MeasuRouting: A Framework for Routing Assisted Traffic Monitoring

Saqib Raza, *Member, IEEE*, Guanyao Huang, *Member, IEEE*, Chen-Nee Chuah, *Senior Member, IEEE*, Srini Seetharaman, *Member, IEEE*, and Jatinder Pal Singh, *Member, IEEE*

***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 toward 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 subpopulations over fixed monitors. We refer to this approach as *MeasuRouting*. The main challenge for MeasuRouting is to work within the constraints of existing intradomain traffic engineering operations that are geared for efficiently utilizing bandwidth resources, or meeting quality-of-service (QoS) constraints, or both. A fundamental feature of intradomain routing, which makes MeasuRouting feasible, is that intradomain 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.**

***Index Terms*—Anomaly detection, intradomain routing, network management, traffic engineering, traffic measurements.**

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 timescales 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, our focus is on the overall *monitoring utility*, defined as a weighted sum of the monitoring achieved over all flows.

The main challenge for MeasuRouting is to work within the constraints of existing intradomain 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 toward 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 involves routers implementing uniform sampling or an approximation of it, with network operators being interested 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 subpopulations. 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 subpopulations that might have medium-sized flows across such routers. A network can have different active and passive measurement infrastructure and algorithms deployed, and MeasuRouting

Manuscript received October 25, 2010; revised April 15, 2011; accepted April 29, 2011; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P. Van Mieghem. Date of publication July 18, 2011; date of current version February 15, 2012. This work was supported in part by the NSF under Grant CNS-0905273.

S. Raza is with Cisco Systems, San Jose, CA 95134 USA (e-mail: sraza@ucdavis.edu).

G. Huang and C.-N. Chuah are with the Department of Electrical and Computer Engineering, University of California, Davis, Davis, CA 95616 USA (e-mail: gyhuang@ucdavis.edu; chuah@ucdavis.edu).

S. Seetharaman is with R&D Labs, Deutsche Telekom, Inc., Los Altos, CA 94022 USA (e-mail: seethara@stanford.edu).

J. P. Singh is with R&D Labs, Deutsche Telekom, Inc., Los Altos, CA 94022 USA, and also with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: jatinder@stanford.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2011.2159991

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 subpopulation can be conjointly routed to avoid maintaining states across different paths. Similarly, if the 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.

This paper is an extended version of our previous work [4], which we believe to be the first attempt to leverage routing as a degree of freedom for monitoring traffic. The present work extends upon [4] as follows.

- The results in [4] indicated that the performance of MeasuRouting is sensitive to the number of paths present between pairs of nodes. It is the relative difference in measurement capacity across such paths between a pair of nodes that is leveraged by MeasuRouting to improve monitoring performance. Whereas [4] showed significant performance gains for MeasuRouting, the choice of experimental networks was restricted to networks with a very low number of paths present between node pairs. This paper reports the results for a more realistic set of networks (higher average degree), contributing to a more realistic performance evaluation of MeasuRouting.
- The fundamental idea behind MeasuRouting is to divide traffic aggregates into subpopulations and then differentially route the traffic subpopulations based on the monitoring capacity of available routes and the relative measurement importance of the traffic subpopulations. It was observed in [4] that the way traffic aggregates are decomposed into multiple subpopulations has an impact on MeasuRouting performance. This paper extends upon [4] by introducing additional and more involved decomposition methods than those presented in [4], resulting in improved MeasuRouting performance.
- We also take a closer look at the solution computation times of MeasuRouting problems and their scalability in Section IV-A-VI. We present an approximation algorithm that allows one to tradeoff MeasuRouting performance for faster computation times.
- Finally, in Section VI, we discuss issues encountered in deploying MeasuRouting solutions in real networks and dynamic environments where both network applications and measurement objectives may keep changing.

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

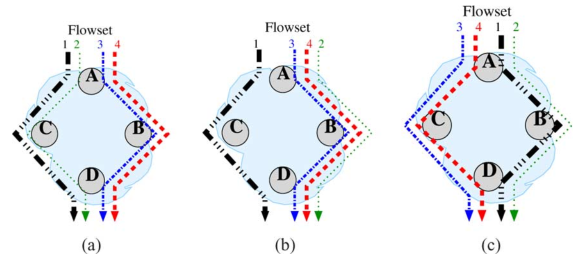

Fig. 1. Illustration of using routing to focus on a traffic subpopulation. In the above example, router B has special sampling of interest to us. To apply this sampling on Flowset 2, we can route through router B, while (b) violating, or (c) being compliant to TE policy. (a) Original. (b) Violating.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

II. MeasuRouting Overview

As mentioned in Section I, MeasuRouting must be cognizant of any implications that rerouting traffic has on 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 $\langle srcip, dstip, srcpt, dstpt, proto\rangle$ for measurement purposes. Common intradomain protocols (IGPs) like OSPF [5] and IS-IS [6] 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 subpopulations that are distinguishable from a measurement perspective but are indistinguishable from a TE perspective. MeasuRouting can, therefore, route our fine-grained measurement traffic subpopulations without disrupting the aggregate routing.
The example depicted in Fig. 1 illustrates this argument. It shows four traffic subpopulations, 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. Fig. 1(a) depicts the original routing that obeys the TE policy. Fig. 1(b) represents a routing that violates the TE policy in order to route f_2 through router B. However, if the traffic subpopulations are routed as in Fig. 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 Fig. 1(a). Then, the new routing depicted by Fig. 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 Fig. 1(b) becomes a compliant solution.

