# Modeling and mitigating link-flooding distributed denial-of-service attacks via learning in Stackelberg games

Guosong Yang and João P. Hespanha

Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, CA 93106, USA, {guosongyang, hespanha}@ucsb.edu

**Summary.** This work formulates the mitigation of link-flooding distributed denial-of-service attacks as a routing problem among parallel links. In order to address the challenge that the adversary can observe the routing strategy before assigning attack traffic, we model the conflict between routing and attack as a Stackelberg game. For a general class of adversaries, we establish a characterization of an optimal attack strategy that reduces the search space to a finite set, and construct explicit formulae for Stackelberg equilibria and costs for a special class of networks. When the attack objective and capacity are unknown, we propose a learning-based approach that predicts the routing cost using a neural network and minimizes the predicted cost via projected gradient descent. A simulation study is provided to demonstrate the effectiveness of our approach.

## 1 Introduction

A major threat to Internet security today is the distributed denial-of-service (DDoS) attack, in which an adversary attempts to interrupt legitimate users' access to certain network resources by sending superfluous traffic from a vast number of subverted machines (bots). Since the first incident of DDoS attack reported by the Computer Incident Advisory Capability in 2000 [1], many large-scale DDoS attacks have been launched against crucial infrastructures and services [2,3]. Moreover, there has been a drastic and persistent increase in the size and frequency of DDoS attacks every year. The 14th annual Worldwide Infrastructure Security Report from NETSCOUT Systems, Inc. showed that the global max attack size reached 1.7 Tbps in 2018, a 273% increase from 2017 [4]. The same report also found that 94% of enterprises observed DDoS attacks on their encrypted traffic in 2018, nearly twice the percentage as 2017.

As pointed out in [3], most DDoS attacks exploit one or both of the following methods: 1) disrupting legitimate users' service by exhausting server resources, and 2) disrupting legitimate users' connectivity by exhausting link

bandwidth. The second method (called link-flooding DDoS) proves to be especially effective and stealthy as it can be executed without sending attack traffic to the victims. Novel link-flooding DDoS attacks such as the Coremelt attack [5] and the Crossfire attack [6] have attracted substantial research interest, since they allow individual bots participating in the coordinated attack to keep their transmission rates below detection thresholds, yet effectively exhausts the bandwidth of target links. Existing results on mitigating link-flooding DDoS attacks mainly focus on developing techniques to distinguish attack traffic from legitimate one [7–12]. However, the adversary can often rotate attack traffic among different sets of bots and links to maintain congestion [6], which motivates us to consider scenarios where the detection of attack traffic is impossible. In [13], the authors studied the modeling and mitigation of the Coremelt attack [5] in such scenarios.

Game theory provides a systematic framework for modeling the conflict between a router and an adversary orchestrating a DDoS attack [14–17]. Existing game-theoretical results on mitigating DDoS attacks mainly focus on Nash equilibrium—a tuple of actions for which no player has a unilateral incentive to change; see, e.g., [18–20] and references therein. However, most DDoS attacks are characterized by asymmetric information: the router is unaware of the attack objective and capacity a priori, whereas the adversary is able to observe the routing action and assign attack traffic accordingly [6]. Since asymmetric information often leads to scenarios with no Nash equilibrium, we consider instead a hierarchical game model proposed by Stackelberg [21]. In our two-player Stackelberg game, the router (called the leader) selects its action first, and then the adversary (called the follower), informed of the router's choice, selects its own action. Standard conditions for existence of a Stackelberg equilibrium are weaker than those of a Nash equilibrium [16, p. 181].

Stackelberg games have been applied to network problems with asymmetric information in applications such as routing [22], scheduling [23], and channel allocation [24]. They have also been applied to various real-world security domains and have lead to practical implementations including the ARMOR program at the Los Angeles International Airport [25], the IRIS program used by the US Federal Air Marshals [26], and counterterrorism programs for crucial infrastructures such as power grid and oil reserves [27, 28]. A recent work [29] analyzed Nash and Stackelberg equilibria for a routing game that is similar but more restrictive than the one considered in this work.

Since the router only has incomplete information of the attack objective and capacity, a fundamental question is whether an iterative data-driven routing algorithm can converge to a Stackelberg equilibrium as it adjusts routing based on historical outcomes. For our Stackelberg game, standard game-theoretical learning processes such as Fictitious play [30, 31] and gradient response [32, 33] cannot be applied. Most existing results on learning in Stackelberg games are limited to linear and quadratic costs and finite action sets [34–36], which are too restrictive for modeling the DDoS attacks of inter-

est. In [37], the authors proposed a learning-based approach for Stackelberg games that could simultaneously estimate the attack strategy and minimize the routing cost, based on adaptive control techniques and hysteresis switching. A motivation for the current work is to extend the results from [37] for more complicated problems such as mitigating link-flooding DDoS attacks based on neural network techniques.

In this work, we formulate the mitigation against link-flooding DDoS attacks as a routing problem among parallel links. Our goal is to construct an optimal routing strategy that minimizes the damage caused by the attack, with the caveat that the adversary is able to observe the routing strategy before assigning attack traffic. The network concepts and the routing and attack optimization problems are described in Section 2. Our formulation provides a hight level of generality by allowing scenarios where the "attack" is actually benign network traffic or a combination of benign and malicious traffic.

In Section 3, we model the conflict between routing and attack as a Stackelberg game. In Stackelberg games, it is common for the follower to have multiple optimal actions against a leader's action, leading to different equilibria for "optimistic" or "pessimistic" leaders that assume the follower will play the optimal actions that are best or worst for the leader, respectively. Therefore, we adopt the notions of strong and weak Stackelberg equilibria, in which the former corresponds to an optimistic router and the latter a pessimistic one.

In Section 4, we consider a general class of adversaries that may have different priorities for different links, and establish a characterization of an optimal attack strategy by showing that it belongs to a finite set. Moreover, we construct explicit formulae for strong and weak Stackelberg equilibria and costs for a special class of networks. Our results shows that there is no weak Stackelberg equilibrium unless the adversary has the same priority for all links, in which case the strong Stackelberg equilibrium is also a weak one. Nevertheless, a pessimistic router can always achieve a cost arbitrarily close to the strong Stackelberg cost.

In Section 5, we considered the more general scenario where the router does not know the attack objective or capacity and thus cannot predict the attack action or the cost associated with a specific routing action. We propose a learning-based approach that predicts the routing cost using a neural network trained based on historical data, and then minimizes the predicted routing cost via projected gradient descent. A simulation study is provided in Section 6 to demonstrate the effectiveness of our leaning-based approach. Section 7 concludes the chapter with a brief summary and an outlook for future research directions.

*Notations:* Let $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, and $\mathbb{Z}_{>0}$ be the sets of nonnegative real numbers, positive real numbers, and positive integers, respectively. Denote by $\mathbf{1}_n$ the vector $(1, \ldots, 1) \in \mathbb{R}^n$. Denote by $\|\cdot\|$ the Euclidean norm for vector. For a vector $v \in \mathbb{R}^n$, denote by $v_i$ its $i$-th scalar component and write $v = (v_1, \ldots, v_n)$.

For a set $\mathcal{S} \subset \mathbb{R}^n$, denote by $|\mathcal{S}|$ its cardinality, and by $\partial\mathcal{S}$ and $\overline{\mathcal{S}}$ its boundary and closure, respectively.

## 2 Routing and attack in communication network

We are interested in modeling and mitigating link-flooding DDoS attacks. Many such attacks focus on flooding parallel links that form a so-called bottleneck of the network to maximize congestion [38]. Therefore, we abstract the communication network as a directed graph of $L$ parallel bottleneck links connecting a source to a destination, and focus our discussion on routing traffic among them. The set of links is denote by $\mathcal{L} := \{1, \dots, L\}$.

Let $R > 0$ be the total *desired transmission rate* of legitimate traffic that a router needs to distribute among the $L$ parallel links. The traffic distribution is represented by a *routing assignment* vector $r \in \mathbb{R}^L_{\geq 0}$ that satisfies

$$\sum_{l \in \mathcal{L}} r_l = 1.$$

If there was no congestion on a link $l \in \mathcal{L}$, then the transmission rate of legitimate traffic from the router on link $l$ would be the *desired user rate* $r_l R$.

Motived by link-flooding DDoS attacks such as the Crossfire attack [6], we assume that an adversary disrupts communication by injecting superfluous traffic on the links according to an *attack assignment* vector $a \in \mathbb{R}^L_{\geq 0}$, and call $a_l$ the *attack rate* on link $l \in \mathcal{L}$. The attack is constrained by a total budget $A > 0$, that is,

$$\sum_{l \in \mathcal{L}} a_l \leq A.$$

For each link $l \in \mathcal{L}$, there is a constant capacity $c_l > 0$ that upper-bounds its total transmission rate. If the sum of the desired user rate $r_l R$ and the attack rate $a_l$ is larger than the capacity $c_l$, then there is congestion on link $l$ which results in packet drops and retransmissions. In many widely used transmission protocols such as the Transmission Control Protocol (TCP) [39, 40], legitimate users will decrease their transmission rates until the total rate on link $l$ is no longer larger than its capacity $c_l$. By contrast, a malicious adversary aims at sustaining congestion and thus does not decrease the attack rates. Therefore, we model the effect of congestion by a decrease in user rates and define the *actual user rates* by

$$u_l := \min\{r_l R, \max\{c_l - a_l, 0\}\}, \qquad l \in \mathcal{L}. \tag{1}$$

The goal of the router is to maximize the sum of actual user rates. Therefore, it minimizes the *routing cost* defined by

$$J(r, a) := -\sum_{l \in \mathcal{L}} u_l. \tag{2}$$

Motivated by link-flooding DDoS attacks, we call $r_l R$ and $a_l$ the transmission rates of *legitimate* and of *attack* traffic, respectively, but in practice the key distinction is that the former represents traffic that the router aims to protect (in terms of minimizing the routing cost (2) above), whereas the router's objective does not care for the latter. If $a$ was indeed generated by a malicious adversary, it would likely attempt to maximize the routing cost (2), or equivalently, to minimize the *attack cost* defined by

$$H(a, r) := -J(r, a) = \sum_{l \in \mathcal{L}} u_l. \tag{3}$$

However, in general one should consider more general attack costs $H(a, r)$ that may not be precisely the symmetry of $J(r, a)$ and could even be altogether independent of the routing assignment $r$. Not knowing the attack cost $H(r, a)$ is in fact the key motivation for the learning-based approach proposed in this paper. In view of this, we only make the mild assumption that $H(a, r)$ is continuous in $a$ for each fixed $r$, except in Section 4 where we will consider an attack cost for a malicious adversary that generalizes (3).

In summary, the routing optimization problem is given by

$$\begin{aligned}
\min_{r} \quad & J(r, a), && && && \text{(4a)}\\
\text{s.t.} \quad & \sum_{l \in \mathcal{L}} r_l = 1 && && \text{(rate conservation)}, && \text{(4b)}\\
& r_l \geq 0 && \forall\, l \in \mathcal{L} && \text{(nonnegative rate)}, && \text{(4c)}
\end{aligned}$$

and the attack optimization problem is given by

$$\begin{aligned}
\min_{a} \quad & H(a, r), && && && \text{(5a)}\\
\text{s.t.} \quad & \sum_{l \in \mathcal{L}} a_l \leq A && && \text{(attack budget)}, && \text{(5b)}\\
& a_l \geq 0 && \forall\, l \in \mathcal{L} && \text{(nonnegative rate)}. && \text{(5c)}
\end{aligned}$$

In particular, we consider the general case where the router does not even know the total desired transmission rate $R$, and thus do not assume $r_l R \leq c_0$ on any link $l \in \mathcal{L}$. Both the routing cost $J(r, a)$ and the attack cost $H(a, r)$ are functions of the routing assignment $r$ and the attack assignment $a$. Therefore we model the conflict between the router and the adversary using a game-theoretical model defined in the next section.

