

 Open access • Journal Article • DOI:10.1109/TPDS.2013.188

LEISURE: Load-Balanced Network-Wide Traffic Measurement and Monitor Placement — [Source link](#)

[Chia-Wei Chang](#), [Guanyao Huang](#), [Bill Lin](#), [Chen-Nee Chuah](#)

Institutions: [University of California, San Diego](#), [University of California, Davis](#)

Published on: 01 Apr 2015 - [IEEE Transactions on Parallel and Distributed Systems \(IEEE\)](#)

Topics: [Network monitoring](#), [Traffic engineering](#), [Load balancing \(computing\)](#), [Network management](#) and [Internet traffic](#)

Related papers:

- [PrePass: Load balancing with data plane resource constraints using commodity SDN switches](#)
- [Joint VM placement and routing for data center traffic engineering](#)
- [Network Servers Inbound Traffic Load Balancing Based on Central Decision System Results of research and model testing](#)
- [DREAM: dynamic resource allocation for software-defined measurement](#)
- [Prediction of the impact of network switch utilization on application performance via active measurement](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/leisure-load-balanced-network-wide-traffic-measurement-and-3gdrfqz5u>

UC Davis

UC Davis Previously Published Works

Title

LEISURE: Load-balanced network-wide traffic measurement and monitor placement

Permalink

<https://escholarship.org/uc/item/07h083x5>

Journal

IEEE Transactions on Parallel and Distributed Systems, 26(4)

ISSN

1045-9219

Authors

Chang, CW
Huang, G
Lin, B
[et al.](#)

Publication Date

2015-04-01

DOI

10.1109/TPDS.2013.188

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-ShareAlike License, available at <https://creativecommons.org/licenses/by-sa/4.0/>

Peer reviewed

LEISURE: Load-Balanced Network-Wide Traffic Measurement and Monitor Placement

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

*University of California, San Diego

†University of California, Davis

Abstract—Network-wide traffic measurement is of interest to network operators to uncover global network behavior for the management tasks of traffic accounting, debugging or troubleshooting, security, and traffic engineering. Increasingly, sophisticated network measurement tasks such as anomaly detection and security forensic analysis are requiring in-depth fine-grained flow-level measurements. However, performing in-depth per-flow measurements (e.g., detailed payload analysis) is often an expensive process. Given the fast-changing Internet traffic landscape and large traffic volume, a single monitor is not capable of accomplishing the measurement tasks for all applications of interest due to its resource constraint. Moreover, uncovering global network behavior requires network-wide traffic measurements at multiple monitors across the network since traffic measured at any single monitor only provides a partial view and may not be sufficient or accurate. These factors call for coordinated measurements among multiple distributed monitors. In this paper, we present a centralized optimization framework, *LEISURE* (Load-Equalized meaSUREment), for load-balancing network measurement workloads across distributed monitors. Specifically, we consider various load-balancing problems under different objectives and study their extensions to support both fixed and flexible monitor deployment scenarios. We formulate the latter flexible monitor deployment case as an MILP (Mixed Integer Linear Programming) problem and propose several heuristic algorithms to approximate the optimal solution and reduce the computation complexity. We evaluate *LEISURE* via detailed simulations on Abilene and GEANT network traces to show that *LEISURE* can achieve much better load-balanced performance (e.g., 4.75X smaller peak workload and 70X smaller variance in workloads) across all coordinated monitors in comparison to a naive solution (uniform assignment) to accomplish network-wide traffic measurement tasks under the fixed monitor deployment scenario. We also show that under the flexible monitor deployment setting, our heuristic solutions can achieve almost the same load-balancing performance as the optimal solution while reducing the computation times by a factor up to 22.5X in Abilene and 800X in GEANT.

Index Terms—Load-balancing, network-wide traffic measurement

1 INTRODUCTION

ACCURATE traffic measurement is essential to a variety of network management tasks, including traffic engineering (TE), capacity planning, accounting, anomaly detection, and security forensics. For example, deep packet inspection (DPI) allows for post-mortem analysis of network events and helps operators to understand the payload properties of the transiting Internet traffic. Another solution, Network DVR [2], performs selective flow-based trace collection by matching packets against application-specific signatures. However, doing fine-grained flow level measurements is often an expensive process that requires dedicated hardware (e.g., TCAMs [3]), specialized algorithms, (e.g., Bloom Filters [4]), or vast storage capacity. Given the fast-changing Internet traffic landscape and large traffic volume, a single monitor is not capable of accomplishing the measurement tasks from all applications of interest due to its resource constraint. This calls for coordinated measurements among multiple distributed monitors. Also, network-wide traffic measurement at multiple monitors is key to uncovering global network behavior since traffic measured at any single monitor only provides a partial view and may not be sufficient or accurate. For example, a *global iceberg* [5] may have high aggregate volume across many monitors, but may not be detectable at any single monitor.

To perform effective network-wide traffic measurement across multiple distributed monitors, a centralized framework that coordinates measurement responsibilities across different monitors is needed. Sekar et al. [6] proposed CSAMP (Coor-

ordinated Sampling), a centralized hash-based packet selection system as a router-level primitive, to allow distributed monitors to measure disjoint sets of traffic without requiring explicit communications. CSAMP uses an optimization framework to specify the set of flows that each monitor is required to record by considering a hybrid measurement objective that maximizes the total flow-coverage, subject to ensuring that the optimal minimum fractional coverage of the task can be achieved. However, CSAMP does not consider the load-balancing of workloads for multiple measurement tasks across the available monitors, which can lead to situations in which some monitors carry substantially higher workloads than other monitors. Coupled with ever-increasing link rates and traffic volume, the more heavily loaded monitors can easily become overwhelmed. In addition, existing frameworks (e.g., CSAMP) do not consider flexible monitor deployment scenarios and are agnostic to differentiation in the importance of traffic sub-populations or the cost of individual measurement tasks.

In this paper, we present *LEISURE* (Load-Equalized meaSUREment), a new centralized optimization framework to address the network measurement load-balancing problem in various realistic scenarios. Our solution framework takes as inputs a) a routing matrix, b) the network topology and the monitor infrastructure deployment, and c) the measurement requirements of the different measurement tasks, and it decides which available monitors should participate in each specific measurement task and how much they need to measure to optimize the load-balancing objectives. Ideally the *load-balancing objective* is to have identical workload for all monitors where workload denotes the normalized traffic amount that each monitor measures. In this work, the *load-balancing objective*

is mainly defined in terms of the following: 1) minimizing the variance of workloads across all monitors or 2) minimizing the maximum workload among them. Optimal solutions are translated into disjoint sets of required-measured flows that each monitor is assigned to measure. In contrast to CSAMP [6] that aims at maximizing the total flow-coverage without considering load-balancing problems, LEISURE distributes traffic measurement tasks evenly across coordinated monitors such that the required fractional coverage of those tasks can be achieved. We summarize our contributions as follows:

- We present LEISURE and formulate the load-balancing optimization problem for network-wide traffic measurements. We also propose simple heuristic solutions and extend LEISURE to incorporate practical constraints – i.e., (a) with limited measuring resources at monitors, (b) with limited number of deployed monitors, (c) with multiple routing paths (e.g., equal-cost multi-path (ECMP)) for each origin-destination (OD)-pair traffic.
- We further formulate LEISURE as an MIQP (Mixed Integer Quadratic Programming) problem under flexible monitor placement scenarios, and show how it could be reformulated as a standard MILP (Mixed Integer Linear Programming) problem by decoupling two key decision variables. We propose several heuristic algorithms that can approach the performance of the optimal MILP solution, but yet require dramatically shorter computation times.
- We provide detailed evaluations based on the Abilene [7] and GEANT [8] network topologies and traces. Our results show that the significant load-balancing improvement (e.g., 4.75X smaller maximum workload and 70X smaller variance in workloads) is achieved by using LEISURE to optimally distribute the measurement tasks across all coordinated monitors when compared with the naive uniform assignments. We also show that our proposed heuristic algorithms can reduce the computation times by a factor up to 22.5X in Abilene and 800X in GEANT while achieving load-balancing performance that are close to the optimal.

The paper is structured as follows: Section 2 motivates our network load-balancing problem via illustrative examples. Section 3 presents our problem formulations and solutions, Section 4 considers extensions for limited monitor deployment scenarios. Section 5 describes our simulation setup and evaluation results, and Section 6 concludes the paper. Additional supplemental sections discuss related work, implementation issues, and further extensions for considering multi-path routing and multi-measurement task scenarios.

