Can We Beat DDoS Attacks in Clouds?

Shui Yu, *Senior Member, IEEE*, Yonghong Tian, *Senior Member, IEEE*,
Song Guo, *Senior Member, IEEE*, and Dapeng Oliver Wu, *Fellow, IEEE*

Abstract—Cloud is becoming a dominant computing platform. Naturally, a question that arises is whether we can beat notorious DDoS attacks in a cloud environment. Researchers have demonstrated that the essential issue of DDoS attack and defense is resource competition between defenders and attackers. A cloud usually possesses profound resources and has full control and dynamic allocation capability of its resources. Therefore, cloud offers us the potential to overcome DDoS attacks. However, individual cloud hosted servers are still vulnerable to DDoS attacks if they still run in the traditional way. In this paper, we propose a dynamic resource allocation strategy to counter DDoS attacks against individual cloud customers. When a DDoS attack occurs, we employ the idle resources of the cloud to clone sufficient intrusion prevention servers for the victim in order to quickly filter out attack packets and guarantee the quality of the service for benign users simultaneously. We establish a mathematical model to approximate the needs of our resource investment based on queueing theory. Through careful system analysis and real-world data set experiments, we conclude that we can defeat DDoS attacks in a cloud environment.

Index Terms—Cloud computing, DDoS attacks, mitigation, system modelling, resource investment

1 Introduction

In this paper, we attempt to answer one important question: can we beat DDoS attacks in a cloud environment? The answer is positive. One essential issue of DDoS attack and defense is resource competition; if a defender has sufficient resources to counter a DDoS attack, then the attack will be unsuccessful, and vice versa. Unfortunately, the counterparts of clouds, e.g., the client server and the peer-to-peer computing platforms, do not have sufficient resources to beat DDoS attacks. However, a cloud infrastructure provider pools a large amount of resources and makes them easy access in order to handle a rapid increase in service demands [1]. Therefore, it is almost impossible for a DDoS attack to shut down a cloud. However, individual cloud customers (referred to as parties hosting their services in a cloud) cannot escape from DDoS attacks nowadays as they usually do not have the advantage. The good news is that it is highly likely for individual cloud customers to win the battle by taking advantage of the unique features of clouds. In this paper, we explore how to overcome DDoS attacks against individual cloud customers from the resource competition perspective.

Currently, cloud computing has become one of the fastest growing sectors in the IT industry all over the world. Cloud computing features a cost-efficient, ''pay-as-you-go'' business model and flexible architectures, such as SaaS, PaaS and SaaS. A cloud platform can dynamically clone virtual machines in a very quick fashion, e.g., duplicating a gigabyte level server within one minute [2]. Despite the promising business model and hype surrounding cloud computing, security is the major concern for businesses shifting their applications to clouds [3], [4].

Distributed Denial of Service (DDoS) is a major threat to Internet based killer applications for noncloud computing environments, such as independent news web sites, e-business and online games [5]. DDoS attacks are now carried out by botnets. Resent researches [6] have corrected a long held belief that hackers can easily compromise as many computers as they want. Due to the anti-virus and anti-malware effort and software, the number of active bots a botmaster can manipulate is constrained to the hundreds or few thousands level, even though the number of bot footprints may be much larger.

In the early work about DDoS defense, Yau *et al.* [7] treated DDoS attacks as a resource management problem. Recent researches [8], [9], [10] have further demonstrated that the essential issue of DDoS attack and defense is a competition for resources: the winner is the side who possesses more resources in the battle. Different from other computing platforms, a cloud environment usually has profound resources, full control, and dynamic allocation capability of resources. As a result, it is not possible to deny the service of a cloud with the scale of current botnets.

However, an individual cloud customer does not have this advantage of surviving a brute force DDoS attack. Cloud service providers (CPS) usually offer cloud customers two resource provisioning plans: short-term on-demand and long-term reservation. Major cloud providers, such as Amazon EC2 and GoGrid, provide both plans to their customers [11]. If a customer chooses the first plan, then she will be charged based on what she uses. This resource business model is vulnerable to an Economic Denial of Sustainability (EDoS) attack [12], [13]. Moreover, this kind of attack also disturbs the service of clouds who

- *S. Yu is with the School of IT, Deakin University, Victoria, 3125, Australia. E-mail: syu@deakin.edu.au.*
- *Y. Tian is with the School of EECS, Peking University, Beijing, China. E-mail: yhtian@pku.edu.cn.*
- *S. Guo is with the School of CSE, The University of Aizu, Aizuwakamatsu, Japan. E-mail: sguo@u-aizu.ac.jp.*
- *D.O. Wu is with the Department of ECE, University of Florida, FL, USA. E-mail: wu@ece.ufl.edu.*

Manuscript received 8 Apr. 2013; revised 28 June 2013; accepted 7 July 2013. Date of publication 23 July 2013; date of current version 13 Aug. 2014.
Recommended for acceptance by M. Guo.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2013.181

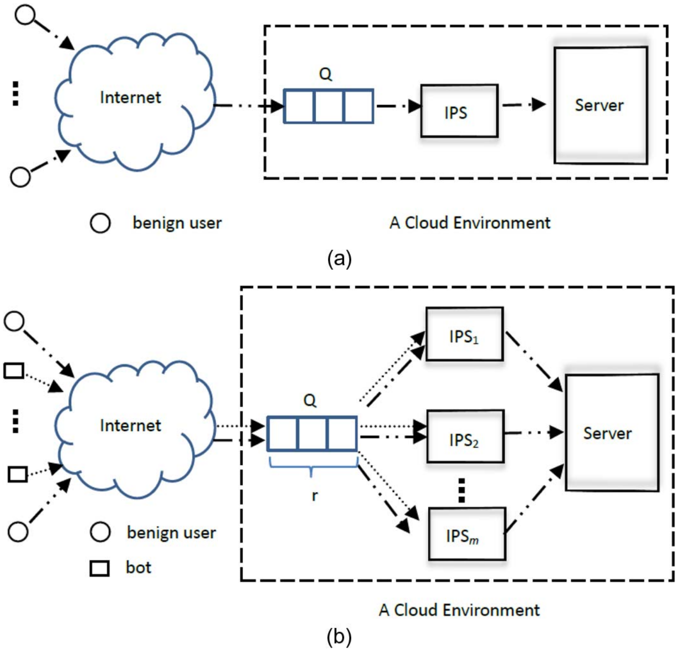


Fig. 1. (a) Cloud hosted server in a nonattack scenario. (b) Cloud hosted server under DDoS attack with the mitigation strategy in place.

allocate resources based on spot instance [14], [15]. On the other hand, if a cloud customer takes the reservation plan, she usually makes the source reservation for the maximum usage of her business. In other words, the reserved resource for her application is limited. As a result, a threat of DDoS attack remains.

As a new business model and computing platform, cloud related research has attracted a lot of attention. There has been plenty of research done on cloud, such as economical modelling [16] and resource optimization [17]. However, research on DDoS attack and defense in a cloud environment is still at an early stage. The available cloud security covers various aspects, such as attack mitigation strategies against DDoS attacks [18] or EDoS attacks [13] in a cloud environment, DDoS defense as a cloud service [19], and security architecture against DDoS attacks in cloud computing [20].