## 3 Stackelberg game model

We are interested in scenarios where there is asymmetric information between the router and the adversary. Specifically, the adversary is able to observe the

routing assignment $r$ before selecting the attack assignment $a$. To model such scenarios, we consider a hierarchical game model called Stackelberg games [21], in which the router (called the leader) selects the routing action $r \in \mathcal{R}$ first, and then the adversary (called the follower), informed of the router's choice, selects the attack action $a \in \mathcal{A}$. The routing action set $\mathcal{R}$ and the attack action set $\mathcal{A}$ are defined by

$$\mathcal{R} := \{r \in \mathbb{R}^L_{\geq 0} : \text{(4b) and (4c) hold}\},$$
$$\mathcal{A} := \{a \in \mathbb{R}^L_{\geq 0} : \text{(5b) and (5c) hold}\}.$$

The game between the router and the adversary is fully defined by the tuple $(\mathcal{R}, \mathcal{A}, J, H)$. Formally, Stackelberg equilibria are defined as follows; cf. [14, Section 3.1], [16, Definition 4.6, p. 179], and [41, 42].

**Definition 1** (Stackelberg)**.** Given a game defined by the tuple $(\mathcal{R}, \mathcal{A}, J, H)$, a routing action $r^*_{\mathrm{s}} \in \mathcal{R}$ is called a *strong Stackelberg equilibrium routing action* if

$$J^*_{\mathrm{s}} := \inf_{r \in \mathcal{R}} \min_{a \in \beta_a(r)} J(r, a) = \min_{a \in \beta_a(r^*_{\mathrm{s}})} J(r^*_{\mathrm{s}}, a), \tag{6}$$

where

$$\beta_a(r) := \arg\min_{a \in \mathcal{A}} H(a, r)$$

denotes the set of optimal attack actions against a routing action $r \in \mathcal{R}$, and $J^*_{\mathrm{s}}$ is known as the *strong Stackelberg routing cost*; a routing action $r^*_{\mathrm{w}} \in \mathcal{R}$ is called a *weak Stackelberg equilibrium routing action* if

$$J^*_{\mathrm{w}} := \inf_{r \in \mathcal{R}} \max_{a \in \beta_a(r)} J(r, a) = \max_{a \in \beta_a(r^*_{\mathrm{w}})} J(r^*_{\mathrm{w}}, a), \tag{7}$$

where $J^*_{\mathrm{w}}$ is known as the *weak Stackelberg routing cost*; and for an $\varepsilon > 0$, a routing action $r^*_\varepsilon \in \mathcal{R}$ is called a *weak $\varepsilon$ Stackelberg routing action* for $(\mathcal{R}, \mathcal{A}, J, H)$ if

$$\max_{a \in \beta_a(r^*_\varepsilon)} J(r^*_\varepsilon, a) \leq J^*_{\mathrm{w}} + \varepsilon. \tag{8}$$

For each fixed $r \in \mathcal{R}$, as the function $H(a, r)$ is continuous in $a$ and the set $\mathcal{A}$ is compact, the set $\beta_a(r)$ is nonempty and compact; thus the minima in (6) and the maxima in (7) and (8) can be attained. However, the functions $\min_{a \in \beta_a(r)} J(r, a)$ and $\max_{a \in \beta_a(r)} J(r, a)$ are not necessarily continuous, and thus may only have infima over the set $\mathcal{R}$.

The difference between strong and weak Stackelberg equilibria stems from scenarios where the optimal attack action is nonunique, that is, where $|\beta_a(r)| > 1$. Such scenarios are common under Stackelberg equilibrium routing actions, a phenomenon similar to the fact that in a mixed-strategy Nash equilibrium, every pure strategy in the support of a player's mixed strategy is a best response to the opponents mixed strategy [15, Lemma 33.2, p. 33]. If $|\beta_a(r)| > 1$, which optimal attack action the adversary plays will affect the resulting routing cost. The strong Stackelberg equilibrium represents the

"optimistic" view that the adversary will select the one that is best for the router, whereas the weak Stackelberg equilibrium represents the "pessimistic" view that the worst one for the router will be selected. The conditions for existence of a strong Stackelberg equilibrium are weaker than those of a weak Stackelberg equilibrium [43], as we shall see in Section 4.1. However, a weak Stackelberg equilibrium is usually more desirable as it provides a guaranteed upper bound for the routing cost, regardless of the adversary's tie-breaking rule. The terminology "strong" and "weak" originates from [42] and is widely used in *Stackelberg Security Games*; see, e.g., [44] and references therein.

In this work, we consider games with incomplete information where the router does not know the attack cost function $H$ or even the attack action set $\mathcal{A}$ and thus cannot predict the set of optimal attack actions $\beta_a(r)$. This level of generality allows us to capture scenarios where the "adversary" is in fact benign and the attack action $a$ may not even depend on the routing action $r$. However, in the following section we do focus our attention on malicious adversaries with a cost function the generalizes (3).

## 4 Optimal attack and Stackelberg equilibria for malicious adversaries

As mentioned in Section 2, a natural candidate for the attack cost $H(a, r)$ for a malicious adversary is given by (3), which makes $(\mathcal{R}, \mathcal{A}, J, H)$ a zero-sum game. In this section, we consider a more general scenario that deviates from the zero-sum case, as the adversary may have different priorities for different links, and the attack cost is defined by

$$H(a, r) := \sum_{l \in \mathcal{L}} \gamma_l u_l, \tag{9}$$

where $\gamma = (\gamma_1, \ldots, \gamma_L) \in (0, 1]^L$ can be viewed as an *attack priority* vector.

From the definition of the actual user rates (1), we see that a malicious adversary with the attack cost (9) has no incentive to assign less attack traffic than the total budget $A$ or to assign more attack traffic to a link than its capacity. Therefore, for each routing action $r \in \mathcal{R}$, there is an optimal attack action $a \in \beta_a(r)$ such that

$$\sum_{l \in \mathcal{L}} a_l = A \tag{10}$$

with

$$a_l \leq c_l \qquad \forall\, l \in \mathcal{L}. \tag{11}$$

For attack actions $a \in \mathcal{A}$ such that (11) holds, we can rewrite the definition of the actual user rates (1) by

$$u_l := \min\{r_l R,\, c_l - a_l\}, \qquad l \in \mathcal{L}.$$

Also, we assume

$$A < \sum_{l \in \mathcal{L}} c_l,$$

since otherwise the adversary can simply fill all the bottleneck links and the attack optimization problem becomes trivial.

In the following result, we establish that there is always an optimal attack action with the attack rates equal to 0 or to the corresponding link capacities for all but one link.

**Theorem 1.** *For each routing action $r \in \mathcal{R}$, there is an optimal attack action $a^* \in \beta_a(r)$ such that* (10) *and* (11) *hold, and there is at most one link $l_0 \in \mathcal{L}$ where $a^*_{l_0} \in (0,\, c_{l_0})$, that is,*

$$a^* \in \mathcal{A}^* := \{a \in \mathcal{A} : (10) \text{ and } (11) \text{ hold, and } |\{l \in \mathcal{L} : a_l \in (0,\, c_l)\}| \leq 1\}. \tag{12}$$

*Proof.* Based on the analysis before Theorem 1, there is an optimal attack action $\bar{a}^* \in \beta_a(r)$ such that (10) and (11) hold. We construct an optimal attack action $a^* \in \beta_a(r) \cap \mathcal{A}^*$ as follows.

1. Let $a = \bar{a}^*$. Then $a \in \beta_a(r)$ and satisfies (10) and (11).
2. If there are two links $l_1, l_2 \in \mathcal{L}$ such that

$$a_{l_1} \in (0, c_{l_1}), \qquad a_{l_2} \in (0, c_{l_2}),$$

   then $a \notin \mathcal{A}^*$ and we go to step 3; otherwise $a \in \mathcal{A}^*$ and we go to step 4.
3. Consider the following two possibilities.
   a) If there is a link $\bar{l}_1 \in \{l_1, l_2\}$ such that $a_{\bar{l}_1} \leq c_{\bar{l}_1} - r_{\bar{l}_1} R$, then the corresponding actual user rate satisfies

$$u_{\bar{l}_1} = \min\{r_{\bar{l}_1} R,\, c_{\bar{l}_1} - a_{\bar{l}_1}\} = r_{\bar{l}_1} R.$$

   We define an attack action $\bar{a} \in \mathcal{A}$ by moving as much attack traffic on $\bar{l}_1$ as possible to the other link $\bar{l}_2 \in \{l_1, l_2\} \backslash \{\bar{l}_1\}$, that is,

$$\bar{a}_l := \begin{cases} \max\{0,\, a_{l_1} + a_{l_2} - c_{\bar{l}_2}\}, & l = \bar{l}_1, \\ \min\{a_{l_1} + a_{l_2},\, c_{\bar{l}_2}\}, & l = \bar{l}_2, \\ a_l, & l \in \mathcal{L} \backslash \{l_1,\, l_2\}. \end{cases} \tag{13}$$

   Then clearly $\bar{a} \in \mathcal{A}$ and satisfies (10) and (11) with $\bar{a}$ in place of $a$, and at least one of $\bar{a}_{\bar{l}_1} = 0$ and $\bar{a}_{\bar{l}_2} = c_{\bar{l}_2}$ holds. Moreover, we have

$$\bar{a}_{\bar{l}_1} \leq a_{\bar{l}_1} \leq c_{\bar{l}_1} - r_{\bar{l}_1} R, \qquad \bar{a}_{\bar{l}_2} \geq a_{\bar{l}_2};$$

   thus the corresponding actual user rates satisfy $\bar{u}_l = u_l$ for all $l \in \mathcal{L} \backslash \{l_1,\, l_2\}$ and

$$\bar{u}_{\bar{l}_1} = \min\{r_{\bar{l}_1}R,\, c_{\bar{l}_1} - \bar{a}_{\bar{l}_1}\} = r_{\bar{l}_1}R = u_{\bar{l}_1},$$
$$\bar{u}_{\bar{l}_2} = \min\{r_{\bar{l}_2}R,\, c_{\bar{l}_2} - \bar{a}_{\bar{l}_2}\} \le \min\{r_{\bar{l}_2}R,\, c_{\bar{l}_2} - a_{\bar{l}_2}\} = u_{\bar{l}_2}.$$

Therefore, the attack costs for $\bar{a}$ and $a$ satisfy

$$H(\bar{a}, r) - H(a, r) = \gamma_{\bar{l}_1}\bar{u}_{\bar{l}_1} + \gamma_{\bar{l}_2}\bar{u}_{\bar{l}_2} - \gamma_{\bar{l}_1}u_{\bar{l}_1} - \gamma_{\bar{l}_2}u_{\bar{l}_2} \le 0;$$

thus $a \in \beta_a(r)$ implies $\bar{a} \in \beta_a(r)$.

b) Otherwise $a_{l_1} \in (c_{l_1} - r_{l_1}R,\, c_{l_1})$ and $a_{l_2} \in (c_{l_2} - r_{l_2}R,\, c_{l_2})$. Let $\bar{l}_1$ be the link between $l_1$ and $l_2$ with a lower priority for the attack, that is,

$$\bar{l}_1 \in \arg\min_{l \in \{l_1,\, l_2\}} \gamma_l$$

(if $\gamma_1 = \gamma_2$, pick an arbitrary one). Again, we define an attack action $\bar{a} \in \mathcal{A}$ by moving as much attack traffic on $l_1$ and $l_2$ as possible to the other link $\bar{l}_2 \in \{l_1,\, l_2\}\backslash\{\bar{l}_1\}$ according to (13). Then clearly $\bar{a} \in \mathcal{A}$ and satisfies (10) and (11) with $\bar{a}$ in place of $a$, and at least one of $\bar{a}_{\bar{l}_1} = 0$ and $\bar{a}_{\bar{l}_2} = c_{\bar{l}_2}$ holds. Moreover, we have

$$\bar{a}_{\bar{l}_1} + \bar{a}_{\bar{l}_2} = a_{\bar{l}_1} + a_{\bar{l}_2}, \qquad \bar{a}_{\bar{l}_2} \ge a_{\bar{l}_2} \ge c_{\bar{l}_2} - r_{\bar{l}_2}R;$$

thus the corresponding actual user rates satisfy

$$\bar{u}_{\bar{l}_1} = \min\{r_{\bar{l}_1}R,\, c_{\bar{l}_1} - \bar{a}_{\bar{l}_1}\} \le c_{\bar{l}_1} - \bar{a}_{\bar{l}_1},$$
$$\bar{u}_{\bar{l}_2} = \min\{r_{\bar{l}_2}R,\, c_{\bar{l}_2} - \bar{a}_{\bar{l}_2}\} = c_{\bar{l}_2} - \bar{a}_{\bar{l}_2}.$$

Therefore, the attack costs for $\bar{a}$ and $a$ satisfy

$$\begin{aligned} H(\bar{a}, r) - H(a, r) &= \gamma_{\bar{l}_1}\bar{u}_{\bar{l}_1} + \gamma_{\bar{l}_2}\bar{u}_{\bar{l}_2} - \gamma_{\bar{l}_1}u_{\bar{l}_1} - \gamma_{\bar{l}_2}u_{\bar{l}_2} \\ &\le \gamma_{\bar{l}_1}(c_{\bar{l}_1} - \bar{a}_{\bar{l}_1}) + \gamma_{\bar{l}_2}(c_{\bar{l}_2} - \bar{a}_{\bar{l}_2}) \\ &\quad - \gamma_{\bar{l}_1}(c_{\bar{l}_1} - a_{\bar{l}_1}) - \gamma_{\bar{l}_2}(c_{\bar{l}_2} - a_{\bar{l}_2}) \\ &= \gamma_{\bar{l}_1}(a_{\bar{l}_1} - \bar{a}_{\bar{l}_1}) + \gamma_{\bar{l}_2}(a_{\bar{l}_2} - \bar{a}_{\bar{l}_2}) \\ &= (\gamma_{\bar{l}_1} - \gamma_{\bar{l}_2})(\bar{a}_{\bar{l}_2} - a_{\bar{l}_2}) \le 0; \end{aligned}$$

thus $a \in \beta_a(r)$ implies $\bar{a} \in \beta_a(r)$.

