Stealthy DGoS Attack under Passive and Active Measurements

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

***Abstract*—As a tool to infer the internal state of a network that cannot be measured directly (e.g., the Internet and all-optical networks), 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 path performances without exposing the manipulated links to network tomography. While existing studies on this attack assume that network tomography only measures the paths actively used for data transfer (by passively recording the performance of data packets), 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 and shed light on possible defenses.**

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

I. Introduction

Network tomography [1] 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 [2], [3], [4], a Boolean model for localizing failures [5], [6], or various probabilistic models for accommodating performance fluctuations (see [1] 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 delaying 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. For linear observation models, recent studies [7], [8] revealed novel attacks that can substantially degrade path performances while misleading network tomography to believe that the manipulated links perform well. 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 defending against attacks. In this work, we harden this intuition by quantifying the damage a stealthy attacker can inflict on a network monitored by network tomography with both passive and active measurements.

A. Related Work

Network tomography is a rich family of network monitoring techniques that infer network internal characteristics from external measurements [1], [9]. Early works focused on *best-effort solutions*, which tried to find the most likely network state from given measurements, obtained by unicast [10], multicast [11], and their variations (e.g., back-to-back unicast [12]). After observing that an arbitrary set of measurements is frequently insufficient for identifying all the link metrics [13], [14], later works aimed at either reducing the ambiguity with given measurements (e.g., [10], [13]), or ensuring identifiability by carefully designing the monitor locations and the measurement paths (e.g., [2], [3], [4]). All these works assume a *benign setting*, where the links behave consistently and the measurements are truthful.

Very few works have considered network tomography in an *adversarial setting*. In [7], optimizations were formulated to design the manipulations at compromised nodes in order to

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

cause the maximum degradation while scapegoating certain benign links as the cause of poor performance; however, the set of compromised nodes is not optimized. In [8], 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 at each path traversing at least one compromised link). However, both works assumed that only the data paths are monitored by network tomography, and thus the manipulation of any measurement path will affect the end user's performance.

B. Summary of Contributions

Our goal is to quantify the impact of DGoS attack in networks monitored by network tomography based on both passive and active measurements.

1) We extend the attack model in [8] to include both passive measurement paths (for data packets) and active measurement paths (for probes), 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 strategy and develop efficient approximations.

3) We evaluate the proposed attack strategies on real Internet topologies. Our evaluations show that: (i) the proposed strategies achieve more severe performance degradation than intuitive alternatives and thus better reveal the potential damage of DGoS attack, (ii) even a few compromised links can cause significant damage to end-to-end communications, and (iii) adding active measurements provides little protection if these measurements are only between the communicating terminals.

Roadmap. Section II formulates the generalized DGoS attack. Section III presents the optimality conditions and the associated algorithms for attack design. Section IV evaluates the proposed algorithms. Finally, Section V concludes the paper.

II. Problem Formulation

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 $R = (r_{ij})_{p_i \in P, l_j \in L}$ 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 detects link anomalies by finding a solution $\widehat{\mathbf{x}}$ to $R\widehat{\mathbf{x}} = \mathbf{y}$ 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 before-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 metric, which can be infinity; assume that $\tau \leq \tau_{\max}$.

We note that the solution to $R\widehat{\mathbf{x}} = \mathbf{y}$ may not be unique as R may not have a full column rank [13], [14]. In this case, we consider a powerful network tomography solver that can compute all the possible solutions to the link metrics.

B. Attack Model

Suppose that an attacker wants to degrade the performance of a subset of paths $P_d \subseteq P$. Paths in $P \setminus P_d$ are monitored by network tomography but not carrying data flows of interest. For example, paths in $P \setminus P_d$ may be only used by probes or non-performance-sensitive packets.

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_j ($l_j \in L$) denote the cost of compromising link l_j, and k 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 [7], [8]:

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 $p_i \in P_m$.

Moreover, the attacker wants to stay stealthy to the detection system by ensuring that (i) the network tomography problem remains feasible under the manipulation, i.e., $R\widehat{\mathbf{x}} = R\mathbf{x} + \mathbf{z}$ is feasible, and (ii) there exists a feasible solution $\widehat{\mathbf{x}}$ according to which all the compromised links appear normal.

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

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

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

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

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

$$\sum_{l_j \in L_m} c_j \leq k, \tag{1e}$$

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

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 [8, (1)] in that: (i) only degradation on the paths in P_d is included in the objective, which allows us to model network tomography based on both passive and active measurements, 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.

