Towards a Bayesian Network Game Framework for Evaluating DDoS Attacks and Defense

Guanhua Yan [*]
Information Sciences Group
Los Alamos National Laboratory
Los Alamos, NM 87545
ghyan@lanl.gov

Ritchie Lee
Carnegie Mellon University
Silicon Valley
NASA Ames Research Park
Moffett Field, CA 94035
ritchie.lee@nasa.gov

Alex Kent
Advanced Computing Solutions Program Office
Los Alamos National Laboratory
Los Alamos, NM 87545
alex@lanl.gov

David Wolpert [†]
Information Sciences Group
Los Alamos National Laboratory
Los Alamos, NM 87545
dwolpert@lanl.gov

ABSTRACT

With a long history of compromising Internet security, Distributed Denial-of-Service (DDoS) attacks have been intensively investigated and numerous countermeasures have been proposed to defend against them. In this work, we propose a non-standard game-theoretic framework that facilitates evaluation of DDoS attacks and defense. Our framework can be used to study diverse DDoS attack scenarios where multiple layers of protection are deployed and a number of uncertain factors affect the decision making of the players, and it also allows us to model different sophistication levels of reasoning by both the attacker and the defender. We conduct a variety of experiments to evaluate DDoS attack and defense scenarios where one or more layers of defense mechanisms are deployed, and demonstrate that our framework sheds light on the interplay between decision makings of both the attacker and the defender, as well as how they affect the outcomes of DDoS attack and defense games.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Security and protection; K.6.5 [**Management of Computing and Information Systems**]: Security and protection

[*]Los Alamos National Laboratory Publication No. LA-UR 12-20831

[†]David Wolpert is also affiliated with Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501.

CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA.
ACM 978-1-4503-1651-4/12/10.

Keywords

DDoS attacks and defense, game theory, Bayesian networks

1. INTRODUCTION

Distributed Denial-of-Service (DDoS) attacks have been plaguing the cyber space for a long time [29, 5, 26]. According to a VeriSign survey [25], almost two thirds of enterprise companies were victims of DDoS attacks, and 11 percent of them were attacked more than six times; also, there is a high price tag to defend against DDoS attacks as the average cost was estimated at as much as 2.5 million dollars. To fight back against DDoS attacks, researchers have proposed a number of countermeasures, many of which have been covered in the comprehensive surveys by Mirkovic and Reiher [14] and Peng *et al.* [17]. Performance evaluation of these methods, however, has been mostly conducted in a static environment (*e.g.*, simulation [9, 11] or a controlled real-world testbed [19, 15]) where behaviors of the attacker and the defender are predefined in an experiment. Although conclusions drawn from these studies shed light on the strengths and/or weaknesses of proposed defense techniques, they typically do not take into account strategic thinking of either the attacker or the defender, and thus fail to characterize accurately the dynamic interactions between the decisions made by the attacker and the defender in a real-world DDoS attack and defense incident.

In a few previous studies, evaluation of DDoS attacks and defense has been performed in a game-theoretic context [22, 27, 31, 28, 20]. These studies typically formulated DDoS attacks and defense as a static game (or a one-shot game) for a specific DDoS defense mechanism such as pushback [12], and studied the behaviors of the attacker and the defender in an equilibrium state (e.g., Nash equilibrium). Although these studies offer insights into how the attacker and the defender behave in a strategic manner in a DDoS attack and defense game, they suffer from the following three major shortcomings. First, these previous efforts mainly focused on the equilibrium states, particularly Nash equilibrium, in

the strategy spaces of the attacker and the defender. In practice, however, both the attacker and the defender have only bounded rationality due to limited information or resources, which makes them difficult, if not impossible, to always take the best-response actions. This is particularly true for DDoS attacks and defense because they usually take place within a short period of time. Second, most previous studies targeted one specific defense mechanism by the defender, in which the defender's decisions lie in how she sets the parameters in the method, and did not consider existence of multiple layers of protection against DDoS attacks commonly deployed in modern enterprise networks. Traditionally, game theorists use the mixed strategy approach to integrate multiple possible actions by a player with a probability distribution, but this method is inappropriate for modeling defense in depth against DDoS attacks, as it fails to model correlation among changes to the system state by different defense mechanisms. Third, in a real-life DDoS attack and defense incident, the strategic thinking of both the attacker and the defender is affected by many uncertain factors, such as how many legitimate users are using the service, how much traffic is generated from each user, and random packet dropping due to congestion at routers. Standard game theoretic methods such as extensive form games commonly used in the previous studies often ignore or simplify possible distributions behind these random events, and thus do not provide a seamless and coherent way of quantifying the effects of these uncertain factors.

Due to these constraints, game theory has not yet been widely applied in practice to evaluate DDoS attacks and defense. Against this backdrop, in this work we develop a game-theoretic evaluation framework, which is able to model different sophistication levels of strategic thinking by the attacker and the defender, offers great freedom in choosing distributions characterizing legitimate traffic, and provides a seamless method for reasoning among uncertain factors in DDoS attacks and defense. Based on a semi network-form game theoretical model proposed recently by Lee and Wolpert [10], our evaluation framework leverages a Bayesian graphical model for system state inference, where a set of random variables are used to characterize system states in DDoS attacks and defense and a directed acyclic graph to model their conditional dependencies. For uncertain factors such as the number of legitimate users which are decided by the nature, the modeler can use distributions inferred from historical data of the network. Dependencies among random variables are modeled based on the underlying mechanisms that govern the changes of system states. For instance, we can model different packet scheduling mechanisms at routers and different blocking methods by firewalls in this evaluation framework. Due to the flexibility of modeling dependencies among random variables, the framework also allows us to study multiple layers of protection against DDoS attacks. Also, the level-K reasoning used in the framework allows us to model players of different sophistication levels. A naïve defender, for instance, may think that DDoS attacks would never take place, and a more sophisticated defender, however, will take into consideration how the attacker responds if she does this and how she should further respond accordingly. Level-K thinking naturally reflects such type of reasoning in practice by assuming that a player's best response at a certain level depends on her observations from the previous one.

In this work, we use the proposed framework to evaluate DDoS attacks and defense in a typical enterprise network that deploys three layers of defense against DDoS attacks, including adding extra bandwidth to the external link, blocking suspicious traffic, and limiting the traffic rate from each source. Assuming that the attacker uses a botnet to launch DDoS attacks, he has the freedom to choose the number of bots in the attack and the sending rate per bot. We also model normal traffic from legitimate users, and take into consideration factors such as costs in adding more bandwidth to the external link, capacity of the server, and the detection rate and false alarm rate of the intrusion detection system. Using a variety of experiments with diverse parameter settings, we demonstrate how different sophistication levels of strategic thinking lead to different outcomes of a DDoS attack and defense game, how different system parameters (e.g., average number of legitimate users, bandwidth price, server capacity, performance of intrusion detection systems) affect the decisions made by the attacker and the defender, respectively, and how multiple layers of defense against DDoS attacks complement each other when they are put into effect simultaneously.

The remainder of the paper is organized as follows. In Section 2, we briefly introduce the related work. Section 3 provides the motivation behind using the semi network-form game to analyze DDoS attacks and defense. We introduce system variables used in this framework in Section 4, how to model conditional dependencies among them in Section 5, and level-K reasoning in Section 6. We show how to use this framework to analyze scenarios with a single layer of defense in Section 7 and those with multiple layers of defense in Section 8. Section 9 draws concluding remarks of this work.

2. RELATED WORK

The long history of DDoS attacks has inspired numerous defense techniques. Mirkovic and Reiher proposed a taxonomy of DDoS attack and defense mechanisms [14], and Peng *et al.* presented a survey of network-based DoS attacks and defense techniques [17]. Game theory has been previously applied to gain insights into cyber security issues. Roy *et al.* surveyed game-theoretic solutions to network security applications, largely along the line of the types of games used (i.e., static and dynamic games) and whether the information available to the players is perfect or imperfect, and complete or incomplete [18]. Manshaei *et al.* surveyed previous works on applying game-theoretic techniques to address security and privacy problems [13].

Previously, there were a few efforts on conducting game-theoretic analysis of DDoS attacks and defense. Zang *et al.* applied a Bayesian game model to analyze the defense against DDoS attack traffic with unclear signatures [31]. In their model, the defender is uncertain about the type of the traffic origin, which can be either a legitimate user or an attacker, and thus infers it using Bayesian rules. In [22], Snyder *et al.* introduced a DDoS traffic injection game, which is a two-person zero-sum game with imperfect knowledge. In the model developed by Wu *et al.* [27], the attacker attempts to optimize the attack effect by choosing the most effective attack traffic sending rate or number of zombie machines to send out attack traffic, while the defender optimizes the effectiveness of filtering attack traffic at the firewall. The entire game works in a continuous setting and the Nash equilibrium strategy can be computed analytically. In [28], Xu

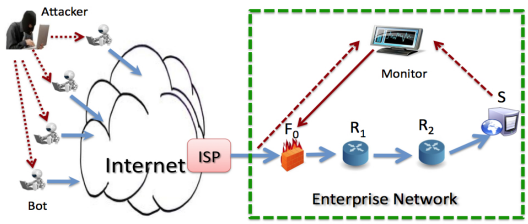


Figure 1: DDoS attack and defense scenario for an enterprise network

and Lee proposed a defense system against DDoS attacks and analyzed its performance in a game-theoretic context. Khirwadkar *et al.* developed a repeated game model based on the fictitious play process for pushback-based DDoS defense [7]. The key idea of their approach is that each player estimates the mixed strategy of the opponent's actions based on her previous observations, and then plays the pure strategy that is the best response accordingly. In [30], Yan and Eidenbenz proposed a truthful mechanism to provide economic incentives to ISPs to defend against DDoS attacks in a noncooperative environment.

Besides pushback and firewall filtering, client puzzle is another DoS/DDoS defense mechanism, in which a client has to solve a computational puzzle before the server commits resources to deal with his or her request. Game theory has also been applied to study this type of protection against DDoS attacks [2, 6]. In this work, we do not consider defense based on client puzzles, although they in principle can be incorporated into our evaluation framework.

3. MOTIVATION

In this work, we consider DDoS attacks that target a high-value server (e.g., a DNS server or an HTTP server) in an enterprise network. Figure 1 illustrates an DDoS attack and defense scenario for an enterprise network with a critical webserver, S. In this example, the attacker acquires a number of bots to send DDoS attack traffic to server S. In the enterprise network, the defender (i.e., network administrator) monitors the utilization level of the link from ISP to firewall F_0, and also the utilization level of the webserver. Once the defender infers that an attack is taking place, she takes actions to counter against it, including adding extra bandwidth to external link $ISP \rightarrow F_0$, blocking suspicious traffic, and limiting the maximum traffic rate from each source.