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 subpopulation, 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 subpopulations and might impose limits on the cardinality or the membership of such traffic subpopulations. 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 [7]–[9], is outside the scope of this paper. Sections 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: 1) the TE policy; 2) the topology and monitoring infrastructure deployment; and 3) the size and importance of traffic subpopulations.

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 intradomain 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)

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

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. Section IV discusses this in detail. We summarize the notations in Table I.

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:

$$\text{maximize} \quad \beta$$
$$\text{subject to}$$
$$\sum_{i:(i,j)\in E}\gamma^y_{ij} - \sum_{k:(j,k)\in E}\gamma^y_{jk} = 0 \qquad y\in\theta, j\neq \mathrm{in}_y, \mathrm{out}_y \quad (2)$$
$$\sum_{i:(i,j)\in E}\gamma^y_{ij} - \sum_{k:(j,k)\in E}\gamma^y_{jk} = -1 \qquad y\in\theta, j = \mathrm{in}_y \quad (3)$$
$$(1+\epsilon)\sigma^\Gamma \geq \sigma^\gamma \quad (4)$$
$$\gamma^y_{ij} \geq 0 \qquad y\in\theta, (i,j)\in E. \quad (5)$$

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

TABLE I
SUMMARY OF NOTATIONS

Notation	Description
$G(V,E)$	Network with V giving the set of nodes and E giving the set of edges
Θ	Set of macro-flowsets
θ	Set of micro-flowsets
Φ_x	Traffic demand for macro-flowset $x \in \Theta$
ϕ_y	Traffic demand of micro-flowset $y \in \theta$
Γ_{ij}^x	Fraction of total demand of macro-flowset $x \in \Theta$ on link $(i,j) \in E$
$\mathcal{S}_{(i,j)}$	Sampling characteristic of link $(i,j) \in E$
$\mathcal{I}_y$	Sampling utility of the micro-flowset $y \in \theta$
$\gamma_{(i,j)}^y$	Micro-flowset routing decision variable giving the fraction of total demand of micro-flowset $y \in \theta$ incident on link $(i,j) \in E$
β	Sampling Resolution Function (the objective to maximize)

by (2) and (3). LTD requires that the aggregate TE policy is not violated, as represented by (4). 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 (4) 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 nonnegative 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 Fig. 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 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:

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

Equation (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.

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 Fig. 2. We have two macro-flowsets x_1 and x_2 having the same traffic demands, i.e., $\Phi_{x_1} = \Phi_{x_2}$. Fig. 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 $x_1(\Upsilon_{x_1})$ across the path (A,C,B,D), and the micro-flowsets in macro-flowset $x_2(\Upsilon_{x_2})$ 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. However, 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

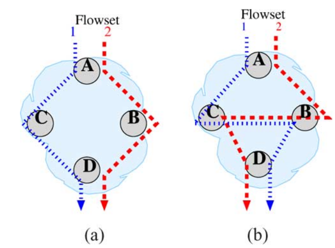

Fig. 2. MeasuRouting can violate routing semantics. (a) Original. (b) Violating.

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

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 (6)]:

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

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* $\Gamma_{ij}^{x:y\in\Upsilon_x} > 0$.

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 nonnegativity 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 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 Sections 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 $\mathrm{in}_f \in V$ and egress node $\mathrm{out}_f \in V$. $f \in \mathcal{F}$ belongs to macro-flowset x if and only if $(\mathrm{in}_f, \mathrm{out}_f) = (\mathrm{in}_x, \mathrm{out}_x)$. 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 [5] and IS-IS [6] for intradomain 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 [10] 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 total number of points earned, by routing our traffic across the given topology. This total number of points is given by the following:

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

$\upsilon^{-1}(f)$ in (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

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

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

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

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

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

The MeasuRouting formulation requires us to specify the sampling utility function for each micro-flowset. Toward 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

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

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

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

Section 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. Section 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 [11]. 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 generate sampling rates for each link following uniform distribution between 0 and 0.1. For one realization of link sampling rates and traffic demand, we repeat the experiments 50 times with different flow sampling utilities generated from the same distribution. The measurement gains are fairly stable. They fluctuate within 0.01% of the average value, and the standard deviation is around 5×10^{-5}. In Sections V and VI, we only present the average measurement gain. We introduce two more topologies (AS3257 and AS6461) with larger numbers of multipaths than [4]. Synthetical topologies generated by BRITE [12] are used to study the time complexity of MeasuRouting. We set the MeasuRouting parameters to the values given in Table II for all experiments in Section IV-A, unless specified otherwise.