2 MOTIVATING EXAMPLE

Consider the toy example shown in Fig. 1 with traffic demands from three OD-pairs: SF→NY, LA→Seattle, and Chicago→Atlanta, each with 120 units of traffic (IP flows). Suppose the measurement task imposed by the network operator is to measure all traffic from these three OD-pairs. One naive approach is to simply always measure the traffic for each OD-pair at the ingress router, as shown in Fig. 1(b). In this case, monitors would only need to be deployed in SF, LA, and Chicago, each with a worst-case (peak) measurement workload of 120 units of traffic. Similarly, the traffic for each OD-pair can be measured at the egress router, as shown in Fig. 1(c). In this case, monitors would instead need to be deployed in NY,

Seattle, and Atlanta, each with the same peak measurement workload of 120 units. Both of these approaches only need 3 monitors to accomplish the assigned measurement tasks in this example, but each monitors needs to be able to handle 120 units of traffic.

On the other hand, assume all routers are equipped with monitors that are capable of performing the measurement tasks. In this setting, rather than concentrating the measurement workloads on a small set of monitors, we can instead load-balance the measurement workloads across the monitors so that the peak measurement workload on these monitors is minimized, and thereby minimizing the processing requirements of these monitors. This can be achieved by assigning a *fraction* of the required measurement traffic to each monitor for which the monitor is responsible. One simple strategy is to simply *uniformly* distribute the required measurement traffic of each OD-pair to the monitors along its routing path, as depicted in Fig. 1(d). For example, the 120 units of traffic for SF→NY is measured uniformly across the monitors placed in SF, Denver, Kansas City, Indianapolis and NY. Each takes the measurement responsibility for 24 units of traffic. Similarly, the monitors in LA, Denver, Seattle and Chicago, Indianapolis, Atlanta each takes the measurement responsibility for 40 units of traffic for the LA→Seattle and Chicago→Atlanta traffic, respectively. The monitor with the highest measurement workload is therefore most likely be the router with the largest number of OD-pairs passing through it (e.g., 64 units of measurement traffic in Denver/Indianapolis).

Alternatively, another load-balancing method is to distribute the required measurement traffic of each OD-pair to the monitors along the path in inverse proportion to the traffic passing through them, as shown in Fig. 1(e). For example, the traffic passing through SF, Denver, Kansas City, Indianapolis and NY is 120, 240, 120, 240 and 120, respectively. Based on an inverse proportion calculation, SF, Kansas City and NY should measure 30 units of the SF→NY traffic ($\frac{120^{-1}}{120^{-1}+240^{-1}+120^{-1}+240^{-1}+120^{-1}} \times 120$), while Denver and Indianapolis should measure 15 units for SF→NY. Similarly, the monitors in LA and Seattle should measure 48 units of the LA→Seattle traffic, while Denver should measure 24 units for LA→Seattle, and the monitors in Chicago and Atlanta should measure 48 units of the Chicago→Atlanta traffic, while Indianapolis should measure 24 units for Chicago→Atlanta.

Although both methods achieve significant reduction in the maximum measurement workload, as compared to the naive approaches (e.g., 120→64, 120→48), the maximum measurement workload can actually be further reduced to 40 units by using our proposed LEISURE approach to solve the global load-balancing optimization problem, as shown in Fig. 1(f). In this optimal solution, the SF→NY traffic is measured uniformly by only *three* monitors (SF, Kansas City, and NY) instead of five, each with 40 units of traffic while Denver and Indianapolis are *not* involved in the measurement of the SF→NY traffic. This in turn allows the equal splitting of the LA→Seattle traffic and the Chicago→Atlanta traffic across all three routers in each of its respective path, which results in all monitors having the same perfectly load-balanced measurement workload of 40 units.

The above shows that by deciding which monitors should participate in which measurement task, and how much they should measure, LEISURE can load-balance the measurement

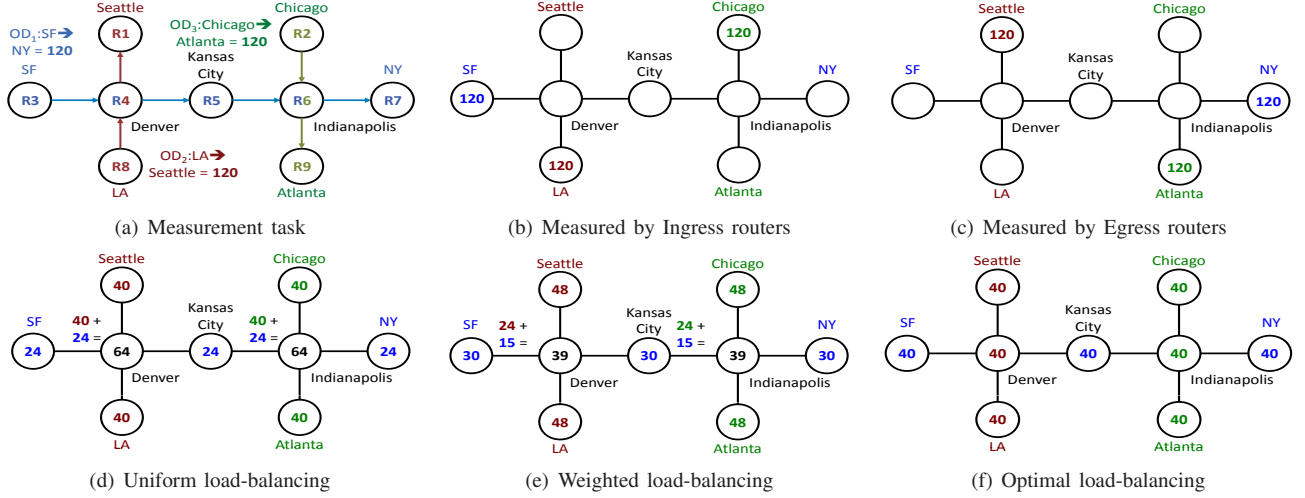


Fig. 1. Different load-balancing approaches for our toy example, which includes three OD-pair traffic as our measurement task (i.e., SF→NY, LA→Seattle, and Chicago→Atlanta, each with 120 units of traffic).

tasks optimally while the required fractional coverage of those tasks can be fulfilled. Next, we seek to find globally optimal load-balancing solutions by deploying LEISURE under different network conditions (e.g., topology, traffic demand, routing matrix, etc), measurement objectives (e.g., minimize maximum workload, maximize measurement utility, etc), and resource constraints (e.g., subset of routers are capable of monitoring, some monitors have lower capacities, etc).

3 LEISURE FRAMEWORK

We now present a load-balancing optimization framework to cover network-wide monitoring objectives while respecting router resource constraints. ISPs typically specify their network-wide measurement tasks in terms of OD-pairs. To cover these measurement assignments, LEISURE needs both the traffic demand and routing information, which are readily available to network operators, as described in [9]. In general, LEISURE is a centralized architecture that allocates disjoint sets of required-measurement flows in OD-pairs to each router given global network-wide information. The global information includes: a) network topology, monitoring infrastructure deployment, b) traffic demand, routing matrix, and c) measurement requirements and the associated cost for each measurement task.

The problem formulation builds up from the simplest case in which we assume the following: 1) all routers are deployed with monitors and capable of measurement; 2) each OD-pair follows a single router-level path by Open Shortest Path First (OSPF); and 3) there is only one measurement task for every monitor. These constraints are gradually relaxed in Section 4.

3.1 Basic Model

Let $G(V, E)$ represent our network topology, where V is the set of routers (monitors) and E is the set of directed links. Each router V_i ($i = 1 \dots M$) has two factors that limit its measurement ability: memory and bandwidth. We abstract them into a single resource constraint C_{v_i} ($i = 1 \dots M$), the number of flows that router V_i can measure in a given measurement interval.

An OD-pair, $OD_x, x \in |V| \times |V - 1|$, represents the set of flows between the same pair of ingress/egress routers for which an aggregated routing placement is given. The set of

all $|V| \times |V - 1|$ OD-pairs is given by $\Theta: \Theta = |V| \times |V - 1|$. Φ_x characterizes the traffic demand (IP flows) of the OD-pair $OD_x, x \in \Theta$ in a given measurement interval (e.g., 5 minutes). P_x represents the given routing strategy (router-level path) for every OD-pair $OD_x, x \in \Theta$.

a_x denotes the desired coverage fraction of IP flows of OD_x that is required to measure, which is imposed by the network operator. Therefore, the total required measurement traffic (number of flows), β , introduced to all routers is simply a summation of traffic demand per OD-pair times a_x as $\beta = \sum_{x \in \Theta} \Phi_x \times a_x$.

