MeasuRouting: A Framework for Routing Assisted Traffic Monitoring

Saqib Raza, Guanyao Huang and Chen-Nee Chuah
University of California, Davis
Email: {sraza,gyhuang,chuah}@ucdavis.edu

Srini Seetharaman and Jatinder Pal Singh
Deutsche Telekom R&D Laboratories, USA
Email: {srini.seetharaman, jatinder.singh}@telekom.com

***Abstract*—Monitoring transit traffic at one or more points in a network is of interest to network operators for reasons of traffic accounting, debugging or troubleshooting, forensics, and traffic engineering. Previous research in the area has focused on deriving a placement of monitors across the network towards the end of maximizing the monitoring utility of the network operator for a given traffic routing. However, both traffic characteristics and measurement objectives can dynamically change over time, rendering a previously optimal placement of monitors suboptimal. It is not feasible to dynamically redeploy/reconfigure measurement infrastructure to cater to such evolving measurement requirements. We address this problem by strategically routing traffic sub-populations over fixed monitors. We refer to this approach as *MeasuRouting*.**

The main challenge for MeasuRouting is to work within the constraints of existing intra-domain traffic engineering operations that are geared for efficiently utilizing bandwidth resources, or meeting Quality of Service (QoS) constraints, or both. A fundamental feature of intra-domain routing, that makes MeasuRouting feasible, is that intra-domain routing is often specified for aggregate flows. MeasuRouting, can therefore, differentially route components of an aggregate flow while ensuring that the aggregate placement is compliant to original traffic engineering objectives. In this paper we present a theoretical framework for MeasuRouting. Furthermore, as proofs-of-concept, we present synthetic and practical monitoring applications to showcase the utility enhancement achieved with MeasuRouting.

I. Introduction

Several past research efforts have focused on the optimal deployment of monitoring infrastructure in operational networks for accurate and efficient measurement of network traffic. Such deployment involves both monitoring infrastructure *placement* as well as *configuration* decisions. An example of the former includes choosing the interfaces at which to install DAG cards, and the latter includes tuning the sampling rate and sampling scheme of the DAG cards.

The optimal placement and configuration of monitoring infrastructure for a specific measurement objective typically assumes a priori knowledge about the traffic characteristics. Furthermore, these are typically performed at longer time scales to allow provisioning of required physical resources. However, traffic characteristics and measurement objectives may evolve dynamically, potentially rendering a previously determined solution suboptimal.

We propose a new approach called *MeasuRouting* to address this limitation. MeasuRouting forwards network traffic across routes where it can be best monitored. Our approach is complementary to the well-investigated monitor placement problem [1–3] that takes traffic routing as an input and decides where to place monitors to optimize measurement objectives; MeasuRouting takes monitor deployment as an input and decides how to route traffic to optimize measurement objectives. Since routing is dynamic in nature (a routing decision is made for every packet at every router), MeasuRouting can conceptually adjust to changing traffic patterns and measurement objectives. In this paper, the overall *monitoring utility*, defined as a weighted sum of the monitoring achieved over all flows, is our primary concern.

The main challenge for MeasuRouting is to work within the constraints of existing intra-domain traffic engineering (TE) operations that are geared for efficiently utilizing bandwidth resources, or meeting Quality of Service (QoS) constraints, or both. This paper presents a framework for MeasuRouting that allows rerouting traffic towards the end of optimizing an ISP's measurement objectives, while being compliant to TE constraints. Our framework is generic and can be leveraged for a wide variety of measurement scenarios. We highlight a few examples as follows:

- A simple scenario is when certain routers implement uniform sampling or an approximation of it, and the network operator has greater interest in monitoring a subset of the traffic. MeasuRouting can be used to make important traffic traverse routes that maximize their overall sampling rate.
- Networks might implement heterogeneous sampling algorithms, each optimized for certain kinds of traffic sub-populations. For instance, some routers can implement sophisticated algorithms to give accurate flow-size estimates of medium-sized flows that otherwise would not have been captured by uniform sampling. MeasuRouting can then route traffic sub-populations that might have medium-sized flows across such routers. A network can have different active and passive measurement infrastructure and algorithms deployed, and MeasuRouting can direct traffic across paths with greater measurement potential.
- MeasuRouting can be used to conserve measurement resources. For instance, all packets belonging to a certain traffic sub-population can be conjointly routed to avoid maintaining states across different paths. Similarly, if state at a node is maintained using probabilistic data structures (such as sketches), MeasuRouting can enhance the accuracy of such

structures by selecting the traffic that traverses the node.

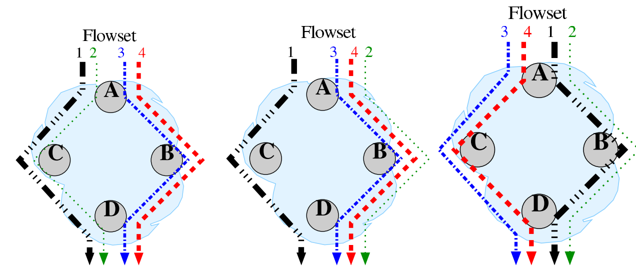
This paper presents a general routing framework for MeasuRouting, assuming the presence of special forwarding mechanisms. We present three flavors of MeasuRouting, each of which works with a different set of compliancy constraints, and we discuss two applications as proofs-of-concept. These MeasuRouting applications illustrate the significant improvement achieved by this additional degree of freedom in tuning how and where traffic is monitored. We believe our work to be the first in formally studying this new degree of freedom.

The rest of this paper is organized as follows: We present an overview of MeasuRouting in § II. § III details the MeasuRouting framework. Our example monitoring applications and a detailed performance evaluation are given in § IV. § V presents related work. We conclude in § VI.

