UC Davis
UC Davis Previously Published Works

Title

Intelligent SDN based traffic (de)Aggregation and Measurement Paradigm (iSTAMP)

Permalink

https://escholarship.org/uc/item/5mj2z64v

Authors

Malboubi, M
Wang, L
Chuah, CN
et al.

Peer reviewed

Intelligent SDN based Traffic (de)Aggregation and Measurement Paradigm (*i*STAMP)

Mehdi Malboubi, Liyuan Wang, Chen-Nee Chuah
Dept. of Electrical & Computer Engineering
University of California, Davis, USA.
{mmalboubi, antwang, chuah}@ucdavis.edu

Puneet Sharma
Hewlet-Packard Laboratories (HP Labs)
Palo Alto, California, USA.
puneet.sharma@hp.com

Abstract—Fine-grained traffic flow measurement, which provides useful information for network management tasks and security analysis, can be challenging to obtain due to monitoring resource constraints. The alternate approach of inferring flow statistics from partial measurement data has to be robust against dynamic temporal/spatial fluctuations of network traffic. In this paper, we propose an intelligent Traffic (de)Aggregation and Measurement Paradigm (iSTAMP), which partitions TCAM entries of switches/routers into two parts to: 1) optimally aggregate part of incoming flows for aggregate measurements, and 2) de-aggregate and directly measure the most informative flows for per-flow measurements. iSTAMP then processes these aggregate and per-flow measurements to effectively estimate network flows using a variety of optimization techniques. With the advent of Software-Defined-Networking (SDN), such real-time rule (re)configuration can be achieved via OpenFlow or other similar SDN APIs. We first show how to design the optimal aggregation matrix for minimizing the flow-size estimation error. Moreover, we propose a method for designing an efficient-compressive flow aggregation matrix under hard resource constraints of limited TCAM sizes. In addition, we propose an intelligent Multi-Armed Bandit based algorithm to adaptively sample the most "rewarding" flows, whose accurate measurements have the highest impact on the overall flow measurement and estimation performance. We evaluate the performance of iSTAMP using real traffic traces from a variety of network environments and by considering two applications: traffic matrix estimation and heavy hitter detection. Also, we have implemented a prototype of iSTAMP and demonstrated its feasibility and effectiveness in Mininet environment.

I. INTRODUCTION

Fine-grained traffic flow measurements provide essential information that is central to network design, operation, management, accounting and security. Direct flow-based measurements such as NetFlow [1] and sFlow [2] offer fine-grained measurements that can support different measurement tasks. However, such an approach not only requires dedicated hardware and specialized algorithms, but it is also often challenging, inefficient or even infeasible to monitor each and every flow due to exploding traffic volume and limited monitoring resources (e.g., the number of Ternary Content Addressable Memory (TCAM) entries, storage capacity and processing power). Therefore, intelligent sampling and streaming algorithms have been proposed to estimate statistics or answer specific queries of, e.g., flow size distributions [3] and approximate size of elephant flows [4]. These solutions are task-specific and lack the full flexibility to dynamically choose which traffic sub-population to measure, and how, depending on application requirements.

An alternate approach is estimating the internal attributes of interest in a network (e.g. traffic intensity between nodes) based on a limited set of measurements using network inference/tomography methods. However, most Network Inference (NI) problems are naturally formulated as ill-posed Under-Determined Linear Inverse (UDLI) problems where the number of measurements are not sufficient to uniquely and accurately determine the solution. Hence, side information from different sources must be incorporated into the problem formulation to improve the estimation precision [5].

On the other hand, many recent network monitoring and security applications require timely estimates of both large and small traffic flows with high precisions [6], [7]. The flows of interest can be sparse, or highly fluctuating over time/space. Hence, the network measurement infrastructure must be agile enough to cope with the dynamic network and traffic conditions. Such a flexible architecture can be achieved due to the recent advent of Software-Defined-Networking (SDN). In fact, SDN enabler, such as OpenFlow, nicely separates the measurement data plane and control plane functions, and provides a capability to control/re-program the internal configurations of switches in dynamic environments. Consequently, SDN allows for more complex network monitoring and management applications [8], [9].

In this paper, we propose an intelligent SDN based Traffic (de)Aggregation and Measurement Paradigm (iSTAMP), in which the measurement modules can be configured on-the-fly to collect fine-grained measurements of specific traffic sub-populations of interest that directly reflect the monitoring application requirements. Based on the philosophy that not all attribute of interests are equally important, iSTAMP utilizes an intelligent sampling algorithm to select the most informative traffic flows, using information gathered throughout the measurement process. It also exploits Compressive Sensing (CS) inference methods that are effective for estimating highly fluctuated sparse unknown quantities from a set of well-defined compressed linear measurements [10], [11].

iSTAMP leverages OpenFlow to dynamically partition the TCAM entries of a switch/router into two parts. In the first part, a set of incoming flows are optimally aggregated to provide well-compressed aggregated flow measurements that can lead to the best estimation accuracy via network inference process. The second portion of TCAM entries are dedicated to track/measure the most *rewarding* flows (defined as flows with the highest impact on the ultimate monitoring application performance) to provide accurate per-flow measurements.

These flows are selected and "stamped" as important (or *rewarding* from monitoring's perspective) using an intelligent Multi-Armed Bandit (MAB) based algorithm. These two sets of measurements (aggregated and sampled flows) are then jointly processed to estimate the size of all network flows using different optimization techniques.

A. Related Work

There is a rich literature on improving the accuracy of traffic flow measurements and estimation. We will briefly discuss the most relevant work here. In [12], ProgMe proposes a re-programmable architecture allowing statistics collection based on the notion of flowsets (arbitrary traffic sub-populations) using a flexible flowset composition language. iSTAMP, in contrast, leverages off-the-shelf components and existing OpenFlow API to configure the TCAM entries and associated counting rules. In particular, it leverages "statistics" field in OpenFlow to collect traffic measurements. Hence, it does not impact the packet forwarding behavior or performance. In addition, from SDN perspective, [8] describes a reconfigurable measurement architecture for hierarchical heavy hitter detection, and then, [9] proposes a re-programmable structure (called OpenSketch) where a variety of sketches for different measurement tasks can be defined and installed by the operator. Furthermore, Duffield [13] develops a theory of fairly distributing a sampling budget for estimating sub-populations flow records. However, SDN-based fine-grained traffic flow measurement under monitoring resource constraints has not been addressed previously. iSTAMP leverages the flexibility provided by SDN to push the boundary of obtaining accurate flow measurements/estimation by combining both coarse (aggregate) and fine-granularity (per-flow) measurements. We build upon the recent achievements in the theory of compressed sensing and on-line learning to develop a theoretical foundation for guiding the design of such an adaptive flow measurement framework under hard resource constraints.

