Towards Accurate Online Traffic Matrix Estimation in Software-Defined Networks

Yanlei Gong[§]
gyl0511@gmail.com

Xiong Wang[§]
wangxiong@uestc.edu.cn

Mehdi Malboubi[¶]
mmalboubi@ucdavis.edu

Sheng Wang[§]
wsh_keylab@ucdavis.edu

Shizhong Xu[§]
xsz@uestc.edu.cn

Chen-Nee Chuah[¶]
chuah@ucdavis.edu

[§]School of Communication and Information Engineering,
University of Electronic Science and Technology of China, China
[¶]Department of Electrical and Computer Engineering,
University of California, Davis, USA

ABSTRACT

Traffic matrix measurement provides essential information for network design, operation and management. In today's networks, it is challenging to get accurate and timely traffic matrix due to the hard resource constraints of network devices. Recently, Software-Defined Networking (SDN) technique enables customizable traffic measurement, which can provide flexible and fine-grain visibility into network traffic. However, the existing software-defined traffic measurement solutions often suffer from feasibility and scalability issues. In this paper, we seek accurate, feasible and scalable traffic matrix estimation approaches. We propose two strategies, called Maximum Load Rule First (MLRF) and Large Flow First (LFF), to design feasible traffic measurement rules that can be installed in TCAM entries of SDN switches. The statistics of the measurement rules are collected by the controller to estimate fine-grained traffic matrix. Both MLRF and LFF satisfy the flow aggregation constraints (determined by associated routing policies) and have low-complexity. Extensive simulation results on real network and traffic traces reveal that MLRF and LFF can achieve high accuracy of traffic matrix estimation and high probability of heavy hitter detection.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]; C.2.3 [**Network Operations**]: Network monitoring; C.2.4 [**Distributed Systems**]: Network operating systems

Keywords

Software-Defined Networking, Software-Defined Measurement, Traffic Matrix Estimation

1. INTRODUCTION

Traffic Matrix (TM) plays an important role in many network tasks, such as network design [1], traffic engineering [1], traffic accounting [2], and performance diagnosis [3], all of which rely on accurate and timely TMs as critical inputs. Due to the important role of TM, TM measurement has attracted extensive attention from the research community in the past decade [4, 5, 6]. However, it is still challenging to accurately measure TM for practical networks in a timely fashion. First, direct measurement of TM on large networks is infeasible due to the hard constraint of network measurement resources (e.g., TCAM entries, memory capacity and processing power). Second, even though TM can be estimated from side information that can be readily obtained such as SNMP link loads and network routing configuration, the TM estimation problem is typically an under-determined linear-inverse problem where the number of measurements are not sufficient to accurately identify the TM. Thus, in order to improve the estimation accuracy, more related side information must be incorporated into the problem formulation. However, this is hard to achieve due to the control plane limitations of traditional networks.

On the other hand, as a promising architecture for future networks, Software-Defined Networking (SDN) [7] has received a significant attention by both industry and academia. SDN paradigm enables the separation of a logically centralized control plane from the underlying data plane. This decoupling brings unique opportunities for traffic measurement. Most importantly, the centralized control plane provides a global view of network resource and enables programmable traffic measurement. Moreover, the data plane in each switch provides several counters for each flow rule in the flow table. Therefore, the SDN has the potential capability of enabling accurate and dynamic traffic measurement.

Recently, a few research efforts have been made to develop dynamic, accurate and scalable traffic measurement frameworks in SDN paradigm. In [8], ProgME proposes a programmable traffic measurement architecture, which allows to collect traffic statistics of user defined sets of flows. To support various measurement tasks, OpenSketch [9] introduces a variety of hash-based sketches, and can configure the sketches dynamically. However, both ProgME [8] and OpenSketch [9] assume specialized hardware support on switches for traffic measurement. In order to avoid using custom hardware for traffic measurement, [10, 11] propose practical traffic measurement solutions should run on commodity network elements, and [12] extends the work in

SOSR 2015, June 17 - 18, 2015, Santa Clara, CA, USA.
Copyright 2015 ACM. ISBN 978-1-4503-3451-8/15/06...$15.00.
DOI: http://dx.doi.org/10.1145/2774993.2775068.

