Stealthy DGoS Attack against Network Tomography: The Role of Active Measurements

Cho-Chun Chiu, *Student Member, IEEE* and Ting He, *Senior Member, IEEE*

***Abstract*—As a tool to infer the internal state of a network that cannot be measured directly, network tomography has been extensively studied under the assumption that the measurements truthfully reflect the end-to-end performance of measurement paths, which makes the resulting solutions vulnerable to manipulated measurements. In this work, we investigate the impact of manipulated measurements via a recently proposed attack model called the *stealthy DeGrading of Service (DGoS) attack*, which aims at maximally degrading the performance of targeted paths without exposing the manipulated links to network tomography. While existing studies on this attack assumed that network tomography only measures the paths actively used for data transfer (via passive measurements), our model allows network tomography to measure a larger set of paths, e.g., by sending probes on some paths not carrying data flows. By developing and analyzing the optimal attack strategy, we quantify the maximum damage of such an attack. We further develop a defense strategy by formulating and solving a Stackelberg game to select the best set of measurement paths under a budget constraint. Our evaluations on real topologies validate the efficacy of the proposed defense strategy while identifying areas for further improvement.**

***Index Terms*—Network tomography, Degrading of Service attack, combinatorial optimization, integer linear programming, measurement design.**

I. Introduction

Network tomography [2] is a family of inference-based techniques to monitor the internal state (e.g., link delays or loss rates) of a network from external measurements. The need of such techniques arises in many networks where the internal network elements are accessible in the data plane but inaccessible in the control plane, e.g., the public Internet and all-optical networks.

Theoretically, network tomography works by inverting a given observation model that captures the relationship between the unknown link states and the observed path states, where specific solutions differ in the observation models they assume, e.g., a linear model for inferring additive link metrics such as delays [3], [4], [5], a Boolean model for localizing failures [6], [7], or various probabilistic models for accommodating performance fluctuations (see [2] and references therein). However, most of the existing works assumed that the measurements truthfully reflect the performance of measurement paths, leaving open what will happen when measurements can be manipulated by an attacker.

Manipulated measurements fundamentally change the problem of network tomography, because instead of only changing the link states (e.g., by imposing the same delay to all the packets traversing a link), the attacker may manipulate different packets traversing the same link differently (e.g., by adding delays for packets with one source-destination pair but not adding delays for packets with another source-destination pair), thus changing the observation model. For example, a link showing two different behaviors for two groups of flows is effectively two different links, each traversed by one group of flows. The impact of manipulated measurements on network tomography only started to be realized recently [8], [9], under linear observation models, where it was shown that the attacker can substantially degrade path performances while misleading network tomography to consider the manipulated links as well-performing links. However, these studies implicitly assumed that network tomography only collects *passive measurements*, i.e., the performances of data packets, and thus only measures the paths used for data transfer.

In practice, however, network tomography can monitor a larger set of paths via *active measurements* obtained from probes. Intuitively, augmenting passive measurements with active measurements exposes the performances of a larger set of paths and thus should help network tomography to defend against attacks. In this work, we harden this intuition by quantifying the maximum damage a stealthy attacker can inflict on a network monitored by network tomography, and developing a defense strategy by selecting the probing paths to minimize this damage under a budget constraint.

A. Related Work

Network tomography is a rich family of network monitoring techniques that infer network internal characteristics from external measurements [2], [10]. Early works focused on *best-effort solutions*, which tried to find the most likely network state from given measurements, obtained by unicast [11], [12], [13], [14], multicast [15], [16], [17], [18], [19], [20], and their variations (e.g., bicast [21], flexicast [22], and back-to-back unicast [23], [24], [20]). After observing that an arbitrary set of measurements is frequently insufficient for identifying all the link metrics [25], [12], [26], [27], [28], later works focused on either reducing the ambiguity (e.g., by imposing a tie breaker [13], [14], [29], or relaxing the objective [27], [30], [31]), or ensuring identifiability by carefully designing the monitor locations and the measurement paths (e.g., [3], [4], [5], [32], [33], [6], [7]). All these works assume a *benign setting*, where the links behave consistently and the measurements are truthful.

C. Chiu (cuc496@psu.edu) and T. He (tzh58@psu.edu) are with the School of Electrical Engineering and Computer Science, the Pennsylvania State University.

Aspects of this work was presented at Globecom'20 [1].

This work was supported by the National Science Foundation under award CCF-1813219.

Very few works have considered network tomography in an *adversarial setting*. In [34], an algorithm was proposed to detect non-neutral links that discriminate packets on different paths, by detecting the links that cause the network tomography problem to be infeasible. There are also other works on detecting network neutrality violations [36], [37], [38], but these are engineering solutions utilizing information beyond end-to-end performances (e.g., ports and other confounding factors), falling out of the scope of network tomography. Studies on network neutrality differ fundamentally from our work in that they do not consider an intelligent adversary that intentionally controls the non-neutral links to evade detection. In contrast, only the following works considered intelligent attacks against network tomography. In [8] [35], optimizations were formulated to design the manipulations at given compromised nodes in order to cause performance degradation while scapegoating certain benign links as the cause of poor performance; however, the set of compromised nodes is not optimized. In [9], a similar but more sophisticated attack model, called the *stealthy DeGrading of Service (DGoS) attack*, was proposed, where the attacker jointly optimizes where to attack (in terms of compromised links) and how to attack (in terms of the manipulation on each path traversing at least one compromised link). However, [8], [35], [9] all assumed that the total performance degradation includes the degradation on all the measurement paths, which implies that only the paths carrying data flows are monitored by network tomography. Our attack model is most similar to [9], except that we accommodate both passive and active measurement paths for network tomography.

In terms of defense, existing solutions [8], [35] focused on detecting and localizing the attacker-controlled links, under the assumption that the attacker neglects certain measurement paths during attack design (and is hence exposed by these paths). In this work, we will consider a more challenging scenario, where the attacker is aware of all (possible) measurement paths, and thus the designed attack is undetectable by construction. Instead, we propose a proactive defense strategy that designs the measurement paths to minimize the maximum impact of all the undetectable attacks. Table I summarizes the comparison between our work and the above existing works.

B. Summary of Contributions

Our goal is to analyze the impact of DGoS attack and develop defenses in networks monitored by network tomography that employs both passive and active measurements.

1) We extend the attack model in [9] to include both passive measurement paths and active measurement paths, where only the performance degradation on passive measurement paths counts towards the damage caused by an attack.

2) We derive sufficient/necessary conditions for an attack strategy to be optimal under the above attack model. Based on these conditions, we establish the hardness of designing the optimal attack, and develop efficient algorithms by converting our problem to well-known integer programming problems.

3) Based on insights about the attack, we develop a defense strategy by selecting the measurement paths to minimize the maximum damage under a budget constraint. Although the optimal defense is very hard to compute as evaluating the maximum damage under a given set of paths is already NP-hard, we show that the complexity can be reduced by minimizing an upper bound on the maximum damage, for which a mixed-integer indefinite quadratic programming is formulated and a polynomial-time greedy algorithm is proposed.

4) We evaluate the proposed attack and defense strategies on a variety of real Internet topologies. Our evaluations show that: (i) the proposed attack strategies achieve significantly more performance degradation than intuitive alternatives and thus better reveal the potential damage of DGoS attack, (ii) compared to only monitoring passive measurement paths, adding a few active measurement paths selected by the proposed defense strategy can notably reduce the performance degradation, but (iii) if not selected carefully, then many more paths need to be monitored to achieve the same level of protection.

Roadmap. Section II formulates the generalized DGoS attack that accounts for both passive and active measurement paths. Section III analyzes the optimal attack and presents algorithms for attack design. Section IV presents our defense strategy and the associated algorithms. Section V evaluates the proposed attack/defense algorithms on real Internet topologies. Finally, Section VI concludes the paper.

II. Problem Formulation

Table II summarizes the main notations used in this paper.

A. Network Model

We model the network 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 (the smaller, the better). We assume that these link metrics are *additive*, i.e., a path metric equals the sum of its link metrics. This is a canonical assumption satisfied by several important performance metrics including delays, jitters, log-success rates, and their statistics.

We assume that this network is monitored by a tomography-based detection system that measures the end-to-end metrics on a set P of paths to detect anomalies on link metrics. Let $P_d \subseteq P$ denote the passive measurement paths (traversed by data packets) and $P \setminus P_d$ the active measurement paths (traversed by probes). Let $\boldsymbol{R} = (r_{ij})_{p_i \in P, l_j \in L}$ 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 $\boldsymbol{R}$. Given the measured path metrics $\mathbf{y} = (y_i)_{p_i \in P}$, network tomography detects link anomalies by finding a solution $\widehat{\mathbf{x}}$ to $\boldsymbol{R}\widehat{\mathbf{x}} = \mathbf{y}$ and then comparing each inferred link metric $\widehat{x}_j$ with the maximum normal delay τ: l_j is considered "normal" if $\widehat{x}_j \leq \tau$ and "abnormal" otherwise. To focus on anomalies caused by the attack, we assume that the pre-attack link metrics are all normal, i.e., $x_j \leq \tau$ ($\forall l_j \in L$). Let $\tau_{\max}$ denote the maximum possible link

[1] Although [8], [35] mentioned the use of probes, their definition of damage metric implies that only passive measurements are considered.

