Measurement-Aware Monitor Placement and Routing: A Joint Optimization Approach for Network-Wide Measurements

Guanyao Huang, Chia-Wei Chang, Chen-Nee Chuah, and Bill Lin

Abstract—**Network-wide traffic measurement is important for various network management tasks, ranging from traffic accounting, traffic engineering, network troubleshooting to security. Previous research in this area has focused on either deriving better monitor placement strategies for fixed routing, or strategically routing traffic sub-populations over existing deployed monitors to maximize the measurement gain. However, neither of them alone suffices in real scenarios, since not only the number of deployed monitors is limited, but also the traffic characteristics and measurement objectives are constantly changing.**

This paper presents an MMPR (Measurement-aware Monitor Placement and Routing) framework that jointly optimizes monitor placement and dynamic routing strategy to achieve maximum measurement utility. The main challenge in solving MMPR is to decouple the relevant decision variables and adhere to the intra-domain traffic engineering constraints. We formulate it as an MILP (Mixed Integer Linear Programming) problem and propose several heuristic algorithms to approximate the optimal solution and reduce the computation complexity. Through experiments using real traces and topologies (Abilene [1], AS 6461 [2], and GEANT [3]), we show that our heuristic solutions can achieve measurement gains that are quite close to the optimal solutions, while reducing the computation times by a factor of 23X in Abilene (small), 246X in AS 6461 (medium), and 233X in GEANT (large), respectively.

Index Terms—**Traffic measurement, routing, traffic engineering.**

I. INTRODUCTION

GIVEN the sheer size and complexity of the Internet today and its increasingly important role in modern-day society, there is a growing need for high-quality network traffic measurements to better understand and manage the network. Obtaining accurate network-wide traffic measurement in an efficient manner is a daunting task given the multi-faceted challenges. First, there is an inherent lack of fine-grained measurement capabilities in the Internet architecture. Second, the rapidly increasing link speeds make it impossible for every router to capture, process, and share detailed packet information. Earlier work on traffic monitoring has focused on improving single-point measurement techniques, such as

G. Huang and C.-N. Chuah are with the University of California at Davis, CA USA (e-mail: gyhuang@ucdavis.edu).

C.-W. Chang and B. Lin are with the University of California at San Diego, CA USA.

sampling approaches [4, 5], estimation of heavy-hitters [6], and methods to channel monitoring resources on traffic sub-populations [7, 8]. To achieve network-wide coverage, previous studies have focused on the optimal deployment of monitors across the network to maximize the monitoring utility (as determined by the network operator) with given traffic routing [9–11]. The optimal placement for a specific measurement objective typically assumes a priori knowledge about the traffic characteristics. However, both traffic characteristics and measurement objectives can dynamically change over time, potentially rendering a previously optimal placement of monitors suboptimal. For instance, a flow of interest can avoid detection by not traversing the deployed monitoring boxes. The optimal monitor deployment for one measurement task might become suboptimal once the objective changes.

To address the limitation mentioned above, MeasuRouting [12] was recently proposed to strategically/dynamically route important traffic over fixed monitors such that it could be best measured. Using intelligent routing, it can cope with the changes of traffic patterns or measurement objectives to maximize measurement utility while meeting existing intra-domain traffic engineering (TE) constraints,

e.g., achieving even load distribution across the network, or meeting Quality of Service (QoS) constraints. It is oblivious of the monitor placement problem. The key idea is that the routes of important and unimportant flows can be exchanged to achieve better measurement and load balancing. However, MeasuRouting is based on the assumption that monitor locations have already been decided a priori and fixed. It does not consider the flexibility of deploying new monitors and replacing old ones, or altering the existing monitor placement strategies.

In practice, current routers deployed in operational networks are already equipped with monitoring capabilities (e.g., Netflow [13], Openflow [14]). Network operators would not turn on all these functionalities because of their associated expensive operation cost [9–11] and measurement redundancy [15], and hence there are potentially hundreds of monitoring points to choose from to achieve network-wide measurements. Given routing could be changed dynamically to aid measurement, the optimal monitor selection/placement strategies may also change to take advantage of this new degree of freedom. Therefore, previous approaches that treat monitor placement and routing as two separate problems may

be sub-optimal (as demonstrated in Section 2 with an example scenario). This naturally leads to the following open question: *Given a network where all links can be monitored, which monitors should be activated and how to strategically route traffic sub-populations over those planned monitors such that both the measurement gain is maximized and the limited resources is best utilized.*

In this paper, we propose an MMPR (Measurement-aware Monitor Placement and Routing) framework that jointly optimizes monitor placement and traffic routing strategy, given traffic characteristics and monitor capacities as inputs. In our framework, the optimal routing strategy is determined for each flowset, which is defined to be any aggregation of flows which share the same ingress/egress routers and have the same routing decision. The goal is to maximize the overall measurement utility, which quantifies how well each individual flow is monitored (e.g., how many bytes or packets are sampled), weighted by its importance. We strive to adhere to the existing intra-domain traffic engineering (TE) constraints such that we maintain similar load distributions in the network (e.g., maximum link utilization) as in default routing case. We also attempt to constrain measurement resources by activating no more than K monitors in arbitrary links.

The properties of monitors and importance of flows in the paper are modeled in a very generic form such that our framework can be applied to a wide variety of measurement scenarios. We assume that the dynamic traffic/measurement changes will stay for long enough time for us to re-optimize monitor placement and flowset routing. Implementation issues for continuous measurement are discussed in Section VII or left as future work. We highlight our contributions as follows:

. We formulate the MMPR problem as an MIQP (Mixed Integer Quadratic Programming) problem, and show how it could be reformulated as a standard MILP (Mixed Integer Linear Programming) problem by decoupling the two key decision variables.

. We investigate several approximate solutions that can approach the performance of the optimal MILP solution, but yet they require dramatically shorter computation times. Our heuristic algorithms include K-Best, Successive Selection, Greedy and Quasi-Greedy.

. We perform detailed simulation studies using real traces and topologies from Abilene [1], AS6461 [2], and GEANT [3]. Our results show that the optimal MMPR solution can achieve measurement gains up to a factor 1.76X better when compared to baseline cases (optimal Placement-only or MR(MeasuRouting)-only). We also show that our heuristic algorithms can achieve measurement utilities that are quite close to the optimal solution, while reducing computation times by a factor of 23X in Abilene, 246X in AS6461, and 233X in GEANT, compared with the MILP (optimal) solution.

The rest of the paper is organized as follows. Section II illustrates through a motivating example the benefits of a joint optimization approach that considers both monitor placement and traffic routing together. Section III formulates the MMPR problem, and Section IV presents our heuristic solutions. Section V presents detailed experimental results using our
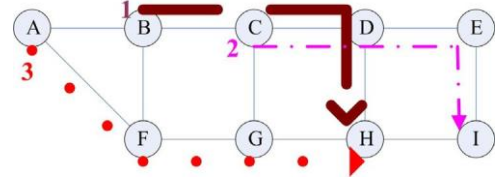

Fig. 1. MMPR motivational example.

proposed methods, and Section VI outlines related work. Finally, Section VII discusses practical implementation issues and concludes the paper.

II. MOTIVATING EXAMPLE

In this section, we showcase the importance of both monitor placement and traffic routing through an illustration. Consider the topology in Figure 1. We define a flow based on the five tuple < $srcip, dstip, srcpt, dstpt, proto$ >. We assume that due to budget considerations, only one monitor is allowed to be deployed in any one of the 12 links. The network operator wants to identify the best monitor location and the best routing for "important" flows, to achieve maximum measurement gain, i.e., measuring as many important flows as possible. At the same time, the operator wants to ensure that the monitor placement and any routing changes have least impact on existing QoS metric, which is defined as the "average path length" of every flow.

Initially there are two important flows, flow 1 and 2, with their default routing show in Figure 1. Obviously the optimal monitor location is on link $C \rightarrow D$ where the two important flows traverse. There are many other unimportant flows (not shown in the figure) from each OD (origin-destination) pair. All of the N unimportant flows (including flow 3) use shortest path routing. Suppose their average path length is ζ.