[10, 11] by enabling concurrent and dynamically instantiated traffic measurement tasks. However, all the solutions proposed in [8, 9, 10, 11, 12] are targeted for measuring a specific set of flows (e.g, Heavy Hitters or Distributed Icebergs), and they are not suitable for TM estimation.

OpenTM [13], DCM [14] and iSTAMP [15] aim to measure TM in SDN. OpenTM and DCM are per-flow based measurement solutions, which directly measure and estimate TM by tracking statistics of each flow. Evidently, OpenTM and DCM are not scalable since the measurement resources (e.g., CPU and TCAM) are limited while the number of flows is large. To meet constraints on the available measurement resources and improve measuring accuracy, iSTAMP infers TM based on both aggregated and the k largest (i.e. the most informative flows) per-flow measurements. iSTAMP seems to make a good tradeoff between the used measuring resources and accuracy, but it also faces the following issues. First, the priority and wildcard based matching strategy used by SDN switches implies that only the flows with a same prefix can be potentially aggregated by one rule, but iSTAMP ignores 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. Most recently, [16] investigates the TM estimation problem in SDN capable data center networks. The feasibility issue of traffic aggregation is considered in [16] based on the assumption that the traffic measurement/aggregation only takes place at the ToR SDN switches of data center networks. The assumption makes the method proposed in [16] hard to apply in general networks. In addition, the complexity of choosing feasible aggregation paths in [16] is also high for large-scale networks.

Based on the shortcomings of existing works, we revisit the TM estimation problem in SDN paradigm, and aim to propose accurate, feasible and scalable traffic measurement strategies in this paper. Here, we say a traffic measurement strategy is *feasible* if it satisfies the measuring resource and flow aggregation constraints. We assume that to save TCAM entries, the rules used for routing flows in each SDN switch are aggregated whenever possible. In theory, the TM can be estimated based on the statistics of these aggregated routing rules. However, to improve the estimation accuracy, we generate traffic measurement rules by deaggregating the aggregated rules, and install the traffic measurement rules in the available TCAM entries of each SDN switch. The controller collects the measurement statistics of TCAM entries periodically, and estimates the per-flow sizes based on these statistics. The main contributions of this paper are summarized as follows:

1) We propose a simple traffic measurement rule generation strategy named Maximum Load Rule First (MLRF) to efficiently generate *feasible* traffic measurement rules.

2) To further improve the TM estimation *accuracy*, we also propose another traffic measurement rule generation strategy named Large Flows First (LFF), which uses the TM estimation results of MLRF as the inputs.

3) We evaluate the performance of MLRF and LFF using traffic traces from real ISP networks. The results verify that MLRF and LFF can achieve *feasible* and *accurate* traffic estimation.

2. THE TRAFFIC MATRIX ESTIMATION IN SDN

2.1 System Model and Assumptions

In this paper, we consider a hybrid SDN network, where only a subset of the nodes are SDN switches while the rest of the nodes are traditional routers. The TM estimation system of SDN contains two parts. In the data plane, the TCAMs in SDN switches match and count packets with wildcard rules. In the control plane, the controller: 1) fetches flow statistics (TCAM counters and SNMP link loads); 2) estimates the TM based on the statistics; 3)designs new measurement rules based on the quality of estimated TM; 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 routing rules. To save TCAM entries, the routing rules are aggregated based on the destination prefixes. To avoid forwarding disruption to network traffic, the routing rules cannot be modified during the traffic measurement process. We assume that the network operators will assign a set of IP prefixes to each node, and this mapping is known a priori. A flow can be 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.

2.2 Problem Formulation

We can model the network as a directed graph $G = (V, L)$, where V and L are the sets of nodes and links, respectively. 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_s and m_s be the total number of TCAM entries and the number of available (i.e. unused or reserved) TCAM entries in SDN node $s(s \in V_{SDN})$, respectively. Let R_s be the set of flow rules of SDN node s $(s \in V_{SDN})$. 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 traffic matrix, where N is the number of flows. Y_S and Y_L have the following relationship with X.