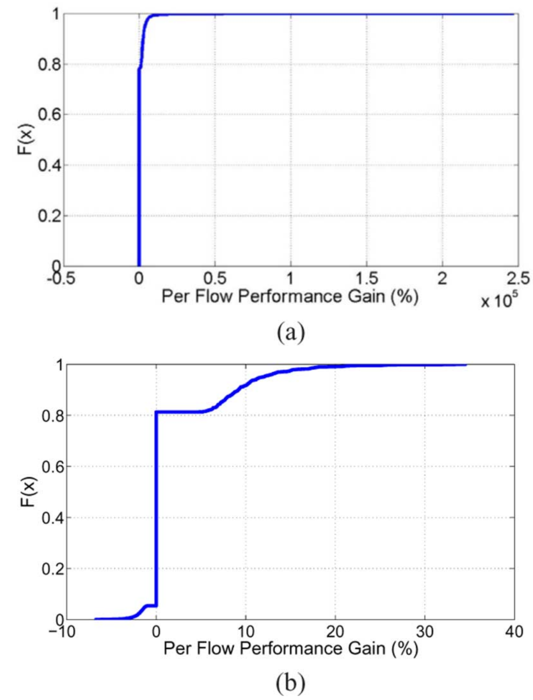

Fig. 3. CDF of per flow performance gain. (a) LTD. (b) NRL.

TABLE II
DEFAULT EXPERIMENTAL PARAMETERS

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	Flow sampling utility	Pareto ($\lambda = 2$)

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 Section III. We conduct our experiment for a 44-node and 88-link RocketFuel topology AS1221 [13]. Fig. 3 shows the cumulative distribution function (cdf) of the per-flow performance gain as described in (12). The per-flow performance gain for a flow is as high as 250 000% 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 for AS1221. 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. Fig. 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 (11), is 131%, 10%, and 9.5% for LTD, NRL, and RSR, respectively.

Consistent with our intuition in Section 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

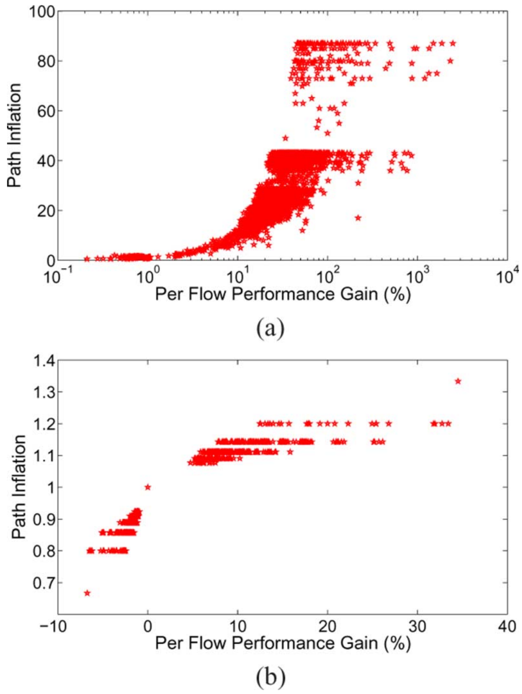

Fig. 4. Path inflation in micro-flowset routing. (a) LTD path inflation versus per-flow performance gain. (b) NRL path inflation versus per-flow performance gain.

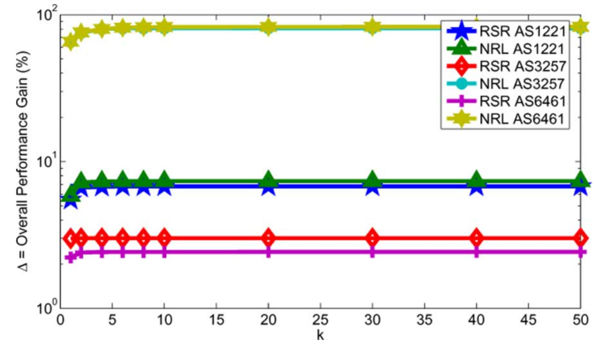

Fig. 5. MeasuRouting performance for different k.

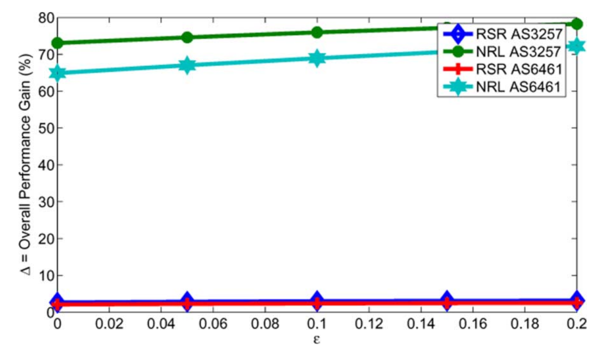

Fig. 6. MeasuRouting performance for different ϵ.