In summary, we have constructed an optimal attack action $\bar{a} \in \beta_a(r)$ so that (10) and (11) hold with $\bar{a}$ in place of $a$, and there is only one link $l \in \{l_1,\, l_2\}$ where $a_l \in (0, c_l)$. Let $a = \bar{a}$ and return to Step 2.

4. Let $a^* = a$. Then $a^* \in \beta_a(r) \cap \mathcal{A}^*$.

As the number of links $L$ is constant, the above algorithm is guaranteed to terminate before running Step 3 for $L$ times. Therefore, an optimal attack action $a^* \in \beta_a(r) \cap \mathcal{A}^*$ exists. □

Based on Theorem 1, it suffices to search for an optimal attack action in the subset $\mathcal{A}^*$ defined by (12) of the attack action set $\mathcal{A}$. Therefore, from now on

we restrict our attention to attack actions from $\mathcal{A}^*$ and the game $(\mathcal{R}, \mathcal{A}^*, J, H)$, and define the corresponding set of optimal attack actions against a routing action $r \in \mathcal{R}$ by

$$\beta_a^*(r) := \arg\min_{a \in \mathcal{A}^*} H(a, r) = \beta_a(r) \cap \mathcal{A}^*.$$

*Remark* 1. Consider the case where the attack action $a = a^{\text{mal}} + a^{\text{ben}}$, where $a^{\text{mal}}$ represents traffic from a malicious adversary and $a^{\text{ben}}$ represents traffic from benign users that do not response to the router, that is, the corresponding best-response set $\beta_{a^{\text{ben}}}(r)$ is constant. Then it is straightforward to see that the result in Theorem 1 still holds with $c_l - a_l^{\text{ben}}$ in place of $c_l$ in (12).

### 4.1 Optimal attack and Stackelberg equilibria for networks with identical links

One can show that, in general, finding an optimal attack action in the set $\mathcal{A}^*$ defined by (12) is at least as hard as solving the NP-hard *knapsack problem* [45]; cf. [29]. However, the problem is simpler when all the parallel links have the same capacity $c_0$. In this case, the set $\mathcal{A}^*$ defined in (12) is the set of attack actions with attack rate $c_0$ on $\lfloor A/c_0 \rfloor$ links and 0 on $L - \lceil A/c_0 \rceil$ links, that is,

$$\begin{aligned}\mathcal{A}^* = \{a \in \mathcal{A} : &\text{ (10) and (11) hold,} \\ &\text{and } |\{l \in \mathcal{L} : a_l = c_0\}| = \lfloor A/c_0 \rfloor \text{ and } |\{l \in \mathcal{L} : a_l = 0\}| = L - \lceil A/c_0 \rceil\}.\end{aligned}$$

For an attack action $a \in \mathcal{A}^*$, we denote by $l_0(a)$ the link on which $a_{l_0(a)} \in (0, c_0)$. The attack rate on the link $l_0(a)$ is a constant given by

$$a_{l_0(a)} = a_{\text{rem}} := A - \lfloor A/c_0 \rfloor c_0.$$

For an attack budget $A$ such that $A/c_0 \in \mathbb{Z}$, we have $\{l_0(a)\} = \emptyset$ and $a_{\text{rem}} = 0$. For a routing action $r \in \mathcal{R}$ and an attack action $a \in \mathcal{A}^*$, the routing cost $J(r, a)$ and attack cost $H(a, r)$ are given by

$$\begin{aligned} J(r,a) &= -u_{l_0(a)} - \sum_{l \in \mathcal{L}: a_l = 0} u_l \\ &= -\min\{r_{l_0(a)} R,\, c_0 - a_{\text{rem}}\} - \sum_{l \in \mathcal{L}: a_l = 0} \min\{r_l R,\, c_0\} \\ H(a,r) &= \gamma_{l_0(a)} u_{l_0(a)} + \sum_{l \in \mathcal{L}: a_l = 0} \gamma_l u_l \\ &= \gamma_{l_0(a)} \min\{r_{l_0(a)} R,\, c_0 - a_{\text{rem}}\} + \sum_{l \in \mathcal{L}: a_l = 0} \gamma_l \min\{r_l R,\, c_0\}. \end{aligned} \tag{14}$$

If additionally

$$\min\{r_l R,\, c_0\} \le c_0 - a_{\rm rem} \qquad \forall\, l \in \mathcal{L}, \tag{15}$$

then an attack $a \in \mathcal{A}^*$ will not cause congestion on $l_0(a)$, and thus

$$J(r,a) = -\sum_{l\in\mathcal{L}:a_l<c_0} \min\{r_l R,\, c_0\}, \qquad H(a,r) = \sum_{l\in\mathcal{L}:a_l<c_0} \gamma_l \min\{r_l R,\, c_0\}. \tag{16}$$

Note that (15) always holds for an attack budget $A$ such that $A/c_0 \in \mathbb{Z}$. A necessary condition for (15) is $A/c_0 - \lfloor A/c_0 \rfloor \le 1 - R/(Lc_0)$.

For the case where all the parallel links have the same capacity $c_0$, Theorem 1 can be extend to the following result, which shows that an optimal attack action can be found after at most $L$ trials.

**Corollary 2.** *Consider the case of equal link capacities $c_l = c_0$ for all $l \in \mathcal{L}$. For a routing action $r \in \mathcal{R}$, the set of optimal attack actions in $\mathcal{A}^*$ satisfies that*

$$\beta_a^*(r) \subset \mathcal{A}_0^*(r) := \{a \in \mathcal{A}^* : \min\{\gamma_l \min\{r_l R,\, c_0\} : a_l = c_0\} \\ \ge \max\{\gamma_l \min\{r_l R,\, c_0\} : a_l = 0\}\}; \tag{17}$$

*equivalently, an attack action $a \in \mathcal{A}^*$ is optimal against $r$ only if there exists a permutation $(l_1^a, \ldots, l_L^a)$ of $\mathcal{L}$ such that*

$$\gamma_{l_1^a} \min\{r_{l_1^a} R,\, c_0\} \ge \cdots \ge \gamma_{l_L^a} \min\{r_{l_L^a} R,\, c_0\} \tag{18}$$

*and*

$$\{l \in \mathcal{L} : a_l = c_0\} \subset \mathcal{L}_1^a \cup \{l^a_{\lceil A/c_0 \rceil}\}, \qquad \{l \in \mathcal{L} : a_l = 0\} \subset \mathcal{L}_2^a \cup \{l^a_{\lfloor A/c_0 \rfloor + 1}\}$$

*with*

$$\mathcal{L}_1^a := \{l_1^a, \ldots, l^a_{\lfloor A/c_0 \rfloor}\}, \qquad \mathcal{L}_2^a := \{l^a_{\lceil A/c_0 \rceil + 1}, \ldots, l_L^a\}.$$

*If additionally* (15) *holds, then*

$$\beta_a^*(r) = \mathcal{A}_1^*(r) := \{a \in \mathcal{A}^* : \min\{\gamma_l \min\{r_l R,\, c_0\} : a_l = c_0\} \\ \ge \max\{\gamma_l \min\{r_l R,\, c_0\} : a_l < c_0\}\}; \tag{19}$$

*equivalently, an attack action $a \in \mathcal{A}^*$ is optimal against $r$ if and only if there exists a permutation $(l_1^a, \ldots, l_L^a)$ of $\mathcal{L}$ such that* (18) *holds and*

$$\{l \in \mathcal{L} : a_l = c_0\} = \mathcal{L}_1^a.$$

*Proof.* Suppose there is an optimal attack action $a^* \in \beta_a^*(r) \backslash \mathcal{A}_0^*(r)$. Then there are two links $l_1, l_2 \in \mathcal{L}$ such that $a^*_{l_1} = c_0$, $a^*_{l_2} = 0$, and $\gamma_{l_1} \min\{r_{l_1} R,\, c_0\} < \gamma_{l_2} \min\{r_{l_2} R,\, c_0\}$. We define an attack action $\bar{a}$ by switching the attack rates on $l_1$ and $l_2$, that is,

$$\bar{a}_l := \begin{cases} a^*_{l_2}, & l = l_1, \\ a^*_{l_1}, & l = l_2, \\ a^*_l, & l \in \mathcal{L}\backslash\{l_1, l_2\}. \end{cases}$$

Then the formula for attack cost in (14) implies

$$H(\bar{a}, r) - H(a^*, r) = \gamma_{l_1} \min\{r_{l_1} R, c_0\} - \gamma_{l_2} \min\{r_{l_2} R, c_0\} < 0,$$

which contradicts the assumption that $a^*$ is optimal against $r$. Hence $\beta^*_a(r) \subset \mathcal{A}^*_0(r)$, that is, (17) holds.

If additionally (15) holds, the same analysis with $a^*_{l_2} < c_0$ shows that $\beta^*_a(r) \subset \mathcal{A}^*_1(r)$. Meanwhile, the formula for attack cost in (16) implies that all attack actions from $\mathcal{A}^*_1(r)$ yield the same attack cost which is the sum of the $L - \lfloor A/c_0 \rfloor$ smallest $\gamma_l \min\{r_l R, c_0\}$. Hence $\beta^*_a(r) = \mathcal{A}^*_1(r)$, that is, (19) holds. □

In the reminder of this section, we investigate Stackelberg equilibrium routing actions and Stackelberg routing costs for the nonzero-sum game $(\mathcal{R}, \mathcal{A}^*, J, H)$, assuming that all links have same capacity $c_0$ and the attack priorities satisfy a condition. We construct explicit formulae for a strong Stackelberg equilibrium routing action and for weak $\varepsilon$ Stackelberg routing actions with arbitrarily small $\varepsilon > 0$. Our results shows that there is no weak Stackelberg equilibrium routing action unless $\gamma_l$ are the same for all links, in which case the strong Stackelberg equilibrium routing action is also a weak one. In practice, a pessimistic router would either play the weak Stackelberg equilibrium routing action when all $\gamma_l$ are the same, or a weak $\varepsilon$ Stackelberg routing action with a small $\varepsilon$ when they are not, with the understanding that there will be an $\varepsilon$ penalty in the latter case.

First, we construct the strong Stackelberg equilibrium routing action and the strong Stackelberg routing cost.

