Stealthy DGoS Attack: DeGrading of Service under the Watch of Network Tomography

Cho-Chun Chiu* and Ting He*
*Pennsylvania State University, University Park, PA, USA. Email: {cuc496,tzh58}@psu.edu

***Abstract*—Network tomography is a powerful tool to monitor the internal state of a closed network that cannot be measured directly, with broad applications in the Internet, overlay networks, and all-optical networks. However, existing network tomography solutions all assume that the measurements are trust-worthy, leaving open how effective they are in an adversarial environment with possibly manipulated measurements. To understand the fundamental limit of network tomography in such a setting, we formulate and analyze a novel type of attack that aims at maximally degrading the performance of targeted paths without being localized by network tomography. By analyzing properties of the optimal attack, we formulate novel combinatorial optimizations to design the optimal attack strategy, which are then linked to well-known problems and approximation algorithms. Our evaluations on real topologies demonstrate the large damage of such attacks, signaling the need of new defenses.**

***Index Terms*—Network tomography, Denial of Service attack, combinatorial optimization, approximation algorithm.**

I. Introduction

Timely and accurate knowledge of network internal state (e.g., link delays/jitters/loss rates/bandwidths) is essential for many network management functions such as traffic engineering, load balancing, and service placement, which actively adapt control parameters such as the routes, the rates, and even the destinations (e.g., via service placement) according to the current network state.

Traditionally, network administrators obtain the network state by directly measuring internal network elements through local support (e.g., SNMP agents) or special diagnostic tools (e.g., traceroute). This approach has the limitation that it requires the support of internal network devices, e.g., to run SNMP agent or respond to ICMP probes, which has severe limitations in networks where such support is unreliable [1], [2], [3] or unavailable [4], [5].

Network tomography [6] provides a powerful approach for monitoring the internal state of closed networks. Instead of directly measuring the internal elements, network tomography infers the states of these elements (e.g., link delays) from end-to-end measurements (e.g., path delays) between special nodes participating in monitoring, referred to as *monitors*. As network tomography only requires the cooperation from monitors, it has broad applications in monitoring networks where only a subset of nodes cooperate, e.g. the Internet [1], [2], [3], overlay networks [7], and all-optical networks [4], [5].

Despite substantial research on network tomography, most existing solutions hinge on a fundamental assumption: *the measurements correctly reflect the performance of measurement paths*. Consider the canonical application of inferring additive link metrics (e.g., delays, jitters, log-success rates) from the sum metrics on measurement paths. As illustrated in Fig. 1 (a), normally the measured path metrics will equal the sum of link metrics on each path, yielding a linear observation model: $R\mathbf{x} = \mathbf{y}$, where $\mathbf{x} = (x_j)_{l_j \in L}$ is the column vector of unknown link metrics (L: set of links), $\mathbf{y} = (y_i)_{p_i \in P}$ is the column vector of measured path metrics (P: set of measurement paths), and $R = (r_{ij})_{p_i \in P, l_j \in L}$ is the *measurement matrix* with $r_{ij} \in \{0, 1\}$ indicating whether path p_i traverses link l_j. Network tomography infers the link metrics by "inverting" the observation model, i.e., solving for $\hat{\mathbf{x}}$ that satisfies $R\hat{\mathbf{x}} = \mathbf{y}$ (the solution may not be unique).

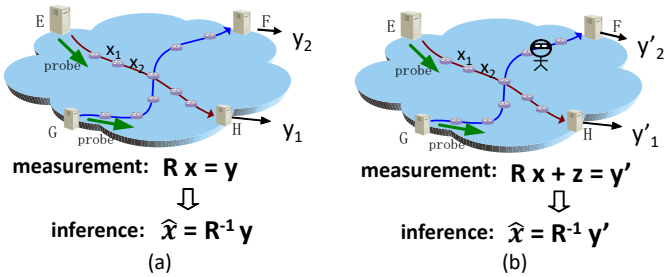

Fig. 1. (a) network tomography in benign setting; (b) network tomography in adversarial setting.

However, if some links are controlled by an attacker (referred to as *compromised links*) as illustrated in Fig. 1 (b), then the attacker can manipulate the measurements on paths traversing these links, e.g., by introducing additional delays, jitters, or losses. This yields a modified observation model: $R\mathbf{x} + \mathbf{z} = \mathbf{y}'$, where $\mathbf{y}'$ is the vector of observed path metrics under the attack, and $\mathbf{z} = (z_i)_{p_i \in P}$ is the vector of *manipulations* controlled by the attacker. For example, the attacker can be a malicious or hacked Internet Service Provider (ISP) that tries to launch a targeted attack on a certain content provider, whose paths to clients are modeled by P, from a set of links it controls in the public Internet. Note that this model is different from $R(\mathbf{x}+\mathbf{z}) = \mathbf{y}'$, as the attacker can manipulate packets on different paths differently at the same link, e.g., delaying packets belonging to one path but not delaying packets belonging to another path. An unsuspecting network tomography algorithm will try to explain the measurements according to the original observation model by trying to find $\hat{\mathbf{x}}$ satisfying $R\hat{\mathbf{x}} = \mathbf{y}'$. This can cause many issues, such as lack of feasible solutions [8] and incorrect fault diagnosis [9].

In this work, we aim to understand the fundamental limit of a stealthy attacker in *maximally degrading the performance*

Research was sponsored by NSF under Award Number 1813219.

of end-to-end communications without being localized by network tomography. Such understanding will not only quantify the limitation of existing network tomography algorithms but also provide insights for the design of new algorithms that are suitable for adversarial environments.

A. Related Work

Since introduced by Vardi [10], network tomography has expanded to a rich family of network monitoring techniques that infer network internal characteristics from external measurements [6], [11]. Early works focused on *best-effort solutions*, which tried to find the most likely network state from given measurements, obtained by unicast [12], [13], [14], [15], multicast [16], [17], [18], [19], [20], [21], and their variations (e.g., bicast [22], flexicast [23], and back-to-back unicast [24], [25], [21]). After observing that an arbitrary set of measurements is frequently insufficient for identifying all the link metrics [26], [13], [27], [7], [28], later works aimed at either reducing ambiguity by imposing a tie breaker (e.g., [14], [15], [29]) or relaxing the objective (e.g., [7], [30], [31]), or ensuring identifiability by carefully designing the monitor locations and the paths to measure [32], [33], [34], [35], [36], [5], [37], [38], [39], [40]. All these works assume a *benign setting*, where the links behave consistently.

In contrast, very few works have considered network tomography in an *adversarial setting*, where links can behave inconsistently for different paths. In [8], the problem is tackled in the context of a non-neutral network, where some links can discriminate packets sent on different paths. In [9], the problem is tackled in the context of an attacker that can manipulate the measurements traversing malicious nodes, with a primary goal of scapegoating certain benign links as the cause of poor performance. While our problem setting is similar to [9], our results differ significantly as explained in Section II-C.

B. Summary of Contributions

The main contributions of this work are:

1) We formulate a novel attack, called *stealthy DeGrading of Service (DGoS) attack*, that aims at maximally degrading the performances of end-to-end communications by manipulating the performances of compromised links, without letting these links localized by network tomography.

2) To understand the fundamental limit of this attack, we develop algorithms to explicitly design which links to compromise and how to manipulate the performances of these links. We show that selecting which links to compromise is a novel combinatorial optimization problem that is NP-hard. By linking this problem to several well-known problems, we leverage existing algorithms to achieve guaranteed approximation.

3) We further consider a constrained number of compromised links. We show that the constrained link selection problem is another novel combinatorial optimization problem which reduces to the previous problem as the constraint relaxes.

4) Our evaluations on real topologies show that the proposed attack can significantly degrade communication performance (injecting 4–30 seconds of delay per path) without exposing the compromised links to network tomography.