II. MEASUROUTING OVERVIEW

As mentioned in the previous section, MeasuRouting must be cognizant of any implications that rerouting traffic has on Traffic Engineering (TE) policy. They are three fundamental ways in which MeasuRouting enhances traffic monitoring utility without violating TE policy:

- TE policy is usually defined for aggregated flows. On the other hand, traffic measurement usually deals with a finer level of granularity. For instance, we often define a flow based upon the five tuple ⟨*srcip, dstip, srcpt, dstpt, proto*⟩ for measurement purposes. Common intra-domain protocols (IGPs) like OSPF [4] and IS-IS [5] use link weights to specify the placement of traffic for each Origin-Destination (OD) pair (possibly consisting of millions of flows). The TE policy is oblivious of how constituent flows of an OD pair are routed as long as the aggregate placement is preserved. It is possible to specify traffic sub-populations that are distinguishable from a measurement perspective but are indistinguishable from a TE perspective. MeasuRouting can, therefore, route our fine-grained measurement traffic sub-populations without disrupting the aggregate routing. The example depicted in Figure 1 illustrates this argument. It shows four traffic sub-populations, f_1, f_2, f_3, and f_4, that have the same ingress and egress nodes. Suppose that f_1, f_2, f_3, and f_4 are of equal size. Router B has some dedicated monitoring equipment, and it is important for the network operator to monitor f_2. Our TE policy is to minimize the maximum link utilization. Figure 1(a) depicts the original routing that obeys the TE policy. Figure 1(b) represents a routing that violates the TE policy in order to route f_2 through router B. However, if the traffic sub-populations are routed as in Figure 1(c), f_2 is allowed to pass through the dedicated monitoring equipment, and the routing is indistinguishable from the original from the perspective of our TE policy. It is important to note that the aggregate traffic must span multiple paths in order for MeasuRouting to be useful in this way. If the aggregate traffic traverses a single path then no opportunity exists to differentially route subsets of the traffic.
- The second way in which MeasuRouting is useful stems from the definition of TE objectives. TE objectives may be oblivious to the exact placement of aggregate traffic and only take cognizance of summary metrics such as the maximum link utilization across the network. An aggregate routing that is slightly different from the original routing may still yield the same value of the summary metric. Suppose f_2 and f_3 pertain to two different OD pairs in Figure 1(a). Then the new routing depicted by Figure 1(c) changes the aggregate traffic placement discussed above. However, from a TE perspective the total link utilization of all links remains the same.
- Finally, a network operator can specify a certain permissible level of TE policy violations. Such a specification would enable a tradeoff between the advantage derived from MeasuRouting and adherence to TE policy. For instance, if the the network operator is willing to allow a 33% increase in the maximum link utilization, the routing in Figure 1(b) becomes a compliant solution.



(a) Original (b) Violating (c) Compliant

Fig. 1. Illustration of using routing to focus on a traffic sub-population. In the above example, router B has special sampling of interest to us. To apply this sampling on Flowset 2, we can route through router B, while (b) violating, or (c) being compliant to TE policy.

The above discussion deals with the requirement that MeasuRouting must operate within the confines of the TE policy. The other equally important challenge is that any MeasuRouting solution should be physically realizable according to the constraints of the underlying forwarding mechanisms. For instance, in order to selectively route a certain traffic sub-population, the capability must exist to execute the requisite forwarding. This introduces a host of issues. It would require state to be maintained for all traffic sub-populations, and might impose limits on the cardinality or the membership of such traffic sub-populations. Other concerns may stem from the exact routing protocols used to implement MeasuRouting. For instance, a routing protocol may impose a constraint that traffic between a pair of nodes may only traverse paths that are along shortest paths with respect to certain link weights. We address a few of these issues in this paper. However, the main focus of this paper is to investigate the potential monitoring benefits of, and to present an underlying theoretical framework for MeasuRouting. The actual forwarding, which can potentially be implemented using programmable routers [6–8], is outside the scope of this paper. § V and § VI touch on some of these auxiliary concerns.

III. MEASUROUTING FRAMEWORK

We now present a formal framework for MeasuRouting in the context of a centralized architecture. A centralized architecture refers to the case where the algorithm deciding how distributed nodes will route packets using MeasuRouting has global information of *a)* the TE policy, *b)* the topology and monitoring infrastructure deployment, and *c)* the size and importance of traffic sub-populations.

A. Definitions

$G(V, E)$ represents our network, where V is the set of nodes and E is the set of directed links.

A *macro-flowset* represents a set of flows for which an aggregate routing placement is given. In the context of intra-domain IP routing, a macro-flowset comprises all flows between an OD pair. For MPLS networks, macro-flowsets can be defined as all flows between an ingress-egress pair in the same QoS class. Our only requirement is that flows in a macro-flowset have the same ingress and egress nodes. In this paper we consider all flows between an OD pair to constitute a single macro-flowset. The set of all $|V| \times |V-1|$ macro-flowsets is given by Θ. Γ^x_{ij} gives the fraction ($[0,1]$) of the traffic demand belonging to macro-flowset x placed along link (i, j). $\{\Gamma\}^{x\in\Theta}_{(i,j)\in E}$ is an input to the MeasuRouting problem and represents our *original routing*. We assume $\{\Gamma\}^{x\in\Theta}_{(i,j)\in E}$ is a *valid* routing i.e. flow conversation constraints are not violated and it is compliant with network TE policy.

A macro-flowset may consist of multiple *micro-flowsets*. θ denotes the set of micro-flowsets. There is a many-to-one relationship between micro-flowsets and macro-flowsets. Υ_x represents the set of micro-flowsets that belong to the macro-flowset x. We represent the fraction of traffic demands belonging to micro-flowset y, placed along link (i, j) by γ^y_{ij}. $\{\gamma\}^{y\in\theta}_{(i,j)\in E}$ represents our *micro-flowset routing* and gives the decision variables of the MeasuRouting problem. We use in_z and out_z to denote the ingress and egress nodes of micro/macro-flowset z, respectively. $\{\Phi\}_{x\in\Theta}$ and $\{\phi\}_{x\in\theta}$ represent the traffic demands or sizes of the macro-flowsets and micro-flowsets, respectively. It follows that $\Phi_x = \sum_{y\in\Upsilon_x} \phi_y$.

We define our measurement infrastructure and measurement requirement in abstract terms. $\{\mathcal{S}\}_{(i,j)\in E}$ denotes the *sampling characteristic* of all links. The sampling characteristic is the ability of a link to sample traffic. It could be a simple metric like the link sampling rate. $\{\mathcal{I}\}_{y\in\theta}$ denotes the *sampling utility* of the micro-flowsets. This is a generic metric that defines the importance of measuring a micro-flowset. $\{\mathcal{S}\}_{(i,j)\in E}$ and $\{\mathcal{I}\}_{y\in\theta}$ are inputs to our problem.