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 $y\in\Upsilon_x$. Fig. 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. Fig. 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 Section 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 Section 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. Fig. 5 shows the overall performance gain (Δ) for different values of k in three ISPs (AS1221, AS3257, AS6261). 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 5, MeasuRouting shows significant performance gain. Moreover, we see that there are diminishing returns for increasing k.

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. Fig. 6 shows how the performance improves with increasing ϵ for AS3257 and AS6461. We omit AS1221 since its performance is consistently inferior. An interesting result is that even for $\epsilon=0$, both NRL and RSR have positive Δ. In fact, even with $\epsilon=0$, we have $\Delta\approx 70\%$ for NRL in AS3257. 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 Multipath Routing: We also evaluate the effect of network size on the MeasuRouting performance gain. Fig. 7 compares the overall performance gain Δ for NRL between AS1221 (44 nodes, 88 links), AS3257 (41 nodes, 174 links), AS6461 (19 nodes, 68 links), and AS1239 (52 nodes, 168 links) [13] for different ϵ. We omit RSR since it is consistently inferior than NRL. We see that the performance gain is the largest in AS3257. This stems from our observation in Section II that making disaggregated routing decisions for different micro-flowsets corresponding to the same OD pair

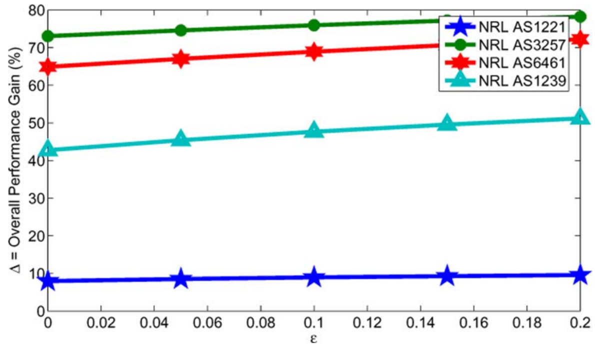

Fig. 7. MeasuRouting performance for different networks.

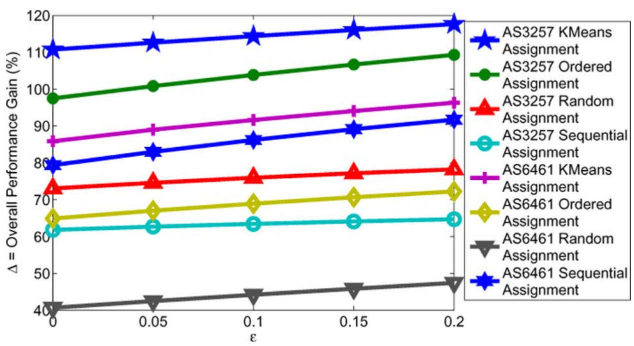

Fig. 8. MeasuRouting performance for different micro-flowset assignments.

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. AS3257 has better performance than AS1221 and AS6461 because of its larger topology. However, although AS1239 has larger topology than AS3257, its performance is inferior since it has fewer links per OD pair. The performance therefore depends on both the diversity of routing paths and topology size.

In this study, we chose ECMP for simplicity. A number of routing schemes provide a greater multiplicity of paths than ECMP [14]. MeasuRouting stands to perform much better with such routing schemes.

5) Micro-Flowset Composition Methods and Sampling Utility Diversity: Since we cluster together flows with high sampling importance (ordered flow to micro-flowset assignment), 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 NRL using other flow to micro-flowset assignment schemes for AS3257 and AS6461 in Fig. 8. We compare the Ordered Assignment with three other methods. The Random Assignment method was presented in [4]. In this paper, we introduce two additional assignment schemes. The schemes are detailed as follows.

1) Random Assignment: We 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.

TABLE III
IMPACT OF MICRO-FLOWSET UTILITY DISTRIBUTION

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%

2) KMeans Assignment: In Ordered Assignment and Random Assignment, the number of flows in two different micro-flowsets belonging to the same macro-flowset differ by no more than one. However, it is possible to have a variable number of flows in different micro-flowsets belonging to the same macro-flowset. The KMeans Assignment is one such assignment in which we cluster all flows in a macro-flowset into k subsets such that flows in each subset have similar sampling importance. Each micro-flowset is then assigned flows clustered into its corresponding subset. The objective is to minimize the intracluster variance in terms of the sampling importance of flows. We use the KMeans++ algorithm to compute the assignment [15].
3) Sequential Assignment: In this assignment, we first arrange all flows in a macro-flowset in decreasing order of their sampling importance. Starting from the first to the $(k-1)$th micro-flowset of a macro-flowset, the ith micro-flowset is assigned the i most important flows that are remaining. All flows that are not assigned are assigned to the kth micro-flowset.

