Impact Evaluation of DDoS Attacks on DNS Cache Server Using Queuing Model

Zheng Wang [1,2] and Shian-Shyong Tseng [3]

[1] Computer Network Information Center, Chinese Academy of Sciences
Beijing 100190 - China
[2] China Organizational Name Administration Center
Beijing 100028 - China
[e-mail: wangzheng@conac.cn]
[3] Department of Information Science and Applications, Asia University
Taichung 41354 - Taiwan
[e-mail: sstseng@asia.edu.tw]

Received November 6, 2012; revised March 8, 2013; accepted April 2, 2013; published April 30, 2013

Abstract

Distributed Denial-of-Service (DDoS) attacks towards name servers of the Domain Name System (DNS) have threaten to disrupt this critical service. This paper studies the vulnerability of the cache server to the flooding DNS query traffic. As the resolution service provided by cache server, the incoming DNS requests, even the massive attacking traffic, are maintained in the waiting queue. The sojourn of requests lasts until the corresponding responses are returned from the authoritative server or time out. The victim cache server is thus overloaded by the pounding traffic and thereafter goes down. The impact of such attacks is analyzed via the model of queuing process in both cache server and authoritative server. Some specific limits hold for this practical dual queuing process, such as the limited sojourn time in the queue of cache server and the independence of the two queuing processes. The analytical results are presented to evaluate the impact of DDoS attacks on cache server. Finally, numerical results are provided for further analysis.

Keywords: DNS, DDoS, Cache server, Queuing process

This work was supported in part by the National Key Technology R&D Program of China under the grant number 2012BAH16B00 and the National Science Foundation for Distinguished Young Scholars of China under the grant number 61003239.

http://dx.doi.org/10.3837/tiis.2013.04.017

1. Introduction

The Domain Name System (DNS) is a fundamental and indispensable component of the modern Internet [1-2]. In essence, the DNS is a globally distributed and decentralized database of network identities. Its most common use by far is to resolve host names into Internet addresses. Furthermore, a growing number of other applications, such as SMTP, ENUM and SIP also depend on the DNS in order to route messages through appropriate application level gateways [3]. As a result, the availability of the DNS can affect the availability of a large number of Internet applications. Ensuring the DNS data availability is an essential part of providing a robust Internet.

A Denial-of-Service (DoS) attack is an attack in which one or more machines target a victim and attempt to prevent the victim from doing useful work. Almost all Internet services are vulnerable to denial-of-service attacks of sufficient scale. In most cases, sufficient scale can be achieved by compromising enough end-hosts (typically using a virus or worm) or routers, and using those compromised hosts to perpetrate the attack. Such an attack is known as a Distributed Denial-of-Service (DDoS) attack. However, there are also many cases where a single well-connected end-system can perpetrate a successful DoS attack.

While authoritative servers are generally recognized as the victim of DDoS attacks, there is hardly awareness that cache servers are also vulnerable to the flooding DNS query traffic. We show in this paper that the incoming DNS requests received by cache servers should be temporally kept in it, waiting for the response from authoritative server. The sojourn of these unanswered requests occupies the resolution resources of cache server, or even exhausts them under massive attacking traffic. Although the retaining requests can be dropped by cache server when time out, the analytical results on the impact of such attacks are still necessary for performance evaluation and attack defending.

In this paper we model the query resolution service in cache server and authoritative server as a dual queuing process. The specific queuing model is characterized by limited sojourn time in the queue of cache server and the independence of the two queuing processes. This makes it different from the typical queuing processes which are extensively studied by previous works [4-7]. The queuing process is analytically solved and some important performance measures are provided. We also present numerical results as well as some further analysis based on them.

The rest of the paper is organized as follows: some related works are presented in Section 2; Section 3 will provide a dual queuing model for resolution of DNS queries and obtain a fairly complete solution for the queuing model; Some numerical results as well as further analysis of them are given in Section 4; Finally, Section 5 will conclude this paper.

2. Related Work

In recent years there have been a number of proposals for protecting the DNS against DoS or DDoS attacks. Yang et al. [8] have proposed to augment the DNS structure with additional pointers that are used in order to access children zones. But it cannot enhance the DNS resilience against DDoS attacks that target cache servers. Parka et al. [9] have proposed to add a lookup peer-to-peer service between the stub-resolvers and the cache servers, which can be used in order to defend against DDoS attack towards cache servers. Ballani et al. [10] have proposed to use the information stored in the stale cache to answer the query for mitigating the

impact of DoS attacks on DNS. Unfortunately, this proposal violates the semantics of record expiration as defined for DNS, which have less protocol compatibility and may hinder its adoption. All of the efforts mentioned above have not provided an analytical model for evaluating the impact of DDoS attacks on cache servers, which guides the design of protecting or defending mechanism.