To analyze the strategies of both the attacker and the defender, we build an evaluation framework based on *semi network-form games*, which were previously proposed to model pilots' behaviors during mid-air encounters [10]. The semi network-form game model uses a Bayesian network for probabilistic inference of uncertain system states. Nodes in the Bayesian network are classified into two categories: *chance nodes*, which represent states of non-human components, and *decision nodes*, each of which describes how a player makes a decision based on her observed system states. Conditional probability distributions of chance nodes are predefined according to the physical law or mechanism that governs dependencies among system states. In a player's thinking process, she evaluates possible outcomes of the game (i.e., system states) due to a certain decision she could have made, based on a predefined *utility function*. Due to difficulty of obtaining a closed-form formula of the posterior probability distribution from the Bayesian network given some observed information, a player is assumed to sample it in order to approximate system states. To speed up the sampling process, the semi network-form game takes advantage of the d-separation property of a Bayesian network and likelihood weighting [8]. The tricky part in sampling is, however, how to deal with decision nodes, as no predefined conditional probability distributions are associated with them. The semi network form game applies the following methods. First, when a player samples from her own decision node, she use a predefined *satisficing function* that reflects her preferences in her decision making. For instance, when a person sees a car driving towards her, she would prefer to shun towards the left with probability 30% (choice 1) and the right with probability 70% (choice 2). Hence, when she samples from her decision node, she would have 30% of samples with choice 1 and 70% with choice 2, but which choice she eventually takes depends on her evaluation of the outcome based on her utility function. Second, when player A samples from player B's decision node, she uses the classical level-K strategy [3] for reasoning. With this strategy, player A's decision at a certain level k depends on how player B reasons at level $k-1$. The rigorous definition of a semi network-form game model is provided in Appendix A.

Using the semi network-form game model to evaluate DDoS attacks and defense has a few advantages. First, it provides a seamless way of integrating multiple defense mechanisms. The Bayesian network in it naturally models dependencies among these components. For example, adding extra bandwidth to the external link allows more traffic to arrive at the firewall, which thus has to deal with more packets when blocking suspicious traffic. Such dependencies are difficult to capture by traditional game theoretic methods such as the mixed strategy approach. Second, uncertain factors such as how much legitimate traffic is destined to the server and intensity of background traffic can be conveniently modeled as chance nodes in the framework. Third, the level-K reasoning strategy offers a natural way for modeling the sophistication levels in strategic reasoning by both the attacker and the defender. A naïve attacker, for example, would think there is no defense at level 0, but a more sophisticated attacker would take the defender's potential defense into consideration with higher k in reasoning.

In the following sections, we shall describe our framework which applies the semi network-form game model to evaluate DDoS attacks and defense. We will first introduce a number of random variables to describe system states in Section 4, and in the ensuing section, we will explain how to use a Bayesian network to model dependencies among these system variables. In Section 6, we will discuss how each player uses this Bayesian network to reason about the adversary's move and accordingly derive her best response action.

4. SYSTEM VARIABLES

In this section, we present the random variables that are used to model the system state. Table 1 summarizes all the variables introduced to describe the system state. These variables fall into three groups. The first group describes the network state, such as how much traffic arrives at the external link and how much traffic is served by the server. Since the network state changes after the defender takes mitigation actions, different variables are defined to reflect this change. The second and third groups of variables describe possible

Table 1: Summary of system parameters

Var.	Meaning
$\boldsymbol{A}$	Attacker's decision
$\boldsymbol{J}$	State of attacker's traffic arriving at external link
$\boldsymbol{L}$	State of external legitimate users' traffic
$\boldsymbol{X_h^{bf}}$	Output of foreground traffic from h before D's move
$\boldsymbol{X_h^{af}}$	Output of foreground traffic from h after D's move
$\boldsymbol{G_h}$	Rate of background traffic traversing resource h
$\boldsymbol{O_d^{bf}}$	Defender's observation before D's move
$\boldsymbol{D}$	Defender's mitigation parameters
$\boldsymbol{Y_{sb}}$	Output of foreground traffic after static blocking
$\boldsymbol{Y_{rl}}$	Output of foreground traffic after rate limiting

actions of the attacker and the defender, respectively. In the following, we shall explain these system variables in detail.

Network state. We first define parameters to characterize the system state in an enterprise network. Let $\mathcal{C}$ denote the set of resources vulnerable to DDoS attacks on the path of inbound attack traffic to the target server in the enterprise network. We call set $\mathcal{C}$ *critical resource set* and a resource in $\mathcal{C}$ a *critical resource*. A non-critical resource is assumed to have sufficient capacity so that no traffic is dropped due to it. In the example shown in Figure 1, we simply assume that $\mathcal{C}$ contains only the link $ISP \to F_0$ and the server.

There are partial ordering relationships among some critical resources. In the previous example, an attack packet must traverse firewall F_0 before arriving at the server. In this work, we assume that there is no conflict due to such ordering relationships. That is to say, if an attack packet has to use resource h_1 before resource h_2, no other attack packet should use resource h_2 before resource h_1. This is a reasonable assumption because in an enterprise network setting, there is usually a unique path from the perimeter firewall to the server. Such partial ordering constraint prevents cycles in the Bayesian graphical model described later.

There are two types of traffic that traverses a critical resource: *foreground* traffic refers to those packets that are destined to the server, and *background* traffic means those that do not go to the server. For critical resource h, we also introduce two vector variables: $\boldsymbol{X_h^{bf}}$ and $\boldsymbol{X_h^{af}}$ to describe the *output* state of foreground traffic from the resource *before* and *after* the defender's mitigation action, respectively. Each of these variables is a vector with the following representation:

$$\boldsymbol{X} = \langle \boldsymbol{X}[1], \boldsymbol{X}[2], \boldsymbol{X}[3] \rangle, \tag{1}$$

where $\boldsymbol{X}[1]$ denotes the utilization level of the resource, $\boldsymbol{X}[2]$ and $\boldsymbol{X}[3]$ denote the states of attack traffic and legitimate users' traffic destined to the server, respectively.

In this work, we focus on application-level DDoS attacks. Hence, we measure the *traffic rate* in terms of the number of *transactions* per second. To simplify our analysis later, we assume that a transaction has a constant number of packets and contains a constant number of bytes. Another important issue is how to represent the state of traffic traversing a resource. One approach is representing the traffic state as the aggregate number of transactions per second. This method, however, suffers when modeling the effect of the defender's mitigation. For example, the rate limiting scheme is usually performed per individual source IP, and aggregating the transaction rates over all sources (either malicious or benign) thus loses the information at the granularity relevant to rate limiting. Hence, we need to have full knowledge on the rate of each flow. We thus let $\boldsymbol{X}[2]$ and $\boldsymbol{X}[3]$ be a *rate vector* in which each element represents the rate of a flow. For clarity, we also use $size(\boldsymbol{X}[i])$, where $i = 2$ or 3, to denote the number of flows in $\boldsymbol{X}[i]$. To model legitimate traffic, we use rate vector $\boldsymbol{L}$ to characterize transactions generated from legitimate users. Hence, the number of source IP addresses in legitimate transactions is $size(\boldsymbol{L})$ and the total legitimate users' traffic rate is $\sum_{k=1}^{size(\boldsymbol{L})} \boldsymbol{L}[k]$.

A critical resource such as the link $ISP \to F_0$ is usually used not just for the traffic destined to the server. Hence, $\boldsymbol{G_h}$ is used to denote the rate of background traffic that traverses h. This rate is represented at the level that is relevant to resource h. In the example, for the link from $ISP \to F_0$, the background traffic is represented in terms of bytes per second, rather than transactions per second.

Attacker's move space. The goal of the attacker is to deplete the resources at the target. As now DDoS attacks are often launched with automatic software such as Stacheldraht [23], which requests bots to send attack traffic with the same rate, we assume that the attacker picks the same rate for each bot. Hence, the action space of the attacker can be characterized with a tuple:

$$\boldsymbol{A} = \langle \boldsymbol{A}[1], \boldsymbol{A}[2] \rangle, \tag{2}$$

where $\boldsymbol{A}[1]$ denotes the number of bots used in the attack, and $\boldsymbol{A}[2]$ the transaction rate from each bot. When the attack traffic from all bots reach the external link, its traffic is denoted as a rate vector $\boldsymbol{J}$.

Defender's move space. The defender monitors the network state to detect ongoing DDoS attacks. The defender's observed network state obviously depends on the monitoring scheme deployed. We consider the following monitoring scheme: first, the defender monitors the utilization level of every resource in the critical resource set $\mathcal{S}$; second, the defender monitors two statistics of the traffic destined to the server at the firewall, number of unique source IPs and average transaction rate from each source IP. We define the defender's observed network state before her mitigation action, $\boldsymbol{O_d^{bf}}$, where $\boldsymbol{O_d^{bf}}[i]$, for $i = 1, 2, 3, 4$, represents the utilization level of the external link, the utilization level of the server, the number of source IPs destined to the server, and the average transaction rate per flow, respectively.

Once the defender infers that a DDoS attack is going on based on the network state observed, her potential moves include requesting the upstream ISP to increase the downlink bandwidth (*bandwidth inflation*), blocking traffic from suspicious source IP addresses (*static blocking*), and limiting the maximum transaction rate allowed per source IP address (*rate limiting*). As these actions can be taken simultaneously, we use $\boldsymbol{D}$ to denote the defender's mitigation parameters:

$$\boldsymbol{D} = \langle \boldsymbol{D}[1], \boldsymbol{D}[2], \boldsymbol{D}[3] \rangle, \tag{3}$$

where $\boldsymbol{D}[1]$ denotes the fraction of increased bandwidth from the upstream ISP, $\boldsymbol{D}[2]$ whether static blocking is enabled (1 if enabled and 0 if disabled), and $\boldsymbol{D}[3]$ the upper limit on the maximum transaction rate allowed per source IP.

For ease of presentation, we assume the defender performs static blocking before rate limiting, both taking place at the firewall. To characterize the state of foreground traffic after each of these two actions by the defender, we introduce two other variables $\boldsymbol{Y_{sb}}$ and $\boldsymbol{Y_{rl}}$, each of which contains two rate vectors, to denote the rates of foreground traffic after the

defender performs static blocking and rate limiting, respectively. For $\boldsymbol{Y_t}$, where $t \in \{sb, rl\}$, $\boldsymbol{Y_t}[1]$ and $\boldsymbol{Y_t}[2]$ are the rate vectors for the attack traffic and the legitimate traffic, respectively.

