

Robust Network Tomography in the Presence of Failures

S. Tati*, S. Silvestri*, T. He[†], T. La Porta*

*Department of Computer Science and Engineering
Pennsylvania State University, USA

{tati, simone, tlp}@cse.psu.edu

[†]IBM T.J. Watson Research Center

Yorktown, NY, USA

the@us.ibm.com

Abstract—In this paper, we study the problem of selecting paths to improve the performance of network tomography applications in the presence of network element failures. We model the robustness of paths in network tomography by a metric called *expected rank*. We formulate an optimization problem to cover two complementary performance metrics: robustness and probing cost. The problem aims at maximizing the expected rank under a budget constraint on the probing cost. We prove that the problem is NP-Hard. Under the assumption that the failure distribution is known, we propose an algorithm called *RoMe* with guaranteed approximation ratio. Moreover, since evaluating the expected rank is generally hard, we provide a bound which can be evaluated efficiently. We also consider the case in which the failure distribution is not known, and propose a reinforcement learning algorithm to solve our optimization problem, using *RoMe* as a subroutine. We run a wide range of simulations under realistic network topologies and link failure models to evaluate our solution against a state-of-art path selection algorithm. Results show that our approaches provide significant improvements in the performance of network tomography applications under failures.

I. INTRODUCTION

In the Internet and complex wide-area networks, network management involves a wide range of tasks such as fault detection, performance diagnosis, resource allocation, route selection and congestion control. Most of these tasks require a complete knowledge of internal network state and network topology. Network tomography techniques [1], [2], [3] are proposed to acquire this information efficiently probing only end-to-end (e2e) paths from monitors located at the network edges, instead of directly monitoring every network element. Applications of network tomography include, but are not limited to, inference of individual link performance metrics from given e2e path measurements [1], network topology inference [2], and estimation of the complete set of e2e measurements from an incomplete set [3].

A commonly adopted approach in network tomography is to formulate a linear system that models the relationship between path measurements and individual link metrics. Given the *candidate paths* between monitors, state-of-art solutions in network tomography select a subset of these paths, determined by finding an arbitrary *basis* of the linear system. By probing the paths in a basis, previous approaches [1], [4], [3] reduce the

overhead of collecting e2e measurements while maximizing the performance.

Existing work assumes a simple network model, where all network elements are reliable. However, failure of network elements are common events in modern networks due to maintenance procedures, hardware malfunctions, energy outages, or disasters [5]. The typical duration of link failures in IP networks [5] are longer than the lengths of time windows for measurement collection [6] in network tomography. Hence, the link failures may prevent the collection of some measurements, and this degrades the performance of network tomography applications. As a result, previous approaches may perform poorly in the presence of failures.

Furthermore, in practice, probing a path and centrally collecting measurements at a Network Operating Center (NOC) where network tomography applications are executed, both incur costs. These costs vary across different paths. As a result, the set of paths to probe has to be carefully selected in order to maximize the performance of network tomography within a budget constraint.

In this paper, for the first time, we consider the optimization of path selection in network tomography in the presence of network element failures and heterogeneous probing costs. We consider two different scenarios. In the first scenario, we assume that the distribution of network elements failures is known. We model the robustness of a set of paths by means of a metric called *Expected Rank (ER)*, defined as the expected rank of the linear system corresponding to the remaining paths after the failures occur.

We formulate the measurement selection problem as an optimization problem which maximizes *ER* under a given budget constraint on the probing cost. We show that the problem is NP-Hard, but recognize a special property of *ER* that allows the development of an approximate algorithm called Robust Measurements (RoMe), which has an approximation factor of $1 - \frac{1}{\sqrt{e}}$. We note that an exact implementation of RoMe would be hindered by the high complexity in evaluating *ER*. To address this issue, we derive an analytical bound on *ER* that can be evaluated efficiently. Moreover, we show that our solution becomes optimal in the more constrained case of selecting only linearly independent paths.

In the second scenario, we assume no prior knowledge on

This work is published in the conference of IEEE International Conference on Distributed Computing Systems (ICDCS), 2014 at Madrid, Spain.

the failure distribution. We propose a reinforcement learning algorithm, called Learning with Submodular Rewards (LSR), in order to learn the failure distribution through the observations of e2e measurements. LSR makes use of RoMe during the learning process. We prove that under certain conditions LSR has bounded performance with respect to the optimal strategy that would be performed by knowing the failure distribution.

The main contributions of this paper are the following:

- We show, for the first time, that a proper selection of paths can significantly improve the performance of network tomography applications in the presence of failures.
- We define an optimization problem for path selection which maximizes the robustness against failures under a budget constraint on probing cost, assuming a known failure distribution. We show that the problem is NP-Hard in general and provide an efficient solution. Moreover, we show that our solution becomes optimal in the more constrained case when selected paths must be linearly independent.
- We propose a reinforcement learning algorithm to pick a set of robust paths while learning path robustness, when no prior knowledge about failure distribution is available.
- We confirm through simulations the benefit of our approach in comparison with previous solutions in realistic network scenarios. The results show that our proposed solution significantly improves the performance of network tomography applications under failures.

The rest of the paper is organized as follows. Section II motivates the need of robust measurement design amidst failures. Section III formulates the problem. Sections IV and V present our solutions under known and unknown failure distributions, respectively. Section VI evaluates our solutions against benchmarks. Section VII reviews related work. Finally, Section VIII concludes the paper.

II. NETWORK TOMOGRAPHY AMIDST FAILURES

In this section, we briefly provide the background of network tomography techniques, and show how failures of network elements can significantly affect their performance.

A. Background on network tomography

Network tomography techniques [1], [2], [3] model the network as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. A subset of the nodes $M \subseteq V$, usually nodes at the edge of the network, may act as *monitors*. A single path is assumed between each pair of monitors, as usually provided by routing algorithms in the Internet [7]. We denote by R_M the set of *candidate paths* between monitors that can be selected for probing. Monitors probe each other to collect *e2e measurements* over the selected paths. In this work, we consider typical applications of network tomography including inferring the individual link metrics from e2e measurements and inferring the complete set of e2e

measurements from a subset. A typical assumption in these applications is that the metric of interest is additive across links, which holds for delay and logarithm of packed delivery rate.

Given the set of candidate probing paths R_M , we define a *path matrix* \mathbf{A} of size $|R_M| \times |E|$. If a path $q_i \in R_M$ contains link j then $\mathbf{A}[i, j] = 1$; otherwise $\mathbf{A}[i, j] = 0$. Let $y_{R_M} \in \mathbb{R}^{|R_M|}$ be a column vector, where y_i represents an e2e measurement of path q_i . Also, let $x \in \mathbb{R}^{|E|}$ be a column vector where x_j is the unknown metric of link l_j . Since additive link metrics are considered, we can write a linear system for all the e2e measurements of paths in R_M as,

$$\mathbf{A}x = y_{R_M} \quad (1)$$

The rank of matrix \mathbf{A} , hereafter denoted by $r(\mathbf{A})$, is equal to the number of linearly independent rows in \mathbf{A} (*row rank*) and also to the number of linearly independent columns of \mathbf{A} (*column rank*).

In real large-scale networks, the linear system is usually underdetermined, since the rank is smaller than the number of links [3], i.e., $r(\mathbf{A}) < |E|$. A *basis* of paths $R_b \subseteq R_M$ is a maximal set of linearly independent paths, where we say that paths are independent if all corresponding rows in the matrix \mathbf{A} are linearly independent.

Previous approaches in network tomography [1], [4], [3], probe an arbitrary basis of paths determined by rank decomposition techniques. A basis can be used to determine the unique solution for a maximal set of links [1], called *identifiable* links, from the linear system in Eq. 1. Moreover, it allows to reconstruct the set of all e2e measurements in R_M , resulting in a scalable monitoring system in large networks [3]. These approaches, assume that all network elements are reliable and that paths have homogeneous probing costs. As a result the may perform poorly in the presence of failures and heterogeneous costs.

B. Impact of link failures

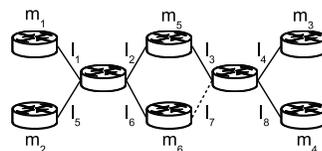


Fig. 1: An example

In this section, we motivate our problem by showing how a careful selection of candidate paths improves the performance of network tomography applications in the presence of failures. We first provide an illustrative example, and then give some motivating results for realistic networks.