In this paper, we propose a practical dynamic resource allocation mechanism to confront DDoS attacks that target individual cloud customers. In general, there is one or several access points between a cloud data center and the Internet. Similar to firewalls, we place our Intrusion Prevention System (IPS) at these locations to monitor incoming packets. When a cloud hosted server is under a DDoS attack, the proposed mechanism will automatically and dynamically allocate extra resources from the available cloud resource pool, and new virtual machines will be cloned based on the image file of the original IPS using the existing clone technology [21], [22]. All IPSs will work together to filter attack packets out, and guarantee the quality of service (QoS) for benign users at the same time. When the volume of DDoS attack packets decreases, our mitigation system will automatically reduce the number of its IPSs, and release the extra resources back to the available cloud resource pool.

As aforementioned, the essential issue to defeat a DDoS attack is to allocate sufficient resources to mitigate attacks no mater how efficient our detection and filtering algorithms are. In order to estimate our resource demands and QoS for benign users in a DDoS battle, we employ queueing theory to undertake performance evaluation due to its extensive deployment in could performance analysis, such as in [23], [24], [25].

It should be noted that our goal for this paper is to explore the possibility of defeating DDoS attacks in a cloud environment from a technical and resource competition point of view. We therefore do not involve specific DDoS detection methods, and do not involve too many business issues which may be caused by our mitigation proposal. With the proposed system in place, we believe most DDoS attacks can be defeated, if not all attacks. This will make cloud customers more confident in shifting their businesses to cloud platforms.

The contributions of this paper are summarized as follows:

- We point out that DDoS attacks do threaten individual cloud customers. However, by taking advantage of the cloud platform, we can overcome DDoS attacks, which is difficult to achieve for noncloud platforms. To the best of our knowledge, this paper is an early feasible work on defeating DDoS attacks in a cloud environment.
- We propose a dynamic resource allocation mechanism to automatically coordinate the available resources of a cloud to mitigate DDoS attacks on individual cloud customers. The proposed method benefits from the dynamic resource allocation feature of cloud platforms, and is easy to implement.
- We establish a queueing theory based model to estimate the resource allocation against various attack strengths. Real-world data set based analysis and experiments help us to conclude that it is possible to defeat DDoS attacks in a cloud environment with affordable costs.

The remainder of this paper is organized as follows. The mitigation mechanism is discussed in Section 2. We present system modelling and analysis of the proposed method in Section 3, and design an algorithm for the mitigation mechanism in Section 4. Performance evaluations are conducted in Section 5, and we present further discussion in Section 6. Finally, we summarize this paper and discuss future work in Section 7. The related work can be found from the online supplementary file of this paper which is available in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.181.

2 DDoS Attack Mitigation in Clouds

In this section, we propose a mechanism to dynamically allocate extra resources to an individual cloud hosted server when it is under DDoS attack.

First of all, we examine the features of a cloud hosted virtual server in a nonattack scenario. As shown in Fig. 1a, similar to an independent Internet based service, a cloud hosted service includes a server, an intrusion prevention system (IPS in the diagram), and a buffer for incoming packets (queue Q in the diagram). The IPS is used to protect the specific server of the hosted service. All packets of benign users go through the queue, pass the IPS and are served by the server. In general, the number of benign users is stable, and we suppose the virtual IPS and virtual server have been allocated sufficient resources, and therefore the quality of service (QoS) is satisfactory to users.

When a DDoS attack occurs against the hosted virtual server, a large number of attack packets are generated by botnets, and pumped to queue Q. In order to identify these attack packets and guarantee the QoS of benign users, we have to invest more resources to clone multiple IPSs to carry out the task. We propose to clone multiple parallel IPSs to achieve the goal as shown in Fig. 1b.

The number of IPSs we need to achieve our goal depends on the volume of the attack packets. As discussed previously, the attack capability of a botnet is usually limited, and the required amount of resources to beat the attack is usually not very large. In general, it is reasonable to expect a cloud can manage its reserved or idle resources to meet demand.

3 System Modelling and Analysis

In this section, we first discuss how to model the system in general, and then establish an executable mathematical model to approximate the resource demands on various attack strengths using queueing theory for the proposed mitigation method. Following this model, we will also provide a thorough analysis of the system.

3.1 System Modelling in General

In general, we treat our studied system as a black box, and observe its input and output with respect of time t. We denote the input as $a(t)$, the output as $b(t)$, and the system function of the black box as $h(t)$. We then have a relationship among these three functions as follows.

$$b(t) = a(t) * h(t), \tag{1}$$

where $*$ is the convolution operation.

In order to obtain solutions for the output, and for most of the cases, we map $a(t)$ and $h(t)$ into another domain using different transform techniques, such as Laplace-transform, Z-transform, and so on. We use the Laplace transform in this paper. The Laplace transform of $a(t)$ is defined as follows

$$A(s) \triangleq \int a(t)e^{-st}dt. \tag{2}$$

Similarly, we can obtain $H(s)$ from $h(t)$. Let $B(s)$ be the Laplace transform of $b(t)$, and we obtain $B(s)$ through the following equation

$$B(s) = A(s) \cdot H(s). \tag{3}$$

Once $B(s)$ is in place, we can calculate $b(t)$ using the inverse Laplace transform,

$$b(t) = \frac{1}{2\pi i}\int B(s)e^{st}ds. \tag{4}$$

In our case, $a(t)$ represents the arrival distribution, $h(t)$ is the system service distribution. In the queueing theory, our studied system can be modeled as $G/G/m$, namely, general arrival distribution and general service rate distribution. However, for this general model, the analysis will be very complex, and we may not have computationally attractable methods to calculate the numerical results of these models [26]. For example, we cannot obtain $A(s)$, $H(s)$ from $a(t)$, $h(t)$ most of the time, and we cannot obtain $b(t)$ even if $B(s)$ is in place sometimes. As a result, researchers have to approximate the complex $G/G/m$ model to solvable models in order to proceed with analysis and prediction. To date, only the $M/M/m$ model (exponential arrival rate and service rate) can offer a closed form result as these distributions possess wonderful properties, such as additive and memoryless [27]. We will also follow this mainstream method for our analysis on the proposed mitigation strategy.

3.2 Approximation of the Proposed System

As widely applied in cloud performance analysis [23], [24], [25], we make a few reasonable assumptions and approximations in order to make our modelling, analysis and the following experiments feasible and practical. There are:

- Whether or not there is a DDoS attack, we suppose the number of benign users is stable, and we suppose the cloud is big enough and has sufficient reserved or idle resources to overcome a DDoS attack on a cloud customer.
- We suppose the arrival rate to the system follows the Poisson distribution when a DDoS attack is ongoing. We know the arrivals of a server in a nonattack case obey the Poisson distribution. When a DDoS attack is ongoing, there are many more packets to the system, and a general conclusion from queueing theory is that a large number of arrival rate can be approximated as a Poisson distribution [27]. Therefore, we use the Poisson distribution as the arrival distribution for both attack and nonattack cases in this paper.
- We suppose the service rate of each individual IPS follows an exponential distribution, which is common in queueing analysis.

In order to measure the performance of the system, we use *average time in system* of packets as a metric of QoS in this paper. We denote T_n (n stands for normal) as the acceptable average time in system for packets of benign users in nonattack cases. In general, T_n is a constant. In attack cases, the average time in system varies because the number of attack packets changes. Therefore, we denote it as $T_a(t)$ for a given time point t (a stands for attack).