5. MODELING CONDITIONAL DEPENDENCIES WITH BAYESIAN NETWORK

In the previous section, we have introduced a number of random variables to describe the system state in a DDoS attack and defense scenario. The dependencies among these variables are governed by the underlying scheduling mechanism deployed by each critical resource and also the ordering relationships among critical resources imposed by network connectivity. We use $\vdash$ to describe the dependence relationship. That is to say, $\boldsymbol{Z_0} \vdash \boldsymbol{Z_1}, \boldsymbol{Z_2}$ means that variable $\boldsymbol{Z_0}$ conditionally depends on both variables $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$. In the following, we describe how to model the conditional dependencies with a Bayesian network. To explain the basic principles, we take our best-effort guesses about the models that characterize conditional dependencies among the system variables. In a practical setting, we can adjust these models accordingly based on realistic data.

Critical resource dependencies. As we only consider a single server, the critical resources in set $\mathcal{S}$ form a path $h_0 = ISP \rightarrow F_0, h_1, ..., h_m = S$ where output of foreground traffic from h_i (i.e., $\boldsymbol{X}^{\boldsymbol{bf}}_{\boldsymbol{h_i}}$) depends on that from the previous one (i.e., $\boldsymbol{X}^{\boldsymbol{bf}}_{\boldsymbol{h_{i-1}}}$) and its background traffic before the defender's mitigation action:

$$\boldsymbol{X}^{\boldsymbol{bf}}_{\boldsymbol{h_i}} \vdash \boldsymbol{X}^{\boldsymbol{bf}}_{\boldsymbol{h_{i-1}}}, \boldsymbol{G_{h_i}}, \quad i = 1, 2, ..., m. \tag{4}$$

The input traffic fed to the external link includes attack traffic, legitimate users' traffic destined to the server, and background traffic that traverses the external link. Hence, we have the following for the external link $ISP \rightarrow F_0$:

$$\boldsymbol{X}^{\boldsymbol{bf}}_{\boldsymbol{ISP \rightarrow F_0}} \vdash \boldsymbol{J}, \boldsymbol{L}, \boldsymbol{G_{ISP \rightarrow F_0}}. \tag{5}$$

After the defender's mitigation action, the path formed by the critical resources remains intact, except that the defender's actions of static blocking and rate limiting affect the traffic rate at the firewall. Also, the output rate of foreground traffic from the external link after the defender's mitigation action depends on $\boldsymbol{D}$, which includes how much bandwidth has been added to the external link. Hence we have:

$$\boldsymbol{X}^{\boldsymbol{af}}_{\boldsymbol{ISP \rightarrow F_0}} \vdash \boldsymbol{J}, \boldsymbol{L}, \boldsymbol{G_{ISP \rightarrow F_0}}, \boldsymbol{D}. \tag{6}$$

As the attacker's traffic seen at the external link depends on how the attacker generates his attack traffic, we have:

$$\boldsymbol{J} \vdash \boldsymbol{A}. \tag{7}$$

Moreover, the output rate of foreground traffic after static blocking depends on both the output rate of foreground traffic from the external link, and the defender's decision on whether static blocking is enabled:

$$\boldsymbol{Y_{sb}} \vdash \boldsymbol{X}^{\boldsymbol{af}}_{\boldsymbol{ISP \rightarrow F_0}}, \boldsymbol{D}. \tag{8}$$

Similarly, the output rate of foreground traffic after rate limiting depends on the output rate of foreground traffic from the static blocking component, and the defender's choice on the upper limit on the maximum transaction rate allowed per source IP:

$$\boldsymbol{Y_{rl}} \vdash \boldsymbol{Y_{sb}}, \boldsymbol{D}. \tag{9}$$

For critical resources after the firewall on the path, we have:

$$\begin{cases} \boldsymbol{X}^{\boldsymbol{af}}_{\boldsymbol{h_i}} \vdash \boldsymbol{Y_{rl}}, \boldsymbol{G_{h_i}} & \text{if } i = 1 \\ \boldsymbol{X}^{\boldsymbol{af}}_{\boldsymbol{h_i}} \vdash \boldsymbol{X}^{\boldsymbol{af}}_{\boldsymbol{h_{i-1}}}, \boldsymbol{G_{h_i}} & \text{if } i > 1 \end{cases} \tag{10}$$

In the following, we shall elaborate on the conditional dependencies in Equations (4)-(10).

Attack traffic. Although the attacker specifies the number of bots used in the attack and the rate of transactions sent from each bot, the traffic coming out of each bot's physical machine, however, is subject to random distortion, and rates of attack traffic may be changed due to congestion before packets arrive at the target enterprise network. Let the true transaction rate from *any bot* arriving at the target enterprise network be $\boldsymbol{K}q$, where $0 \leq \boldsymbol{K} \leq n-1$ and q is the *traffic resolution*. Note that in reality, traffic rates are in the continuous space. To model them in our framework, we discretize them and let the largest possible rate for each flow be $(n-1)q$. We further assume that given $\boldsymbol{A} = \langle M, R \rangle$, we sample $\boldsymbol{K}$ using the following modified Poisson distribution:

$$\mathbb{P}\{\boldsymbol{K} = k\} = g(k) = \begin{cases} \frac{R^k e^{-k}}{Ck!} & \text{if } 0 \leq k \leq n-1 \\ 0 & \text{if } k \geq n \end{cases} \tag{11}$$

where $C = \sum_{k=0}^{n-1} \frac{R^k e^{-k}}{k!}$ is the normalizer. We can further easily obtain the attacker's traffic $\boldsymbol{J}$ from Equation (11). Note that $size(\boldsymbol{J})$ is equal to M.

FCFS scheduling. It is observed that Equations (4), (5), and (10) all take one of the following forms:

$$\begin{cases} \langle \boldsymbol{U^l_h}, \boldsymbol{U^a_h}, \boldsymbol{U^u_h} \rangle \vdash \langle \boldsymbol{V^a_h}, \boldsymbol{V^u_h} \rangle, \boldsymbol{G_h} \\ \langle \boldsymbol{U^l_h}, \boldsymbol{U^a_h}, \boldsymbol{U^u_h} \rangle \vdash \langle \boldsymbol{V^l_h}, \boldsymbol{V^a_h}, \boldsymbol{V^u_h} \rangle, \boldsymbol{G_h} \end{cases} \tag{12}$$

where $\boldsymbol{U^a_h}$ and $\boldsymbol{U^u_h}$ are the output rate vectors of attack and legitimate foreground traffic from resource h, respectively, $\boldsymbol{V^a_h}$ and $\boldsymbol{V^u_h}$ are the input rate vectors of attack and legitimate foreground traffic fed to resource h, respectively, $\boldsymbol{U^l_h}$ is the utilization of resource h, $\boldsymbol{V^l_h}$ is the utilization level of the previous critical resource, and $\boldsymbol{G_h}$ is the background traffic rate. As $\boldsymbol{V^l_h}$ is not used for computation, we can do the computation in the same manner for both cases. The dependency shown in Equation (12) is dictated by the scheduling mechanism deployed by the resource. Here we consider only the widely deployed FCFS (First Come First Serve) packet scheduling scheme, but the proposed framework can be easily extended to other scheduling schemes such as WFQ (Weighted Fair Queueing) [24].

Let $\Phi(h)$ denote the capacity of resource h. If resource h is a link, then $\Phi(h)$ gives the bandwidth of that link, and $\boldsymbol{G_h}$ gives the background traffic rate in terms of bytes per second. Otherwise, if resource h is a server, then $\Phi(h)$ provides the maximum number of transactions the server can handle per second, and both $\Phi(h)$ and $\boldsymbol{G_h}$ are measured in the number of transactions per second. Let constant b be the number of bytes in a transaction. In order to deal with two different types of resources, we introduce notation $\tau(h)$:

$$\tau(h) = \begin{cases} b & h \text{ is a link} \\ 1 & h \text{ is a server} \end{cases} \tag{13}$$

We also define $\boldsymbol{W_h}$ as follows:

$$\boldsymbol{W_h} = \boldsymbol{G_h}/\tau(h) + \sum_{k=1}^{size(\boldsymbol{V^a_h})} \boldsymbol{V^a_h}[k] + \sum_{k=1}^{size(\boldsymbol{V^u_h})} \boldsymbol{V^u_h}[k], \tag{14}$$

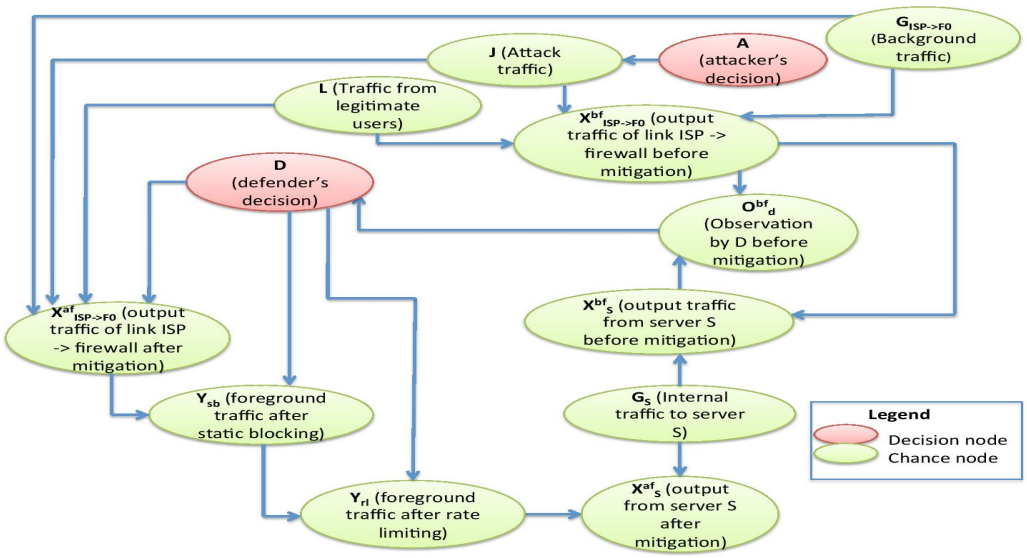


Figure 2: DDoS attack and defense modeled as a semi network-form game

which is to say, the handy variable $\boldsymbol{W_h}$ is actually the aggregate input transaction rate fed to resource h.