We find that KMeans Assignment has the best performance compared to the other three methods. The performance of Sequential Assignment is very unstable across ISP. It is better than the Ordered Assignment for AS6461, but even worse than Random Assignment for AS3257.

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. Thus, for $M \gg k$, the overall distribution of micro-flowset sampling utilities tend 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 III 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 toward increasing the variance in the distribution of micro-flowset sampling utility. More intelligent assignment schemes may use different numbers of flows per micro-flowset to increase the diversity in the sampling utilities of micro-flowsets.

6) Computation Time and Approximation Algorithms: Reference [4] looked exclusively at the measurement performance gains of MeasuRouting. In this paper, we take a look at the computational complexity and the scalability of our MeasuRouting problem. Two major factors affecting the complexity of the optimization problem are the number of macro-flowsets (Θ) and the number of micro-flowsets (θ). In our formulation, the number

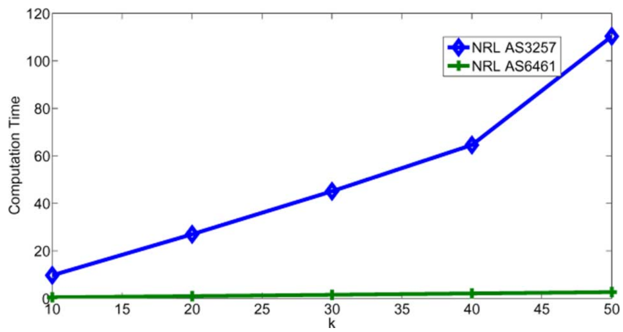

Fig. 9. Computation time with increased micro-flowset number.

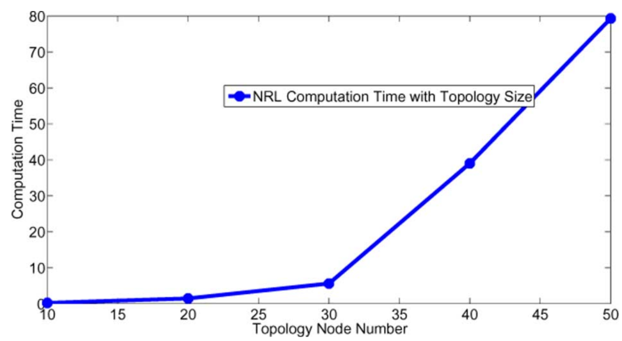

Fig. 10. Computation time with increased network size.

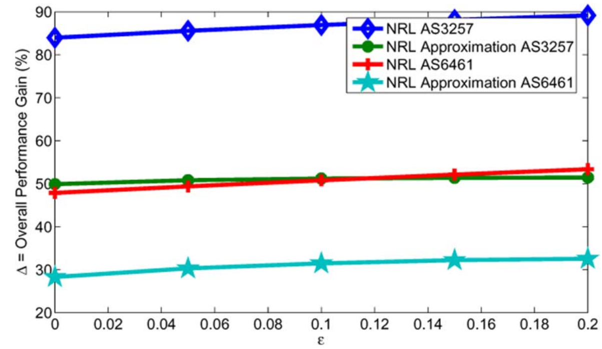

Fig. 11. Approximation MeasuRouting algorithm performance.

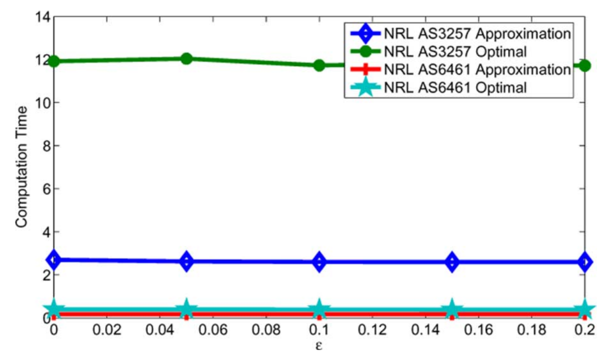

Fig. 12. Approximation MmeasuRouting algorithm computation time.

of macro-flowsets is equal to the number of OD pairs, which depends on the size of the topology. The number of micro-flowsets is a configurable parameter that represents the granularity at which macro-flowsets can be decomposed and differentially routed.

The results for computation time are in Figs. 9 and 10. For both figures, the units of y-axis are seconds, $\epsilon = 0.1$, and we use NRL routing scheme. Topologies in Fig. 10 are generated by BRITE [12], in which we fix $\frac{|E|}{|V|} = 4$ and $k = 10$. Results suggest that the computation time strongly depends on $|\gamma^y_{(i,j)}|$, the cardinality of decision variables used for linear programming. When the topology is fixed, $|\gamma^y_{(i,j)}|$ grows linearly with k. The computation time therefore increases almost linearly. However, when the topology size increases, $|E|$ grows linearly with $|V|$, and the number of OD pairs grows with $|V|^2$. The computation time therefore approximately increases with $|V|^3$.