Let d_i^x denote the fraction of traffic demand (IP flows) of OD_x that router V_i samples/measures (i.e., $d_i^x = \frac{\text{measured flows in } \Phi_x}{\Phi_x}$) while L_i denotes the total traffic (number of IP flows) that router V_i measures for all OD-pairs, $OD_x, x \in \Theta$ normalized by β . The summation of L_i for all routers V_i ($i = 1 \dots M$) then equals 1. We have:

$$\beta = \sum_{x \in \Theta} \Phi_x \times a_x \quad (1)$$

$$L_i = \frac{1}{\beta} \sum_{x: V_i \in P_x} d_i^x \times \Phi_x \quad \forall i \quad (2)$$

$$\sum_{i=1}^M L_i = 1 \quad (3)$$

Our decision variable is d_i^x . The first constraint of d_i^x is that the value of d_i^x is bounded between 0 and 1 as Eq. (4). The second constraint is that the summation of d_i^x along the path P_i for each OD-pair $OD_x, x \in \Theta$ is a_x , as Eq. (5). If router V_i is not in the routing path P_x of OD-pair $OD_x, x \in \Theta$ ($V_i \notin P_x$), d_i^x is inherently 0. The third constraint is that the measured fraction of β for each monitor V_i should not exceed its measurement ability (resource constraint) C_{v_i} as Eq. (6). Notations are also summarized in Table I.

$$0 \leq d_i^x \leq 1 \quad \forall x, i \quad (4)$$

$$\sum_{i: V_i \in P_x} d_i^x = a_x \quad \forall x \in \Theta \quad (5)$$

$$\sum_{x: V_i \in P_x} d_i^x \times \Phi_x \leq C_{v_i} \quad \forall i \quad (6)$$

TABLE 1
Notations

Notation	Description
OD_x	represent a set of flows between the same pair of ingress/egress routers
Θ	the set of all $ V \times V - 1 $ OD-pairs: $OD_x, x \in \Theta$
Φ_x	characterizes the traffic demand (IP flows) of OD-pair $OD_x, x \in \Theta$
P_x	represents the given routing strategy for OD-pair $OD_x, x \in \Theta$
a_x	the fraction of Φ_x (IP flows) of OD_x that is required to measure
d_i^x	the fraction of Φ_x (IP flows) of OD_x that router V_i measures
β	the total required measurement traffic (number of IP flows)
L_i	the total traffic (number of IP flows) that V_i measured normalized by β
α	load-balancing objective

TABLE 2
 d_i^x for each approach with the toy example shown in Fig. 1

	d_3^1	d_4^1	d_5^1	d_6^1	d_7^1	d_7^2	d_4^2	d_8^2	d_5^3	d_6^3	d_9^3	$MAX(L_i)$	$VAR(L_i)$	# of monitors	Decision
LB(ingress)	1	0	0	0	0	0	0	1	1	0	0	120/360	0.025	3	local
LB(egress)	0	0	0	0	1	1	0	0	0	1	1	120/360	0.025	3	local
LB(uniform)	1/5	1/5	1/5	1/5	1/5	1/3	1/3	1/3	1/3	1/3	1/3	64/360	0.00167	9	local
LB(weighted)	1/4	1/8	1/4	1/8	1/4	2/5	1/5	2/5	2/5	1/5	2/5	48/360	0.000484	9	global
LB(optimal)	1/3	0	1/3	0	1/3	1/3	1/3	1/3	1/3	1/3	1/3	40/360	0	9	global

3.2 Problem Formulation

We define our load-balancing objective in abstract form α , which can be any term as long as it captures load-balancing performance (i.e., identical workload for all monitors). The overall optimization objective of LEISURE is to minimize α that each router operates within its resource constraint by given parameter a_x , the required fractional coverage per OD-pair imposed by the network operator. In this section, we formulate and study three different optimization problems that correspond to three different load-balancing objective α : *min-VAR*, *min-MAX* and *min-VAR-given-MAX*.

3.2.1 Minimize Variance Problem (*min-VAR*)

In this problem, we denote α as the variance of L_i across all participating routers¹. The intuition is that with more even workload L_i for all routers, the variance is smaller (e.g., variance=0 stands for ideal load-balancing objective where $L_i = \frac{1}{M}$ for all M routers). We have:

$$\alpha = VAR(L_i) = \frac{\sum_{i=1}^M (L_i - \bar{L})^2}{M} \quad (7)$$

$$\bar{L} = \frac{1}{M} \sum_{i=1}^M L_i = \frac{1}{M} \cdot 1 \quad (8)$$

This optimization problem is formulated as:

minimize $\alpha=VAR(L_i)$
subject to

$$\frac{1}{\beta} \sum_{x:V_i \in P_x} d_i^x \times \Phi_x = L_i \quad \forall i \quad (9)$$

$$\sum_{i:V_i \in P_x} d_i^x = a_x \quad \forall x \in \Theta \quad (10)$$

$$\sum_{x:V_i \in P_x} d_i^x \times \Phi_x \leq C_{v_i} \quad \forall i \quad (11)$$

$$0 \leq d_i^x \leq 1 \quad \forall x, i \quad (12)$$

1. We use ‘‘population variance’’ instead of ‘‘sample variance’’ as our objective function since we already know the number of monitors m .

3.2.2 Minimize Maximum Problem (*min-MAX*)

In this problem, we denote α as the maximum value of L_i across all routers:

$$\alpha = MAX(L_i) \quad i = 1 \dots M \quad (13)$$

The intuition is that when LEISURE keeps minimizing the maximum value of L_i for all monitors by adjusting decision variables d_i^x , other smaller L_i will increase, eventually they will reach some equilibrium state that no more adjustments it can do to lower the $MAX(L_i)$ without increasing other L_i above $MAX(L_i)$. The problem formulation shares the same constraints as *min-VAR* problem, Eq.(9-12), except that the objective function is different: minimize $\alpha = MAX(L_i), i = 1 \dots M$.

3.2.3 Minimize Variance with Max-Constraint Problem (*min-VAR-given-MAX*)

This problem involves two phases. In the first step, we formulate the *min-MAX* problem given in Section 3.2.2 to find the minimum achievable maximum value L_{max} (L_{max} = minimized $MAX(L_i), i = 1 \dots M$) for all routers to cover the total required-measurement IP flows, β . Then we seek for any opportunity to further re-distribute the measurement task (workload) evenly within this constraint. Therefore in the second step, we introduce additional constraints to the *min-VAR* problem given in Section 3.2.1 to limit the L_i for each router V_i to be at most L_{max} . We then minimize the variance of L_i across all routers. Specifically, we only need to introduce the following constraint to the *min-VAR* problem:

$$L_i = \frac{1}{\beta} \sum_{x:V_i \in P_x} d_i^x \times \Phi_x \leq L_{max} \quad \forall i \quad (14)$$

Therefore the *min-VAR-given-MAX* problem actually combines the *min-VAR* and *min-MAX* problems.

3.3 Optimal/Heuristic Solutions

We seek for the optimal d_i^x assignments for the above three problems. There is a variety of optimization tools that we can leverage. Specifically, the optimal solutions can be found by using a Quadratic Programming (QP) formulation for the *min-VAR* problem and a Linear Programming (LP) formulation for the *min-MAX* problem. The combined problem, *min-VAR-given-MAX*, can be solved in a two-phase manner by using LP

first and QP follows. We refer these three optimal solutions of LEISURE as LB(min-VAR), LB(min-MAX), and LB(min-VAR-given-MAX), respectively.