Finally, we define the *sampling resolution function* (β):

$$\beta : (\{\gamma\}^{y\in\theta}_{(i,j)\in E}, \{\mathcal{S}\}_{(i,j)\in E}, \{\mathcal{I}\}_{y\in\theta}) \rightarrow \Re \qquad (1)$$

β assigns a real number representing the monitoring effectiveness of a micro-flowset routing for given link sampling characteristics and micro-flowset sampling utilities. The objective of MeasuRouting is to maximize β. Specifying β, $\{\mathcal{S}\}_{(i,j)\in E}$, and $\{\mathcal{I}\}_{y\in\theta}$ defines a concrete MeasuRouting application. § IV discusses this in detail.

B. Classes of MeasuRouting Problems

We now define three classes of MeasuRouting problems, each differing in the level of required conformance to the original routing:

1) Least TE Disruption MeasuRouting (LTD): The basic version of our MeasuRouting problem, referred to as LTD, can be formulated as the following:

$$\text{maximize } \beta$$

$$\text{subject to}$$

$$\sum_{i:(i,j)\in E} \gamma^y_{ij} - \sum_{k:(j,k)\in E} \gamma^y_{jk} = 0 \qquad y \in \theta, j \neq in_y, out_y \quad (2)$$

$$\sum_{i:(i,j)\in E} \gamma^y_{ij} - \sum_{k:(j,k)\in E} \gamma^y_{jk} = -1 \qquad y \in \theta, j = in_y \quad (3)$$

$$(1+\epsilon)\sigma^\Gamma \geq \sigma^\gamma \quad (4)$$

$$\gamma^y_{ij} \geq 0 \qquad y \in \theta, (i,j) \in E \quad (5)$$

It tries to maximize β by computing a micro-flowset routing, $\{\gamma\}^{y\in\theta}_{(i,j)\in E}$, that obeys the flow conservation constraints given by Eq. 2 and 3 The only constraint for LTD is that the aggregate TE policy is not violated. This constraint is represented by Equation 5. σ^Γ gives the value of the TE metric of the original macro-flowset routing. Similarly σ^γ is a function of the micro-flowset routing that gives the corresponding value of the TE metric for it. Equation 5 specifies that σ^γ does not exceed σ^Γ by more than a certain percentage, signified by a tolerance parameter ϵ. Traditionally, the TE metric is some measure of the utilization of network links. For instance, σ^Γ and σ^γ can represent $|E|$ element row vectors giving link utilizations. Alternatively, they can be single non-negative numbers representing the utilization of the most congested link. The definition of the TE metric depends upon the network's TE policy.

2) No Routing Loops MeasuRouting (NRL): The flow conservation constraints in LTD do not guarantee the absence of loops. In Figure 1, it is possible that the optimal solution of LTD may involve repeatedly sending traffic between routers A, B and C in a loop, so as to sample it more frequently while still obeying the flow conservation and TE constraints. Such routing loops may not be feasible or desirable in real-world routing implementations. We, therefore, propose NRL which ensures that the micro-flowset routing is loop-free. Loops are avoided by restricting the set of links along which a micro-flowset can be routed. This restriction is accomplished by supplementing the LTD problem with the following additional constraint:

$$\gamma^y_{ij} = 0 \qquad y \in \theta, (i,j) \notin \Psi_{x:y\in\Upsilon_x} \qquad (6)$$

Eq. 6 states that only links included in $\Psi_{x:y\in\Upsilon_x}$ may be used for routing micro-flowset $y \in \theta$. We restrict the membership of $\Psi_{x:y\in\Upsilon_x}$ such that the induced graph of $\Psi_{x:y\in\Upsilon_x}$ forms a directed acyclic graph. Since there are no cycles in the graph induced by $\Psi_{x:y\in\Upsilon_x}$, the micro-flowset routing does not contain any loops. We guarantee that a feasible routing exists for each micro-flowset by stipulating that the following implication is always true:

$$\Gamma_{ij}^{x:y\in\Upsilon_x} > 0 \Rightarrow (i,j) \in \Psi_{x:y\in\Upsilon_x} \tag{7}$$

There could be multiple ways of constructing $\Psi_{x:y\in\Upsilon_x}$. An example construction is given in Algorithm 1.

Algorithm 1

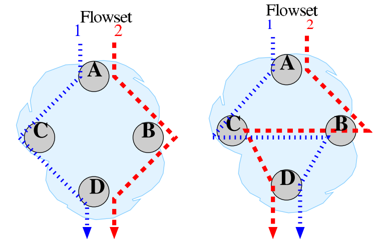
$\Psi_{x:y\in\Upsilon_x} = \emptyset$
for all $(i,j) \in E$ **do**
  if $\Gamma_{ij}^{x:y\in\Upsilon_x} > 0$ **then**
    $\Psi_{x:y\in\Upsilon_x} \leftarrow \Psi_{x:y\in\Upsilon_x} \cup \{(i,j)\}$
  end if
end for
$\hat{E} \leftarrow E/\Psi_{x:y\in\Upsilon_x}$
{A specific order of choosing links in $\hat{E}$ may be specified for the following part}
for all $(i,j) \in \hat{E}$ **do**
  if Induced graph of $\Psi_{x:y\in\Upsilon_x} \cup \{(i,j)\}$ is acyclic **then**
    $\Psi_{x:y\in\Upsilon_x} \leftarrow \Psi_{x:y\in\Upsilon_x} \cup \{(i,j)\}$
  end if
  $\hat{E} \leftarrow \hat{E}/\{(i,j)\}$
end for

3) Relaxed Sticky Routes MeasuRouting (RSR): NRL ensures that there are no routing loops. However, depending upon the exact forwarding mechanisms and routing protocol, NRL may still not be feasible. To further elaborate this point consider the example in Figure 2. We have two macro-flowsets x_1 and x_2 having the same traffic demands i.e., $\Phi_{x_1} = \Phi_{x_2}$. Figure 2(a) represents our original routing that sends all traffic belonging to x_1 along the path (A, C, D), and that belonging to x_2 along (A, B, D). MeasuRouting can set $\{\gamma\}_{(i,j)\in E}^{y\in\theta}$ such that we route the micro-flowsets in macro-flowset x_1 (Υ_{x_1}) across the path (A, C, B, D), and the micro-flowsets in macro-flowset x_2 (Υ_{x_2}) across the path (A, B, C, D). Note that the utilization on all links will remain the same except for (B, C) and (C, B). Assuming that the TE policy is oblivious to the load on links (B, C) and (C, B), the micro-flowset routing is a feasible solution for both LTD and NRL. But this might not be feasible in practice given the routing implementation. For instance, consider the destination based shortest path routing paradigm followed in IP routing. The original routing implied that links (B, C) and (C, B) were not along the shortest path from A to D. The new routing would, therefore, require the micro-flowsets from A to D to be routed across a link that is not part of the shortest path from A to D. This may not be achievable given the underlying routing mechanisms.