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. The Case of $k = \infty$

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 unbudgeted case, we establish the sufficient and the 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 minimal total traversal number, i.e., $\mathcal{C}^*_{P'} := \{L' \in \mathcal{C}_{P'} | T(L') \leq T(L''), \forall L'' \in \mathcal{C}_{P'}\}$.

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 total traversal number, i.e., $L_m \in \mathcal{C}^*_P$.

Proof. see [15]. □

Remark: Theorem III.1 generalizes [8, 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.

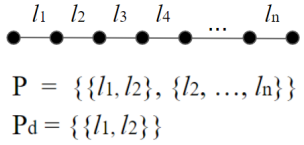

Fig. 1. $L_m \in \mathcal{C}^*_P$ is not necessary for optimality.

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 necessary by a simple example. In the example shown in Fig. 1, suppose that $\sum_{k=2}^{n} x_k \geq \tau_{\max}$. 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. Nevertheless, we will show that forming a cut of 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. See [15]. □

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. See [15]. □

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.

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 total traversal number by P_d:

$$\min_{L_m} \sum_{l_j \in L_m} w_j \tag{2a}$$

$$\text{s.t. } L_m \in \mathcal{C}_P, \tag{2b}$$

$$L_m \subseteq L \tag{2c}$$

GALS is similar to the *adversarial link selection (ALS)* problem in [8], except 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 [8], GALS is also NP-hard. Moreover, similarly to the reduction of ALS to the *weighted set cover (WSC)* problem [8], GALS can also be

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;

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 [16] that this greedy algorithm has the best approximation factor for WSC, which is $\Theta(\log |P|)$ in our case.

B. The Case of $k < \infty$

In the general case, 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 for each link l_j, the *generalized constrained adversarial link selection* (GCALS) problem aims at:

$$\max_{L_m} T_m := \sum_{l_j \in L'_n} w_j \tag{3a}$$

$$\text{s.t.} \sum_{l_j \in L_m} c_j \leq k, \tag{3b}$$

$$L_m \subseteq L, \tag{3c}$$

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. See [15]. □

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

Theorem III.5. The GCALS problem (3) is NP-hard.

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

input : P, P_d, k
output: Compromised links L_m

1 $k' \leftarrow k$ `// remaining budget`
2 $L_c \leftarrow L \setminus \{l_j \in L | c_j > k'\}$ `// candidate links`
3 $L_m \leftarrow \emptyset$;
4 $P_m \leftarrow \emptyset$;
5 $(\alpha'_j, \beta'_j, \gamma'_j)_{l_j \in L} \leftarrow$ solving the LP relaxation of (4);
6 **while** $P_m \subset P$ *and* $L_c \neq \emptyset$ **do**
7 Find the link $l_j \in L_c$ with the largest ratio $\frac{\alpha'_j |P_j \setminus P_m|}{c_j}$;
8 $k' \leftarrow k' - c_j$;
9 $L_c \leftarrow (L_c \setminus \{l_j\}) \setminus \{l_{j'} \in L_c | c_{j'} > k'\}$;
10 $L_m \leftarrow L_m \cup \{l_j\}$;
11 $P_m \leftarrow P_m \cup P_j$;
12 return L_m;

Proof. See [15]. □

Next, we will develop a solution by formulating this problem as an integer linear programming (ILP) problem (w_j is defined as in Definition 1, the other parameters defined as in (1)):

$$\max_{\alpha_j, \beta_j, \gamma_j} \sum_{l_j \in L} \gamma_j w_j \tag{4a}$$

$$\text{s.t.} \sum_k \alpha_k r_{ik} \geq r_{ij} \beta_j \qquad \forall l_j \in L, \forall p_i \in P, \tag{4b}$$

$$\sum_{l_j \in L} c_j \alpha_j \leq k \qquad \forall l_j \in L, \tag{4c}$$

$$\gamma_j \leq 1 - \alpha_j \qquad \forall l_j \in L, \tag{4d}$$

$$\gamma_j \leq \beta_j \qquad \forall l_j \in L, \tag{4e}$$

$$\gamma_j \geq \beta_j - \alpha_j \qquad \forall l_j \in L, \tag{4f}$$

$$\alpha_j, \beta_j, \gamma_j \in \{0, 1\} \qquad \forall l_j \in L. \tag{4g}$$

Lemma III.6. The optimization (4) is equivalent to the optimization (3), where $l_j \in L_m$ if and only if $\alpha_j = 1$.