We note that T_n and $T_a(t)$ in this paper do not include the time spent in the normal service of the server because this time is the same for both attack and nonattack cases. In other words, the system we study here only includes queue Q and the original IPS or multiple IPSs.

In order to guarantee the QoS of benign users in attack cases, we need to dynamically allocate resources into the battle, and make sure $T_a(t) \leq T_n$ for any time point t.

We use a function $R(\cdot)$ to represent the resource investment. Let variable x be the expected system performance, such as average time in system of requests. Obviously,

$R(\cdot)$ depends on x and time point t. We therefore denote it as $R(x, t)$. We also simplify it as $R(x)$ or $R(t)$ if it is clear in the context.

As shown in Fig. 1b, we model our mitigation system as an M/M/m queue, namely, one incoming queue with an infinite buffer size, the arrivals following the Poisson distribution, and $m (m \geq 2)$ multiple servers each with an exponential service rate.

With the system model in hand, we can transform our mitigation problem into an optimization problem: minimizing the resource investment $R(t)$ while guaranteeing the QoS for benign users in attack cases. We formulate the problem as follows

$$\begin{aligned} &mini.R(t) \\ &\quad \text{s.t.} \\ &\qquad T_a(t) \leq T_n. \end{aligned} \tag{5}$$

3.3 Resource Investment Analysis

In order to decide on the investment for a expected quality of service, we have to define an executable investment function $R(x)$ with respect to a system performance expectation x. Variable x could be a vector to represent specific requirements of different resources, such as $x = \langle CPU, memory, IO, bandwidth \rangle$.

For feasibility reasons, we define $R(x)$ as a linear and nondecreasing function. Let x, y be two different system performance expectations. Then we have the following properties of this investment function

$$\begin{cases} R(x) & = 0, \qquad\qquad\quad x = 0 \qquad\quad (a) \\ R(x) & \leq R(y), \qquad\qquad 0 \leq x \leq y \quad (b) \\ R(ax + by) & = aR(x) + bR(y), \quad a, b \in \mathbb{R} \qquad (c). \end{cases} \tag{6}$$

In practice, the current CSPs, such as Amazon EC2, offer resources in terms of instance. An instance includes a fixed amount of various resources, e.g. memory and IO. In other words, an instance is the basic unit for resource allocation. In this case, equation (6) does reflect this practice very well.

3.4 System Analysis for Nonattack Cases

For a web based service in nonattack cases, it is generally accepted that the arrival rate of queue Q follows the Poisson distribution, whose probability density function is defined as

$$\mathbb{P}\{X = k\} = \frac{\lambda^k e^{-\lambda}}{k!}, \; k = 0, 1, \ldots. \tag{7}$$

Equation (7) describes the probability of k arrivals for a given time interval.

For nonattack cases as shown in Fig. 1a, the system can be naturally modeled as an $\mathrm{M/M/1/\infty}$ queue. We denote the packet arrival rate as λ, and the service rate of the IPS as μ.

People usually derive a parameter called *utility rate* or *busy rate* as the ratio of the arrival rate and the service rate. In this case, we denote it as

$$\rho_n = \frac{\lambda}{\mu}. \tag{8}$$

Usually, we need to make sure $\rho_n < 1$ in order to keep the system in a stable state.

Based on queueing theory [27], we know the probability of the system stays state π_k (namely, there are k packets in the system) is

$$\begin{cases} \pi_0 = 1 - \frac{\lambda}{\mu} \\ \pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0. \end{cases} \tag{9}$$

The probability density of the time in system is

$$\mathbb{P}(T = t) = (\mu - \lambda) e^{-(\mu - \lambda)t}, \tag{10}$$

for $t > 0$. The average time spent in the IPS system is

$$T_n = \frac{1}{\mu - \lambda} = \frac{1}{\left(\frac{1}{\rho_n} - 1\right)\lambda}. \tag{11}$$

Naturally, we assume T_n meets users' expectations of service. We will use T_n as a benchmark of QoS for benign users when the cloud hosted server is under a DDoS attack.

3.5 System Analysis for Attack Cases

In the case of a cloud customer being subjected to a DDoS attack as shown in Fig. 1b, and based on our proposal, the cloud will clone multiple IPSs to counter the attack in order to guarantee the QoS for benign users.

It is natural that we model the mitigation system using the M/M/m model: Poisson arrival rate and multiple (m) servers with an exponential service rate.

For the sake of neatness in the analysis, we make the following definition:

Attack strength is the total number of arrivals to a victim for a given time interval when a DDoS attack is ongoing.

From this definition, we know an attack strength includes both benign packets and attack packets. For the sake of simplicity, we represent an attack strength as $r (r \geq 1)$ (where r is a real number) times of the arrival rate of nonattack cases. As we denote the arrival rate of nonattack cases as λ, an attack strength is therefore denoted as $r\lambda$. The service rate for each IPS is still μ as it was in the nonattack case, and all IPSs share the workload. Once again, based on queueing theory [27], we have the following system service rate μ_k (k servers in service).

$$\mu_k = \min[\mathrm{k}\mu, \mathrm{m}\mu] = \begin{cases} \mathrm{k}\mu & k \leq m \\ \mathrm{m}\mu & m \leq k. \end{cases} \tag{12}$$

We obtain the $\pi_k (0 \leq k \leq \infty)$ (the probability of k packets in the system) as follows.

$$\pi_k = \begin{cases} \pi_0 \frac{(m\rho)^k}{k!} & k \leq m \\ \pi_0 \frac{\rho^k m^m}{m!} & m \leq k, \end{cases} \tag{13}$$

where ρ is the system busy rate, which is defined in a multiple homogeneous server case as

$$\rho = \frac{r\lambda}{m\mu}. \tag{14}$$

Similarly, we have to make sure $\rho < 1$ in order to keep the system in a stable state.

In equation (13), π_0 represents the probability of a state of the system that there are no packets in the queue, including the initial state of the system. π_0 is an important parameter in queueing analysis, and it is defined as follows in the M/M/m model.

$$\pi_0 = \left[1 + \sum_{k=1}^{m-1} \frac{(m\rho)^k}{k!} + \sum_{k=m}^{\infty} \frac{(m\rho)^k}{k!} \frac{1}{m^{k-m}}\right]^{-1}. \tag{15}$$

Opposite to state π_0, we have π_{m+}, which is the probability that a packet has to wait when it arrives in the system. π_{m+} is expressed as

$$\begin{aligned}\pi_{m+} &= \sum_{k=m+1}^{\infty} \pi_k \\ &= \pi_0 \frac{(m\rho)^m}{m!(1-\rho)} \qquad (16) \\ &= 1 - \sum_{k=0}^{m} \pi_0 \frac{(m\rho)^k}{k!}. \qquad (17)\end{aligned}$$

From a system viewpoint, we are interested in the average time spent in the system, $\overline{T_a(t)}$. In this paper, the number of servers, m, is also a factor on $\overline{T_a(t)}$. We therefore express it more explicitly as $\overline{T_a(t,m)}$, which is given as follows.

$$\begin{aligned}\overline{T_a(t,m)} &= \mathbb{E}[T_a(t,m)] \\ &= \frac{1}{r\lambda}\left(m\rho + \rho \frac{(m\rho)^m}{m!} \frac{\pi_0}{(1-\rho)^2}\right). \qquad (18)\end{aligned}$$