We consider a topology with 8 nodes and 8 links as shown in Figure 1. In this example, we assume that monitors are placed on nodes that are labeled m_1, \dots, m_6 . The candidate paths are given in Figure 2 (a). The path matrix \mathbf{A} is shown in Figure 2 (b). In the example, we focus on the application in which we are interested in identifying the metrics of the links from e2e measurements.

Monitors	Path
(m_1, m_3)	$q_1 = (l_1, l_6, l_7, l_4)$
(m_2, m_4)	$q_2 = (l_5, l_6, l_7, l_8)$
(m_2, m_3)	$q_3 = (l_5, l_6, l_7, l_4)$
(m_1, m_4)	$q_4 = (l_1, l_6, l_7, l_8)$
(m_1, m_2)	$q_5 = (l_1, l_5)$
(m_3, m_4)	$q_6 = (l_4, l_8)$
(m_1, m_5)	$q_7 = (l_1, l_2)$
(m_2, m_5)	$q_8 = (l_5, l_2)$
(m_5, m_3)	$q_9 = (l_3, l_4)$
(m_5, m_4)	$q_{10} = (l_3, l_8)$
(m_2, m_6)	$q_{11} = (l_3, l_7)$
(m_1, m_6)	$q_{12} = (l_1, l_6)$
(m_2, m_6)	$q_{13} = (l_5, l_6)$
(m_6, m_3)	$q_{14} = (l_7, l_4)$
(m_5, m_6)	$q_{15} = (l_7, l_8)$

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8
q_1	1	0	0	1	0	1	1	0
q_2	0	0	0	0	1	1	1	1
q_3	0	0	0	1	1	1	1	0
q_4	1	0	0	0	0	1	1	1
q_5	1	0	0	0	1	0	0	0
q_6	0	0	0	1	0	0	0	1
q_7	1	1	0	0	0	0	0	0
q_8	0	1	0	0	1	0	0	0
q_9	0	0	1	1	0	0	0	0
q_{10}	0	0	1	0	0	0	0	1
q_{11}	0	0	1	0	0	0	1	0
q_{12}	1	0	0	0	0	1	0	0
q_{13}	0	0	0	0	1	1	0	0
q_{14}	0	0	0	1	0	0	1	0
q_{15}	0	0	0	0	0	0	1	1

Fig. 2: (a) Candidate paths, (b) path matrix.

In the linear system corresponding to the path matrix \mathbf{A} , any basis enables the unique identification of all links, since \mathbf{A} has rank 8 and there are 8 unknown link metrics. However, this is not the case when failures occur. In this example, we consider the failure of link l_7 , due to which the paths that contain this link are not available to provide e2e measurements.

Previous path selection approaches [1], [4], [3], select an arbitrary basis such as $R_1 = (q_1, q_2, q_4, q_{11}, q_{15}, q_5, q_6, q_7)$. When l_7 fails, only paths q_5, q_6 and q_7 can be successfully probed in R_1 , hence the provided rank is 3. The surviving paths do not cover the links l_3 and l_6 at all, moreover the corresponding linear system has infinite solutions for each link.

Given the possible failure of link l_7 , we can select a more robust basis such as $R_2 = (q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12})$. The only path affected by the failure in R_2 is q_{11} . The corresponding linear system uniquely identifies the metrics of all links except l_7 . In addition, we can also conclude, from the failures of path q_{11} , that the failed link is l_7 .

To further motivate our problem, we present some simulation results, in Figure 3, where we plot the average ranks provided by two arbitrary bases and by all paths in R_M , as we increase the number of concurrent link failures. In these simulations, we use the realistic topology AS1239 from the Rocketfuel Project [8] and we consider 1600 candidate paths in R_M (detailed simulation settings are provided in Section VI-A). The bases provide different ranks under failures, and as expected the complete set of paths provides a higher rank than both.

These results highlight the need of choosing a robust basis and also show the benefits of probing redundant paths in addition to a basis. Since probing a path incurs a cost and generates overhead, it is often undesirable to probe all paths in R_M . In this paper, we address both robustness and cost by formulating an optimization problem for path selection which aims at maximizing the expected rank under failures subject to a budget constraint. We initially consider a scenario where the failure distribution is known, and then we propose a reinforcement learning approach to learn this distribution from measurements.

Note that some papers on network tomography [9] focus solely on localizing network failures. Our objective is orthogonal to theirs, as we focus on inferring performance metrics of non-failed links through robust path selection. As mentioned in Section I, this is a significant problem due to the fact that the time of measurement collection [6] is typically much smaller than length of link failures [5].

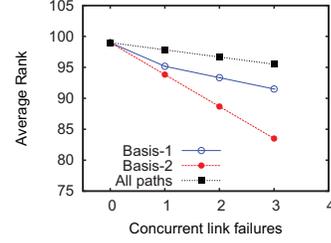


Fig. 3: Rank of a basis under failures

III. PROBLEM FORMULATION

In this section we introduce our optimization problem, after defining the expected rank function and the probing cost model.

A. Robustness of Measurements

We first formalize the concept of robustness for a single path under link failures by defining its *expected availability*, and then we define the robustness for a set of multiple paths by introducing the *expected rank*. We introduce these concepts assuming that a statistical knowledge of the failure distribution over the links in the network is available. However, in Section V we relax this assumption and propose a reinforcement learning approach to learn the failure distribution while probing.

We divide time into epochs and assume that the availability state (available/failed) of a link is persistent within an epoch, but varies i.i.d. across epochs. We also assume that availability states of different links are independent. We define a *failure vector*, $v \in \{0, 1\}^{|E|}$, such that $v[i] = 1$ if link l_i has failed in the current epoch and $v[i] = 0$ otherwise. We define a probability distribution $\mathbb{P} : \{0, 1\}^{|E|} \rightarrow [0, 1]$ over the set of all possible failure vectors, which is determined by the link failure probabilities as:

$$\mathbb{P}(v) = \prod_{i=1}^{|E|} \left((p_i v[i]) + (1 - v[i])(1 - p_i) \right) \quad (2)$$

where p_i is the failure probability of link l_i . We model the robustness of a path $q \in R_M$ under link failures by the expected availability, EA , calculated over all possible failure vectors.

$$EA(q) = \sum_{v \in \{0, 1\}^{|E|}} q(v) \quad (3)$$

where v is a given failure vector and $q(v)$ is an indicator function (it is equal to 1 if q is available, or 0 otherwise). Using the Total Probability Theorem, $EA(q)$ can be efficiently calculated by observing that q is available only if none its links have failed, and hence $EA(q) = \prod_{l_i \in q} (1 - p_i)$.

Similarly, we define the expected rank for a set of paths over all possible failure vectors. It evaluates the rank provided by the subset of paths that remain available after link failures, averaged over all link failure scenarios.

Definition 1 (Expected Rank). *Given a network $G = (V, E)$ and a probability distribution over failure vectors $\mathbb{P} : \{0, 1\}^{|E|} \rightarrow [0, 1]$, the expected rank for a set of paths R is defined as,*

$$ER(\mathbf{R}) = \sum_{v \in \{0,1\}^{|E|}} r(\mathbf{R}_v) \mathbb{P}(v) \quad (4)$$

where $\mathbf{R}_v \subseteq \mathbf{R}$ is the subset of paths that do not fail in the failure scenario v and the function r is the rank function defined on a set of paths¹.

Unlike EA , ER cannot be efficiently calculated due to possible linear dependence of paths. Generally, computing $ER(\mathbf{R})$ requires the enumeration of all possible failure vectors in $\{0,1\}^{|E|}$, resulting in an exponential complexity. We propose a low complexity bound to approximate ER in Section IV-C.

B. Probing Cost

Collecting an e2e measurement on a path q incurs a *probing cost* that is denoted by $PC(q)$. The probing cost of different paths are independent. We consider three main components in $PC(q)$: the run-time cost incurred when sending probes on the path, the run-time costs incurred when the NOC collects the measurements from monitors at the endpoints of q , and the access cost incurred by the NOC in accessing the monitor nodes belonging to other network administration domains, which can be in the case of multi-ownership wide-area networks.

Since the probing cost of paths are independent of each other, the probing cost function for a set of paths $\mathbf{R} \subseteq \mathbf{R}_M$, denoted by $PC(\mathbf{R})$, is the sum of the probing costs of the paths in \mathbf{R} . We assume that the network manager has a maximum budget \mathbf{B} and can probe any set of paths $\mathbf{R} \subseteq \mathbf{R}_M$ such that the incurred probing cost $PC(\mathbf{R})$ is no more than \mathbf{B} .