RSR ensures that the micro-flowset routing does not route a macro-flowset's traffic along a link that the macro-flowset's traffic was not routed along in the original routing. This is accomplished by supplementing LTD with the following additional constraint (instead of using Eq. 6):

$$\gamma_{ij}^{y} = 0 \qquad\qquad y \in \theta, \Gamma_{ij}^{x:y\in\Upsilon_x} = 0 \tag{8}$$

Note that RSR is a special case of NRL with $\Psi_{x:y\in\Upsilon_x}$ constructed such that a link $(i,j) \in \Psi_{x:y\in\Upsilon_x}$ *if and only if* $\Gamma_{ij}^{x:y\in\Upsilon_x} > 0$.



(a) Original (b) Violating

Fig. 2. MeasuRouting can violate routing semantics.

C. Comparing MeasuRouting Problems

All the three MeasuRouting problems (LTD, NRL, RSR) represent different degrees of restrictions. LTD is the most flexible but may result in routing loops or traffic between an OD pair traversing links it does not traverse in the original routing. NRL disallows loops but may result in routing semantics being violated. RSR ensures loop-free routing as well as adherence to routing semantics. Consequently, we expect the best measurement gains for LTD, NRL, and RSR in that order. Our formulation makes a simplifying assumption about the micro-flowset routing. We assume that traffic can be distributed in any proportion across the set of permissible links for the macro-flowset as long as TE metric is not violated. This may or may not be possible depending upon the underlying forwarding mechanism. If not, then this would impose further restrictions on the micro-flowset routing. The focus of this paper is to study the potential gains of MeasuRouting. LTD, NRL, and RSR can be construed to represent the best case performance.

Note that the flow conservation constrains and the non-negativity constraints are linear functions. If the TE metric function σ^{γ} is linear then the TE constraint is also linear. Therefore, if the elements of the objective function (β) are also linear functions of the the decision variables, LTD, NRL, and RSR become linear programming (LP) problems. This implies that they are solvable in polynomial time.

IV. Performance Evaluation

This section evaluates the performance of MeasuRouting for specific monitoring applications. A MeasuRouting application can be defined by specifying the sampling resolution function (β), and its constituents i.e., link sampling characteristics $(\{\mathcal{S}\}_{(i,j)\in E})$ and micro-flowset sampling utilities $(\{\mathcal{I}\}_{y\in\theta})$. We proceed to define and study two MeasuRouting applications in § IV-A and § IV-B. For both applications we consider the utilization of the most congested link as our TE metric, i.e., σ^{Γ} and σ^{γ} represent the maximum link utilization resulting from the original and micro-flowset routing, respectively. We also have a common definition of the link sampling characteristics across both our applications. The sampling characteristic of a link (i,j), $\mathcal{S}_{(i,j)}$ is equal to $p_{ij} \in [0,1]$, where p_{ij} represents the known sampling rate of link (i,j).

We have a set of flows $\mathcal{F}$. Each flow $f \in \mathcal{F}$ has an associated ingress node $in_f \in V$ and egress node $out_f \in V$. $f \in \mathcal{F}$ belongs to macro-flowset x if and only if $(in_f, out_f) =$

(in_x, out_x). We represent the traffic demand of flow f by b_f, and the importance or utility of sampling it by i_f. We define k to be the total number of micro-flowsets for each macro-flowset. We use $\upsilon_{y\in\theta}$ to represent the set of flows that belong to the micro-flowset y.

It follows that the aggregate traffic demand for macro-flowset x is given by $\Phi_x = \sum_{f\in\mathcal{F}_x} b_f$. Most IP networks use link-state protocols such as OSPF [4] and IS-IS [5] for intra-domain routing. In such networks, every link is assigned a cost and traffic between any two nodes is routed along minimum cost paths. Setting link weights is the primary tool used by network operators to control network load distribution and to accomplish TE objectives. We use the popular local search meta-heuristic in [9] to optimize link weights with respect to our aggregate traffic demands $\{\Phi\}_{x\in\Theta}$. The optimized link weights are then used to derive our original routing $\{\Gamma\}_{(i,j)\in E}^{x\in\Theta}$.

Our applications are differentiated on the basis of the set of flows $\mathcal{F}$, and how we assign the sampling importance i_f and the traffic demand b_f of each flow $f \in \mathcal{F}$. For both our applications we can consider the importance of a flow f, i_f, to be the points we earn if we were to sample a byte for that flow. We wish to maximize the the total number of points earned, by routing our traffic across the given topology. This total number of points is given by the following:

$$\Delta_{\text{MR}} = \sum_{f\in\mathcal{F}} \sum_{(i,j)\in E} p_{ij} i_f b_f \gamma_{ij}^{\upsilon^{-1}(f)} \tag{9}$$

$\upsilon^{-1}(f)$ in Eq. 9 denotes the micro-flowset to which flow f belongs. In the default case, where we do not employ MeasuRouting, all flows are routed according to the original routing $\{\Gamma\}_{(i,j)\in E}^{x\in\Theta}$. Hence, the total number of points for this default case is:

$$\Delta_{\text{default}} = \sum_{f\in\mathcal{F}} \sum_{(i,j)\in E} p_{ij} i_f b_f \Gamma_{ij}^{\Upsilon^{-1}(f)} \tag{10}$$

$\Upsilon^{-1}(f)$ in Eq. 10 denotes the macro-flowset to which flow f belongs. Therefore, the performance gain as a result of MeasuRouting is given by:

$$\Delta = \frac{\Delta_{\text{MR}} - \Delta_{\text{default}}}{\Delta_{\text{default}}} \tag{11}$$

Our objective is to maximize Δ. The performance gain for a single flow $f \in \mathcal{F}$ can also be found in an analogous manner given by Eq. 12.

$$\frac{\sum_{(i,j)\in E} p_{ij} i_f b_f (\gamma_{ij}^{\upsilon^{-1}(f)} - \Gamma_{ij}^{\Upsilon^{-1}(f)})}{\sum_{(i,j)\in E} p_{ij} i_f b_f \Gamma_{ij}^{\upsilon^{-1}(f)}} \tag{12}$$