It is important to clarify that, consistent with the objective of our paper, the gains we report represent the theoretical maximum value. The solution times are therefore for the best performance. Real networks may impose additional realistic constraints, which may reduce or increase the complexity of finding optimal solution. They may also use approximations that fit into specific requirements. In order to further reduce the computation time for linear programming, we devise simple approximation algorithms in which only the n most important and n least important flowsets are allowed to be routed differently from the original routing. Results for $n = 1$ and $k = 10$ are shown in Figs. 11 and 12. It decreases the computation time to one quarter of the original value. The performance gain β is also decreased. However, a large β still exists, which is approximately half of the original value.

B. Deep Packet Inspection Trace Capture

For the toy problem in Section 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 [16]), or specialized algorithms that are prone to errors (e.g., DPI with Bloom Filters [17]), 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. Toward 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 (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 $\mathcal{Q}(i)$, $u(i)$ is equal to zero. Equation (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 [18]. 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 Section 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 to route future traffic and 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. Fig. 13 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 nine-node Abilene network. This study is only intended to provide a proof of concept. Network operators can define their own utility functions ($u(\cdot)$) over their own fields of interest. MeasurRouting can be leveraged to enhance the quality of traces captured for their specific objectives.

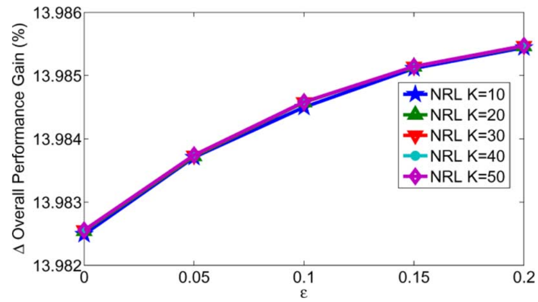

Fig. 13. MeasuRouting performance for DPI trace capture.

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 subpopulations; and 3) placing monitor agents at certain strategic network locations. We summarize existing work in these three areas.

Claffy *et al.* [19] compared various sampling approaches at both packet-based and time-based granularities [19]. Several other research efforts aim to improve estimation of "heavy-hitter" traffic volume, flow-size distributions, traffic matrices, or flow durations [20]–[27]. Recent work has demonstrated that conventional sampling techniques can obscure statistics needed to detect traffic anomalies [28] or execute certain anomaly detection algorithms [29]. All these previous works highlight the importance of being able to focus on specific traffic subpopulations. Reference [30] 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 [31], [32] 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], [33], and [34], 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. Discussions

MeasuRouting empowers 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.

A. Implementation Issues

While our current work provides theoretical bounds on the maximal performance gain through MeasuRouting, the actual implementation depends on the routing control plane. If network traffic and measurement applications remain constant, MeasuRouting can simply route/reroute important flows through the dedicated monitors. However, in reality, both measurement objectives and traffic characteristics keep changing.

The implementation of dynamic MeasuRouting involves three challenges: 1) how to dynamically assess the importance of traffic flows; 2) how to aggregate flows (and hence take a common action for them) in order to conserve routing table entries; and 3) how to achieve traffic routing/rerouting in a manner that is least disruptive to normal network performance while maximizing the measurement utility. The first challenge requires that measurement results be communicated to the routing control plane at run-time. The solution to the second challenge is application-dependent. For example, prefix-based routing is usually used to save routing table entries. The third challenge calls for simply dynamic/distributed computation of routing decisions, rather than centralized LP solver, to avoid possible computation overhead or traffic dynamicity.

MeasuRouting can be implemented over OpenFlow [8], which is a practical control mechanism for enterprise or data-center networks. The OpenFlow controller can reroute/route traffic on the fly according to specifically programmed modules. We have implemented an OpenFlow-based prototype of MeasuRouting for one measurement application: global iceberg detection and capture. Our experiments suggest that dynamic MeasuRouting is achievable in practice. The implementation details are out of the scope of this paper and can be found in [35]. We will also explore other applications in our future work.

We had mentioned that the performance of MeasuRouting is sensitive to the number of paths present between pairs of nodes. MeasuRouting leverages the relative difference in measurement capacity across multiple paths between a pair of nodes. This obviously depends upon the network topology and whether multiple paths exist at all. Additionally, the number of paths available for micro-flowset routing is a function of the number of paths used in the original routing. MeasuRouting performance will be better if the original routing uses multiple paths between a single OD pair. The implementation of multiple-path routing depends on the routing protocols. ISPs using OSPF and IS-IS generally use Equal Cost Multipath (ECMP) [5], which results in multiple paths. In fact, heuristics optimizing links weights seek to leverage ECMP to split traffic between an OD pair across multiple paths [10]. Other routing algorithms can exist that result in even more multiplicity of paths between OD pairs [14], [36].