Suppose now flow 3 becomes important over time, with its default route $A \rightarrow F \rightarrow G \rightarrow H$. With the current monitor placement (previously determined to be optimal), flow 3 will not be monitored at all. In order to capture this flow, a second monitor (additional resources) will be needed along the path $A \rightarrow F \rightarrow G \rightarrow H$, if the routing remain unchanged. Alternatively, a dynamic routing approach like MeasuRouting would redirect flow 3 through link $C \rightarrow D$ (assuming the resulting link utilization is below a desired threshold). However, the detour increases path length for flow 3 from 3 to 4. Since every other flow uses shortest path routing, the average path length increases from $\frac{N\zeta+6}{N+2}$ to $\frac{N\zeta+7}{N+2}$, which clearly has a negative impact on the QoS metric.

Instead, it would be better to move the monitor from $C \rightarrow D$ to link $G \rightarrow H$, and redirect the flow 1 and 2 both through link $G \rightarrow H$. The new routes for flow 1 and 2 can be $B \rightarrow F \rightarrow G \rightarrow H$ and $C \rightarrow G \rightarrow H \rightarrow I$, respectively. As such, no flow has increased its path length, i.e. average path length remains $\frac{N\zeta+6}{N+2}$. All flows can be monitored with only one monitor (without additional resources).

One other practical concern is that the redirection of flow 1 and 2 may overload link $G \rightarrow H$. This can be simply avoided by switching flow 2 with another unimportant flow from B to H, as long as that flow has equal traffic amount

HUANG *et al.*: MEASUREMENT-AWARE MONITOR PLACEMENT AND ROUTING: A JOINT OPTIMIZATION APPROACH FOR NETWORK-WIDE ...

51

and was originally routed through $B \rightarrow F \rightarrow G \rightarrow H$. Flow 3 can be similarly treated by switching with a flow originally routed as $C \rightarrow G \rightarrow H \rightarrow I$. MeasuRouting [12] has already shown ways to switch flows for better measurement. In our situation, by switching flow 1 and 2 with other unimportant flows, both average path length and link load can be preserved at initial conditions. The same scenario may lead to different optimal solutions (the new placement location and new routes) with other TE metric definitions. The problem becomes more complicated for larger topology, with more important flows and different TE metrics.

The example above reveals that MeasuRouting without considering changing monitor placement (referred to as MeasuRouting- or MR-only) may become suboptimal. Similarly, changing the optimal monitor placement alone (referred to as Placement-only) without the flexibility in re-routing may be infeasible without introducing additional measurement resources (e.g., adding a second monitor in this example). A better monitor placement combined with strategic routing can achieve optimal solution while meeting the QoS or TE constraints. This motivates us to formulate the joint optimization problem of both monitor placement and traffic routing under the MMPR framework and propose optimal solutions that achieve best measurement utility with limited monitor resources. We will later compare the performance of optimal MMPR solution with MR-only and Placement-only in Figure 2. In the example above, Placement-only strategy will miss flow 3 completely. Both MR-only and MMPR can monitor all flows. However, MR-only increases the average path length (QoS metric) to $\frac{N\zeta+7}{N+2}$, which is undesirable, while MMPR reduces it to $\frac{N\zeta+6}{N+2}$

The main focus on this paper is to provide a *theoretical* framework for MMPR problem and examine the cost/performance trade-offs for the optimal solution and a variety of heuristic approaches. There are several practical issues which remain to be addressed in order to realize MMPR solutions. For example, MMPR assumes prior knowledge of traffic importance, which is usually inaccurate in practice. All the related implementation issues will be discussed in Section VII.

III. MMPR Framework

We now present a formal framework for MMPR in the context of a centralized architecture, which jointly optimizes monitor placement and traffic routing assuming it has global knowledge of a) the network topology, b) the size and importance of traffic sub-populations, c) the monitor capability, and d) the TE policy.

A. Definition

$G(V, E)$ represents our network, where V is the set of nodes and E is the set of directed links. $M = |V|$ is the total number of links. An OD pair represents a set of flows between the same pair of ingress/egress nodes for which an aggregated routing placement is given. The set of all $|V| \times |V - 1|$ OD pairs is given by Θ. Γ^x_{ij} denotes the fraction ($[0, 1]$) of the traffic demand belonging to OD pair x placed along link (i, j). $\{\Gamma\}^{x\in\Theta}_{(i,j)\in E}$ is an input to the MMPR problem and represents

our *original routing*. We assume $\{\Gamma\}^{x\in\Theta}_{(i,j)\in E}$ is a *valid* routing, i.e. flow conservation constraints are not violated and it is compliant with the network TE policy.

An OD pair may consist of multiple flows where some of them have higher measuring importance than others. The purpose of traffic measurement is to capture those important flows as much as possible. However, it is impractical to enforce individual routing decision for each flow. On the other hand, flows are aggregated as flowsets according to flow semantics, e.g: prefix based routing. In this paper, we define flowset to be any aggregation of flows which share the same ingress/egress routers and have the same routing decision. We use θ to denote the set of mutually exclusive flowsets and Y_x to denote the set of flowsets that belongs to the OD pair x. Each flow is assigned to one flowset in θ.

We denote the fraction of traffic demand of flowset y placed along link (i,j) as γ^y. $\{\gamma\}^{y\in\theta}$ represents our flowset routing decision variables and is the set of decision variables of the MMPR problem. According to this definition, flows belonging to the same flowset y should have the same routing. We denote $\{\Phi\}_{x\in\Theta}$ and $\varphi_y\}_{\in\theta}$ to be the traffic demands (e.g., the sizes) for the OD pair Θ and flowset θ, respectively. It follows that $\Phi_x = \sum_{y\in\Upsilon_x} \varphi_y$. φ_y denotes the measurement utility of the flowset y. This is a generic metric that defines the importance of measuring a flowset, which is related to the importance of its individual flows.

In this paper, we assume traffic measurements are conducted on links. We define our measurement infrastructure and measurement requirement in abstract terms. $\{S\}_{(i,j)\in E}$ denotes the measurement characteristic of all links, i.e. the ability of a link to measure traffic. For example, $S_{(i,j)}$ can be equal to p_{ij}, the sampling rate of link (i, j). Since packet sampling is the de facto deployed measurement method, we will use p_{ij} and $\{S\}_{ij}$ interchangeably to denote the measurement ability of each link, and we discuss other possible measurement functions in Section III-C. In summary, $\{S\}_{(i,j)\in E}$ and $\varphi_{y\in\theta}$ are inputs given to our MMPR problem. Another input to the MMPR problem is K, the maximum number of monitors that would be turned on inside the network. In this paper, monitors can be turned on any of the M links. The (0,1) boolean variable u_{ij} is used to denote the placement strategy. Finally, we use a *measurement resolution function* (β) to characterize the overall performance of traffic measurement. β assigns a real number representing the monitoring effectiveness of flowset routing, flowset utility, and monitor placement strategy for given measurement characteristics. The objective of MMPR is to maximize β. Notations are summarized in Table I.

$$\beta : (\{\gamma\}^y_{(i,j)}, \{S\}_{(i,j)}, \{I\}_y, u_{(i,j)}) \rightarrow R \qquad (1)$$

B. Formulation

In our problem, we can formulate the measurement gain through two kinds of popular reward models [10]. Let utility function T_y denote the benefit gained by monitoring flowset y. We assume that there is no additional benefit gained by repeatedly monitoring the same traffic. Thus T_y can be

TABLE I
SUMMARIZATION OF NOTATIONS

Notation	Description
x	OD pair
y	flowset
Θ	set of OD pairs
φ	set of flowsets
Φ_x	traffic demands of the OD pair x
φ_y	traffic demands of the flowset y
Γ_{ij}^{x}	original routing for OD pair x
$S_{(i,j)}, P_{(i,j)}$	measurement characteristic of link (i,j)
$\mathbf{I}_y$	measurement utility of the flowset y
T_y	measurement gain of flowset y
$\gamma_{(i,j)}^{y}$	routing decision variable for flowset y
$u_{(i,j)}$	monitor placement variable for link (i, j)
β	optimization objective

expressed in either of two ways:

$$T_y = 1 - \sum_{(i,j)\in E} (1 - p_{ij}u_{ij}\gamma_{ij}^{y}) \qquad (2)$$

$$T_y = \sum_{(i,j)\in E} p_{ij}\, u_{ij}\, \gamma_{ij}^{y} \qquad (3)$$

Equation (3) approximates equation (2) if $p_{ij}u_{ij}\gamma_{ij}^{y}$ is very small. This is true for most core-networks since the sheer traffic volume/speed prohibits high rate measurement. Equation (2) models the case where monitors independently sample flows, while in Equation (3), monitors measure non-overlapping traffic. This can be achieved by CSamp [15] like methods, in which disjoint hash-based filters are placed before flows get sampled. In this paper, we use the later reward model since it is linear, allowing us to better compare the various MMPR solutions.