**Theorem 3.** *Consider the case of equal link capacities $c_l = c_0$ for all $l \in \mathcal{L}$ and attack priorities $\gamma_l$ such that*

$$\frac{1/\gamma_l}{\sum_{\bar{l} \in \mathcal{L}} 1/\gamma_{\bar{l}}} < \frac{c_0 - a_{\text{rem}}}{R} \qquad \forall\, l \in \mathcal{L}. \tag{20}$$

*For the game $(\mathcal{R}, \mathcal{A}^*, J, H)$, there is a strong Stackelberg equilibrium routing action $r^* \in \mathcal{R}$ defined by*

$$r^*_l := \frac{1/\gamma_l}{\sum_{\bar{l} \in \mathcal{L}} 1/\gamma_{\bar{l}}}, \qquad l \in \mathcal{L}, \tag{21}$$

*and the strong Stackelberg routing cost is given by*

$$J^* := \min_{a \in \beta^*_a(r^*)} J(r^*, a) = -\sum_{l \in \mathcal{L}^r_1} r^*_l R, \tag{22}$$

*where* $\mathcal{L}_1^r$ *is a subset of* $L - \lfloor A/c_0 \rfloor$ *links* $l \in \mathcal{L}$ *with the largest* $r_l^*$*, that is, there exists a permutation* $(l_1^r, \ldots, l_L^r)$ *of* $\mathcal{L}$ *such that*

$$r_{l_1^r}^* \geq \cdots \geq r_{l_L^r}^*,$$

*and*

$$\mathcal{L}_1^r = \{l_1^r, \ldots, l_{L-\lfloor A/c_0 \rfloor}^r\}.$$

*If additionally the attack budget* $A \geq c_0$*, then* $r^*$ *defined by* (21) *is the unique strong Stackelberg equilibrium routing action.*

Before proving Theorem 3, we observe that the routing action $r^*$ defined by (21) satisfies (15) with $r^*$ in place of $r$ due to (20). Moreover, we have

$$\gamma_l \min\{r_l^* R,\, c_0\} = \frac{R}{\sum_{\bar{l} \in \mathcal{L}} 1/\gamma_{\bar{l}}} \qquad \forall\, l \in \mathcal{L},$$

which, combined with (19), implies

$$\beta_a^*(r^*) = \mathcal{A}^*. \tag{23}$$

Therefore, if the router plays $r^*$, then all attack actions $a \in \mathcal{A}^*$ will yield the same attack cost

$$H^* := H(a, r^*) = \frac{(L - \lfloor A/c_0 \rfloor)R}{\sum_{l \in \mathcal{L}} 1/\gamma_l}$$

given by the formula for attack cost in (16).

*Proof of Theorem 3.* Clearly, $r^*$ defined by (21) satisfies $r^* \in \mathcal{R}$. Then the equality in (22) follows from (23) and the formula for routing cost in (16).

Consider an arbitrary $r \in \mathcal{R}$ such that $r \neq r^*$, and let

$$\mathcal{L}_1 := \{l \in \mathcal{L} : r_l < r_l^*\}, \qquad \mathcal{L}_2 := \{l \in \mathcal{L} : r_l \geq r_l^*\}.$$

Then

$$r_l R < r_l^* R < c_0 - a_{\text{rem}} \qquad \forall\, l \in \mathcal{L}_1 \tag{24}$$

and

$$\gamma_{l_1} r_{l_1} < \frac{1}{\sum_{l \in \mathcal{L}} 1/\gamma_l} \leq \gamma_{l_2} r_{l_2} \qquad \forall\, l_1 \in \mathcal{L}_1, \forall\, l_2 \in \mathcal{L}_2. \tag{25}$$

Note that (15) may not holds for $r$ since $\min\{r_l R,\, c_0\} \leq c_0 - a_{\text{rem}}$ may not hold for $l \in \mathcal{L}_2$. Consider an arbitrary optimal attack action $a \in \beta_a^*(r)$ against $r$. Then there are two possibilities.

1. If $|\mathcal{L}_1| \geq L - \lfloor A/c_0 \rfloor$, then from (17) and (25) we have $\{l \in \mathcal{L} : a_l = 0\} \subset \mathcal{L}_1$. If (15) holds, then from (19) and (25) we have $\{l \in \mathcal{L} : a_l < c_0\} \subset \mathcal{L}_1$. Next, we show that, even if (15) does not hold, the link $l_0(a)$ on which $a_{l_0(a)} = a_{\text{rem}} \in (0, c_0)$ still satisfies $l_0(a) \in \mathcal{L}_1$. Indeed, suppose $l_0(a) \in \mathcal{L}_2$. Then the formula for attack cost in (14) implies

$$H(a,r) = \gamma_{l_0(a)} \min\{r_{l_0(a)}R,\, c_0 - a_{\rm rem}\} + \sum_{l\in\mathcal{L}:a_l=0} \gamma_l r_l R.$$

Meanwhile, as $|\mathcal{L}_1| \geq L - \lfloor A/c_0 \rfloor$, there is at least one link $\bar{l}_1 \in \mathcal{L}_1$ on which $a_{\bar{l}_1} = c_0$. We define an attack action $\bar{a}$ by switching the attack rates on $l_0(a)$ and $\bar{l}_1$, that is,

$$\bar{a}_l := \begin{cases} a_{\rm rem}, & l = \bar{l}_1, \\ c_0, & l = l_0(a), \\ a_l, & l \in \mathcal{L}\backslash\{\bar{l}_1,\, l_0(a)\}. \end{cases}$$

Then

$$H(\bar{a},r) = \gamma_{\bar{l}_1} r_{\bar{l}_1} R + \sum_{l\in\mathcal{L}:a_l=0} \gamma_l r_l R < H(a,r),$$

where the inequality follows from (24) and (25), which contradicts the assumption that $a$ is optimal against $r$. Hence $\{l \in \mathcal{L} : a_l < c_0\} \subset \mathcal{L}_1$ holds regardless of whether (15) holds. Let $\bar{\mathcal{L}}_1$ be a subset of $L - \lfloor A/c_0 \rfloor$ links $l \in \mathcal{L}_1$ with the largest $r_l$. Then the formula for routing cost in (14), together with (24), implies

$$J(r,a) \geq -\sum_{l\in\bar{\mathcal{L}}_1} r_l R > -\sum_{l\in\bar{\mathcal{L}}_1} r_l^* R \geq -\sum_{l\in\mathcal{L}_1^r} r_l^* R = J^*.$$

2. If $|\mathcal{L}_1| < L - \lfloor A/c_0 \rfloor$, then from (17) and (25) we have $\mathcal{L}_1 \subset \{l \in \mathcal{L} : a_l < c_0\}$. In the following, we assume that there is a link $l_0(a) \in \mathcal{L}_2$ on which $a_{l_0(a)} = a_{\rm rem} \in (0, c_0)$, and let $\bar{\mathcal{L}}_2$ be a subset of $L - \lceil A/c_0 \rceil - |\mathcal{L}_1|$ links $l \in \mathcal{L}_2$ with the largest $r_l$. (The cases where $l_0(a) \in \mathcal{L}_1$ or $l_0(a)$ does not exists can be proved along the same lines while removing the terms related to $l_0(a)$ and letting $\bar{\mathcal{L}}_2$ be a subset of $L - \lfloor A/c_0 \rfloor - |\mathcal{L}_1|$ links $l \in \mathcal{L}_2$ with the largest $r_l$. In particular, if $l_0(a) \in \mathcal{L}_1$ then (24) implies $r_{l_0(a)}R < c_0 - a_{\rm rem}$.) From (4b) we have

$$\begin{aligned} -\sum_{l\in\mathcal{L}_1} (r_l - r_l^*)R &= \sum_{l\in\mathcal{L}_2} (r_l - r_l^*)R \geq \big(r_{l_0(a)} - r_{l_0(a)}^*\big)R + \sum_{l\in\bar{\mathcal{L}}_2} (r_l - r_l^*)R \\ &\geq \big(\min\{r_{l_0(a)}R, c_0 - a_{\rm rem}\} - r_{l_0(a)}^* R\big) + \sum_{l\in\bar{\mathcal{L}}_2} (\min\{r_l R,\, c_0\} - r_l^* R), \end{aligned}$$

where the last inequality is strict if $r_{l_0(a)}R > c_0 - a_{\rm rem}$. Hence the formula for routing cost in (14) implies

$$
\begin{aligned}
J(r,a) &\geq -\min\{r_{l_0(a)}R, c_0 - a_{\rm rem}\} - \sum_{l\in\mathcal{L}_1} r_l R - \sum_{l\in\bar{\mathcal{L}}_2} \min\{r_l R,\, c_0\} \\
&= -\big(\min\{r_{l_0(a)}R, c_0 - a_{\rm rem}\} - r^*_{l_0(a)}R\big) - \sum_{l\in\mathcal{L}_1}(r_l - r^*_l)R \\
&\quad - \sum_{l\in\bar{\mathcal{L}}_2}(\min\{r_l R,\, c_0\} - r^*_l R) - r^*_{l_0(a)}R - \sum_{l\in\mathcal{L}_1} r^*_l R - \sum_{l\in\bar{\mathcal{L}}_2} r^*_l R \\
&\geq -r^*_{l_0(a)}R - \sum_{l\in\mathcal{L}_1} r^*_l R - \sum_{l\in\bar{\mathcal{L}}_2} r^*_l R \\
&\geq -\sum_{l\in\mathcal{L}_1^r} r^*_l R = J^*,
\end{aligned}
$$

where the second inequality is strict if $r_{l_0(a)}R > c_0 - a_{\rm rem}$.
Next, we show that if the attack budget $A \geq c_0$, then we have $J(r,a) > J^*$ even if $r_{l_0(a)}R \leq c_0 - a_{\rm rem}$. Indeed, as $r_{l_0(a)}R \leq c_0 - a_{\rm rem}$, there is no congestion on $l_0(a)$. Hence the routing and attack costs are given by the corresponding formulae in (16), and thus similar analysis to the proof of Corollary 2 shows that $a \in \mathcal{A}_1^*(r)$ defined in (19). Then as $A \geq c_0$, there is a link

$$
\bar{l}_2 \in \arg\max_{l\in\mathcal{L}} \gamma_l \min\{r_l R,\, c_0\}
$$

on which $a_{\bar{l}_2} = c_0$. Moreover, we have $r_{\bar{l}_2} > r^*_{\bar{l}_2}$ as $r \neq r^*$; thus $\bar{l}_2 \in \mathcal{L}_2$. Let $\bar{\mathcal{L}}_2$ be a subset of $L - \lfloor A/c_0 \rfloor - |\mathcal{L}_1|$ links $l \in \mathcal{L}_2 \backslash \{\bar{l}_2\}$ with the largest $r_l$. Then (4b) implies

$$
\begin{aligned}
-\sum_{l\in\mathcal{L}_1}(r_l - r^*_l)R &> \sum_{l\in\mathcal{L}_2\backslash\{\bar{l}_2\}}(r_l - r^*_l)R \\
&\geq \sum_{l\in\bar{\mathcal{L}}_2}(r_l - r^*_l)R \geq \sum_{l\in\bar{\mathcal{L}}_2}(\min\{r_l R,\, c_0\} - r^*_l R).
\end{aligned}
$$

Hence the formula for routing cost in (16) implies

$$
\begin{aligned}
J(r,a) &\geq -\sum_{l\in\mathcal{L}_1} r_l R - \sum_{l\in\bar{\mathcal{L}}_2} \min\{r_l R,\, c_0\} \\
&= -\sum_{l\in\mathcal{L}_1}(r_l - r^*_l)R - \sum_{l\in\bar{\mathcal{L}}_2}(\min\{r_l R,\, c_0\} - r^*_l R) \\
&\quad - \sum_{l\in\mathcal{L}_1} r^*_l R - \sum_{l\in\bar{\mathcal{L}}_2} r^*_l R \\
&> -\sum_{l\in\mathcal{L}_1} r^*_l R - \sum_{l\in\bar{\mathcal{L}}_2} r^*_l R \\
&\geq -\sum_{l\in\mathcal{L}_1^r} r^*_l R = J^*.
\end{aligned}
$$

In summary, we have established that

$$J^* = \min_{a \in \beta_a^*(r^*)} J(r^*, a) \leq \min_{a \in \beta_a^*(r)} J(r, a) \qquad \forall\, r \neq r^*; \tag{26}$$

thus $r^*$ is a strong Stackelberg equilibrium routing action and $J^*$ is the strong Stackelberg routing cost for $(\mathcal{R}, \mathcal{A}^*, J, H)$. If additionally the attack budget $A \geq c_0$, then the inequality in (26) is strict; thus $r^*$ is the unique strong Stackelberg equilibrium routing action. □

Next, we establish that $J^*$ defined in (22) is also the weak Stackelberg routing cost for $(\mathcal{R}, \mathcal{A}^*, J, H)$, by constructing a weak $\varepsilon$ Stackelberg routing action for an arbitrarily small $\varepsilon > 0$; cf. [43, Section 6] for a related result for finite games with mixed strategies.