According to the FCFS scheduling mechanism, when there is no congestion (i.e., $\boldsymbol{W_h} \leq \Phi(h)/\tau(h)$), no transaction is dropped; otherwise, transactions are dropped with the same probability from each source IP. More specifically, if $\boldsymbol{W_h} \leq \Phi(h)/\tau(h)$, we have:

$$\begin{cases} \boldsymbol{U_h^l} & = & \frac{\tau(h)\boldsymbol{W_h}}{\Phi(h)} \\ \boldsymbol{U_h^a}[k] & = & \boldsymbol{V_h^a}[k] \quad \forall k : 1 \leq k \leq size(\boldsymbol{U_h^a}) \\ \boldsymbol{U_h^u}[k] & = & \boldsymbol{V_h^u}[k] \quad \forall k : 1 \leq k \leq size(\boldsymbol{U_h^u}) \end{cases} \tag{15}$$

and if $\boldsymbol{W_h} > \Phi(h)/\tau(h)$, we have:

$$\begin{cases} \boldsymbol{U_h^l} & = & 1 \\ \boldsymbol{U_h^a}[k] & = & \frac{\Phi(h)\boldsymbol{V_h^a}[i]}{\tau(h)\boldsymbol{W_h}} \quad \forall k : 1 \leq k \leq size(\boldsymbol{U_h^a}) \\ \boldsymbol{U_h^u}[k] & = & \frac{\Phi(h)\boldsymbol{V_h^u}[i]}{\tau(h)\boldsymbol{W_h}} \quad \forall k : 1 \leq k \leq size(\boldsymbol{U_h^u}) \end{cases} \tag{16}$$

It is noted that Equation (6) does not take the form shown in Equation (12). Instead, it takes the form $\langle \boldsymbol{U_h^a}, \boldsymbol{U_h^u} \rangle \vdash \langle \boldsymbol{V_h^a}, \boldsymbol{V_h^u} \rangle, \boldsymbol{G_h}, \boldsymbol{D}$. The computation is similar, though, except that $\Phi(h)$ in Equations (15) and (16) (including their corresponding conditions) have to be replaced with the new capacity, which is $(1 + \boldsymbol{D}[1]) \cdot \Phi(h)$.

Static blocking. Static blocking is performed on traffic destined to the server at each source IP level. We assume a simple Bernoulli process for both attack traffic and legitimate users' traffic: each source IP address in the attack traffic is classified as a bot with probability p_0 (true positive rate), and each source IP address in legitimate users' traffic is classified as a bot with probability p_1 (false positive rate).

If $\boldsymbol{D}[2] = 1$, which means that static blocking is enabled, then we have the following for $j = 1, 2$:

$$\boldsymbol{Y_{sb}}[j][k] \mid \boldsymbol{X^{af}_{ISP \to F_0}}[j+1][k], \boldsymbol{D} \sim \begin{cases} \boldsymbol{X^{af}_{ISP \to F_0}}[j+1][k] & \text{with probability } 1 - p_{j-1} \\ 0 & \text{with probability } p_{j-1} \end{cases}$$

Otherwise, if $\boldsymbol{D}[2] = 0$, which means that static blocking is disabled, then we always have

$$\begin{cases} \boldsymbol{Y_{sb}}[1][k] = \boldsymbol{X^{af}_{ISP \to F_0}}[2][k] \\ \boldsymbol{Y_{sb}}[2][k] = \boldsymbol{X^{af}_{ISP \to F_0}}[3][k] \end{cases} \tag{17}$$

Rate limiting. Recall $\boldsymbol{D}[3]$ is the maximum transaction rate allowed per source IP. For each source IP, if its rate is above the threshold $\boldsymbol{D}[3]$, its traffic is reshaped to make its rate become $\boldsymbol{D}[3]$, irrespective of whether it is a bot or not. Therefore, we have:

$$\boldsymbol{Y_{rl}}[j][k] = \begin{cases} \boldsymbol{Y_{sb}}[j][k] & \text{if } \boldsymbol{Y_{sb}}[j][k] \leq \boldsymbol{D}[3] \\ \boldsymbol{D}[3] & \text{if } \boldsymbol{Y_{sb}}[j][k] > \boldsymbol{D}[3] \end{cases}, \tag{18}$$

for both $j = 1$ and $j = 2$.

Observed state by the defender. Recall that the defender monitors the utilization level of each critical resource, and also the number of unique source IPs and the average transaction rate from each source IP in the traffic destined to the server. The statistics of traffic destined to the server can be obtained from $\boldsymbol{X^{bf}_{ISP \to F_0}}$. Hence, we have:

$$\boldsymbol{O_d^{bf}} \vdash \boldsymbol{X^{bf}_{ISP \to F_0}}, \boldsymbol{X_S^{bf}}. \tag{19}$$

More specifically, we have the following:

$$\begin{cases} \boldsymbol{O_d^{bf}}[1] & = & \boldsymbol{X^{bf}_{ISP \to F_0}}[1] \\ \boldsymbol{O_d^{bf}}[2] & = & \boldsymbol{X_S^{bf}}[1] \\ \boldsymbol{O_d^{bf}}[3] & = & \sum_{j=2}^{3} size(\boldsymbol{X^{bf}_{ISP \to F_0}}[j]) \\ \boldsymbol{O_d^{bf}}[4] & = & \frac{\sum_{j=2}^{3} \sum_{k=1}^{size(\boldsymbol{X^{bf}_{ISP \to F_0}}[j])} \boldsymbol{X^{bf}_{ISP \to F_0}}[j][k]}{\sum_{j=2}^{3} size(\boldsymbol{X^{bf}_{ISP \to F_0}}[j])} \end{cases} \tag{20}$$

As the defender's decision is made based on her observed network state before her mitigation action, the following holds:

$$\boldsymbol{D} \vdash \boldsymbol{O_d^{bf}}. \tag{21}$$

Construction of Bayesian Network. After all dependencies defined, we can construct a Bayesian network. Each state variable listed in Table 1 is a node in the Bayesian

network. When there is such a dependency relationship $\boldsymbol{Z_0} \vdash \boldsymbol{Z_1}, \boldsymbol{Z_2}, ..., \boldsymbol{Z_m}$, then we create a directed edge from each of $\boldsymbol{Z_i}$, where $1 \leq i \leq m$, to $\boldsymbol{Z_0}$. The conditional probability density functions are also defined accordingly based on how each state variable is calculated. Following the same example, its corresponding Bayesian network is illustrated in Figure 2.

6. LEVEL-K REASONING

In contrast to previous attempts of applying game theory to analyze DDoS attacks and defense, our evaluation framework considers bounded rationality [21] for both the attacker and the defender. Bounded rationality reflects many real-world situations in which people tend to satisfice rather than maximizing their preferences in decision making due to limited information or time available or the difficulty involved in sophisticated reasoning. To this end, we assume that both the attacker and the defender adopt the "level-K thinking" strategy [3], in which each player optimizes her best response at level k based on her observations from the previous level. Under the level-K reasoning model, we further define *satisficing distributions*, *utility functions*, and *level-0 distributions* for both the attacker and the defender.

Satisficing distributions λ. The satisficing distributions in the semi network-form game indicate a player's preferences in her decision making given an observed system state. They serve the purpose of sampling from a player's decision space in her reasoning. In our model, the attacker initiates the attack, and his decision does not depend on any other system variables. We also assume that the traffic rate from each bot is uniformly drawn from $\mathcal{K}$ by the attacker. On the other hand, let $\mathcal{B}$ denote the set of the number of bots that the attacker would like to choose in an attack. The number of bots used in the attack is uniformly drawn from $\mathcal{B}$.

The defender's satisficing distribution hinges on her observed network state $\boldsymbol{O_d^{bf}}$. The defender's observed network state helps her make decisions on **(1)** whether to perform a specific mitigation action (bandwidth inflation, static blocking, and rate limiting) and **(2)** how to set the parameters if a mitigation action is performed.

For each of these mitigation actions, we model the defender's satisficing function with a two-step process including whether to perform it and next how to set corresponding parameters. We use simple delta functions and logistic functions to model the probability with which the defender adopts a specific mitigation action based on her observed network state. Note that the delta function $\delta(x)$ returns 1 if x is true or 0 otherwise. The cumulative density function of the logistic distribution is $Q(x; \mu, s) = 1/(1 + e^{-(x-\mu)/s})$.

Let $\boldsymbol{D_{bi}}, \boldsymbol{D_{sb}}, \boldsymbol{D_{rl}}$ denote whether bandwidth inflation, static blocking, and rate limiting is adopted, respectively. We have:

$$\begin{cases} \lambda\{\boldsymbol{D_{bi}} = 1 \mid \boldsymbol{O_d^{bf}}\} &= \delta(\boldsymbol{O_d^{bf}}[1] \geq t_{bi}) \\ \lambda\{\boldsymbol{D_{sb}} = 1 \mid \boldsymbol{O_d^{bf}}\} &= \delta(\boldsymbol{O_d^{bf}}[2] \geq t_s) \cdot Q(\boldsymbol{O_d^{bf}}[3]; \mu_{sb}, s_{sb}) \\ \lambda\{\boldsymbol{D_{rl}} = 1 \mid \boldsymbol{O_d^{bf}}\} &= \delta(\boldsymbol{O_d^{bf}}[2] \geq t_s) \cdot Q(\boldsymbol{O_d^{bf}}[4]; \mu_{rl}, s_{rl}) \end{cases} \quad (22)$$

That is to say, when the utilization level of the external link exceeds threshold t_{bi}, bandwidth inflation is enabled; only when the server utilization level exceeds threshold t_s, will static blocking be enabled according to a logistic distribution function $Q(\boldsymbol{O_d^{bf}}[3]; \mu_{sb}, s_{sb})$. Similarly, when the server utilization level exceeds threshold t_s, rate limiting is enabled according to a logistic distribution function $Q(\boldsymbol{O_d^{bf}}[4]; \mu_{rl}, s_{rl})$.

In our model, the defender samples $\boldsymbol{D}[1]$, $\boldsymbol{D}[2]$, and $\boldsymbol{D}[3]$ in Equation (3) independently given $\boldsymbol{O_d^{bf}}$. If the defender decides to perform bandwidth inflation (i.e., $\boldsymbol{D_{bi}} = 1$), we assume that she samples the fraction of increased bandwidth uniformly from a set $\mathcal{F}$; otherwise, $\boldsymbol{D}[1]$ is always 0. In the case of static blocking, we note that $\boldsymbol{D}[2] = \boldsymbol{D_{sb}}$. Hence, Equation (22) gives us $\lambda\{\boldsymbol{D}[2] \mid \boldsymbol{O_d^{bf}}\}$. Similar to the case of bandwidth inflation, we assume that the defender samples an upper limit uniformly from set $\mathcal{H}$ when $\boldsymbol{D_{rl}} = 1$.