The MeasuRouting formulation requires us to specify the sampling utility function for each micro-flowset. Towards this end we define the sampling utility function as $\mathcal{I}_{y\in\theta} = \sum_{f\in\upsilon_y} i_f b_f$. Thus, the sampling utility of a micro-flowset is the sum of sampling utilities of its flows weighted by the flow sizes. We then define the sampling resolution function (β) for both our applications as:

$$\beta = \sum_{y\in\theta} \sum_{(i,j)\in E} p_{ij} \mathcal{I}_y \gamma_{ij}^y \tag{13}$$

$$= \sum_{y\in\theta} \sum_{(i,j)\in E} p_{ij} \gamma_{ij}^y \sum_{f\in\upsilon_y} i_f b_f \tag{14}$$

Parameter	Description	Value/Distribution
M	Flows per macro-flowset	3000
k	Micro-flowsets per macro-flowset	10
ϵ	TE violation threshold	0.1
i_f	Micro-flowset sampling utility	Pareto ($\lambda = 2$)

TABLE I
DEFAULT EXPERIMENTAL PARAMETERS

Note that according to our definition $\beta = \Delta_{MR}$. Therefore, for a given flows to micro-flowset assignment, maximizing β is equivalent to maximizing Δ_{MR} and Δ.

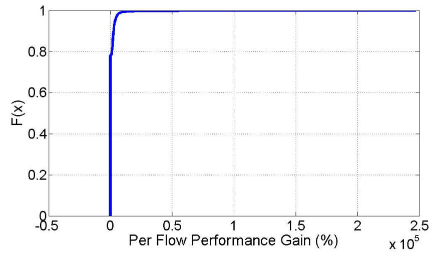
§ IV-A discusses a synthetic application where i_f and b_f are synthetically generated. We use our toy application to provide a general evaluation and sensitivity analysis for MeasuRouting. § IV-B applies MeasuRouting in a practical context. Specifically we leverage MeasuRouting to optimize the mix of packets captured for subsequent deep packet inspection.

A. Synthetic Application

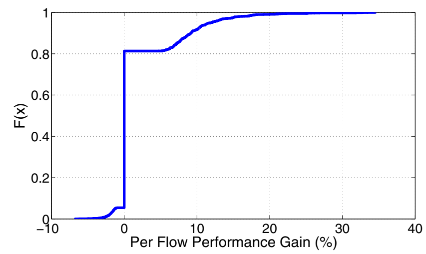
We first study MeasuRouting with flows having synthetically generated sampling importance and sizes. We specify distributions from which the flow sampling importance and size are randomly generated.

Each flow $f \in \mathcal{F}$ is assigned to a micro-flowset. All flows belonging to the same micro-flowset y have the same routing γ_{ij}^y. It follows that we have the greatest degree of freedom if each flow is assigned to a unique micro-flowset. This will allow each flow to be routed independently. However, this might not be scalable from both a computational and implementation perspective. We, therefore, have k micro-flowsets per macro-flowset. We also have $M \geq k$ flows for each macro-flowset. Each of the M flows in $\mathcal{F}$ belonging to a particular macro-flowset is assigned to one of its corresponding k micro-flowsets. There can be multiple ways of making such an assignment. The assignment scheme that we use assigns an equal number of flows to each of the k micro-flowsets of a macro-flowset, and ensures that the sampling importance of each flow in micro-flowset i is greater than the the sampling importance of each flow in the micro-flowset $i+1$. We stick to this assignment scheme for the rest of this section, unless specified otherwise. In order to get the size or traffic demand of each flow, we first generate aggregate traffic demands for each OD pair using a Gravity Model [10]. The traffic demand of flow f, b_f, is then set equal to the traffic demand of its corresponding OD pair divided by M. We set the MeasuRouting parameters to the values given in Table I for all experiments in § IV-A, unless specified otherwise.

1) Preliminary Comparison of MeasuRouting Problems: We first conduct a preliminary evaluation of the performance of the three MeasuRouting problems (LTD, NRL, and RSR) described in § III. We conduct our experiment for a 44 node

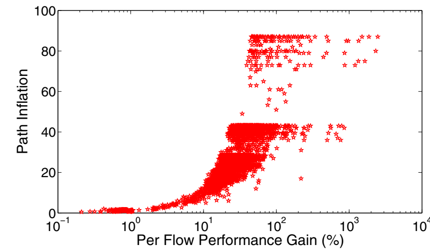


(a) LTD

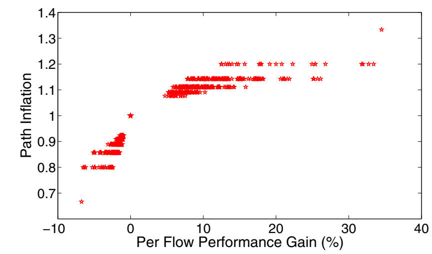


(b) NRL

Fig. 3. CDF of Per Flow Performance Gain



(a) LTD Path Inflation vs. Per Flow Performance Gain



(b) NRL Path Inflation vs. Per Flow Performance Gain

Fig. 4. Path Inflation in Micro-Flowset Routing

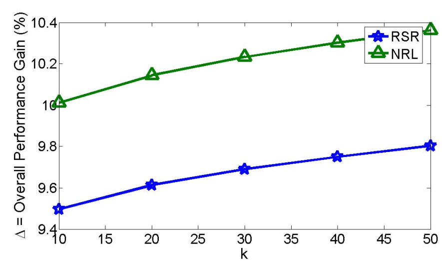


Fig. 5. MeasuRouting Performance for different k

and 88 link RocketFuel topology AS1221 [11]. Figure 3 shows the cumulative distribution function of the per flow performance gain as described in Equation 12. The per flow performance gain for a flow is as high as 250000% and 35% for LTD and NRL, respectively. We do not show results for RSR since its performance is very close to NRL. This is because Algorithm 1 introduces a very small number of additional paths. Some flows also have negative performance gain since MeasuRouting may divert flows with lower sampling importance away from paths with better sampling resources in order to allot them to flows with higher sampling importance. Figure 3 also shows that a significant fraction of flows have 0% performance gain, most probably because their micro-flowset routing remains unchanged from the original routing. The overall performance gain, Δ (Equation 11), is 131%, 10%, and 9.5% for LTD, NRL, and RSR respectively.