TABLE I
Comparison with Related Works

paper	intelligent attack	optimized places of attack	active measurements	defense strategy
[34]	no	no	no	detection
[8], [35]	yes	no	no[1]	detection
[9]	yes	yes	no	N/A
this work	yes	yes	yes	mitigation

TABLE II
Notations

L, L_m, L_n	all/compromised/uncompromised links
P, P_c	measurement paths, candidate measurement paths
P_d	passive measurement paths
P_m, P_n	compromised/uncompromised measurement paths
L'_n	uncompromised links only on compromised paths
w_j	traversal number of link $l_j \in L$
$T(L')$	total traversal number of a set of links L'
$\mathcal{C}_{P'}$	all cuts of a set of paths P'
$\mathcal{C}^*_{P'}$	all cuts of P' with minimum total traversal number
$\boldsymbol{R}$, $\mathbf{r}_i$	measurement matrix, i-th row in measurement matrix
$\mathbf{x}$, $\widehat{\mathbf{x}}$	true/inferred link metrics
$\mathbf{y}$	measured path metrics
$\mathbf{z}$	adversarial manipulations of path metrics
$\boldsymbol{c^a}$	costs of compromising links
$\boldsymbol{c^d}$	costs of monitoring paths
k^a, k^d	attack/defense budget
τ	maximum metric of a normal link
$\tau_{\max}$	maximum possible link metric
$\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$	variables of attack optimization (see (14))
$\boldsymbol{\delta}$	variable of defense optimization (see (20))

metric, which is only used to ensure finite objective values and can be arbitrarily large. We assume that $\tau \leq \tau_{\max}$.

Remark: We note that the solution to $\boldsymbol{R}\widehat{\mathbf{x}} = \mathbf{y}$ may not be unique as $\boldsymbol{R}$ may not have a full column rank [25], [12], [26], [27], [28]. In this case, network tomography can pick a solution by optimizing certain objective functions, e.g., minimizing the number of abnormal links [13], [14] or maximizing the posterior probability [29]. We do not assume any specific objective function in this work.

B. Attack Model

Suppose that an attacker wants to degrade the performance of paths in P_d without being localized by network tomography. In this sense, P_d also represents the set of paths targeted by the attacker (e.g., paths to/from certain hosts of interest).

The attack is mounted by first controlling a subset $L_m \subseteq L$ of links and then modifying the path metrics by $\mathbf{z} = (z_i)_{p_i \in P}$ through these links. Let c^a_j ($l_j \in L$) denote the cost of compromising link l_j, and k^a denote the budget of the attacker. We call L_m the *compromised links* and $L_n := L \setminus L_m$ the *uncompromised links*. We call the paths $P_m \subseteq P$ traversing at least one link in L_m the *compromised paths*, and the rest $P_n := P \setminus P_m$ the *uncompromised paths*.

To ensure that the attack is feasible, we adopt the following assumptions from [8], [9]:

1) Only the metrics of compromised paths can be manipulated, i.e., $z_i = 0$ for any $p_i \in P_n$.
2) The manipulation can only degrade (not improve) path performance, i.e., $z_i \geq 0$ for any $p_i \in P_m$.

The first assumption ensures that the attacker can only manipulate the performance of packets traversing at least one of the compromised links, and the second assumption ensures that the manipulation is feasible (e.g., the injected delay is non-negative). Moreover, to stay stealthy, the attacker must ensure that (i) the network tomography problem remains feasible under the manipulation, i.e., $\boldsymbol{R}\widehat{\mathbf{x}} = \boldsymbol{R}\mathbf{x} + \mathbf{z}$ is feasible, and (ii) there exists a feasible solution to $\widehat{\mathbf{x}}$, according to which all the compromised links appear normal.

Remark: When $\boldsymbol{R}$ is rank-deficient, there are multiple feasible solutions to $\widehat{\mathbf{x}}$, and the above conditions (i–ii) provide certain stealthiness in the sense that the detection system cannot say for sure that any of the compromised links are abnormal. If how network tomography resolves ambiguity (e.g., [13], [14], [29]) is known to the attacker, then he can achieve stronger stealthiness by imposing an additional constraint that the desired $\widehat{\mathbf{x}}$ (which indicates all the compromised links as normal) will be the solution selected by network tomography. We leave the investigation under this formulation to future work.

Using a change of variable $\mathbf{z} = \boldsymbol{R}(\widehat{\mathbf{x}} - \mathbf{x})$, we formulate the problem of optimal attack design as follows:

$$\max_{L_m, \widehat{\mathbf{x}}} \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{1a}$$

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

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

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

$$\sum_{l_j \in L_m} c^a_j \leq k^a, \tag{1e}$$

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

where $P_n = \{p_i \in P : \, p_i \cap L_m = \emptyset\}$ and $L_n = L \setminus L_m$ by definition. In words, (1) designs "where to attack" (represented by L_m) and "how to attack" (represented by $\widehat{\mathbf{x}}$) to maximize the total degradation on the paths of interest (1a), subject to feasibility (1b), stealthiness (1c)(1d), and budget constraints (1e). The above formulation generalizes the stealthy DGoS attack proposed in [9, (1)] in that: (i) only degradation on the paths in P_d is included in the objective, which allows us to model both passive and active measurements for network tomography, and (ii) a budget constraint (1e) is added to capture the resource constraint faced by a realistic attacker. As shown later, these differences lead to subtle but critical changes in the solution.

C. Defense Model

As the detection system cannot access individual links (hence the need of network tomography) and the DGoS attack is designed to evade detection, the best defense from the perspective of the detection system is to minimize the damage of such an attack. Specifically, while the passive measurement paths P_d are usually dictated by the needs of data flows, the active measurement paths $P \setminus P_d$ are controlled by network tomography and thus can be designed to mitigate the attack. Similar ideas of designing the measurements have been widely applied to network tomography in the benign setting [3], [4], [5], [32], [33], [6], [7].

Let P_c denote the set of candidate measurement paths (e.g., all the routing paths involving the terminals controlled by network tomography), including the paths in P_d. Suppose that monitoring each path $p_i \in P_c$ incurs a cost of c_i^d, and the defender (i.e., measurement designer) has a budget of k^d. As passive measurements are byproducts of data communications and do not consume extra network resources, we assume that $c_i^d \equiv 0$ for all $p_i \in P_d$, which implies that $P_d \subseteq P$.

We formulate the problem of optimal defense design as the following bilevel optimization:

$$\min_{P \subseteq P_c} \max_{L_m, \widehat{\mathbf{x}}} \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{2a}$$

$$\begin{aligned}
\text{s.t. } & \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) = 0, && \forall p_i \in P_n, && \text{(2b)}\\
& \tau_{\max} \geq \widehat{x}_j \geq 0, && \forall l_j \in L_n, && \text{(2c)}\\
& \tau \geq \widehat{x}_j \geq 0, && \forall l_j \in L_m, && \text{(2d)}\\
& \sum_{l_j \in L_m} c_j^a \leq k^a, && && \text{(2e)}\\
& \sum_{p_i \in P} c_i^d \leq k^d, && && \text{(2f)}\\
& L_m \subseteq L. && && \text{(2g)}
\end{aligned}$$

The above problem is a *Stackelberg game*, where the defender is the leader, and the attacker is the follower. The two players interact via the bilevel optimization (2). At the upper-level, the defender selects the measurement paths P out of P_c to minimize the worst-case performance degradation that can be caused by the attacker, where (2f) captures the budget constraint for the defender. At the lower-level, the attacker designs the parameters L_m and $\widehat{\mathbf{x}}$ of the DGoS attack as in (1) to achieve the maximum damage on P_d while evading detection.

Remark 1: The bilevel optimization implicitly assumes that the action of the defender (i.e., P) is known to the attacker. This can happen if the attacker is able to monitor all the candidate paths (e.g., by intersecting traffic at the gateway router of a targeted organization) to identify the active paths traversed by data packets or probes. Generally, letting $\hat{P}$ denote the attacker's estimate of the paths monitored by network tomography, the assumption of $\hat{P} = P$ leads to a conservative defense strategy, and the actual attack can be non-stealthy (if $P \setminus \hat{P} \neq \emptyset$) or less effective (if $\hat{P} \setminus P \neq \emptyset$). However, as the attacker's knowledge is unknown to the defender at the time of measurement design, this assumption allows us to plan for the worst case.

Remark 2: The defense formulation in (2) aims at mitigating the impact of the worst attack subject to budget and stealthiness constraints. If other attack strategies are used under the designed measurement paths P, they will be non-stealthy, more expensive, or less damaging. The objective value of (2) under a given P represents the maximum performance degradation that can be caused by a stealthy attack within budget k^a if the paths in P are monitored, and the actual performance degradation can only be smaller.

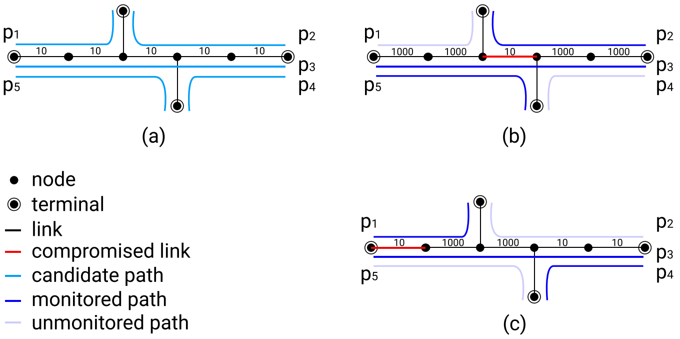


Fig. 1. Example: (a) assume $P_d = \{p_3\}$ and initial delay is 10 ms on each link; (b) under monitored paths $\{p_2, p_3, p_5\}$, the maximum delay degradation on P_d is 3960 ms (link labels indicate $\widehat{\mathbf{x}}$), (c) under monitored paths $\{p_1, p_3, p_4\}$, the maximum delay degradation on P_d is 1980 ms.