B. Future Directions

MeasuRouting requires the knowledge of traffic importance in order to route micro-flowsets differently. We emphasize that such prior knowledge need not be accurate in practice. Most applications actually provide vague traffic statistics in the first step, and MeasuRouting serves to gain detailed information. Suppose we want to measure flow-size distribution for small/medium flows. One possible solution is to first estimate large flow identities by [23] and direct the large flows away from measurement boxes. The monitor boxes can be devised with many small-sized counters, such that small/medium flows can be more accurately maintained. In this example, there is no need to accurately measure flow sizes in the first step. Meanwhile, we observe 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.

Our current work only considered the simplest measurement applications where measurement resolution function β is linear with the sampled packets bytes. This is true for DPI trace capture since the measurement utility is directly proportional to the sampled amount of traffic. However, real applications are much more complicated. For instance, β is modeled as concave functions in [2]. Our linear objective function only provides a proof of concept. Besides the objective function, how to decide the proper flow utility also remains a problem. For instance, in certain applications such as flow-size estimation, it is less important to sample many packets from large flows, compared to equal number of small flows. It remains a problem how to determine a proper flow utility function based on flow size. Lastly, our current work did not consider how to avoid possible measurement inaccuracy. For instance, uniform packet sampling, the *de facto* implemented measurement method, introduces great inaccuracy for many applications. All these issues are application-dependent, and we will explore them in the future work.

Acknowledgment

The authors would like to thank A. 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 *Proc. ACM CoNEXT*, Toulouse, France, Oct. 2005, pp. 71–82.

[2] K. Suh, Y. Guo, J. Kurose, and D. Towsley, "Locating network monitors: Complexity, heuristics and coverage," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, vol. 1, pp. 351–361.

[3] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran, "Reformulating the monitor placement problem: Optimal network-wide sampling," in *Proc. ACM CoNEXT*, Lisboa, Portugal, Dec. 2006, Article no. 5.

[4] S. Raza, G. Huang, C.-N. Chuah, S. Seetharaman, and J. P. Singh, "MeasuRouting: A framework for routing assisted traffic monitoring," in *Proc. IEEE INFOCOM*, San Deigo, CA, Mar. 2010, pp. 1–9.

[5] "OSPF," The Internet Society, RFC 2328, 1998 [Online]. Available: http://tools.ietf.org/html/rfc2328

[6] "IS-IS," The Internet Society, RFC 1142, 1990 [Online]. Available: http://tools.ietf.org/html/rfc1142

[7] C. Wiseman, J. Turner, M. Becchi, P. Crowley, J. DeHart, M. Haitjema, S. James, F. Kuhns, J. Lu, J. Parwatikar, R. Patney, M. Wilson, K. Wong, and D. Zar, "A remotely accessible network processor-based router for network experimentation," in *Proc. ACM/IEEE ANCS*, San Jose, CA, Nov. 2008, pp. 20–29.

[8] "The OpenFlow Switch Consortium," Stanford University, Stanford, CA [Online]. Available: http://www.openflowswitch.org

[9] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, "The Click modular router," in *Proc. ACM SOSP*, Charleston, SC, Dec. 1999, pp. 217–231.

[10] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, Tel-Aviv, Isreal, Mar. 2000, vol. 2, pp. 519–528.

[11] A. Medina, N. Taft, S. Battacharya, C. Diot, and K. Salamatian, "Traffic matrix estimation: Existing techniques and new directions," in *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002, pp. 161–174.

[12] "BRITE," Boston University, Boston, MA [Online]. Available: http://www.cs.bu.edu/BRITE/

[13] N. Spring, R. Mahajan, and T. Anderson, "The causes of path inflation," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 113–124.

[14] Y. Ohara, S. Imahori, and R. V Meter, "MARA: Maximum alternative routing algorithm," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 298–306.