C. Problem Statement

Our problem maximizes the expected rank under a budget constraint on probing cost. It is defined as follows:

Definition 2 (Budget-constraint optimization problem). *Given a set of paths \mathbf{R}_M , the expected rank function $ER : 2^{\mathbf{R}_M} \rightarrow \mathbb{R}^+$, the probing cost function PC and a budget \mathbf{B} , find the set $\mathbf{R}^* \subseteq \mathbf{R}_M$ such that:*

$$\begin{aligned} \mathbf{R}^* &= \arg \max_{\mathbf{R} \subseteq \mathbf{R}_M} ER(\mathbf{R}), \\ \text{s.t. } PC(\mathbf{R}) &\leq \mathbf{B} \end{aligned}$$

where $2^{\mathbf{R}_M}$ is the power set of \mathbf{R}_M .

IV. OPTIMIZATION WITH STATISTICAL KNOWLEDGE OF FAILURES

In this section we consider the case in which the link failure distribution is known. Even in this case, the optimization problem defined in Section III-C is NP-Hard, as proved by the following theorem. The complete proof is provided in the appendix.

Theorem 3. *The budget-constraint optimization problem is NP-Hard.*

¹Given a set of paths \mathbf{R} , $r(\mathbf{R})$ is the rank of the submatrix of A consisting of rows corresponding to the paths in \mathbf{R} .

In the following, we propose the RoMe algorithm to solve the problem with a provable approximation bound. In addition, we show that, by considering the more constrained setting where selected paths are linearly independent and paths have unitary probing cost, RoMe provides an optimal solution.

A. RoMe: An Approximate Solution

Although proved to be NP-Hard, our problem has a special property that allows an approximate solution. Specifically, our objective function ER belongs to the family of *submodular* functions. A function is *submodular* according to the following definition.

Definition 4 (Submodular function). *Given a finite ground set E and a function $f : 2^E \rightarrow \mathbb{R}_+$, then $\forall A \subseteq B \subseteq E$, $e \in E$ f is submodular iff $f(A \cup e) - f(A) \geq f(B \cup e) - f(B)$.*

We prove that ER is submodular in the following theorem; the proof is given in the appendix.

Theorem 5. *Given a set of paths \mathbf{R}_M , the function $ER : 2^{\mathbf{R}_M} \rightarrow \mathbb{R}^+$ is submodular.*

Submodular function optimization theory provides solutions, with provable approximation bounds, to many NP-Hard problems [10]. In this paper, we propose an algorithm called *Robust Measurements* (RoMe) that adopts recent advances in this field [11] to provide a solution with proven approximation bounds through a greedy strategy.

Algorithm 1: the RoMe algorithm

Input: Set \mathbf{R}_M , budget \mathbf{B} , cost function $PC : 2^{\mathbf{R}_M} \rightarrow \mathbb{R}^+$, objective function $ER : 2^{\mathbf{R}_M} \rightarrow \mathbb{R}^+$
Output: A subset of \mathbf{R}_M

- 1 $\bar{\mathbf{R}} = \arg \max_{q \in \mathbf{R}_M} \{ER(\{q\}) : PC(\{q\}) \leq \mathbf{B}\};$
- 2 $\mathbf{R}_{out} = \emptyset;$
- 3 $\mathbf{R} = \mathbf{R}_M;$
- 4 **while** $\mathbf{R} \neq \emptyset$ **do**
- // Weight computation
- 5 **forall the** $q \in \mathbf{R}$ **do**
- 6 $w_q = \frac{ER(\mathbf{R}_{out} \cup \{q\}) - ER(\mathbf{R}_{out})}{PC(\{q\});}$
- 7 $q_{max} = \arg \max_{q \in \mathbf{R}} w_q;$
- 8 **if** $PC(\mathbf{R}_{out}) + PC(\{q_{max}\}) \leq \mathbf{B}$ **then**
- 9 $\mathbf{R}_{out} \leftarrow \mathbf{R}_{out} \cup \{q_{max}\};$
- 10 $\mathbf{R} = \mathbf{R} \setminus \{q_{max}\};$
- 11 **if** $ER(\mathbf{R}_{out}) > ER(\bar{\mathbf{R}})$ **then**
- 12 **return** \mathbf{R}_{out}
- 13 **else**
- 14 **return** $\bar{\mathbf{R}}$

The pseudo-code of RoMe is shown in Algorithm 1. It incrementally constructs a set of paths \mathbf{R}_{out} , which is initially empty (line 2). At each iteration, the algorithm picks an unconsidered path q , which has maximum weight w_q . The weight is defined as the increase in the function ER that q provides, divided by its cost (line 6). The path q is included in \mathbf{R}_{out} only if the budget is not exceeded (line 8). The algorithm returns the best solution between the set \mathbf{R}_{out} , and the best single path solution $\bar{\mathbf{R}}$, which is the path that maximizes the expected rank function and has a cost within the budget (lines 1, 11-14).

As indicated before, RoMe provides a solution which is within a provable approximation bound as stated by the following theorem. In the appendix we prove that ER satisfies the conditions of the theorem.

Theorem 6 ([11]). *Given a ground set E , a submodular function $f : 2^E \rightarrow \mathbb{R}_+$ and an budget \mathbf{B} . If f is non-decreasing and $f(\emptyset) = 0$ then the greedy algorithm produces a solution which is at least $1 - \frac{1}{\sqrt{e}}$ times the optimal value.*

We point out that an exact implementation of RoMe would have a very high computational complexity due to the inherent exponential complexity in computing ER . This makes the straightforward implementation highly unfeasible. In Section IV-C, we propose an efficient method to reduce the complexity. In the next section we show that in the common setting adopted by previous works, our algorithm provides an optimal solution.

B. Optimization Under Linear Independence Constraint

Previous works on network tomography consider paths with unit probing costs and only select a set of linearly independent paths for probing [1], [2], [3]. In this more constrained setting, RoMe solves the optimization problem optimally. This is proven by using the theory of *matroids*. A matroid is defined as follows:

Definition 7 (Matroid [12]). *A matroid \mathbb{M} is a pair (E, \mathcal{I}) , where E is a finite ground set and \mathcal{I} is a non-empty collection such that $\mathcal{I} \subseteq 2^E$, with the following properties:*

- $\forall A \subset B \subseteq E$, if $B \in \mathcal{I}$, then $A \in \mathcal{I}$.
- $\forall A, B \in \mathcal{I}$ with $|B| > |A|$, $\exists x \in B \setminus A$ s.t. $A \cup \{x\} \in \mathcal{I}$

We define $\mathbb{M}_L = (R_M, \mathcal{I}_L)$, where R_M is the set of all paths and \mathcal{I}_L contains the sets $R \subseteq R_M$ such that paths in R are linearly independent. Under unitary cost, the budget is equal to the maximum number of paths that can be probed, i.e. we require for each path set $R \in \mathcal{I}_L$ that $|R| \leq B$. For any value of B , it can be shown that \mathbb{M}_L is a matroid [12].

The following lemma shows that in the setting of matroid constraint, where selected paths are independent, the function ER is modular, i.e. $ER(R) = \sum_{q \in R} ER(\{q\})$. The complete proof is given in the appendix.

Lemma 8. *Given a set of candidate paths R_M and the corresponding matroid $\mathbb{M}_L = (R_M, \mathcal{I}_L)$, the function $ER : \mathcal{I}_L \rightarrow \mathbb{R}^+$ is modular.*

We slightly modify RoMe, to solve the problem under the linear independence constraint. At each iteration, the path q with maximum weight is added to the R_{out} only if it is linearly independent from the already selected paths (line 7). Since paths have unit cost, the while loop terminates if adding an additional path would violate the budget constraint.

Since the ER function is modular when paths are independent, the greedy approach achieves an optimal solution, as stated by the following theorem:

Theorem 9 ([12]). *Given a matroid $M = (E, \mathcal{I})$ and a monotone modular function $f : 2^{R_M} \rightarrow \mathbb{R}_+$, the greedy algorithm finds an optimal solution $R^* = \max_{R \in \mathcal{I}} f(R)$.*

C. Efficient Computation of ER

To analyze the complexity of RoMe it is important to analyze the complexity of evaluating the function ER , since the algorithm calls ER $O(|R_M|^2)$ times. This is required to determine the maximum increment in ER , i.e., the weight w_q for selecting a path $q \in R$ at each iteration. The worst-case complexity of the function ER for any arbitrary set $R \subseteq R_M$ is:

$$O(ER(R)) = 2^{|E|} O(|E| \times |R|^2),$$

where $|E|$ is the number of links and $2^{|E|}$ is the total number of failure scenarios. Hence, the complexity of a straightforward implementation of RoMe is $O(2^{|E|} \times |E| \times |R_M|^4)$. In this section, we reduce the complexity of the algorithm by proposing an approach that efficiently computes an approximated value of ER .