Besides the optimal solutions, we introduce one simple heuristic method called LB(weighted) under the assumption that routers can always fulfill assigned measured tasks (e.g., no resource constraints for all routers in Eq. (6)). LB(weighted) calculates d_i^x in inverse-proportion to the total required-measurement traffic amount (IP flows) passing through router V_i . The rationale behind it is that routers with larger required-measurement IP flows passing through should be assigned with fewer IP flows to measure in order to achieve load-balancing. Let β_i denote the total required measurement traffic passing through router V_i , which can be calculated using Eq. (15). The d_i^x assignment for LB(weighted) is formulated as:

$$\beta_i = \sum_{x:V_i \in P_x} \Phi_x \cdot a_x \quad \forall i \in V \quad (15)$$

$$d_i^x = \frac{\frac{1}{\beta_i}}{\sum_{i:V_i \in P_x} \frac{1}{\beta_i}} \times a_x \quad \forall x, i \quad (16)$$

Although LB(weighted) does not necessarily lead to the optimal solution, its computation time is very fast compared to the time required to solve QP or LP optimization problems for LB(min-VAR), LB(min-MAX), and LB(min-VAR-given-MAX). In Section 5, we also compare their performances with the following three simple naive strategies:

- LB(ingress): the required measurement traffic $\Phi_x \cdot a_x$ for each OD-pair $OD_x, x \in \Theta$ is only measured at ingress routers.
- LB(egress): the required measurement traffic $\Phi_x \cdot a_x$ for each OD-pair $OD_x, x \in \Theta$ is only measured only at egress routers.
- LB(uniform): the required measurement traffic $\Phi_x \cdot a_x$ for each OD-pair $OD_x, x \in \Theta$ is measured evenly across the routers on its routing path P_x .

Table 2 summarizes the corresponding d_i^x for each approach with the toy example presented in Fig. 1. In this example, LB(min-VAR), LB(min-MAX), and LB(min-VAR-given-MAX) all have the same optimal load-balancing performance (i.e., $MAX(L_i) = \frac{40}{360}$ and $VAR(L_i) = 0$), which we denote as LB(optimal). In comparison, LB(ingress) and LB(egress) have poorest load-balancing performance but with least number of deployed monitors. LB(uniform) outperforms them but needs more monitors (e.g., 9 instead of 3 monitors in our toy example). LB(weighted) and LB(optimal) which consider global required measurement traffic can have better load-balancing performance compared to the local approaches (e.g., LB(ingress), LB(egress) and LB(uniform)), where LB(optimal) has the optimal load-balancing performance but needs much more computation time.

4 MEASUREMENT WITH LIMITED MONITORS

In this section, we extend previous formulations to consider limited monitor deployment scenarios. In practice, not every router is equipped with monitor and capable of measurement. Suppose only K out of the M routers are deployed with monitors and thus have measurement capability. We assume each OD-pair $OD_x, x \in \Theta$ has at least one router on its

routing path P_x which is capable of measurement to fulfill the measurement tasks imposed by the network operator. Our formulation includes two problems: 1) measurement with fixed monitor deployment problem, and 2) measurement with flexible monitor deployment problem.

4.1 Fixed Monitor Deployment Problem

In the first case, we assume that these K monitors have been deployed in routers and are fixed. Our goal is to distribute required measurement tasks to these limited K routers. It can be simply solved by changing the routing index P_x as follows: we exclude router V_i from P_x if V_i is not equipped with monitor and unable to measure (e.g., $P_x^* = P_x - \{V_i\}$ for all OD-pair $OD_x, x \in \Theta$). Variance calculation should also be modified accordingly since we now only have K monitors instead of M . All constraints remain the same except that P_x are replaced by P_x^* in Eq. (9)-(12).

$$P_x^* = P_x - \{V_i\}, \text{ if } V_i \text{ is not deployed with monitor} \quad (17)$$

$$VAR(L_i) = \frac{\sum_{i=1}^K (L_i - \bar{L})^2}{K} \quad (18)$$

$$\bar{L} = \frac{1}{K} \sum_{i=1}^K L_i = \frac{1}{K} \quad (19)$$

4.2 Flexible Monitor Deployment Problem

In the second case, the location of K monitors have not been decided and they are flexible to be deployed in any router. This problem includes not only the *distribution* of measurement tasks, but also the *placement* of monitors. To formulate this problem, we introduce additional decision variables u_i , where $u_i = 1$ if router V_i is selected to deploy a monitor, and $u_i = 0$ otherwise. The summation of u_i is therefore to be K . We assume every monitor has identical limited measurement capability (resource constraint) as C_m . The problem is formulated below with load-balancing objective as $\alpha = MAX(L_i)$. Note that it is no longer an LP/QP problem since $u_i, i \in V$ are Boolean variables.

minimize α

subject to

$$\frac{1}{\beta} \sum_{x:V_i \in P_x} d_i^x \times \Phi_x \times u_i = L_i \quad \forall i \quad (20)$$

$$\sum_{i:V_i \in P_x} d_i^x \times u_i = a_x \quad \forall x \in \Theta \quad (21)$$

$$\sum_{x:V_i \in P_x} d_i^x \times \Phi_x \times u_i \leq C_m \quad \forall i \quad (22)$$

$$\sum_{i=1}^M u_i = K \quad (23)$$

$$0 \leq d_i^x \leq 1 \quad \forall x, i \quad (24)$$

$$u_i \in \{0, 1\} \quad \forall i \quad (25)$$

In this model, L_i is the summation of the product of Φ_x, d_i^x and u_i . Therefore the objective function α is related to the product of two decision variables u_i and d_i^x , and the optimization problem falls into the MIQP (Mix Integer Quadratic Programming) category. In order to avoid quadratic programming, we could introduce z_i^x to decouple $d_i^x \times u_i$ by using Equations (26) to (28). It is easy to see their equivalence.

When $u_i = 0$, $z_i^x = 0$ from (27); and when $u_i = 1$, $z_i^x = d_i^x$ from (28).

$$z_i^x = d_i^x \times u_i \quad (26)$$

$$0 \leq z_i^x \leq u_i \quad (27)$$

$$d_i^x + u_i - 1 \leq z_i^x \leq d_i^x \quad (28)$$

Although we could reduce the MIQP problem to the MILP (Mix Integer Linear Programming) problem by introducing z_i^x , the new MILP problem actually has doubled number of decision variables. This is because the cardinality of $u_i \ll$ the cardinality of d_i^x in practice. Fortunately, the decision variable d_i^x (for distributing measurement tasks) is highly dependent on the decision variable u_i (for monitor placement). If $u_i = 0$ (i.e., router V_i is not selected to deploy a monitor), router V_i cannot participate in any specific measurement task. It means d_i^x , the fraction of Φ_x (IP flows) for each OD-pair, $OD_x, x \in \Theta$ that router V_i measures should be zero (i.e., $d_i^x = 0$). On the other hand when $u_i = 1$ (i.e., router V_i is selected to deploy a monitor and capable of measurement), d_i^x could be any decimal bounded between 0 and 1. Therefore we can directly use d_i^x to substitute $d_i^x \times u_i$ in Eq. (20)-(22) to avoid quadratic programming but with a new constraint, $0 \leq d_i^x \leq u_i, \forall x, i$ to replace Eq. (24). It is easy to see their equivalence. The formulation now becomes MILP and keeps the original number of decision variables.

4.3 Optimal MILP/Heuristic Solutions

The optimal solution searches for the best d_i^x and u_i assignments for the hybrid load-balancing and placement problem under the assumption of using limited flexible K monitors instead of M to minimize the maximum measurement workload across them (e.g., minimize $MAX(L_i), i = 1 \dots K$). The simplified formulation is MILP problem since u_i is a binary decision variable and d_i^x is a continuous decision variable. There is a variety of optimization tools that we can leverage. In particular, we use an MILP solver (e.g., CPLEX [10]) to find the optimal solution. We refer to this solution as ‘‘Optimal’’. For small to medium size networks, the optimal load-balancing with placement solution can be readily found. However, given that the time-complexity of MILP problems are in general NP-hard, the solvers are not fast enough for large networks.

It is easy to see that the hybrid load-balancing and placement problem becomes a LP (Linear Programming) problem if the monitor placement strategy is given (i.e., with fixed u_{ij}). Therefore, all of our proposed heuristic solutions tend to decide the monitor locations first. In this section, we propose two heuristic solutions to approximate the optimal performance: ‘‘LB-Successive Selection’’ and ‘‘LB-Greedy’’. Both of them iteratively select monitors to disable, based on the different planned monitor placement strategy decided from the previous iteration. They all start from an initial configuration under the assumption that all M routers are fully deployed with monitors. We refer to this initial configuration as the ‘‘All-On’’ stage. The monitor-disable process is repeated until only K out of M monitors are left.