D. Motivating Example

For the attacker, while intuitively compromising all the links gives the attacker the most flexibility in manipulating the measurements, and should therefore be the optimal strategy, we have shown that this is not true via a counterexample [9], since compromised links will also impose limitations on how much degradation can be injected on paths due to the stealthiness constraint. For the defender, intuitively monitoring the paths in $P_c \setminus P_d$ that cover more links in $\cup_{p \in P_d} p$, in addition to P_d, would better mitigate the performance degradation injected by the attacker on P_d. However, we will show that this strategy is generally suboptimal. Consider the example in Fig. 1 (a), where $P_d = \{p_3\}$. Suppose that before the attack, each link has a delay of 10 ms, $\tau = 10$ ms, and $\tau_{\max} = 1000$ ms. Fig. 1 (b) shows the optimal attack parameters $(L_m, \widehat{\mathbf{x}})$ under the monitoring of paths $\{p_2, p_3, p_5\}$, increasing the delay on p_3 by 3960 ms. In Fig. 1 (c), under the monitoring of paths $\{p_1, p_3, p_4\}$, the maximum delay degradation on p_3 is only 1980 ms. Even though paths $\{p_2, p_5\}$ cover more links on p_3 than paths $\{p_1, p_4\}$, monitoring $\{p_1, p_4\}$ by active measurements will provide better protection for the data flow on p_3, as these paths provide more fine-grained information for network tomography and hence make it harder for the attacker to inject delays without being localized. This example demonstrates the potential to limit the damage of stealthy DGoS attacks by carefully selecting the (active) measurement paths and the nontrivialness of such selection.

III. Optimal Attack Strategy

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. Meanwhile, optimizing L_m is a combinatorial optimization problem, with an objective $F(L_m)$ that denotes the optimal value of (1a) under a given L_m. The main challenge is that the objective function $F(L_m)$ is not an explicit function of the decision variable L_m. Below, we propose two approaches to turn $F(L_m)$ into an explicit function of L_m, which then lead to efficient algorithms.

A. Attack under Unlimited Budget

First, consider the case that the attacker has an unlimited budget, i.e. the constraint (1e) is removed.

1) Property of the Optimal L_m: In the case of unlimited budget, we will establish sufficient/necessary conditions for a given L_m to be optimal for (1). To this end, we introduce the following definitions.

Definition 1. Given P and P_d, we define:

1) the traversal number $w_j := \sum_{p_i \in P_d} r_{ij}$ for link l_j as the number of paths in P_d that traverse l_j,
2) $T(L') := \sum_{l_j \in L'} w_j$ as the total traversal number of a set of links L',
3) a set of links L' as a cut of a set of paths P' if every $p_i \in P'$ traverses at least one link in L',
4) $\mathcal{C}_{P'}$ as the collection of all the cuts of P', and
5) $\mathcal{C}^*_{P'}$ as the collection of all the cuts of P' with the minimum total traversal number, i.e., $\mathcal{C}^*_{P'} := \{L' \in \mathcal{C}_{P'} | T(L') \le T(L''),\ \forall L'' \in \mathcal{C}_{P'}\}$.

Based on these definitions, we can state the optimality conditions as follows.

Theorem III.1. A set of compromised links L_m is optimal if it is a cut of P with the minimal $T(L_m)$, i.e., $L_m \in \mathcal{C}^*_P$.

Proof. **Step 1.** We claim that if there exists an uncompromised path, then by carefully selecting a link to compromise, the objective value (1a) will increase monotonically. To show this, suppose that under an initial solution $L_m^{(0)}$, there is at least one uncompromised path $p_{i^*} \in P_n^{(0)}$. Then we are going to compromise a link on path p_{i^*}. Given $L_m = L_m^{(0)}$, the optimization in (1) is reduced to:

$$\max \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{3a}$$

$$\text{s.t. } \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) = 0, \qquad \forall p_i \in P_n^{(0)}, \tag{3b}$$

$$\tau_{\max} \ge \widehat{x}_j \ge 0, \qquad \forall l_j \in L_n^{(0)}, \tag{3c}$$

$$\tau \ge \widehat{x}_j \ge 0, \qquad \forall l_j \in L_m^{(0)}. \tag{3d}$$

Let $\widehat{\mathbf{x}}^{(0)}$ be the optimal $\widehat{\mathbf{x}}$ for (3). We observe that there must exist a link $l_{j^*} \in p_{i^*}$ for which $\widehat{x}_{j^*}^{(0)} \le \tau$, as otherwise (i.e., $\widehat{x}_j^{(0)} > \tau$ for all $l_j \in p_{i^*}$), we will have $\mathbf{r}_{i^*}\widehat{\mathbf{x}}^{(0)} > |p_{i^*}|\tau \ge \mathbf{r}_{i^*}\mathbf{x}$, where $|p_{i^*}|$ is the hop count of p_{i^*}. This contradicts with $\mathbf{r}_{i^*}\widehat{\mathbf{x}}^{(0)} = \mathbf{r}_{i^*}\mathbf{x}$ according to constraint (3b). As a result, for the link l_{j^*}, adding a constraint $\widehat{x}_{j^*} \le \tau$ is not going to change the optimal solution.

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

$$\max \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{4a}$$

$$\text{s.t. } \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) = 0, \qquad \forall p_i \in P_n^{(1)}, \tag{4b}$$

$$\tau_{\max} \ge \widehat{x}_j \ge 0, \qquad \forall l_j \in L_n^{(1)}, \tag{4c}$$

$$\tau \ge \widehat{x}_j \ge 0, \qquad \forall l_j \in L_m^{(1)}, \tag{4d}$$

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. 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 (3) with the added constraint $\widehat{x}_{j^*} \le \tau$ remains feasible for (4). Therefore, if $f(\widehat{\mathbf{x}})$ denotes the objective function (4a) and $\widehat{\mathbf{x}}^{(1)}$ is the optimal $\widehat{\mathbf{x}}$ for (4), then $f(\widehat{\mathbf{x}}^{(1)}) \ge f(\widehat{\mathbf{x}}^{(0)})$. This implies that one of the optimal solutions must be a cut in $\mathcal{C}_P$.

Step 2. Next, we are going to show that among all the cuts in $\mathcal{C}_P$, the optimal cut must be the one that minimizes the $T(L_m)$. 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

$$\max \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \tag{5a}$$

$$\text{s.t. } \tau_{\max} \ge \widehat{x}_j \ge 0, \qquad \forall l_j \in L_n, \tag{5b}$$

$$\tau \ge \widehat{x}_j \ge 0, \qquad \forall l_j \in L_m. \tag{5c}$$

It is easy to see that the optimal solution to (5) 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 (5) equals

$$\sum_{p_i \in P_d} (\tau \sum_{l_j \in L_m} r_{ij} + \tau_{\max}(\sum_{l_j \in L} r_{ij} - \sum_{l_j \in L_m} r_{ij})) - \sum_{p_i \in P_d} \mathbf{r}_i \mathbf{x}$$
$$= (\tau - \tau_{\max}) \sum_{p_i \in P_d} \sum_{l_j \in L_m} r_{ij} + \tau_{\max} \sum_{p_i \in P_d} \sum_{l_j \in L} r_{ij} - \sum_{p_i \in P_d} \mathbf{r}_i \mathbf{x}, \tag{6}$$

where only the first term of (6) depends on L_m. As $\tau - \tau_{\max} \le 0$, maximizing (6) is equivalent to minimizing $\sum_{p_i \in P_d} \sum_{l_j \in L_m} r_{ij} = \sum_{l_j \in L_m} w_j$. Thus, all the cuts with the minimum $T(L_m)$ are equally optimal for (1). Since as proved in step 1, one of the optimal solutions must be a cut in $\mathcal{C}_P$, every $L_m \in \mathcal{C}^*_P$ is optimal for (1). □

Remark: Theorem III.1 generalizes [9, Theorem III.1], which states that in the case of $P_d = P$, the minimal traversal cut of P achieves optimality, where the traversal number of a link is defined as the total number of paths in P that traverse it. Theorem III.1 extends this statement to the case of $P_d \subseteq P$ by redefining the traversal number for a link to only count the paths in P_d that traverse this link.

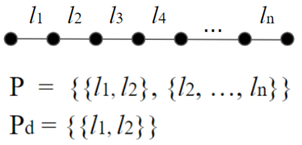


Fig. 2. $L_m \in \mathcal{C}^*_P$ is not necessary for optimality.

While Theorem III.1 gives a sufficient condition to achieve optimality, it does not rule out other possibilities. We show that $L_m \in \mathcal{C}_P^*$ is not always necessary by a simple example. In the example shown in Fig. 2, suppose that $\sum_{i=2}^n x_i \geq \tau_{\max}$. It is easy to see that the optimal solution can be $L_m = \{l_1\}$ ($\widehat{x}_1 = \tau$, $\widehat{x}_2 = \tau_{\max}$) or $L_m = \{l_2\}$ ($\widehat{x}_1 = \tau_{\max}$, $\widehat{x}_2 = \tau$). The optimal solution $\{l_1\} \notin \mathcal{C}_P^*$ shows that $L_m \in \mathcal{C}_P^*$ is not a necessary condition. Generally, it may not be necessary to compromise an active measurement path if its metric is sufficiently large ($\geq \tau_{\max}$). Nevertheless, we will show that compromising all the paths in P_d is necessary under mild conditions.

Theorem III.2. If $\tau > x_j$ ($\forall l_j \in L$), a set of compromised links L_m is optimal only if $L_m \in \mathcal{C}_{P_d}$.

Proof. We prove the claim by contradiction. Assume that an optimal solution to (1) is $\widehat{\mathbf{x}}^{(0)}$ and $L_m^{(0)} \notin \mathcal{C}_{P_d}$. Since $L_m^{(0)} \notin \mathcal{C}_{P_d}$, there must exist an uncompromised path $p_{i^*} \in P_n^{(0)}$ such that $p_{i^*} \in P_d$. We will show that the performance degradation can be strictly increased by compromising p_{i^*}.