B. Our contributions

iSTAMP is simple, generic, and efficient with the ability to intelligently track and measure the most "rewarding" flows, and optimally aggregate others. iSTAMP adaptively allocate(re-allocate) TCAM entries amongst the constantly evolving flows to achieve the highest flow estimation accuracy. Therefore, it is robust in dynamic environments (with highly fluctuating flows over time/space) and under hard resource constraints where, TCAM entries, storage capacity, and network bandwidth are severely limited. In fact, iSTAMP can be easily deployed on commodity OpenFlow-enabled routers/switches to enhance the performance of various network monitoring applications, including Traffic Matrix (TM) estimation, Traffic Engineering (TE), Heavy Hitter Identification (HHI) and different security applications [7]. It also offers timely estimates of network flows with low computation and communication overhead between control and data planes. Furthermore, it significantly reduces the required storage for monitoring tasks.

Our main contributions are summarized as follow:
• In the context of compressed sensing, we, for the first time, formulate and solve the problem of designing an optimal binary aggregation (or observation) matrix to maximize the estimation accuracy while providing a compressed form of measurements. Moreover, we propose a method for designing an efficient-compressive flow aggregation matrix under hard resource constraints of limited TCAM sizes.
• We propose a simple and efficient MAB based algorithm to adaptively track and measure the most *rewarding* flows.
• We evaluate the performance of iSTAMP using real traffic traces from a variety of network environments and by considering two applications: Origin-Destination Flow (ODF) estimation and heavy hitter identification. Furthermore, we implement a prototype of iSTAMP and demonstrate its feasibility and effectiveness in Mininet environment.

The rest of this paper is organized as follows. Section II provides an overview of iSTAMP and different network inference techniques that we have used. Section III describes our optimal aggregation matrix design procedure along with our intelligent MAB based flow sampling algorithm. In Section IV, we evaluate the performance of iSTAMP considering two main applications: ODF estimation and heavy hitter detection. Section V summarizes our most important results.

II. System Description and Problem Statement

Figure 1 shows the general block diagram of the iSTAMP framework where the TCAM at the switch or router is partitioned into two parts for (i) aggregate measurements, and (ii) per-flow monitoring of selected flows, respectively. The optimal aggregate statistics from the first part provide a set of well-formed compressed linear measurements as input to a network inference process to perform flow size estimation. The second part of TCAM is allocated for tracking individual flows that are deemed "important" (or rewarding) for the monitoring application in question. This process is called intelligent sampling. For illustrative purpose, we consider Origin-Destination Flow (ODF) size estimation, and more generally, traffic matrix (as a measure of origin-destination traffic intensity between nodes) estimation, as the driving application in this section. In this context, ODFs with the largest volume will be the 'important' candidates for per-flow monitoring. Note that a single router may only observe a subset of ODFs (or partial TM) in the network. We will first discuss how iSTAMP can be deployed on a single router/switch to estimate the ODFs it observes. The framework can then be extended to a distributed monitoring case, where multiple routers running iSTAMP can coordinate to estimate the complete TM for the network.

Assume there are n ODFs in the network and T entries of TCAM at the switch which can be used for network measurement where $n >> T$. At each measurement interval τ, K out of T entries are used to track and measure the most rewarding ODFs and m entries are used to optimally aggregate other ODFs (i.e. $T = m + K$). For this purpose, the flexibility of OpenFlow is used to install TCAM wildcard matching rules (prefix keys) c_i and collect associated statistical counts y_i in each measurement interval τ_t. At the controller, which can be co-located on the switch or reside on a separate machine, the measurement statistics/counts are processed. In the next epoch τ_{t+1}, the best ODFs (determined based on the application requirement) for direct measurement are selected and their corresponding prefix keys are installed in K TCAM entries. This process is called de-aggregation and the TCAM lookup mechanism, which is based on highest wildcard matching rule, facilitates the implementation of this process. In addition, m
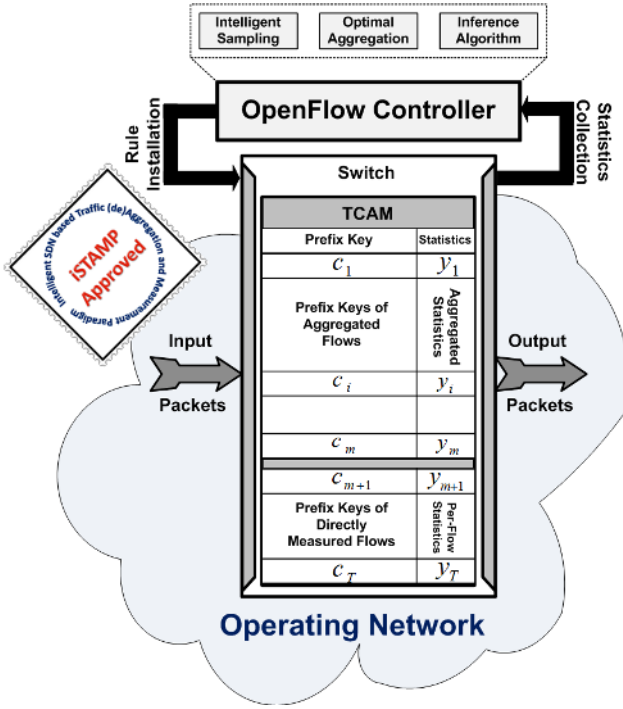
Fig. 1: iSTAMP network measurement framework: a general perspective.

entries of TCAM is used for optimal aggregation of $n - K$ flows. The controller can poll the statistic counts periodically or in different measurement intervals, the frequency of which is limited by practical switch/network constraints.

At each epoch t, the set of measurement statistics Y^t is represented by the under-determined linear system of equation shown in Eq.(1) where Y_g^t denotes the aggregated measurements and Y_k^t denotes the direct per-flow measurements. In this equation, A^t is an $(T \times n)$ measurement matrix consisting of an $(m \times n)$ binary aggregation matrix A_g^t, where non-zero entries in each row of A_g^t represents the set of flows aggregated (or mapped) to that entry, and an $(k \times n)$ binary matrix A_k^t, where the only one non-zero entry in each row demonstrates the selected flow for direct measurement in t^{th} measurement interval τ_t. Also, $X^t (= X_g^t \bigcup X_k^t)$ denotes the tth vector of flows where X_g^t is the set of ODFs that are being mapped to m groups for aggregate measurements and X_k^t is the set of directly measured ODFs (i.e. $X_k = Y_k$). Having these measurements, the set of unknown flows X^t can be estimated using the following general optimization formulations Eq.(2) and Eq.(3), where parameters (p, q, λ) are defined based on different optimization techniques. The particular choice $p = 2$, $q = 1$ and $\lambda = \lambda_0 (\in \mathbb{R})$ define a compressed sensing inference technique [10] that is effective for estimating highly fluctuating network flows. Furthermore, Eq.(3) incorporates side information Y_S into the optimization problem to improve the performance of the overall TM estimation process (in the extended multi-point/distributed monitoring case). Here, side information Y_S^t is the SNMP link load measurements $Y_S^t = H X_g^t$ where H is the routing matrix.