LB-Successive Selection: it starts from the initial All-On configuration where all M routers are assumed to be fully deployed with monitors, and iteratively chooses one monitor to disable after optimization process (i.e., minimize $MAX(L_i)$) until only K out of M monitors are left. The selection of which monitor to disable is based on their ranking of

Algorithm 1 LB-Successive Selection Algorithm

```

1: while More than  $K$  monitors are left do
2:   Minimize  $\alpha$  by using all remaining monitors
3:   find the corresponding  $d_i^x$ 
4:   for Each remaining monitor  $V_i \in \hat{M}$  do
5:     Calculate  $L_i$  for each remaining monitor with  $d_i^x$ 
6:     Calculate one of the three metrics with  $d_i^x$ 
7:   end for
8:   Find monitor with minimum  $L_i$ 
9:   if only one monitor has minimum  $L_i$  then
10:     Disable that monitor
11:   else
12:     Disable the monitor with minimum  $L_i$  and least
       performance-metric
13:   end if
14: end while

```

measurement workload (e.g., L_i). We choose the one having least measurement workload across all remaining monitors (e.g., $V_i = \min(L_i), i = 1 \dots \hat{M}$), where \hat{M} stands for the set of remaining routers deployed with monitors. The intuition is that the monitors with higher measurement workloads after optimization process (i.e., minimize $MAX(L_i), \forall i$, the maximum workload across all monitors) take more measurement responsibility for the traffic from some OD-pairs which have few monitors deployed in their routing paths. Therefore, those monitors can not be disabled, otherwise their assigned measurement task can not be further redistributed. If more than two routers have the same minimum measurement workload in each iteration, LB-Successive Selection calculates one of the following three metrics which are served as tie-breaker and disables the one with least value:

- *Least-traffic* ($\sum_{x:V_i \in P_x} \Phi_x$). The intuition is that the monitors with the least amount of traffic passing through them have less freedom to load-balance the measurement tasks for each OD-pair traffic.
- *Least-LB(uniform)*. We use LB(uniform) heuristic mentioned in Section 3.3 to find corresponding measurement workload across all remaining monitors to serve as our second-stage tie-breaker.
- *Least-LB(weighted)*. We use LB(weighted) heuristic to find corresponding measurement workload across all remaining monitors to serve as our second-stage tie-breaker.

In particular, it disables monitor based on their ranking calculated from the previous iteration (Line 12). This means we use the information from the previous iteration (i.e., planned measurement fraction d_i^x) to calculate the metric for each monitor in the current iteration (Line 5-6).

LB-Greedy Algorithm: similar to LB-Successive Selection, the LB-Greedy algorithm also disables one monitor in each iteration, until K monitors are left. However, it is more time-consuming since it tests all remaining monitors one-by-one in each iteration. To test the importance of each monitor, LB-Greedy re-computes the minimized α after turning off each monitor alternately (Line 2-7), which essentially involves numerous optimization procedure (Line 4) mentioned in Section 3.3. Based on the testing of every remaining monitor, it disables the one that has least impact on α (Line 8).

Since the LB-Greedy algorithm exhaustively tests individual

monitor in each iteration, its performance is expected to be close to the optimal MILP solution. However it is still sub-optimal since LB-Greedy only tests individual monitor instead of every possible combination. Besides, the algorithm remains computationally costly, since it tests $O(\hat{M})$ monitors with $O(M)$ LP problems in each iteration. For a moderate sized topology, an MILP solver can sometimes work faster than this LB-Greedy approach. Details are shown in Section 5.3.2.

Algorithm 2 LB-Greedy Algorithm

```

1: while More than  $K$  monitors are left do
2:   for Each remaining monitor  $V_i \in \hat{M}$  do
3:     Disable the monitor,  $V_i$ 
4:     Minimize  $\alpha$  based on remaining monitors
5:     Store  $\alpha$ 
6:     Enable the monitor,  $V_i$ 
7:   end for
8:   Find monitor,  $V_i$ , with smallest  $\alpha \in \hat{M}$  when they are
   disabled
9:    $\hat{M} \leftarrow \hat{M} / \{V_i\}$ 
10: end while

```

5 PERFORMANCE EVALUATION

We evaluated the performance of LEISURE with three optimal solutions, LB(min-VAR), LB(min-MAX) and LB(min-VAR-given-MAX), for different load-balancing objectives in various realistic scenarios on two separate real, large point-of-presence(PoP)-level backbone networks: Abilene [7] and GEANT [8]. We also compare them with several simple naive approaches, namely LB(ingress), LB(egress), LB(uniform), and LB(weighted). Our starting point is to conduct a preliminary evaluation on the basic model in Section 5.2 based on three assumptions: (1) all routers are equipped with monitors that are capable of performing the measurement task, (2) traffic from each OD-pair has a single router-level path by OSPF and (3) there is only one measurement task. We relax the first assumption in Section 5.3 to show our load-balancing ability and the computation time complexity in this more general case. We further relax the second and third assumptions and discuss their evaluations in Supplemental Section 10.

5.1 Experimental Setup and Performance Metrics

We use two real datasets from the Abilene [7] and GEANT networks [8], both of which have been studied and discussed in the research literature. Their data sets are publicly available, including network topology, routing information. Based on these available data sets, we implemented a flow-based trace-driven simulation to conduct our evaluations. For both networks, we use the real traffic matrices provided by a third party [11]. The traffic matrix data sets for the Abilene network are available at [12], and the traffic matrix data sets of the GEANT network are available at [13].

Abilene: A public academic network in the U.S. with 11 nodes interconnected by OC192, 10 Gbits/s links. The traces we use were collected from April 22-26, 2004. The routers in ATLA, CHIN, DENV, HSTN, IPLS, KSCY, LOSA, NYCM, SNVA, STTL and WASH are denoted as R_0, R_1, \dots, R_{10} respectively.

GEANT: It connects a variety of European research and education networks. Our experiments were based on the December 2004 snapshot available at [14], which consists of 23

nodes and 74 links varied from 155 Mbits/s to 10 Gbits/s. The traces we use were collected from April 11-15, 2004.

The traffic matrix we use consists of demands for every OD-pair within a certain time interval (5 mins for Abilene and 15 mins for GEANT). We construct OD-pairs by considering all possible pairs of PoPs and calculate their shortest-path routes. In brief, these traffic matrices are derived from flow information collected at key locations of the network, and is transformed into the demand rate for each OD-pair based on the control plane information.

In the following sections, we assume our target is to measure all traffic (i.e., $a_x = 1, \forall x \in \Theta$). Therefore the workload L_i for router R_i ($i = 1 \dots M$) is defined as the traffic amount that router R_i measured normalized by the total traffic demand. Theoretically, the ideal load-balancing workload L_i for M monitors is $\frac{1}{M}$. However, it might be unachievable due to routing limitations from TE or resource constraints on monitors. In our experiments, we are interested in the following three performance metrics:

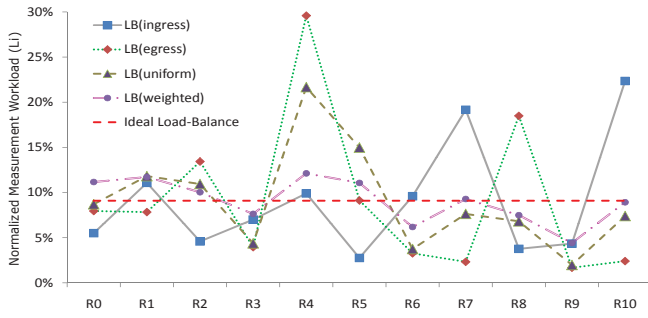
- *Maximum Workload:* We use the maximum value of each monitor’s measurement workload in the entire network to serve as our load-balancing performance metric mainly (e.g., $\text{MAX}(L_i), i=1 \dots M$).
- *Variance of Workload:* The other load-balancing performance metric used in this paper is the variance of workloads across all monitors (e.g., $\text{VAR}(L_i)$).
- *Computation Time:* In our experiment, we only collect computation time for the LP or MILP solver since they usually take much longer time compared to normal numerical computation, and therefore dominate the whole computation time of LEISURE. Meanwhile, the computation time for LP or MILP may vary for different solvers. We therefore do not mix them with other numerical computations.