**Theorem 4.** *Consider the case of equal link capacities $c_l = c_0$ for all $l \in \mathcal{L}$ and attack priorities $\gamma_l$ such that* (20) *holds. For the game $(\mathcal{R}, \mathcal{A}^*, J, H)$, we have*

1. *for an arbitrary $\varepsilon > 0$ such that*

$$\varepsilon < \min\{(L - \lfloor A/c_0 \rfloor)\, r_l^*,\ \lfloor A/c_0 \rfloor((c_0 - a_{\mathrm{rem}})/R - r_l^*)\} \qquad \forall\, l \in \mathcal{L}, \tag{27}$$

*where $r^*$ is the routing action defined by* (21)*, there is a weak $\varepsilon$ Stackelberg routing action $r^\varepsilon \in \mathcal{R}$ defined by*

$$r_l^\varepsilon := \begin{cases} r_l^* - \varepsilon/(L - \lfloor A/c_0 \rfloor), & l \in \mathcal{L}_1^r, \\ r_l^* + \varepsilon/\lfloor A/c_0 \rfloor, & l \in \mathcal{L}_2^r, \end{cases} \tag{28}$$

*where $\mathcal{L}_1^r$ is a subset of $L - \lfloor A/c_0 \rfloor$ links $l \in \mathcal{L}$ with the largest $r_l^*$ as in* (22) *and $\mathcal{L}_2^r := \mathcal{L} \backslash \mathcal{L}_1^r$, and*
2. *the weak Stackelberg routing cost is equal to the strong Stackelberg routing cost $J^*$ given by* (22)*.*

*Proof.* The condition (27) ensures that $r^\varepsilon$ defined by (28) satisfies $r^\varepsilon \in \mathcal{R}$ and

$$r_l^\varepsilon < \frac{c_0 - a_{\mathrm{rem}}}{R} \qquad \forall\, l \in \mathcal{L},$$

and thus (15) holds with $r^\varepsilon$ in place of $r$. Moreover, combining (21) and (28) yields

$$\gamma_{l_1} r_{l_1}^\varepsilon < \frac{1}{\sum_{l \in \mathcal{L}} 1/\gamma_l} < \gamma_{l_2} r_{l_2}^\varepsilon \qquad \forall\, l_1 \in \mathcal{L}_1^r, \forall\, l_2 \in \mathcal{L}_2^r.$$

Then (19) implies that the set of optimal attack actions is given by

$$\beta_a^*(r^\varepsilon) = \{a \in \mathcal{A}^* : \{l \in \mathcal{L} : a_l = c_0\} = \mathcal{L}_2^r\}.$$

Hence the formula for routing cost in (16) implies

$$J(r^\varepsilon, a) = -\sum_{l\in\mathcal{L}_1^r} r_l^\varepsilon R = \varepsilon - \sum_{l\in\mathcal{L}_1^r} r_l^* R = J^* + \varepsilon \qquad \forall\, a \in \beta_a^*(r^\varepsilon).$$

As $\varepsilon > 0$ can be arbitrarily small, we have

$$J^* \geq \inf_{r\in\mathcal{R}} \max_{a\in\beta_a^*(r)} J(r, a).$$

Additionally, Theorem 3 implies

$$J^* = \min_{r\in\mathcal{R}} \min_{a\in\beta_a^*(r)} J(r, a) \leq \inf_{r\in\mathcal{R}} \max_{a\in\beta_a^*(r)} J(r, a).$$

Hence

$$J^* = \inf_{r\in\mathcal{R}} \max_{a\in\beta_a^*(r)} J(r, a),$$

that is, $J^*$ is the weak Stackelberg routing cost and $r^\varepsilon$ is a weak $\varepsilon$ Stackelberg routing action for $(\mathcal{R}, \mathcal{A}^*, J, H)$. □

Based on (23), all attack actions from $\mathcal{A}^*$ are optimal against the strong Stackelberg equilibrium routing action $r^*$ defined by (21). Hence $r^*$ cannot be a weak Stackelberg equilibrium routing action unless all attack actions from $\mathcal{A}^*$ also yield the same routing cost. Combining (21) and the formula for routing cost in (16), we see that is the case if and only if the adversary has the same priority for all links. Moreover, if the adversary has the same priority for all links, then (20) can be replace by a less restrictive condition; the proof is along the lines of that of Theorem 3 and thus omitted here.

**Corollary 5.** *Consider the case of equal link capacities $c_l = c_0$ for all $l \in \mathcal{L}$ and attack priorities such that* (20) *holds. The routing action $r^*$ defined by* (21) *is a weak Stackelberg equilibrium routing action for the game $(\mathcal{R}, \mathcal{A}^*, J, H)$ if and only if there is a constant $\gamma_0 \in (0, 1]$ such that*

$$\gamma_l = \gamma_0 \qquad \forall\, l \in \mathcal{L}. \tag{29}$$

*Moreover, if* (29) *holds, then* (20) *can be replaced by*

$$R < Lc_0$$

*and $r^*$ defined by* (21)*, that is,*

$$r_l^* = R/L, \qquad l \in \mathcal{L},$$

*is still both a strong Stackelberg equilibrium routing action and a weak one.*

## 5 Mitigating attacks via learning

We now focus our attention on scenarios where the "adversary" may be driven by a cost more general than either (3) or (9) and, in fact, the router does not

know the attack cost function $H$ or even the attack action set $\mathcal{A}$, leading to a game with incomplete information. In this scenario, the attack strategy is a *best-response function* $f^* : \mathcal{R} \to \mathcal{A}$ that satisfies

$$f^*(r) \in \beta_a(r) \qquad \forall\, r \in \mathcal{R},$$

but the function $f^*$ (and the set-valued function $\beta_a$ as well) is unknown to the router. Since the router cannot predict the attack action $f^*(r)$ or the routing cost $J(r, f^*(r))$ associated with a specific routing action $r$, it constructs a prediction of $J(r, f^*(r))$ via a learning-based approach that consists of two components:

1. a neural network trained to construct a prediction $\hat{J}(r)$ of the actual routing cost $J(r, f^*(r))$ that results from a routing action $r \in \mathcal{R}$, and
2. a gradient descent algorithm used to minimize the predicted routing cost $\hat{J}(r)$.

### 5.1 Predicting the routing cost

We train a neural network

$$y = \mathrm{NN}(\theta, r)$$

to predict the routing cost $J(r, f^*(r))$ that results from a routing action $r \in \mathcal{R}$, where $\theta$ denotes the parameters of the neural network. The input to the neural network is the routing action $r$; thus there are $L$ neurons in the input layer. Based on this input, the neural network predicts the actual user rates $u$ as defined by (1) and the resulting routing cost $J(r, f^*(r))$ as defined by (2); thus there are $L + 1$ neurons in the output layer. We use $n$ fully connected hidden layers, each of which consists of $m$ *rectified linear units (ReLU)*. Therefore, the input $x^i \in \mathbb{R}^m$ to the $i$-th hidden layer is given by

$$x^i = \mathrm{ReLU}(\theta^i_{\text{weight}} x^{i-1} + \theta^i_{\text{bias}}), \qquad i \in \{1, \ldots, n\},$$

where $x^0 = r \in \mathcal{R}$ is the input and ReLU is the *rectifier* function

$$\mathrm{ReLU}(x) = \max\{x,\, 0\},$$

in which the maximum is taken in each scalar component. The output of the neural network $y \in \mathbb{R}^{L+1}$ is given by

$$y = \theta^{n+1}_{\text{weight}} x^n + \theta^{n+1}_{\text{bias}}.$$

The neural network is visualized in Fig. 1 below. It is trained using a history of past routing actions and the corresponding values of actual user rates and routing costs. It is important to highlight that this neural network can be trained without observing the attack actions $a$ and with no input of the total desired transmission rate of legitimate traffic $R$.

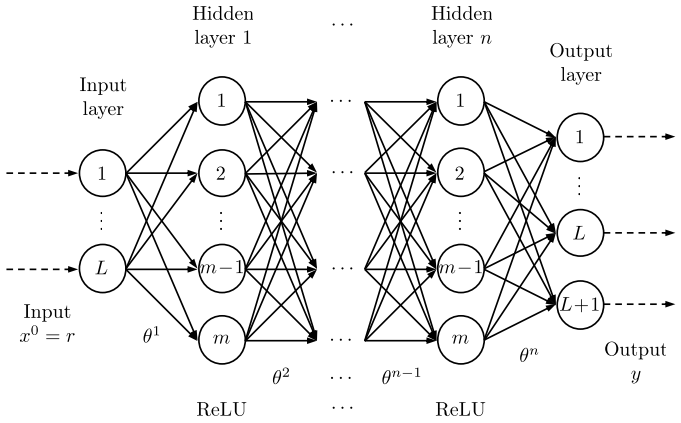



**Fig. 1.** The neural network used to predict the routing cost. There are $L$ neurons in the input layer, $L+1$ neurons in the output layer, and $n$ fully connected hidden layers each with $m$ ReLU neurons.

### 5.2 Minimizing the predicted routing cost

After training, the router adjusts the routing action $r \in \mathcal{R}$ based on a gradient descent algorithm to minimize the *predicted routing cost* $\hat{J}(r) = y_{L+1}$, which is the last component of the output $y = \mathrm{NN}(\theta, r)$ of the neural network. To specify the gradient descent algorithm, we recall the following notions and basic properties from convex analysis; for more details, see, e.g., [46, Section 6] or [47, Section 5.1].

For a closed convex set $\mathcal{C} \subset \mathbb{R}^n$ and a point $v \in \mathbb{R}^n$, we denote by $[v]_{\mathcal{C}}$ the *projection* of $v$ onto $\mathcal{C}$, that is,

$$[v]_{\mathcal{C}} := \operatorname*{arg\,min}_{w \in \mathcal{C}} \|w - v\|.$$

The projection $[v]_{\mathcal{C}}$ satisfies $[v]_{\mathcal{C}} = v$ if $v \in \mathcal{C}$. For a convex set $\mathcal{S} \subset \mathbb{R}^n$ and a point $x \in \mathcal{S}$, we denote by $T_{\mathcal{S}}(x)$ the *tangent cone* to $\mathcal{S}$ at $x$, that is,

$$T_{\mathcal{S}}(x) := \overline{\{h(z-x) : z \in \mathcal{S},\, h > 0\}}.$$

The set $T_{\mathcal{S}}(x)$ is closed and convex, and satisfy $T_{\mathcal{S}}(x) = \mathbb{R}^n$ if $x \in \mathcal{S} \backslash \partial\mathcal{S}$.

The gradient descent algorithm used to minimize the predict routing cost $\hat{J}(r) = y_{L+1}$ is

$$\dot{r} = \big[-\lambda \nabla_r \hat{J}(r)\big]_{T_{\mathcal{R}}(r)} = \big[-\lambda \nabla_{x^n} y_{L+1} \nabla_{x^{n-1}} x^n \cdots \nabla_{x^0} x^1\big]_{T_{\mathcal{R}}(r)}, \tag{30}$$

where $\lambda > 0$ is a preselected constant, and the projection $[\cdot]_{T_{\mathcal{R}}(r)}$ onto the tangent cone $T_{\mathcal{R}}(r)$ is used to guarantee that the routing action $r$ remains inside the routing action set $\mathcal{R}$. Note that it is particularly convenient to use a gradient descent method in our approach, as the gradients $\nabla_{x^n} y_{L+1}$, $\nabla_{x^{n-1}} x^n$, ..., and $\nabla_{x^0} x^1$ are readily available from backpropagation during training.

## 6 Simulation study

In this section, we present simulation results for the learning-based approach described in Section 5 against the malicious adversary described in Section 4.1, for networks with up to 10 parallel links.

In the simulation, all links have the same capacity $c_0 = 1$, the total desired user rate $R = L/2$, and the attack budget $A = \lfloor L/2 \rfloor$. For cases where the adversary has the same priority for all links, we use the attack priority vector $\gamma = \mathbf{1}_L$; otherwise $\gamma \in (0, 1]^L$ will be specified.