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

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 $(j \leq N)$ is forwarded by rule i $(i \leq \sum_{s \in V_{SDN}} n_s)$, and the element $A_L^{ij} \in \{0, 1\}$ indicates whether flow j $(j \leq N)$ is going through link i $(i \leq |L|)$. A_L is given and it is fixed while A_S is determined by the flow rules designed by the controller to provide the most informative aggregate measurements adhering to the routing policy. Having measurements Y_S and Y_L as well as aggregation matrices A_S and A_L, the traffic matrix 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 [15].

$$\begin{aligned} \hat{X} = & \underset{X}{\text{minimize}} \, \|X\|_1 \\ \text{s.t.} \quad & Y_L = A_L X \\ & Y_S = A_S X \\ & X \geq 0 \end{aligned} \tag{2}$$

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:     $\Delta_{min} \leftarrow \frac{1}{2} \cdot l_{old}$  //$\Delta_{min}$ is an indicator of load balance
        between $r_{new}$ and $r_{old}$, and $\Delta_{min} = 0$ 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 $r_{new}$ respectively,
        and update the sets of flows matching the rules $r_{old}$ and
        $r_{new}$ respectively.
36:   end while
37:   $R \leftarrow R_s \cup R$
38: end for
39: return $R$
```

Considering the optimization formulation (2), we can improve the estimation accuracy by generating better A_S. Since A_S is determined by the measurement rules installed in the SDN switches, we can get a better A_S by installing carefully generated traffic measurement rules on the available TCAM entries, which will provide more inputs to the optimization formulation (2). To this end, we generate some measurement rules by deaggregating the 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. In this paper, we mainly focus on the measurement rule generation strategies.

3. THE FLOW MEASUREMENT RULE GENERATION STRATEGIES

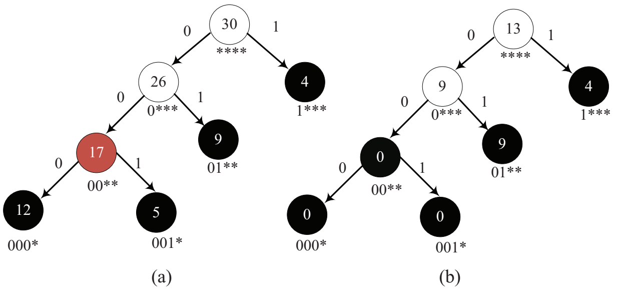

Figure 1: Prefix trie of source IPs.

In practical networks, the rules used for routing are always aggregated to save TCAM entries [17] (e.g., the rules for routing the flows to a same prefix can be aggregate into one rule); consequently flow aggregation measurements are used for TM estimation. However, due to the ill-conditioned and under-determined nature of TM inference problems [18], the direct estimation of TM based on the statistics of those aggregated routing rules may suffer from significant estimation errors. Hence, in order to improve the TM estimation accuracy, we can generate additional rules to measure the traffic under the resource and flow aggregation constraints. In this section, we will present the proposed measurement rule generation strategies called Maximum Load Rule First (MLRF) and Large Flow First (LFF), respectively.

3.1 The Maximum Load Rule First Strategy

For a flow (defined by a source and destination prefixes pair) going through SDN switch i, the controller can easily find out the flow rule matching the flow in SDN switch i by simply checking each rule installed in SDN switch i. Thus, given the set of flows and the routes of the flows, the number flows matching each rule in a SDN switch can be easily computed. Here, we define the load of a rule as the number of flows matching the rule in a SDN switch.