Firstly, we claim that there must exist a link $l_{j^*} \in p_{i^*}$ such that $\widehat{x}_{j^*}^{(0)} < \tau$, as otherwise, we will have $\mathbf{r}_{i^*}\widehat{\mathbf{x}}^{(0)} \geq |p_{i^*}|\tau > \mathbf{r}_{i^*}\mathbf{x}$ (because of $\tau > x_j$), which contradicts with $\mathbf{r}_{i^*}\widehat{\mathbf{x}}^{(0)} = \mathbf{r}_{i^*}\mathbf{x}$ according to (1b).

Next, consider another solution where $L_m = L_m^{(1)} := L_m^{(0)} \cup \{l_{j^*}\}$ and $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}'$, defined as

$$\widehat{x}'_j = \begin{cases} \widehat{x}_j^{(0)}, & \text{if } j \neq j^*, \\ \tau, & \text{if } j = j^*. \end{cases} \tag{7}$$

It is easy to verify that this is a feasible solution to (1).

Let $F(L_m, \widehat{\mathbf{x}})$ be the objective value of (1) under solution $(L_m, \widehat{\mathbf{x}})$. Then

$$F(L_m^{(0)}, \widehat{\mathbf{x}}^{(0)}) - F(L_m^{(1)}, \widehat{\mathbf{x}}') \tag{8a}$$

$$= \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}}^{(0)} - \widehat{\mathbf{x}}') \tag{8b}$$

$$= \sum_{p_i \in P_d} r_{ij^*}(\widehat{x}_{j^*}^{(0)} - \tau). \tag{8c}$$

Since $p_{i^*} \in P_d$, $l_{j^*} \in p_{i^*}$ (i.e. $r_{i^*j^*} = 1$), and $\widehat{x}_{j^*}^{(0)} < \tau$, $\sum_{p_i \in P_d} r_{ij^*}(\widehat{x}_{j^*}^{(0)} - \tau) < 0$, i.e., the objective value of solution $(L_m^{(0)}, \widehat{\mathbf{x}}^{(0)})$ can be increased by another solution, which contradicts the assumption that $(L_m^{(0)}, \widehat{\mathbf{x}}^{(0)})$ is optimal. □

Theorems III.1 and III.2 imply the following condition.

Corollary III.3. If $P_d = P$ and $\tau > x_j$ ($\forall l_j \in L$), then a set of compromised links L_m is optimal if and only if $L_m \in \mathcal{C}_P^*$.

Proof. We know from Theorem III.2 that L_m is optimal only if $L_m \in \mathcal{C}_P$ since $P_d = P$. Then from Step 2 in the proof of Theorem III.1, we know that L_m is optimal in $\mathcal{C}_P$ only if it has the minimal $T(L_m)$ among all the cuts in $\mathcal{C}_P$. This together with Theorem III.1 completes the proof. □

2) Hardness and Algorithm Design: Theorem III.1 implies that finding a minimum-traversal cut $L_m \in \mathcal{C}_P^*$ will give an optimal solution to (1). This reduces (1) to the following combinatorial optimization problem.

Algorithm 1: Greedy GALS

```
input : $P$, $P_d$
output: Compromised links $L_m$
1 $P_m \leftarrow \emptyset$;
2 $L_m \leftarrow \emptyset$;
3 $w_j \leftarrow \sum_{p_i \in P_d} r_{ij}$;
4 while $P_m \neq P$ do
5     Find the link $l_j$ with the smallest ratio $\frac{w_j}{|P_j \setminus P_m|}$;
6     $P_m \leftarrow P_m \cup P_j$;
7     $L_m \leftarrow L_m \cup \{l_j\}$;
8 return $L_m$;
```

Definition 2. Given a set of paths P and a subset $P_d \subseteq P$, the *generalized adversarial link selection* (GALS) problem aims at finding the cut of P with the minimum $T(L_m)$:

$$\min_{L_m \in \mathcal{C}_P} T(L_m) = \sum_{l_j \in L_m} w_j. \tag{9}$$

GALS generalizes the *adversarial link selection (ALS)* problem formulated in [9] in that the traversal number w_j only counts the traversals by paths in P_d. Nevertheless, given P_d, the traversal number of each link is a constant, and thus the solutions for ALS and GALS are the same.

Specifically, since ALS is NP-hard [9], GALS is also NP-hard. Moreover, similarly to the reduction of ALS to the *weighted set cover (WSC)* problem [9], GALS can also be reduced to WSC, and can thus leverage existing algorithms designed for WSC. One such algorithm is the greedy algorithm, shown in Algorithm 1. The algorithm iterates until all the paths are compromised (line 4), where in each iteration, it picks a link with the smallest cost-value ratio (line 5) and adds it to the set of compromised links (lines 6–7). Here, we define the cost-value ratio of link l_j by $w_j/|P_j \setminus P_m|$, where P_j is the set of paths traversing link l_j. It is known [39] that this greedy algorithm has the best approximation factor for WSC, which is $\Theta(\log |P|)$ in our case. The while loop (lines 4–7) is repeated $O(|L|)$ times, each iteration taking $O(|L| \cdot |P|)$ time (due to line 5), leading to an overall complexity of $O(|L|^2|P|)$.

B. Attack under Limited Budget

In the general case of $k^a < \infty$, the attacker may not have sufficient budget to compromise the minimum-traversal cut, and thus the optimal strategy needs to be adapted.

1) Property of the Asymptotically Optimal L_m: For a general L_m, it is difficult to write the optimal value of (1a) wrt $\widehat{\mathbf{x}}$ as an explicit function of L_m. Nevertheless, we find the following approximation to be asymptotically accurate.

Definition 3. Given a set of paths P, a subset $P_d \subseteq P$, and the cost c_j^a for each link l_j, the *generalized constrained adversarial link selection* (GCALS) problem aims at:

$$\max_{L_m} T(L'_n) := \sum_{l_j \in L'_n} w_j \tag{10a}$$

$$\text{s.t.} \quad \sum_{l_j \in L_m} c_j^a \leq k^a, \tag{10b}$$

$$L_m \subseteq L, \tag{10c}$$

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

We show that when $\tau_{\max}$ is large, GCALS is asymptotically equivalent to the original optimization (1).

Theorem III.4. As $\tau_{\max} \to \infty$, $L_m = L^*_m$ is optimal for (1) if and only if L^*_m is an optimal solution to GCALS.

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

$$\sum_{p_i \in P_d} \sum_{l_j \in L} r_{ij}(\widehat{x}_j - x_j) = \sum_{l_j \in L} \sum_{p_i \in P_d} r_{ij}(\widehat{x}_j - x_j). \qquad (11)$$

If $l_j \in L_m$, then $\widehat{x}_j \leq \tau$ by (1d). 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,1c). For a large $\tau_{\max}$, $\widehat{x}_j$ can achieve $\tau_{\max}$ if and only if $l_j \in L'_n$. Thus, as $\tau_{\max} \to \infty$, the optimal value of (11) wrt $\widehat{\mathbf{x}}$ is approximately:

$$\tau_{\max} \sum_{l_j \in L'_n} \sum_{p_i \in P_d} r_{ij} = \tau_{\max} \sum_{l_j \in L'_n} w_j \propto T(L'_n) \qquad (12)$$

That is, the optimal objective value of (1) under a given L_m is asymptotically proportional to $T(L'_n)$, which completes the proof. □

2) Hardness and Algorithm Design: First, we will show that GCALS problem is NP-hard.

Theorem III.5. The GCALS problem (10) is NP-hard.

Proof. The idea is to show that GCALS is actually a generalization of GALS, and hence its NP-hardness is implied by the NP-hardness of GALS.

To this end, consider a special case of GCALS, where it is known that it suffices to optimize L_m among the cuts in $\mathcal{C}_P$. If $L_m \in \mathcal{C}_P$, then $L'_n = L_n$, and hence $T(L'_n) = T(L_n)$. As

$$T(L_n) + T(L_m) = T(L), \qquad (13)$$

where the right-hand side is a constant, maximizing $T(L_n)$ is equivalent to minimizing $T(L_m)$, which is the GALS problem. □

Next, we will develop a solution by formulating this problem as an integer linear programming (ILP) problem:

$$\max_{\alpha_j, \beta_j, \gamma_j} \sum_{l_j \in L} \gamma_j w_j \qquad (14a)$$

$$\text{s.t.} \sum_h \alpha_h r_{ih} \geq r_{ij} \beta_j \qquad \forall l_j \in L, \forall p_i \in P, \qquad (14b)$$

$$\sum_{l_j \in L} c^a_j \alpha_j \leq k^a \qquad \forall l_j \in L, \qquad (14c)$$

$$\gamma_j \leq 1 - \alpha_j \qquad \forall l_j \in L, \qquad (14d)$$

$$\gamma_j \leq \beta_j \qquad \forall l_j \in L, \qquad (14e)$$

$$\gamma_j \geq \beta_j - \alpha_j \qquad \forall l_j \in L, \qquad (14f)$$

$$\alpha_j, \beta_j, \gamma_j \in \{0, 1\} \qquad \forall l_j \in L. \qquad (14g)$$

Lemma III.6. The optimization (14) is equivalent to the optimization (10), where $l_j \in L_m$ if and only if $\alpha_j = 1$.

Proof. Let $\alpha_j \in \{0, 1\}$ be the indicator of whether link l_j is compromised:

$$\alpha_j = \begin{cases} 1, & \text{if } l_j \in L_m, \\ 0, & \text{otherwise}, \end{cases} \qquad (15)$$

which is subject to the budget constraint (14c).