Utility functions. From the standpoint of the attacker, his goal is to maximize the fraction of dropped transactions from legitimate users' traffic. Meanwhile, the attacker also wants to minimize the number of bots used in the attack, as he needs to pay the botherder for each bot used [4]. Hence, a feasible utility function for the attacker is:

$$u_a = \frac{\sum_{k=1}^{size(\boldsymbol{L})} \boldsymbol{L}[k] - \sum_{k=1}^{size(\boldsymbol{X_S^{af}}[3])} \boldsymbol{X_S^{af}}[3][k]}{\sum_{k=1}^{size(\boldsymbol{L})} \boldsymbol{L}[k] \times \boldsymbol{A}[1]}, \quad (23)$$

which is the fraction of dropped legitimate users' transactions normalized by the number of bots used in the attack. The reason that we incorporate the number of bots into the attacker's utility function is because botnets are usually rented for malicious purposes such as DDoS attacks based on the number of bots used [4]. Here, alternative utility functions can also be applied, such as the difference between the attacker's gain (i.e., fraction of dropped legitimate traffic) and his cost on paying bots (after proper scaling). Evaluation of these different utility functions remains as our future work.

On the other hand, the defender values the fraction of served legitimate users' transactions. In the case of bandwidth inflation, the defender may need to pay the fee for increased bandwidth from the upstream ISP. Let the price per increased fraction be ρ. We define the the defender's utility function u_d as follows:

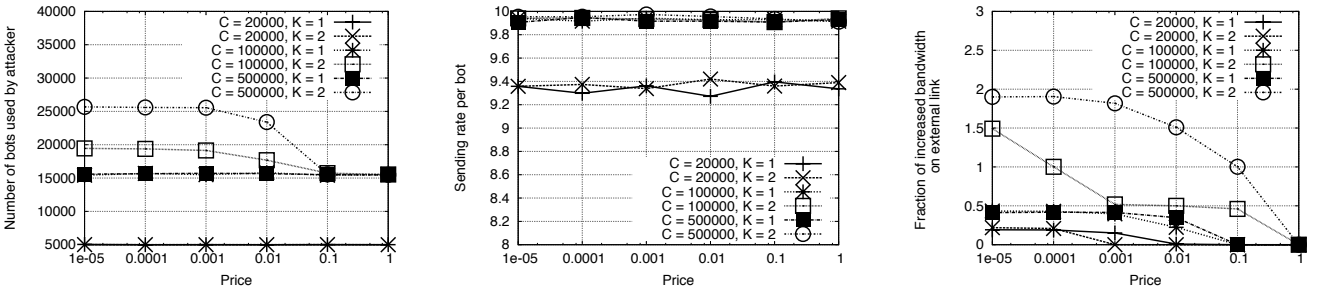
$$u_d = \frac{\sum_{k=1}^{size(\boldsymbol{X_S^{af}}[3])} \boldsymbol{X_S^{af}}[3][k]}{(1 + \rho \boldsymbol{D}[1]) \sum_{k=1}^{size(\boldsymbol{L}[k])} \boldsymbol{L}[k]}. \quad (24)$$

Note that in the Equation (24), the defender's utility is normalized by how much price he pays for the increased fraction of external link's bandwidth.

Level-0 distributions. To bootstrap the level-K thinking process, a player needs to make an assumption about how her opponent behaves nonstrategically, *i.e.*, at level 0. From the defender's viewpoint, we assume that *she thinks* that there is no attack from the attacker. Similarly, we assume that the attacker *thinks* that the defender does not deploy any defense in advance. Hence, in our model, parameter K indicates the sophistication level of the player: the higher K is, the more sophisticated a player's strategic thinking is.

7. ANALYSIS OF INDIVIDUAL DEFENSE

In this section, we use the evaluation framework, implemented with around 10,000 lines of C++ code, to analyze scenarios with a single mitigation scheme deployed. Understanding the intricacies in games played when a single defense scheme is put in place not only offers us insights into its strength and weakness against defending against



(1) number of bots vs. price (2) sending rate per bot vs. price (3) fraction of increased bandwidth vs. price

Figure 3: Decisions vs. different prices under the bandwidth inflation scheme (C: server capacity)

Table 2: Parameter settings in baseline case

Variable	Ref.	Setting
n	Eq. (11)	100
q	Eq. (11)	1.0 trans/sec
b	Eq. (13)	8000 bits
$\Phi(ISP \to F_0)$	Table 1	1Gbps
$\Phi(S)$	Table 1	20K, 100K, and 500K trans/sec
$G(ISP \to F_0)$	Table 1	$\mathcal{N}$(0.5Gbps, (0.2Gbps)^2)
$G(S)$	Table 1	$\mathcal{N}$(150 trans/sec, (50 trans/sec)^2)
L	Table 1	number of users $\sim$ Poisson(5,000)
		transactions/user $\sim$ Uniform(1,3)
$\mathcal{B}$		$\{5000i\}_{i=0,\ldots,10}$
$\mathcal{K}$		$\{kq\}_{k=1,\ldots,10}$
$\mathcal{F}$		$\{0.5i\}_{i=1,\ldots,10}$
$\mathcal{H}$		$\{i\}_{i=1,\ldots,10}$
t_{bi}	Eq. (22)	0.99
t_s	Eq. (22)	0.99
μ_{sb}	Eq. (22)	10000
s_{sb}	Eq. (22)	1000
μ_{rl}	Eq. (22)	5
s_{rl}	Eq. (22)	1

DDoS attacks, but also helps us understand later how the entire defense system works when multiple defense schemes are deployed simultaneously. To do so, we let $\boldsymbol{D_i}$ where $i \in \{bi, sb, rl\}$ always be 0 in Equation (22) if the corresponding defense scheme is not enabled. Table 2 summarizes the parameter settings in the experiments. Due to space limitation, we present only results for bandwidth inflation. For each scenario, we average results from 500 simulation runs. We consider only $K = 1, 2$ as when $K > 2$, it becomes too computationally prohibitive. Fortunately, empirical studies have suggested that people usually reason at only low levels. For instance, K was observed to be mostly 1 or 2 in [16], or almost always no greater than 3 in [1].

7.1 Decisions of attacker and defender

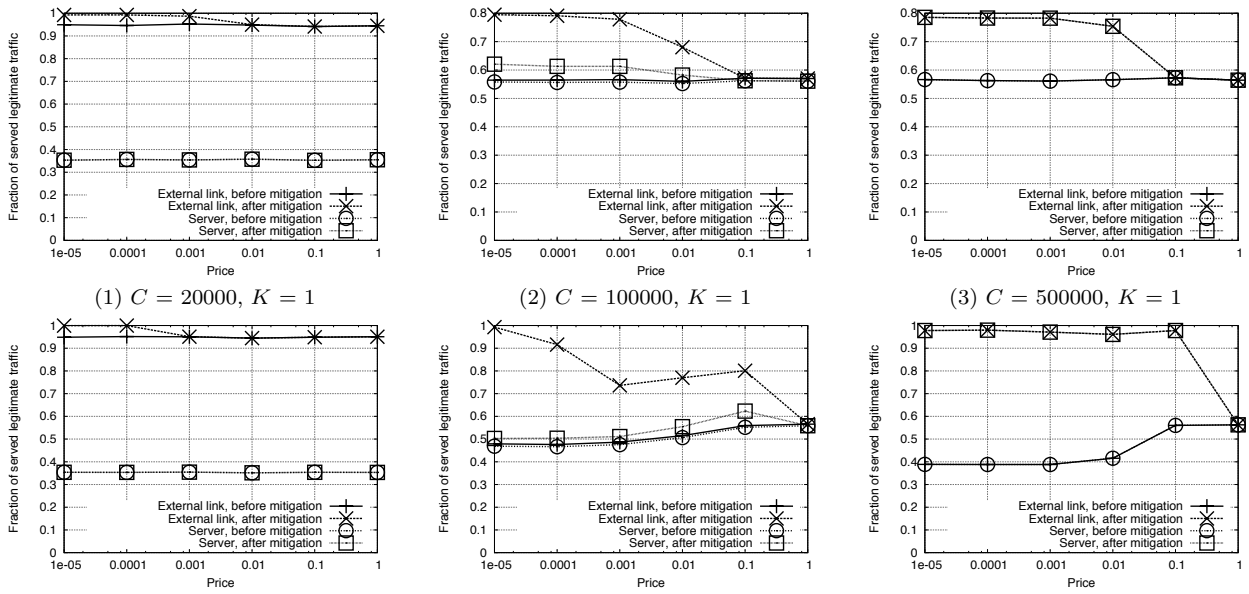
Decisions of the attacker and the defender under different bandwidth prices are shown in Figure 3, from which we make the following observations.

Naïve reasoning ($K = 1$): The number of bots and the sending rate per bot do not change with the bandwidth price ρ, because when $K = 1$, the attacker assumes that the defender does not deploy any mitigation scheme, thereby making the bandwidth price irrelevant. Moreover, when the server capacity is low, the number of bots used by the attacker is only 5000 (the lowest positive in set $\mathcal{B}$), suggesting that the small number of bots is sufficient for flooding the server. When the server capacity is high, the attacker needs more bots to degrade the performance of the server, even though the utility function of the attacker is inversely proportional to the number of bots used. On the other hand, the sending rate per bot chosen by the attacker is close to the maximum value he can use in set $\mathcal{K}$ (i.e., 10 transactions / sec). This is because the utility function of the attacker is not explicitly affected by the sending rate per bot. Note that when the server capacity is low, the sending rate per bot is slightly lower than those when the server capacity is higher. We conjecture that it is because the difference is negligible when the server is overloaded.

By contrast, the defender's decisions are affected by the bandwidth price. Albeit the defender assumes that there is no attack, increasing the bandwidth of the external link helps reduce the fraction of legitimate traffic dropped by the external link due to congestion. With a higher bandwidth price, the defender becomes more reluctant to increase the bandwidth, which agrees with our intuition. When the server capacity is lower, the defender has less incentive to increase the bandwidth because the traffic, even passing through the external link, would still be dropped by the overloaded server. Hence, we observe from Figures 3(3) that the higher the server capacity is, the more likely the defender will increase the bandwidth of the external link when the bandwidth price is not high.

Sophisticated reasoning ($K = 2$). When $K = 2$, the attacker uses only a small number of bots for attacks when the server capacity is low, regardless of the bandwidth price. *However, when the server capacity is high, the attacker tends to reduce the number of bots for attacks when the bandwidth price increases. This is interesting because the bandwidth price is only directly related to the defender's decision (from her utility function).* Actually, when the bandwidth price increases, the attacker figures that the defender is less likely to increase the bandwidth of the external link, and as a result, he tends to use a smaller number of bots for attacks. On the other hand, when the server capacity increases, the attacker figures that the defender tends to increase the bandwidth of the external link by a higher fraction if the bandwidth price is not high; hence, he has to use a larger number of bots to achieve desired attack effects. As in the cases when $K = 1$, the attacker uses a high sending rate per bot for attacks.