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 a SDN switch in each step. MLRF first greedily selects the rule with the maximum load in a 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 of the new rule are the same as the old rule (lines 7, 8, 21 and 25 in 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). Figure 1 shows an example prefix trie for four bits. The number on each prefix node is the load of the rule if it uses the associated source IP prefix. In this example (Figure 1(a)), MLRF will choose 00** as the source IP prefix for the new rule, and the loads of the new rule and the old rule are 17 and 13 respectively when the new is added into the SDN switch. Figure 1(b) shows the loads of the rules using the associated source IP prefixes on the prefix trie when the new rule is

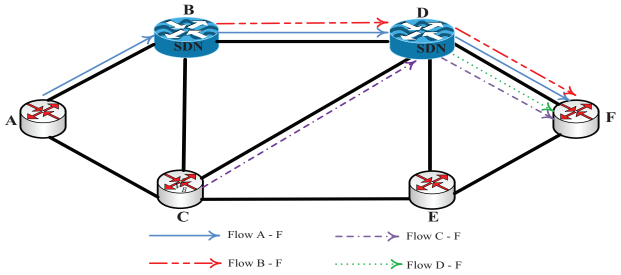

Figure 2: Illustration of flow measurement allocation.

added to the SDN switch.

3.2 The Large Flow First Strategy

It has been shown that in real networks, a small number of large flows may account for more than 80% the traffic volume [8]. Therefore, accurately measuring the large flows can yield the best improvement of overall TM estimation performance. However, how to find out the expected large flows is a problem. To solve this problem, iSTAMP [15] uses a two-phase approach, which measures the sizes of all the flows using available TCAM entries over multiple epochs in the first phase, and selects the k largest flows to measure in the second phase. However, measuring the per-flow sizes is costly and time consuming, especially when the available TCAM entries are limited and the number of flows is large. LFF is also a two-phase approach. But instead of accurately measuring the per-flow sizes, LFF estimates the per-flow sizes based on the statistics of the rules generated by MLRF in the first phase. Although the estimated per-flow sizes may not accurate, they are sufficiently informative for us to find out the real 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.

In hybrid 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 impact on the measurement results. Let us consider the example in Figure 2. There are four flows: $A-F$, $B-F$, $C-F$, and $D-F$. The routes of the flows are indicated by dotted 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, LFF needs to get an optimal solution of FMA. 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 $\forall v_j \in V_{SDN}$, $\sum_{f_i \in F} \psi_{f_i}^{v_j} \leq m_{v_j}$, where m_{v_j} is the number of available TCAM entries in SDN switch v_j.

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

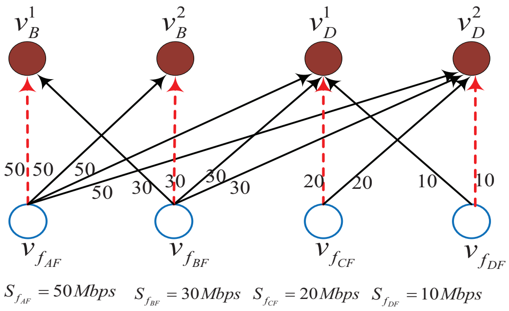

Figure 3: The auxiliary bipartite graph and a maximum weight matching denoted by red dashed lines.

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

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

where S_{f_i} is the size of flow 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, $f(\Psi^*) \geq f(\Psi)$.

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 $(j \leq m_s)$. The weight of the link (v_{f_i}, v_s^j) is set to the estimated size of flow f_i (denoted as S_{f_i}). The auxiliary bipartite graph of the example in Figure 2 is illustrated in Figure 3.

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

Proof. See [19] □

Based on the discussions above, we design the LFF traffic measurement strategy for SDN networks as in Algorithm 2. Since a maximum weight matching of the constructed auxiliary bipartite graph is an optimal flow measurement allocation solution, LFF will generate a rule for each flow (lines 9-15) that corresponds to a link of the maximum weight matching. Accordingly, the generated flow rules are installed in SDN switches and the flow statistics are used to accurately estimate the TM using network inference framework (2). The red dashed lines in Figure 3 denote a maximum weight matching of the auxiliary bipartite graph. In the example,

Algorithm 2 The Large Flow First Measurement Rule Generation Strategy