Consistent with our intuition in § III LTD shows the greatest performance gain since it offers the greatest flexibility for routing micro-flowsets. Part of this flexibility stems from the permissibility of routing loops. In order to gain a better understanding of the characteristics of the solution returned by LTD, we look at the path inflation given by $\sum_{(i,j)\in E} \gamma^{y}_{(i,j)} / \sum_{(i,j)\in E} \Gamma^{x}_{(i,j)}$, where $y \in \Upsilon_x$. Figure 4(a) shows the path inflation for LTD plotted against the per flow improvement. We see that flows with high performance gain have a very high path inflation. The path inflation for some flows exceeds the network diameter implying that LTD makes flows with high sampling importance traverse the same links multiple times. Figure 4(b) shows the path inflation for NRL is significantly smaller than that for LTD. Also, the average path length is 19.407, 3.309, and 3.3098 for LTD, NRL, and RSR respectively, while the original average path length is only 3.2373. Although LTD gives the greatest flexibility loops in the micro-flowset are not likely to be desirable or practically feasible. We, therefore, only focus on NRL and RSR from hereon.

2) Micro-Flowsets Per Macro-Flowset: The number of micro-flowsets per macro-flowset (k) has significant implications on the performance of MeasuRouting. As explained in § II, the ability to make disaggregated routing decisions for subsets of traffic between an OD pair is key for MeasuRouting. The worst case scenario is when $k = 1$, in which any MeasuRouting gains are restricted to the latter two cases delineated in § II. The best scenario is when $k = M$. We can then diversely route each flow in $\mathcal{F}$. However, a larger k value increases the complexity of the MeasuRouting problem. Also, in order to implement MeasuRouting, routers will have to keep separate forwarding state for each of the k micro-flowsets per macro-flowset. Larger values of k might not be practically feasible or desirable. Therefore, a tradeoff exists between the performance gain and scalability of MeasuRouting. Figure 5 shows the overall performance gain (Δ) for different values of k. We see that for both NRL and RSR, Δ monotonically increases with k. A promising result is that even for a reasonably small value of k equal to 10, MeasuRouting shows significant performance gain. Moreover, we see that there are diminishing returns for increasing k.

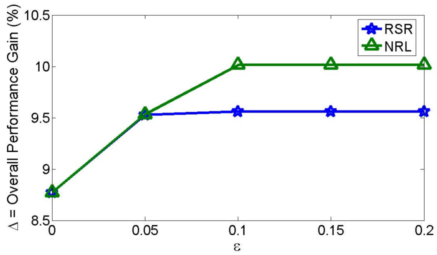


Fig. 6. MeasuRouting Performance for different ϵ

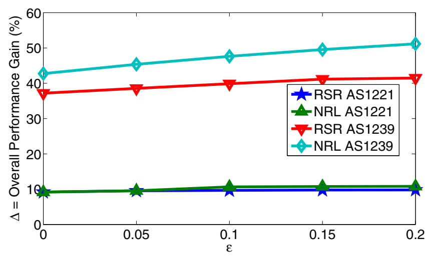


Fig. 7. MeasuRouting Performance for different Networks

3) Relaxing Traffic Engineering Constraints: As is obvious, allowing the traffic engineering constraints to be violated will increase the performance gain for MeasuRouting. Since we use the maximum link utilization as our traffic engineering metric, ϵ represents the permissible percentage increase in the maximum link utilization with respect to the original routing. Figure 6 shows how the performance improves with increasing ϵ. An interesting result is that even for $\epsilon = 0$ both NRL and RSR have positive Δ. In fact, even with $\epsilon = 0$ we have $\Delta \approx 8.8\%$. This is an important result showing that when there is zero tolerance for any traffic engineering violation, diversely routing micro-flowsets allows us to improve traffic monitoring.

4) Network Size and Multi-Path Routing: We also evaluate the effect of network size on the MeasuRouting performance gain. Figure 7 compares the overall performance gain Δ for RSR and NRL between AS1221 and AS1239 (52 nodes, 168 links) [11] for different ϵ. We see that the performance gains are much larger in AS1239. This stems from our observation in § II that making disaggregated routing decisions for different micro-flowsets corresponding to the same OD pair is most useful when there are multiple paths between the OD pair. Our original routing is based upon shortest path routing with respect to the optimized link weights. Since we use Equal Cost Multi-Path (ECMP) [9], some OD pairs have multiple paths. The larger the topology the greater are the number of paths per OD pair [12], and therefore, the greater the performance gain. In this study we chose ECMP for simplicity. A number of routing schemes provide a greater multiplicity of paths than ECMP [12]. MeasuRouting stands to perform much better with such routing schemes.

5) Micro-Flowset Sampling Utility Diversity: Since we cluster together flows with high sampling importance, we maximize the diversity in the sampling importance of different micro-flowsets. The greater this diversity the larger is the benefit of using MeasuRouting to make disaggregated micro-flowset routing decisions. On the other hand, if all micro-flowsets have the same sampling importance then the ability to make disaggregated routing decisions is of little use. We confirm this intuition by plotting the performance of RSR and NRL using another flow to micro-flowset assignment scheme. We simply assign flows of a macro-flowset to its k micro-flowsets in a round-robin fashion. The assignment is oblivious to the sampling importance of the flows. Note that the distribution of the micro-flowset sampling utility for this assignment has less variance than our original assignment. Figure 8 shows that this reduced diversity in the micro-flowset sampling utility results in inferior MeasuRouting performance.

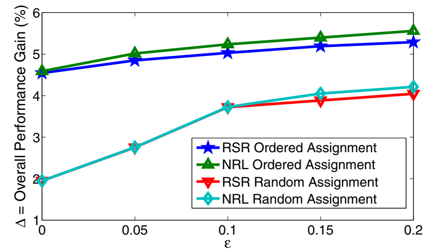


Fig. 8. MeasuRouting Performance for different Micro-Flowset Assignments

Distribution	**NRL Performance Gain (Δ)**
Exponential ($\lambda = 10$)	9.17805%
Pareto ($\lambda = 1$)	9.08674%
Pareto ($\lambda = 2$)	5.99197%
Pareto ($\lambda = 3$)	2.77895%

TABLE II
IMPACT OF MICRO-FLOWSET UTILITY DISTRIBUTION

Another way of altering the diversity is by choosing a different distribution from which to draw the sampling importance i_f of each individual flow $f \in \mathcal{F}$. Recall that micro-flowset sampling utilities are a sum of multiple identically distributed independent random variables. So for $M >> k$, the overall distribution of micro-flowset sampling utilities tends to be Gaussian according to the Central Limit Theorem. In order to make this overall distribution more closely mirror the underlying flow sampling importance distribution, we set $M = 50$ instead of 3000. Table II shows the overall performance gain for different underlying distributions of flow sampling importance. We see that more heavy tailed distributions result in better MeasuRouting performance. The strategy for defining micro-flowsets should, therefore, be geared towards increasing the variance in the distribution of micro-flowset sampling utility. More intelligent assignment schemes may use different number of flows per micro-flowset to increase the diversity in the sampling utilities of micro-flowsets.