$$Y^t = \begin{bmatrix} Y_g^t \\ Y_k^t \end{bmatrix} = A^t X^t \text{ where } A^t = \begin{bmatrix} A_g^t \\ A_k^t \end{bmatrix} \quad (1)$$

$$\hat{X} = \underset{X}{\text{minimize}} \left\| Y_g^t - A_g^t X_g^t \right\|_p^2 + \lambda \left\| X_g^t \right\|_q \quad (2)$$
$$\text{s.t.} \quad X_k^t = A_k^t X_k^t, \quad X_g^t \geq 0$$

$$\hat{X} = \underset{X}{\text{minimize}} \left\| Y_S^t - H X_g^t \right\|_p^2 + \lambda \left\| X_g^t \right\|_q \quad (3)$$
$$\text{s.t.} \quad Y_g^t = A_g^t X_g^t, \quad X_k^t = A_k^t X_k^t, \quad X_g^t \geq 0$$

Considering the above optimization formulations, iSTAMP can be used in different single-point and multi-point measurement scenarios. Accordingly, the vector of ODFs X^t represents different notion of flows at different levels. For example, in single-point scenario, an ODF (defined as a sequence of packets that share the same source/destination IP addresses and that are observed within a given time interval at an observation point in the network) can be traffic between servers in different racks in a data center. In the multi-point scenario, ODFs can be traffic between routers in an ISP network. Under single point measurement, we consider two sub-cases: (a) when OpenFlow Switch (OFS) is used for both routing and measurement, and hence the aggregation matrix A_g cannot be arbitrary in order to preserve routing of the aggregate flows, and (b) the OpenFlow switch is used primarily for flow measurement, and routing does not have to be preserved, as in the case of random or hash-based routing. These two cases are compatible with both formulations Eq.(2) and Eq.(3). In the former case A_g^t is the local routing matrix and part of the TCAM entries is used to directly measure a set of the most informative flows. Note that reserved-empty TCAM entries are always available at switches [14]. In the later case, all TCAM entries of the OFS are used for flow measurement; accordingly, along with directly measured flows, A_g^t can be optimally designed to enhance the accuracy of estimation. Likewise, there are two possible cases in multi-point measurement scenario where installation rules are transmitted from Central Controller (CC) to local OFSs and their measurements are transmitted to the CC. If all TCAM entries of OFSs are used for measurement, then both Eq.(2) and Eq.(3) are applicable and A_g^t can be optimally designed. However, if local OFSs are used for routing and measurement, then only Eq.(2) can be applied because, in this case, $A_g^t \cong H$.

In iSTAMP, the communication overhead between the controller and switch is low because only a few flows are directly measured and other entires are used to aggregate a large number of flows (hence, T instead of n measurement records) which is important under hard resource constraints and in multi-point measurement scenario. iSTAMP is also computationally efficient because it exploits existing low-complexity network inference techniques [10].

III. OPTIMAL FLOW AGGREGATION AND SAMPLING

In this section, we explain how to design the optimal aggregation matrix A_g^t and sample the most rewarding flows, via direct(/per-flow) sampling matrix A_k^t to maximize the performance within the iSTAMP framework. We also discuss the various challenges involved and how we address them.

A. Optimal Compressive Aggregation

To cope with the under-determined nature of Eq.(1) and the fact that today's network traffics are highly fluctuating over time and/or space, compressive sensing inference methods can be effectively applied to estimate the sparse characteristics of network flows. For example, it has been shown that if the

coherency of the observation matrix A_g, defined as μ in Eq.(4), is sufficiently small, then the convex optimization program Eq.(5), can exactly recover a sparse signal with l non-zero quantities (in some basis) from $m = O(l \log(n/l))$ properly designed linear observations [10]. Interestingly, it has been shown that random observation matrices with i.i.d. Gaussian or random ± 1 entries, and sufficient number of rows can achieve small coherence with overwhelmingly high probability. Also, there are efficient algorithms with low computational complexity, such as Orthogonal Matching Pursuit (OMP), for recovery of sparse signals [10]. In the following, for simplicity we drop the notion t and assume A_g is written as $A_{g_{(m \times n)}} := [a_1, ..., a_n] \in \mathbb{R}^{m \times n}$.

$$\mu = \max_{i \neq j} \frac{|a_i^T a_j|}{\|a_i\|_2 \|a_j\|_2} \qquad (4)$$

$$\hat{X} = \min_X \|X\|_1 \quad \text{s.t.} \quad Y_g = A_g X \qquad (5)$$

Studies, in [15], [16], demonstrated that a carefully designed observation matrix can achieve sufficient small coherency to improve the performance and efficiency of CS inference methods. In accordance to this, continues optimization techniques have been used to design the optimal observation matrix by minimizing all off-diagonal elements of the corresponding Gram matrix $G = A_g^T A_g$, that is, by minimizing $\left\| A_g^T A_g - I \right\|_F^2$. The motivation is that minimizing the sum of the squared inner products of all columns(/atoms) of A_g results in an observation matrix with more orthogonal columns which may improve the performance of the CS recovery process [16].

Here, we use the same formulation for designing the optimal flow aggregation (i.e. observation) matrix. However, network switches/routers typically classify the incoming packets (i.e. forward, measure or drop) based on the longest prefix match, i.e., each flow is matched to at most one TCAM entry depending on how the wildcard fields are configured. As a result, our aggregation matrix is typically a binary matrix. Therefore, we introduce the following generic integer optimization program Eq.(6) to design the optimal aggregation matrix where the first constraint emphasizes that A_g is a binary matrix. The second constraint controls the redundancy between aggregated measurements where redundancy is achieved by observing a flow at multiple TCAM entries. The non-zero entries of diagonal matrix ϕ is chosen properly to force the optimization engine to focus on minimizing the sum of off diagonal elements of the objective function $\left\| A_g^T A_g - \phi \right\|_F^2$, and to control the number of ones's in each column.Note that, C_l and C_u are important parameters that are constrained by TCAM lookup limitations. Also, the third constraint represents the feasibility of aggregation process where $\mathcal{F}_i$ denotes the set of candidate flows that can be aggregated at i^{th} TCAM entry (e.g. to preserve routing); this set is known as the ith set of feasible aggregated flows. Note that this constraint can be relaxed in the case of random or hash-based routing, where arbitrary subset of flows can be aggregated into the same TCAM entry. The fourth constraint controls the number of aggregated flows per TCAM entry.

Assuming a_{ij}'s are binary variables, we have shown that such a complex, non-convex and non-linear objective function in Eq.(6) can be reformulated as Eq.(7) that is still a non-linear integer program. Thus, we effectively converted Eq.(7)

to a linear integer optimization program introduced in Eq.(8) that can be solved using standard integer optimization tools, such as CPLEX, to design the optimal aggregation matrix. In Eq.(8), C_j and R_i denote the set of z_j and z_i where corresponding a_{ij} satisfies ② and ④ in Eq.(6), respectively. Note that, if $C_l = C_u = 1$, then the objective function in Eq.(8) is simplified by reducing K_u to $mn(n-1)$ and removing $\sum_{i=1}^{K_v} v_i$; in this case $\phi = I$. It should be noted that, our optimization framework in Eq.(8) can also be used for the design of optimal binary observation matrices which are of particular importance in other network monitoring and sensor networks applications where the entries of A indicates the ability($a_{ij} = 1$)/inability($a_{ij} = 0$) of measuring the jth attribute of interest (x_j) at i^{th} observation and redundant measurements ($C_l > 1$) are permitted.