```
Input: Network topology $G(V, L)$.
Output: The rule sets $R$ for the SDN switches.
$R \leftarrow \emptyset$
estimate the flow sizes based on the statistics of the rules generated by MLRF strategy (Algorithm 1)
sort the flows according to their estimated sizes in decreasing order
for each node $s \in V_{SDN}$ do
    add the routing rules in node $s$ to $R_s$
end for
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
    $r_{old} \leftarrow$ the rule matching flow $f_i$ in set $R_s$
    $r_{new} \leftarrow r_{old}$
    $r_{new}.priority \leftarrow r_{new}.priority + 1$
    $r_{new}.src\_prefix \leftarrow f_i.src\_prefix$   //$f_i.src\_prefix$ denotes the source prefix of flow $f_i$
    $R_s \leftarrow R_s \cup r_{new}$
end for
for each node $s \in V_{SDN}$ do
    $R \leftarrow R \cup R_s$
end for
return $R$
```

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}.

4. PERFORMANCE EVALUATION

4.1 Simulation Setup

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

$$S_{f_i} = S_{af_{sd}} \cdot \frac{len(f_i.src\_prefix)}{\sum\limits_{pref \in P_s} len(pref)} \cdot \frac{len(f_i.dst\_prefix)}{\sum\limits_{pref \in P_d} len(pref)}, \quad (3)$$

where $S_{af_{sd}}$ 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 P_s and P_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 ratio between the number of total TCAM entries and the number of flows, i.e., $r = n \cdot \frac{|V_{SDN}|}{N}$.

Performance Metrics: The metrics used in our performance evaluation are defined in equation (4). P^k_{large} is the average probability of accurately finding out the k largest flows by using the measurement rules generated by MLRF. From the presentation in Section 3.2, we know that the performance of LFF is closely related to P^k_{large}. In equation (4), $I(\cdot)$ returns the indices of flows, sorted in descending order of the flow sizes. NMAE is widely used performance metric for measuring the accuracy of traffic matrix estimation.

$$\begin{aligned} P^k_{large} &= \frac{1}{M}\sum_{i=1}^{M} pr(I(\hat{X^i}) \leq k | I(X^i) \leq k) \\ NMAE &= \frac{1}{M}\sum_{i=1}^{M} \frac{|X^i - \hat{X}^i|}{|X^i|} \end{aligned} \quad (4)$$

4.2 Simulation Results

To yield the best improvement of overall estimation performance, LFF chooses the k largest flows to measure directly. The key challenge here is that the per-flow sizes are unknown. LFF solves the problem by estimating the per-flow sizes based on the statistics of the rules generated by MLRF. So the probability (P^k_{large}) of accurately finding out the k largest flows by using estimated per-flow sizes is critical for LFF. Figure 4 shows P^k_{large} under different k in both Geant and Abilene topologies. From Figure 4, we can see that P^k_{large} increases with the flow aggregation ratio r. Because higher r means that more TCAM entries can be used for traffic measurement, and thus the traffic size estimation accuracy can be improved. Moreover, we can observe that even when the r is low (e.g., 10%), the majority of large flows (more than 75%) included in the k largest flow set can also be found out successfully. It is demonstrated that using the estimated per-flow sizes based on the statistics of the rules generated by MLRF, we can also achieve sufficiently accurate inputs for LLF, and thus we do not need to directly measure every per-flow size in the first phase of LFF.