B. Deep Packet Inspection Trace Capture

For the toy problem in § IV-A we synthetically generated flows and assigned sampling importance and flow sizes. In this section we elucidate a practical application of MeasuRouting using actual traffic traces from a real network, and with a meaningful definition of flow sampling importance. We consider the problem of *increasing the quality of traces* captured for subsequent Deep Packet Inspection (DPI). DPI is a useful process that allows post-mortem analysis of events seen in the network and helps understand the payload properties of transiting Internet traffic. However, capturing payload is often an expensive process that requires dedicated hardware (e.g., DPI with TCAMs [13]), or specialized algorithms that are prone to errors (e.g., DPI with Bloom Filters [14]), or vast storage capacity for captured traces. As a result, operators sparsely deploy DPI agents at strategic locations of the network, with limited storage resources. In such cases, payload of only a subset of network traffic is captured by the dedicated hardware.

Thus, improving the quality of the capture traces for subsequent DPI involves allocating the limited monitoring resources such that the representation of more interesting traffic is increased. We can leverage MeasuRouting to increase the quality of the traces captured, by routing interesting traffic across routes where they have a greater probability of being captured. The sampling rate p_{ij} in this context refers to the fraction of total bytes captured at link (i, j).

We first need to define what constitutes interesting traffic. Towards this end we define a field of interest as a subset of the bits of a packets IP header. This could be any subset. However, without loss of generality we use the field representing the destination port as our field of interest in this study. $u(i)$ is defined as the utility of capturing a packet with a specific destination port i. We infer $u(i)$ using historical data. We assume that we know the probability mass functions $\mathcal{P}$ and $\mathcal{Q}$, that represent the distribution of destination ports in the recent traffic history and the long term traffic history, respectively. We wish to assign utilities such that more packets are captured for flows that are responsible for the difference between $\mathcal{P}$ and $\mathcal{Q}$. We compute $u(i)$ as follows:

$$u(i) = -\ln(1 - |\mathcal{P}(i) - \mathcal{Q}(i)|) \tag{15}$$

According to Eq. 15, the utility of capturing a packet with the destination port equal to i increases with the absolute difference between $\mathcal{P}(i)$ and $\mathcal{Q}(i)$. When $\mathcal{P}(i)$ is equal to $\mathcal{P}(i)$, $u(i)$ is equal to zero. Eq. 15 is just an example utility function and network operators may define their own utility functions depending upon their measurement objectives.

We conduct our study for the Abilene network [15]. We consider a time series of sampled Abilene Netflow records taken at discrete units of time. Specifically we capture Netflow records for Tuesdays between *11:00* and *11:15* (GMT), for the first three months of 2009. This constitutes our long term traffic history. We consider the data of the last couple of Tuesdays in the above trace as our recent traffic history.

We construct our set of flows, $\mathcal{F}$, from the Netflow records constituting our recent traffic history. We set b_f equal to the number of captured bytes for the flow. The sampling importance, i_f, is set to $u(i)$ where i is the destination port of flow f. We use the same mechanism to derive the original routing and link sampling rates as specified in § IV-A.

MeasuRouting returns a micro-flowset routing given by $\{\gamma\}_{(i,j)\in E}^{y\in\theta}$. However, the routing is computed for the recent traffic history. We wish to use it route future traffic and

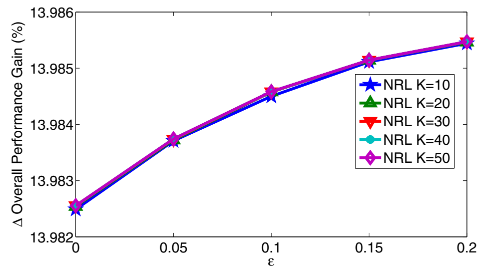


Fig. 9. MeasuRouting Performance for DPI Trace Capture

evaluate the quality of traces captured. To simulate such future traffic we use Netflow records for Tuesdays between *11:00* and *11:15* (GMT) for April 2009. Figure 9 shows the overall performance gain, Δ, for NRL for different k and $\epsilon = 0$. We observe that we get gain of 13.98% without any deviation from TE ($\epsilon = 0$). Furthermore, we observe that the gain is relatively unaffected by the value of k. That can be attributed to the scarcity of multiple paths in the small 9 node Abilene network. This study is only intended to provide a proof-of-concept. Network operators can define their own utility functions ($u(.)$) over their own fields of interest. MeasurRouting can be leveraged to enhance the quality of traces captured for their specific objectives.

V. Related Work

Earlier work in the area of traffic monitoring has focused on 1) inferring characteristics of original traffic from sampled traffic, 2) investigating and improving the effect of oblivious sampling on monitoring certain traffic sub-populations, and 3) placing monitor agents at certain strategic network locations. We summarize existing work in these three areas.

Claffy *et al.* [16] compared various sampling approaches at both packet-based and time-based granularities [16]. Several other research efforts aim to improve estimation of "heavy-hitter" traffic volume, flow-size distributions, traffic matrices, or flow durations [17–19] [20–24]. Recent work has demonstrated that conventional sampling techniques can obscure statistics needed to detect traffic anomalies [25] or execute certain anomaly detection algorithms [26]. All these previous works highlight the importance of being able to focus on specific traffic sub-populations. [27] proposes ways to focus monitoring budget on a specific traffic subpopulation by defining individual bins based on one or more tuples and allocating sampling budget to each bin. The traffic belonging to individual bins are identified using a counting bloom filter. There exists other proposals [28, 29] that also define the traffic subpopulation in a flexible manner.

All of the above mentioned works are orthogonal in nature to our proposal as their work focuses on improving monitoring at one monitor, while our work tries to route traffic to make best use of these monitors. The closest research efforts to ours are those presented in [1–3, 30, 31], which aim to achieve effective coordination across multiple traffic monitors to improve network-wide flow monitoring. The presented techniques adapt the sampling rate to changes in flow characteristics, attempt a different sampling strategy altogether, or apply network-wide constraints, typically to draw inferences about flow size distributions from sampled traffic statistics. However, these research efforts take traffic routing as a given, and do not achieve the best possible monitoring utility. MeasuRouting overcomes any limitations by computing the best possible traffic route for any given placement.