From the defender's side, with a higher bandwidth price, she has less incentive to increase the bandwidth of the external link. When the server capacity is high, not only does she have the same incentive to increase the bandwidth of the external link as she does when $K = 1$, but also she also knows that the attacker would use a higher number of bots for attacks and is thus more willing to increase the bandwidth



(1) $C = 20000$, $K = 1$ (2) $C = 100000$, $K = 1$ (3) $C = 500000$, $K = 1$

(4) $C = 20000$, $K = 2$ (5) $C = 100000$, $K = 2$ (6) $C = 500000$, $K = 2$

Figure 4: Fraction of legitimate traffic served by critical resources vs. different bandwidth prices under the bandwidth inflation scheme (Server capacity: C)

of the external link unless the bandwidth price is high. We thus observe from Figures 3(3) that the differences in the defender's decisions when $K = 2$ are prominent under different server capacities.

7.2 Outcomes of DDoS attack and defense games

The outcomes of the game under varying prices are illustrated in Figure 4. They show the fraction of legitimate traffic that has been served by the external link and the server before and after the mitigation scheme takes effect, respectively. We observe that when the server capacity is low (Figures 4(1,4)), although the majority of legitimate traffic (above 90%) passes through the external link, the server only serves about 35% of legitimate traffic, regardless of whether mitigation takes effect. Under a low bandwidth price, the defender decides to increase the bandwidth on the external link, resulting that almost all legitimate traffic passes through the external link (in both cases of $K = 1$ and $K = 2$).

Medium/high server capacity, $K = 1$: When the server capacity is 100,000, before mitigation takes effect, about 56% of legitimate traffic passes through the external link and is then served by the server, regardless of bandwidth price; after mitigation takes effect, the fraction of legitimate traffic passing through the external link increases, leading to more legitimate traffic served by the server. When the server capacity reaches 500,000, all traffic passing through the external link is served by the server. Also, both fractions of legitimate traffic passing through the external link and served by the server decrease with bandwidth price, as the defender becomes more reluctant to add more bandwidth to the external link with a higher bandwidth price.

Medium/high server capacity, $K = 2$: When the server capacity increases, both the fraction of legitimate traffic passing through the external link and then served by the server *before mitigation takes effect* increase as the bandwidth price increases, because the attacker reduces the number of bots used in the attack (see Figure 3(1)). From Figure 4(5), we note that when the server capacity is 100,000, as bandwidth price increases, the fraction of legitimate traffic passing through the external link first decreases, then increases, and finally decreases again. This interesting phenomenon reflects mixed effects from two factors. On one hand, as bandwidth price increases, the attacker tends to reduce the number of bots used in the attack (see Figure 3(1)). On the other hand, the defender also tends to reduce the fraction of bandwidth added to the external link as bandwidth price increases.

When the server capacity is 100,000, the fraction of legitimate traffic served by the server first increases, and then decreases as bandwidth price increases (Figure 4(5)). The shape of this curve differs from the one corresponding to that served by the external link, because the fraction of attack traffic arriving at the server decreases as bandwidth price increases. But when the server capacity is high (i.e., 500,000), all traffic passing through the external link can be served by the server, thus making the two curves identical in Figure 4(6).

Practical implications: Our analysis of individual defense reveals the following: **(1)** When the attacker and the defender reason at different sophistication levels, their decisions can change significantly, which further affect the outcome of the game. This implies that players' strategic thinking plays an important role in deciding the outcome of a DDoS attack and defense game. From the defender's perspective, if she knows she faces a sophisticated attacker, she knows that the attacker knows some defense would be deployed in advance and thus could use more bots. She needs to take that into consideration in the guessing game to find her best response. Our framework can be used in these situations to help predict actions of the adversary at different sophistication levels and accordingly derive the best strategies for the defender. **(2)** A player's decision can be affected by a parameter that is *not* directly related to her decision.

Table 3: Parameter settings in multi-layer defense experiments

Parameter	Values
Avg. #users	5000 (low), 10000 (medium), 15000 (high)
Server capacity	20000 (low), 100000 (medium), 500000 (high)
Bandwidth price	0.0001 (low), 0.01 (medium), 1 (high)
Detection rate	0.6 (low), 0.8 (medium), 1.0 (high)
False alarm rate	0 (low), 0.1 (medium), 0.2 (high)

When the bandwidth price changes, this affects not only behaviors of the defender whose utility is directly related, but also actions of the attacker, although indirectly. This suggests that game-theoretic evaluation of DDoS attacks and defense could lead to different results from traditional evaluation methods which usually ignore such indirect effects. The defender needs to consider such effects on both sides to find the best response. Our framework can be used in these situations to help the defender predict how changes of system parameters (e.g., bot price, bandwidth price, etc.) would affect the adversary's action and then find the best move to maximize her own utility.

8. ANALYSIS OF MULTI-LAYER DEFENSE

We now consider scenarios where the defender put multiple defense schemes in place. The parameters are set similarly as in Table 2, except those shown in Table 3. In total, we have 243 scenarios due to different combinations of parameter settings. For each scenario, we simulate 256 random samples, and the average decisions made by the attacker and the defender in these samples are shown in Figure 5. As we have a high-dimension input parameter space, we use different colors, shapes and sizes of markers to differentiate different inputs. As some irregular patterns in the figures suggest, the players' best strategies may not have linear relationships with the system parameters. In the following, we will interpret these plots and highlight some interesting observations.

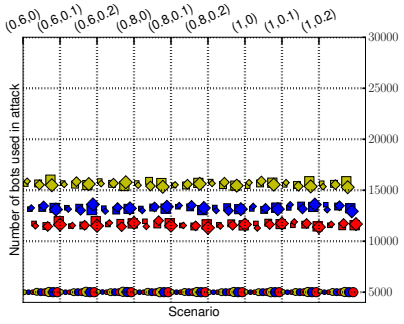
Attacker's decisions: For a naïve attacker (i.e., K is 1), the number of bots used depends on both the server capacity and the average number of users: with a low server capacity, the attacker chooses a low number of bots for attacks (around 5000); with a medium or high server capacity, the attacker increases the number of bots when the average number of users decreases. The naïve attacker also uses a high sending rate per bot (close to 10) in all scenarios.

For a sophisticated attacker (i.e., K is 2), when the server capacity is low, the attacker still uses a small number of bots (close to 5000). When the server capacity is high and the bandwidth price is not high, the attacker uses a high number of bots for the attack (around 25,000). When the server capacity is medium or high, the general trend is that a higher bandwidth price leads to a smaller number of bots used for the attack. Moreover, the detection rate and the false positive rate of the static blocking scheme have little effect on the number of the bots used by the attacker. Regarding the sending rate per bot, the attacker tends to use a high rate for most of the scenarios, but if the server capacity is low (*circles*) a lower sending rate per bot may be used. Interestingly, *the performance of the static blocking scheme affects the sending rate per bot by the attacker: a better performed static blocking scheme with a lower false alarm rate or a higher detection rate could make the attacker use a lower sending rate per bot.*

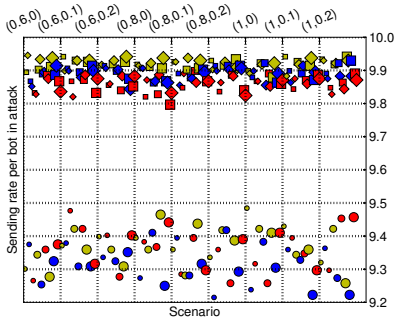
Defender's decisions: *Bandwidth inflation.* The fractions of increased bandwidth on the external link by the defender are shown in Figure 5(3) and 5(6). (1) When the bandwidth price is high (*large-sized markers*), the defender chooses not to increase bandwidth on the external link, regardless of other parameter settings. This is in concert with our intuition. (2) When the bandwidth price is medium (*medium-sized markers*) and the server capacity is medium or high (*blue or red*), if $K = 2$, the fraction of increased bandwidth on the external link does not change much with other parameters but if $K = 1$, it mostly increases with the average number of users (indicated by color). When the bandwidth price is medium and the server capacity is low (*red medium-sized circles*), if $K = 1$, the defender does not add more bandwidth to the external link, but if $K = 2$, her decision seems to be affected by the detection rate of the static blocking scheme: the higher detection rate, the lower fraction of increased bandwidth on the external link. This suggests that *the defender's three layers of defense mechanisms could complement each other: a better performed defense component allows the defender to rely less on others.* (3) When the bandwidth price is low (*small-sized markers*), the defender's decisions become more complicated. For instance, if K is 1, the defender adds a relatively high amount of extra bandwidth to the external link when false positive rate of the static blocking scheme is non-zero, the server capacity is low, and the average number of users is medium (*small-sized blue circles* in Figure 5 (3)), but if $K = 2$, the defender adds a relatively high amount of extra bandwidth when the server capacity is high (*diamonds*), regardless of the performance of the static blocking scheme.

Static blocking. The fraction of samples when the defender enables static blocking are shown in Figure 5(7) and 5(10). We first consider the scenarios when $K = 1$ (Figure 5(7). The defender tends not to enable static blocking when the server capacity is high (*diamonds*), or the server capacity is low but the average number of users is also low (*yellow circles*). She also tends to always enable static blocking when the server capacity is low and the average number of users is high (*red circles*). The defender enables static blocking with probability around 20% when the server capacity is medium (*boxes*). When the server capacity is low and the number of users is medium (*blue circles*), the defender's decision depends on the performance of the static blocking scheme. For instance, if the static blocking scheme does not produce false alarms or if the bandwidth price is high, the defender enables static blocking with probability at around 80%; otherwise, the probability of enabling static blocking is between 40% and 60%. *The results are surprising, because when $K = 1$, the defender does not assume existence of any attack and she thus does not have any incentive to enable static blocking.* Close examination reveals that when the server is overloaded, the defender will sample decisions with static blocking enabled, which would drop legitimate traffic due to false alarms. As the server is overloaded anyway, turning on static blocking produces exactly the same utility for the defender as if she does not.