[15] A. David and V. Sergei, “K-means++: The advantages of careful seeding,” in *Proc. ACM–SIAM SODA*, Philadelphia, PA, 2007, pp. 1027–1035.
[16] F. Yu, R. H. Katz, and T. V. Lakshman, “Gigabit rate packet pattern-matching using TCAM,” in *Proc. IEEE ICNP*, Berlin, Germany, Oct. 2004, pp. 174–183.
[17] S. Dharmapurikar, P. Krishnamurthy, T. Sproull, and J. Lockwood, “Deep packet inspection using parallel bloom filters,” *IEEE Micro*, vol. 24, no. 1, pp. 44–51, Jan.–Feb. 2004.
[18] “The Internet2 network,” 2009 [Online]. Available: http://www.internet2.edu
[19] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, “Application of sampling methodologies to network traffic characterization,” in *Proc. ACM SIGCOMM*, San Francisco, CA, Sep. 1993, pp. 194–203.
[20] B.-Y. Choi and S. Bhattacharyya, “On the accuracy and overhead of Cisco sampled netflow,” in *Proc. ACM SIGMETRICS LSNI*, Banff, Canada, Jun. 2005, pp. 18–23.
[21] N. Duffield and C. Lund, “Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure,” in *Proc. ACM SIGCOMM IMC*, Miami Beach, FL, 2003, pp. 179–191.
[22] C. Estan, K. Keys, D. Moore, and G. Varghese, “Building a better Netflow,” in *Proc. ACM SIGCOMM*, Portland, OR, Sep. 2004, pp. 245–256.
[23] C. Estan and G. Varghese, “New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice,” *Trans. Comput. Syst.*, vol. 21, no. 3, pp. 270–313, 2003.
[24] N. Hohn and D. Veitch, “Inverting sampled traffic,” in *Proc. ACM SIGCOMM IMC*, Miami, FL, Oct. 2003, pp. 222–233.
[25] R. Kompella and C. Estan, “The power of slicing in Internet flow measurement,” in *Proc. ACM SIGCOMM IMC*, Berkeley, CA, Oct. 2005, p. 9.
[26] A. Kumar, M. Sung, J. Xu, and J. Wang, “Data streaming algorithms for efficient and accurate estimation of flow size distribution,” in *Proc. ACM SIGMETRICS*, New York, NY, Jun. 2004, pp. 177–188.
[27] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, “An information-theoretic approach to traffic matrix estimation,” in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 301–312.
[28] 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 *Proc. ACM SIGCOMM IMC*, Rio de Janeiro, Brazil, Oct. 2006, pp. 147–152.
[29] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, “Is sampled data sufficient for anomaly detection?,” in *Proc. ACM SIGCOMM IMC*, Rio de Janeiro, Brazil, Oct. 2006, pp. 165–176.
[30] A. Ramachandran, S. Seetharaman, N. Feamster, and V. Vazirani, “Fast monitoring of traffic subpopulations,” in *Proc. ACM SIGCOMM IMC*, Vouliagmeni, Greece, Oct. 2008, pp. 257–270.
[31] H. V. Madhyastha and B. Krishnamurthy, “A generic language for application-specific flow sampling,” *Comput. Commun. Rev.*, vol. 38, no. 2, pp. 5–16, 2008.
[32] L. Yuan, C.-N. Chuah, and P. Mohapatra, “ProgME: Towards programmable network measurement,” in *Proc. ACM SIGCOMM*, Kyoto, Japan, Aug. 2007, pp. 97–108.
[33] M. R. Sharma and J. W. Byers, “Scalable coordination techniques for distributed network monitoring,” in *Proc. PAM*, Boston, MA, Apr. 2005, pp. 349–352.
[34] 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 *Proc. USENIX NSDI*, San Francisco, CA, Apr. 2008, pp. 233–246.
[35] G. Huang, S. Raza, S. Seetharaman, and C.-N. Chuah, “Dynamic measurement-aware routing in practice,” *IEEE Network*, vol. 25, no. 3, pp. 29–34, May-Jun. 2011.
[36] J. R. D. Xu and M. Chiang, “Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering,” in *Proc. IEEE INFOCOM*, 2008, pp. 466–474.

Saqib Raza (M’04) received the B.S. degree from the Lahore University of Management Sciences, Lahore, Pakistan, in 2004, and the M.S. and Ph.D degrees from the University of California, Davis, in 2007 and 2010, respectively, all in computer science.

He is a Software Engineer with the Data Center Switching Technology Group, Cisco Systems, San Jose, CA.

Guanyao Huang (M’11) received the undergraduate and post-graduate degrees in electrical and computer engineering from the University of Science and Technology of China, Hefei, China, in 2004 and 2007, respectively, and is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of California, Davis.

His current research focuses on network measurement and anomaly detection.

Chen-Nee Chuah (S’92–M’01–SM’06) received the B.S. degree in electrical engineering from Rutgers University, New Brunswick, NJ, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1997 and 2001, respectively.

She is a Professor of electrical and computer engineering with the University of California, Davis. Her research interests lie in the area of communications and computer networks, with emphasis on Internet measurements, network management, cybersecurity, online social networks, and vehicular ad hoc networks.

Srini Seetharaman (M’06) received the Master’s degree from The Ohio State University, Columbus, in 2001, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, in 2007, both in computer science.

He is a member of the Clean Slate Lab, Stanford University, Stanford, CA, and a Senior Research Scientist with Deutsche Telekom R&D Labs, Los Altos, CA. His research interests include networking architectures and protocols, overlay networks, traffic/network monitoring, and green technologies.

Jatinder Pal Singh (M’05) received the B.S. degree from the Indian Institute of Technology, Delhi, India, in 2000, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 2002 and 2005, respectively, all in electrical engineering.

He is a Consulting Assistant Professor with the Department of Electrical Engineering, Stanford University. His current research focus is on next-generation mobile networks and smart-phone platforms, multihomed device communication, energy-efficient location sensing and positioning, and evolving telecommunication networks and architectures.