Combining equations (14) and (18), we have

$$\overline{T_a(t,m)} = \frac{1}{\mu} + \frac{1}{r\lambda} \frac{\left(\frac{r\lambda}{\mu}\right)^m}{m!} \frac{\pi_0}{\left(1 - \frac{r\lambda}{m\mu}\right)^2}. \tag{19}$$

As previously discussed, in order to guarantee the QoS for benign users during a DDoS attack, the condition of equation (5) has to be satisfied. Therefore,

$$\frac{1}{\mu} + \frac{1}{r\lambda} \frac{\left(\frac{r\lambda}{\mu}\right)^m}{m!} \frac{\pi_0}{\left(1 - \frac{r\lambda}{m\mu}\right)^2} \le \frac{1}{\mu - \lambda}. \tag{20}$$

For simplicity, let

$$f(r,m) = \frac{\lambda}{\mu} - (\mu - \lambda) \frac{(r\lambda)^{m-1}}{m!\mu^m} \frac{\pi_0}{\left(1 - \frac{r\lambda}{m\mu}\right)^2}, \tag{21}$$

where π_0 is determined by equation (15).

Combining (21) and (20), we have the constrain for the optimization as

$$f(r,m) \ge 0. \tag{22}$$

Moreover, we note that equation (22) is under the following constrains

$$\begin{cases} r\frac{\lambda}{\mu} < m & (a) \\ r > 1 & (b) \\ m = 2, 3, \ldots, & (c) \end{cases} \tag{23}$$

where condition (a) comes from equation (14).

If equation (22) does not hold, then it is time to invest more resources to clone one or more IPSs against the ongoing attack.

Usually, a cloud has sufficient idle or reserved resources, which can be used to counter brute force DDoS attacks. We denote the resource for one IPS as R_{IPS}, and the available reserved resources of a cloud as R_c. The maximum IPSs that we can use is then $\lfloor \frac{R_c}{R_{IPS}} \rfloor$. In a strict sense, on top of the constrains in (23), we have to have one more constrain as follows

$$m \le \left\lfloor \frac{R_c}{R_{IPS}} \right\rfloor + 1. \tag{24}$$

4 DDoS Mitigation Algorithm for a Cloud

In this section, we present the related algorithm for the proposed mitigation strategy.

4.1 DDoS Detection Methods

As aforementioned, DDoS defense in cloud essentially depends on resources no matter which defense methods we use. Therefore, in our mitigation algorithm, we do not involve specific detection methods, rather, we focus on the resource management aspect of detection. In the online supplementary file, we list a few DDoS detection methods that could be implemented in cloud for interested readers.

4.2 DDoS Mitigation Algorithm in Cloud

In the algorithm, we first observe the arrival patterns in nonattack cases for a protected server, and extract the parameters μ and λ. Moreover, we also identify the resources for the current IPS, R_{IPS}, and the available or idle resources R_c of the cloud.

When a DDoS attack is detected by the original IPS, we then clone one IPS based on the image of the original IPS, and calculate the average time in system for the current status. If $\overline{T_a(t,m)} > T_n$, then we clone one more IPS for the filtering task. As the battle continues, and we find $\overline{T_a(t,m)} < T_a(t, m-1)$, then it is time to reduce one IPS and release the resources back to the cloud available resource pool.

The details of the dynamic resource allocation algorithm against DDoS attacks on a cloud customer can be found from the online supplementary file of this paper.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed dynamic resource allocation method for DDoS mitigation in a cloud from various perspectives. We first study the performance for nonattack scenarios, then investigate the performance of the proposed mitigation method against an ongoing DDoS attack, and then estimate the cost for the proposed mitigation methods.

First of all, we summarize the key statistics of DDoS attacks in a global scenario from highly referred literature [6], [28], and present them in Table 1.

A cloud usually has profound resources. We use the Amazon EC2 as an example and show the related data in Table 2.

TABLE 1
Key Statistics of DDoS Attacks

Feature	Attack duration [28]	Attack rate [28]	Sources per attack session [6]
Value	5 minutes	500 requests/s	Around 1000

Based on Tables 1 and 2, we can conclude that it is not possible to deny the service of a cloud data center as a data center possesses profound resources against the attack capability of a DDoS attack.

On the other hand, we are interested in observing the workload of individual web sites for our following experiments. In order to obtain this data, we observed two popular web sites (one news web site and one e-business web site) of a data center of a major ISP. We counted the requests for each web site every 30 seconds for a day. We processed the data and present the number of requests in seconds in Fig. 2. From these results, we can see that the requests for a popular web site is usually less than 10 requests per second. It is generally unwise to reserve too many idle resources as it becomes costly. For the news web site, we suppose the owner reserves resources for a maximum need of 10 requests per second. As Moore *et al.* [28] indicated, the average attack rate is 500 requests or packets per second. This means a web site faces 50 times the workload of its maximum capacity. It is not difficult to conclude that a DDoS attack is highly likely to be successful. This confirms our claim that a DDoS attack is still a critical threat to individual cloud hosted services.

As discussed previously, we use average time in system as a metric for our performance evaluation in the following experiments. Therefore, let us firstly explore the average time in system for nonattack cases, which is modeled as an M/M/1 queue. We want to know the impact on the average time in a system from different arrival rates under different service rates. Following (11), we obtained the results of experiments shown in Fig. 3. These results indicated that when an IPS server is heavily loaded, e.g., $\mu = 10$ (therefore, $\rho_n \rightarrow 1$ when $\lambda \rightarrow 10$), T_n increases in an exponential way. On the other hand, when the IPS server's workload is suitable, e.g., $\mu = 15$ (therefore, $\rho_n \rightarrow 2/3$ when $\lambda \rightarrow 10$), T_n is relatively stable for various arrival rate λ.

From this experiment, we know that the workload of an IPS should be kept within a suitable range. If it is too low, say $\rho_n < 0.5$, then we waste some capability of the system. On the other hand, if it is too high, say $\rho_n \rightarrow 1$, then we degrade the quality of service for benign users. We summarize this in the following observation.

5.1 Observation 1

We prefer the busy rate as high as possible under the condition that the average time in system is acceptable.

TABLE 2
Estimated Key Resources of Amazon EC2

Resource	Servers	Bandwidth
Value	500,000	1Gb/Instance

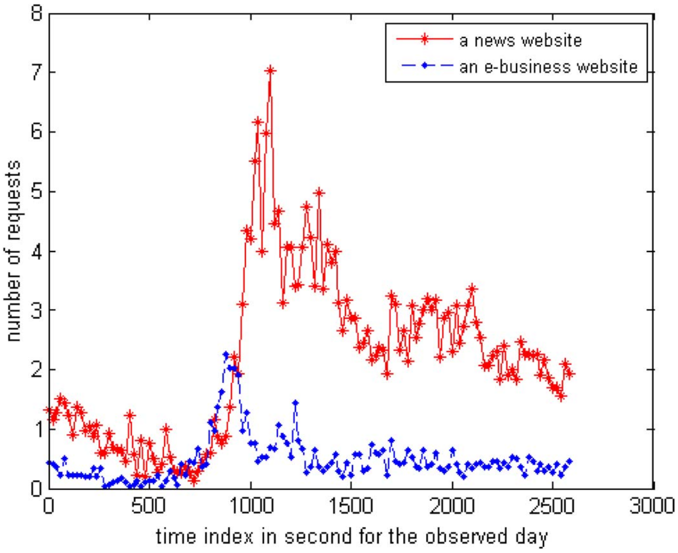


Fig. 2. Requests per second for two popular web sites of a major ISP data center.