There are several ways to reduce the complexity of combinatorial functions such as ER by approximating the value of the function using only a subset of combinations. One of the popular approaches is the Monte Carlo method, in which several random failure scenarios are generated and the achieved rank is averaged to compute the approximated value. However, due to the large number of possible failure scenarios, a large number of Monte Carlo runs should be generated to achieve a reliable approximation for ER . Clearly, this approach is not scalable, because the required runs increase very quickly as the number of links increases. Therefore, we introduce a more efficient and scalable analytical approach to compute an approximation of ER .

Let us consider a set of paths $R \subseteq R_M$, which contains both linearly independent and dependent paths. We denote the rank of R under failures by Z_R ; note that Z_R is a random variable and $ER(R)$ is the expected value $\mathbb{E}(Z_R)$. To analyze $\mathbb{E}(Z_R)$, we partition R into R_{ind} and R_{dep} , where R_{ind} is a maximal set of linearly independent paths in R and $R_{dep} = R \setminus R_{ind}$. Note that there are multiple partitions possible and we pick an arbitrary one.

Let X_q be a random variable which is equal to 1 if q is available, or 0 otherwise. The expected value $\mathbb{E}(X_q)$ equals the expected availability $EA(q)$ given in Eq. 3. Each path $q \in R_{dep}$ can be expressed as a linear combination of some paths in R_{ind} . Let $R_q \subseteq R_{ind}$ be the set of paths in which q is linearly dependent on, i.e., the paths which have non-zero coefficients in the linear combination. Let us consider the set $\hat{R}_q = R_{ind} \cup \{q\}$ and the corresponding rank random variable $Z_{\hat{R}_q}$. We can express $\mathbb{E}(Z_{\hat{R}_q})$ as:

$$\mathbb{E}(Z_{\hat{R}_q}) = \sum_{q \in R_{ind}} \mathbb{E}(X_q) + \mathbb{E}(D_q), \quad (5)$$

where D_q is a random variable that takes value 1 if q is available and at least one path in R_q has failed, or 0 otherwise. In other words, q contributes to the rank only when all its links are available and at least one path on which it depends is not available. Let L_{R_q} be the set of links in the paths of R_q but not in q , the expected value of D_q can be calculated as:

$$\mathbb{E}(D_q) = \prod_{l_i \in q} (1 - p_i) \left(1 - \prod_{l_i \in L_{R_q}} (1 - p_i)\right), \quad (6)$$

where we recall that p_i is the failure probability of link

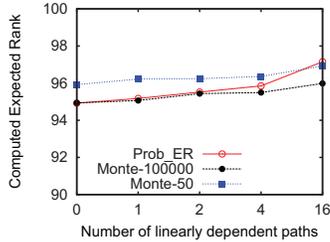


Fig. 4: Comparing ER computation for different approaches

l_i . The formula in Eq. 6 cannot be easily generalized by considering more redundant paths, because of the complex interdependencies in the paths. However, we can use it to upper bound the value of (Z_R) as:

$$(Z_R) \leq \sum_{q \in R_{ind}} (X_q) + \sum_{q \in R_{dep}} (D_q), \quad (7)$$

where (D_q) is calculated according to Eq. 6. The righthand side of Eq. 7 is an upper bound on (Z_R) because it assumes that each available path $q \in R_{dep}$ contributes to the rank as long as it cannot be represented as a linear combination of available paths in R_{ind} , ignoring the possibility that q could be represented by the available paths in $R_{ind} \cup (R_{dep} \setminus \{q\})$.

We evaluate the proposed bound in Eq. 7 against the true value of ER through a numerical example. We use the realistic topology AS1239 from the Rocketfuel Project [8] and 1600 candidate paths in R_M (detailed simulation settings are provided in Section VI-A). We pick an arbitrary basis from R_M , and then add additional paths (that are linearly dependent on the basis) as needed to evaluate the impact of the number of linearly dependent paths on the ER and its approximations. Since enumerating all the $2^{|E|}$ failure scenarios is computationally infeasible, in this example we approximate it by the Monte Carlo method using a large number (10^5) of Monte Carlo runs.

Figure 4 shows the values of the “true” ER (MC-100000) and our bound (ProbBound) as we vary the number of linearly dependent paths. Besides validating that Eq. 7 is indeed an upper bound, the plot also shows that it provides a close approximation to the true ER when the number of linearly dependent paths is small. In addition, we approximate ER using the Monte Carlo method with a small number (50) of runs (MC-50). We see that with fewer runs, the Monte Carlo method incurs a larger approximation error at a small number of linearly dependent paths.

We note that the proposed bound particularly suits the need of RoMe, because a mistake in selecting paths in initial iterations (when there are few linearly dependent paths in R_{out}) affects all subsequent iterations, and therefore it is more important to approximate ER closely for those cases. Indeed, our evaluations confirm the advantage of using the bound instead of the Monte Carlo method to approximate ER in RoMe (see Section VI-B).

V. OPTIMIZATION WITHOUT STATISTICAL KNOWLEDGE OF FAILURES

In this section, we consider the problem of maximizing the expected rank under a budget constraint when the failure distri-

bution is not known. Through probing we can only observe the availability of paths and not of the individual links. In general, we cannot determine the link failure vector from the observed path availabilities as the solution may not be unique, which makes it difficult to learn the original (link) failure distribution. Meanwhile, our observations readily allow the learning of path failure distributions. In previous sections, we have seen the significance of expected path availabilities in the computation of ER . This motivates us to learn expected path availabilities while trying to maximize ER using a reinforcement learning approach. We now explain in detail how to cast our problem into the framework of reinforcement learning.

We denote the random variable representing the availability of path $q_i \in R_M$ during the n -th epoch by $X_i(n) \in \{0, 1\}$, where $X_i(n) = 1$ if all links of path q_i are available during the epoch, or 0 otherwise. We refer to θ_i as the unknown mean of $X_i(n)$, which equals to the expected availability of path q_i as defined in Eq. 3. We denote the total number of candidate paths in R_M by N .

At each epoch n , we perform an *action* $R(n)$, selected in an action space \mathcal{A} , which represents the set of paths that are probed at this epoch. We assume that \mathcal{A} only contains *maximal path sets* under the given budget constraint \mathbf{B} , without loss of generality. We define L as the maximum number of paths probed in an epoch, i.e., $L = \max_{R \in \mathcal{A}} |R|$.

Our goal is to maximize the expected rank by learning expected path availabilities. Strictly speaking, expected path availabilities do not give sufficient information to calculate the value of ER , as path availabilities can be correlated due to shared links. To address this issue, we modify our objective function to an approximated ER , denoted by $ER(R; \theta)$, which is the expected rank of available paths in R under the assumption that path availabilities are independent. In contrast, we refer to $EA(R; \theta) = \sum_{q_i \in R} \theta_i$ as the expected availability of path set R .

When an action $R(n)$ is performed, the realization of the random variables corresponding to the paths in $R(n)$ are observed. We define the *reward* of an action $R(n)$ as the rank of paths in $R(n)$ that are available in this epoch:

$$\mathcal{R}_{R(n)}(n) = r(\{q_i \in R(n) : X_i(n) = 1\}). \quad (8)$$

By definition, $(\mathcal{R}_{R(n)}) = ER(R(n); \theta)$. Following the convention in reinforced learning, we evaluate the performance of our solution by *regret*, which is defined as the expected difference between the cumulative reward that could be obtained by selecting the optimal action at each time and the reward actually achieved. At a given epoch n , the regret is defined as:

$$\mathfrak{R}(n) = nER(R^*; \theta) - \sum_{i=1}^n ER(R(i); \theta), \quad (9)$$

where R^* denotes the optimal set of probing paths that maximizes $ER(R; \theta)$, if θ is known. Our goal is to determine a strategy of taking actions such that the regret is minimized over time.