Due to constraint (14b), we know that (i) if there is at least one uncompromised path p_i traversing link l_j, i.e., $\exists i$ such that $\sum_h \alpha_h r_{ih} = 0$ and $r_{ij} = 1$, then $\beta_j \leq 0$, and (ii) otherwise, i.e., $\sum_h \alpha_h r_{ih} \geq 1$ for every i such that $r_{ij} = 1$, then $\beta_j \leq 1$. Moreover, constraints (14d–14g) collectively imply that $\gamma_j = \beta_j(1 - \alpha_j)$, i.e., the objective (14a) is to maximize $\sum_{l_j \in L} \beta_j (1 - \alpha_j) w_j$. Since $(1 - \alpha_j) w_j$ is non-negative, the optimal value of β_j should equal its upper bound, i.e.,

$$\beta_j = \begin{cases} 1 & \text{if } l_j \text{ is only traversed by paths in } P_m, \\ 0 & \text{otherwise.} \end{cases} \qquad (16)$$

Therefore, $\gamma_j = \beta_j(1 - \alpha_j) = 1$ if and only if l_j is an uncompromised link that is only traversed by compromised paths, i.e., $l_j \in L'_n$. Thus, the objective (14a) is equivalent to (10a), which completes the proof. □

The ILP formulation (14) allows us to leverage techniques for solving ILP to solve the GCALS problem (10). In particular, one commonly-used approach is to relax the ILP into an LP by relaxing the integer constraint (14g) into $\alpha_j, \beta_j, \gamma_j \in [0, 1]$. After solving this LP relaxation for a fractional solution, we can use various rounding techniques to convert it into a feasible solution to the original problem. A rounding scheme we find to be particularly effective is as follows. For a link l_j, we define the value-cost ratio as $\frac{\alpha'_j |P_j \setminus P_m|}{c^a_j}$, where P_j is the set of paths traversing link l_j and α'_j is the fractional solution of α_j from the LP relaxation. We then iteratively select links into L_m until reaching the budget, where in each iteration, we select the link l_j with the largest value-cost ratio. We refer to this algorithm as "LP relaxation with rounding (LP-R)", for which the pseudo code is given in Algorithm 2. It remains open whether there exists a polynomial-time algorithm with approximation guarantee for GCALS.

The while loop (lines 5–10) is repeated $O(|L|)$ times, and each iteration takes $O(|L||P|)$ time. Thus the while loop takes $O(|L|^2|P|)$ time. Before the while loop, the LP (line 4) needs to be solved. In most cases, the time solving the LP dominates the overall time complexity. Therefore, the time complexity of Algorithm 2 is equivalent to solving a LP problem. For example, its complexity will be $O(|L|^4(|P| + |L|)^{2.5})$ if using Vaidya's algorithm [40] to solve the LP.

IV. Optimal Defense Strategy

Armed with explicit characterizations of the optimal attack strategy, we now solve the defender's problem (2).

A. Defense under Unlimited Budget

Intuitively, monitoring more paths will increase the observability of network tomography and hence reduce the damage caused by undetectable attacks. We confirm this intuition in the following lemma.

Algorithm 2: LP relaxation with Rounding (LP-R)

```
input : $P, P_d, k^a$
output: Compromised links $L_m$
$L_c \leftarrow L \setminus \{l_j \in L | c^a_j > k^a\}$;
// candidate links
$L_m \leftarrow \emptyset$;
$P_m \leftarrow \emptyset$;
$(\alpha'_j, \beta'_j, \gamma'_j)_{l_j \in L} \leftarrow$ solving the LP relaxation of (14);
while $P_m \subset P$ and $L_c \neq \emptyset$ do
    Find the link $l_j \in L_c$ with the largest ratio $\frac{\alpha'_j |P_j \setminus P_m|}{c^a_j}$;
    $k^a \leftarrow k^a - c^a_j$;
    $L_c \leftarrow (L_c \setminus \{l_j\}) \setminus \{l_{j'} \in L_c | c^a_{j'} > k^a\}$;
    $L_m \leftarrow L_m \cup \{l_j\}$;
    $P_m \leftarrow P_m \cup P_j$;
return $L_m$;
```

Lemma IV.1. Given any measurement paths P, monitoring one more path can only decrease the maximum damage in (2a).

Proof. We prove the lemma by arguing that removing (i.e., not monitoring) a measurement path can only increase the maximum damage. Let $(L_m^{(0)}, \widehat{\mathbf{x}}^{(0)})$ be the optimal attack design when the set of measurement paths is P, achieving damage $d(P)$. After removing $p \in P$ from the measurement paths, the attacker's problem remains the same, except that the constraint (2b) corresponding to p (if p is uncompromised) is removed. This means that $(L_m^{(0)}, \widehat{\mathbf{x}}^{(0)})$ remains a feasible solution to the attacker's problem, and thus the maximum damage under measurement paths $P \setminus \{p\}$ is no smaller than $d(P)$. □

According to Lemma IV.1, if k^d is unlimited, then we should simply monitor all the candidate paths, which minimizes the maximum damage that can be caused by the attacker.

B. Defense under Limited Budget

Now we focus on the more general and practical case when the defender has a limited budget $k^d < \sum_{p_i \in P_c} c^d_i$, which requires a careful selection of which candidate paths to monitor.

1) Hardness: We start by establishing the hardness of this problem.

Theorem IV.2. The defense optimization (2) is NP-hard.

Proof. We first prove that the decision version of the attack optimization (1) is NP-hard by a reduction from ALS [9], which aims at finding a cut of P with the minimum total traversal number by P. Given an instance of ALS, we can construct its corresponding attack optimization problem by setting $P_d = P$, $\tau_{\max} = 2$, $\tau = 1$, and $x_j = 0, \forall l_j \in L$. The objective value for the constructed problem is

$$\max \sum_{p_i \in P_d} \mathbf{r}_i(\widehat{\mathbf{x}} - \mathbf{x}) \leq 2 \sum_{p_i \in P_d} \|\mathbf{r}_i\|_1 \leq 2|P||L|. \tag{17}$$

Moreover, the optimal objective value will be an integer, since under constraints (1b) (1c) (1d), the optimal value of $\widehat{x}_j$ would be the following:

$$\widehat{x}_j = \begin{cases} 0, & \text{if } l_j \in \cup_{p \in P_n} p, \text{ due to (1b)}, \\ 1, & \text{if } l_j \in L_m, \text{ due to (1d)}, \\ 2, & \text{otherwise, due to (1c)}. \end{cases} \tag{18}$$

As a result, the optimal objective value d^* of the constructed attack optimization is an integer in $[0, 2|P||L|]$. It can be computed in $O(log(|P||L|))$ time by a binary search based on a solution to the related decision problem: is d^* smaller than d? Since $\tau = 1 > 0 = x_j$, according to Corollary III.3, the optimal L_m^* for the constructed problem is the optimal solution to the corresponding instance of ALS. Thus, the optimal objective value OPT^{ALS} of this instance of ALS can be derived from d^* according to (6):

$$OPT^{ALS} = 2 \sum_{p_i \in P_d} \sum_{l_j \in L} r_{ij} - d^*. \tag{19}$$

Since the time complexity of this reduction is polynomial and solving OPT^{ALS} is NP-hard [9], the decision version of the attack optimization (1) is NP-hard.

Next, consider the decision problem related to the defense optimization (2): is there a design of P under which the maximum damage is smaller than d? Given a *certificate* $P^* \subseteq P_c$, the *verifier* must solve the decision version of the attack optimization for $P = P^*$, which is NP-hard. In other words, the decision problem of the defense optimization can not be verified in polynomial time, unless $P = NP$. Since the related decision problem is not in NP (unless $P = NP$), the defense optimization (2) is not in NP [41], and is thus NP-hard. □

2) Algorithm Design: By Theorem III.4, when $\tau_{\max}$ is large, GCALS is asymptotically equivalent to the attack optimization (1). Although both problems are NP-hard, GCALS has an ILP formulation (14), which can be relaxed into an LP to be solved efficiently. Therefore, we propose to use the LP relaxation of GCALS as a proxy of the attack optimization (1) to guide the defense optimization.

Based on this idea, we reformulate the defense optimization (2) by replacing the lower-level optimization with the LP relaxation of (14):

$$\min_{\boldsymbol{\delta}} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}} \boldsymbol{\gamma}^T \boldsymbol{w} \tag{20a}$$

$$\text{s.t. } 1 - \delta_i + \mathbf{r}_i \boldsymbol{\alpha} \geq r_{ij} \beta_j \qquad \forall l_j \in L, \forall p_i \in P_c, \tag{20b}$$

$$\boldsymbol{\gamma} \leq \mathbf{1}^{|L|} - \boldsymbol{\alpha}, \tag{20c}$$

$$\boldsymbol{\gamma} \leq \boldsymbol{\beta}, \tag{20d}$$

$$\boldsymbol{\gamma} \geq \boldsymbol{\beta} - \boldsymbol{\alpha}, \tag{20e}$$

$$(\boldsymbol{c^a})^T \boldsymbol{\alpha} \leq k^a, \tag{20f}$$

$$(\boldsymbol{c^d})^T \boldsymbol{\delta} \leq k^d, \tag{20g}$$

$$\mathbf{0}^{|L|} \leq \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \leq \mathbf{1}^{|L|}, \tag{20h}$$

$$\boldsymbol{\delta} \in \{0, 1\}^{|P_c|}, \tag{20i}$$

where except for $\mathbf{r}_i$, all the vectors are column vectors, $\mathbf{0}^{|L|}$ and $\mathbf{1}^{|L|}$ denote $|L|$-dimensional vectors of all 0's or 1's, and all the inequalities are element-wise inequalities. Here $\boldsymbol{\delta} := (\delta_i)_{p_i \in P_c}$ is the defense decision variable, where

$$\delta_i = \begin{cases} 1, & \text{if } p_i \in P, \text{ i.e., } p_i \text{ is monitored}, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

and $\boldsymbol{\alpha} := (\alpha_j)_{l_j \in L}$, $\boldsymbol{\beta} := (\beta_j)_{l_j \in L}$, $\boldsymbol{\gamma} := (\gamma_j)_{l_j \in L}$ are attack decision variables as in (14). Other than introducing the upper-level optimization, a minor difference between (20) and (14)

Algorithm 3: Greedy Defense

```
input : $P_c$, $P_d$, $k^d$, $\boldsymbol{c}^d$, $k^a$, $\boldsymbol{c}^a$
output: paths to monitor $P$
$P \leftarrow P_d$;
$P_c \leftarrow (P_c \setminus P_d) \setminus \{p_i \in P_c | c^d_i > k^d\}$;
while $P_c \neq \emptyset$ do
    Find a path $p_i \in P_c$ with the largest $\frac{V_P(p_i;\, P_d, \boldsymbol{c}^a, k^a)}{c^d_i}$;
    $P \leftarrow P \cup \{p_i\}$;
    $k^d \leftarrow k^d - c^d_i$;
    $P_c \leftarrow (P_c \setminus \{p_i\}) \setminus \{p_{i'} \in P_c | c^d_{i'} > k^d\}$;
return $P$;
```

is the addition of $(1-\delta_i)$ to (20b), which allows β_j to be 1 as long as link l_j is only traversed by compromised paths or unmonitored paths. In other words, the links only traversed by compromised paths or unmonitored paths will be used to explain the cause of performance degradation. It is easy to verify that for any given $\boldsymbol{\delta}$, (20) is the same as (14), except that (20h) relaxes the corresponding integer constraint (14g).