VI. Discussion & Future Directions

We empower network monitoring by intelligently routing flows of interest through static monitoring agents in a network. To the best of our knowledge this is the first work to present a comprehensive measurement oriented and traffic engineering compliant routing framework. Our routing framework is generic and can be leveraged for specific monitoring objectives and traffic characteristics. Our investigation of MeasuRouting applications highlights essential factors impacting its performance. MeasuRouting gains increase for networks with multiple paths between node pairs, and with original routing that utilizes these multiple paths. This is not a big concern for large-sized ISPs. Even for mid-sized ISPs, algorithms [12] exist to increase the diversity in the original routing. We did not observe significant differences between NRL and RSR. This is because Algorithm 1 is very basic and does not significantly increase the number of paths available for routing a micro-flowset. Deriving a better algorithm would further increase the performance of NRL and is a candidate for future work. We also observed that the diversity in the sampling utility of different micro-flowsets has a bearing upon MeasuRouting performance. MeasuRouting stands to gain tremendously from micro-flowset definition strategies that increase this diversity. We plan to explore such strategies in much greater detail.

Future work also involves deploying MeasuRouting in a realistic network substrate that is programmable. Specifically, we intend to implement MeasuRouting over OpenFlow [7]. The controller in such a system will feature an online version of the MeasuRouting algorithm that takes into account forecasting information of flows to determine an approximate solution at runtime. We would like to study the effect of additional constraints imposed by such realistic network substrates. We believe that our framework provides a firm foundation for routing assisted traffic engineering. We speculate that, in conjunction with programmable routing agents, our framework will stimulate further MeasuRouting applications.

VII. Acknowledgements

The authors would like to thank Anja Feldmann for her insightful comments and suggestions.

References

[1] C. Chaudet, E. Fleury, I. G. Lassous, H. Rivano, and M.-E. Voge, "Optimal positioning of active and passive monitoring devices," in *Proceedings of ACM CoNEXT*, 2005, pp. 71–82.

[2] K. Suh and Y. Guo and J. Kurose and D. Towsley, "Locating network monitors: Complexity, heuristics and coverage," in *Proceedings of IEEE INFOCOM*, March 2005.

[3] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran, "Reformulating the monitor placement problem: Optimal network-wide sampling," in *Proceedings of ACM CoNEXT*, 2006.

[4] "OSPF," http://tools.ietf.org/html/rfc2328, April 1998.

[5] "IS-IS," http://tools.ietf.org/html/rfc1142, February 1990.

[6] C. Wiseman et al., "A remotely accessible network processor-based router for network experimentation," in *Proceedings of the 4th ACM/IEEE ANCS*, 2008, pp. 20–29.

[7] "The OpenFlow Switch Consortium," http://www.openflowswitch.org.

[8] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, "The Click modular router," in *Proceedings of the 17th ACM SOSP*, December 1999, pp. 217–231.

[9] B. Fortz and M. Thorup, "Internet TE by Optimizing OSPF Weights," in *Proceedings of IEEE INFOCOM*, March 2000.

[10] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian, "Traffic matrix estimation: Existing techniques compared and new directions," in *Proceedings of ACM SIGCOMM*, August 2002.

[11] N. Spring, R. Mahajan, and T. Anderson, "Quantifying the Causes of Path Inflation," in *Proceedings of ACM SIGCOMM*, 2003.

[12] Y. Ohara, S. Imahori, and R. V. Meter, "MARA: Maximum alternative routing algorithm," in *Proceedings of IEEE INFOCOM*, April 2009.

[13] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern-matching using tcam," in *Proceedings of IEEE ICNP*, 2004, pp. 174–183.

[14] S. Dharmapurikar, P. Krishnamurthy, T. Sproull, and J. Lockwood, "Deep packet inspection using parallel bloom filters," in *IEEE Micro*, 2003, pp. 44–51.

[15] "The abilene network," http://www.internet2.edu.

[16] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," in *Proceedings of ACM SIGCOMM*, 1993.

[17] B.-Y. Choi and S. Bhattacharyya, "On the Accuracy and Overhead of Cisco Sampled NetFlow," in *Proceedings of ACM SIGMETRICS Workshop on Large Scale Network Inference (LSNI)*, June 2005.

[18] N. Duffield, C. Lund, and M. Thorup, "Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure," in *Proceedings of ACM SIGCOMM*, 2003, pp. 179–191.

[19] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a Better NetFlow," in *Proceedings of ACM SIGCOMM*, 2004.

[20] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice," *ACM Transactions on Computer Systems*, vol. 21, no. 3, pp. 270–313, August 2003.

[21] N. Hohn and D. Veitch, "Inverting sampled traffic," in *Proceedings of ACM SIGCOMM*, 2003.

[22] R. Kompella and C. Estan, "The Power of Slicing in Internet Flow Measurement," in *Proceedings of ACM SIGCOMM*, 2005.

[23] A. Kumar, M. Sung, J. Xu, and J. Wang, "Data streaming algorithms for efficient and accurate estimation of flow size distribution," in *Proceedings of ACM SIGMETRICS*, 2004, pp. 177–188.

[24] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An Information-Theoretic Approach to Traffic Matrix Estimation," in *Proceedings of ACM SIGCOMM*, 2003, pp. 301–312.

[25] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and Identification of Network Anomalies Using Sketch Subspaces," in *Proceedings of ACM Internet Measurement Conference*, 2006.

[26] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, "Is sampled data sufficient for anomaly detection?" in *Proceedings of ACM IMC*, 2006.

[27] A. Ramachandran, S. Seetharaman, N. Feamster, and V. Vazirani, "Fast Monitoring of Traffic Subpopulations," in *Proceedings of ACM Internet Measurement Conference*, 2008.

[28] H. V. Madhyastha and B. Krishnamurthy, "A Generic Language for Application-Specific Flow Sampling," *ACM Computer Communication Review*, vol. 38, no. 2, April 2008.

[29] L. Yuan, C.-N. Chuah, and P. Mohapatra, "ProgME: Towards Programmable Network MEasurement," in *Proceedings of ACM SIGCOMM*, 2007.

[30] M. R. Sharma and J. W. Byers, "Scalable Coordination Techniques for Distributed Network Monitoring," in *Proceedings of Passive and Active network Measurement (PAM) workshop*, April 2005.

[31] V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen, "CSAMP: A System for Network-Wide Flow Monitoring," in *Proceedings of USENIX NSDI*, 2008.