5.2 Basic Load-Balancing Comparison

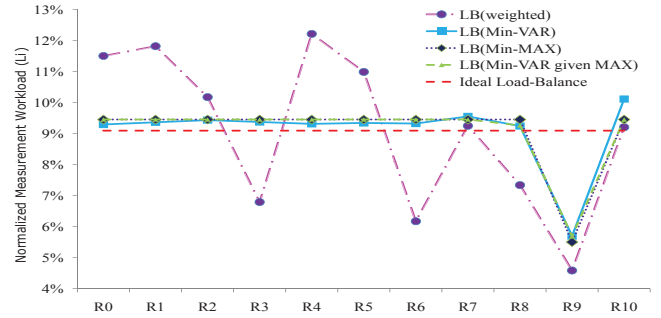
In this section, we compare the load-balancing performance of all approaches based on two assumptions (ubiquitous monitors and single path routing). Table 3 compares $\text{MAX}(L_i)$ of all monitors for different approaches. For GEANT, our optimal load-balancing solutions can reduce $\text{MAX}(L_i)$ by a factor of 4.75X(= $\frac{28.79\%}{6.06\%}$) when compared to the naive approach of LB(ingress) and 2.27X(= $\frac{13.73\%}{6.06\%}$) when compared to LB(uniform). Similar gains can be seen in the results for Abilene as well. Fig. 2 plots in more details the L_i values of 11 monitors and 23 monitors for different load-balancing approaches in Abilene and GEANT networks, respectively².

Another relative performance measure is to see how close the maximum workloads are in comparison to the ideal load-balancing case of $\bar{L} = \frac{1}{M}$, as given by Eq.(8). For Abilene and GEANT, the ideal \bar{L} is 9.09%(= $\frac{1}{11}$) and 4.35%(= $\frac{1}{23}$), respectively. However, the $\text{MAX}(L_i)$ of LB(ingress) for Abilene and GEANT are 19.16% and 28.79%, respectively, which are 2.11X and 6.62X worse than the ideal case. For simple heuristic approaches, they still have large $\text{MAX}(L_i)$ values compared to the ideal case: e.g., 21.67% (2.3X worse) for LB(uniform) in Abilene and 10.67% (2.4X worse) for LB(weighted) in

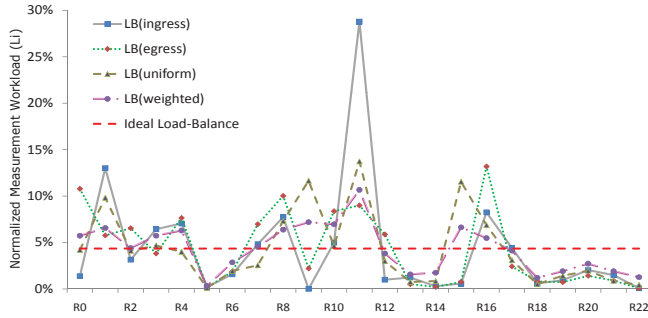
² Ideally if the network topology is fully-meshed, LEISURE can achieve the ideal load-balance performance. However due to the limitation of network topology and routing strategy(e.g., OSPF), the peripheral routes are not able to share the majority measurement task concentrated in core network (e.g., R9(Seattle) in Abilene [7]).



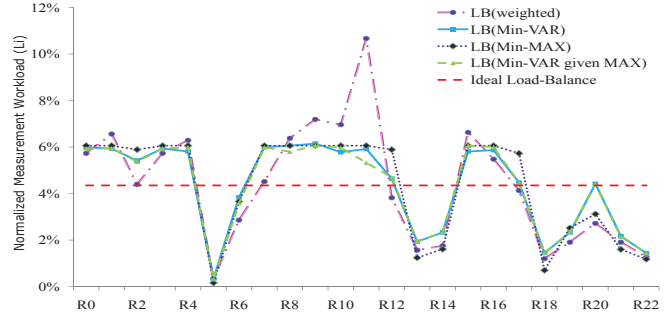
(a) Simple Heuristic Approaches in Abilene



(b) Optimal Approaches in Abilene



(c) Simple Heuristic Approaches in GEANT



(d) Optimal Approaches in GEANT

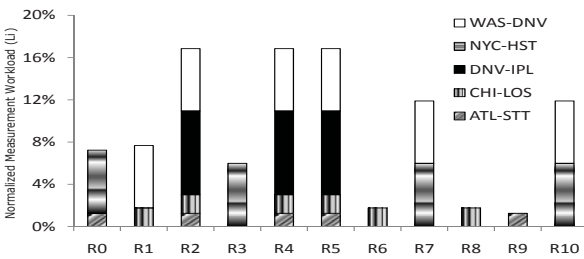
Fig. 2. Measurement load distribution for different approaches in Abilene and GEANT

TABLE 3
Comparisons on Maximum value of L_i

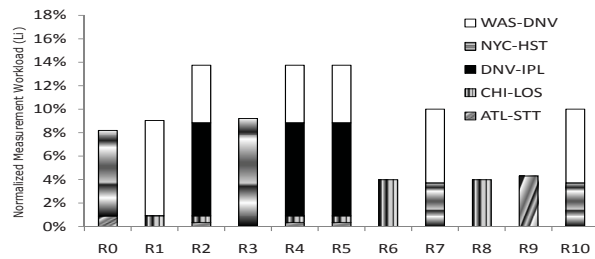
Network	Naive Approaches		Heuristic Approaches		Optimal Load-Balancing		
	ingress	egress	uniform	weighted	min-VAR	min-MAX	min-VAR given MAX
Abilene	19.16%	29.59%	21.67%	12.12%	10.11%	9.45%	9.45%
GEANT	28.79%	13.19%	13.73%	10.67%	6.15%	6.06%	6.06%

TABLE 4
Comparisons on Variance of L_i

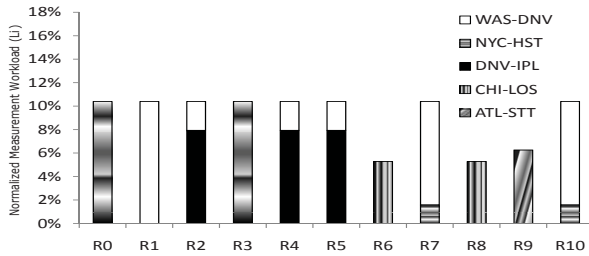
Network	Naive Approaches		Heuristic Approaches		Optimal Load-Balancing		
	ingress	egress	uniform	weighted	min-VAR	min-MAX	min-VAR given MAX
Abilene	0.004107	0.007366	0.003158	0.000602	0.000105	0.000131	0.000105
GEANT	0.003978	0.001626	0.001594	0.000662	0.000378	0.000495	0.000378



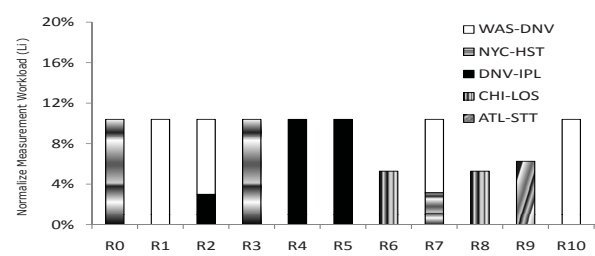
(a) LB(Uniform)



(b) LB(Weighted)



(c) LB(Min-MAX)



(d) LB(Min-VAR given MAX)

Fig. 3. Detailed Abilene results for five OD-pairs. Optimal solutions allow nodes to be excluded from measurement if they are already overloaded.

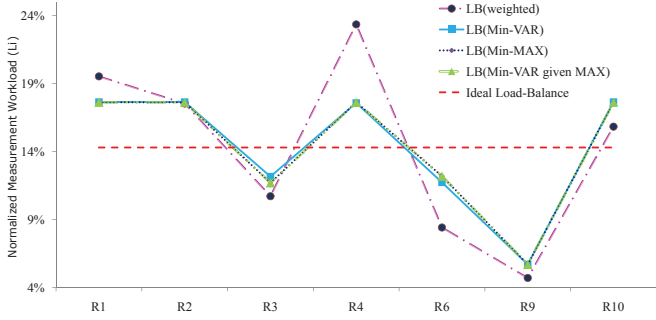


Fig. 4. Measurement load distribution with limited 7 out of 11 monitors in Abilene.