Relationship between (20) *and* (2)*:* The relaxation of integer constraints in (20h) helps to reduce the computation complexity. Specifically, the defense optimization (2) is not in NP, but (20) is in NP, since given a certificate P^*, its decision problem can be verified by solving as an LP. On the other hand, this relaxation changes the defense objective from minimizing the maximum damage to minimizing an upper bound on the maximum damage. Empirically, we find that even though the gap between the upper bound and the actual maximum damage is not negligible, the trend is preserved: minimizing the upper bound tends to minimize the maximum damage.

Greedy heuristic: The new formulation (20) enables us to apply the greedy heuristic to obtain a (possibly suboptimal) solution in polynomial time. Given a set of measurement paths P, define $D(P;\, P_d, \boldsymbol{c}^a, k^a)$ as the optimal objective value of (20) under $\boldsymbol{\delta}$ defined as in (21) (recall from Definition 1 that P_d determines the traversal numbers $\boldsymbol{w} := (w_j)_{l_j \in L}$). Given $p_i \notin P$, define $V_P(p_i;\, P_d, \boldsymbol{c}^a, k^a)$ as the decrease of this objective value by monitoring one more path p_i, i.e., $V_P(p_i;\, P_d, \boldsymbol{c}^a, k^a) := D(P;\, P_d, \boldsymbol{c}^a, k^a) - D(P \cup \{p_i\};\, P_d, \boldsymbol{c}^a, k^a)$. Algorithm 3 iteratively selects paths into P such that in each iteration, the selected path maximizes $\frac{V_P(p_i;\, P_d, \boldsymbol{c}^a, k^a)}{c^d_i}$, i.e., the reduction of the (upper bound on the) maximum damage per unit cost. The iteration continues until the defense budget k^d is exhausted.

Since $D(P;\, P_d, \boldsymbol{c}^a, k^a)$ can be evaluated by an LP solver, $O(|P_c|)$ paths are examined in each iteration, and there are at most $O(|P_c|)$ iterations, the time complexity of Algorithm 3 is $O(\Gamma |P_c|^2)$, where Γ is the time for solving the LP for $D(P;\, P_d, \boldsymbol{c}^a, k^a)$, e.g., $\Gamma = O(|L|^{4.5} |P_c|^{2.5})$ if using Vaidya's algorithm [40][2]. Although this complexity is a high-order polynomial in $|L|$ and $|P_c|$, we argue that it is acceptable in practice as the algorithm is only run offline. Moreover, while not theoretically guaranteed, we empirically find Algorithm 3 to be near-optimal for (20).

Exact solution: To solve (20) exactly, we convert the bi-level optimization into a single-level optimization, which can then be solved numerically for small problem instances.

Theorem IV.3. The optimization (20) is equivalent to the following optimization problem:

$$\min_{\boldsymbol{\delta}, \boldsymbol{b}} \boldsymbol{a}^T \boldsymbol{b} \tag{22a}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{b} \geq \boldsymbol{d}, \tag{22b}$$

$$(\boldsymbol{c^d})^T \boldsymbol{\delta} \leq k^d, \tag{22c}$$

$$\boldsymbol{\delta} \in \{0,1\}^{|P_c|}, \tag{22d}$$

$$\boldsymbol{b} \geq \boldsymbol{0}^{1+|L|(|P_c|+6)}, \tag{22e}$$

where $\boldsymbol{a}$ and $\boldsymbol{d}$ are defined as follows:

$$\boldsymbol{a} = \begin{bmatrix} k^a \\ \boldsymbol{1}-\boldsymbol{\delta} \\ \vdots \\ \boldsymbol{1}-\boldsymbol{\delta} \\ \boldsymbol{1} \\ \boldsymbol{0} \end{bmatrix} \begin{array}{l} \\ \} \, 1^{\text{st}} \ |P_c| \text{ rows} \\ \\ \} \, |L|^{\text{th}} \ |P_c| \text{ rows} \\ \} \, 4|L| \text{ rows} \\ \} \, 2|L| \text{ rows} \end{array}, \tag{23}$$

$$\boldsymbol{d} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{w} \end{bmatrix} \begin{array}{l} \} \, |L| \text{ rows} \\ \} \, |L| \text{ rows} \\ \} \, |L| \text{ rows} \end{array}. \tag{24}$$

The coefficient matrix $\boldsymbol{A}$ is defined in (26), where $\boldsymbol{R} = (r_{ij})_{p_i \in P_c, l_j \in L}$ is the matrix representation of P_c as defined in Section II-A, $\boldsymbol{I} \in \mathbb{R}^{|L| \times |L|}$ is the identity matrix, and $\boldsymbol{M}_j \in \mathbb{R}^{|L| \times |L|}$ $(j = 1, \ldots, |L|)$ is zero everywhere except that the j-th diagonal entry is 1.

Proof. The idea is to take the dual of the lower-level maximization problem in (20), which yields:

$$\min_{\boldsymbol{b}} \boldsymbol{a}^T \boldsymbol{b} \tag{25a}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{b} \geq \boldsymbol{d}, \tag{25b}$$

$$(\boldsymbol{c^d})^T \boldsymbol{\delta} \leq k^d, \tag{25c}$$

$$\boldsymbol{\delta} \in \{0,1\}^{|P_c|}, \tag{25d}$$

$$\boldsymbol{b} \geq \boldsymbol{0}^{1+|L|(|P_c|+6)}, \tag{25e}$$

where $\boldsymbol{b}$ (a column vector) is the dual variable, $\boldsymbol{\delta}$ is a given solution to the upper-level optimization problem, and the other parameters are defined as in (22). Since the lower-level optimization problem is an LP, its dual problem, which is a minimization, has the same optimal objective value due to the strong duality of LP. Once we transform the lower-level maximization problem into a minimization problem (25), the original minimax problem (20) becomes a minimization problem (22). □

Let $f := \boldsymbol{a}^T \boldsymbol{b}$ denote the objective function of (22). Since f is twice differentiable, the Hessian $\nabla^2 f$ can be evaluated. As f is quadratic in the variables $\boldsymbol{\delta}$ and $\boldsymbol{b}$, $\nabla^2 f \neq \boldsymbol{0}$. However, f is only linear in individual variables, and thus $\frac{\partial^2}{\partial \delta_i^2} f = 0$ for each δ_i and $\frac{\partial^2}{\partial b_j^2} f = 0$ for each b_j, i.e., all the diagonal elements of $\nabla^2 f$ are zero. This implies that the eigenvalues of

[2] This algorithm has a worst-case complexity of $O((n+m)^{1.5} nB)$ for an LP with n variables, m constraints, and B input bits. In our case, $n = O(|L|)$, $m = O(|L||P_c|)$, and $B = O(|L|^2 |P_c|)$.

$$\boldsymbol{A} = \left[\begin{array}{cccccccccc} \boldsymbol{c^a} & -\boldsymbol{R}^T & \cdots & -\boldsymbol{R}^T & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{M}_1\boldsymbol{R}^T & \cdots & \boldsymbol{M}_{|L|}\boldsymbol{R}^T & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{0} & \underbrace{\boldsymbol{0}}_{|P_c|\text{ columns}} & \cdots & \boldsymbol{0} & \underbrace{\boldsymbol{0}}_{|L|\text{ columns}} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & -\boldsymbol{I} \end{array}\right] \begin{array}{l} \}\,|L| \text{ rows} \\ \}\,|L| \text{ rows} \\ \}\,|L| \text{ rows} \end{array} \tag{26}$$

TABLE III
PARAMETERS OF ISP TOPOLOGIES

Network	#nodes	#links	#candidate terminals[4]
Bics	33	48	16
BTN	53	65	25
Colt	153	191	45
Cogent	197	245	21
AS 20965	968	8283	75
AS 8717	1778	3755	1075

$\nabla^2 f$ sum to zero. However, the eigenvalues cannot be all zero, as otherwise the eigen decomposition $\nabla^2 f = Q^T \Lambda Q = \boldsymbol{0}$, contradicting with $\nabla^2 f \neq \boldsymbol{0}$. Thus, $\nabla^2 f$ must have at least one negative eigenvalue and at least one positive eigenvalue. That is, f is non-convex in $\boldsymbol{\delta}$ and $\boldsymbol{b}$, and (22) is a mixed-integer indefinite quadratic programming (MIIQP) problem. MIIQP is generally NP-hard, but can be solved for small instances (e.g., by the branch-and-bound algorithm [42]).