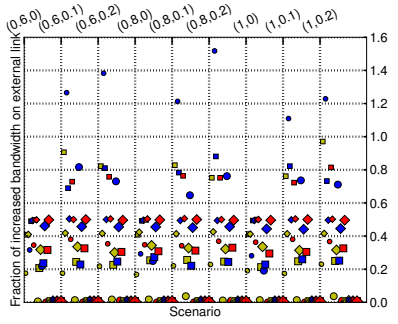
When $K = 2$ (Figure 5(10)), the defender disables static blocking when the server capacity is high (*diamonds*), and almost always enables static blocking when the server capacity is low and the average number of users is medium or high (*blue and red circles*). When the server capacity is medium (*boxes*), the defender enables static blocking in



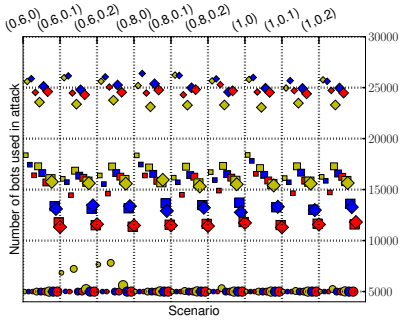
(1) bots, K = 1



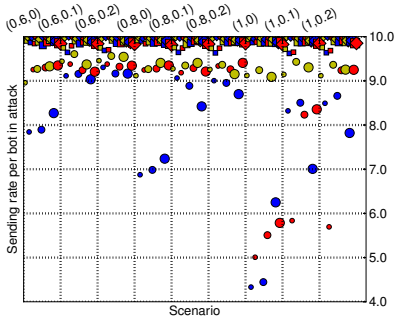
(2) rate per bot, K = 1



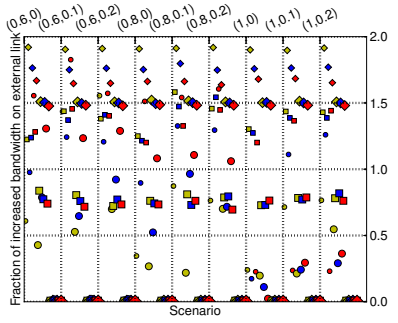
(3) bandwidth inflation, K = 1



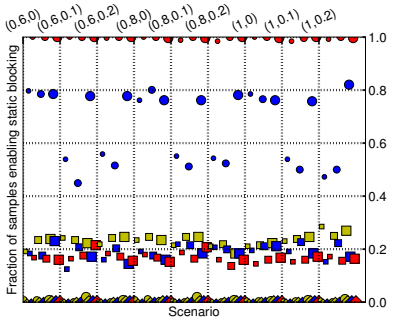
(4) bots, K = 2



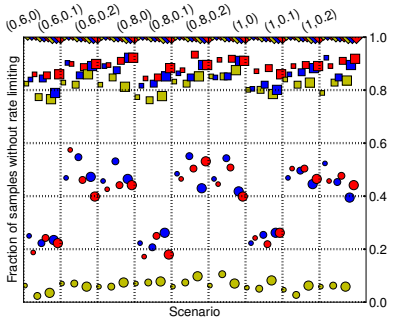
(5) rate per bot, K = 2



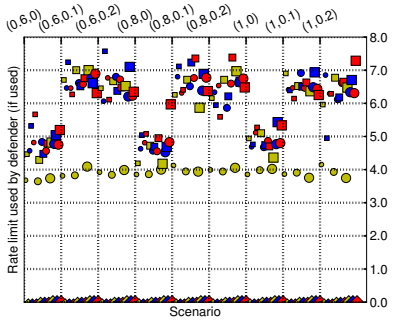
(6) bandwidth inflation, K = 2



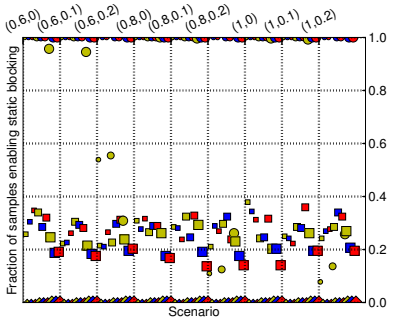
(7) static blocking, K = 1



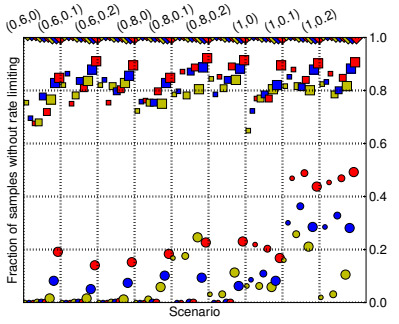
(8) no rate limit, K = 1



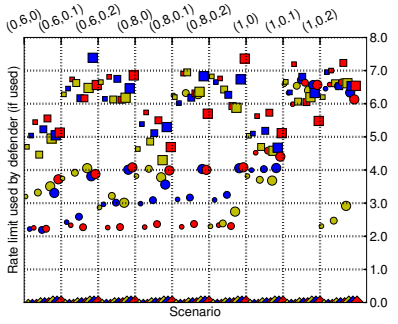
(9) rate limit, K = 1



(10) static blocking, K = 2



(11) no rate limit, K = 2



(12) rate limit, K = 2

Figure 5: Decisions made by the attacker and the defender when multiple layers of protection are deployed. Shape: Circle (capacity = 20000), Box (capacity = 100000), Diamond (capacity = 500000); Color: Yellow (Avg. #users = 5000), Blue (Avg. #users = 10000), Red (Avg. #users = 15000). The X-axis: (a, b) means the detection rate is a and the false positive rate is b. Size of each marker implies the corresponding bandwidth price. Best viewed in color.

about 23% of the samples, and the fraction slightly varies with the average number of users and the bandwidth price. When the server capacity is low and the average number of users is low (*yellow circles*), the defender's decision depends on the performance of the static blocking scheme as well as the bandwidth price; hence, it differs from what she would have done if $K = 1$.

Rate limiting. The decisions regarding rate limiting are shown in Figures 5(8-9) and 5(11-12). When K is 1, one interesting observation is that when the server capacity is low and the average number of users is medium or high (*red or blue circles*), *the defender's decision depends on the performance of the static blocking scheme*: if the static blocking scheme does not generate false positives, in around 21% of samples, the defender disables rate limiting; otherwise, the fraction of samples in which the defender does not enable rate limiting is about between 40% and 60%. In the remaining samples when the defender enables rate limiting, the rate limit is affected by the performance of the static blocking scheme. *One may wonder why the performance of the static blocking scheme affects the defender's decisions in rate limiting since when $K = 1$, the defender assumes no attacks.* This is actually because if the static blocking scheme is prone to false alarms, some legitimate traffic would be dropped by the static blocking scheme, and if the rate limiting component sees that the sending rate per source is below a certain threshold, it would not enable rate limiting. *Revelation of such correlation in changes of system states by different layers of protection is made possible due to reasoning based on Bayesian networks but is difficult under traditional game theoretic models.*

We next consider the results when $K = 2$. In this case, the defender's decisions regarding rate limit when the server capacity is high or medium are similar to those when $K = 1$ (*boxes and diamonds*). But when the server capacity is low (*circles*), we observe that if the detection rate of the static blocking scheme is not perfect, then the defender is more likely to enable rate limiting with a high probability and use a low rate limit (if the rate limiting scheme is enabled); otherwise if the detection rate of the static blocking scheme is 100%, the defender is more likely to disable rate limiting and in cases where it is not enabled, she tends to use a higher rate limit. *This further confirms our previous observation that the different layers of defense mechanisms can complement each other, and having a better performed component allows the defender to rely less on other components in the defense system.*

Practical implications: Our analysis of multi-layer defense reveals the following: **(1)** In some scenarios the deployment of multi-layer defense against DDoS attacks allows the defender to rely less on a specific defense mechanism if another is well performed, but the degree of reduction depends on various factors, such as the server capacity and intensity of legitimate users' traffic. When the defender needs to decide whether to invest on a new defense technology, say, a better intrusion detection system, she needs to understand relationships between it and other defense schemes. A number of scenarios have shown having a better-performed traffic blocking scheme does not necessarily mean it is useful, because other defense schemes can complement a poorly-performed one. Hence, our framework is instrumental in identifying those scenarios where investment on a new defense technology will indeed pay off. **(2)** When there are multiple layers of defense, players' decisions become complicated due to their nonlinear relationships with system parameters. Our framework can help the defender get better outcome or control defense mechanisms in these situations, due to its ability to infer the best action (e.g., optimal settings of defense parameters) given an observed network state. Moreover, our framework is capable of sorting through a number of uncertain factors (e.g., number of normal users, amount of DDoS traffic, and fraction of normal traffic falsely classified by the intrusion detection system) to find optimal strategies for the defender.

9. CONCLUSIONS

The goal of this work is to explore new methodology for evaluating DDoS attacks and defense. Although there were a few previous attempts in using traditional game theory tools to model DDoS attacks and defense, our contribution in this work is development of a game-theoretic framework that is able to model much more complicated scenarios in DDoS attacks and defense. This framework uses Bayesian networks for players to infer system states in a probabilistic manner. Level-K thinking is used to model different sophistication levels of the players. Observations from our experiments show that taking strategic thinking of players into account brings a new perspective to analysis of DDoS attacks and defense, and it is necessary to do so if we want to gain a realistic glimpse into behaviors of both players in DDoS attacks and defense.

In our future work, we plan to incorporate incomplete information into our evaluation framework by letting each player define a separate Bayesian network for her own reasoning, rather than having a single Bayesian graphical model as described in Sections 4 and 5. Such flexibility will allow each player to model more fine-grained knowledge about the adversary in her decision making process.

In this study we mainly focused on games in which the attacker and the defender reason at the same level K, but it would be interesting to see how players thinking at different sophistication levels affect the outcomes of the games. In another plan of our future work, we would like to extend the static game model used in our current framework to dynamic games where decisions of both the attacker and the defender change over time. Techniques such as reinforcement learning can be used to model how a player learns about the adversary's strategies dynamically. This will further improve the richness of our framework.

Appendix A: Semi Network-Form Game [10]

The formal definition of semi network-form game uses the following notations. Δ_Z denotes the probabilistic simplex over a space Z, and $\Delta_{Z|Y} = \times_{y \in Y} \Delta_Z$ is the space of all possible conditional distributions of $z \in Z$ conditioned on $y \in Y$. As in standard game theory, for any particular player i, $-i$ is used to denote the set of all players excluding player i.

A *semi network-form game* involving N players is a quintuple (G, X, u, R, π) that satisfies the following conditions:

1. G is a finite directed acyclic graph $\{V, E\}$, where V and E are the sets of nodes and edges in it, respectively. For any node $v \in V$, we let $pa(v)$ denote the set of parent nodes of v.

2. X is a Cartesian product of $|V|$ finite sets, each with at least two elements. X_v denotes the set corresponding to $v \in V$, and $X_{pa(v)}$ the Cartesian product of sets for all elements in $pa(v)$. Intuitively, X_v contains all possible states at node v and $X_{pa(v)}$ all possible states of v's parent nodes.

3. u is a function $X \to \mathbb{R}^N$. It also can be seen as a set of N utility functions $u_i : X \to \mathbb{R}$ for each player i.