Proof. see [15]. □

The ILP formulation (4) allows us to leverage techniques for solving ILP to solve the GCALS problem (3). In particular, one commonly-used approach is to relax the ILP into an LP by relaxing the integer constraint (4g) 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 $\frac{\alpha'_j |P_j \setminus P_m|}{c_j}$, 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.

IV. Performance Evaluation

In order to understand the potential damage of the generalized DGoS attack, we evaluate the proposed algorithms as well as benchmarks on real network topologies.

A. Experiment Setup

1) Network topology: We use real network topologies from public datasets, whose parameters are shown in the TABLE I. The first four topologies are Point of Presence (PoP)-level topologies from the Internet Topology Zoo [17], and the last two topologies are router-level topologies from the CAIDA project [18].

2) Paths: For each topology, we select a given number of terminals from low-degree nodes (degree ≤ 2), and compute P as the shortest paths (in hop count) between terminals, with ties broken arbitrarily. We then select a subset of paths in P as P_d uniformly at random.

3) Other parameters: Before the attack, each link has a delay sampled from the interval of $[0, 20)$ (ms) uniformly at random. The cost[1] of compromising each link is drawn uniformly at random from the interval of $[0, 2)$. A link is considered "normal" if its delay is within 150 ms, i.e., $\tau = 150$. The maximum delay at a link is 2000 ms, i.e., $\tau_{\max} = 2000$.

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

i) "Random selection" ('random'): This algorithm selects links uniformly at random within the given budget.
ii) "Top traversal" ('top traversal'): Based on the intuition that compromising the most traversed links will allow the attacker to control the most paths, this algorithm sorts the links by their traversal numbers in descending order, and then selects links in this order under 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 (4) is used as probabilities for selecting links.
iv) "ILP" ('ILP'): This solution directly solves the ILP (4) by a commercial optimizer called Gurobi, 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).

[1]The cost is relative to the budget and is thus unitless. We set the average cost to 1 so that a budget of k will allow the attacker to compromise k randomly selected links on the average, while the optimal strategy may compromise more or fewer.

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

TABLE I
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

TABLE II
Parameters for Evaluating Attacks under Varied Budget

Network	#terminals	budget k	$\lvert P_d \rvert$	$\lvert P \rvert$
Bics	15	$5, 7, 9, 11, 13, \infty$	50	105
BTN	15	$5, 7, 9, 11, 13, \infty$	50	105
Cogent	20	$5, 7, 9, 11, 13, \infty$	100	190
Colt	20	$5, 7, 9, 11, 13, \infty$	100	190
AS 8717	30	$5, 7, 9, 11, 13, \infty$	200	435
AS 20965	30	$5, 7, 9, 11, 13, \infty$	200	435

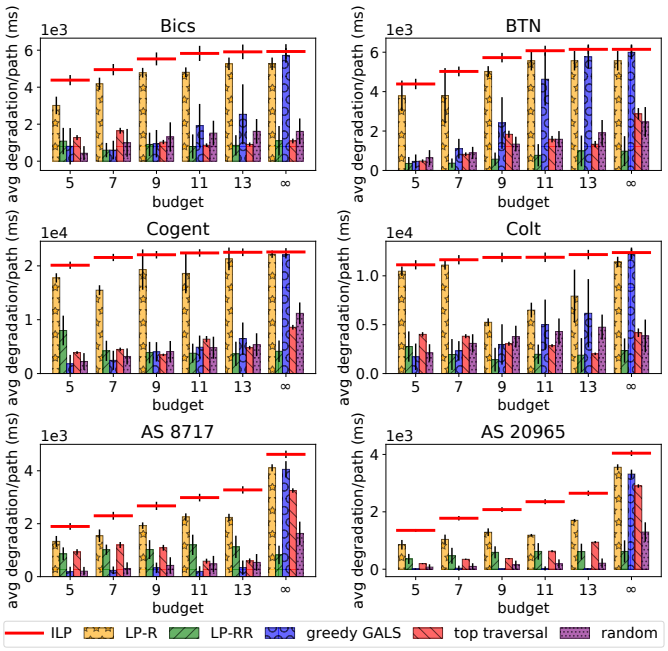

Fig. 2. Average delay degradation per path when budget k varies

B. Experiment Results

We evaluate the average performance degradation over the paths in P_d (plus/minus one standard deviation) under each attack design, computed over 100 Monte Carlo runs.