$$\text{Maximize } \beta \qquad (4)$$

$$\beta = \sum_{y\in\theta} \mathbf{I}_y T_y \qquad (5)$$

$$= \sum_{y\in\theta} \sum_{(i,j)\in E} \mathbf{I}_y p_{ij}\, u_{ij}\, \gamma_{ij}^{y} \qquad (6)$$

$$\gamma_{ij}^{y} \geq 0, \; \forall\, y \in \theta, (i,j) \in E \qquad (7)$$

$$u_{ij} \in \{0, 1\}, \; \forall\, (i,j) \in E \qquad (8)$$

In our model, T_y is the summation of the product of p_{ij}, γ_{ij}^{y} and u_{ij}. Therefore the objective function β is related to the product of two decision variables u_{ij} and γ_{ij}^{y}, and the optimization problem falls into the MIQP (Mix Integer Quadratic Programming) category. In order to avoid quadratic programming, we introduce z_{ij}^{y} to decouple u_{ij}, γ_{ij}^{y} by Equations (10) and (11). It is easy to see their equivalence. When $u_{ij} = 0$, $z_{ij}^{y} = 0$ from (10); and when $u_{ij} = 1$, $z_{ij}^{y} = \gamma_{ij}^{y}$ from (11).

$$z_{ij}^{y} = \gamma_{ij}^{y} \times u_{ij} \qquad (9)$$

$$0 \leq z_{ij}^{y} \leq u_{ij} \qquad (10)$$

$$\gamma_{ij}^{y} + u_{ij} - 1 \leq z_{ij}^{y} \leq \gamma_{ij}^{y} \qquad (11)$$

After we substitute (10-11) to (6), the formulation becomes MILP (Mixed Integer Linear Programming) instead of MIQP:

$$\text{Maximize } \beta \qquad (12)$$

$$\beta = \sum_{y\in\theta} \sum_{(i,j)\in E} \mathbf{I}_y p_{ij}\, z_{ij}^{y} \qquad (13)$$

$$0 \leq z_{ij}^{y} \leq u_{ij} \qquad (14)$$

$$\gamma_{ij}^{y} + u_{ij} - 1 \leq z_{ij} \leq \gamma_{ij}^{y} \qquad (15)$$

$$\gamma_{ij}^{y} \geq 0, \forall\, y \in \theta, (i,j) \in E \qquad (16)$$

$$u_{ij} \in \{0, 1\}, \forall\, (i,j) \in E \qquad (17)$$

We set the maximum number of allowed monitors to be no more than K:

$$\sum_{(i,j)\in E} u_{ij} \leq K \qquad (18)$$

After introducing MMPR, the new routing should not violate the TE metric (e.g., maximum link utilization) by more than a certain threshold, as compared with original routing. We use σ^{Γ} and σ^{γ} to denote TE metric of original routing and new routing, respectively. We introduce a threshold s, which bounds the violation of TE metric.

$$\sigma^{\gamma} \leq (1 + s)\sigma^{\Gamma} \qquad (19)$$

The traffic constraints can be formulated as follows:

$$\sum_{i:(i,j)\in E} \gamma_{ij}^{y} - \sum_{k:(j,k)\in E} \gamma_{jk}^{y} = 0 \qquad y \in \theta, j \neq in_y, out_y \qquad (20)$$

$$\sum_{i:(i,j)\in E} \gamma_{ij}^{y} - \sum_{k:(j,k)\in E} \gamma_{jk}^{y} = -1 \qquad y \in \theta, j = in_y \qquad (21)$$

$$\sum_{i:(i,j)\in E} \gamma_{ij}^{y} - \sum_{k:(j,k)\in E} \gamma_{jk}^{y} = 1 \qquad y \in \theta, j = out_y \qquad (22)$$

In MMPR, to maximize β, important flowsets might get repeatedly routed through monitors. In reality, loop-free routing is desirable to avoid huge delays. MeasuRouting [12] proposed two methods (RSR and NRL) to provide candidate routes which are loop-free. They pre-calculate allowable acyclic paths for each OD pair. The optimization problem then selects the best routes from these candidates. In this paper, we borrow the idea of NRL (No Routing Loops MeasuRouting [12]). It allows us to select paths other than original routing Γ^{x}, by introducing $\Psi_{x:y\in\Upsilon}$:

$$\gamma_{ij}^{y} = 0 \qquad y \in \theta, (i,j) \notin \Psi_{x:y\in\Upsilon_x} \qquad (23)$$

Equation (23) states that only links included in $\Psi_{x:y\in\Upsilon_x}$ may be used for routing flowset y. We use the heuristic algorithm in [12] to construct these paths. For each OD pair, it iteratively adds new link to $\Psi_{x:y\in\Upsilon_x}$ in decreasing order of sampling rate, as long as it does not introduce any loops.

C. Extensions

In this section, we extend our formulation and discuss some related issues. First, our formulation only introduces parameter K to bound the number of monitors, without formulating any detailed cost functions. In reality, the operation cost of

monitors also depend on their sampling rates. Let f_{ij} (and g_{ij}) denote the unit monitor deployment (and operation) cost at link (i, j), and B (and C) represents the maximum budget for deployment (and operation) cost. We could add these two constraints as follows:

$$\sum_{(i,j) \in E} u_{ij} \times f_{ij} \leq B \qquad (24)$$

$$\sum_{y \in \theta} \sum_{(i,j) \in E} p_{ij} u_{ij} \gamma^y_{ij} \times g_{ij} \times \varphi_y \leq C \qquad (25)$$

We can also treat the sampling rate, p_{ij}, as another decision variable if the operator tends to better configure the operation cost of monitors. The new problem becomes complicated since both the optimization objective (13) and the constraint (25) become quadratic. In reality, it is difficult to compare and tune the settings for different measurements. It is impractical to mathematically compare these costs with measurement gains. Instead, we formulate the fundamental situation where the cost is only related to the number of monitors, and each monitor has fixed configuration. It is equivalent to the case where f_{ij} is identical to all of the monitors.

Second, our formulation is based on "uniform" measurement. That means, each monitor will treat any traffic that traverse it equally. The objective function then becomes linear. In reality, more sophisticated measurements can intelligently adapt to different flows [7, 8]. Because of this, the measurement gain function β might become nonlinear, or, other parameters are needed to reflect the difference in how flows are measured by the same monitor. We will explore different measurement methods (e.g.: flow sampling, flexsample [7], etc) in our future work. Our current formulation applies to any measurement scheme where all packets are treated equally by the same monitor. For example, DPI (deep packet inspection) can be simply viewed as $p_{ij} = 1$.

Finally, our formulation can be easily extended to Placement-only problem. It is defined to maximize β with respect to the decision variable u_{ij} only, while flowsets are routed along their original routes:

$$\text{Maximize } \beta \qquad (26)$$

$$\beta = \sum_{y \in \theta} \sum_{(i,j) \in E} \mathbf{I}_y p_{ij} \, u_{ij} \, \Gamma^x_{y \in \Upsilon_x} \qquad (27)$$

$$u_{ij} \in \{0, 1\}, \forall (i, j) \in E \qquad (28)$$

IV. MMPR SOLUTIONS

In this section, we first describe the optimal MMPR solution by solving the associated MILP problem in Section IV-A. Since the time-complexity of MILP is generally NP-hard, we propose several heuristic solutions to approximate the optimal performance: "K-Best", "Successive Selection", "Greedy" and "Quasi-Greedy". It is easy to see that MMPR becomes a LP (Linear Programming) problem if the monitor placement strategy is given (i.e., with fixed u_{ij}). Therefore, all of our heuristic solutions tend to decide the monitor locations first. They all start from an initial configuration in which all M monitors are fully deployed. We refer to this initial configuration as the "All-On" stage.

In particular, we first propose K-Best (Section IV-B), the most lightweight algorithm among our heuristic methods. It directly disables M_K monitors according to their performance in the All-On case, based on some ranking metrics (e.g. traffic amount, topology, link capacity, etc). We then propose several increasingly complex algorithms; "Successive Selection", "Greedy", and "Quasi-Greedy", that iteratively select monitors to disable, based on the planned monitor placement strategy decided from the previous iteration. This process is repeated until only K monitors are left. The Successive Selection algorithm (Section IV-C) uses the same heuristic metrics as K-Best to successively disable monitors at each iteration. The Greedy and Quasi-Greedy (Section IV-D) algorithms are the most complex since they select monitors to disable in each iteration by testing them.