Figure 5 compares the NMAE of MRLF, LLF, iSTAMP (with BAT) and iSTAMP (with EAT) [15], where EAT (Exponential Aggregation Technique) and BAT (Block Aggregation Technique) are two different aggregation matrix design strategies used in iSTAMP. 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 [15]. It is notable that both BAT and EAT do not consider the flow aggregation feasibility. So though the aggregation matrix generated by EAT and BAT are good for traffic matrix estimation, it may not be feasible in practice. From Figure 5, we can observe that as expected, the NMAE of the four methods decreases with the increasing of the flow aggregation ratio r. Most importantly, we also can observe that the NMAE of our proposed LFF is very close to that of iSATMP+BAT (the differences are within 0.1) and iSTAMP+EAT, and in Abilene, the NMAE of LFF is even much better than that of iSATMP+EAT. Our results demonstrate that LFF can generate feasible traffic

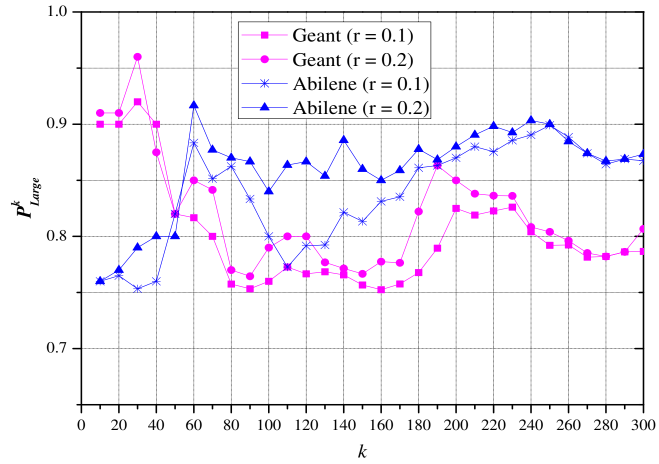

Figure 4: P^k_{large} of MLRF under different k in Geant and Abilene topologies.

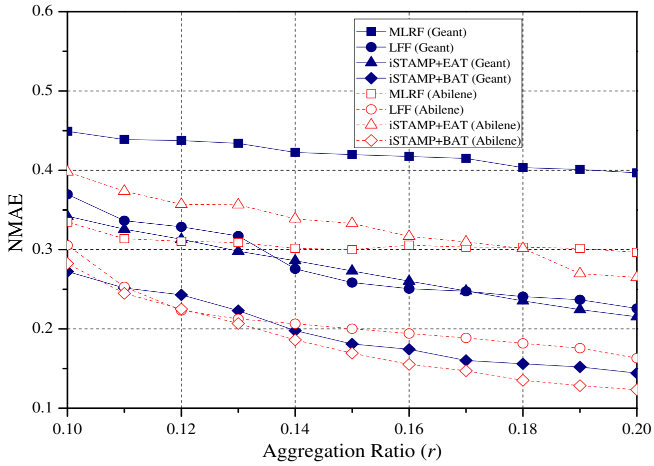

Figure 5: NMAE in Geant and Abilene topologies when r varies.

measurement rules that can achieve high traffic matrix estimation accuracy. Comparing with LFF, MRLF has higher NMAE. However, MRLF is a simple algorithm with low-computational complexity and it can provide useful information for the LFF (as shown in Figure 4). So MRLF is also a meaningful approach for HH detection and providing the outline of the traffic matrix.

Figure 6 shows the NMAE of different methods when the number of SDN switches varies in Geant. Since the capacity of TCAM is very limited, the flow aggregation 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 is set as 75. Under this setting, the flow aggregation ratio of Geant network is about 15% when 50% of the nodes are SDN-capable. In Figure 6, the NMAE of all the methods decreases quickly with the increasing number of deployed SDN switches. When 50% of the nodes are SDN-capable (the flow aggregation ratio is about 15%), the NMAE of LFF is about 0.1 for Geant. This demonstrates that even if a small number of SDN switches are deployed in the network, the traffic matrix estimation accuracy can be significantly improved. We obtain similar results in Abilene network, which which is included in our technical report [19].

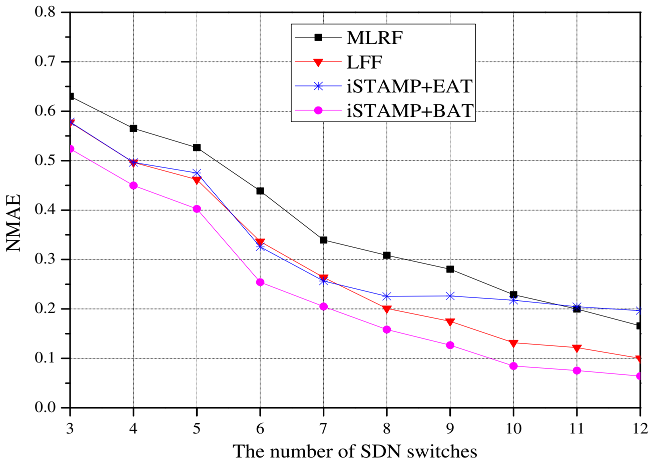

Figure 6: NMAE in Geant topology when the number of SDN switches varies.

In addition, MLRF and LFF can also be used for Heavy Hitter (HH) detection. To evaluate the effectiveness of using MLRF and LFF for HH detection, we also compute the average probability of detection (P^d_{HH}) [15] of MLRF and LFF. The results indicate that both MRLF and LLF can achieve very high probability of detection even when the aggregation ratio r is low (e.g., $r = 0.1$). Detailed results can be found in our technical report [19].

5. CONCLUSIONS