First, we increase the budget k as in Table II to evaluate the impact of a stronger attacker. Fig. 2 shows that the proposed algorithm for the budgeted case (LP-R) achieves much bigger damage than the benchmark heuristics for a wide range of k, whereas the proposed algorithm for the unbudgeted case (Greedy GALS) is only effective when k is large. Moreover,

TABLE III
PARAMETERS FOR EVALUATING ATTACKS UNDER VARIED $|P|$ ($k = 10$)

Network	#terminals	$\lvert P_d \rvert$	$\lvert P \rvert$
Bics	15	50	50 , 60, 70, 80, 90, 105
BTN	15	50	50 , 60, 70, 80, 90, 105
Cogent	20	100	100, 120, 140, 160, 180, 190
Colt	20	100	100, 120, 140, 160, 180, 190
AS 8717	30	240	240, 280, 320, 360, 400, 435
AS 20965	30	240	240, 280, 320, 360, 400, 435

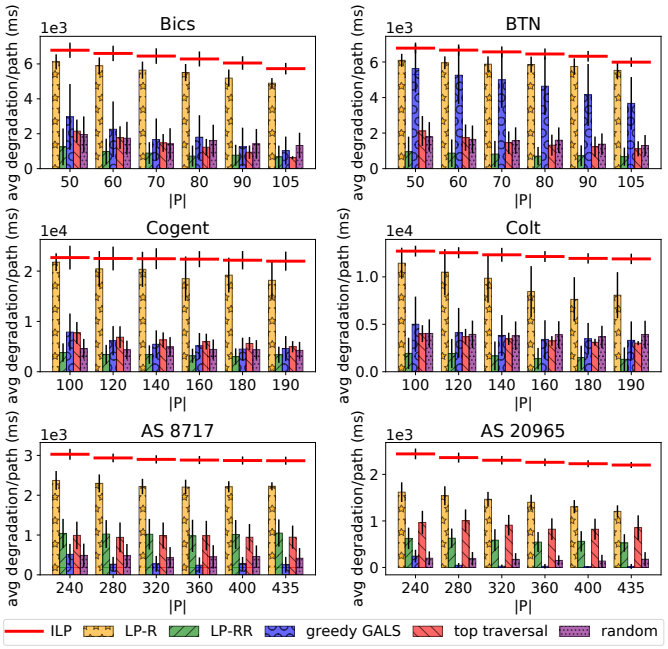

Fig. 3. Average delay degradation per path when $|P|$ varies

the attacker can cause significant damage with only a few compromised links (e.g., causing > 1 second of delay per path when compromising an average of 5 links). Note that LP-R is non-monotone for Colt because it does not utilize the budget optimally (recall that the optimal solution is NP-hard as shown in Theorem III.5).

Next, we fix k and $|P_d|$ but increase $|P|$ (and hence $|P| - |P_d|$) as in Table III to evaluate the impact of monitoring more paths by network tomography. Fig. 3 shows that the damage achieved by all the attack strategies, especially the optimal strategy (ILP), decreases very slowly in $|P|$.

Discussion: The above results provide a number of insights for defending against DGoS attacks: (i) it is important to defend against intelligent attack strategies as they can achieve substantially more damage than simplistic ones, (ii) it is important to guard against compromised network elements (nodes/links) from the bottom up as even a few compromised elements can cause a big damage, and (iii) simply monitoring more paths is not sufficient to make network tomography robust against DGoS attacks. For (iii), however, the added paths in our experiments are only between the terminals, and it remains open how much protection can be achieved by carefully designing the paths, which is left for future work.

V. CONCLUSION

We have quantified the impact of DGoS attack by formulating and computing the maximum damage that an attacker can inflict on end-to-end communications through compromised links, without exposing these links to a tomography-based detection system based on both passive and active measurements. By establishing optimality conditions, we convert the attack design problem into novel combinatorial optimization problems and develop efficient algorithms. Our evaluations on real network topologies reveal significant damage of the DGoS attack and provide insights for future defenses.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] ——, "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.

[7] 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.

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

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

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

[11] 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.

[12] 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.

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

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

[15] C. Chiu and T. He, "Stealthy dgos attack under passive and active measurements," Technical Report, May 2020. [Online]. Available: https://sites.psu.edu/nsrg/files/2020/05/Stealthy_DGoS_Attack_under_passive_and_active_measurement_report.pdf

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

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

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