$$\underset{A_g}{\text{minimize}} \left\| A_g^T A_g - \phi \right\|_F^2 = \underset{A_g}{\text{minimize}} \ Tr[(A_g^T A_g - \phi)^2] \qquad (6)$$

$s.t.$

① : $\{a_{ij}\}_{(i,j)=(1,1)}^{(m,n)} \in \{0,1\}$, ② : $C_l \leq \sum_{i=1}^{m} a_{ij} \leq C_u \ \forall j = 1:n$

③ : $\{a_{ij}\}_{j=1}^n \in \mathcal{F}_i \ \forall i = 1:m$, ④ : $M_l \leq \sum_{j=1}^{n} a_{ij} \leq M_u \ \ \forall i = 1:m$

$$\underset{A_g}{\text{minimize}} \left\| A_g^T A_g - \phi \right\|_F^2 = \underset{A_g}{\text{minimize}} \{ \sum_{i=1}^{m} \sum_{j=1}^{n} (1 - 2\phi_{jj}) a_{ij} + \qquad (7)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1,k \neq j}^{n} a_{ij} a_{ik} + \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1,k \neq i}^{m} a_{ij} a_{kj} +$$

$$\sum_{i=1}^{m} \sum_{j=1,j \neq i}^{m} \sum_{k=1}^{n} \sum_{l=1,l \neq k}^{n} a_{ik} a_{il} a_{jk} a_{jl} + \sum_{j=1}^{n} \phi_{jj}^2 \}$$

$$\underset{Z,U,V}{\text{minimize}} \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} (1 - 2\phi_{jj}) z_{(i-1)n+j} + \sum_{j=1}^{n} \phi_{jj}^2 + \sum_{i=1}^{K_u} u_i + \sum_{i=1}^{K_v} v_i \right\} \qquad (8)$$

where $A^T(:) = Z = [z_1, ..., z_{mn}], U = [u_1, ..., u_{K_u}], V = [v_1, ..., v_{K_v}]$, $K_u = mn(n-1) + nm(m-1)$ and $K_v = m(m-1)n(n-1)$
$s.t.$

a) for each multiplication $z_j z_k$: $u_i \leq z_j, u_i \leq z_k, u_i \geq 0$,
$u_i \geq z_j + z_k - 1$

b) for each multiplication $z_j z_k z_l z_m$: $v_i \leq z_j, v_i \leq z_k, v_i \leq z_l, v_i \leq z_m$,
$v_i \geq 0, v_i \geq z_j + z_k + z_l + z_m - 3$

c) $C_l \leq \sum_{j \in C_j} z_j \leq C_u$, d) $M_l \leq \sum_{i \in R_i} z_i \leq M_u$

e) $z_i \in \{0,1\}$, and f) $z_i \in$ corresponding $\mathcal{F}_i$

B. Illustrative Example

To evaluate the performance of this optimization technique we describe two examples. First, in our formulation in Eq.(8), we set $m = 4$, $n = 6$ and $C_l = C_u = 1$ (i.e. each flow can only be mapped to one TCAM entry and redundant measurements are not permitted) to compute the optimal aggregation matrix $A_{g_1}^{Opt} = [1,1,0,0,0,0;0,0,1,1,0,0;0,0,0,0,1,0;0,0,0,0,0,1;]$ where its coherency is $\mu = 1$. Then, using the same set up, we initialize $C_l = C_u = 2$ (i.e. each flow can be mapped to more than one TCAM entry and redundant measurements are permitted)

and compute the optimal aggregation matrix $A_{g_2}^{Opt} = [0, 1, 1, 1, 0, 0; 0, 0, 0, 1, 1, 1; 1, 0, 1, 0, 0, 1; 1, 1, 0, 0, 1, 0]$ where its coherency is $\mu = 0.5$. To measure the real performances of these two aggregation matrices, we ran Monte-Carlo simulations on randomly selected real traffic traces X_g, from Geant network (Table I), to compute the measurement vectors $Y_{g_1} = A_{g_1}^{Opt} X_g$ and $Y_{g_2} = A_{g_2}^{Opt} X_g$. Next, we use the compressive sensing inference technique (i.e. Eq.(2) with (p, q, λ)=(2, 1, 0.01) and without considering direct measurements) to estimate $\hat{X}_g$ and compute Normalized-Mean-Square-Error (Eq.(10)). Accordingly, $NMSE$ is computed as $NMSE_1 = 0.4547$ and $NMSE_2 = 0.1416$ which shows that our optimization formulation in Eq.(8) is successful in generating optimal binary aggregation matrices with low coherency that has important affect on the estimation accuracy.

C. Challenges in Optimal Aggregation Matrix Design and Solutions

The wildcard matching rule in current TCAM lookup technology does not allow a particular flow to be mapped into more than one TCAM entry, that is, $C_l = C_u = 1$. This imposes hard constraints on our optimization program Eq.(8), which otherwise could have benefited from redundant measurements when there can be multiple ones in each column and the accuracy of the estimation can be improved by reducing the coherency of A_g.

To cope with this practical constraint, two solutions are proposed. First, we make use of auxiliary data, such as SNMP link counts, as redundant measurements (observations) of these flows. In fact, we observed that network routing matrices have small average coherency, for example, $\bar{\mu}_{Abilene} = 0.10$ and $\bar{\mu}_{Geant} = 0.04$ (see Table I). Therefore SNMP link counts, that are easily provided, can be incorporated into our flow estimation formulation where feasible optimal aggregated statistics can be used as side information to improve the accuracy of the inference algorithm. Accordingly, having routing matrix H, the SNMP link counts are provided using $Y_S^t = HX^t$ in our estimation framework in Eq.(3). This approach is compatible with both of our single-point and multi-point measurement scenarios. Second, we introduce an efficient-compressive aggregation method in Section.III-E.

D. Optimal Flow Sampling under Hard Resource Constraints

The notion of sampling here denotes the process of sequential allocation of K out of T entries of TCAM at epoch τ_t to target flows, which are selected based on the information collected up to that time. The main goal is to adaptively track and measure the most rewarding(/informative) traffic flows that, if measured accurately, can yield the best improvement of overall measurement utility (the exact performance metric is dependent on the monitoring application). For this purpose, multi-armed bandit sequential resource allocation algorithms are used. A classic MAB problem involves a number of independent arms, each of which, when played, offers random reward drawn from a distribution with unknown mean. At each time, a player chooses a subset of arms to play, aiming to maximize reward or minimize regret over some time horizon T_c [17]. The Restless Multi-Armed Bandit (RMAB) optimal policies [18], [19] can be applied effectively to our framework for intelligent flow sampling in dynamic networks where: a) flows are independent; b) the dynamics of flows are not-known, and c) flow sizes can vary stochastically with time. In general, the optimal policies consists of two phases: exploration or learning phase, in which system dynamics is learned, and exploitation phase, where the most rewarding flow(s) (i.e. arm(s)) are measured (i.e. played).