All the proposed heuristics seek the least important monitors (in accordance to some metric) to disable and then maximize the measurement gain β. They all start from the All-On stage and gradually exclude monitors until K are left. Our approach is complementary to previous work on monitor placement [9, 10] that starts with zero monitors and gradually add new monitors until there are K of them. The reason for our design is the following: whenever monitors are chosen, the best routing for the flowsets needs to be re-calculated, which may change substantially after new monitors are introduced. Instead of testing possible placement and flowset routing, it is more straightforward to disable unimportant monitors from a stage with more enabled monitors. We therefore propose algorithms that start from the All-On stage.

A. Optimal Solution

The optimal solution searches for the best γ^y_{ij} and u_{ij} assignments for the MMPR problem. The MMPR formulation is an MILP problem since u_{ij} is a binary decision variable and γ^y_{ij} is a continuous decision variable. There is a variety of optimization tools that we can leverage. In particular, the optimal solution can be found using an MILP solver (e.g., CPLEX [16]). We refer to this solution as "Optimal". For small to medium size networks, the optimal MMPR solution can be readily found. However, given that MILP problems are in general NP-hard, the solvers are not fast enough for large networks.

B. K-Best Algorithm

The K-best algorithm disables M_K monitors in a single step, based on their performance in the All-On stage. It starts from the All-On configuration and calculates the maximum achievable β and optimal traffic assignment γ^y_{ij}. It then ranks all monitors in ascending order using one of the following metrics and directly disables the top $M K\_$ monitors:

· *Least-utility* ($\sum_y p_{ij} \gamma^y_{ij}$) We disable the monitors with the least measurement utilities. Since measurement utility is the same as our optimization objective, we expect this metric will achieve the best β.

· *Least-traffic* ($\sum_y \gamma^y_{ij} \varphi_y$) The intuition behind this metric is that the monitors with the least amount of traffic passing through them are also expected to have the least contribution to the overall measurement utility.

- *Least-importance* ($\sum_{y} \gamma_{ij}^{y} I_{y}$). This metric only considers the flowset importance, regardless of the sampling rate. It treats all flowset with the same traffic demand and all monitors with the same sampling rate.

- *Least-rate* (p_{ij}). We disable monitors with the least sampling rates since they are the least capable.

- *Least-neighbor* ($\sum_{k:(k,i)\in E} 1 + \sum_{k:(j,k)\in E} 1$). From a topology perspective, the monitors that are the least connected are also likely to provide the least amount of freedom to MMPR for routing optimization.

The K-Best algorithm greatly saves computation time since only two LP problems are involved. The first LP decides the γ_{ij}^{y} for the All-On stage. Ranked in ascending order using one of the above metrics, the top $M-K$ monitors are disabled. Then, with these $M-K$ monitors turned off, a second LP is solved to maximize β using MeasuRouting [12]. However, since K-Best ranks the importance of each monitor based on metrics evaluated from the initial All-On stage, the measurement gain is predicted to diverge from the optimal.

C. Successive Selection Algorithm

Algorithm 1 Successive Selection Algorithm

1: **while** More than K monitors are left **do**
2:     Maximize β by using all remaining monitors
3:     find the corresponding γ_{ij}^{y}
4:     **for** Each remaining monitor $(i,j) \in \hat{M}$ **do**
5:        Calculate its performance metric for one of the five principles with γ_{ij}^{y}.
6:     **end for**
7:     Disable D monitors with least performance-metric
8: **end while**

The Successive Selection algorithm also starts from the initial All-On configuration with all M monitors and iteratively chooses D monitors to disable. Here, we use the same five metrics introduced in Section IV-B. The selection of which D monitors to disable is based on the ranking of remaining monitors $\hat{M}$ using one of the five metrics. In particular, it disables D monitors based on their ranking calculated from the previous iteration (Line 7). This means we use the information from the previous iteration (i.e., planned routes γ_{ij}^{y}, etc.) to calculate the metric for each monitor in the current iteration (Line 5).

Note that if the metric used is either the "least-rate" or the "least-neighbor" metric, both Successive Selection and K-Best will have the same selection of monitors and measurement gain since the metrics do not involve γ_{ij}^{y}.

D. Greedy Algorithm

Similar to Successive Selection, the Greedy algorithm also disables D monitors in each iteration, until K monitors are left. However, it is more complicated since it tests all remaining monitors $\hat{M}$ in each iteration. In order to test a monitor, it re-computes the maximized β after turning it off (Line 2-7), which essentially involves using MeasuRouting [12] (Line 4). Based on the testing of every remaining monitor, it disables D of them that have least impact on β (Line 8).

Algorithm 2 Greedy Algorithm

1: **while** More than K monitors are left **do**
2:     **for** Each remaining monitor $(i,j) \in \hat{M}$ **do**
3:        Disable the monitor
4:        Maximize β based on remaining monitors
5:        Store β
6:        Enable the monitor
7:     **end for**
8:     Find D monitors with largest $\beta \in \hat{M}$ when they are disabled
9:     $\hat{M} \leftarrow \hat{M}/\{(i,j) \in D\}$
10: **end while**

Since the Greedy algorithm exhaustively tests individual monitors at each iteration, its performance is hypothesized to be close to the optimal solution. It is still suboptimal since it tests individual monitors instead of every possible combination. However, the algorithm remains computationally costly, since it tests $O(\hat{M})$ monitors with $O(\hat{M})$ LP problems in each iteration. For a moderate sized topology, an MILP solver can sometimes work faster than this greedy approach. To reduce the computation time, we propose a less heavy-weighted algorithm called "Quasi-Greedy", which is a derivation of the Greedy algorithm. In Quasi-Greedy, instead of testing every remaining monitor, it only tests λ fraction candidates, where $0 < \lambda < 1$. We use C to denote candidate sets.

Algorithm 3 Quasi-Greedy Algorithm (λ)

1: **while** More than K monitors are left **do**
2:     Maximize β by using all remaining monitors
3:     Calculate measurement utility of each monitor $(i,j) \in \hat{M}$
4:     Choose $C=\lambda$ fraction remaining monitor $(i,j) \in \hat{M}$ as candidates
5:     **for** Each candidate monitor $\in C$ **do**
6:        Disable the monitor
7:        Maximize β based on remaining monitors
8:        Store β
9:        Enable the monitor
10:     **end for**
11:     Find D monitors $\in C$ with largest β when they are disabled
12:     $\hat{M} \leftarrow \hat{M}/\{(i,j) \in D\}$
13: **end while**

The candidates C are chosen based on the least-utility metric (Line 4), where utility is defined as $\sum_{y} p_{y} \gamma_{ij}^{y} \cdot I_{y}$. It benchmarks how much utility a monitor measures (Line 3). In each iteration, the Quasi-Greedy algorithm re-computes all the corresponding β by turning off one-by-one the remaining monitors in C to find the least important D monitor to disable (Line 5-11). It then disables these chosen D monitors from the remaining monitor set, $\hat{M}$ (Line 12). Besides least-utility, candidates can also be identified by using other heuristic metrics defined for the K-Best algorithm (Line 3).

E. Algorithm Examples

Suppose we have $M = 32$, $K = 24$ and $D = 4$. The K-Best algorithm directly disables $8 = 32 - 24$ monitors in a single step. On the other hand, the Successive Selection algorithm involves two iterations. In each iteration, $D = 4$ monitors are selected for exclusion based on their performance in the previous iteration, in accordance to one of the metrics defined in Section IV-B. The Greedy (Quasi-Greedy) algorithm also disables $D = 4$ monitors in each iteration. However, it selects the least important 4 monitors based on testing every (candidate) monitor one-by-one and solving the corresponding LP problem in each iteration, which is very time consuming. For the Greedy algorithm, it involves solving 32 and 28 LP problems in the first and second iteration, respectively. For the Quasi-Greedy algorithm, it involves solving $32 \times \lambda$ and $28 \times \lambda$ LP problems in the first and second iteration, respectively.

V. EVALUATION

In this section, we evaluate the performance of MMPR. The experiment settings are described in Section V-A. Section V-B presents the traces and the metrics of performance/cost to benchmark MMPR solutions. Section V-C discusses our evaluation results in detail.