Roadmap. Section II formulates our problem. Section III designs the attack in the unconstrained case, which is evaluated in Section IV. Section V addresses the constrained case. Finally, Section VI concludes the paper.

II. Problem Formulation

A. Network Model

We model the network monitored by network tomography as an undirected graph $\mathcal{G} = (N, L)$, where N is the set of nodes and L the set of links. Each link $l_j \in L$ is associated with an unknown metric x_j that describes its performance (e.g., average link delay). We assume that these link metrics are *additive*, i.e., the metric of a path equals the sum of its link metrics, which is a canonical model representing important performance metrics including delays, jitters, log-success rates, and many other statistics.

B. Network Tomography Model

Suppose that a set of users of the above network (or their administrator) send traffic through $\mathcal{G}$ along a set of paths P, and use network tomography to monitor the received performances at individual links. For example, a content provider delivering content to its customers through the public Internet can use network tomography to monitor the received performances at links in different ISPs to detect violations of network neutrality [8]. Let $R = (r_{ij})_{p_i \in P, l_j \in L}$ be the matrix representation of P, called the *measurement matrix*, where $r_{ij} \in \{0,\ 1\}$ indicates if path p_i traverses link l_j. Let $\mathbf{r}_i = (r_{ij})_{l_j \in L}$ be the i-th row in R. Given the measured path metrics $\mathbf{y} = (y_i)_{p_i \in P}$, network tomography seeks to find a solution $\widehat{\mathbf{x}}$ to the link metrics that can explain the measurements, i.e., $R\widehat{\mathbf{x}} = \mathbf{y}$.

We note that the solution is generally non-unique as R may not be full-column-rank. This issue, known as the *lack of identifiability*, has been widely recognized [26], [13], [27], [7], [28]. Instead of making a limiting assumption that R must be full-column-rank as in [9], we allow an arbitrary R, and consider a generic network tomography solver that can compute the set of all feasible solutions.

C. Attack Model

Suppose that an attacker attempts to degrade the performance of P by manipulating the performances of certain compromised links. Let $L_m \subseteq L$ denote the set of *compromised links* and $L_n = L \setminus L_m$ the set of *uncompromised links*. Accordingly, the paths $P_m \subseteq P$ traversing at least one compromised link are called *compromised paths*, and the remaining paths $P_n = P \setminus P_m$ are called *uncompromised paths*. The attacker can only control the performance at compromised links.

Let $\mathbf{z} = (z_i)_{p_i \in P}$ denote the vector of *manipulations*, where z_i is the increment in the metric of path $p_i \in P$ caused by the attacker. It is easy to see that $\mathbf{z}$ must satisfy the following constraints [9]:

1) Only the metrics of compromised paths can be manipulated, i.e., $z_i = 0$ for any $p_i \in P_n$.
2) Path performances can only be degraded (not improved) due to manipulation, i.e., $z_i \geq 0$ for any $p_i \in P_m$.

Moreover, to stay stealthy, the attacker must preserve feasibility of the network tomography problem to hide the presence of artificial manipulations, i.e., after the manipulations, there must exist at least one solution $\widehat{\mathbf{x}}$ that satisfies $R\widehat{\mathbf{x}} = R\mathbf{x} + \mathbf{z}$. In addition, he must protect the compromised links from detection, i.e., among all the feasible solutions to $\widehat{\mathbf{x}}$, there must be at least one solution that does not flag any of the compromised links as bad links. We define "bad links" as those whose metrics are above a given threshold τ. In practice, there may also be an upper bound on link metrics, denoted by $\tau_{\max}$ (which can be ∞), e.g., the maximum delay implied by the limited buffer size at a network interface. To avoid trivial cases, we assume that $x_j \leq \tau \leq \tau_{\max}$ for all $l_j \in L$.

We formulate the attacker's goal as the following optimization, called the *stealthy DeGrading of Service (DGoS) attack*:

$$\max \sum_{p_i \in P_m} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{1a}$$

$$\text{s.t. } \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) = 0, \quad \forall p_i \in P_n, \tag{1b}$$

$$\mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \geq 0, \quad \forall p_i \in P_m, \tag{1c}$$

$$\tau_{\max} \geq \widehat{x}_j \geq 0, \quad \forall l_j \in L_n, \tag{1d}$$

$$\tau \geq \widehat{x}_j \geq 0, \quad \forall l_j \in L_m, \tag{1e}$$

$$L_m \subseteq L. \tag{1f}$$

This is an optimization of L_m and $\widehat{\mathbf{x}}$, where L_m specifies the links to compromise, and $\widehat{\mathbf{x}}$, denoting (one of the feasible solutions to) the inferred link metrics, is used to compute the *actual manipulations* $\mathbf{z}$ to inject onto the paths by

$$\mathbf{z} = R(\widehat{\mathbf{x}} - \mathbf{x}). \tag{2}$$

Computing the manipulations by (2) automatically ensures feasibility of the network tomography problem. Note that this does not require the compromised links to behave consistently across paths, as illustrated in Fig. 2. The objective (1a) is to maximize the total performance degradation on paths in P, measured by the increase in the sum path metric. Constraints (1b,1c) ensure that manipulations are feasible, i.e., only performed on compromised paths to degrade the performance. Constraint (1e) ensures that the attack cannot be localized by network tomography, as all the compromised links perform normally according to the inferred link metrics.

Remark: A similar problem of *scapegoating attack* has been studied in [9], which tries to fool network tomography while degrading path performances. Despite the similarity, [9] substantially differs from our work in that: (i) the scapegoating attack is required to mislead network tomography to detect certain uncompromised links as bad links, while we do not impose such constraints; (ii) [9] assumes the measurement matrix to be full-column-rank, which is frequently violated in practice [26], [13], [27], [7], [28], while we do not make such assumption; (iii) [9] assumes that the set L_m of compromised links is given, while we treat it as a decision variable, which allows us to model a more intelligent attacker that strategically places attacks. In fact, the selection of L_m significantly impacts the capability of an attack and is the focus of this work.

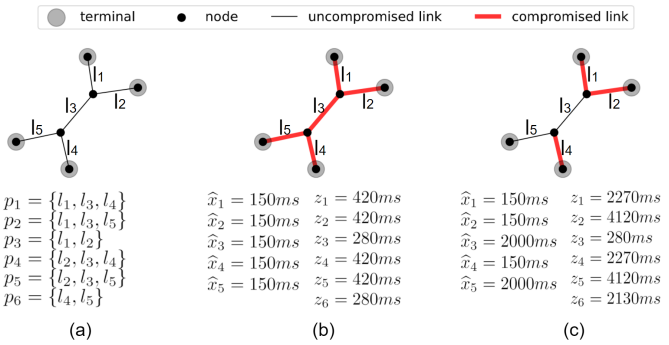

Fig. 2. Example: (a) input, (b) optimal manipulations under one L_m, (c) optimal manipulations under another L_m.

D. Example

Consider the example in Fig. 2 (a). Suppose that before the attack, each link has a delay of 10 ms, $\tau = 150$ ms, and $\tau_{\max} = 2000$ ms. Fig. 2 (b) shows the optimal manipulations under an intuitive selection of L_m—compromising all the links. In this case, the attack can cause 2240 ms of extra delay in total, by injecting a delay of z_i onto path p_i at some of the compromised links traversed by p_i. Fig. 2 (c) shows the optimal manipulations under another selection of L_m, which, although having fewer compromised links, is able to cause 15190 ms of extra delay, as the uncompromised links l_3 and l_5 can be used to explain the large delays of paths p_1, p_2, p_4, p_5, p_6 to network tomography without exposing the compromised links. Note that the inferred link metrics can differ from the actual metrics, and the compromised links can behave inconsistently across paths, e.g., in Fig. 2 (c), link l_1 injects no more than 280 ms of delay onto p_3 but 4120 ms of delay onto p_2. This example shows that DGoS attack can cause large damage without being localized, and the amount of damage critically depends on the selection of L_m.