The above reinforcement learning problem has been studied for the special case of linear (i.e. modular) reward functions in [13], where an algorithm called Learning with Linear Reward (LLR) is proposed with a regret bounded by $O(L^3 N \log n)$.

However, the submodularity of the ER function for a general set of paths violates the linear reward assumption of LLR. In this paper we extend the results in [13] and propose an algorithm, called Learning with Submodular Rewards (LSR), which provides formal bounds on the achieved regret for submodular reward functions under certain conditions.

A. Learning with Submodular Rewards (LSR)

The algorithm LSR keeps track of the empirical expected availability $\hat{\theta}_i$ for each path q_i , and a counter μ_i representing the number of times that q_i has been probed. We denote by $\hat{\theta} = (\hat{\theta}_i)_{i=1}^N$ and $\mu = (\mu_i)_{i=1}^N$ the vectors of empirical availabilities and counters for all paths, respectively.

The algorithm has two phases: *initialization* and *learning*. During the initialization phase, N actions are played such that each random variable is observed at least once. During the optimization phase of epoch n , the action $R(n)$ is picked such that:

$$R(n) = \arg \max_{R \in \mathcal{A}} ER(R; \hat{\theta} + \mathbf{C}) \quad (10)$$

where $\mathbf{C} = (\sqrt{(L+1) \log n / \mu_i})_{i=1}^N$ is the vector of confidence intervals of empirical path availabilities, as a function of time and number of observations. After probing paths in $R(n)$, we update $(\hat{\theta}_i, \mu_i)_{i=1}^N$. The pseudo-code of the LSR algorithm is shown in Algorithm 2.

Algorithm 2: LSR algorithm

```

// INITIALIZATION PHASE
1 n = 0;
2 for i = 1 to N do
3   Play an action R such that  $q_i \in R$ ;
4   Update  $\hat{\theta}$  and  $\mu$ ;
5   n++;
// OPTIMIZATION PHASE
6 while True do
7    $R(n) = \arg \max_{R \in \mathcal{A}} ER(R; \hat{\theta} + \mathbf{C})$ ;
8   Play  $R(n)$ ;
9   Update  $\hat{\theta}$  and  $\mu$ ;
10  n++;

```

Note that, the optimization problem in Eq. 10 is equivalent to the problem that we discussed in Section IV, and hence it is NP-Hard. We apply the greedy algorithm RoMe proposed in Section IV-A together with the bound of ER proposed in Section IV-C to solve the problem efficiently. Under the assumption of independent path availabilities, the bound in Eq. 7 for a set $R \subseteq R_M$ is reduced to:

$$ER(R; \theta) \leq \sum_{q_i \in R_{ind}} \theta_i + \sum_{q_i \in R_{dep}} \theta_i \left(1 - \prod_{q_j \in R_{q_i}} \theta_j\right) \quad (11)$$

B. Performance Analysis

If \mathcal{A} only contains sets of linearly independent paths, as discussed in the problem under matroid constraint in Section IV-B, LSR reduces to LLR proposed in [13], in which case $ER(R; \hat{\theta} + \mathbf{C}) = EA(R; \hat{\theta} + \mathbf{C})$ becomes a linear reward function (for any $R \in \mathcal{A}$). Consequently, the performance

analysis of LLR applies, yielding a regret upper bound of $O(L^3 N \log n)$.

In the general case of budget constraint \mathcal{A} contains sets of linearly dependent paths, which makes $ER(R; \hat{\theta} + \mathbf{C})$ a strictly submodular function. It is difficult to analyze the accuracy in approximating $ER(R; \theta)$, as the relationship between the expected rank and individual path availabilities is highly combinatorial. However, when the solution to the problem of maximizing $ER(R; \theta)$ is unique, assuming known θ , and it is a set of linearly independent paths, the problem becomes tractable as discussed in the following.

Let \mathcal{S} be the set of all suboptimal path sets, i.e., $\mathcal{S} = \{R \in \mathcal{A} : ER(R; \theta) < ER(R^*; \theta)\}$. We denote by $\Delta := \max_{R \in \mathcal{A}} (ER(R^*; \theta) - ER(R; \theta))$ the maximum gap between candidate path sets in terms of expected rank, and by $\delta := EA(R^*; \theta) - \max_{R \in \mathcal{S}} EA(R; \theta)$ the minimum gap between R^* and suboptimal path sets in terms of expected availability (note that δ can be negative). We have the following bound on the regret of LSR², a sketch proof of which is given in the appendix and the full proof is provided in [14].

Theorem 10. *If R^* is a linearly independent set, and $\delta > 0$, then the regret of LSR at epoch n is bounded by*

$$\mathfrak{R}(n) \leq \Delta N \left[\left(\frac{2L}{\delta} \right)^2 (L+1) \log n + 1 + \frac{\pi^4}{45} L \right] \quad (12)$$

$$= O\left(\frac{\Delta}{\delta^2} N L^3 \log n\right). \quad (13)$$

Clearly the bound in Theorem 10 only applies to a subset of cases. The following lemma gives a more explicit description of these cases in terms of the Knapsack solution, the proof is provided in the appendix.

Lemma 11. *A sufficient condition of the conditions in Theorem 10 is that the solution to the Knapsack Problem of maximizing $EA(R; \theta)$ under the same budget constraint (assuming known θ) is unique, and is a linearly independent set.*

C. Discussions

Any online algorithm, including LSR, that tries to maximize the expected rank with no prior knowledge of path availability distributions encounters three sources of suboptimality: (i) suboptimality due to unknown path availability distributions, (ii) suboptimality in the definition of the reward function, and (iii) suboptimality in selecting paths given estimated path availability distributions. Our regret analysis focuses on (i), but (ii) and (iii) also contribute to the overall gap between the learning algorithm and the optimal solution. We will evaluate the aggregate impact of (i-iii) via simulations in Section VI-B.

VI. EVALUATION

In our evaluation setup, we take the following inputs: a network topology, a routing protocol, a set of nodes which

²Although the proof assumes an exact solution to the optimization on line 7, analogous proof can show the same bound for a modified regret when line 7 can only be solved approximately as in our case, where the modified regret compares the actual reward to the reward of the approximate solution based on true θ (given by RoMe). Note that the use of the approximated ER in Eq. 11 does not affect the regret bound as pointed out in the proof.

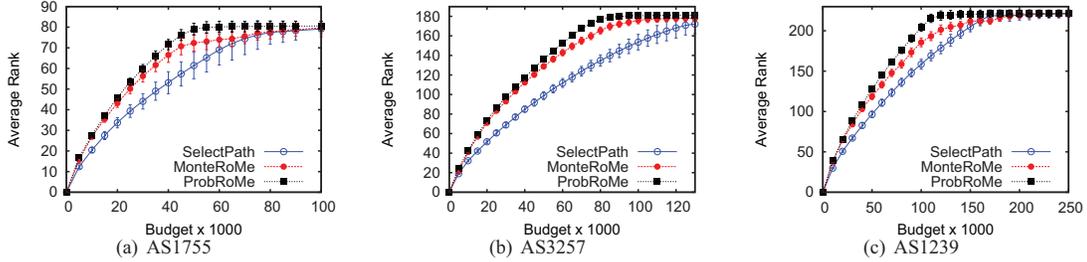


Fig. 5: Performance with varying budget

act as sources and destinations of the e2e paths, and models of probing cost and link failures. We separately evaluate our solutions against benchmarks for cases when the failure distribution is known and unknown. We first describe our evaluation setup in Section VI-A and then in Section VI-B we provide results that show the benefits of our proposed algorithms when compared to state-of-art approaches.

A. Evaluation Setup

We consider realistic ISP topologies from the Rocketfuel Project [8]. We select the topologies of three autonomous systems with labels AS1755, AS3257 and AS1239. The numbers of nodes and edges of each AS are presented in Table I; AS1755, AS3257 and AS1239 are representatives for small, medium, and large topologies, respectively. The nodes in the considered autonomous systems are backbone routers.

AS no. (type)	No. of Nodes	No. of Links
AS1755 (Small)	87	161
AS3257 (Medium)	161	328
AS1239 (Large)	315	972

TABLE I: Details of topologies

In our simulations, we randomly select a subset of nodes that can function as monitors which are defined in Section II-A. Each monitor acts as either a source or a destination of paths. We also perform simulations for the case in which a monitor can act as both a source and a destination. We do not show the results for this case due to space limitation, as we observe trends similar to the presented results.