4. R is a partition of V into $N+1$ subsets. Each of $R(i)$, where $1 \leq i \leq N$, contains a single element, which is a *decision* node, and all elements in $R(N+1)$ are *chance* nodes. $\mathsf{D} = \cup_{i=1}^{N} R(i)$ and $\mathsf{C} = R(N+1)$ represent the set of decision nodes and chance nodes, respectively.

5. π is a function from $v \in R(N+1) \to \Delta_{X_v|\times_{v' \in pa(v)} X_{v'}}$. That is to say, π assigns to every chance node $v \in R(N+1)$ a conditional probability distribution of v conditioned on the states of its parents.

In a semi network-form game, each player needs to infer the system state based on her observed information, and then takes the best-response action. A player performs statistical inference by treating G, augmented with conditional probability functions in π, in the semi network-form game as a Bayesian graphical model. Typically, it is difficult to derive a closed form of posterior probability distributions for a Bayesian network conditioned on some observed information, and hence sampling techniques such as forward sampling and importance sampling are widely used for approximating these distributions. In the context of semi network-form games, however, the challenge is how to sample from a decision node, as in the definition of a semi network-form game no conditional probability distributions are specified for these nodes. Note that π is only defined for chance nodes in the model.

To address this challenge, *level-K relaxed strategies* are proposed. The key idea is level-K thinking, in which a player's reasoning is recursively defined with level K. Define the level-K relaxed strategy of a decision node $v \in \mathsf{D}$, where $K \geq 1$, to be $\Lambda^{K-1}(X_{v'} \mid X_{pa(v')})$. We have the following definitions for a semi network-form game (G, X, u, R, π):

- $U = V \setminus \{v, pa(v)\}$,
- $P^K(X_v \mid X_{pa(v)}) = \pi(X_v \mid X_{pa(v)})$ if $v \in \mathsf{C}$,
- $P^K(X_v \mid X_{pa(v)}) = \Lambda^{K-1}(X_v \mid X_{pa(v)})$ if $v \in \mathsf{D}$,
- $P^K(X_Z) = \prod_{v'' \in Z} P^K(X_{v''} \mid X_{pa(v'')})$.

As level-K relaxed strategies are recursively defined, we specify a *level-0 distribution* $\Lambda^0(X_v \mid x_{pa(v)}) \in \Delta_{X_v|\times_{v' \in pa(v)} X_{v'}}$ for every decision node. With these notations, the level-K relaxed strategy for any decision node $v \in \mathsf{D}$ is defined as follows. First, for every decision node, a *satisficing function* distribution $\lambda(X_v \mid x_{pa(v)}) \in \Delta_{X_v|\times_{v' \in pa(v)} X_{v'}}$ is specified to indicate a player's preference in sampling from the space of her possible moves. Also specify two integers M and M' used in the sampling process. Suppose that decision node v corresponds to player i, where $1 \leq i \leq N$. Next, perform the following process independently for each $x_{pa(v)} \in X_{pa(v)}$:

1. Choose M independent and identically distributed (IID) samples of $\lambda(X_v \mid x_{pa(v)})$. After removing all duplicates, we obtain a set $\{x'_v(j) : j = 1, \ldots, m\}$ with m elements;

2. For each j where $j = 1, \ldots, m$, choose M' IID samples of the joint distribution:

$$P^K(X_V \mid x'_v(j), x_{pa(v)}) = \prod_{v' \in V} P^K(X_{v'} \mid X_{pa(v')}) \delta_{X_{pa(v)}, x_{pa(v)}} \delta_{X_v, x'_v(j)};$$

and let $\{x'_V(k; x'_v(j)) : k = 1, \ldots M'\}$ be the set of these samples. In the above equation, the delta function $\delta_{a,b}$ returns 1 if $a = b$ or 0 otherwise. Here, rejection sampling is applied: if in the sample X_V node v's state is not $x'_v(j)$ or its parents' states are not $x_{pa(v)}$, the sample is simply rejected.

3. For each j where $j = 1, \ldots, m$, we estimate player i's utility under her possible action $x'_v(j)$:

$$\hat{u}_i^K(x'_U(; x'_v(j)), x'_v(j), x_{pa(v)}) = \frac{1}{M'} \sum_{k=1}^{M'} u_i(x'_V(k, x'_v(j))). \quad (25)$$

4. Player i takes the best-response action $x'_v(j^*)$ where

$$j^* = \text{argmax}_j [\hat{u}_i^K(x'_U(; x'_v(j)), x'_v(j), x_{pa(v)})].$$

The rejection sampling applied by the level-K relaxed strategy can be extremely costly, as a large number of samples are rejected because they do not contain observed information (node v's parents' states) or the player's action sampled from her satisficing function. To further improve the performance of the level-K relaxed strategy, the following two optimization techniques can be adopted:

D-separation. Let X, Y, Z be three sets of nodes in a Bayesian network. X and Y are called *d-separated* given Z if there is no active trail between any node in X and any node in Y given Z [8]. Intuitively speaking, given Z we can guarantee the independence between X and Y, if X and Y are d-separated given Z. In the context of semi network-form games, we can find the set of nodes independent of a decision node v given the set of its parent nodes $pa(v)$. For these nodes, we can sample them independently first and then combine them with the samples for the other nodes as described in the level-K relaxed strategy.

Likelihood weighting. Rejection sampling is computationally expensive because it rejects samples if they are incompatible with the evidence. Likelihood weighting is a method that forces samples to take appropriate values at evidence nodes and then compensate it with a weight characterizing the likelihood that the evidence nodes take the forced values in the sample. In the semi network-form game, calculation of the utility function in Equation (25) is adjusted accordingly to take the weight of each sample into account. Also, we do not have to normalize the weights as unnormalized weights are sufficient for comparing utilities under different actions.

Acknowledgment

The authors acknowledge and appreciate the support provided for this work by the Los Alamos National Laboratory Directed Research and Development Program (LDRD, project number 20110093DR) and NASA Aviation Safety Program SSAT project.

10. REFERENCES

[1] A. Arad and A. Rubinstein. The 11-20 money request game: Evaluating the upper bound of k-level reasoning. Technical report, Tel Aviv University Working Paper, May 2010.

[2] B. Bencsath, I. Vajda, and L. Buttyan. A game based analysis of the client puzzle approach to defend against DoS attacks. In *Proceedings of the 2003 International Conference on Software, Telecommunications and Computer Networks*, pages 763–767, 2003.

[3] C. F. Camerer. *Behavioral game theory: experiments in strategic interaction*. Princeton University Press, 2003.

[4] N. Christin, S. Egelman, T. Vidas, and J. Grossklags. It's all about the Benjamins: An empirical study on incentivizing users to ignore security advice. In *Proceedings of IFCA Financial Cryptography'11*, pages 16–30, Saint Lucia, February 2011.

[5] `http://edition.cnn.com/2008/TECH/04/18/cnn.websites/`.

[6] M. Fallah. A puzzle-based defense strategy against flooding attacks using game theory. *IEEE Transactions on Dependable And Secure Computing*, 7:5–19, January 2010.

[7] T. Khirwadkar, K. C. Nguyen, D. M. Nicol, and T. Basar. Methodologies for evaluating game theoretic defense against DDoS attacks. In *Proceedings of the 2010 Winter Simulation Conference*, 2010.

[8] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[9] I. Kotenko and A. Ulanov. Simulation of internet DDoS attacks and defense. In *Proceedings of the 9th international conference on Information Security*, ISC'06, pages 327–342, 2006.

[10] R. Lee and D. Wolpert. Game theoretic modeling of pilot behavior during mid-air encounters. In *Decision Making with Imperfect Decision Makers*, pages 75–111. Springer, 2012.

[11] Z. Li, Y. Xiang, and D. He. Computational intelligence and security. chapter Simulation and Analysis of DDoS in Active Defense Environment, pages 878–886. Springer-Verlag, 2007.

[12] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, 32:62–73, July 2002.

[13] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J.-P. Hubaux. Game theory meets network security, 2010. Submitted to ACM Survey.

[14] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communications Review*, 34(2), April 2004.

[15] J. Mirkovic, S. Wei, A. Hussain, B. Wilson, R. Thomas, S. Schwab, S. Fahmy, R. Chertov, and P. Reiner. DDoS benchmarks and experimenter's workbench for the deter testbed. In *Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom'07)*, pages 1–7, May 2007.

[16] R. Nagel. Unraveling in guessing games: An experimental study. *American Economic Review*, 85(5):1313–26, December 1995.

[17] T. Peng, C. Leckie, and K. Ramamohanarao. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, 39, April 2007.

[18] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, pages 1–10, 2010.

[19] D. Schmidt, S. Suriadi, A. Tickle, A. Clark, G. Mohay, E. Ahmed, and J. Mackie. A distributed denial of service testbed. In Jacques Berleur, Magda Hercheui, and Lorenz Hilty, editors, *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*, volume 328 of *IFIP Advances in Information and Communication Technology*. Springer Boston, 2010.

[20] P. Shi and Y. Lian. Game-theoretical effectiveness evaluation of DDoS defense. In *Proceedings of the Seventh International Conference on Networking (ICN'08)*, pages 427 –433, April 2008.

[21] H. A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[22] M. E. Snyder, R. Sundaram, and M. Thakur. A game-theoretic framework for bandwidth attacks and statistical defenses. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks*, 2007.

[23] `http://www.sans.org/security-resources/malwarefaq/stacheldraht.php`.

[24] D. Stiliadis and A. Varma. Latency-rate servers: a general model for analysis of traffic scheduling algorithms. *IEEE/ACM Transactions on Networking*, 6(5):611 –624, October 1998.

[25] `http://www.internetnews.com/security/article.php/3933046/How+Much+Does+a+DDoS+Attack+Cost.htm`.

[26] `http://www.pcmag.com/article2/0,2817,2374063,00.asp`.

[27] Q. Wu, S. Shiva, S. Roy, C. Ellis, and V. Datla. On modeling and simulation of game theory-based defense mechanisms against DoS and DDoS attacks. In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10, pages 159:1–159:8. ACM, 2010.

[28] J. Xu and W. Lee. Sustaining availability of web services under distributed denial of service attacks. *IEEE Transactions on Computers*, 52(2):195 – 208, feb. 2003.

[29] `http://news.cnet.com/2100-1023-236621.html`.

[30] G. Yan and S. Eidenbenz. DDoS mitigation in non-cooperative environments. In *Proceedings of the 7th international IFIP-TC6 networking conference*, NETWORKING'08, Singapore, 2008.

[31] W. Zang, P. Liu, and M. Yu. How resilient is the Internet against DDoS attacks? – a game theoretic analysis of signature-based rate limiting. *International Journal of Intelligent Control and Systems*, 12(4):307–316, December 2007.