V. Performance Evaluation

To understand the potential damage of the generalized DGoS attack and validate the efficacy of the proposed defense, we evaluate the proposed algorithms as well as their benchmarks on real Internet topologies. We implement the algorithms in Python[3], and solve the numerical optimization problems by the commercial optimizer Gurobi.

A. Experiment Setup

1) Network Topology: We use real Internet topologies from public datasets, whose parameters are shown in Table III. The first four topologies are Point of Presence (PoP) level topologies from the Internet Topology Zoo [43], and the last two topologies are router-level topologies from the CAIDA project [44].

2) Candidate Measurement Paths: For each topology, we select a given number of terminals[5] from low-degree nodes (degree ≤ 2), and compute P_c as the set of shortest paths (in hop count) between all pairs of terminals, with ties broken arbitrarily. We then randomly select a subset of paths in P_c as P_d, i.e., the paths carrying active data flows.

3) Other Parameters: Before the attack, each link has a delay sampled from the interval of $[0, 20)$ (ms) uniformly at random. The cost of compromising each link is drawn uniformly at random from the interval of $[0, 2)$, where a lower cost indicates a more vulnerable link (e.g., built by an older technology or managed by a less trusted provider) and vice versa. The cost of monitoring a path in $P_c \setminus P_d$ is chosen randomly between 1 and 2, modeling the cost of recruiting the endpoints to participate in active measurements. Note that these costs are relative to the attack/defense budget and are thus unitless. A link is considered "normal" if its delay is within 150 ms, i.e., $\tau = 150$. The maximum delay of a link is set to 2000 ms, i.e., $\tau_{\max} = 2000$.

4) Benchmarks for Attack: We compare the proposed attack design algorithms, Greedy GALS (Algorithm 1) and LP-R (Algorithm 2), with three heuristics and an optimal solution:

i) "Random selection" ('random'): This algorithm randomly selects compromised links within the given budget.

ii) "Top traversal" ('top traversal'): Based on the intuition that compromising the most traversed links will give the attacker the most control, this algorithm sorts the links by their traversal numbers in descending order, and then selects compromised links in this order within the budget.

iii) "LP relaxation with Randomized Rounding" ('LP-RR'): To benchmark the proposed rounding scheme in Algorithm 2, we evaluate a randomized rounding scheme, where the fractional solution $(\alpha'_j)_{l_j \in L}$ to the LP relaxation of (14) is used as probabilities for selecting links.

iv) "ILP" ('ILP'): This solution directly solves the ILP (14) by the Gurobi optimizer, which performs an exhaustive search in the worst case.

Under each selection of the compromised links L_m, we solve the optimization (1) in $\widehat{\mathbf{x}}$ to compute the total performance degradation under the optimal manipulations, measured by the total delay injected by the attacker over all the paths in P_d. Because of this, all these benchmarks can be considered instances of an improved version of the "maximum-damage scapegoating attack" in [8], [35] that aims at injecting the maximum degradation on the paths in P_d, with L_m chosen by the above methods.

5) Benchmarks for Defense: We compare the proposed defense design algorithm, Greedy Defense (Algorithm 3), with a baseline and an "optimal" solution:

i) "Random selection" ('random'): This algorithm randomly selects additional paths (besides P_d) to monitor within the defense budget.

ii) "Maximum Coverage selection" ('max cover'): According to the intuition that monitoring the paths traversing more links used by P_d will provide more protection, this algorithm selects the path p_i that maximizes $|cover(p_i)|/c_i^d$ in each iteration until exhausting the defense budget, where $cover(p_i) := \{l_j \in p_i \cap (\cup_{p \in P_d} p) \setminus (\cup_{p \in P_s} p)\}$ and P_s is the set of paths that are already selected.

iii) MIIQP: This algorithm uses the Gurobi optimizer to directly solve (22), which minimizes an upper bound on the maximum damage given by the LP relaxation of (14).

Under each design of the measurement paths P, we solve the attack optimization (1) optimally (by first computing the

[3] Code: https://github.com/cuc496/Analysis-of-Active-measurement.

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

[5] The number of terminals is our evaluation is limited due to its impact on the complexity of Algorithm 3, which is a high-order polynomial in $|P_c|$ that is in turn quadratic in the number of terminals.

TABLE IV
PARAMETERS FOR ATTACKS UNDER VARIED k^a AND UNLIMITED k^d

Network	#terminals	$\lvert P_d \rvert$	$\lvert P \rvert$
Bics	15	10	105
BTN	15	10	105
Cogent	15	10	105
Colt	20	10	190
AS 8717	20	10	190
AS 20965	20	10	190

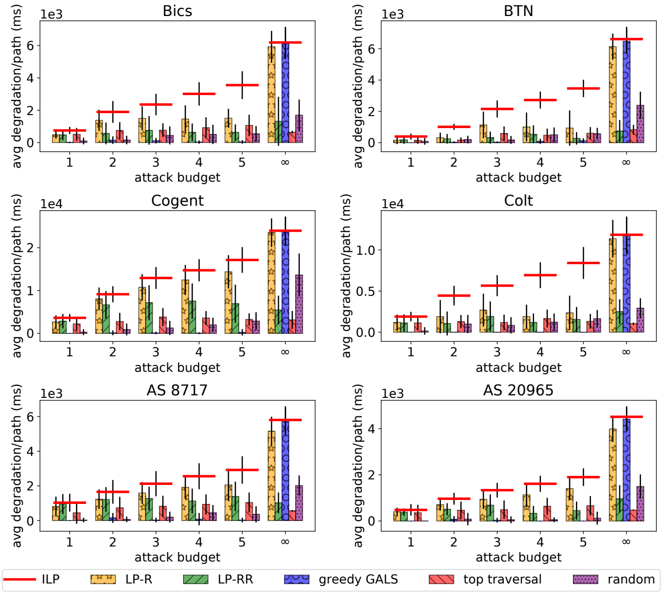


Fig. 3. Comparison of attack strategies when attack budget k^a varies.

optimal L_m by solving the ILP (14) and then solving the remaining LP in $\widehat{\mathbf{x}}$) to compute the maximum performance degradation under P. Due to the high complexity of the defense optimization (which is not even in NP), the optimal defense strategy cannot be computed in reasonable time even for small problem instances, and is hence skipped.

B. Experiment Results on Attack

To evaluate the potential damage of DGoS attack, we evaluate the average performance degradation over the paths in P_d (plus/minus one standard deviation) under each attack design, computed over 20 Monte Carlo runs.

First, we increase the attack budget k^a under the parameters in Table IV to evaluate attackers of growing strength. To understand the fundamental impact of DGoS attack, we set $P = P_c$ in this experiment, which corresponds to the case of unbudgeted defense according to Lemma IV.1. Fig. 3 shows that the proposed attack design algorithm for the budgeted case (LP-R) achieves much bigger damage than the benchmarks for a wide range of k^a, whereas the proposed algorithm for the unbudgeted case (Greedy GALS) is only effective when k^a is large. Moreover, the attacker can cause significant damage by compromising only a few links (e.g., causing > 1 second of delay per path in AS8717 when compromising an average of 2 links at attack budget 2). Note that LP-R is sometimes non-monotone in k^a (e.g., for Colt), because it is generally suboptimal and hence may not utilize the budget optimally (recall that designing the optimal attack strategy is NP-hard according to Theorem III.5).

TABLE V
PARAMETERS FOR ATTACKS UNDER VARIED $\lvert P \rvert$

Network	#terminals	attack budget	$\lvert P_d \rvert$
Bics	15	2	10
BTN	15	2	10
Cogent	15	3	10
Colt	20	3	10
AS 8717	20	3	10
AS 20965	20	3	10

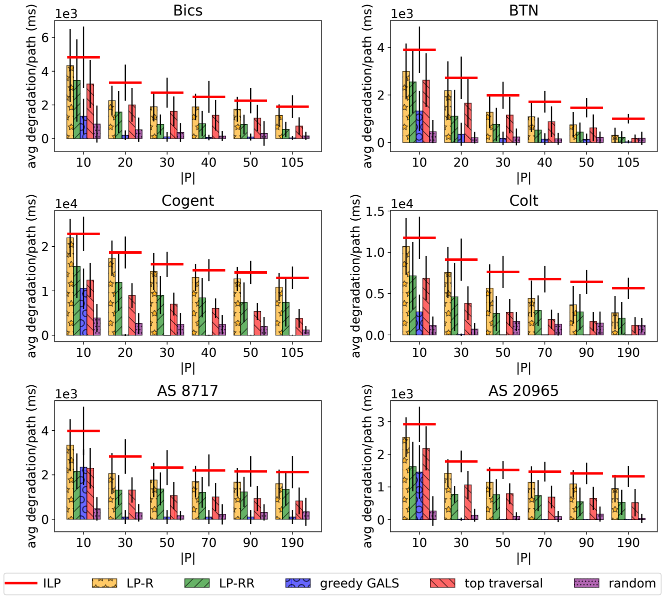


Fig. 4. Comparison of attack strategies when #measurement paths $\lvert P \rvert$ varies.

Next, we fix k^a and $\lvert P_d \rvert$ but increase $\lvert P \rvert$ (and hence the number of active measurement paths in $P \setminus P_d$) under the parameters in Table V to evaluate the impact of monitoring more paths. To understand the protection provided by simply monitoring more paths (in addition to P_d), we randomly select the active measurement paths from $P_c \setminus P_d$ in this experiment; the results under more sophisticated measurement design will be shown in Section V-C. Fig. 4 shows that the damage achieved by all attack strategies decreases with the increase of $\lvert P \rvert$, verifying the intuition that monitoring more paths makes network tomography more effective at thwarting attacks.