In our problem, we would like to identify, track, and measure the K largest flows amongst n flows using K out of T entries of TCAM for two purposes. First, accurate measurements of the largest flows can improve the ability of our optimization techniques in Eq.(2) and Eq.(3) for fine-grained flow estimation. Second, this can enhance the capability of the monitoring system in Heavy Hitter Identification (HHI) where Heavy Hitters (HH) are flows with a flow size larger than a threshold θ. Note that HHI is of particular importance in both network management and security applications. For a flow sampling strategy, to function effectively and efficiently in dynamic environments and under hard resource monitoring constraints (where $T << n$), the duration of the learning phase must be short. This is important because in many applications measurement intervals τ_ts are long, for example 5-15 minutes (see Table I) which leads to very long learning durations. Therefore, after a comprehensive survey, we adopted and modified the upper confidence bound algorithm in [18] based on our application requirements and propose our Modified Upper Confidence Bound algorithm (Alg. 1).

In MUCB algorithm, we modify the original UCB algorithm [18] to choose the K most rewarding flows. For this purpose, first, the time horizon T_c is determined. Then all T entries of TCAM are used to measure all n incoming flows over $\lceil \frac{n}{T} \rceil$ measurement interval and the indicies of incoming flows, sorted in descending order, are reported including the indicies of K heaviest flows (I_k^t) and the indicies of flows that must be aggregated (I_g^t). Accordingly, the direct flow measurement matrix A_k^t is defined as: $A_k^t(:, I_k^t) = 1$. In computing flow indicies the first term $\bar{x}_j$ favors measurement resources toward flows that are historically larger. In fact, larger flows that are measured more frequently are dominated by the first term and smaller flows that are observed less frequently are dominated by the second term. To improve the agility of the algorithm in dynamic environment, for observing a variety of flows, we also modify the original UCB algorithm [18] by multiplying the first term $\bar{x}_j$ by a coefficient α where $\alpha \leq 1$. In this algorithm, the parameter T_c can be adjusted by the user. We also propose the following method (Eq.(9)) to use the temporal auto-correlation of flows to compute T_c where $t_{R_x^j}(\beta)$ is the delay (t_d) that the auto-correlation of jth flow x_j (i.e. $R_x^j := E[x_j(t)x_j(t-t_d)]$) drops to $\beta \max(R_x^j)$. Typically, β is set to 0.5, but smaller β leads to larger T_c.

$$T_c = \frac{1}{n} \sum_{j=1}^{n} t_{R_x^j}(\beta) \quad \text{and} \quad \beta \leq 1 \tag{9}$$

This algorithm is simple to implement and learning phase is very short which is an important factor in network monitoring applications under hard resource constraints. It is also effective because the set I_k^t indicates highly possible large flows; accordingly, it is very efficient for heavy hitter detection in dynamic environments. Our initial exploration using real traffic traces shows that the average probability of error in selecting

Algorithm 1 Modified Upper Confidence Bound (MUCB)

Input: Time horizon T_c.
Output: At each epoch t, the set of sorted indicies of incoming flows (I^t) in descending order where $I^t = I_k^t \bigcup I_g^t$.
while True **do**
  - Set $t = 1$, measure all n flows $\{x_j\}_{j=1}^n$ using all T entries of TCAM over $\lceil \frac{n}{T} \rceil$ epochs, and set $t_c = n$.
  while $t < T_c$ **do**
    - Compute flow indicies $I_j^t = \alpha \bar{x}_j + \sqrt{\frac{2ln(t_c)}{t_j}}$ for all flows $j = 1, ..., n$ where $\bar{x}_j$ is the average flow size for j^{th} flow, t_j is the number of times flow j has been measured so far and t_c is the overall number of measurements done so far.
    - Sort the set I^t and report indicies in descending order as $I^t = I_k^t \bigcup I_g^t$.
    - Allocate k measurement entries to the k flows with indicies in I_k^t and measure them.
    - $t = t + 1$, $t_c = t_c + K$.
  end while
end while

Algorithm 2 Exponential Aggregation Technique (EAT)

Input: Aggregation parameters ρ and δ.
Output: Aggregation Matrix A_g^t.
Initialization: Set $i_c = 0$ and $A_g^t = 0_{m \times n}$.
for i = m to 1 **do**
  - $r = \lceil \rho(n - K)\frac{1}{\delta^i} \rceil + 1$
  for j = 1 to r **do**
    - $A_g^t(i, I_g^t(i_c + j)) = 1$
  end for
  - $i_c = i_c + r$
end for

K largest flows using Alg. 1 is fairly low; for example, Pe_{Avg} for Abilene and Geant networks are $Pe_{Avg}^{Abilene} = 0.2320$ and $Pe_{Avg}^{Geant} = 0.1483$, respectively. On the other hand, the set I_g^t in Alg. 1 declares an important order for aggregated flows which can be used to design a near optimal aggregation matrix.

E. Efficient-Compressive Aggregation in Practice

Now lets suppose the set I_g^t is given where its elements declare the indicies of network flow sizes in descending order. Therefore, using the fact that grouping attributes with the same quantity can improve estimation accuracy in UDLI problems, then, an efficient-compressive aggregation matrix can be designed to aggregate $n - K$ flows using m entries of TCAM under hard resource constraints. For this purpose, the following exponential aggregation algorithm is proposed (Alg. 2) where more measurement resources (TCAM entries) are allocated to the earlier indicies in the set I_g^t, indicating larger flows. Then, a large number of smaller flows with more stable behaviors are aggregated using smaller number of TCAM entries. Through more detailed exploration using real traffic traces, we have measured and found that flow indicies reported by Alg. 1 ($\{I_g^t\}_{t=1}^{T_c}$) exhibit a well spatio-temporal stability. Accordingly, applying our optimization formulation in Eq.(2), provides a very precise estimates for both large and small flows. Hence, our framework is effective not only for heavy hitter detection but also for estimating sub-population flow sizes which is of particular importance to network security applications such as DDOS attack detection. Parameters ρ and δ, which controls the number of flows aggregated per entries, are defined by the user.

IV. *i*STAMP PERFORMANCE EVALUATION

In this section, the effectiveness of our network measurement and inference framework in Figure 1 is justified. For this purpose, three real networks (Table I) with different configurations are considered. The Abilene [20] and GEANT [21] networks are well known networks with publicly available traffic traces. The routing matrices $H_{Abilene}$ and H_{Geant} are (30×144) and (74×529) binary matrices with full row-ranks. To consider a more dynamic environment with highly varying flows, we processed the publicly available data center packet traces in [22] and randomly chose a subset of rapidly fluctuating flows over time/space. We also assumed a Fat-Tree topology where the number of servers communicating among different racks (n_{Ext}) vary and ECMP routing is used between aggregation and core levels. In this case, flows are defined as traffic between servers in different racks.