III. Optimal Attack Strategy

Although (1) is a joint optimization of both L_m and $\widehat{\mathbf{x}}$, we will show that the main challenge is in optimizing L_m, which can be reduced to a novel "minimum cut" problem.

A. Optimizing $\widehat{\mathbf{x}}$ under Given L_m

Given the set of compromised links L_m, (1) is a *linear program (LP)* in $\widehat{\mathbf{x}}$ that can be solved in polynomial time by standard LP solvers, and the result gives the optimal manipulation vector (under the given L_m) by (2). Nevertheless, there are several simplifications that can be used to speed up the solution for large networks.

First, we observe that constraint (1c) has no effect on the optimal solution, as it only imposes a lower bound on $\widehat{\mathbf{x}}$, while the objective (1a) tries to increase $\widehat{\mathbf{x}}$. We can thus drop this constraint without changing the optimal solution to $\widehat{\mathbf{x}}$.

Furthermore, we observe that the dimension of the solution space can be reduced. To this end, we rewrite (1) after dropping constraint (1c) in a vector form:

$$\max \mathbf{1}_{|P_m|} R_m(\widehat{\mathbf{x}} - \mathbf{x}) \tag{3a}$$

$$\text{s.t. } R_n\widehat{\mathbf{x}} = R_n\mathbf{x}, \tag{3b}$$

$$\boldsymbol{\phi} \geq \widehat{\mathbf{x}} \geq \mathbf{0}, \tag{3c}$$

where $\mathbf{1}_{|P_m|}$ is the $1\times|P_m|$ vector of 1's, $R_m = (\mathbf{r}_i)_{p_i\in P_m}$ and $R_n = (\mathbf{r}_i)_{p_i\in P_n}$ are the sub-measurement matrices representing all the compromised/uncompromised paths, respectively, and $\boldsymbol{\phi} := (\phi_j)_{l_j\in L}$ is the vector of upper bounds on $\widehat{x}_j$ in (1d,1e), i.e.,

$$\phi_j := \begin{cases} \tau & \text{if } l_j \in L_m, \\ \tau_{\max} & \text{if } l_j \in L_n. \end{cases} \tag{4}$$

The "$\geq$" in (3c) means element-wise $\geq$.

To reduce the dimension for optimization (3), we perform a change of variable as follows. Since $x_j \leq \tau$ ($\forall l_j \in L$), it is easy to see that $\widehat{\mathbf{x}} = \mathbf{x}$ is a feasible solution to (3). Let B be a matrix whose columns form a basis of null(R_n), the null space of R_n. Let nullity(R_n) denote the *nullity* of R_n, i.e., the dimension of null(R_n). Then $\widehat{\mathbf{x}} = B\mathbf{c} + \mathbf{x}$ will always satisfy $R_n\widehat{\mathbf{x}} = R_n\mathbf{x}$ for any (nullity(R_n) $\times$ 1)-vector $\mathbf{c}$. Substituting $\widehat{\mathbf{x}}$ by $B\mathbf{c} + \mathbf{x}$, (3) is transformed into:

$$\max\ \mathbf{1}_{|P_m|}R_mB\mathbf{c} \tag{5a}$$
$$\text{s.t. } \boldsymbol{\phi} - \mathbf{x} \geq B\mathbf{c} \geq -\mathbf{x}, \tag{5b}$$

which is an optimization in $\mathbf{c}$.

Compared to (3), the number of decision variables in (5) is reduced from the number of links to the nullity of R_n. By the *rank-nullity theorem*, rank(R_n) + nullity(R_n) = $|L|$, and hence the reduction will be significant when rank(R_n) is large, i.e., the number of linearly independent uncompromised paths is large.

B. Property of the Optimal L_m

To facilitate the optimization of L_m, we first investigate the property of the optimal solution. As is shown in Section II-D, simply compromising all the links is generally suboptimal, as the attacker will have to make all the link metrics appear normal (i.e., $\widehat{x}_j \leq \tau$ for all $l_j \in L$), which limits the amount of performance degradation he can inject on each path.

Generally, compromising a link l_j can have two contradicting effects:

1) previously uncompromised paths that traverse l_j can now be controlled by the attacker, which removes some constraints of the type (1b) and hence may increase the objective value;
2) instead of constraint (1d), l_j will be subject to a tighter constraint (1e), which may decrease the objective value.

Due to these contradicting effects, it is not obvious what is the optimal set of links to compromise.

Our main result is a closed-form characterization of the optimal set of compromised links. To present this result, we introduce the following definitions.

Definition 1. Given a set of paths P, we define:

1) the *traversal number* of link l, denoted by w_l, as the number of paths in P that traverse link l;
2) a *cut* C of P as a subset of links such that every $p \in P$ traverses at least one link in C;
3) the *minimum-traversal cut* C^* of P as the cut of P with the minimum total traversal number, i.e., $\sum_{l\in C^*} w_l \leq \sum_{l\in C} w_l$ for any cut C.

Theorem III.1. The optimal set of compromised links L_m^* (i.e., the optimal solution to L_m in (1)) is the minimum-traversal cut of P.

We will prove this theorem in two steps. **Step 1** is to show that L_m^* must be a cut of P, as otherwise the attacker will be able to improve his objective value by compromising one more link.

Lemma III.2. Suppose that for the initial set of compromised links $L_m^{(0)}$, there is at least one uncompromised path p_{i^*}. Then there must exist an uncompromised link $l_{j^*} \in p_{i^*}$, such that compromising l_{j^*} increases the total performance degradation, i.e., $\Gamma(L_m^{(0)} \cup \{l_{j^*}\}) \geq \Gamma(L_m^{(0)})$, where $\Gamma(L')$ is the optimal objective value of (1) when $L_m = L'$.

Proof. Let $L_m^{(0)}$ ($L_n^{(0)}$) be the initial set of compromised (uncompromised) links, $P_m^{(0)}$ ($P_n^{(0)}$) be the initial set of compromised (uncompromised) paths, and $\widehat{\mathbf{x}}^{(0)}$ be the optimal solution to $\widehat{\mathbf{x}}$ when $L_m = L_m^{(0)}$. By assumption, $p_{i^*} \in P_n^{(0)}$.

First, we observe that there must exist a link $l_{j^*} \in p_{i^*}$ for which $\widehat{x}_{j^*}^{(0)} \leq \tau$, as otherwise (i.e., $\widehat{x}_j^{(0)} > \tau$ for all $l_j \in p_{i^*}$), we will have $\mathbf{r}_{i^*}\widehat{\mathbf{x}}^{(0)} > |p_{i^*}|\tau \geq \mathbf{r}_{i^*}\mathbf{x}$, where $|p_{i^*}|$ is the hop count on p_{i^*}. This contradicts with $\mathbf{r}_{i^*}\widehat{\mathbf{x}}^{(0)} = \mathbf{r}_{i^*}\mathbf{x}$ according to constraint (1b).

Next, for the above link l_{j^*}, adding a constraint $\widehat{x}_{j^*} \leq \tau$ to (1) will not change the optimal solution when $L_m = L_m^{(0)}$. That is, $\widehat{\mathbf{x}}^{(0)}$ remains an optimal solution to the following optimization in $\widehat{\mathbf{x}}$

$$\max \sum_{p_i\in P_m^{(0)}} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{6a}$$
$$\text{s.t. } \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) = 0, \qquad \forall p_i \in P_n^{(0)}, \tag{6b}$$
$$\tau_{\max} \geq \widehat{x}_j \geq 0, \qquad \forall l_j \in L_n^{(0)} \setminus \{l_{j^*}\}, \tag{6c}$$
$$\tau \geq \widehat{x}_j \geq 0, \qquad \forall l_j \in L_m^{(0)} \cup \{l_{j^*}\}. \tag{6d}$$

Note that we can omit constraint (1c) as explained in Section III-A.