C. Experiment Results on Defense

To evaluate the protection provided by carefully designing the (active) measurement paths, we evaluate the average performance degradation over P_d (plus/minus one standard deviation) under each design of P and the corresponding optimal attack strategy, computed over 20 Monte Carlo runs.

First, we fix the attack budget k^a as in Table V and gradually increase the defense budget k^d. Fig. 5 confirms that monitoring a larger set of paths by active measurements helps to protect data flows from stealthy DGoS attacks regardless of how the active measurement paths are selected. However, by carefully selecting these paths, we can achieve the same level

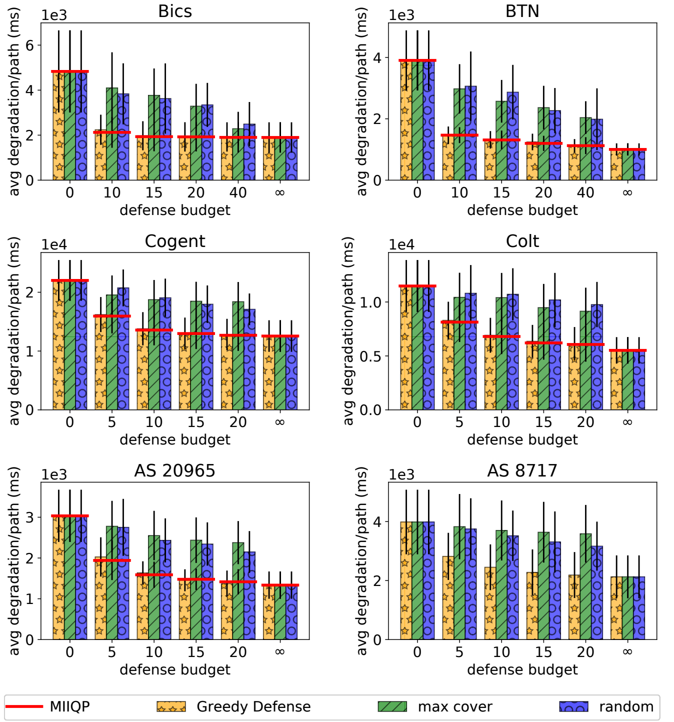


Fig. 5. Comparison of defense strategies when defense budget varies.

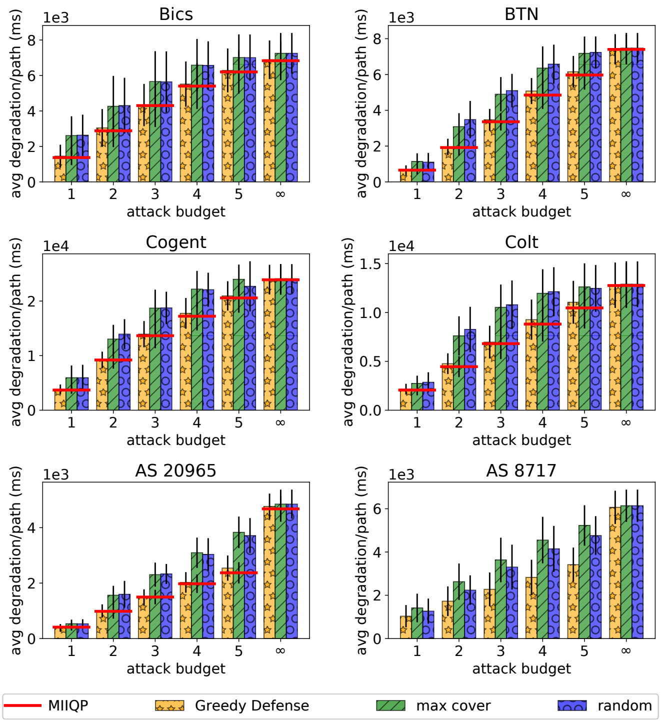


Fig. 6. Comparison of defense strategies when attack budget varies.

TABLE VI
PARAMETERS FOR ATTACKS UNDER VARIED k^a AND FIXED k^d

Network	#terminals	$\lvert P_d \rvert$	$\lvert P_c \rvert$	k^d
Bics	15	10	105	5
BTN	15	10	105	5
Cogent	15	10	105	10
Colt	20	10	190	10
AS 8717	20	10	190	15
AS 20965	20	10	190	15

of protection at a fraction of cost. For example, the proposed solution (Greedy Defense) achieves almost the same protection as monitoring all the 95-180 candidate active measurement paths by only monitoring 10 paths. Moreover, Greedy Defense performs as well as MIIQP while being more computationally efficient (MIIQP is skipped for AS 8717 due to its high complexity, which is exponential in the worst case).

Next, we vary the attack budget k^a while fixing the defense budget k^d as in Table VI. Fig. 6 confirms the efficacy of Greedy Defense when compared to randomly selecting the active measurement paths (random) and selecting the active measurement paths to cover the most links carrying data flows (max cover), which are in turn better than not performing active measurements at all as shown in Fig. 5. In particular, the intuitive heuristic of max cover performs as poorly as random selection, as it fails to model a strategic attacker as done by Greedy Defense. The cost of using Greedy Defense is its computational complexity, e.g., at $k^d = 10$, running this algorithm on average takes 56.85 seconds for BTN, 61.15 seconds for Bics, 1315.7 seconds for Cogent, 808.6 seconds for Colt, 1067.35 seconds for AS8717, 1310.5 seconds for AS20965. Nevertheless, we argue that this cost can be worthwhile in practice as the computation will occur offline. Meanwhile, once the attacker has an unlimited budget, defenses based on designing the measurement paths are no longer effective as the attacker can compromise every path, and other defenses are needed.

Summary: The above results provide a number of insights about DGoS attacks and their defenses: (i) it is important to model intelligent attack strategies as they can achieve substantially more damage than simplistic strategies; (ii) while monitoring more paths always helps the defender, carefully selecting the measurement paths can achieve the same protection at a much lower cost; (iii) even under optimized design of measurement paths, the attacker can still cause significant damage by manipulating internal network elements, signaling the importance of securing network elements as a first line of defense. We note, however, that the third conclusion is drawn under the limitation that measurement paths must start/end at terminals and follow the shortest paths, and it remains open how much protection can be achieved by deploying dedicated monitors and/or controlling the routing of probes, which has been used to ensure identifiability for network tomography in the benign setting [3], [4], [5], [32], [33], [6], [7]. We leave further investigation of this idea to future work.

VI. CONCLUSION

By formulating and analyzing a generalized DGoS attack, we quantified the maximum damage that an attacker inside the network can inflict on end-to-end communications without exposing the compromised links to network tomography, while modeling both passive and active measurements. By establishing optimality conditions, we connected the optimal attack design problem to well-known optimization problems and developed efficient algorithms. We further developed a polynomial-time defense algorithm by formulating and solving a Stackelberg game to optimize the measurement paths in the presence of a limited budget and an intelligent attacker. Our evaluations on real network topologies highlighted the importance of modeling intelligent attackers, and validated the efficacy of the proposed defense. Meanwhile, our results also showed that monitoring the default routing paths between communicating terminals may not be sufficient, indicating the need for the development of further defenses.

References

[1] C. Chiu and T. He, "Stealthy DGoS attack under passive and active measurements," in *IEEE Globecom*, 2020.

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

[3] 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.

[4] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "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.

[5] 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.

[6] 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.

[7] ——, "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.

[8] 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.

[9] C. Chiu and T. He, "Stealthy DGoS attack: Degrading of service under the watch of network tomography," in *IEEE INFOCOM*, 2020.

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

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

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

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

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

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

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

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

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

[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] 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.

[33] 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.

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

[35] S. Zhao, Z. Lu, and C. Wang, "Measurement integrity attacks against network tomography: Feasibility and defense," December 2019.

[36] R. Beverly, S. Bauer, and A. Berger, "The Internet is not a big truck: Toward quantifying network neutrality," in *Springer PAM*, 2007.

[37] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting network neutrality violations with causal inference," in *ACM CoNEXT*, 2009.

[38] U. Weinsberg, A. Soule, and L. Massoulie, "Inferring traffic shaping and policy parameters using end host measurements," in *IEEE INFOCOM*, 2011.

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

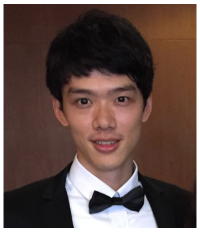
[40] P. M. Vaidya, "Speeding-up linear programming using fast matrix multiplication," in *IEEE FOCS*, 1989.

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* MIT Press, 2009.

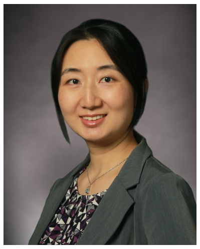
[42] J. Clausen, "Branch and bound algorithms—principles and examples," Technical Report, University of Copenhagen, 1999.

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

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



Cho-Chun Chiu (S'20) received the B.S. degree in Space Science and Engineering from National Central University in 2009 and M.S. in Mechanical Engineering from National Taiwan University in 2011. He is a Ph.D. student in Computer Science and Engineering at the Pennsylvania State University, advised by Prof. Ting He. His research interest includes computer networking, network security, differential privacy, and federated learning.



Ting He (SM'13) received the Ph.D. degree in electrical and computer engineering from Cornell University in 2007. Dr. He is an Associate Professor in the School of Electrical Engineering and Computer Science at Pennsylvania State University, University Park, PA. Her work is in the broad areas of computer networking, network modeling and optimization, and statistical inference. Dr. He is a senior member of IEEE, an Associate Editor for IEEE Transactions on Communications (2017-2020) and IEEE/ACM Transactions on Networking (2017-2021), and an Area TPC Chair of IEEE INFOCOM (2021). She received multiple Outstanding Contributor Awards from IBM, and multiple paper awards from ITA, ICDCS, SIGMETRICS, and ICASSP.