GEANT. On the other hand, our three optimal load-balancing solutions presented in Fig. 3 and Table 3 perform very close to the theoretical ideal case: 10.11%, 9.45%, and 9.45% for LB(min-VAR), LB(min-MAX), and LB(min-VAR given MAX), respectively, as compared to the ideal case of 9.09% for Abilene. Similarly, our three optimal solutions are 6.15%, 6.06%, and 6.06%, respectively, as compared to the ideal case of 4.35% for GEANT.

Table 4 compares $\text{VAR}(L_i)$ across all monitors for different approaches. For Abilene, our optimal load-balancing solutions can reduce $\text{VAR}(L_i)$ by a factor of 70X ($=\frac{0.007366}{0.000105}$) when compared to the naive approach of LB(egress), and over 30X ($=\frac{0.003158}{0.000105}$) when compared to LB(uniform). Similar improvements in variance can be seen for GEANT as well.

To better understand why our optimal solutions can achieve more evenly distributed measurement load, we use traffic from only five OD-pairs in Abilene³ to show the detailed load assignment in Fig. 3 (WAS-DNV, NYC-HST, DNV-IPL, CHI-LOS and ATL-STT with 66.5 MB, 44.9 MB, 44.6 MB, 19.8 MB and 11.7 MB, respectively). In Fig. 3(a), although LB(uniform) distributes each OD-pair traffic to all monitors in the path uniformly (e.g., WAS-DNV with 6 monitors), the aggregated workload for overall measurement task in each monitor is still unbalanced (e.g., L_i for all routers R_i ($i = 1 \dots 10$) are distributed between 1% to 17%). LB(weighted) in Fig. 3(b) improves the load-balancing performance due to the global view it has but still load-balanced poorly (e.g., L_i distributed between 4% to 14%). In contrast, the optimal solutions can achieve much better load-balancing performance (e.g., L_i distributed between 5.5% to 10.5%) by excluding some monitors from measuring certain OD-pair traffic (e.g., R4 and R5 do not measure traffic for WAS-DNV OD-pair in Fig. 3(d)).

5.3 Limited Number of Monitors

In this section, we relax our first assumption to the case that only a subset of routers are deployed monitors and capable of measurement. We further evaluate LEISURE in the cases of fixed and flexible monitor deployment.

5.3.1 Fixed Monitor Deployment Scenario

In the first case, we assume there are $K = 7$ out of $M = 11$ routers are deployed with fixed monitors in Abilene. The routers which are excluded to deployed monitors are R_0 ,

3. The notations of these OD-pairs and their routing information could be found in [7], [12].

R_5 , R_7 and R_8 ⁴. Therefore LEISURE can only distribute the measurement task to the remaining 7 monitors. We omit naive approaches and focus on heuristic (i.e., LB(weighted)) and optimal approaches in Fig. 4. Compared with ubiquitous case in Fig. 2(a), the ideal load-balancing workload is increased from 9.09% to 14.29%. For LB(min-VAR), LB(min-MAX) and LB(min-VAR given MAX), the $\text{MAX}(L_i)$ is only increased from 9.67% to 17.61%. However, for heuristic approaches, $\text{MAX}(L_i)$ increased from 12.12% to 23.33% for LB(weighted), and 21.67% to 35.86% for LB(uniform). We observe that LEISURE with these three optimal solutions for different load-balancing objectives only increased 7.94% workload for $\text{MAX}(L_i)$, which are close to 5.2% for the theoretical ideal case and are much better than 11.21% for LB(weighted) and 14.19% for LB(uniform).

5.3.2 Flexible Monitor Deployment Scenario

In the second case, we assume there are limited K out of $M = 11$ routers are deployed with flexible monitors in Abilene (e.g., $K=11, 10, 9, 8, 7$ and 6). Therefore LEISURE can only distribute the measurement task to these K monitors. Fig. 5 plots the detailed L_i (normalized measurement workload) values of monitors for different load-balancing approaches in Abilene. Compare to the ideal load-balancing case of $\bar{L} = \frac{1}{K}$, as given by Eq.(8) where K is the limited number of deployed monitors, our optimal MILP solution (e.g., LB(Optimal)) performs almost the same as the ideal case. As shown in Fig. 5(a), for $K=11, 10, 9, 8, 7$ and 6 in Abilene, the L_{max} ($L_{max} = \text{minimized MAX}(L_i), i = 1 \dots M$) of LB(Optimal) is 9.43%, 10.0%, 11.11%, 12.5%, 14.29% and 16.96%, respectively while the ideal \bar{L} is 9.09% ($=\frac{1}{11}$), 10.0% ($=\frac{1}{10}$), 11.11% ($=\frac{1}{9}$), 12.5% ($=\frac{1}{8}$), 14.29% ($=\frac{1}{7}$) and 16.67% ($=\frac{1}{6}$).

Next we show that the proposed heuristic algorithms, “Greedy” and “Successive Selection” can achieve nearly equivalent load-balancing performance, L_{max} , as LB(Optimal) while the potentially long computation time could be reduced dramatically. In Fig. 5(b), the L_{max} for $K=11, 10, 9, 8, 7$ and 6 in LB-Greedy almost have the same values as in LB(Optimal) in Fig. 5(a) although their deployment of monitors might be different. The same observation could be also found in LB-Successive Selection(traffic)/LB-Successive Selection(uniform), and LB-Successive Selection(weighted)⁵ (e.g., see Fig. 5(c)). Compared to the simple basic LB(weighted) whose L_{max} for $K=11, 10, 9, 8, 7$ and 6 is 12.12%, 12.74%, 13.70%, 14.89%, 16.85% and 19.28% respectively as shown in Fig. 5(d), LB-Greedy/LB-Successive Selection(weighted) improves the load-balancing performance a lot, not to mention the basic LB(uniform).

6 CONCLUSION

In this paper, we proposed an optimization framework, LEISURE, for load-balancing network-wide traffic measurements across coordinated monitors in the network. This is an important problem because without judiciously load-balancing measurement tasks across monitors, the existing monitors can be overwhelmed due to competing measurement tasks or

4. The reason to choose those 4 excluded routers is to maintain the fact that at least one capable monitor in each OD-pair’s route to fulfill the measurement tasks imposed by the network operator.

5. We omit the result of LB-Successive Selection(traffic)/LB-Successive Selection(uniform) since they perform close to LB-Successive Selection(weighted).

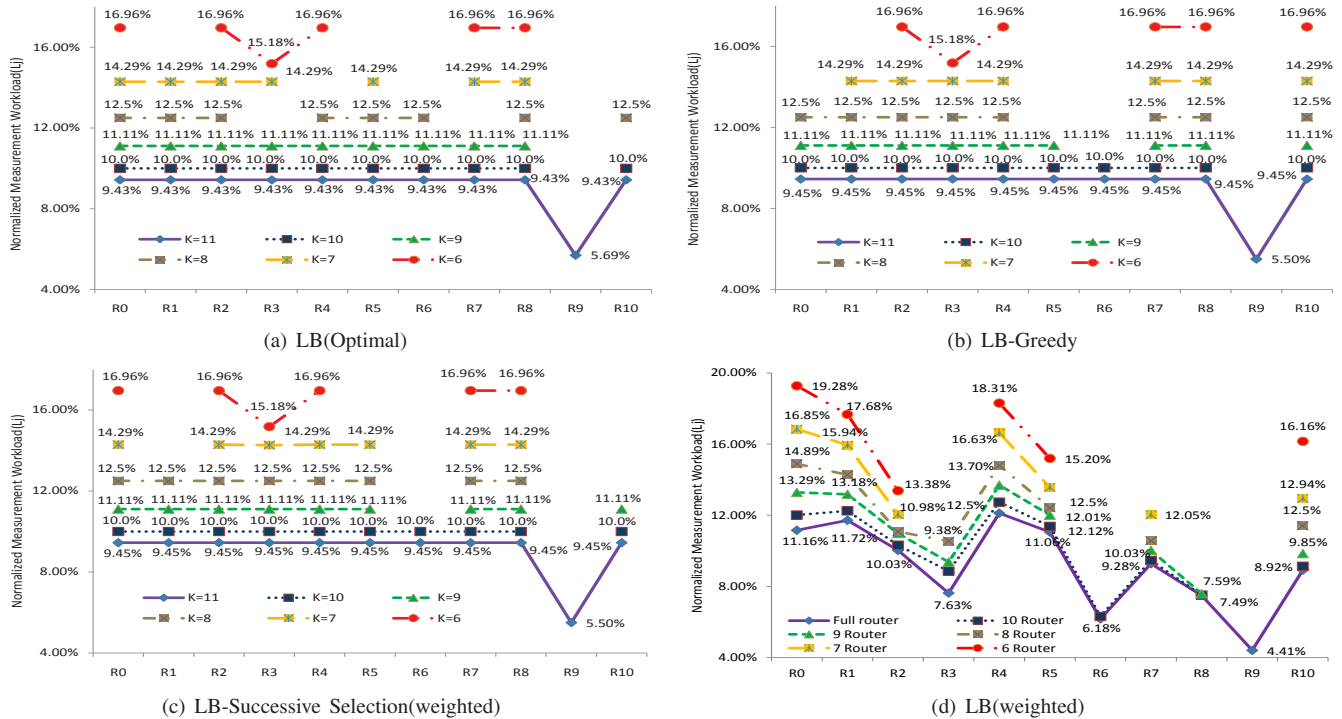


Fig. 5. Measurement load distribution for different approaches in Abilene with limited K flexible deployed monitor

unexpectedly large traffic volume. Based on our simulation measurement studies using the Abilene and GEANT networks, we show that LEISURE can achieve up to 4.75X smaller maximum measurement workload and 70X smaller variance across all coordinated monitors.