In this problem, a router that mistakenly believes the adversary is playing a constant attack action will regret its choice no matter which routing action is played; thus there is no Nash equilibrium for the game $(\mathcal{R}, \mathcal{A}^*, J, H)$ [17, p. 106]. The results in Section 4.1 show that there is a strong Stackelberg equilibrium routing action given by (21) but not necessarily a weak Stackelberg equilibrium routing action, while the strong and weak Stackelberg routing costs are both $J^*$ given by (22). However, even if the adversary always plays an optimal attack action that is worst for the router, the router is able to approach the weak Stackelberg routing cost $J^*$ using a weak $\varepsilon$ Stackelberg routing action $r^\varepsilon$ closed to $r^*$ given by (28).

In each simulation, a neural network is constructed using Python 3.7.7 and PyTorch 1.3.1. It has $n = 3$ fully connected hidden layers, each with $m$ ReLU neurons. One of the focus for this simulation study is to construct the predicted routing cost $\hat{J}(r)$ using a small set of training data. Therefore, we train the neural network using only 500 samples with randomly generated routing actions (sampled from the 101-st to the 600-th unit of time). The neural network is trained for 2000 epochs for each simulation, with the batch size 80 and the learning rate 0.001. After training, the gradient algorithm (30) is applied to minimize the predicted routing cost $\hat{J}(r)$ generated by the neural network, until the $T = 10000$-th unit of time. The constant in (30) is set by $\lambda = 0.001$. The simulation results are shown in Fig. 2–13 below. In Fig. 2, 4, 6, 8, 10, 11, 12, and 13, the horizontal axis is in $\times 10^4$ units of time.

### 6.1 Discussion

In Table 1 below, we summarize the simulation parameters and final results at time $T$, including the number of links $L$, the attack priority vector $\gamma$, the number of neurons in hidden layers $m$, the strong Stackelberg equilibrium routing action $r^*$ defined by (21), the weak Stackelberg routing cost $J^*$ defined by (22), the final actual and predicted routing costs $J(T) := J(r(T), f^*(r(T)))$ and $\hat{J}(T) := \hat{J}(r(T))$, the relative difference $|J(T)/J^* - 1|$, the percentage prediction error $|\hat{J}(T)/J(T) - 1|$, and the corresponding figures.

Based on Table 1 and Fig. 2–13, we make the following observations:

1. Even with a small set of training data (500 samples), our learning-based approach is able to provide a routing cost $J(T)$ that is reasonably close to the weak Stackelberg routing cost $J^*$.

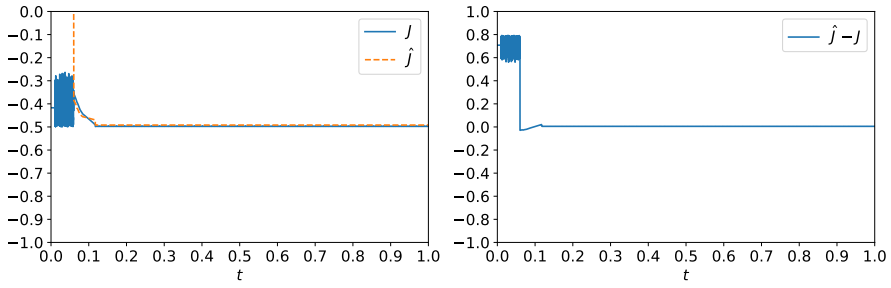
**Table 1.** Simulation parameters and results.

| $L$ | $\gamma$ | $m$ | $r^*$ | $J^*$ | $J(T)$ | $\hat{J}(T)$ | $\left\lvert\frac{J(T)}{J^*}-1\right\rvert$ | $\left\lvert\frac{\hat{J}(T)}{J(T)}-1\right\rvert$ | Fig. |
|---|---|---|---|---|---|---|---|---|---|
| 2 | $\mathbf{1}_2$ | 16 | $\frac{1}{2}\mathbf{1}_2$ | $-\frac{1}{2}$ | $-0.50$ | $-0.49$ | 0.60% | 1.04% | 2, 3 |
| 2 | $\left(\frac{2}{3},1\right)$ | 32 | $\left(\frac{3}{5},\frac{2}{5}\right)$ | $-\frac{3}{5}$ | $-0.60$ | $-0.58$ | 0.77% | 3.40% | 4, 5 |
| 3 | $\mathbf{1}_3$ | 16 | $\frac{1}{3}\mathbf{1}_3$ | $-\frac{1}{2}$ | $-0.48$ | $-0.47$ | 3.52% | 3.39% | 6, 7 |
| 3 | $\left(1,\frac{3}{4},1\right)$ | 32 | $\left(\frac{3}{10},\frac{2}{5},\frac{3}{10}\right)$ | $-\frac{3}{5}$ | $-0.56$ | $-0.54$ | 6.26% | 3.85% | 8, 9 |
| 4 | $\mathbf{1}_4$ | 32 | $\frac{1}{4}\mathbf{1}_4$ | $-1$ | $-0.96$ | $-0.92$ | 4.27% | 3.85% | 10 |
| 4 | $\left(\frac{1}{2},1,1,1\right)$ | 64 | $\left(\frac{2}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5}\right)$ | $-\frac{6}{5}$ | $-1.15$ | $-1.12$ | 3.79% | 3.05% | 11 |
| 6 | $\mathbf{1}_6$ | 64 | $\frac{1}{6}\mathbf{1}_6$ | $-\frac{3}{2}$ | $-1.46$ | $-1.38$ | 2.58% | 5.57% | 12 |
| 10 | $\mathbf{1}_{10}$ | 128 | $\frac{1}{10}\mathbf{1}_{10}$ | $-\frac{5}{2}$ | $-2.30$ | $-2.19$ | 7.94% | 4.96% | 13 |

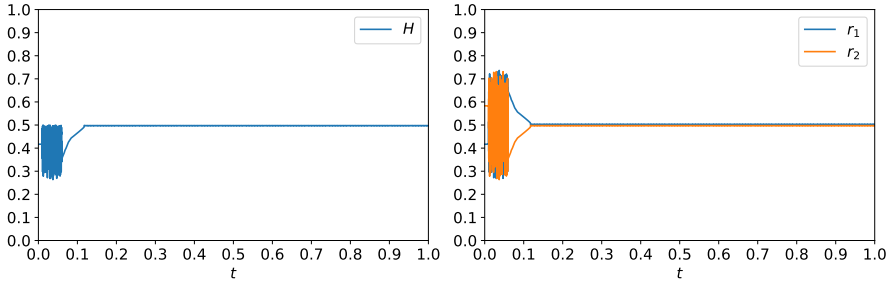
2. Figures 5 and 9 demonstrate that our neural networks succeed at approximating the routing cost in spite of strong discontinuities in the function $J(r, f^*(r))$ that result from nonunique optimal attack actions against a strong Stackelberg equilibrium routing action. However, this is clearly a challenge for high-dimensional problems and motivates the need for further research.
3. In practice, the selection of the routing action $r$ only cares about minimizing the last component of the neural network's output (the predicted routing cost $\hat{J}(r)$ above). However, we found that the performance of the neural network improved significantly when it was trained to also predict the actual user rates $u$. Our conjecture is that the additional dimensions in training data force the hidden layers to "respond" to the actual user rates and thus provides a more persistent structure for the neural network. This phenomenon will also be a topic for future research.

## 7 Conclusion

We formulated the mitigation of link-flooding DDoS attacks as a routing problem among parallel links. A Stackelberg game model was constructed to address the challenge that the adversary can observe the routing strategy before assigning attack traffic. For a general class of adversaries, we characterized an optimal attack that belongs to a finite set, and constructed explicit formulae for Stackelberg equilibria and costs for a special class of networks. For the more general case of unknown attack cost and capacity, we proposed a learning-based approach that predicts the routing cost using a neural network and then minimizes the predicted routing cost via projected gradient descent.




(a) Actual and predicted routing costs $J, \hat{J}$ (b) Prediction error $\hat{J} - J$




(c) Attack cost $H$ (d) Routing action $r$

**Fig. 2.** Simulation results for a network of $L = 2$ links and an attack priority vector $\gamma = \mathbf{1}_2$, using $m = 16$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-0.50$ and $-0.49$, respectively, where the former is within 0.60% from the weak Stackelberg routing cost $J^* = -0.5$. The prediction error $\hat{J} - J$ converges to 0.01, which is within 1.04% from $J(T)$. The attack cost $H$ converges to 0.50. The routing action $r$ converges to $(0.50, 0.50)$, which is close to the weak Stackelberg equilibrium routing action $r^* = \mathbf{1}_2/2$.

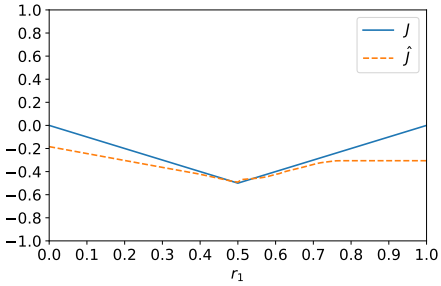
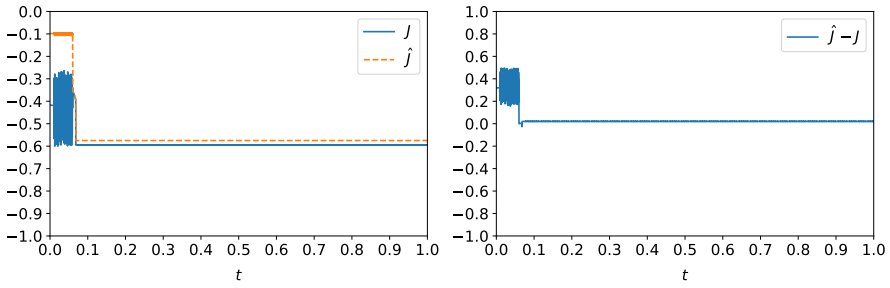
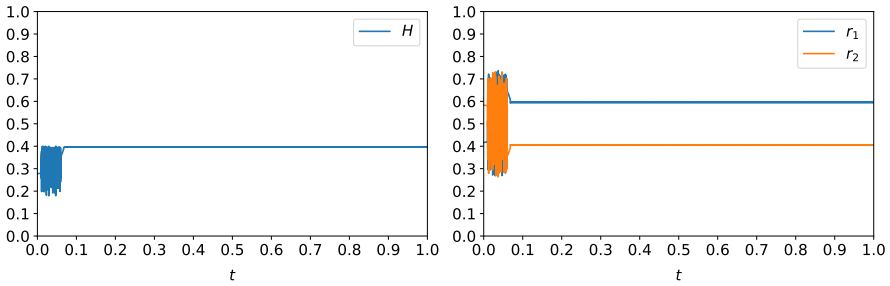



**Fig. 3.** Actual and predicted routing cost functions $J(r, f^*(r))$ and $\hat{J}(r)$ for the simulation in Fig. 2.




(a) Actual and predicted routing costs $J, \hat{J}$ (b) Prediction error $\hat{J} - J$




(c) Attack cost $H$ (d) Routing action $r$

**Fig. 4.** Simulation results for a network with $L = 2$ links and the attack priority vector $\gamma = (2/3, 1)$, using $m = 32$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-0.60$ and $-0.58$, respectively, where the former is within 0.77% from the weak Stackelberg routing cost $J^* = -3/5$. The prediction error $\hat{J} - J$ converges to 0.02, which is within 3.40% from $J(T)$. The attack cost $H$ converges to 0.40. The routing action $r$ converges to $(0.60, 0.40)$, which is close to the strong Stackelberg equilibrium routing action $r^* = (3/5, 2/5)$.