Secondly, we studied the performance when a DDoS attack was ongoing. As previously discussed, we have multiple IPS servers in this case, and the model is M/M/m.

For the system of multiple IPS servers, π_0 is an important element, and is also involved in the calculation of other items. We expect a good understanding of π_0 against the number of duplicated servers (m) for a given busy rate. The experiment results are shown in Fig. 4.

In contrast to π_0, π_{m+} is also important to us because it is a critical point where incoming packets have to wait for service, which is expressed in (16), and the experimental results are shown in Fig. 5. The results indicate that: 1) for a given number of duplicated IPS servers, the higher ρ is, the less probability of packet queueing; 2) for a given ρ, the probability of packet queueing decreases when there are more duplicated servers (this is intuitively straightforward). From this perspective, we obtain the following observation.

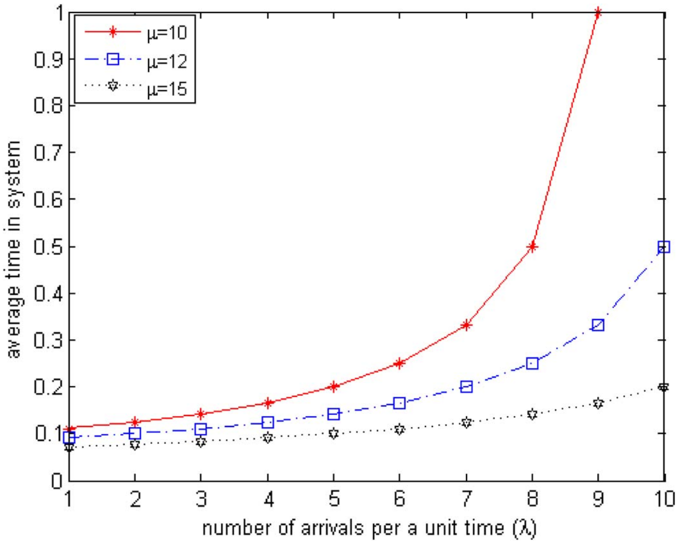


Fig. 3. Average time in system against arrival rate under different service rates for nonattack cases.

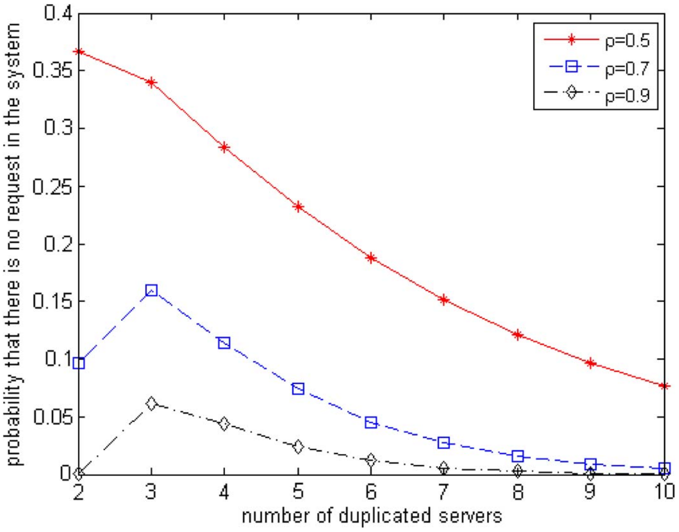


Fig. 4. Relationship between π_0 and the number of duplicated IPS servers for a given busy rate.

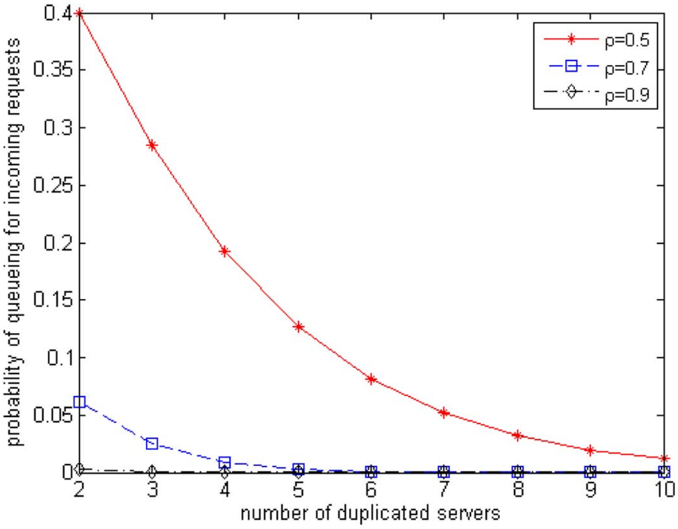


Fig. 5. Relationship between π_{m+} and the number of duplicated IPS servers for a given busy rate.

5.2 Observation 2

In order to reduce the queueing probability, we prefer the busy rate to be high.

We note there is a contradiction between *observation* 1 and *observation* 2. Intuitively, there should exist an equilibrium for the busy rate that balances the needs from both sides. However, this is beyond the scope of this paper, and will be an avenue for future research.

To evaluate the performance of the proposed mitigation method, we desperately want to know how we can beat an ongoing DDoS attack using minimum resources. In other words, how can we hold equation (22) under the constrains of equation (23) (and equation (24) if applicable).

In the following experiments, we set the service rate of the original IPS as $\mu = 10$, therefore, there are three variables, λ (arrival rate for nonattack cases), r (attack strength as defined before), and m (number of duplicated IPSs), which have an impact on our results.

In order to match our previous experiments, we conduct three experiments for $\lambda = 5, 7,$ and 9, respectively. For a given λ, we observe the variation of $f(r, m)$. The results are shown in Figs. 6a, 6b, 6c, which show complete information about the metric $f(r, m)$. As previously discussed, if $f(r, m) < 0$, this means the average time in system for the proposed method is greater than that of nonattack cases, namely, the quality of service for benign users in an attack case is worse than they expect. In order to guarantee the QoS, we need to keep $f(r, m) \geq 0$, which is of more interest to us. Therefore, we repeat the three simulations and only display the $f(r, m) \geq 0$ parts, as shown in Figs. 6a.1, 6b.1, 6c.1, respectively. When $f(r, m) > 0$, this means benign users enjoy an even better QoS than they had in nonattack cases. This occurs by the cloud service provider investing more resources into the service.

From the results of Figs. 6a.1, 6b.1, 6c.1, we find the solution space is roughly divided into two parts: the right hand part (low r and high m part) and the left hand part. Obviously, the right hand part is not what we expect because it requires a large amount of resources (represented by m) for a low attack strength case (represented by r).

CSPs prefer to minimize their investment of resources, namely, to make sure $f(r, m) \rightarrow 0^+$ any time. Based on Figs. 6a.1, 6b.1, 6c.1, we extract the critical points of $f(r, m) = 0$, and demonstrate them in Fig. 7.

The relationship between r and m in Fig. 7 looks linear. However, this is not true. We therefore list some of the numerical results in Table 3 for readers' reference.