In this paper, we leverage the re-configuration capability and flexible flow rules of SDN to enhance the accuracy of traffic matrix estimation. In SDN, the accuracy and feasibility of traffic measurement are closely related to the flow rules installed in SDN switches. To achieve feasible and accurate traffic matrix estimation, we proposed two traffic measurement rule generation strategies, named MLRF and LFF. MLRF and LFF generates traffic measurement rule by de-aggregating the aggregated routing rules. The flow aggregation feasibility is guaranteed in MLRF and LFF, and the complexity of MLRF and LFF is also low. Finally, we have conducted extensive performance evaluation on real networks and traffic traces; the results have confirmed that MRLF and LFF can achieve feasible and accurate traffic matrix estimation.

6. ACKNOWLEDGMENTS

We would also like to thank the anonymous reviewers for their valuable comments. This work is partially supported by NSF CNS-1321115, 973 Program (2013CB329103), NSFC Fund (61301153, 61271165, 61201128), Program for Changjiang Scholars and Innovative Research Team (PCSIRT) in University.

7. REFERENCES

[1] M. Pioro and D. Medhi. *Routing, Flow, And Capacity Design in Communication And Computer Networks.* AMorgan Kaufmann, San Francisco, CA, 2004.

[2] C. Estan and G.Varghese. New directions in traffic measurement and accounting. *SIGCOMM Computer Communication Review*, 32(4):323–336, 2002.

[3] A. Curtis, J. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee. Devoflow: scaling flow management for high-performance networks. In *SIGCOMM*, 2011.

[4] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads. In *SIGMETRICS*, 2003.

[5] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot. Traffic matrices: balancing measurements, inference and modeling. In *SIGMETRICS*, 2005.

[6] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot. Spatio-temporal compressive sensing and internet traffic matrices. *IEEE/ACM Transactions on Networking*, 20:662–676, 2012.

[7] 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*, 38(2):69–74, 2008.

[8] L. Yuan, C. Chuah, and P. Mohapatra. Progme: towards programmable network measurement. *IEEE/ACM Transactions on Networking*, 19(1):115–128, 2011.

[9] M. Yu, L. Jose, and R. Miao. Software defined traffic measurement with opensketch. In *NSDI*, 2013.

[10] M. Moshref, M. Yu, and R. Govindan. Resource/accuracy tradeoffs in software-defined measurement. In *HotSDN*, 2013.

[11] L. Jose, M. Yu, and J. Rexford. Online measurement of large traffic aggregates on commodity switches. In *ACM-Hot-ICE*, 2011.

[12] M. Moshref, M. Yu, R. Govindan, and A. Vahdat. Dream: dynamic resource allocation for software-defined measurement. In *SIGCOMM*, 2014.

[13] A. Tootoonchian, M. Ghobadi, and Y. Ganjali. Opentm: traffic matrix estimator for openflow networks. In *PAM*, 2010.

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

[15] M. Malboubi, L. Wang, C. Chuah, and P. Sharma. Intelligent sdn based traffic (de)aggregation and measurement paradigm (istamp). In *INFOCOM*, 2014.

[16] Zhiming Hu and Jun Luo. Cracking network monitoring in dcns with sdn. In *INFOCOM*, 2015.

[17] X. Zhao, Y. Liu, L. Wang, and B. Zhang. On the aggregatability of router forwarding tables. In *INFOCOM*, 2010.

[18] M. Malboubi, C. Vu, C-N. Chuah, and P. Sharma. Decentralizing network inference problems with multiple-description eusion estimation (mdfe). In *INFOCOM*, 2013.

[19] Y. Gong, X. Wang, M. Mehdi, S. Wang, S. Xu, and C. Chuah. Accurate realization of online traffic matrix measurement and estimation in software-defined networking. *Technical Report ECE-CE-2015-1, UC Davis*, March, 2015, http://web.ece.ucdavis.edu/cerl/techreports/2015-1/.