A. Experiment Settings

We first define the settings for individual flows. We denote the set of flows as F, the traffic demand of flow f as b_f, and the importance of sampling it as i_f. We use $\upsilon_y \in \theta$ to represent the set of flows that belong to the flowset y. For our evaluation, we specify the measurement utility function of each flowset to be the following: $\Gamma_{y \in \theta} = \sum_{f, f i \upsilon_y} i_f b_f$. The importance of a flow f, can be viewed as points we earn if a byte of it is sampled. The optimization objective of MMPR is to maximize β. It is easy to see that β can be expressed in another way: $\sum_{f \in F} i_f b_f T_{\upsilon^{-1}}(f)$, which is exactly the total number of points earned by MMPR. Here $\upsilon^{-1}(f)$ denotes the flowset to which flow f belongs.

Most IP networks use link-state protocols such as OSPF [17] and IS-IS [18] 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. In this paper, we use the popular local search meta-heuristic in [19] to optimize link weights with respect to our aggregate traffic demands. The optimized link weights are then used to derive our original routing $\{\Gamma^{x \in \theta}\}_{(i,j) \in E}$. To avoid randomness in [19], we conduct experiments for the same setting five times, and only show the average results.

We have the greatest degree of freedom if each flow is assigned to a unique flowset. However, this is not scalable from both a computation and an implementation perspective. Therefore, we have q flowsets per OD pair. We also have $L \geq q$ flows for each OD pair. Each of the L flows in F belonging to a particular OD pair is assigned to one of the flowsets. There can be multiple ways of making such an assignment. In this paper, we randomly assign an equal number of flows to each of the q flowsets.

Table II lists the values for the MMPR parameters used for all the experiments in Section V-C. We generate sampling rates

TABLE II
DEFAULT EXPERIMENTAL PARAMETERS

Parameter	Description	Value/Distribution
q	Flowsets per OD pair	10
s	TE violation threshold	0.1
i_f	Flow importance	Pareto ($\lambda = 2$)

for each link using uniform distribution between 0 and 0.1. For one realization of link sampling rate and traffic demand, we repeat the experiments 10 times with different flow importance i_f generated from the Pareto distribution. We present the average measurement gain, unless specified otherwise. We use CPLEX [16] to find optimal solutions for the LP and MILP problems. For all the heuristic algorithms, we choose $D = 4$ and $D = 8$ in Abilene and AS6461/GEANT network, respectively. The algorithms with a larger D disable more monitors in each iteration. However, our evaluation results suggest that the performance is actually insensitive to the value of D, and the results are omitted here.

B. Traces and Performance Metrics

We use these three topologies in our experiments:

- *Abilene*: It is a public academic network in the U.S. with 11 nodes interconnected by 28 OC192 (10 Gbps) links. The traces used were from April 22-26, 2004 [1].
- *AS6461*: It is a RocketFuel [2] topology with 19 nodes and 68 links. To generate artificial traces, we first generate aggregate traffic demands for each OD pair using a Gravity Model [20]. The traffic demand of flow f, b_f, is then set equal to the traffic demand of its corresponding OD pair divided by L, where $L = 3000$.
- *GEANT*: It connects a variety of European research and education networks. Our experiments are based on the December 2004 snapshot [3], which consists of 23 nodes and 74 links ranging from 155 Mbps to 10 Gbps.

In our experiments, besides the measurement gain β and the TE metric in terms of MLU (maximum link utilization), we are also interested in the following four performance metrics:

- *Computation Time*: In our experiment, we only collect computation time for the LP or MILP solver. These parts usually take longer time than normal numerical computation, and are therefore the dominant part for our solutions. Meanwhile, the computation time for LP or MILP may vary for different solvers. We therefore do not mix them with other numerical computation.
- *F&T TE metric*: We use MLU as the TE metric in Equation (19). Besides MLU, we are also interested in the F&T metric [19], which is defined as weighted summation of link utilization of all the links. F&T characterizes the performance of entire network.
- *APLI(Average Path Length Inflation)*: It is defined as the ratio of $\sum_y \sum_{(i,j) \in E} Y^y_{(i,j)} \varphi_y$ and $\sum_y \sum_{(i,j) \in E} \Gamma^x_{(i,j)} \varphi_y$.

APLI reflects how flows get detoured. We expect important flows to have large path inflation since they are re-routed towards monitors.
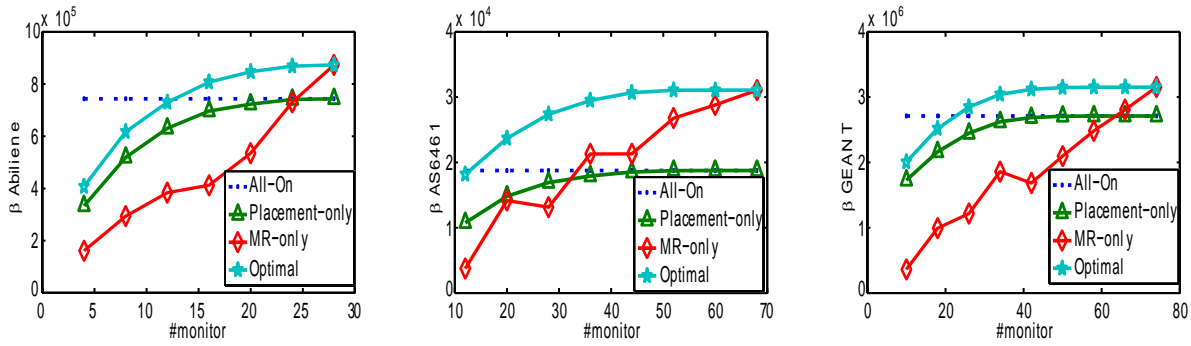
Fig. 2. Compare optimal MMPR with default cases for Abilene, AS6461 and GEANT.

. *Monitor Selection Overlap η*: It is defined as a ratio. The numerator is the number of monitors that are both selected by the heuristic and optimal solution. The denominator is the total number of selected monitors. A key part of the MMPR problem is to select the best monitor locations. This η metric reflects how heuristics select monitors.

C. Evaluation Results

In this section, we first compare Optimal MMPR with two baseline cases in Section V-C1 and show that MMPR can have better measurement gain up to 1.17X and 2.6X when compared to Placement-only and MR-only, respectively, for Abilene, 1.71X and 6X for AS6461, and 1.14X and 6.6X for GEANT. Section V-C2 presents detailed performance comparison of our proposed heuristic algorithms belonging to three different categories: K-Best, Successive Selection and Quasi-Greedy. Section V-C3 shows that all our proposed heuristic algorithms in each category perform very close to the Optimal MMPR solution and can reduce the computation by a factor of 23X in Abilene, 246X in AS6461, and 233X in GEANT.

In all the figures below, we use "KB", "SS", and "QG" to denote K-Best, Successive Selection and Quasi-Greedy, respectively. For example, "KB/utility" means K-Best algorithm with the least-utility ranking metric. Optimal MMPR is denoted as "Optimal" for short. For all the figures on computation time, the unit is second.

1) Optimal Solution vs. Default Cases: We first compare the optimal solution of MMPR with MR-only and Placement-only, using the same experimental settings (Section V-A). Placement-only was defined and formulated in Section II. MR-only, on the other hand, first randomly selects K monitors, and then finds the optimal routing γ^y_{ij} using MeasuRouting [12]. As shown in Figure 2. We see that optimal MMPR can have better measurement gain up to 1.17X($\frac{87}{74}$) and 2.6X($\frac{4}{1.5}$) compared to Placement-only and MR-only, respectively, for Abilene, 1.71X($\frac{3}{1.75}$) and 6X($\frac{1.8}{0.3}$) for AS6461, and 1.14X($\frac{3.1}{2.7}$) and 6.6X($\frac{2}{0.3}$) for GEANT.

We also present the performance of another baseline case, "All-On", in which every monitor is on and flowsets are routed by the default routing $\Gamma^x_{(i,j)}$. As shown in Figure 2, the optimal β of MMPR is better than the "All-On" case, even with only a small fraction of monitors turned on. Without strategic routing, even deploying monitors everywhere does not guarantee a comparable performance gain compared with

MMPR with a small number of monitors. As shown in these figures, MMPR can achieve the same measurement gain as the "All-On" case, but it can save 16(=28-12), 56(=68-12), and 54(=74-20) monitors in the case of Abilene, AS6461, and GEANT, respectively. Meanwhile, the computation time for optimal solution is fairly long (around 4 minutes) for AS6461, and increases to around 6 minutes for the GEANT network.