Moreover, after compromising link l_{j^*}, i.e., for $L_m = L_m^{(0)} \cup \{l_{j^*}\}$, the optimization (1) becomes

$$\max \sum_{p_i\in P_m^{(0)}} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) + \sum_{p_i\in P_m^{(1)}\setminus P_m^{(0)}} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{7a}$$
$$\text{s.t. } \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) = 0, \qquad \forall p_i \in P_n^{(1)}, \tag{7b}$$
$$\tau_{\max} \geq \widehat{x}_j \geq 0, \qquad \forall l_j \in L_n^{(1)}, \tag{7c}$$
$$\tau \geq \widehat{x}_j \geq 0, \qquad \forall l_j \in L_m^{(1)}, \tag{7d}$$

where $L_m^{(1)}$ ($L_n^{(1)}$) is the new set of compromised (uncompromised) links, and $P_m^{(1)}$ ($P_n^{(1)}$) is the new set of compromised (uncompromised) paths.

Finally, since $P_n^{(1)} \subseteq P_n^{(0)}$, $L_n^{(1)} = L_n^{(0)} \setminus \{l_{j^*}\}$, and $L_m^{(1)} = L_m^{(0)} \cup \{l_{j^*}\}$, any feasible solution to (6) remains feasible for (7). In particular, $\widehat{\mathbf{x}}^{(0)}$ is a feasible solution to (7), with an objective value of $\sum_{p_i\in P_m^{(0)}} \mathbf{r}_i(\widehat{\mathbf{x}}^{(0)} - \mathbf{x}) = \Gamma(L_m^{(0)})$. Thus, under the optimal solution to (7), the objective value $\Gamma(L_m^{(0)} \cup \{l_{j^*}\})$ must be no smaller than $\Gamma(L_m^{(0)})$. □

Step 2 is to show that among all the cuts, L_m^* must be the one that minimizes the total traversal number.

Lemma III.3. Among all the cuts of P, the optimal set of links to compromise is the cut with the minimum total traversal number.

Proof. By definition, if L_m is a cut of P, then $P_m = P$ and $P_n = \emptyset$, which simplifies (1) for a given L_m to

$$\max \sum_{p_i \in P} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{8a}$$

$$\text{s.t. } \tau_{\max} \geq \widehat{x}_j \geq 0, \qquad \forall l_j \in L_n, \tag{8b}$$

$$\tau \geq \widehat{x}_j \geq 0, \qquad \forall l_j \in L_m. \tag{8c}$$

It is easy to see that the optimal solution to (8) is $\widehat{x}_j = \tau$ if $l_j \in L_m$ and $\widehat{x}_j = \tau_{\max}$ if $l_j \in L_n$. Under this solution, the objective value of (8) equals

$$\begin{aligned} &\sum_{p_i \in P} (m_i \tau + (|p_i| - m_i)\tau_{\max}) - \sum_{p_i \in P} \mathbf{r}_i \mathbf{x} \\ &= (\tau - \tau_{\max}) \sum_{p_i \in P} m_i + \tau_{\max} \sum_{p_i \in P} |p_i| - \sum_{p_i \in P} \mathbf{r}_i \mathbf{x}, \end{aligned} \tag{9}$$

where m_i is the number of compromised links on path p_i and $|p_i|$ is the total number of links on p_i. Only the first term $(\tau - \tau_{\max}) \sum_{p_i \in P} m_i$ depends on L_m.

As $\tau - \tau_{\max} \leq 0$, maximizing (9) is equivalent to minimizing $\sum_{p_i \in P} m_i$. We further note that

$$\sum_{p_i \in P} m_i = \sum_{p_i \in P} \sum_{l \in p_i} \mathbb{1}_{l \in L_m} = \sum_{l \in L_m} \sum_{p_i \in P} \mathbb{1}_{l \in p_i} = \sum_{l \in L_m} w_l, \tag{10}$$

where $\mathbb{1}_{\cdot}$ is the indicator function. Thus, the optimal solution to L_m among all the cuts is the cut with the minimum total traversal number. □

Proof of Theorem III.1. By Lemma III.2, L_m^* must be a cut of P. Then by Lemma III.3, it must have the minimum total traversal number among all the cuts. Therefore, L_m^* must be the minimum-traversal cut. □

Remark: The minimum-traversal cut of P may not be unique. From the proof of Theorem III.1, we see that all the minimum-traversal cuts are equally optimal.

Theorem III.1 implies that given a set of targeted paths P, the optimal set L_m of links to compromise is the solution to a novel combinatorial optimization problem as follows.

Definition 2. Given a set of paths P, the *adversarial link selection (ALS)* problem is to find the cut of P with the minimum total traversal number.

C. Hardness Analysis

Below we show the hardness of ALS by connecting it to several well-known hard problems in combinatorial optimization in both the general case and a nontrivial special case.

1) ALS is NP-hard: We show that ALS for an arbitrary set of P is NP-hard. To show this, we consider the corresponding decision problem: determine whether a set of paths P has a cut with a given total traversal number T. We will prove that the decision version of ALS is NP-hard by showing a reduction from the exact cover problem [41].

Proof of ALS being NP-hard. Given a set of elements of $E = \{e_1, e_2, \ldots, e_n\}$ and a collection S of subsets of E, an exact cover is a subcollection S^* of S such that each element in E is covered once and only once by sets in S^*. To determine if there exists an exact cover is NP-complete [41].

The exact cover problem can be reduced to the following instance of ALS. We construct a set of paths $P = \{p_1, p_2, \ldots, p_n\}$ in one-one correspondence with the set of elements $E = \{e_1, e_2, \ldots, e_n\}$. Similarly, we construct a set of links $L = \{l_1, l_2, \ldots, l_m\}$ in one-one correspondence with the collection of sets $S = \{s_1, s_2, \ldots, s_m\}$. The relationship between the paths and the links is such that link l_i is traversed by path p_j if and only if set s_i covers element e_j. Note that such construction is always possible as we allow P to contain arbitrary paths in the general case of ALS. Then we claim that there exists an exact cover S^* of E if and only if the constructed instance of ALS has a cut with a total traversal number of $|P|$.

Suppose that there exists an exact cover S^*, i.e., $E \subseteq \bigcup_{s \in S^*} s$ and $\sum_{s \in S^*} |s| = |E|$. According to the above construction, the corresponding set of links $C^* = \{l_i : s_i \in S^*\}$ must cut each path in P once and only once, and hence C^* is a cut with a total traversal number of $|P|$.

Conversely, suppose that the constructed set of paths P has a cut C^* with a total traversal number of $|P|$. By Definition 1, C^* must cut each path in P once and only once. According to the construction, the corresponding subcollection $S^* = \{s_i : l_i \in C^*\}$ must cover each element in E once and only once, i.e., S^* is an exact cover of E. □

2) Hardness of all-possible-paths ALS: Now consider a special case where P contains all possible paths between a given set K of terminals. This case models networks that employ advanced routing mechanisms such as source routing or Software Defined Networking (SDN), that allow traffic to be routed on any path between a pair of terminals. We call the ALS problem in this special case *all-possible-paths ALS*.

All-possible-paths ALS can reduce to the Multiway Cut problem [42]. Also known as the Multiterminal Cut problem, the Multiway Cut problem is a graph division problem, where given an undirected graph $\mathcal{G}(V, E)$ with link weights $w : E \to \mathbb{R}^+$ and a set of terminals $K \subseteq V$, we want to find a subset of links with the minimum total weight to cut all the paths between the terminals. When the number of terminals equals 2, the Multiway Cut problem becomes the min-cut problem, which can be solved efficiently by the max-flow algorithms. We see that all-possible-paths ALS is a special case of Multiway Cut, where the weights are the traversal numbers. We note that the two problems are not equivalent: in Multiway Cut, the link weights are arbitrary; in all-possible-paths ALS, the link weights are the traversal numbers, which are determined by the network topology and the locations of terminals.