Each configuration is defined by selecting different values for T and K and α, and by choosing an appropriate aggregation matrix. Also, Alg. 1 is used to directly measure the highest possible large flows. In addition, we evaluate the performance of three different aggregation techniques: 1) Block Aggregation Technique (BAT); 2) Random Aggregation Technique (RAT), and 3) Exponential Aggregation Technique (EAT). In BAT, m entries of TCAM are equally shared among $(n - K)$ flows where each block contains a set of $\lceil \frac{n-K}{m} \rceil$ consecutive aggregable flows which models the mapping of flows into TCAM entries in many practical cases. RAT is an abstract model that we used, through a Monte-Carlo process, to show the performance of our measurement framework where the routing/aggregation structure is not under our control. In EAT, our aggregation method in Alg. 2 is used to efficiently allocate TCAM entries to flows, and to produce a well compressed form of linear aggregated measurements. We also use different network inference methods in Eq.(2)-Eq.(3) to verify the compatibility of our framework with a variety of optimization techniques which can be efficiently solved by existing low-complexity algorithms [10]. In the following, using Eq.(2) for flow estimation based on aggregated statistics and without Side Information, is denoted by "w/o SI". We use "w/ SI" to denote the case where SNMP link loads are incorporated into our formulation using Eq.(3).

Eq.(10) defines the metrics used in our performance evaluation. $NMSE$ is a metric that we used to measure the accuracy of ODF estimation for each configuration. To justify the effectiveness of our framework for HH detection, we first set the threshold θ as a fraction of the link capacity C_L, and then, the average probability of detection (P_{HH}^d) and the average probability of false alarm (P_{HH}^{fa}) are computed as in Eq.(10). Given n, these metrics are measured based on two

Network	Date	Duration	Resolution	TM Size ($n \times T_c$)
GEANT [21]	2005-01-08	1 week	15 min.	529×672
Abilene [20]	2004-05-01	1 week	5 min.	144×2016
Data Center (Univ1) [22]	2010	$\approx$ 1 hour	1 sec	$48(/192) \times 3500$

TABLE I: Real Datasets under study.

quantities by varying T and K. First, T/n represents the ratio of the number of TCAM entries, used for both aggregation and direct measurement in our framework, and the number of flows. Note that low T/n asserts the hard resource constraint regime where a large number of flows (n) has been represented by T per-flow and aggregated measurements. Thus, T/n also indicates the Flow Compression Ratio (FCR). Second, K/T represents the ratio of the number of TCAM entries used for direct measurement and the total number of available TCAM entries.

Furthermore, we measure the effectiveness of our measurement framework in Sub-Population (SP) flow size estimation, which is of particular importance in some security applications such as DDoS detection. In Eq.(10), SP^t denotes the subset of I_g^t where $X(I_g^t) < \theta_l$ and SP_l^t denotes the lth disjoint subset of SP^t where the sum of flows in SP_l^t can be large (note that $\theta_l < \theta$). Also, π_0^t and π_1^t are prior probabilities which are computed experimentally. In our Monte-Carlo evaluation process, $|SP_l^t|$ is randomly chosen from interval $[\alpha_l(n-K) \quad \alpha_u(n-K)]$ where parameters α_l and α_u are selected properly to cover a wide range of SP sizes. Due to space limitation, we also summarize our results by averaging our performance criteria in Eq.(10) over different choices of T and K (e.g. $NMSE_{Avg}$).

$$NMSE = \frac{1}{T_c} \sum_{t=1}^{T_c} \frac{\left\| X^t - \hat{X}^t \right\|_2}{\left\| X^t \right\|_2}$$

$$P_{HH}^d = \frac{1}{T_c} \sum_{t=1}^{T_c} Pr\left(\hat{X}^t \geq \theta | X^t \geq \theta \right)$$

$$P_{HH}^{fa} = \frac{1}{T_c} \sum_{t=1}^{T_c} Pr\left(\hat{X}^t \geq \theta | X^t < \theta \right)$$

$$SPE = \frac{1}{T_c} \sum_{t=1}^{T_c} \frac{1}{|SP^t|} \sum_{l \in SP^t} \frac{\left| \left\| \hat{X}_{SP_l^t} \right\|_2 - \left\| X_{SP_l^t} \right\|_2 \right|}{\left\| X_{SP_l^t} \right\|_2} \quad (10)$$

$$P_{SP}^d = \frac{1}{T_c} \sum_{t=1}^{T_c} \pi_1^t Pr\left(\left\| \hat{X}_{SP^t} \right\|_2 \geq \theta | \left\| X_{SP^t} \right\|_2 \geq \theta \right)$$
$$+ \frac{1}{T_c} \sum_{t=1}^{T_c} \pi_0^t Pr\left(\left\| \hat{X}_{SP^t} \right\|_2 \leq \theta | \left\| X_{SP^t} \right\|_2 \leq \theta \right)$$

$$P_{SP}^{fa} = \frac{1}{T_c} \sum_{t=1}^{T_c} Pr\left(\left\| \hat{X}_{SP^t} \right\|_2 \geq \theta | \left\| X_{SP^t} \right\|_2 < \theta \right)$$

Figure 2 shows the $NMSE$ for different configurations on Geant network where Eq.(2) and Eq.(3) with (p,q,λ)=(2,1,0.01), as CS inference techniques, are used for flow size estimation. In the absence of side information a better accuracy is achieved at higher K/T's. However, in general, the optimal choice for the number of direct flow measurements K which yields to the minimum $NMSE$ (indicated by solid black points) is obtained at $K/T < 1$. In other words, the best accuracy is often achieved by allocating at least part of the TCAM entries for direct measurement and the remaining for flow aggregation. Among these, EAT can enhance the estimation accuracy, for example, $NMSE_{Avg}^{BAT} = 0.38$ and $NMSE_{Avg}^{EAT} = 0.29$; this improvement is significantly higher at low T/n's. Incorporating side information SNMP link counts and using Eq.(3) can remarkably improve the precision of the estimation using BAT, RAT and EAT methods. In this case, EAT still can improve the accuracy ($NMSE_{Avg}^{BAT} = 0.19$

Aggregation Technique	$NMSE_{Avg}$ (w/o SI)	$\bar{P}_{HH}^d$ (w/o SI)	$\bar{P}_{HH}^{fa}$ (w/o SI)
BAT	0.3857	0.7762	0.0696
RAT	0.5239	0.4508	0.0900
EAT	0.3602	0.7282	0.0496
Aggregation Technique	$NMSE_{Avg}$ (w/ SI)	$\bar{P}_{HH}^d$ (w/ SI)	$\bar{P}_{HH}^{fa}$ (w/ SI)
BAT	0.1663	0.8473	0.0396
RAT	0.2578	0.7361	0.0571
EAT	0.1762	0.8173	0.0341
Aggregation Technique	SPE_{Avg} (w/o SI)	$\bar{P}_{SP}^d$ (w/o SI)	$\bar{P}_{SP}^{fa}$ (w/o SI)
BAT	0.4943	0.7960	0.1360
Aggregation Technique	SPE_{Avg} (w/ SI)	$\bar{P}_{SP}^d$ (w/ SI)	$\bar{P}_{SP}^{fa}$ (w/ SI)
BAT	0.1765	0.9019	0.0086

TABLE II: The average performance for Abilene network where (p,q,λ)=$(\infty,0,0)$ in Eq.(2)-Eq.(3), $\alpha = 1$, $\theta = 0.01C_L$, $\theta_l = 0.0025C_L$, (α_l, α_u)=(0.01,0.2), $C_L = 1Mbps$, $\rho = 1$ and $\delta = 5$.

and $NMSE_{Avg}^{EAT} = 0.16$); however, since aggregated measurements are interpreted as side information in Eq.(3), the difference between BAT and EAT is low. This is an important fact that is used to address the aggregation feasibility in iSTAMP (see Section IV-B). In addition, although increasing T/n improves the accuracy, the slope of the improvement is slow after some near-optimal FCR (e.g. $T/n \approx 0.1$).