As discussed in Section II-A, we assume a single path between each monitor pair. We use Dijkstra’s weighted shortest path algorithm as the routing algorithm to determine these paths using inferred weights of Rocketfuel topologies [8]. To determine the probing cost of paths, we use the cost model that is explained in Section III-B. The access cost (AC) for candidate monitors is picked randomly from two different classes: 0 and 300 with equal probability, which represent self-owned and peer-owned monitors, respectively. The run-time costs are linear functions of hop lengths in the corresponding paths with a weight of 100; the weight is chosen to make the run-time cost comparable to the access cost.

We adopt the link failure model that is proposed by Athina et al. [5] to determine the failure distribution. We only focus on independent link failures which are the most common type of failures in IP and wide area networks [15], [5]. The model determines link failure probabilities by first specifying the

number of failures per link and then normalizing it by the total number of failures. Ordering the links into decreasing order of link failure probabilities, the model classifies the first 2.5% of links as high failure links and the rest as low failure links. Let $n(l)$ denote the number of failures $n(l)$ for a link l (with the l -highest failure probability). According to the model, $n(l) \propto l^{-0.73}$ for high failure links and $n(l) \propto l^{-1.35}$ for low failure links. Moreover, $n(1) = 1000$ for the highest failure link.

In our simulations, we evaluate the robustness for a set of paths R = by considering the provided *rank* and *link identifiability* under failures. The link identifiability of a set R is defined as the number of links for which it is possible to determine a unique solution by solving the linear system corresponding to the paths in R , as discussed in Section II-A. We show average, standard deviation and cumulative distribution function (CDF) of these metrics, calculated by randomly selecting 5 sets of monitors and generating 500 failure scenarios for each set. The 500 failure scenarios are generated randomly using probabilities of link failures, and these may include the case with no link failures in the network.

B. Results

In the following, we refer to RoMe with the probabilistic bound of ER described in Section IV-C as *ProbRoMe*. We compare it to *MonteRoMe*, where RoMe approximates ER using the Monte Carlo method over 50 randomly generated failure scenarios. We compare our algorithms with an existing approach called SelectPath [3]. SelectPath selects an arbitrary maximal set of linearly independent paths (an arbitrary basis) using Cholesky decomposition.

Since there is no existing algorithm that solves the problem under budget constraint, we reasonably modify SelectPath for comparison purposes. This is done by greedily modifying the basis selected by the original SelectPath to fit the budget. In particular, if the cost of the basis is below the budget, we greedily add paths in increasing order of probing cost, until the budget allows. On the contrary, if the cost of the basis exceeds the budget, we greedily remove paths in decreasing order of cost, until the budget constraint is met.

We also present results for *MatRoMe* that solves the problem under unitary cost and linear independence constraint, as described in Section IV-B. We compare *MatRoMe* to the original SelectPath³. In this case, we set the budget \mathbf{B} as the

³ *MatRoMe* uses SVD decomposition, which is more accurate than Cholesky decomposition adopted by SelectPath for the rank computation.

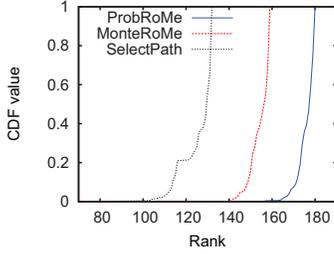


Fig. 6: CDF of rank

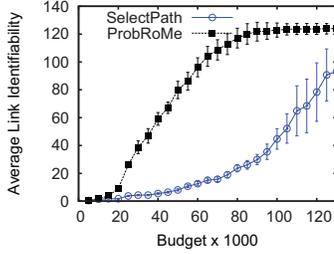


Fig. 7: Link identifiability with varying budget

rank of all candidate paths, i.e., number of paths in a basis.

1) *Optimization with Statistical Knowledge of Failures:* In this section, we show the results for all three topologies: AS1755, AS3257 and AS1239. We consider 400 (AS1755), 1600 (AS3257), and 2500 (AS1239) candidate paths.

In Figure 5, we plot the average rank of the selected paths under failures along with its standard deviation as we vary the budget. With increase in the budget, the rank increases for all algorithms because the number of selected paths increases. In all plots, ProbRoMe and MonteRoMe significantly outperform SelectPath, showing the importance of considering robustness when selecting probing paths. In particular, SelectPath requires twice as much budget in order to maximize the rank compared to ProbeRome. The performance of ProbRoMe is better than MonteRoMe, because the computation of ER in ProbRoMe is more accurate when the number of linearly dependent paths is small, as illustrated in Section IV-C. However, the true advantage of ProbRoMe is that it has a much lower runtime complexity than MonteRoMe. In our simulations, the run-time of MonteRome is around five times longer than ProbeRome on average in various cases. Moreover, in the case of ProbRoMe, the standard deviation in the rank is much lower than MonteRoMe and SelectPath in all the plots in Figure 5. This further confirms the performance advantage of the ProbRoMe algorithm.

To provide a more detailed comparison, in Figure 6, we show the CDF of the rank for the case, where there are 1600 candidate paths in AS3257 and a budget of 80,000. We see that ProbRoMe provides a higher rank than other approaches in all scenarios, showing uniformly good performance. This is also observed in other cases.

Next, we evaluate the link identifiability. In Figure 7, we plot the average link identifiability along with its standard deviation for AS3257 with 1600 candidate paths. We compare SelectPath with the best of our proposed approaches, ProbeRome, as seen in Figure 5. Similar to rank, the link

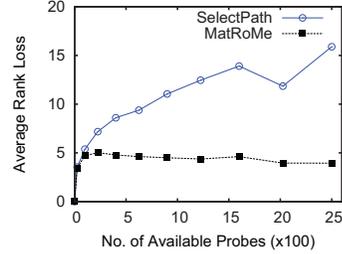


Fig. 8: Optimization under linear independence

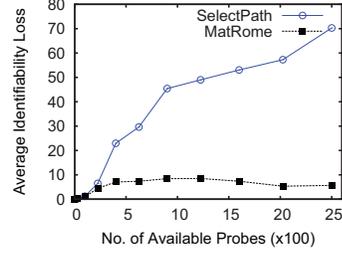


Fig. 9: Optimization under linear independence

identifiability increases with the increase in budget for both algorithms. Again, ProbRoMe performs better than SelectPath, both in average and in standard deviation. Compared with rank, we observe that ProbRoMe provides much larger improvement over SelectPath for link identifiability. This is because even a small loss in rank can lead to a big loss in link identifiability. We observe similar trends for other topologies, results are omitted due to space limitation.

We now show the results for a more constrained case of linear independence and unitary cost of paths, described in Section IV-B. We compare our solution, MatRoMe, to the original SelectPath algorithm. Since the budget is fixed for a given set of candidate paths (to its rank), we vary the number of candidate paths and consider slightly different performance metrics. We evaluate the loss in rank for each simulated failure scenario when compared to the case of no failure; this performance metric is called *rank loss*. Similarly, we evaluate *link identifiability loss*.

In Figures 8 and 9, we plot rank loss and link identifiability loss for AS1239 topology with respect to the number of candidate probing paths. It is evident from both plots that MatRoMe performs better than SelectPath, again illustrating the importance of considering path robustness when selecting probing paths. As we increase the number of candidate paths, rank loss and link identifiability loss for MatRoMe do not vary much. In contrast, both performance metrics increase significantly for SelectPath, because of the increase in the total number of bases, since it picks an arbitrary basis without taking account of robustness of paths. Together, these results suggest that the advantage of MatRoMe over SelectPath is more prominent for large networks with many candidate paths.

2) *Optimization Without Statistical Knowledge of Failures:* We present the results for the LSR algorithm by varying the available budget. We consider the final set of paths selected by LSR, according to the learned path availabilities, after 500 and 1000 epochs. We compare LSR to ProbRoMe, for which we assume that the failure distribution is known, and to SelectPath.

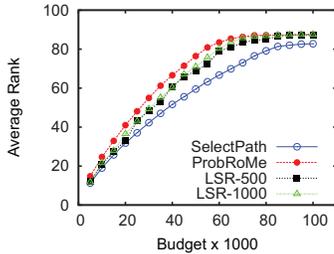


Fig. 10: Performance of reinforced learning approach

We consider the topology AS3257 and 400 candidate paths.