In order to estimate the financial cost of mitigating DDoS attacks using our proposed strategy, we use Amazon EC2 as an example. Currently, the prices of Amazon EC2 Pricing for Standard On-Demand Instances are listed in Table 4 [29]. We take the default setting of a small Linux instance in our following calculation.

We suppose the legitimate traffic volume is 10 requests per second based on our real-world data set (refer to Fig. 2). At the same time, based on DDoS attack characteristics (refer to Table 1), we take the attack rate as 500 requests per second. Therefore, the attack strength is 50. Under different normal workloads (measured by busy rate), we need different numbers of duplicated IPSs to carry out the mitigation task. The number of duplicated IPSs can be extracted from Table 3. By combing all these parameters, we obtained a monetary cost in terms of duration of attacks as shown in Fig. 8.

We should note that a long time and high volume DDoS attack is very rare. For example, Moore *et al.* [28] have indicated that the average attack duration is around 5 minutes, and the rate of a repeat attack is quite low. This may contributed by a few reasons. First of all, long time DDoS attacks will expose botnets to defenders, and therefore, bots will be removed by network administrators. Secondly, it is hard for attackers to organize a large number of active bots to carry out lengthy attacks, e.g., time zones have an impact on the number of active bots [30].

In order to have a straight concept of the monetary cost, we list some of the numerical results from Fig. 8 in Table 5.

From Table 5, we can see the defense cost for most DDoS attacks on a victim is less than US$1 per month if the attack

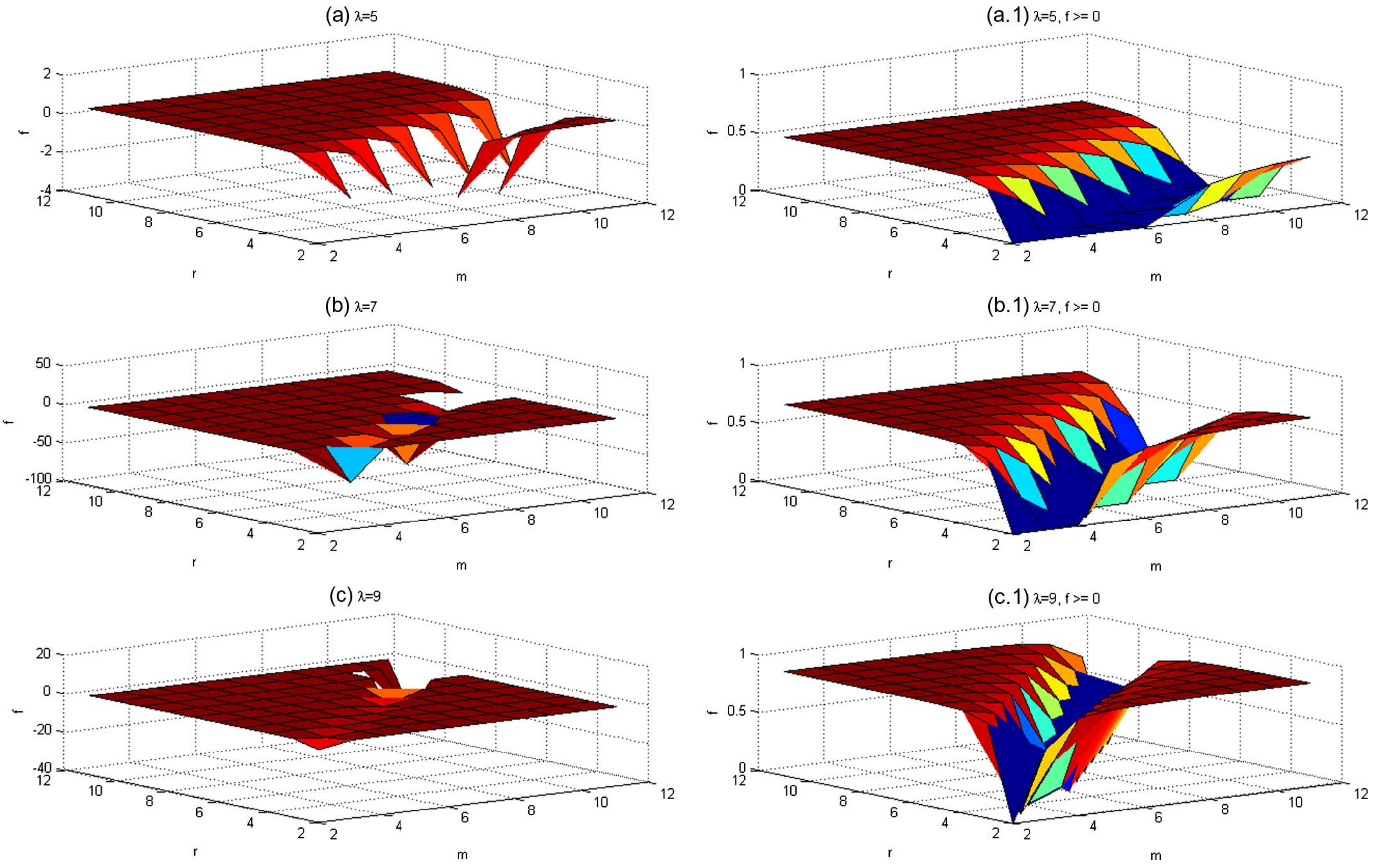


Fig. 6. Performance of defense systems under DDoS attack (compared to nonattack cases) with a different number of duplicated IPSs m, different attack strength r, and different arrival rate λ (with fixed service rate $\mu = 10$). (a) Function f with $\lambda = 5$, (a.1) function $f \geq 0$ with $\lambda = 5$. (b) function f with $\lambda = 7$, (b.1) function $f \geq 0$ with $\lambda = 7$. (c) function f with $\lambda = 9$, (c.1) function $f \geq 0$ with $\lambda = 9$.

happens every fortnight based on the observation of [28]. A dedicated attack for 1 day or 1 week costs defenders around US$50 or US$350, respectively. We note that this kind of lengthy attack occurs with a low probability as they can be easily found by CSPs, and subsequent actions can be taken to terminate them.

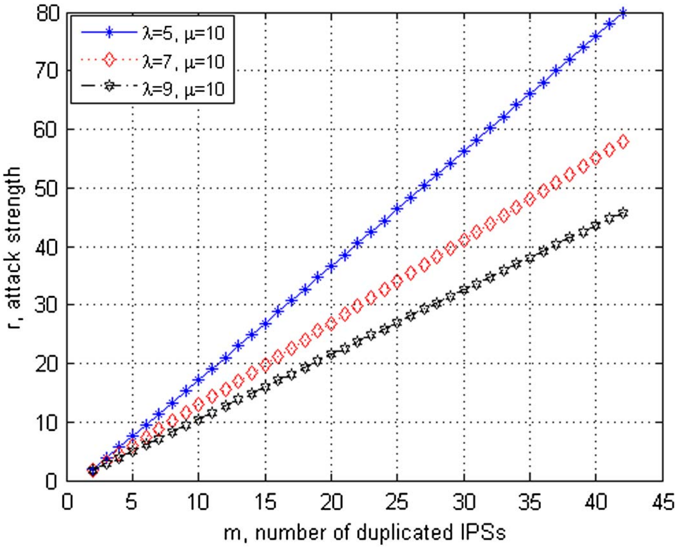


Fig. 7. Relationship between attack strength r and minimum number of duplicated IPSs to guarantee QoS for benign users.