It is known that Multiway Cut is NP-hard, even in a very special case when all the links have unit weights.

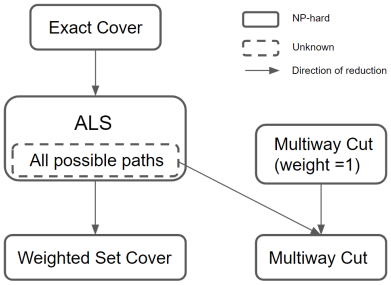

Fig. 3. Relationship between ALS and known NP-hard problems.

Theorem III.4 ([42])**.** The Multiway Cut problem is NP-hard for all $|K| \geq 3$, even if all the link weights are equal to 1.

The hardness of all-possible-paths ALS still remains an open question. Based on Theorem III.4, we conjecture that all-possible-paths ALS is NP-hard, since it is also a special case of Multiway Cut.

Fig. 3 summarizes the relationship between ALS and known NP-hard problems, where the arrows indicate the direction of reduction. As shown in Section III-D1, ALS can reduce to the Weighted Set Cover (WSC) problem, which is also NP-hard.

D. Approximation Algorithms

As ALS is NP-hard, there is no polynomial-time exact algorithm for it unless P = NP. As we mentioned, ALS reduces to Weighted Set Cover (WSC), and all-possible-paths ALS reduces to Multiway Cut. Below we will use known approximation algorithms designed for WSC and Multiway Cut to solve ALS and all-possible-paths ALS, respectively.

1) The greedy algorithm for ALS: Given a set of elements $E = \{e_1, e_2, \ldots, e_n\}$ and a collection $S = \{s_1, s_2, \ldots, s_m\}$ of subsets of E, where each s_i has a weight of w_i, WSC aims at finding the subcollection S^* that covers E with the minimum total weight.

We reduce ALS (for arbitrary P) to WSC as follows. Given a set of paths $P = \{p_1, p_2, \ldots, p_n\}$ traversing a set of links $L = \{l_1, l_2, \ldots, l_m\}$, we construct a set of elements $E = \{e_1, e_2, \ldots, e_n\}$ in one-one correspondence with the paths, and a collection of sets $S = \{s_1, s_2, \ldots, s_m\}$ in one-one correspondence with the links, such that set s_i covers element e_j if and only if link l_i is on path p_j, as illustrated in Fig. 4. Each set s_i has a weight w_i that equals the traversal number of link l_i. It is easy to see that finding the cut with the minimum total traversal number is equivalent to finding the subcollection of sets to cover all the elements with the minimum total weight. We note that in the constructed instance of WSC, the weight of a set always equals its cardinality (i.e., $w_i = |s_i|$), and thus ALS is a special case of WSC.

We apply a well-known greedy algorithm [43], designed for solving WSC, to the ALS problem. The algorithm iterates until all the paths are compromised, where in each iteration, it picks a link with the smallest cost-value ratio and adds the paths traversing it to the set of compromised paths. For a link l, we define the cost-value ratio by $\frac{|P_l|}{|P_l \setminus P_m|}$, where P_l is the set of paths traversing link l. Since the link weight equals $|P_l|$, this ratio is the cost we pay for each newly compromised path, if link l is selected. The pseudocode is shown in Algorithm 1.

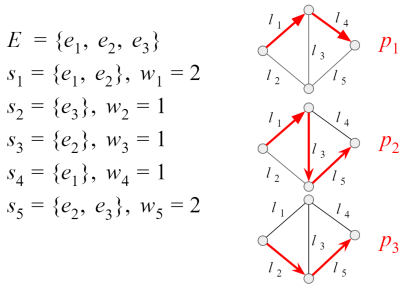

Fig. 4. Reducing ALS to Weighted Set Cover

Algorithm 1: Greedy ALS

input : Paths P
output: Compromised links L_m

$P_m \leftarrow \emptyset$;
$L_m \leftarrow \emptyset$;
while $P_m \neq P$ **do**
  Find the link l with the smallest ratio $\frac{|P_l|}{|P_l \setminus P_m|}$;
  $P_m \leftarrow P_m \cup P_l$;
  $L_m \leftarrow L_m \cup l$;
return L_m;

Although straightforward, this greedy algorithm is known to have the best approximation guarantee for WSC [43]. Applied to our problem, it guarantees the following.

Theorem III.5 ([43])**.** Algorithm 1 achieves an approximation factor of $H_{|P|} = 1 + \frac{1}{2} + \ldots + \frac{1}{|P|} = \Theta(\log |P|)$ for ALS, i.e., $T^{\text{greedy}} \leq H_{|P|} T^{\text{opt}} = \Theta(log|P|) T^{\text{opt}}$, where T^{greedy} is the total traversal number achieved by Algorithm 1 and T^{opt} is the minimum total traversal number of all the cuts of P.

However, our ultimate goal is to maximize the performance degradation measured by (1a). We can substitute $\sum_{p_i \in P} m_i$ by T^{greedy} in (9) to get the corresponding objective value.

Corollary III.6. Using Algorithm 1 to select the compromised links and the LP (5) to compute the manipulations achieves a total performance degradation of

$$(\tau - \tau_{\max}) T^{\text{greedy}} + \tau_{\max} \sum_{p_i \in P} |p_i| - \sum_{p_i \in P} \mathbf{r}_i \mathbf{x} \geq$$
$$(\tau - \tau_{\max}) H_{|P|} T^{\text{opt}} + \tau_{\max} \sum_{p_i \in P} |p_i| - \sum_{p_i \in P} \mathbf{r}_i \mathbf{x}, \quad (11)$$

where T^{greedy} and T^{opt} are defined as in Theorem III.5.

2) CKR relaxation with rounding for all-possible-paths ALS: As mentioned in Section III-C2, all-possible-paths ALS reduces to the Multiway Cut problem, which means we can apply algorithms for Multiway Cut to all-possible-paths ALS.

Calinescu et al. [44] proposed an approach called *CKR relaxation* for Multiway Cut, for which it has been proved that it is NP-hard to achieve a better integrality gap than CKR relaxation for any fixed number of terminals, assuming the Unique Games Conjecture to hold [45]. In a minimization problem, the *integrality gap* is the ratio between the objective value of the optimal integer solution and that of its relaxation, i.e., $\text{OPT}_{\text{int}}/\text{OPT}_{\text{relaxation}}$. We first formulate the Multiway Cut problem as an integer program, and then introduce its CKR relaxation. Given a set V of nodes, a set E of links with

weights $(w_{v,v'})_{(v,v')\in E}$, and a set K ($K \subseteq V$) of terminals, the Multiway Cut problem aims at solving

$$\min \frac{1}{2} \sum_{(v,v')\in E} \sum_{t\in K} w_{v,v'} |x_{v,t} - x_{v',t}| \tag{12a}$$

$$\text{s.t.} \sum_{t\in K} x_{v,t} = 1, \qquad \forall v \in V, \tag{12b}$$

$$x_{t,t} = 1, \qquad \forall t \in K, \tag{12c}$$

$$x_{v,t} \in \{0,\ 1\}, \qquad \forall v \in V,\ t \in K, \tag{12d}$$

where $x_{v,t}$ is the decision variable indicating whether node v will be connected to terminal t after the cut.