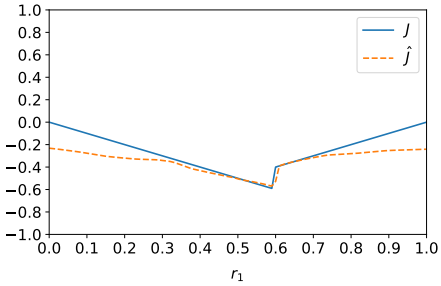
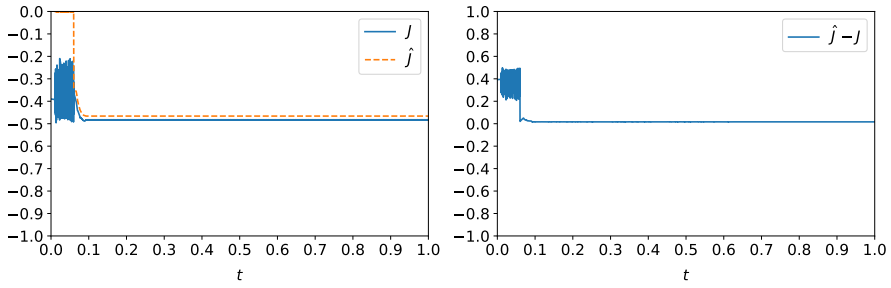


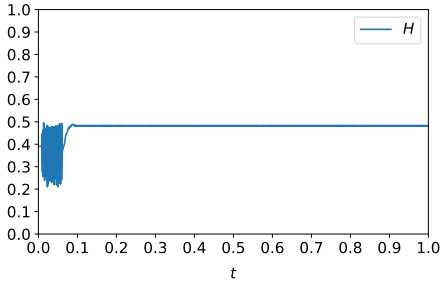

**Fig. 5.** Actual and predicted routing cost functions $J(r, f^*(r))$ and $\hat{J}(r)$ for the simulation in Fig. 4.



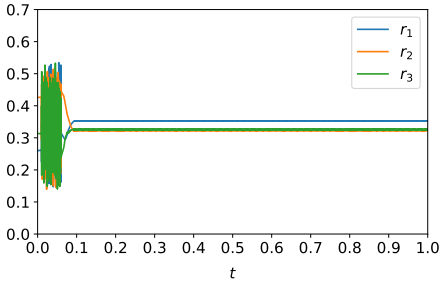

(a) Actual and predicted routing costs $J, \hat{J}$




(b) Prediction error $\hat{J} - J$




(c) Attack cost $H$




(d) Routing action $r$

**Fig. 6.** Simulation results for a network of $L = 3$ links and an attack priority vector $\gamma = \mathbf{1}_3$, using $m = 16$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-0.48$ and $-0.47$, respectively, where the former is within 3.52% from the weak Stackelberg routing cost $J^* = -1/2$. The prediction error $\hat{J} - J$ converges to 0.02, which is within 3.39% from $J(T)$. The attack cost $H$ converges to 0.48. The routing action $r$ converges to $(0.35, 0.32, 0.33)$, which is close to the weak Stackelberg equilibrium routing action $r^* = \mathbf{1}_3/3$.

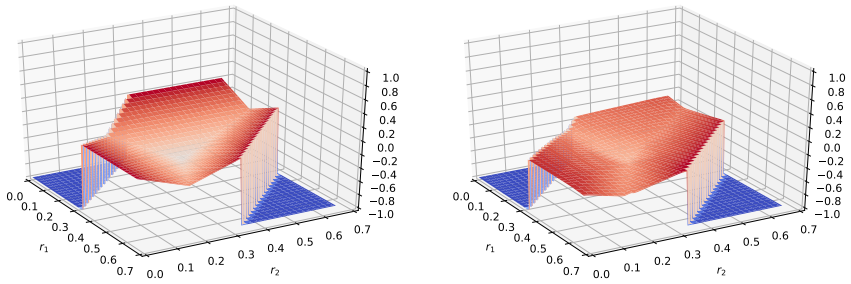
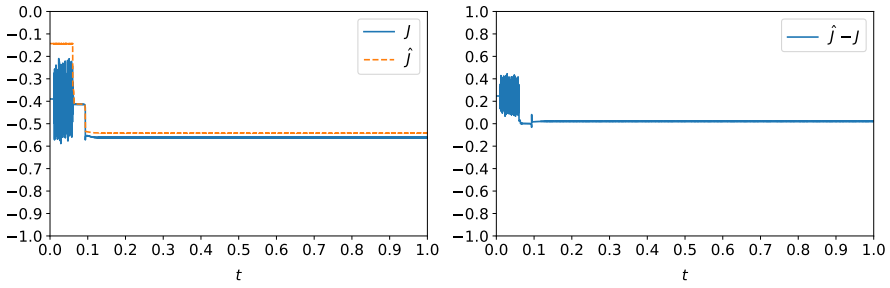
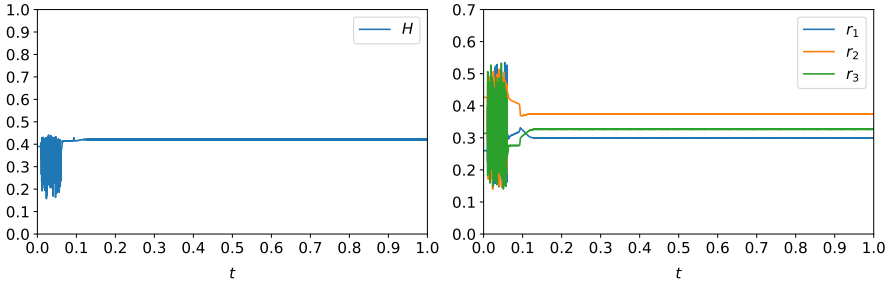



**Fig. 7.** Actual and predicted routing cost functions $J(r, f^*(r))$ and $\hat{J}(r)$ for the simulation in Fig. 6.




(a) Actual and predicted routing costs $J, \hat{J}$ (b) Prediction error $\hat{J} - J$




(c) Attack cost $H$ (d) Routing action $r$

**Fig. 8.** Simulation results for a network of $L = 3$ links and an attack priority vector $\gamma = (1, 0.5, 0.5)$, using $m = 32$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-0.56$ and $-0.54$, respectively, where the former is within 6.26% from the weak Stackelberg routing cost $J^* = -3/5$. The prediction error $\hat{J} - J$ converges to 0.02, which is within 3.85% from $J(T)$. The attack cost $H$ converges to 0.42. The routing action $r$ converges to $(0.30, 0.37, 0.33)$, which is close to the strong Stackelberg equilibrium routing action $r^* = (3/10, 2/5, 3/10)$.

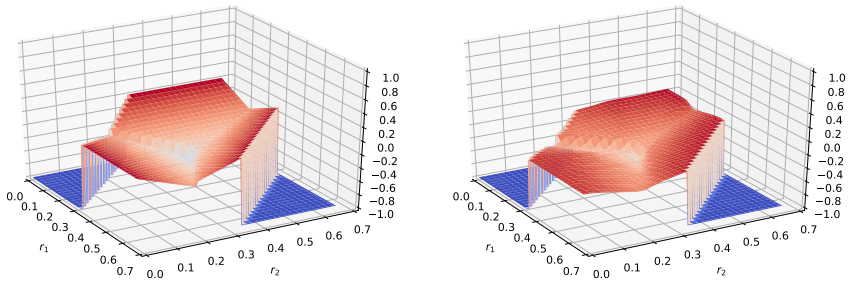
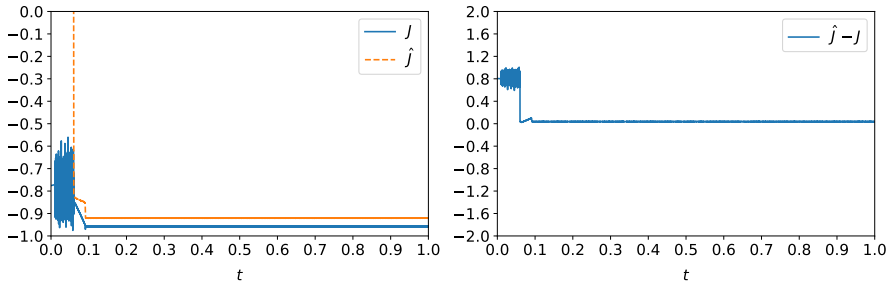


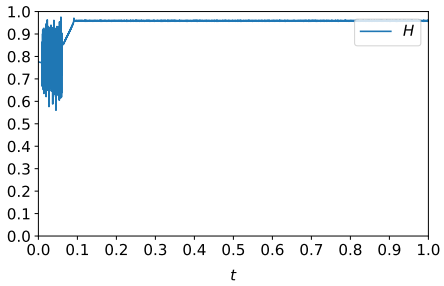

**Fig. 9.** Actual and predicted routing cost functions $J(r, f^*(r))$ and $\hat{J}(r)$ for the simulation in Fig. 8.



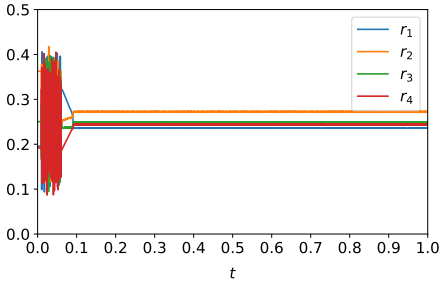

(a) Actual and predicted routing costs $J, \hat{J}$




(b) Prediction error $\hat{J} - J$




(c) Attack cost $H$

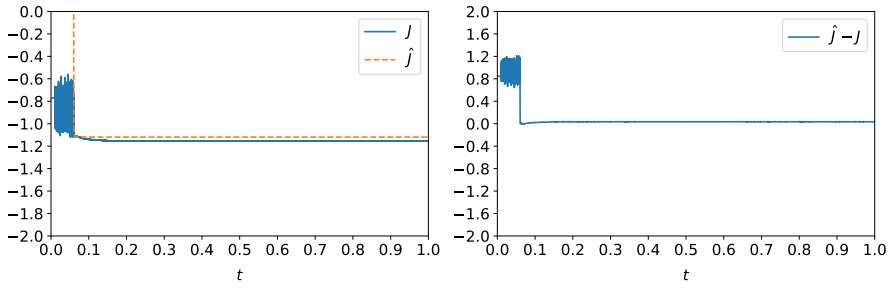



(d) Routing action $r$

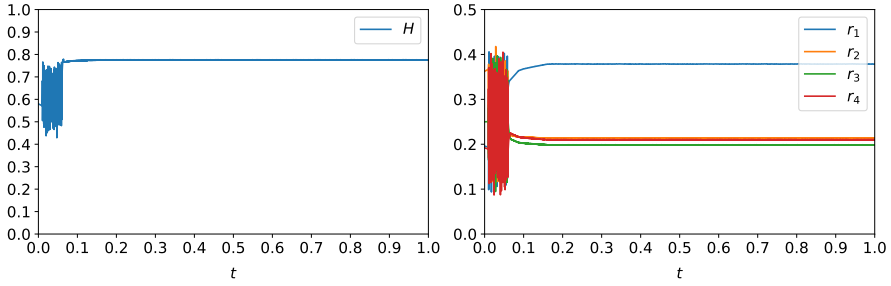
**Fig. 10.** Simulation results for a network of $L = 4$ links and an attack priority vector $\gamma = \mathbf{1}_4$, using $m = 32$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-0.96$ and $-0.92$, respectively, where the former is within 4.27% from the weak Stackelberg routing cost $J^* = -1$. The prediction error $\hat{J} - J$ converges to 0.04, which is within 3.85% from $J(T)$. The attack cost $H$ converges to 0.96. The routing action $r$ converges to $(0.24, 0.27, 0.25, 0.24)$, which is close to the weak Stackelberg equilibrium routing action $r^* = \mathbf{1}_4/4$.

The effectiveness of our approach was demonstrated through a simulation study.

A future research direction is to extend our results to scenarios where the attack objective is partially known and incorporate the additional information about attack to our learning-based approach. For example, if the attack was known to belong to the finite set $\mathcal{A}^*$ defined in (12), how should the router adjust the neural network to improve its efficiency? Other future research topics include to test our approach in more complex networks and to generalize our results to the case of multiple routers and/or adversaries.