2) Sensitivity Analysis of Heuristic Algorithms: Due to the potentially long computation times required to solve for the optimal MMPR, we propose several heuristic algorithms to reduce the computation time complexity. They are categorized as "K-Best", "Successive Selection", "Greedy" and "Quasi-Greedy". We omit performance results for Greedy since it is computationally too costly.

We first compare K-Best algorithms (using different metrics) with the optimal solution in Figure 3. As expected, using the least-utility metric achieves the best β (very close to optimal) in all three topologies. It achieves 2.36X($\frac{2.6}{1.1}$) higher measurement gain compared to using the least-importance metric, but only increases 1.1X($\frac{1.22}{1.1}$) in computation time in AS6461.

As mentioned earlier, the computation time is only collected for the LP or MILP solver. Results show that using different ranking metrics lead to very similar computation times. From the perspective of an LP solver, an unsuitable monitor placement means either more steps are needed to achieve the optimal β (which is more time consuming), or there is no way to achieve very large β (which means shorter solving time). If we also consider other numerical computations (i.e: computation of each metric, ranking monitors based on metric values), "least-utility", "least-traffic", and "least-importance" definitely take longer, since the calculation of these three metrics involve all flows and monitors. In contrast, "least-rate" and "least-neighbor" only need topology information.

Figure 4 compares the Successive Selection algorithms with different ranking metrics, and the same trend is observed in Abilene, AS6461, and GEANT networks. We omit "least-rate" and "least-neighbor" cases since they have

similar measurement gain as in the corresponding K-Best case. Successive Selection with the least-utility metric also achieves the best performance. Similar to Fig.3, the three metrics share very close computation time (figures are omitted). It mostly depends on the number of iterations, which is linear with respect to the number of monitors in the Successive Selection algorithm.
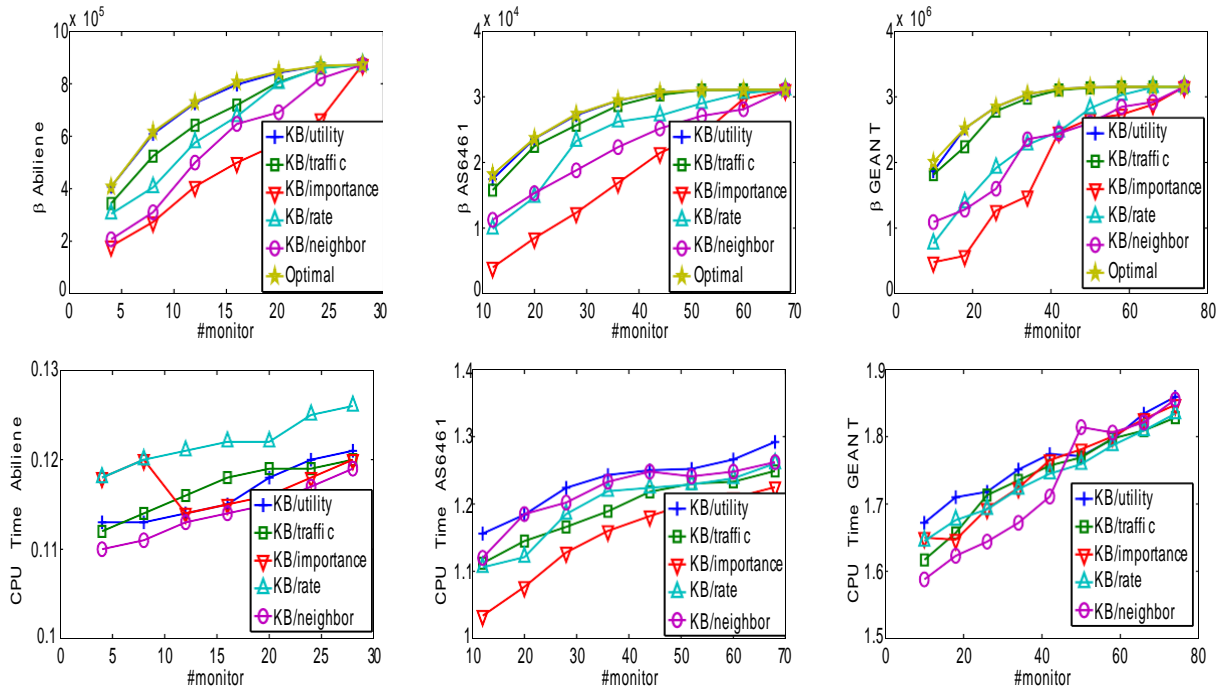
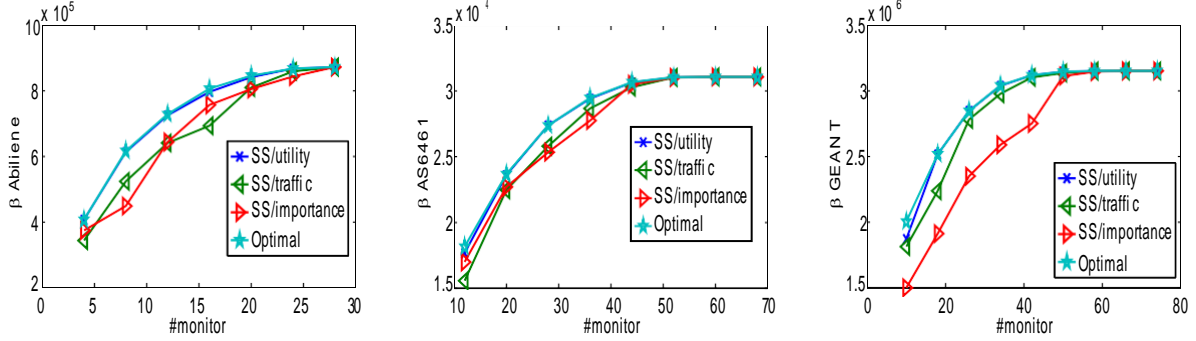Fig. 3. MMPR performance for K-best algorithms for Abilene, AS6461 and GEANT.

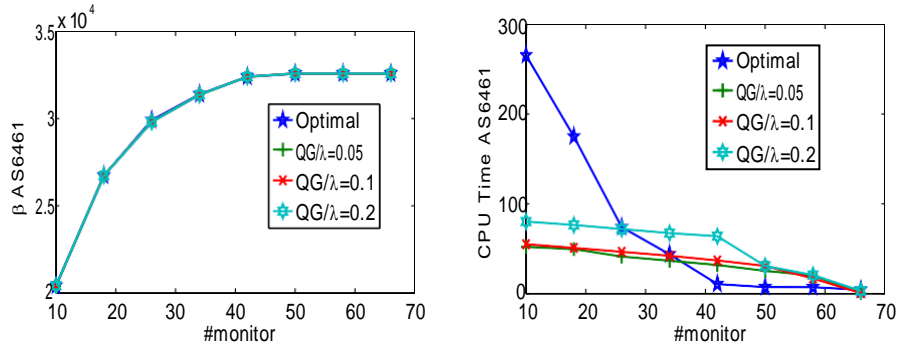Fig. 4. MMPR performance for successive selection algorithms for Abilene, AS6461 and GEANT.

Fig. 5. MMPR performance for quasi-greedy algorithm for AS6461.

Finally, we compare the Quasi-Greedy algorithm (with different λ values) against the optimal solution in Figure 5. Since Quasi-Greedy is still computationally intensive, we only present results for AS6461. Note that there are no obvious improvements on measurement gain for larger λ's. However, the computation time increases substantially with larger λ's. This implies that even with a smaller number of candidates, the Quasi-Greedy algorithm can perform very close to the optimal

and saves computation time.