By relaxing the integer constraint (12d), we can get the following convex optimization:

$$\min \frac{1}{2} \sum_{(v,v')\in E} \sum_{t\in K} w_{v,v'} |x_{v,t} - x_{v',t}| \tag{13a}$$

$$\text{s.t.} \sum_{t\in K} x_{v,t} = 1, \qquad \forall v \in V, \tag{13b}$$

$$x_{t,t} = 1, \qquad \forall t \in K, \tag{13c}$$

$$x_{v,t} \geq 0, \qquad \forall v \in V,\ t \in K. \tag{13d}$$

Replacing $|x_{v,t} - x_{v',t}|$ by a new variable $y_{v,v',t}$ s.t.

$$y_{v,v',t} \geq x_{v,t} - x_{v',t}, \qquad \forall (v,v') \in E,\ t \in K, \tag{14}$$

$$y_{v,v',t} \geq x_{v',t} - x_{v,t}, \qquad \forall (v,v') \in E,\ t \in K, \tag{15}$$

converts (13) into an LP [43], i.e., an LP relaxation of (12).

For each node v and each terminal t, the solution $\bar{x}_{v,t}$ to the LP relaxation can be viewed as the probability of assigning v to (the connected component containing) t after the cut. A rounding scheme is used to convert this fractional value to either 0 or 1, subject to the constraint (13b). Different rounding schemes lead to different approximation factors. Specifically, the randomized rounding scheme achieves an approximation factor of 1.5 [43], and the best-known rounding scheme can achieve an approximation factor of 1.2965 [46].

Finally, given the rounded value of $x_{v,t}$ ($\forall v \in V,\ t \in K$), the cut is the set of all the links whose endpoints are assigned to different terminals, i.e., $L_m = \{(v,v') \in E : \exists t,\ t' \in K \text{ with } t \neq t',\ \text{satisfying } x_{v,t} = x_{v',t'} = 1\}$.

By similar argument as Corollary III.6, we can bound the overall performance of the attack as follows.

Corollary III.7. Using CKR relaxation with an α-approximation rounding scheme to select the compromised links and the LP (5) to compute the manipulations achieves a total performance degradation of

$$(\tau - \tau_{\max})T^{\text{CKR}} + \tau_{\max} \sum_{p_i\in P} |p_i| - \sum_{p_i\in P} \mathbf{r}_i\mathbf{x} \geq (\tau - \tau_{\max})\alpha T^{\text{opt}} + \tau_{\max} \sum_{p_i\in P} |p_i| - \sum_{p_i\in P} \mathbf{r}_i\mathbf{x}, \tag{16}$$

where T^{CKR} is the total traversal number of the links selected by CKR relaxation, and T^{opt} is the minimum total traversal number of all the multiway cuts between the terminals.

TABLE I summarizes the performance guarantee of the aforementioned algorithms in solving ALS.

TABLE I
APPROXIMATION ALGORITHMS FOR ALS

algorithm	case	approximation factor
Greedy ALS	general	$\Theta(\log \lvert P \rvert)$
CKR relaxation	all-possible-paths	α[1]

TABLE II
PARAMETERS OF ISP TOPOLOGIES

Network	size	#nodes	#links	#candidate terminals[2]
Bics	small	33	48	16
BTN	small	53	65	25
Colt	medium	153	191	45
Cogent	medium	197	245	21
AS 20965	large	968	8283	75
AS 8717	large	1778	3755	1075

IV. PERFORMANCE EVALUATION

We conduct simulations to evaluate the capabilities of an intelligent attacker employing our strategies in comparison with benchmarks, based on real ISP topologies. To be concrete, we consider delay-based DGoS attacks, where the attacker tries to inject the maximum amount of delay onto a set of targeted paths, while the user of these paths tries to localize links with abnormally large delays by network tomography.

A. Experiment Setup

1) Network topology: We use real network topologies from public datasets, whose parameters are shown in the TABLE II. The first four topologies are Point of Presence (PoP)-level topologies from the Internet Topology Zoo [47], and the last two topologies are router-level topologies from the CAIDA project [48]. We classify the topologies into small, medium, and large networks. For each topology, we select a given number of terminals uniformly at random from low-degree nodes (degree ≤ 2), and repeat this selection for 20 times.

2) Parameter setting: For each topology and each set of selected terminals, we compute the paths in P in two ways:
(i) *All possible paths*: In this case, P contains all the cycle-free paths between the terminals. Note that cutting all the cycle-free paths is equivalent to cutting all the paths between the terminals. Since the number of all the cycle-free paths can grow exponentially with the network size, we only evaluate this case on the small networks.
(ii) *Shortest paths*: In this case, P only contains one shortest path (in hop count) for each pair of terminals, with ties broken arbitrarily. Since there are only $\binom{|K|}{2}$ paths for $|K|$ terminals, we evaluate this case on the medium–large networks.

Before the attack, each link has a delay of 10 ms. A link is considered as "normal" if its delay is within 150 ms, i.e., $\tau = 150$. The maximum delay at a link is 2000 ms, i.e., $\tau_{\max} = 2000$.

[1]The constant α depends on the rounding scheme, e.g., 1.5 for randomized rounding and 1.2965 for the rounding scheme in [46].

[2]For Bics, these are all the nodes with degree ≤ 2; for the other networks, these are all the nodes with degree one.

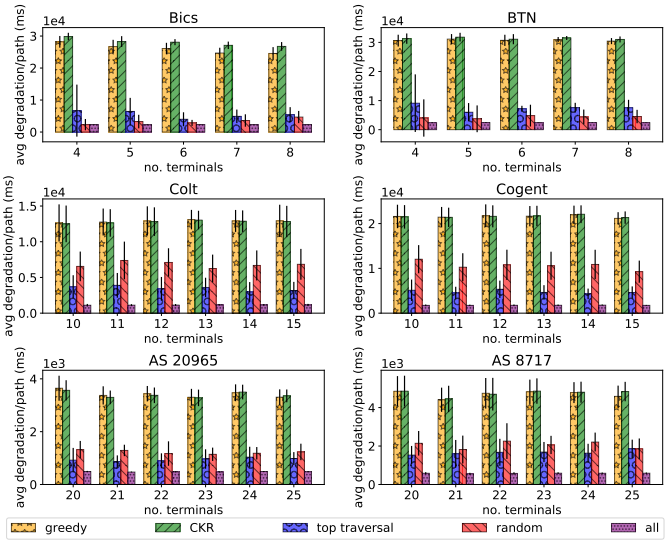

Fig. 5. Average delay degradation for the case of all possible paths (top 2) and the case of shortest paths (bottom 4)

3) Benchmarks: We compare the two proposed algorithms, Algorithm 1 ('greedy') and CKR relaxation with randomized rounding ('CKR'), with the following three heuristics for selecting the set of compromised links:

(i) "Random selection" ('random'): To illustrate the capability of an attacker who cannot actively select which links to compromise, this algorithm selects k links uniformly at random, where k is set to the number of compromised links selected by 'CKR' to facilitate comparison.

(ii) "Top traversal" ('top traversal'): Based on the intuition that compromising the most traversed links will provide control over more paths, this algorithm selects the k links with the largest traversal numbers. Again, to facilitate comparison, k is set to the number of compromised links selected by 'CKR'.

(iii) "Compromise all" ('all'): Compromising all the links is also a very intuitive approach to maximize the damage the attacker can cause to the network.

Under each selection of compromised links, we solve the LP (5) to compute the total performance degradation (measured by the total amount of delay injected by the attacker over all the paths) under the optimal manipulations.

B. Results

Overall, we observe that the proposed algorithms ('greedy' and 'CKR') perform similarly to each other and significantly better than the heuristic algorithms. More importantly, these algorithms show that it is possible to introduce significant delay on communication paths without being localized by network tomography, signaling the need of new defenses.