Figure 10 shows the average rank achieved by the algorithms. LSR has performance close to ProbRoMe, showing that LSR is effective in learning path availabilities and hence maximizing the rank under failures, even when the failure distribution is initially not known. As expected, LSR has better performance as the number of epochs increases, because of the learned knowledge becomes more accurate knowledge. SelectPath has worse performance than LSR in both cases, highlighting the inefficacy of this approach in the presence of failures.

VII. RELATED WORK

There is no prior work that considers the specific problem of optimizing the robustness of e2e measurements under link failures in the context of network tomography.

Zheng et al. [1] consider the problem of selecting the minimum number of paths that can be used to uniquely determine the solution for a set of target links. Chen et al. [3] propose a method to determine the complete set of e2e measurements from an incomplete set, resulting in a scalable monitoring system in large networks. These works minimize the number of paths to be probed by using rank revealing decomposition methods to determine a basis. However, they are agnostic to the effects of network failures on the performance.

Current state of the art in network tomography [16], [17], [18] consider the problem of minimizing the number of monitors required to acquire the e2e measurements. These approaches focus on different optimization objectives with respect to our paper and do not consider failures in the network. Furthermore, controlled routing is often assumed [17], [18], such as source routing, which is impractical in common networks.

The use of network tomography techniques in the context of network failures has been considered by Nguyen et al. [19], [4]. Differently from our approach, these papers exploit network tomography to identifying links that experience poor performance, instead of optimizing the performance under failures.

VIII. CONCLUSION

In this paper we study, for the first time, the problem of selecting a robust set of candidate paths to improve the performance of network tomography applications under failures.

We define a function called expected rank to capture the robustness of a set of paths and formulate an optimization

problem that aims at maximizing the expected rank under a budget constraint on probing cost. We show that the problem is NP-Hard. We proved that our objective function is submodular, and we define an approximate algorithm called RoMe. We also propose an upper bound to the expected rank function which can be efficiently calculated. Furthermore, we show that in the more constrained case in which selected paths are linearly independent and paths have unit cost, RoMe provides an optimal solution.

We tackle the issue of unknown failure distribution from the perspective of reinforcement learning. We propose the LSR algorithm which learns path availabilities through probing and maximizes the expected rank over different epochs.

We evaluate our solutions through simulations based on real network topologies and realistic failure models. Results show that our solutions can significantly improve the performance of network tomography over existing techniques in the presence of failures.

Acknowledgements

We acknowledge the initial inputs of Don Towsley and Ananthram Swami. This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

- [1] Q. Zheng and G. Cao, "Minimizing probing cost and achieving identifiability in probe-based network link monitoring," *IEEE Transactions on Computers*, vol. 62, no. 3, pp. 510–523, 2011.
- [2] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, "Maximum likelihood network topology identification from edge-based unicast measurements," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 11–20, 2002.
- [3] Y. Chen, D. Bindel, H. Song, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," *ACM SIGCOMM Comp. Com. Rev.*, vol. 34, no. 4, pp. 55–66, 2004.
- [4] H. X. Nguyen and P. Thiran, "Active measurement for multiple link failures diagnosis in ip networks," *Springer Passive and Active Measurement Workshop (PAM)*, pp. 185–194, 2004.
- [5] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," *IEEE INFOCOM*, vol. 4, pp. 2307–2317, 2004.
- [6] H. X. Nguyen and P. Thiran, "Network loss inference with second order statistics of end-to-end flows," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07, 2007, pp. 227–240.
- [7] J. He and J. Rexford, "Toward internet-wide multipath routing," *IEEE Network*, vol. 22, no. 2, pp. 16–21, 2008.
- [8] "Rocketfuel Project: Internet Topologies," <http://www.cs.washington.edu/research/networking/rocketfuel/>.
- [9] H. Yao, S. Jaggi, and M. Chen, "Passive network tomography for erroneous networks: A network coding approach," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5922–5940, 2012.
- [10] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, 2012.
- [11] A. Krause and C. Guestrin, "A note on the budgeted maximization of submodular functions," *Tech. Report at CMU*, 2005.

- [12] J. Lee, *A first course in combinatorial optimization*. Cambridge University Press, 2004, vol. 36.
- [13] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1466–1478, 2012.
- [14] S. Tati, S. Silvestri, T. He, and T. La Porta, "Robust network tomography in the presence of failures," *Pennsylvania State University Technical Report*, 2014. [Online]. Available: http://www.cse.psu.edu/~simone/papers/icdcs_tech_rep.pdf
- [15] G. Iannaccone, C.-n. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an ip backbone," *ACM SIGCOMM*, pp. 237–242, 2002.
- [16] R. Kumar and J. Kaur, "Practical beacon placement for link monitoring using network tomography," *IEEE Selected Areas in Communications*, vol. 24, no. 12, pp. 2196–2209, 2006.
- [17] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 3, pp. 906–916, 2012.
- [18] L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami, "Efficient identification of additive link metrics via network tomography," *IEEE ICDCS*, 2013.
- [19] H. X. Nguyen and P. Thiran, "The boolean solution to the congested ip link location problem: Theory and practice," in *IEEE INFOCOM 2007*, 2007, pp. 2117–2125.

APPENDIX

Theorem. *The budget-constraint optimization problem is NP-Hard.*

Proof: In order to prove the NP-Hardness, we provide a reduction from the knapsack problem. Let us consider a general knapsack instance: a capacity \mathbf{B} and a set of items $\Omega = \{s_1, \dots, s_n\}$, where each item $s_i \in \Omega$ has a value v_i and a weight w_i . The goal is to find a set $S^* \subseteq \Omega$ whose items provide maximum value and do not exceed the capacity of the knapsack.

We translate the general knapsack instance into an instance of our problem by defining a set of candidate paths R_M in which we construct a path q_i , of length one, for each item $s_i \in \Omega$. Different paths are disjoint. We set $PC(q_i) = w_i$ and the failure probability of the single link in q_i to $p_i = 1 - \frac{v_i}{TC}$, where $TC = \sum_{s_i \in \Omega} v_i$. Since q_i has a single link, $ER(\{q_i\}) = 1 - p_i$. Finally, we consider the budget \mathbf{B} as the capacity of the knapsack instance.

Our problem aims at finding a set R^* whose paths provide maximum expected rank and incur a cost no more than \mathbf{B} . Since paths in R_M are linearly independent, for each $R \subseteq R_M$, $ER(R) = \sum_{q_i \in R} ER(\{q_i\})$, which is equivalent to the objective function of the knapsack problem.

A solution to our problem can be translated into a solution of the knapsack instance by selecting the items corresponding to the paths in R^* . As a result, solving our problem is at least as hard as solving knapsack and thus our problem is NP-Hard. ■

Theorem. *Given a set of paths R_M , the function $ER : 2^{R_M} \rightarrow \mathbb{R}^+$ is submodular.*

Proof: To prove the submodularity of ER , we need show that, given two sets A, B of paths, such that $A \subseteq B \subseteq R_M$, and a path $q \in R_M$, we have:

$$ER(A + q) - ER(A) \geq ER(B + q) - ER(B) \quad (14)$$

where we use the notation $A + q$ for $A \cup \{q\}$ and similarly for $B + q$. For a set $A \subseteq R_M$, the ER function can be rewritten as:

$$ER(A) = \sum_{i=1}^n r(A^{v_i}) \alpha_i$$

where $n = 2^{|E|}$, $\alpha_i = \mathbb{P}(v_i)$ and A^{v_i} is the set of available paths in A under the failure scenario v_i . Let us order the failure scenarios such that under the failure scenario v_i the path q is available if $i \in [1, n_1]$ and it is not available if $i \in [n_1 + 1, n]$. We can write the inequality given in Eq. 14 as:

$$\sum_{i=1}^{n_1} r(A^{v_i} + q) \alpha_i + \sum_{i=n_1+1}^n r(A^{v_i}) \alpha_i - \sum_{i=1}^n r(R_A^{v_i}) \alpha_i \geq \sum_{i=1}^{n_1} r(B^{v_i} + q) \alpha_i + \sum_{i=n_1+1}^n r(B^{v_i}) \alpha_i - \sum_{i=1}^n r(R_B^{v_i}) \alpha_i$$

The above formula can be rewritten as:

$$\sum_{i=1}^{n_1} \alpha_i (r(A^{v_i} + q) - r(A^{v_i})) \geq \sum_{i=1}^{n_1} \alpha_i (r(B^{v_i} + q) - r(B^{v_i}))$$

Since $A \subseteq B$ and the rank function is submodular, for each $i \in [1, n_1]$ we have that $r(A^{v_i} + q) - r(A^{v_i}) \geq r(B^{v_i} + q) - r(B^{v_i})$. As a result, the above inequality holds and the function ER is submodular. ■

Lemma. *Given a set of paths R_M , the function $ER : 2^{R_M} \rightarrow \mathbb{R}^+$ is non-decreasing and $ER(\emptyset) = 0$.*

Proof: Let us consider a generic set $A \subseteq R_M$ and a path $q \in R_M$. According to the definition of the ER function, for each failure scenario $v \in \{0, 1\}^{|E|}$, $r(R_{A+q}^v) \geq r(R_A^v)$ because the element q may increase the rank provided by the probed paths. As a result, $ER(A + q) \geq ER(A)$ and thus ER is non-decreasing.