3) Comparing K-Best, SS, and QG: In this section, we compare all three heuristic algorithms with the optimal MMPR solution. Results from the previous section show that "least-utility" is the most effective metric for ranking the importance of monitors. We therefore adopt "least-utility" metric as a basis for comparing the K-Best, Successive Selection, and Quasi-Greedy methods.
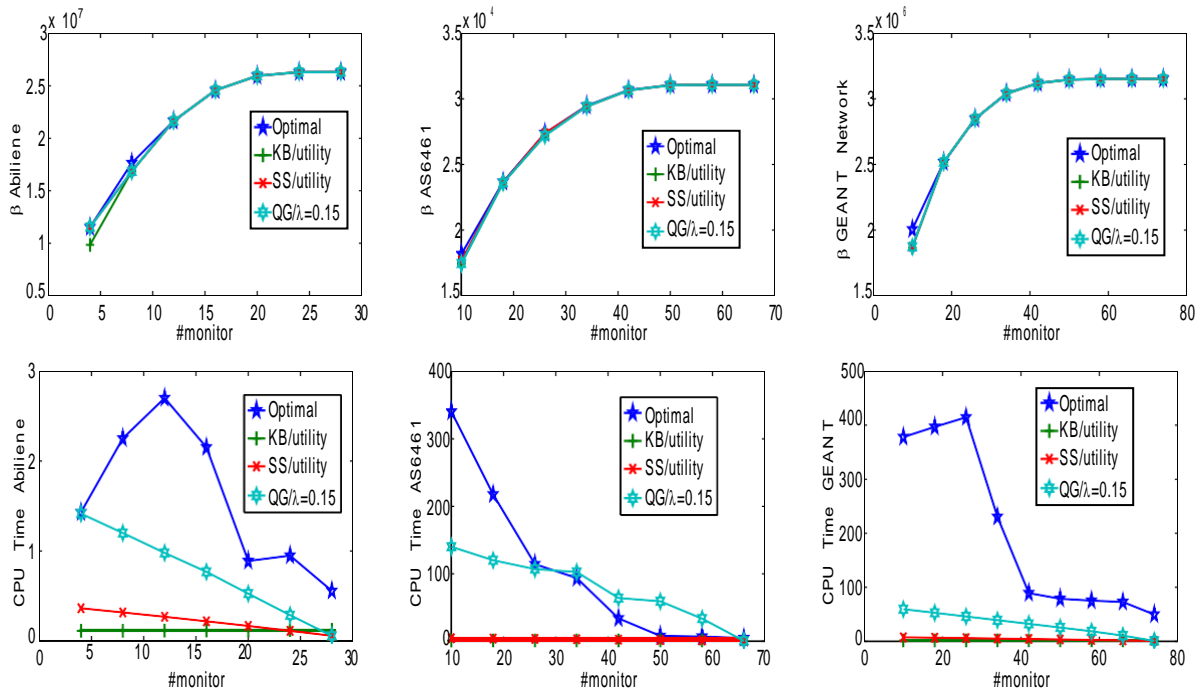
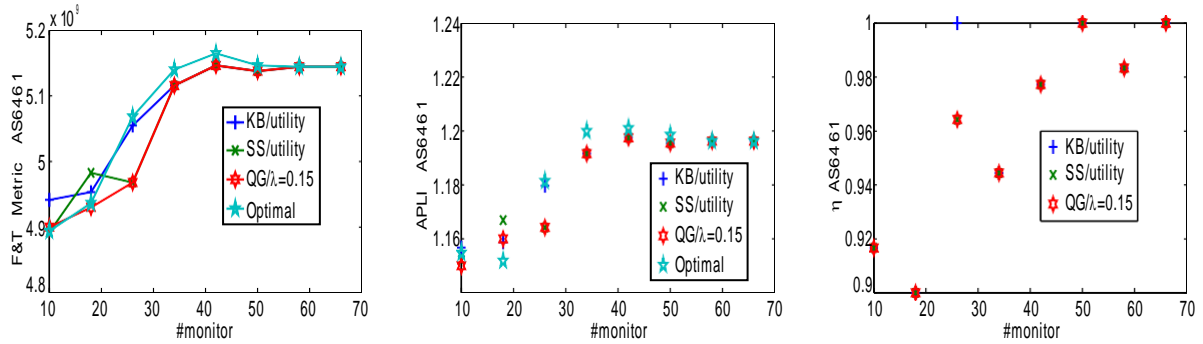Fig. 6. Compare heuristic algorithms for Abilene, AS6461 and GEANT.

Fig. 7. Compare other performance metrics for AS6461.

For the Quasi-Greedy algorithm, we present results using $\lambda = 0.15$. It tests $0.15M$ monitors in each iteration to choose D monitors to disable. Figure 6 shows the achieved measurement gain β and computation time for all the algorithms for all three topologies. In addition, we present F&T metric, APLI (Average Path Length Inflation), and η (Monitor Selection Overlap) in Figure 7. Only results for AS6461 are shown, but the same trends are observed for the other two topologies. We make the following observations based on our results:

. The maximum β's are very close for all algorithms. Both K-Best and Successive Selection algorithms are practical for large networks; their computation times are much less than the optimal case. Their best metric is "least-utility". Although K-Best is slightly worse than Successive Selection for Abilene, their achievable β's are almost the same for a large network like GEANT.

. Quasi-Greedy approach is very costly. However, its measurement gains are not noticeably better than the other heuristics. Therefore, there is no need to iteratively test

monitors one-by-one to decide which ones to disable. We can just simply disable monitors based on their performance metrics in the previous iteration.

. As shown in Figure 6, K-Best achieves almost the same measurement gain as MMPR optimal, but reduces computation times by a factor of 23X($\frac{2.7}{0.12}$), 246X($\frac{320}{1.3}$) and 233X($\frac{400}{1.71}$) for Abilene, AS6461, and GEANT, respectively, while Successive Selection reduces computation times by a factor of 10X($\frac{2.7}{0.26}$), 64X($\frac{320}{5}$), and 66X($\frac{400}{6}$) for Abilene, AS6461, and GEANT, respectively. Quasi-Greedy also saves computation times by a factor of 3X($\frac{300}{100}$) for AS6461. In practice, K-Best is the best choice since it greatly reduces computation time with measurement gains that are very close to the optimal.

. Values for F&T metric and APLI both increase with larger number of monitors. With more monitors, MMPR will put more weight on improving measurement gains, at a cost to the traffic engineering and packet forwarding performance. For example, because the same threshold $s = 0.1$ is used to bound TE violation in Equation (19), all algorithms finally

achieve the same MLU in every case (graphs are omitted here). However, both F&T metric and APLI increase with more monitors. For example, the 20% increase in APLI implies longer end-to-end forwarding delay, which may be acceptable for non-real-time traffic. To meet more stringent QoS requirements, they can be introduced as constraints in the MMPR formulation.

. The optimal solution does not necessarily achieve the best F&T or APLI results, since the optimal solution only optimizes for measurement gains with bounded violation of MLU. Some of the heuristics work better in preserving the overall network performance.

. η (Monitor Selection Overlap) shown in Figure 7 for AS6461 provides insights into why the performance of different algorithms are so close to the optimal. All the heuristic algorithms select almost the same set of monitor locations (e.g., 92%-100%) as the optimal solution, with the ratio approaching one as the number of available monitors increases. The same trend is observed for Abilene and GEANT topologies (results not shown here).

VI. RELATED WORK

Previous work mostly studied traffic measurement on a single monitor. They either infer traffic characteristics from sampled data [4, 5, 21–23] or use measurement schemes other than sampling for special traffic sub-populations [7, 8, 24–27].

Recently, researchers have begun investigating network-wide traffic measurement problems. Existing approaches [9–11] generally define and solve some monitor placement problem for fixed traffic characteristics and monitoring objectives. [9] defines utility functions for the sampled traffic. The problem is to maximize the overall utility with bounded measurement operation/deployment cost. It models variations of this problem, proves their complexities, and proposes heuristic algorithms. [11] improves upon [9] by performing a more rigorous analysis to indicate the convergence of any heuristic solution. Most recently, [28] proposes Successive c-Optimal design to optimize the deployment and sampling rate of large IP networks. However, their measurement goal is traffic matrix estimation. In contrast, MMPR is not restricted to any special measurement goal. None of them are suitable for changing traffic conditions or monitoring objectives.

Our work builds upon the recently proposed MeasuRouting paradigm [12], which proposes to assist traffic monitoring by intelligently routing traffic sub-populations over the corresponding monitors. It assumes fixed and random monitor placement, and routes flowsets based on their different measurement importance. It maximizes the overall measurement gain β under the constraint that σ is preserved at decent levels. With the freedom of intelligent routing, flows can better utilize the existing monitor infrastructure. Our work extends this framework by carefully choosing monitor locations. Our formulation also builds upon CSamp [15] like methods, to ensure non-overlapping measurement across monitors. [15] sets distinct hash filters on each monitor such that they capture different traffic sub-populations.

VII. DISCUSSION AND CONCLUSIONS

In this paper, we presented MMPR, a theoretical framework that jointly optimizes monitor placement and dynamic routing strategy to achieve maximum measurement utility, with limited monitoring resources. We formulated optimal MMPR as an MILP problem and proposed four heuristic algorithms to reduce the computation complexity: "K-Best", "Successive Selection", "Greedy" and "Quasi-Greedy". We performed detailed comparative study of these algorithms on three topologies, using both real traces and synthetic data. Our results suggest that the simplest algorithm, "K-Best", is actually the best choice in practice. It achieves measurement gains that are quite close to the optimal solutions, but it reduces the computation time by a factor of 246X in the best case in our experiments.