Therefore, iSTAMP achieves high estimation accuracy using a small fraction of available TCAM entries at low FCRs and using different aggregation techniques. Hence, it can produce well-compressed aggregated flow measurements and remarkably decrease the required storage capacity and reduce the communication overhead between control and data planes. Furthermore, Figure 3 represents the effectiveness of iSTAMP for reliably detecting both heavy hitters and heavy sub-populations. Among these, a high probability of detection and low probability of false alarms are achieved for HH detection where threshold θ is set to 10% of link capacity $C_L = 10Mpbs$.

This figure also shows that the detection capability for heavy sub-pupolations (with $\theta_l = 0.01C_L$) is still high for many choices of T and K. Under very hard resource constraints, although the sub-population detection perfromance is lower, it is still acceptable and it improves, rapidly.

In addition, Table II summarizes our results for Abilene network. These results re-emphasize the flexibility and capability of iSTAMP for different network monitoring applications under hard resource constraints. Note that, for both Geant and Abilene networks, T_c is set to the duration of the data in Table I (i.e. small β in Eq.(9)) which is the worst case scenario. Better accuracy can be achieved with smaller T_c's.

Figure 4 and Figure 5 show the performance of our measurement framework under very hard resource constraint $T/n = 0.0625$ and with SNMP side information using Eq.(3) in a data center environment with highly fluctuating flows. Here, the aggregation matrix is a given block aggregation matrix and routing matrix H is computed for the fat-tree topology assuming the ECMP routing between aggregation and core level [22]. As it was explained in Section.III, in dynamic environments we need to explore different flows more frequently, which is achieved by choosing a smaller α in Alg. 1. The direct consequence of this effect is seen in Figure 4 where our performance metrics have been improved

by decreasing α. To enhance the accuracy of estimation and improve the capability of the system for HHI in dynamic environments, iSTAMPS needs to repeatedly learn the behavior of flows. This is achieved by shortening the time horizon T_c. Figure 5 shows the effect of different choices of T_c using different values of β in Eq.(9) where n_{Ext} vary. Here, $(\beta = 0.5, n_{Ext} = 2)$, $(\beta = 0.5, n_{Ext} = 4)$, $(\beta = 0.2, n_{Ext} = 2)$ and $(\beta = 0.2, n_{Ext} = 4)$ correspond to $T_c = 186(s)$, $T_c = 260(s)$, $T_c = 1051(s)$ and $T_c = 1085(s)$, respectively. Also, $n_{Ext} = 2$ and $n_{Ext} = 4$ yields the number of flows $n = 48$ and $n = 192$. It is clear that, even in a very hard resource constraint $(T/n << 1)$, our measurement framework can obtain a good estimation accuracy and detecting capability by choosing optimum K/T. In practice, the designer of the system must appropriately set the controlling parameters α and β based on the history of the system and the required performances through a trial and error process.

A. Feasibility Study using Mininet

Mininet is a network testbed for developing OpenFlow and SDN experiments [23]. We create two networks that emulate Geant and Data Center environments and feed these two networks with real traffic traces listed in Table I. Table III summarizes the results of our implementation in Mininet, demonstrating the effectiveness and feasibility of iSTAMP in production environments. Geant(1) and Data Center denote single point measurement scenario and Geant(2) denotes multi-point measurement scheme involving multiple capableness routers running iSTAMP framework.

B. Discussion on Aggregation Feasibility

In iSTAMP the feasibility of the flow aggregation process is important. In our optimal aggregation matrix design (Eq.(8)), the feasibility can be simply modeled as another linear constraint and our CPLEX engine can efficiently solve it. However, in our abstract flow estimation models (Eq.(2)-Eq.(3)), we assume flows can be aggregated without any constraints. In the absence of SNMP link load measurements, BAT is the model used to address this constraint where feasible aggregable flows are grouped together and Eq.(2) is solved for flow size estimation. In the presence of SNMP side information, the feasibility constraint of the aggregation now has less impact on the overall estimation performance. In this case, Eq.(3) is used to provide fine grained flow estimates and aggregated flow measurements act as side information for this optimization problem to improve the estimation accuracy, as shown in Figure 2. In essence, iSTAMP is flexible enough to cope with different aggregation constraints.

Network/Config.	$T = 50, K = 0$	$T = 50, K = 5$	$T = 50, K = 10$
Geant(1)	0.8930	0.6171	0.4909

Network/Config.	$T = 24, K = 0$	$T = 24, K = 12$
Data Center	0.6175	0.2924

Network/Config.	$T = 50, K = 0$	$T = 50, K = 5$
Geant(2)	0.8910	0.6212

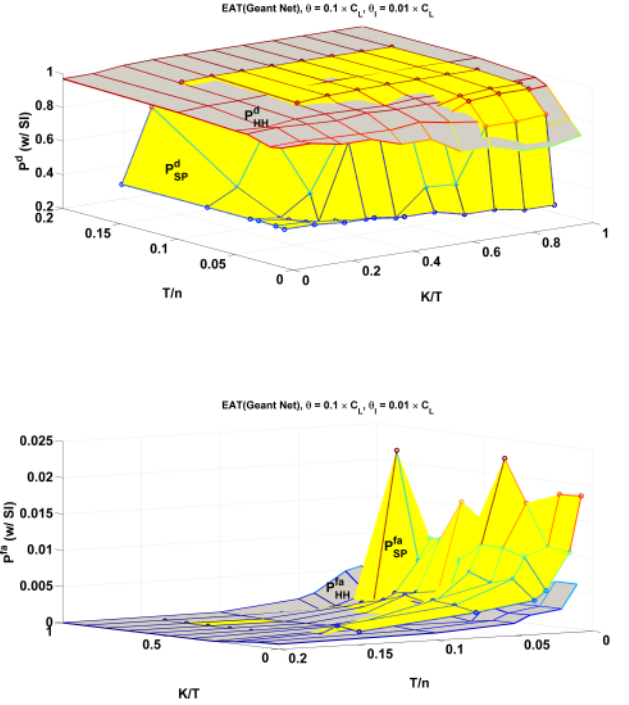*TABLE III:* $NMSE$(w/o SI) for different networks in Mininet implementation.