To complete the proof of the Theorem, we need to prove that $ER(\emptyset) = 0$. This follows trivially by the definition of the rank function, as probing no path provides no rank. ■

Lemma. *Given a set of paths R_M and the corresponding matroid $\mathbb{M}_L = (R_M, \mathcal{I}_L)$, the function $ER : \mathcal{I}_L \rightarrow \mathbb{R}^+$ is modular.*

Proof: In order to prove the Lemma, we prove that for any given set of paths $R \in \mathcal{I}_L$, the function ER can be rewritten as the sum of positive weights given by the expected availabilities of the paths in R .

Since $R \in \mathcal{I}_L$, it contains only independent paths. Hence, for any failure scenario v , $R_v \subseteq R$ is also in \mathcal{I}_L . As a result, under a failure scenario v , the rank provided by the set R_v is $|R_v|$. We can rewrite $ER(R)$ as follows:

$$\begin{aligned} ER(R) &= \sum_{v \in \{0,1\}^{|E|}} r(R_v) \mathbb{P}(v) \\ &= \sum_{v \in \{0,1\}^{|E|}} |R_v| \mathbb{P}(v) = \\ &= \sum_{v \in \{0,1\}^{|E|}} \sum_{q \in R} \mathbb{1}_{q \in R_v} \mathbb{P}(v) = \\ &= \sum_{q \in R} \sum_{v \in \{0,1\}^{|E|}} \mathbb{1}_{q \in R_v} \mathbb{P}(v) = \sum_{q \in R} EA(q) \end{aligned}$$

Since in the case of independent paths the ER function is the sum of positive weights, the function is modular. ■

Theorem. *If R^* is a linearly independent set, and $\delta > 0$, then the regret of LSR at slot n is bounded by*

$$\mathcal{R}(n) \leq \Delta N \left[\left(\frac{2L}{\delta} \right)^2 (L+1) \log n + 1 + \frac{\pi^4}{45} L \right] \quad (15)$$

$$= O\left(\frac{\Delta}{\delta^2} N L^3 \log n\right). \quad (16)$$

Proof sketch: The essence of the proof is to convert the problem with submodular reward to one with linear reward, which is then completed by arguments analogous to those in the proof LLR [13].

We introduce a few notions that will be used in the proof. For each slot t , let $\mu_i(t)$ denote the number of times probing (and hence observing availabilities of) path $q_i \in R_M$ ($i = 1, \dots, N$), $\hat{\theta}_{\mu(t)} := (\hat{\theta}_{i, \mu_i(t)})_{i=1}^N$ the empirical expected path availabilities under sample size $\mu(t)$, and $\mathbf{C}_{\mu(t)} := (C_{t, \mu_i(t)})_{i=1}^N$ ($C_{t, \mu_i(t)} :=$

$\sqrt{(L+1)\log t/\mu_i(t)}$ the confidence intervals. Define a counter $T_i(n)$ as follows: whenever a suboptimal path set is selected at time n , we increment $T_i(n)$ by 1 for $i = \arg \min_{q_j \in R(n)} \mu_j(n)$ (ties broken arbitrarily). It is known from [13] (the corresponding notation is $\hat{T}_i(n)$) that $T_i(n)$ has two properties: (i) $\sum_{i=1}^N T_i(n)$ equals the total number of times selecting suboptimal path sets among the first n slots, and (ii) $T_i(n) \leq \mu_j(n)$ for all $q_j \in R(n+1)$.

We first bound $T_i(n)$ by assuming it is always incremented in the first l slots for a positive integer l :

$$\begin{aligned} T_i(n) &\leq l + \sum_{t=N}^n \mathbb{I} \left\{ ER(R^*; \hat{\theta}_{\mu(t)} + \mathbf{C}_{\mu(t)}) \right. \\ &\quad \left. \leq ER(R(t+1); \hat{\theta}_{\mu(t)} + \mathbf{C}_{\mu(t)}); T_i(t) \geq l \right\}, \end{aligned} \quad (17)$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Because R^* is linearly independent, and expected rank is upper bounded by expected availability, we further have

$$\begin{aligned} &\mathbb{I} \left\{ ER(R^*; \hat{\theta}_{\mu(t)} + \mathbf{C}_{\mu(t)}) \leq ER(R(t+1); \hat{\theta}_{\mu(t)} + \mathbf{C}_{\mu(t)}) \right\} \\ &\leq \mathbb{I} \left\{ \sum_{q_i \in R^*} (\hat{\theta}_{i, \mu_i(t)} + C_{t, \mu_i(t)}) \leq \sum_{q_j \in R(t+1)} (\hat{\theta}_{j, \mu_j(t)} + C_{t, \mu_j(t)}) \right\}, \end{aligned} \quad (18)$$

which converts an indicator on the submodular reward function $ER(\cdot)$ to an indicator on a linear function. Note that the above bound still holds if we replace $ER(R; \theta)$ by its upper bound in Eq. 11 in the algorithm. Therefore, our regret bound applies to both the original LSR and a modified LSR based on the approximated $ER(R; \theta)$ function (approximated by its upper bound in Eq. 11).

From now on, we can apply techniques similar to the proof of LLR to bound the righthand-side (RHS) of (18). Specifically, probability of the RHS is upper bounded by sum of the probabilities of the following events:

- 1) $\sum_{q_i \in R^*} \hat{\theta}_{i, \mu_i(t)} \leq \sum_{q_i \in R^*} \theta_i - \sum_{q_i \in R^*} C_{t, \mu_i(t)}$;
- 2) $\sum_{q_j \in R(t+1)} \hat{\theta}_{j, \mu_j(t)} \geq \sum_{q_j \in R(t+1)} \theta_j + \sum_{q_j \in R(t+1)} C_{t, \mu_j(t)}$;
- 3) $\sum_{q_i \in R^*} \theta_i < \sum_{q_j \in R(t+1)} \theta_j + 2 \sum_{q_j \in R(t+1)} C_{t, \mu_j(t)}$.

Under the conditions of $T_i(t) \geq l$ and $t \leq n$, we can bound the probabilities of events (1) and (2) by $Lt^{-2(L+1)}$ using the *Chernoff-Hoeffding Bound*. We can make the probability of event (3) zero by setting $l = \lceil (2L/\delta)^2 (L+1) \log n \rceil$. See the full proof in [14] for details. Applying these bounds to (17) yields that

$$\mathbb{E}[T_i(n)] \leq \left(\frac{2L}{\delta} \right)^2 (L+1) \log n + 1 + \frac{\pi^4 L}{45}. \quad (19)$$

Since the overall regret is bounded by $\Delta \sum_{i=1}^N \mathbb{E}[T_i(n)]$, we have

$$\mathcal{R}(n) \leq \Delta N \left[\left(\frac{2L}{\delta} \right)^2 (L+1) \log n + 1 + \frac{\pi^4 L}{45} \right]. \quad (20)$$

■

Lemma. *A sufficient condition of the conditions in Theorem 10 is that the solution to the Knapsack Problem of maximizing $EA(R; \theta)$ under the same budget constraint (assuming known θ) is unique, and is a linearly independent set.*

Proof: Let R' be the optimal Knapsack solution. Since the expected availability upper bounds the expected rank, we see that any other path set $R \in \mathcal{A}$ ($R \neq R'$) satisfies

$$ER(R; \theta) \leq EA(R; \theta) < EA(R'; \theta) = ER(R'; \theta), \quad (21)$$

where the last equality is because R' is an independent set. Thus, $R' = R^*$ is also the optimal solution to maximizing the expected

rank. Since R' is the unique maximum of expected availability and the total number of path sets is finite, its gap to suboptimal path sets δ must be positive. ■