Queuing theory is used to approximately model a real queuing situation or system, so the queuing behavior can be analyzed mathematically [11]. General queue model are extensively studied by previous works, and the main results can be found in [12]. For the particular problem of queuing with limited waiting time, various transient and stationary results are obtained for the system in [4-7]. But none of these investigations has covered the specific dual queue model for the impact analysis of DDoS attacks on cache server, which is solved by this paper. Moreover, the limited waiting time queue is inherently different from the limited sojourn time queue studied in this paper although the difference may be literally subtle.

The feasibility of the queuing modeling relies on the assumptions on the service process of DNS servers and the arrival process of DNS queries.

V. Bhaskar et al. discussed open queuing network models with single and multiple servers [13]. Queuing models containing both single and multiple servers, namely hybrid models, are considered in [14-16]. However, the service processes of servers modeled by these works are mostly complicated and hardly applicable to the formal analysis of queuing modeling. Bhaskar and Lavanya modeled a Pentium processor as a queuing network and deduced an equivalent single-queue–single-server model for the original queuing network [17]. The work paved the way to model the service of DNS name servers as a simple single-queue–single-server. Moreover, thanks to the overwhelmingly homogeneity of DNS traffic load on the DNS name servers, the reliability of the model is guaranteed.

One important research area in the context of networking focuses on developing traffic models which can be applied to the performance analysis and evaluation of the Internet [18-26]. Although it is proven not suitable to fit a simple Poisson-based model to capture internet traffic burstiness [18, 19], there are some Poisson process variations proposed to tackle the problem of burstiness [20-23]. One way is to use the so called compound Poisson process, where packet arrivals happen in bursts (or batches), the interbatch times are independent and exponentially distributed (that is, they represent a Poisson process), and the batch sizes are random [20]. Another Poisson-based arrival model addresses the limitations of the Poisson model by introducing dependence between packets traveling between the same end-nodes [21]. Thomas et al. [22] revisit the Poisson assumption by new measurements and show that unlike the older data sets, current network traffic can be well represented by the Poisson model for sub-second time scales. Besides, Cao et al. [23] demonstrated that as the load increases, the laws of superposition of marked point processes push the arrivals toward Poisson. And if the link speed is high enough, the traffic can get quite close to Poisson and independence before the push-back begins in force. These results reverse the commonly-held presumption that Internet traffic is everywhere busty and that multiplexing gains do not occur. One of the merits of Poisson model lie in its analytical simplicity for the performance evaluation of network processing equipments e.g. the DNS cache servers and name servers, which is usually too complicated to derive the closed-form expressions under the queuing model. Thus many previous works on performance analysis of network systems and behavior investigation of network traffic fed Poisson arrival traffic into the queuing system [24-26]. Based on the above considerations, Poisson process, representing the query arrival process for the queuing model in this paper, is an approximate modeling of the real DNS traffic at least for

the preliminary investigation and meanwhile capable of simplifying our mathematical analysis.

Another area related to this research is on the DoS or DDoS attack detection. Ruoyu Yan et al. proposed a new scheme to detect DDoS attacks within a router [28]. The scheme is based on the tree characteristic of DDoS attack, features of IF flows and properties of adaptive filters. Zhu Jian-Qi et al. studied the characteristics of traffic composition in the local-world network environment, and presented an effective method for detecting DoS attacks [29]. These work mainly focus on the DoS or DDoS attack traffic analysis while our work is more concerned about modelling the processing of DoS or DDoS attack traffic.

3. Queuing Model for DNS Resolution

In this section, we present our analytic framework for modeling the DNS resolution. Clients rely on DNS primarily to map service names to the IP addresses of the corresponding servers. Typically, clients issue their queries to a local resolver (recursive server). The local resolver then maps each query to a matching resource record set (hereon simply referred to as a matching record) and returns it in the response. Each record is associated with a time-to-live (TTL) value and resolvers are allowed to cache a record till its TTL expires; beyond this, the record is evicted from the cache. According to the DNS name resolution procedure, we break up the system into two components for our analysis:

1. The cache servers which are analyzed as servers with an equivalent service rate shared with other cache servers from the authoritative server.
2. The authoritative server with the first come, first serve discipline, no capacity constraints, and general arrival distribution aggregated from all cache servers and general service time distribution.

Propagation delays between cache server and authoritative server are usually much lower than the queuing delays. In this paper, we thus focus on the service queuing delays at both servers.

3.1 Queuing Model for Cache Servers

A cache server first searches its cache for a DNS request emitted by a client once receiving it. If hit, the cache server response immediately with the cached records. Otherwise, the cache server has to forward the request to the authoritative server. Since the resource cost as well as the response delay in the case of cache hit is negligible compared with that of cache miss, we only consider the cache miss requests in our analytical framework. The cache server queries the authoritative server for the cache misses DNS question, meanwhile maintains its status as unanswered in a waiting queue. The queue is timed according to a set value of timeout and is scheduled to drop the expired waiting queries. The waiting queries also leave the queue when their answers come from the authoritative server. As the status of each query in the waiting queue should be kept by the cache server, larger waiting queue means more system resources occupation. Furthermore, the length of waiting queue is largely limited by the timeout mechanism which dequeued the persistently unanswered queries. However, the number of waiting queries in cache server is likely to surge under DDoS flooding queries, which exhaust the resources of cache server and thereby cause the failure of DNS resolution for legitimate queries. Therefore, timeout mechanism may play an indispensable role in alleviating the impact of DDoS attacks. It should be noted that the timeout mechansim is not able to make any distinction between DDoS attacking queries and normal queries.