REFERENCES

- [1] C.-W. Chang and G. Huang and B. Lin and C.-N. Chuah, "LEISURE: A Framework for Load-Balanced Network-Wide Traffic Measurement," in *Proceedings of the 7th ACM/IEEE ANCS*, 2011.
- [2] C.-W. Chang, A. Gerber, B. Lin, S. Sen, and O. Spatscheck, "Network dv: A programmable framework for application-aware trace collection," in *Passive and Active Measurement Conference*, 2000.
- [3] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern-matching using team," in *Proceedings of IEEE ICNP*, 2004, pp. 174–183.
- [4] S. Dharmapurikar, P. Krishnamurthy, T. Sproull, and J. Lockwood, "Deep packet inspection using parallel bloom filters," in *IEEE Micro*, 2003, pp. 44–51.
- [5] G. Huang, A. Lall, C.-N. Chuah, and J. Xu, "Uncovering Global Icebergs in Distributed Monitors," in *IEEE IWQoS*, 2009.
- [6] V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen, "CSAMP: A System for Network-Wide Flow Monitoring," in *Proceedings of USENIX NSDI*, 2008.
- [7] V. Chan, "Near-term future of the optical network in question?" *IEEE J. Sel. Areas Commun.*, vol. 25, 2007.
- [8] M. Yoo, C. Qiao, and S. Dixit, "Optical burst switching for service differentiation in the next-generation optical Internet," *IEEE Commun. Magazine*, vol. 39, no. 2, pp. 98–104, 2001.
- [9] Y. Zhang, M. Roughan, N. Duffield, , and A. Greenberg, "Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads," in *Proceedings of ACM SIGMETRICS*, 2003.
- [10] "Ibm cplex software," <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>.
- [11] "Totem, a toolbox for traffic engineering methods, february 2005," <http://totem.info.ucl.ac.be>.
- [12] "Abilene traffic matrices," <http://www.cs.utexas.edu/~y Zhang/research/AbileneTM>.
- [13] "Totem geant traffic matrices," <http://totem.info.ucl.ac.be/dataset.html>.
- [14] "Geant topology available at," http://www.geant.net/upload/pdf/GEANT_Topology_12-2004.pdf.
- [15] A. Ramachandran, S. Seetharaman, N. Feamster, and V. Vazirani, "Fast Monitoring of Traffic Subpopulations," in *Proceedings of ACM Internet Measurement Conference*, 2008.
- [16] L. Yuan, C.-N. Chuah, and P. Mohapatra, "ProgME: Towards Programmable Network MEasurement," in *Proceedings of ACM SIGCOMM*, 2007.
- [17] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies," in *Proceedings of ACM SIGCOMM*, 2004.
- [18] K. Suh and Y. Guo and J. Kurose and D. Towsley, "Locating network monitors: Complexity, heuristics and coverage," in *Proceedings of IEEE INFOCOM*, March 2005.
- [19] M. R. Sharma and J. W. Byers, "Scalable Coordination Techniques for Distributed Network Monitoring," in *Proceedings of Passive and Active network Measurement (PAM) workshop*, April 2005.
- [20] C. Chaudet, E. Fleury, I. Lassous, H. Rivano, and M.-E. Voige, "Optimal positioning of active and passive monitoring devices," in *Proceedings of ACM CoNEXT*, 2005, pp. 71–82.
- [21] G. Sagnol, M. Bouhtou, and S. Gaubert, "Successive c-optimal designs: A scalable technique to optimize the measurements on large networks," in *Proceedings of ACM SIGMETRICS*, 2010.
- [22] J. S. Cuxart, P. B. Ros, R. Kompella, and N. Duffield, "Cuckoo Sampling: Robust Collection of Flow Aggregates under a Fixed Memory Budget," in *Proceedings of IEEE INFOCOM Mini-Conference*, April 2012.
- [23] M. Lee, M. Hajjat, R. Kompella, and S. Rao, "RelSamp: Preserving Application Structure in Sampled Flow Measurements," in *Proceedings of IEEE INFOCOM*, April 2011.
- [24] M. Saxena and R. Kompella, "A framework for efficient class-based sampling," in *Proceedings of IEEE INFOCOM Mini-Conference*, April 2009.
- [25] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran, "Reformulating the monitor placement problem: Optimal network-wide sampling," in *Proceedings of ACM CoNEXT*, 2006.
- [26] "Cisco ios netflow," <http://www.cisco.com/warp/public/732/Tech/netflow/>.
- [27] "The OpenFlow Switch Consortium," <http://www.openflowswitch.org>.
- [28] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. V. D. Merwe, "Design and implementation of a Routing Control Platform," in *Proceedings of USENIX NSDI*, 2005.
- [29] H. Ballani and P. Francis, "CONMan: A Step Towards Network Manageability," in *Proceedings of ACM SIGCOMM*, 2007.
- [30] S. Raza, G. Y. Huang, C.-N. Chuah, S. Seetharaman, and J. P. Singh, "MeasuRouting: A Framework for Routing-Assisted Traffic Monitoring," in *IEEE INFOCOM*, 2010.
- [31] G. Huang, C.-W. Chang, C.-N. Chuah, and B. Lin, "Measurement-Aware Monitor Placement and Routing: A Joint Optimization Approach for Network-Wide Measurements," *IEEE Transactions on Network and Service Management*, vol. 9, pp. 48–59, March 2012.
- [32] N. Duffield and M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation," in *Proceedings of ACM SIGCOMM*, 2001.



Chia-Wei Chang holds a Ph.D. degree in Electrical and Computer Engineering from the University of California, San Diego. He also received his B.S. and M.S. degrees in Communication Engineering from the National Chiao-Tung University, Hsinchu, Taiwan. Dr. Chang received the 2004 Prize of the Graduate Student thesis Contest from CIEE, Taiwan, and the CAL-IT² 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.



Guanyao Huang holds a Ph.D. degree in Electrical and Computer Engineering from the University of California, Davis. He also received undergraduate and post-graduate degrees in Electrical Engineering from the University of Science and Technology of China, Hefei, China. His current research focuses on network measurement, anomaly detection, and online social 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 a Professor in Electrical and Computer Engineering and Adjunct Professor in Computer Science and 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 Qualcomm Institute in industry-sponsored research efforts. His research has led

to over 160 journal and conference publications, including a number of Best Paper awards and nominations. He also holds 4 awarded patents. He has served on the executive and technical program committee of many ACM and IEEE conferences. He currently serves as the Associate Editor for the ACM Transactions on Design Automation of Electronics Systems.



Chen-Nee Chuah is a Professor in Electrical and Computer Engineering at the University of California, Davis. She received her Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley. Her research interests include Internet measurements, network management, anomaly detection, online social networks, and vehicular ad hoc networks. Chuah is an ACM Distinguished Scientist. She received the NSF CAREER Award in 2003, and the Outstanding Junior Faculty Award from

the UC Davis College of Engineering in 2004. In 2008, she was named a Chancellor's Fellow of UC Davis. She has served on the executive/technical program committee of several ACM and IEEE conferences. She served as the Associate Editor for IEEE/ACM Transactions on Networking from 2008-2013.