1) Case of all possible paths: In the case that P contains all the possible paths between the terminals, the results are shown in Fig. 5 (top 2). The y-axis is the performance of the attacker measured by the average injected delay per path (plus/minus one standard deviation), computed over 20 sets of randomly selected terminals, and the x-axis is the number of terminals. In this experiment, 'CKR' performs the best as expected, as it has the best approximation guarantee. Both 'CKR' and 'greedy' perform much better than the heuristic algorithms, demonstrating the importance of carefully selecting the compromised links in modeling the capabilities of intelligent attackers. Among the heuristic algorithms, 'top traversal' performs the best, as it leads to more compromised paths than 'random'. However, it is not sufficient to just compromise more paths. To prevent the compromised links from being detected as bad links by network tomography, the attacker needs to ensure constraint (1e). Therefore, compromising too many links can reduce the attacker's capability in injecting delays. This is why 'all' performs the worst.

2) Case of shortest paths: Similar results are shown in Fig. 5 (bottom 4) for the case where P only contains the shortest paths between the terminals. We see that 'greedy' and 'CKR' still significantly outperform the other algorithms. However, 'CKR' is not always the best any more, because it is not designed for this case. In particular, 'CKR' will select links to cut all the possible paths between the terminals, while the ALS problem in this case only needs to cut the shortest paths. Because of that, 'CKR' may compromise more links than necessary, which reduces the attacker's capability to manipulate the path delays.

In both cases, the best algorithm is able to inject significant delays (4–30 seconds/path) without exposing the compromised links to network tomography.

V. Constrained Attacks

So far we have assumed that the attacker can compromise any subset of links. In practice, however, there are usually constraints on which and/or how many links the attacker is capable of compromising. To shed light on the impact of such constraints, we analyze the optimal attack strategy under the constraint that $|L_m| \leq k$ for a given $k \in \mathbb{Z}^+$. We leave the investigation of other types of constraints to future work.

A. Asymptotic Property of the Optimal L_m

As the cardinality constraint generally invalidates Theorem III.1, new results are needed to reveal properties of the optimal set of compromised links under this constraint. First, we observe that Lemmas III.2 and III.3 imply the following.

Lemma V.1. Let L_m^* be the optimal set of compromised links under a cardinality constraint k. Then:

1) if L_m^* is not a cut of P, its cardinality must equal k;
2) if L_m^* is a cut of P, it must have the minimum total traversal number among all the cuts of cardinality $\leq k$.

Proof. By Lemma III.2, if L_m^* is not a cut and $|L_m^*| < k$, then we must be able to improve L_m^* by adding another link to it, leading to a contradiction with the optimality of L_m^*.

By Lemma III.3, the optimal choice of L_m among the cuts is the one that minimizes the total traversal number. The proof still holds if choices are limited to cuts of cardinality $\leq k$. □

What is missing is an exact description of the optimal choice of L_m from all the cuts and non-cuts of up to k links. To this end, we have the following result.

Theorem V.2. If $\tau_{\max} \gg \mathbf{r}_i\mathbf{x}$ ($\forall p_i \in P$) and $\tau_{\max} \gg \tau$, then the optimal set L_m^* of compromised links under a cardinality constraint k is the one achieving:

$$\max \sum_{l_j \in L_n'} \sum_{p_i \in P_m} r_{ij} =: T_m \tag{17a}$$

$$\text{s.t. } L_m \subseteq L,\ |L_m| \leq k, \tag{17b}$$

where $L_n' := L_n \setminus \bigcup_{p \in P_n} p$ is the set of uncompromised links that are only traversed by compromised paths.

Proof. We rewrite the objective function (1a) as

$$\sum_{p_i \in P_m} \sum_{l_j \in L} r_{ij}(\widehat{x}_j - x_j) = \sum_{l_j \in L} \sum_{p_i \in P_m} r_{ij}(\widehat{x}_j - x_j). \tag{18}$$

If $l_j \in L_m$, then $\widehat{x}_j \leq \tau$ by (1e). If $l_j \in L_n$, then $\widehat{x}_j \leq \min(\tau_{\max}, \min_{i:\, p_i \in P_n, r_{ij}=1} \mathbf{r}_i\mathbf{x})$ by (1b,1d). For a large $\tau_{\max}$, $\widehat{x}_j$ can achieve $\tau_{\max}$ if and only if $l_j \in L_n'$. Thus, when $\tau_{\max}$ is large, (18) can be simplified to:

$$\sum_{l_j \in L} \sum_{p_i \in P_m} r_{ij}(\widehat{x}_j - x_j) \approx \tau_{\max} \sum_{l_j \in L_n'} \sum_{p_i \in P_m} r_{ij}. \tag{19}$$

Therefore, the objective in (1a) is asymptotically equivalent to the objective in (17a) (as $\tau_{\max} \to \infty$). □

We note that while Theorem V.2 is only proved for the case of large $\tau_{\max}$, it reduces to Theorem III.1 when L_m is restricted to cuts. This is because if L_m is a cut, then $L_n' = L_n$ and $P_m = P$, and hence $\sum_{l_j \in L_n'} \sum_{p_i \in P_m} r_{ij} = \sum_{l_j \in L_n} \sum_{p_i \in P} r_{ij}$, which is the total traversal number of all the uncompromised links. Maximizing the total traversal number of the uncompromised links is equivalent to minimizing the total traversal number of the compromised links.

B. Heuristic Algorithms and Evaluation

Theorem V.2 implies another novel combinatorial optimization problem formulated by (17), which we call the *constrained adversarial link selection (CALS)* problem. Due to space limitation, we will present initial results on solving CALS below, and leave further investigation to future work.

Heuristic algorithms: Although Algorithm 1 ('greedy ALS') can be easily adapted to satisfy the cardinality constraint (17b) by stopping after selecting k links, it is not designed for the objective of CALS. Nevertheless, we can modify it for CALS, referred to as *Greedy CALS* ('greedy CALS'). Greedy CALS has the same structure as Algorithm 1, except:

- line 3 is replaced by "while $P_m \neq P$ and $|L_m| < k$";
- line 4 is replaced by "find the link l yielding the maximum increase in T_m" (where T_m is defined in (17a)).

Moreover, the heuristics "random selection" and "top traversal" (see Section IV-A3) can also be easily adapted to satisfy the cardinality constraint (17b).

Evaluation results: We evaluate these algorithms under the setup in Section IV-A, except that we fix the number of terminals and vary the constraint k. Table III shows the parameter values, where the last column gives the number of links selected by the original, unconstrained Greedy ALS (Algorithm 1) in order to cut all the paths in P. The results are shown in Fig. 6, where $k = \infty$ is the unconstrained case.

TABLE III
PARAMETERS FOR EVALUATING CONSTRAINED ATTACKS

Network	#terminals	k	$\lvert L_m \rvert$ by Greedy ALS[3]
Bics	8	[2, 6]	13.85 (1.85)
BTN	8	[2 ,6]	7.4 (0.86)
Colt	15	[9, 13]	15.5 (1.02)
Cogent	15	[9, 13]	20.8 (1.63)
AS 20965	25	[21, 25]	73.35 (9.84)
AS 8717	25	[21, 25]	44.25 (8.28)

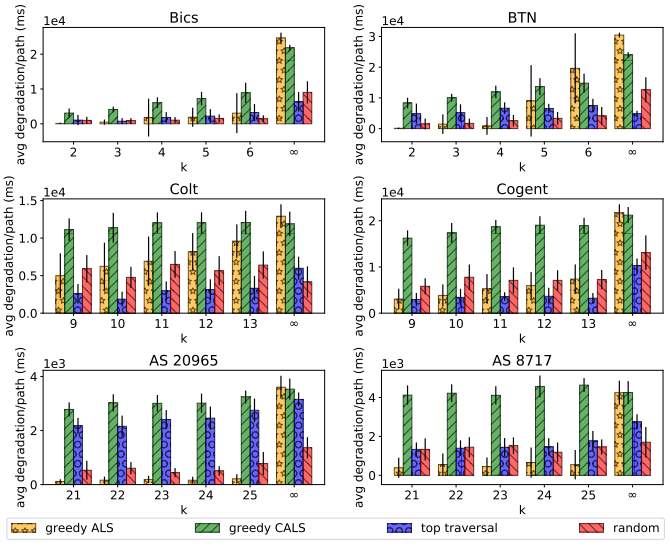

Fig. 6. Average delay degradation under a constrained number of compromised links

First of all, the result shows that it is necessary to change the objective from minimizing the total traversal number to maximizing T_m (17a) when we (the attacker) are not able to compromise a cut due to constraints. This is indicated by the poor performance of 'greedy ALS' compared to 'greedy CALS' when k is small. Meanwhile, when we are able to compromise a cut ($k = \infty$), 'greedy CALS' performs almost as well as 'greedy ALS', and far better than the other heuristics. These observations indicate that 'greedy CALS' better models the attacker's capability in the general case.