Based on these results, we claim that the proposed mitigation strategy is practical and feasible.

6 Further Discussion

To the best of our knowledge, this paper is an early work to discuss mitigating DDoS attacks for individual cloud customers. As a new research field, there are many issues to be further investigated and improved. Due to the limitations of knowledge, space and time, we have only discussed the mechanism in this paper. There are many promising avenues to be explored further. We list some of them here based on our understanding.

First of all, the analysis model can be further improved. We establish the analysis model for an ongoing DDoS attack using the M/M/m model, which simplifies our analysis and makes our experiments feasible. In practice, we need to further identify the distributions that we used in this paper, especially the service rate distribution of the IPS servers.

TABLE 3
Critical Points for $f(r, m) = 0$ with $\mu = 10$

r ($\lambda = 5$)	17.15	36.601	46.40	56.22	66.07	75.93
r ($\lambda = 7$)	12.88	26.90	33.95	41.01	48.08	55.16
r ($\lambda = 9$)	10.52	21.53	27.05	32.57	38.09	43.62
m	10	20	25	30	35	40

TABLE 4
Amazon EC2 Pricing for Standard On-Demand Instances

Instance Type	Linux (per hour)	Windows (per hour)
Small(Default)	\$0.060	\$0.115
Medium	\$0.120	\$0.230
Large	\$0.240	\$0.460
Extra Large	\$0.480	\$0.920

TABLE 5
Some Numerical Mitigation Costs under Different DDoS Attack Durations

r (under $\mu = 10$)	5 minutes	1 hour	1 day	1 week
50 (with $\lambda = 5$)	\$0.15	\$1.8	\$43.2	\$302.4
50 (with $\lambda = 7$)	\$0.175	\$2.1	\$50.4	\$352.8
50 (with $\lambda = 9$)	\$0.23	\$2.76	\$66.24	\$463.7

Secondly, global optimization is expected. As we know, there are several parameters in the system, such as the service rate μ and the busy rate ρ_n for nonattack cases. These parameters have an impact on the resources we need for DDoS mitigation when an attack is ongoing.

Thirdly, we suppose the IPS system is ideal, namely, all attack packets are filtered and all benign packets go through. In practice, we need to consider the false negatives and false positives of the IPS system. Moreover, we may also include the protected server in our consideration of performance evaluation.

Finally, current clouds are distributed systems, and a cloud is usually a composite of a number of data centers. A cloud customer is generally hosted by one data center. If a data center runs out of reserved resources during a battle against a DDoS attack, the question remains how to use the reserved resources of other data centers to beat the ongoing attack.

7 Summary and Future Work

In this paper, we point out that DDoS attacks are still an effective tool for cyber criminals to shut down individual cloud customers, even though it is almost impossible to deny the service of a cloud platform. At the same time, we also note that a cloud possesses a potential to counter this kind of brute force attack by using its profound resources. Motivated by this, we design a strategy to dynamically allocate idle or reserved cloud resources to those cloud customers who are experiencing DDoS attacks in order to defeat the attacks, and at the same time guaranteeing the quality of service for benign users.

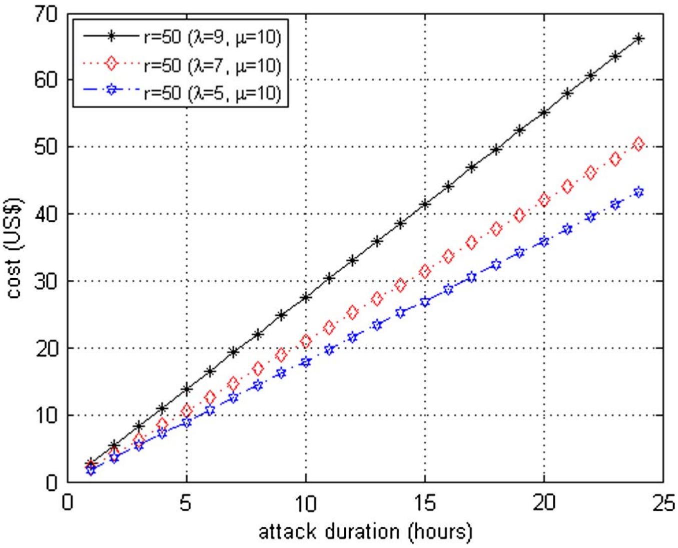


Fig. 8. Cost in US\$ for a victim who faces 50 times attack traffic of its legitimate traffic (which are measured with different arrival rate λ under the condition $\mu = 10$).

We establish a queueing theory based model for the proposed DDoS attack mitigation strategy in a cloud environment. We thoroughly analyze the proposed method. Extensive real-world data set based experiments and simulations confirm our claim that we can beat DDoS attacks on individual cloud hosted services with an affordable cost to cloud customers.

As a rarely explored new area of research, there is plenty of work expected to be completed in the near future. As future work, we firstly attempt to improve the M/M/m model to a more general model, such as the M/G/m model. Secondly, we want to explore what should we do if a cloud data center runs out of resources during a battle. Thirdly, we would like to discover whether it is possible for attackers to rent the resources of a cloud to carry out their attacks on servers hosted by the same or other clouds. Finally, real cloud environment tests for the proposed method are expected in the near future.

Acknowledgment

Y. Tian is the corresponding author.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, Feb. 2009.

[2] C. Peng, M. Kim, Z. Zhang, and H. Lei, "Vdn: Virtual Machine Image Distribution Network for Cloud Data Centers," in *Proc. INFOCOM*, 2012, pp. 181-189.

[3] S. Subashini and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1-11, Jan. 2011.

[4] R. Bhadauria, R. Chaki, N. Chaki, and S. Sanyal, "A Survey on Security Issues in Cloud Computing," *CoRR*, vol. abs/1109.5388, 2011.

[5] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the dos and ddos Problems," *ACM Comput. Surv.*, vol. 39, no. 1, pp. 1-3, 2007.

[6] M.A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "My Botnet is Bigger Than Yours (Maybe, Better Than Yours): Why Size Estimates Remain Challenging," in *Proc. 1st Conf. HotBots*, 2007, p. 5.

[7] D.K.Y. Yau, J.C.S. Lui, F. Liang, and Y. Yam, "Defending Against Distributed Denial-of-Service Attacks with Max-Min Fair Server-Centric Router Throttles," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 29-42, Feb. 2005.

[8] S. Yu, S. Guo, and I. Stojmenovic, "Can We Beat Legitimate Cyber Behavior Mimicking Attacks from Botnets?" in *Proc. INFOCOM, 2012, pp. 2851-2855.*

[9] Y. Chen, K. Hwang, and W.-S. Ku, "Collaborative Detection of ddos Attacks over Multiple Network Domains," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 12, pp. 1649-1662, Dec. 2007.

[10] J. Francois, I. Aib, and R. Boutaba, "Firecol, a Collaborative Protection Network for the Detection of Flooding ddos Attacks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1828-1841, Dec. 2012.

[11] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing," *IEEE Trans. Serv. Comput.*, vol. 5, no. 2, pp. 164-177, Apr./June 2012.