The theoretical study of MMPR framework can be extended by introducing other constraints or variations. For instance, as discussed earlier, measurement deployment/operation cost can be formulated in more concrete forms. Meanwhile, how to decide the proper flow utility function and measurement objective function remain open problems across different measurement applications. Furthermore, it would be interesting to treat sampling rate as another degree of freedom [9, 11], to let monitors dynamically adjust their monitoring capability. All these issues will be explored in future work.

MMPR, as well as MeasuRouting, require the prior knowledge of traffic importance in order to route flowsets differently. Such information need not be accurate in practice. Real measurements usually conduct hypothesis test procedures [8, 29]. They first obtain some global knowledge of the traffic, and zoom into the suspected traffic sub-population for more detailed analysis. Consider measuring flow size distribution for small/medium flows as our target, MMPR/MeasuRouting can depend on external modules to first estimate large flow identities. MMPR/MeasuRouting then directs the large flows away from measurement boxes, which are devised with many small-sized counters such that small/medium flows can be better maintained. In this example, there is no need to accurately measure flow sizes in the first step.

There are also many implementation issues of MMPR that need to be addressed. One important issue is to determine which exact routing protocols are used. A routing protocol that strictly routes traffic between an OD pair along only the shortest paths may provide less opportunities to 're-route' important flowsets through monitors. A centralized routing controller, e.g. [14], is able to detour flows away from the shortest path. Meanwhile, MMPR requires that the traffic be dynamically routed/rerouted. Such dynamic forwarding mechanism can be implemented using programmable routers [14, 30, 31]. Besides this, two other dynamics issues are also important: how to estimate flow importance dynamically and how configure routing table entries dynamically. Our recent work in [32] summarizes these challenges for MeasuRouting and proposes corresponding solutions for one measurement application: global iceberg detection and capture. The solutions are also applicable to MMPR, which builds upon MeasuRouting. MMPR extends MeasuRouting by introducing the opportunity to turn on and off monitors. In reality,

operators should avoid frequently switching monitor status. We plan to implement MMPR in OpenFlow [14] or other programmable routing platforms in future work.

REFERENCES

[1] "The abilene network," http://www.internet2.edu.
[2] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP network topologies with Rocketfuel," in *Proc. 2002 ACM SIGCOMM*.
[3] "Geant topology," http://archive.geant.net/upload/pdf/Topology¥Oct ¥_2004.pdf
[4] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," in *Proc. 1993 ACM SIGCOMM*.
[5] N. Hohn and D. Veitch, "Inverting sampled traffic," in *Proc. 2003 ACM SIGCOMM*.
[6] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice," *ACM Trans. Computer Syst.*, vol. 21, no. 3, pp. 270–313, Aug. 2003.
[7] A. Ramachandran, S. Seetharaman, N. Feamster, and V. Vazirani, "Fast monitoring of traffic subpopulations," in *Proc. 2008 ACM Internet Measurement Conference*.
[8] L. Yuan, C.-N.. Chuah, and P. Mohapatra, "ProgME: towards programmable network measurement," in *Proc. 2007 ACM SIGCOMM*.
[9] C. Claude, F. Eric, L. I. Guérin, R. Hervé, and V. Marie-Emilie, "Optimal positioning of active and passive monitoring devices," in *Proc. 2005 ACM CoNEXT*.
[10] K. Suh, Y. Guo, J. F. Kurose, and D. F. Towsley, "Locating network monitors: complexity, heuristics and coverage," in *Proc. 2005 IEEE INFOCOM*.
[11] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran, "Reformulating the monitor placement problem: optimal network-wide sampling," in *Proc. 2006 ACM CoNEXT*.
[12] S. Raza, G. Huang, C.-N. Chuah, S. Seetharaman, and J. P. Singh, "Measurouting: a framework for routing assisted traffic monitoring," in *2010 IEEE Infocom*.
[13] "Cisco netflow," http://www.cisco.com/.
[14] "The OpenFlow Switch Consortium," http://www.openflowswitch.org.
[15] 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. 2008 USENIX NSDI*.
[16] "IBM cplex software," http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.
[17] "OSPF," http://tools.ietf.org/html/rfc2328, Apr. 1998.
[18] "IS-IS," http://tools.ietf.org/html/rfc1142, Feb. 1990.
[19] B. Fortz and M. Thorup, "Internet TE by optimizing OSPF weights," in *Proc. 2000 IEEE INFOCOM*.
[20] A. Medina, N. Taft, S. Bhattacharyya, C. Diot, and K. Salamatian, "Traffic matrix estimation: existing techniques compared and new directions," in *Proc. 2002 ACM SIGCOMM*.
[21] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, "Is sampled data sufficient for anomaly detection?" in *Proc. 2006 ACM IMC*.
[22] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a better NetFlow," in *Proc. 2004 ACM SIGCOMM*.
[23] B.-Y. Choi and S. Bhattacharyya, "On the accuracy and overhead of Cisco sampled NetFlow," in *Proc. 2005 ACM SIGMETRICS Workshop on LSNI*.
[24] H. V. Madhyastha and B. Krishnamurthy, "A generic language for application-specific flow sampling," *ACM Computer Commun. Rev.*, vol. 38, no. 2, Apr. 2008.
[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 *Proc. 2006 ACM Internet Measurement Conference*.
[26] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice," *ACM Trans. Computer Systems*, vol. 21, no. 3, pp. 270–313, 2003.
[27] P. Loiseau, P. Gonçalves, S. Girard, F. Forbes, and P. Vicat-Blanc Primet, "Maximum likelihood estimation of the flow size distribution tail index from sampled packet data," in *Proc. 2009 ACM SIGMETRICS*.
[28] G. Sagnol, M. Bouhtou, and S. Gaubert, "Successive c-optimal designs: a scalable technique to optimize the measurements on large networks," in *Proc. 2010 ACM SIGMETRICS*.
[29] A. Sridharan and T. Ye, "Tracking port scanners on the IP backbone," in *Proc. 2007 ACM SIGCOMM*.
[30] R. Morris, E. Kohler, J. Jannotti, and F. Kaashoek, "The Click modular router," in *Proc. 1999 ACM SOSP*, pp. 217–231.
[31] C. Wiseman *et al.*, "A remotely accessible network processor-based router for network experimentation," in *Proc. 2008 ACM/IEEE ANCS*.
[32] G. Huang, S. Raza, S. Seetharaman, and C.-N. Chuah, "Dynamic measurement-aware routing in practice," *IEEE Network Mag.*, vol 25, no. 3, pp. 29–34, May 2011.

Guanyao Huang received the undergraduate and post-graduate degrees in Electrical Engineering from the University of Science and Technology of China, Hefei, China, 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, anomaly detection, and online social networks.
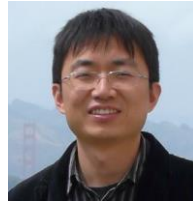
Chia-Wei Chang received the B.S. and M.S. degrees in communication engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 2002 and 2004, respectively. He currently pursues his Ph.D. degree in electrical, computer engineering of UCSD from 2005. Mr. Chang received the 2004 Prize of the Graduate Student thesis Contest from CIEE, Taiwan and CAL-IT^2 Fellowship Award 2005 from UCSD. His research interests generally lie in information and coding theory, multimedia application, network management and measurement, security and anomaly detection.

Chen-Nee Chuah is a Professor in Electrical and Computer Engineering at the University of California, Davis. She received her B.S. from Rutgers University, and her M. S. and Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley. Her research interests lie in the area of communications and computer networks, with emphasis on Internet measurements, network management, cyber-security, online social networks, and vehicular ad hoc networks.

Bill Lin holds a BS, a MS, and a Ph.D. degree in Electrical Engineering and Computer Sciences from the University of California, Berkeley. He is currently on the faculty of Electrical and Computer Engineering at the University of California, San Diego, where he is actively involved with the Center for Wireless Communications (CWC), the Center for Networked Systems (CNS), and the California Institute for Telecommunications and Information Technology (CAL-IT^2) in industry-sponsored research efforts. His research has led to over 150 journal and conference publications. He also holds 4 awarded patents.