VI. CONCLUSION

This work helps to establish the fundamental limit of network tomography in adversarial environments by formulating and analyzing a novel type of attack, called the stealthy DeGrading of Service (DGoS) attack, that aims at maximally degrading the performance of targeted paths without being localized by network tomography. Through careful analysis, we derive explicit properties of the optimal attack strategy. The derived properties allow us to link our problem to well-known combinatorial optimization problems, and leverage existing algorithms with approximation guarantees. Our evaluations on real topologies show that the proposed attack can significantly degrade communication performances without being localized by network tomography, signaling the need of new defenses.

[3]It shows 'mean (standard deviation)' computed over 20 sets of terminals.

REFERENCES

[1] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data," in *ACM CoNEXT*, 2007.

[2] S. Zarifzadeh, M. Gowdagere, and C. Dovrolis, "Range tomgoraphy: Combining the practicality of boolean tomography with the resolution of analog tomography," in *ACM IMC*, 2012.

[3] D. Ghita, K. Argyraki, and P. Thiran, "Toward accurate and practical network tomography," *ACM SIGOPS Operating Systems Review*, vol. 47, no. 1, pp. 22–26, January 2013.

[4] N. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. Chan, "Non-adapative fault diagnosis for all-optical networks via combinatorial group testing on graphs," in *IEEE INFOCOM*, 2007.

[5] S. Ahuja, S. Ramasubramanian, and M. Krunz, "SRLG failure localization in optical networks," *IEEE/ACM Transations on Networking*, vol. 19, no. 4, pp. 989–999, Auguest 2011.

[6] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statistical Science*, 2004.

[7] Y. Zhao, Y. Chen, and D. Bindel, "Towards unbiased end-to-end network diagnosis," in *ACM SIGCOMM*, 2006.

[8] Z. Zhang, O. Mara, and K. Argyraki, "Network neutrality inference," in *ACM SIGCOMM*, 2014.

[9] S. Zhao, Z. Lu, and C. Wang, "When seeing isn't believing: On feasibility and detectability of scapegoating in network tomography," in *IEEE ICDCS*, 2017.

[10] Y. Vardi, "Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Assoc.*, pp. 365–377, 1996.

[11] M. Coates, A. O. Hero, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Magzine*, vol. 19, pp. 47–65, 2002.

[12] M. F. Shih and A. O. Hero, "Unicast inference of network link delay distributions from edge measurements," in *IEEE ICASSP*, 2001.

[13] V. N. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of internet link lossiness," in *IEEE INFOCOM*, April 2003.

[14] N. Duffield, "Simple network performance tomography," in *ACM SIGCOMM conference on Internet measurement*, 2003.

[15] ——, "Network tomography of binary network performance characteristics," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5373–5388, December 2006.

[16] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicase-based inference of network internal loss characteristics," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2462–2480, November 1999.

[17] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. Presti, and V. Paxson, "The use of end-to-end multicast measurements for characterizing internal network behavior," *IEEE Communications Magazine*, vol. 38, no. 5, pp. 152–159, May 2000.

[18] N. Duffield and F. Lo Presti, "Multicast inference of packet delay variance at interior network links," in *IEEE INFOCOM*, 2000.

[19] F. Lo Presti, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal delay distributions," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 761–775, Dec. 2002.

[20] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, December 2006.

[21] N. Duffield, F. LoPresti, V. Paxson, and D. Towsley, "Network loss tomography using striped unicast probes," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 697–710, August 2006.

[22] B. Xi, G. Michailidis, and V. Nair, "Estimating network loss rates using active tomography," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1430–1448, December 2006.

[23] E. Lawrence, G. Michailidis, and V. N. Nair, "Network delay tomography using flexicast experiments," *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 68, no. 5, pp. 785–813, 2006.

[24] M. Coates and R. Nowak, "Network tomography for internal delay estimation," in *IEEE ICASSP*, May 2001.

[25] M. F. Shih and A. O. Hero, "Unicast-based inference of network link delay distributions using mixed finite mixture models," *IEEE Transactions on Signal Processing, Special Issue on Signal Processing in Networking*, vol. 51, no. 9, pp. 2219–2228, August 2003.

[26] O. Gurewitz and M. Sidi, "Estimating one-way delays from cyclic-path delay measurements," in *IEEE INFOCOM*, 2001.

[27] Y. Chen, D. Bindel, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *ACM SIGCOMM*, 2004.

[28] A. Chen, J. Cao, and T. Bu, "Network tomography: Identifiability and Fourier domain estimation," in *IEEE INFOCOM*, 2007.

[29] H. Nguyen and P. Thiran, "The Boolean solution to the congested IP link location problem: Theory and practice," in *IEEE INFOCOM*, 2007.

[30] 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, March 2013.

[31] S. Tati, S. Silvestri, T. He, and T. LaPorta, "Robust network tomography in the presence of failures," in *IEEE ICDCS*, 2014.

[32] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, 2012.

[33] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Identifiability of link metrics based on end-to-end path measurements," in *ACM IMC*, 2013.

[34] ——, "Inferring link metrics from end-to-end path measurements: Identifiability and monitor placement," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1351–1368, June 2014.

[35] L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami, "Efficient identification of additive link metrics via network tomography," in *IEEE ICDCS*, 2013.

[36] S. Ahuja, S. Ramasubramanian, and M. Krunz, "SRLG failure localization in all-optical networks using monitoring cycles and paths," in *IEEE INFOCOM*, 2008.

[37] S. Cho and S. Ramasubramanian, "Localizing link failures in all-optical networks using monitoring tours," *Elsevier Computer Networks*, vol. 58, pp. 2–12, January 2014.

[38] L. Ma, T. He, A. Swami, D. Towsley, K. Leung, and J. Lowe, "Node failure localization via network tomography," in *ACM IMC*, 2014.

[39] L. Ma, T. He, A. Swami, D. Towsley, and K. Leung, "On optimal monitor placement for localizing node failures via network tomography," *Elsevier Performance Evaluation*, vol. 91, pp. 16–37, September 2015.

[40] ——, "Network capability in localizing node failures via end-to-end path measurements," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 434–450, February 2017.

[41] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[42] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, "The complexity of multiterminal cuts," *SIAM Journal on Computing(SICOMP)*, vol. 23, 1994.

[43] V. V. Vazirani, *Approximation Algorithm*. Springer, 2001.

[44] G. Clinescu, H. Karloff, and Y. Rabani, "An improved approximation algorithm for multiway cut," *Computer and System Sciences*, vol. 60, pp. 564–574, June 2000.

[45] R. Manokaran, J. S. Naor, P. Raghavendra, and R. Schwartz, "Sdp gaps and ugc hardness for multiway cut, 0-extension and metric labeling," in *ACM STOC*, 2008.

[46] A. Sharma and J. Vondrk, "Multiway cut, pairwise realizable distributions, and descending thresholds," in *ACM STOC*, 2014.

[47] "The Internet Topology Zoo," http://www.topology-zoo.org/dataset.html.

[48] "Center for Applied Internet Data Analysis: Macroscopic Internet Topology Data Kit (ITDK)," http://www.caida.org/data/internet-topology-data-kit/.