Fig. 3: P^d and P^{fa} in Geant Network with different configurations for both HH and SP detection where (α_l, α_u)=(0.01,0.1), $\rho = 1$ and $\delta = 1.975$.
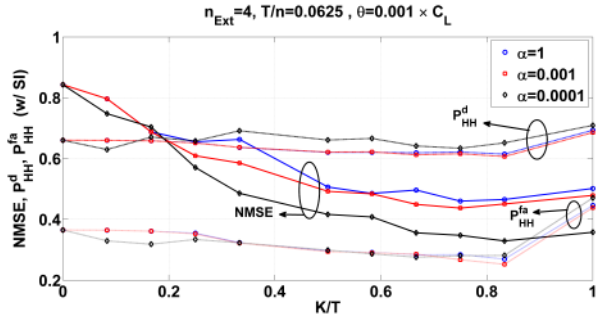

Fig. 4: $NMSE$, P^d_{HH} and P^{fa}_{HH} for Data Center with different configurations where (p, q, λ)=(2,0,0) in Eq.3.
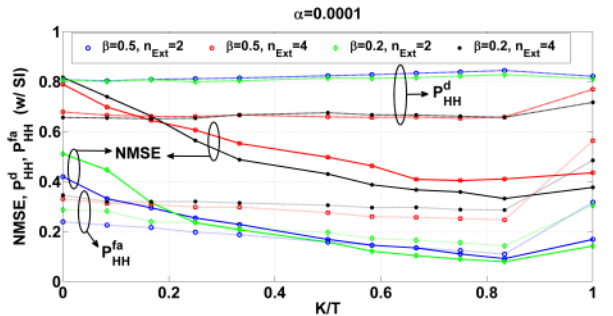

Fig. 5: $NMSE$, P^d_{HH} and P^{fa}_{HH} for Data Center with different configurations where (p, q, λ)=(2,0,0) in Eq.3 and β (i.e. T_c) and n_{Ext} vary.
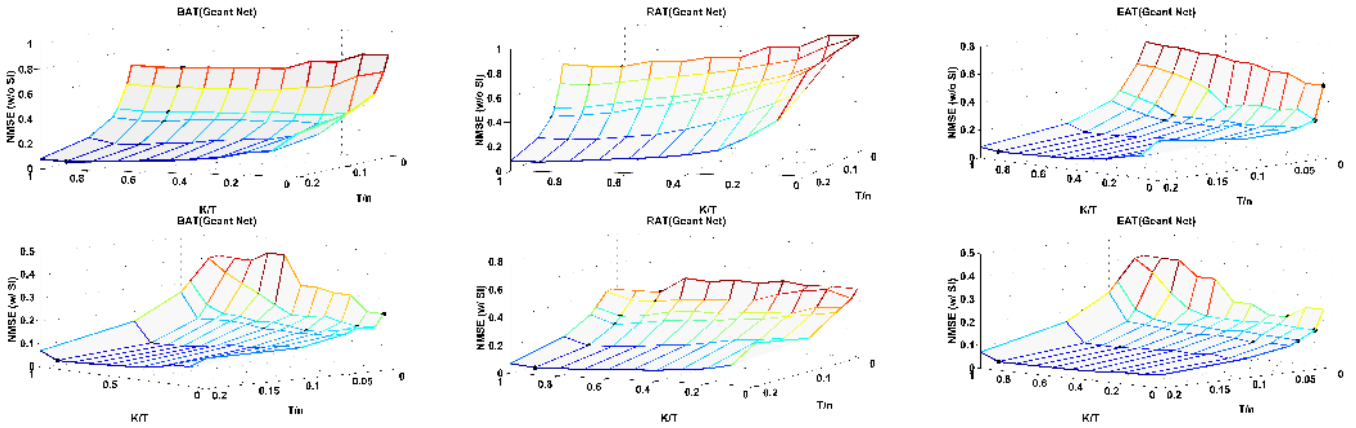
Fig. 2: $NMSE$ for Geant Network with different configurations using Eq.(2)-Eq.(3) with (p,q,λ)=(2,1,0.01) $\rho = 1$ and $\delta = 1.975$.

V. CONCLUSION

In this paper we introduced iSTAMP, an intelligent network measurement framework, where the flexibility provided by SDN is used to optimally aggregate and sample the most informative flows, simultaneously. We also developed an integer optimization formulation to design the optimal aggregation matrix and introduced an efficient-compressive flow aggregation technique. In addition, we proposed an efficient MAB based flow sampling algorithm to select the most rewarding flows with highest influence on the estimation accuracy. Our results showed that iSTAMP is a simple, generic and efficient framework providing accurate fine-grained flow measurements in dynamic environments and in different applications.

VI. ACKNOWLEDGEMENTS

REFERENCES

[1] "Cisco netflow," At: http://en.wikipedia.org/wiki/Netflow.

[2] "Sampled netflow," At: http://www.cisco.com/en/US/docs/ios/12\0s/feature/guide/12s\sanf.html.

[3] A. Kumar, M. Sung, J. Xu, and J. Wang, "Data streaming algorithms for efficient and accurate estimation of flow size distribution," 2004.

[4] A. Kumar and J. Xu, "Sketch guided sampling - using on-line estimates of flow size for adaptive data collection," in *INFOCOM*, 2006.

[5] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: Making use of multiple data sources," *ACM-SIGMETRICS*, 2006.

[6] T. Benson, A. Anand, A. Akella, and M. Zhang, "Microte: Fine grained trafc engineering for data centers," *ACM-IMC*, 2011.

[7] P. Ayres, H. Sun, H. Chao, and W. Lau, "Alpi: A ddos defense system for hogh-speed networks," *IEEE Journal oc Selected Areas in Communications*, vol. 24, pp. 1864–1876, 2006.

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

[9] M. Yu, L. Jose, and R. Miao, "Software defined traffic measurement with opensketch," *ACM-USENIX*, 2013.

[10] Y. C. Eldar and G. Kutyniok in *Compressed Sensing: Theory and Applications*, Cambridge Univ. Press, 2012.

[11] R. M, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 662–676, 2012.

[12] L. Yuan, C. Chuah, and P. Mohapatra, "Progme: towards programmable network measurement," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, 2011.

[13] N. Duffield, "Fair sampling across network flow measurements," *ACM SIGMETRICS*, 2012.

[14] H. Chao in *High Performance Switches and Routers*, Wiley, 2007.

[15] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Sanei, "On optimization of the measurement matrix for compressive sensing," *IEEE- EURASIP*, 2010.

[16] L. Zelnik-Manor, K. Rosenblum, and Y. Eldar, "Sensing matrix optimization for block-sparse decoding," *IEEE Trans. on Signal Proc.*, vol. 59, no. 9, pp. 4300–4312, 2011.

[17] J. Gittins, K. Glazebrook, and R. Weber in *Multi-armed Bandit Allocation Indices*, Wiley, 2011.

[18] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, 2002.

[19] K. L. H. Liu and Q. Zhao, "Learning in a changing world: Restless multi-armed bandit with unknown dynamics," *IEEE Trans. on Info. Theory*, vol. 59, no. 3, pp. 1902–1916, 2012.

[20] "Abilene traffic." www.cs.utexas.edu/ yzhang/research/AbileneTM.

[21] "Geant network:." http://totem.info.ucl.ac.be/dataset.html.

[22] D. M. T. Benson, A. Akella, "Network traffic characteristics of data centers in the wild," *ACM-IMC*, 2010. Data Center Data is Available At: http://pages.cs.wisc.edu/~tbenson/IMC10_Data.html.

[23] "Miniet," At: http://mininet.org/.