[12] J. Idziorek, M. Tannian, and D. Jacobson, "Insecurity of Cloud Utility Models," *IT Prof.*, vol. 15, no. 2, pp. 22-27, Mar./Apr. 2012.

[13] M.H. Sqalli, F. Al-Haidari, and K. Salah, "Edos-Shield-a Two-Steps Mitigation Technique against Edos Attacks in Cloud Computing," in *Proc. UCC*, 2011, pp. 49-56.

[14] Q. Wang, K. Ren, and X. Meng, "When Cloud Meets Ebay: Towards Effective Pricing for Cloud Computing," in *Proc. INFOCOM*, Mar. 2012, pp. 936-944.

[15] S. Yi, A. Andrzejak, and D. Kondo, "Monetary Cost-Aware Checkpointing and Migration on Amazon Cloud Spot Instances," *IEEE Trans. Serv. Comput.*, vol. 5, no. 4, pp. 512-524, Fourth Quarter 2012.

[16] J. Cao, K. Hwang, K. Li, and A. Zomaya, "Optimal Multiserver Configuration for Profit Maximization in Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1087-1096, June 2012.

[17] H. Wang, F. Wang, J. Liu, and J. Groen, "Measurement and Utilization of Customer-Provided Resources for Cloud Computing," in *Proc. INFOCOM*, 2012, pp. 442-450.

[18] R. Lua and K.C. Yow, "Mitigating ddos Attacks with Transparent and Intelligent Fast-Flux Swarm Network," *IEEE Netw.*, vol. 25, no. 4, pp. 28-33, July/Aug. 2011.

[19] P. Du and A. Nakao, "Ddos Defense as a Network Service," in *Proc. NOMS*, 2010, pp. 894-897.

[20] J. Chen, Y. Wang, and X. Wang, "On-Demand Security Architecture for Cloud Computing," *Computer*, vol. 45, no. 7, pp. 73-78, July 2012.

[21] R. Wartel, T. Cass, B. Moreira, E. Roche, M. Guijarro, S. Goasguen, and U. Schwickerath, "Image Distribution Mechanisms in Large Scale Cloud Providers," in *Proc. CloudCom*, 2010, pp. 112-117.

[22] J. Zhu, Z. Jiang, and Z. Xiao, "Twinkle: A Fast Resource Provisioning Mechanism for Internet Services," in *Proc. INFOCOM*, 2011, pp. 802-810.

[23] H. Khazaei, J.V. Misic, and V.B. Misic, "Performance Analysis of Cloud Computing Centers using m/g/m/m+r Queuing Systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 936-943, May 2012.

[24] H. Khazaei, J.V. Misic, V.B. Misic, and S. Rashwand, "Analysis of a Pool Management Scheme for Cloud Computing Centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 5, pp. 849-861, May 2013.

[25] H. Khazaei, J.V. Misic, and V.B. Misic, "Performance of Cloud Centers with High Degree of Virtualization Under Batch Task Arrivals," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2429-2438, Dec. 2013.

[26] J.F.C. Kingman, "The First Erlang Century-and the Next," *Queueing Syst.*, vol. 63, no. 1–4, pp. 3-12, Dec. 2009.

[27] L. Kleinrock, *Queueing Systems*. Hoboken, NJ, USA: Wiley, 1975, .

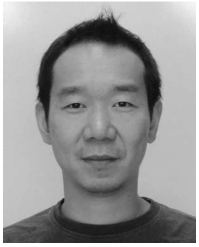
[28] D. Moore, C. Shannon, D.J. Brown, G.M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115-139, May 2006.

[29] [Online]. Available: http://aws.amazon.com/ec2/pricing/

[30] D. Dagon, C. Zou, and W. Lee, "Modeling Botnet Propagation using Time Zones," in *Proc. 13th NDSS*, 2006, pp. 1-18.

Shui Yu received the BEng and MEng degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1993 and 1999, respectively, and the PhD degree from Deakin University, Victoria, Australia, in 2004. Currently, he is a Senior Lecturer with the School of Information Technology, Deakin University, Victoria, Australia. His research interests include networking theory, network security, and mathematical modeling. Dr. Yu actively serves his research communities in various roles, which include the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, and three other international journals, IEEE INFOCOM TPC members, symposium co-chairs of IEEE ICC 2014, IEEE ICNC 2013 and 2104, and many different roles of international conference organizing committees. He has published nearly 100 peer-reviewed papers, including top journals and top conferences, such as IEEE TPDS, IEEE TIFS, IEEE TFS, IEEE TMC, and IEEE INFOCOM. He is a Senior Member of IEEE and a member of AAAS.

Yonghong Tian received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. Currently, he is a Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include computer vision and multimedia analysis and coding. Dr. Tian is currently a Young Associate Editor of the *Frontiers of Computer Science In China*. He is the author or coauthor of over 80 technical articles in refereed journals and conferences. He was the recipient of the Second Prize of National Science and Technology Progress Awards in 2010; the best performer in the TRECVID content-based copy detection (CCD) task (2010-2011); the top performer in the TRECVID retrospective surveillance event detection (SED) task (2009-2012); and the winner of the WikipediaMM task in ImageCLEF 2008. He is a Senior Member of the IEEE.



Song Guo received the PhD degree in computer science from the University of Ottawa, Ottawa, Canada, in 2005. Currently, he is a Senior Associate Professor at the School of Computer Science and Engineering, the University of Aizu, Aizu-Wakamatsu, Japan. His research interests are mainly in the areas of protocol design and performance analysis for reliable, energy-efficient, and cost-effective communications in wireless networks. Dr. Guo is an Associate Editor of the *IEEE Transactions on Parallel and Distributed Systems* and an editor of Wireless Communications and *Mobile Computing*. He is a Senior Member of the IEEE and the ACM.

Dapeng Oliver Wu received the PhD degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003. Since 2003, he has been on the faculty of Electrical and Computer Engineering Department at the University of Florida, Gainesville, USA where he is a Professor. His research interests are in the areas of networking, communications, signal processing, computer vision, and machine learning. Dr. Wu received the University of Florida Research Foundation Professorship Award in 2009, the AFOSR Young Investigator Program (YIP) Award in 2009, the ONR Young Investigator Program (YIP) Award in 2008, the NSF CAREER award in 2007, the IEEE Circuits and Systems for Video Technology (CSVT) Transactions Best Paper Award for Year 2001, and the Best Paper Awards in IEEE GLOBECOM 2011 and International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006. Currently, he serves as an Associate Editor for *IEEE Transactions on CSVT*. He is the founder of *IEEE Transactions on Network Science and Engineering*. He was the founding Editor-in-Chief of the Journal of Advances in Multimedia between 2006 and 2008, and an Associate Editor for *IEEE Transactions on Wireless Communications* and *IEEE Transactions on Vehicular Technology* between 2004 and 2007. He is also a guest-editor for *IEEE Journal on Selected Areas in Communications (JSAC)*, Special Issue on Cross-layer Optimized Wireless Multimedia Communications. He has served as the Technical Program Committee (TPC) Chair for IEEE INFOCOM 2012 and the TPC chair for IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium, and as a Member of the executive committee and/or technical program committee of over 80 conferences. He has served as the Chair for the Award Committee and the Chair of Mobile and wireless multimedia Interest Group (MobIG), Technical Committee on Multimedia Communications, IEEE Communications Society. He is an IEEE Fellow.