(a) Actual and predicted routing costs $J, \hat{J}$ (b) Prediction error $\hat{J} - J$

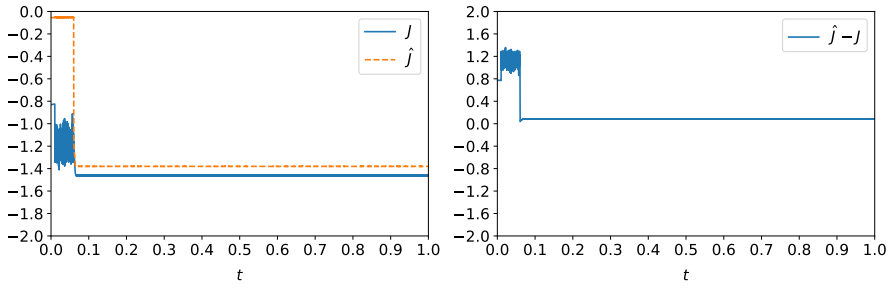



(c) Attack cost $H$ (d) Routing action $r$

**Fig. 11.** Simulation results for a network of $L = 4$ links and an attack priority vector $\gamma = (0.5, 1, 1, 1)$, using $m = 64$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-1.15$ and $-1.12$, respectively, where the former is within 3.79% from the weak Stackelberg routing cost $J^* = -6/5$. The prediction error $\hat{J} - J$ converges to 0.04, which is within 3.05% from $J(T)$. The attack cost $H$ converges to 0.78. The routing action $r$ converges to $(0.38, 0.21, 0.20, 0.21)$, which is close to the strong Stackelberg equilibrium routing action $r^* = (2/5, 1/5, 1/5, 1/5)$.

## References

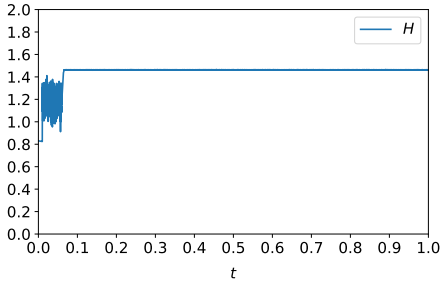
1. P. J. Criscuolo, "Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht," Lawrence Livermore National Laboratory, University of California, Tech. Rep., 2000.
2. J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, Apr. 2004.
3. S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of defense mechanisms against Distributed Denial of Service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.
4. "NETSCOUT's 14th Annual Worldwide Infrastructure Security Report," NETSCOUT Systems, Inc, Tech. Rep., 2019.



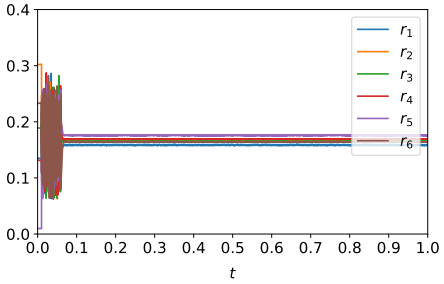

(a) Actual and predicted routing costs $J, \hat{J}$




(b) Prediction error $\hat{J} - J$



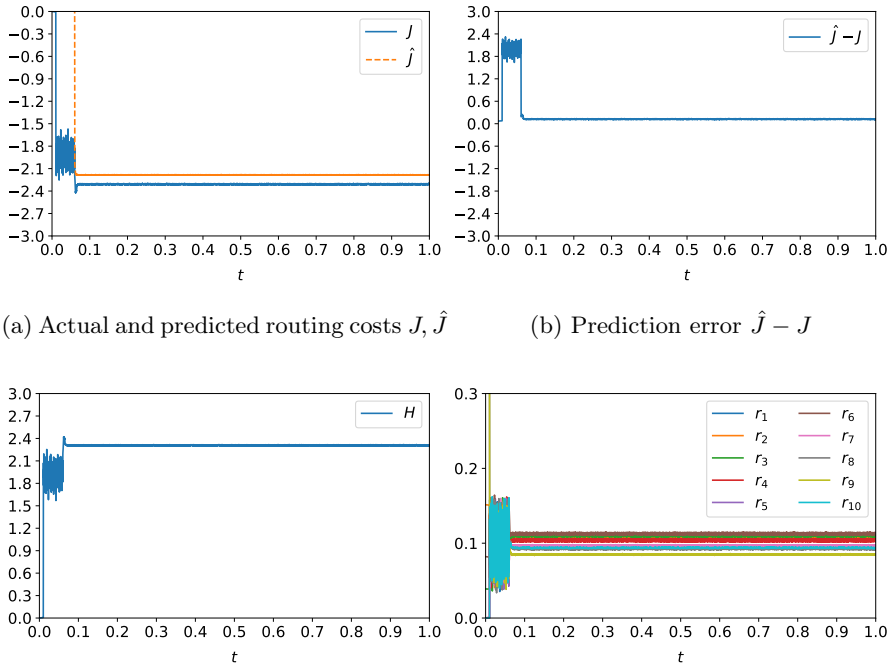

(c) Attack cost $H$




(d) Routing action $r$

**Fig. 12.** Simulation results for a network of $L = 6$ links and an attack priority vector $\gamma = \mathbf{1}_6$, using $m = 64$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-1.46$ and $-1.38$, respectively, where the former is within 2.58% from the weak Stackelberg routing cost $J^* = -3/2$. The prediction error $\hat{J} - J$ converges to 0.08, which is within 5.57% from $J(T)$. The attack cost $H$ converges to 1.46. The routing action $r$ converges to $(0.16, 0.17, 0.16, 0.17, 0.18, 0.16)$, which is close to the weak Stackelberg equilibrium routing action $r^* = \mathbf{1}_6/6$.

5. A. Studer and A. Perrig, "The Coremelt attack," in *16th European Symposium on Research in Computer Security*, 2011, pp. 37–52.
6. M. S. Kang, S. B. Lee, and V. D. Gligor, "The Crossfire attack," in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 127–141.
7. S. B. Lee, M. S. Kang, and V. D. Gligor, "CoDef: Collaborative defense against large-scale link-flooding attacks," in *9th ACM conference on Emerging Networking Experiments and Technologies*, 2013, pp. 417–428.
8. M. S. Kang, V. D. Gligor, and V. Sekar, "SPIFFY: Inducing cost-detectability tradeoffs for persistent link-flooding attacks," in *2016 Network and Distributed System Security Symposium*, 2016, pp. 1–15.
9. C. Liaskos, V. Kotronis, and X. Dimitropoulos, "A novel framework for modeling and mitigating distributed link flooding attacks," in *35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.




(a) Actual and predicted routing costs $J, \hat{J}$

(b) Prediction error $\hat{J} - J$

(c) Attack cost $H$

(d) Routing action $r$

**Fig. 13.** Simulation results for a network of $L = 10$ links and an attack priority vector $\gamma = \mathbf{1}_{10}$, using $m = 128$ ReLU neurons in each hidden layer. The actual and predicted routing costs $J$ and $\hat{J}$ converge to $-2.30$ and $-2.19$, respectively, where the former is within 7.94% from the weak Stackelberg routing cost $J^* = -5/2$. The prediction error $\hat{J} - J$ converges to 0.11, which is within 4.96% from $J(T)$. The attack cost $H$ converges to 2.30. The routing action $r$ converges to $(0.10, 0.11, 0.11, 0.10, 0.09, 0.11, 0.10, 0.09, 0.09, 0.09)$, which is close to the weak Stackelberg equilibrium routing action $r^* = \mathbf{1}_{10}/10$.

10. A. Aydeger, N. Saputro, K. Akkaya, and M. Rahman, "Mitigating Crossfire attacks using SDN-based moving target defense," in *2016 IEEE 41st Conference on Local Computer Networks*, 2016, pp. 627–630.
11. D. Gkounis, V. Kotronis, C. Liaskos, and X. Dimitropoulos, "On the interplay of link-flooding attacks and traffic engineering," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 2, pp. 5–11, Apr. 2016.
12. L. Xue, X. Ma, X. Luo, E. W. W. Chan, T. T. N. Miu, and G. Gu, "LinkScope: Toward detecting target link flooding attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2423–2438, Oct. 2018.
13. G. Yang, H. Hosseini, D. Sahabandu, A. Clark, J. P. Hespanha, and R. Poovendran, "Modeling and mitigating the Coremelt attack," in *2018 American Control Conference*, 2018, pp. 3410–3416.
14. D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

15. M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.
16. T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. SIAM, 1999.
17. J. P. Hespanha, *Noncooperative Game Theory: An Introduction for Engineers and Computer Scientists*. Princeton University Press, 2017.
18. T. Alpcan and T. Başar, *Network Security: A Decision and Game-Theoretic Approach*. Cambridge University Press, 2010.
19. X. Liang and Y. Xiao, "Game theory for network security," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 472–486, 2013.
20. C. T. Do, N. H. Tran, C. Hong, C. A. Kamhoua, K. A. Kwiat, E. Blasch, S. Ren, N. Pissinou, and S. S. Iyengar, "Game theory for cyber security and privacy," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–37, May 2017.
21. H. von Stackelberg, *Market Structure and Equilibrium*. Springer, 2011, transl. from German.
22. Y. A. Korilis, A. A. Lazar, and A. Orda, "Achieving network optima using Stackelberg routing strategies," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 161–173, Feb. 1997.
23. T. Roughgarden, "Stackelberg scheduling strategies," *SIAM Journal on Computing*, vol. 33, no. 2, pp. 332–350, Jan. 2004.
24. M. Bloem, T. Alpcan, and T. Başar, "A Stackelberg game for power control and channel allocation in cognitive radio networks," in *2nd International Conference on Performance Evaluation Methodologies and Tools*, 2007, pp. 1–9.
25. J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport," in *7th International Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 125–132.
26. J. Tsai, S. Rathi, C. Kiekintveld, F. Ordóñez, and M. Tambe, "IRIS - A tool for strategic security allocation in transportation networks," in *8th International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 37–44.
27. G. G. Brown, W. M. Carlyle, J. Salmerón, and K. Wood, "Analyzing the vulnerability of critical infrastructure to attack and planning defenses," in *Emerging Theory, Methods, and Applications*, J. C. Smith, Ed. INFORMS, Sep. 2005, pp. 102–123.
28. G. G. Brown, M. Carlyle, J. Salmerón, and K. Wood, "Defending critical infrastructure," *Interfaces*, vol. 36, no. 6, pp. 530–544, Dec. 2006.
29. D. Grimsman, J. P. Hespanha, and J. R. Marden, "Stackelberg equilibria for two-player network routing games on parallel networks," in *2020 American Control Conference*, 2020, to be published.
30. G. W. Brown, "Iterative solution of games by fictitious play," in *Activity Analysis of Production and Allocation*, T. C. Koopmans, Ed. John Wiley & Sons, 1951, pp. 374–376.
31. J. Robinson, "An iterative method of solving a game," *The Annals of Mathematics*, vol. 54, no. 2, pp. 296–301, Sep. 1951.
32. G. W. Brown and J. von Neumann, "Solutions of games by differential equations," in *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker, Eds. Princeton University Press, 1952, vol. I, ch. 6, pp. 73–80.
33. J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, Jul. 1965.

34. M. Brückner and T. Scheffer, "Stackelberg games for adversarial prediction problems," in *17th ACM International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 547–555.
35. J. Marecki, G. Tesauro, and R. Segal, "Playing repeated Stackelberg games with unknown opponents," in *11th International Conference on Autonomous Agents and Multiagent Systems*, vol. 2, 2012, pp. 821–828.
36. A. Blum, N. Haghtalab, and A. D. Procaccia, "Learning optimal commitment to overcome insecurity," in *Neural Information Processing Systems 2014*, 2014, pp. 1826–1834.
37. G. Yang, R. Poovendran, and J. P. Hespanha, "Adaptive learning in two-player Stackelberg games with continuous action sets," in *58th IEEE Conference on Decision and Control*, 2019, pp. 6905–6911.
38. M. S. Kang and V. D. Gligor, "Routing bottlenecks in the Internet: Causes, exploits, and countermeasures," in *2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 321–333.
39. J. Postel, "User Datagram Protocol," Internet Engineering Task Force, Internet Standard, 1980.
40. T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno modification to TCP's fast recovery algorithm," Internet Engineering Task Force, Internet Standard, 2012.
41. G. Leitmann, "On generalized Stackelberg strategies," *Journal of Optimization Theory and Applications*, vol. 26, no. 4, pp. 637–643, Dec. 1978.
42. M. Breton, A. Alj, and A. Haurie, "Sequential Stackelberg equilibria in two-person games," *Journal of Optimization Theory and Applications*, vol. 59, no. 1, pp. 71–97, Oct. 1988.
43. B. von Stengel and S. Zamir, "Leadership with commitment to mixed strategies," Centre for Discrete and Applicable Mathematics, London School of Economics, Tech. Rep., 2004.
44. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe, "Computing optimal randomized resource allocations for massive security games," in *8th International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 689–696.
45. H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.
46. R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*. Springer, 1998.
47. J.-P. Aubin, *Viability Theory*. Birkhäuser, 1991.