More precisely, timeout can be defined as being triggered by a timer. The timer is started at

the time the query is sent from the cache server to the authoritative server, and meanwhile, put into the waiting queue as an unanswered query. Once its timer reaches the set timeout length, the query staying unanswered is evicted from the waiting queue. The other case that the query in the waiting queue is dequeued happens when the query gets an answer from the authoritative server before its timer reaches the set timeout length. The pseudocode for the timeout is expressed as follows:

```
The query is issued from the cache server and put into the waiting queue;
Its timer starts;
While (its timer has not reached the set timeout length)
{
        If (receipt of an answer from the authoritative server)
        {
        break;
        }
}
The query is dequeued;
```

The queuing model is illustrated in **Fig. 1**. Clients issue their queries to the local cache server. The cache server searches its cache for the matching record and then returns the matching record to the clients if cache hits. Otherwise, the cache miss queries are forwarded by the cache server to the authoritative servers. Meanwhile, the cache miss queries are maintained in a waiting queue in the cache server, awaiting the response from the authoritative server. The waiting queue conducts a time out mechamism which limits the sojourn time of its queries. Those queries staying long than a limited sojourn time and getting no response are evicted from the waiting queue. All queries arriving at the authoritative server are also queued to be provided with a resolution service. The responses are returned by the authoritative server to the cache server. The response finds its matching query in the waiting queue of the cache server, and the matching query is thus evicted from the waiting queue. Finally, the cache server returns the response to its clients.

Let there be N cache servers $C_1, C_2, \ldots, C_N$ in the whole system and on an average, λ_i DNS requests that miss the cache arrive at C_i (i =1, 2, …, N) per second with an arbitrary and independent interarrival time distribution $F_i(\xi)$. Let the average service rate at C_i be μ_i (i =1, 2, …, N) per second. Because cache servers rely on authoritative server to resolve the queries, μ_i is actually the equivalent service rate of sojourning queries (not time out) allocated by authoritative server for C_i (i =1, 2, …, N).

Definition 1. *A queuing process of limited sojourn time is the queuing process that no customer can stay in the server longer than a time interval.*

The definition of "limited sojourn time" is slightly different from that of the frequently cited "limited waiting time" [4-7]. The latter holds that when a customer starts to get service before the limited time, it should never be interrupted until the service time ends, while the former can stop the service of a customer at the time limit. Thus it may happen that a customer leaves the system without being completely served under limited sojourn time.

According to the time out mechanism of cache servers, let us suppose the queuing process in

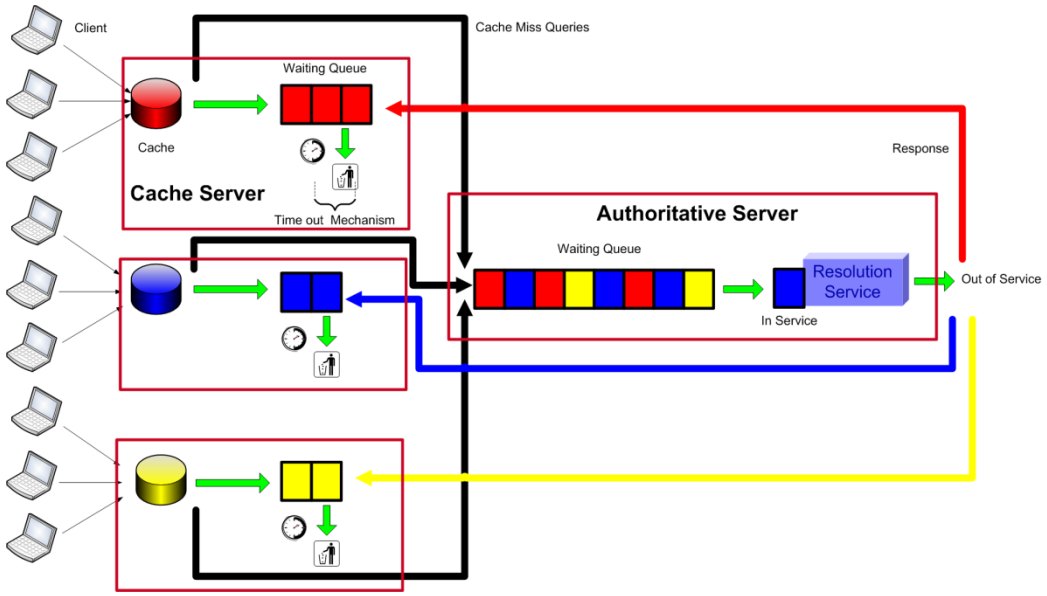

Fig. 1. Illustration of queuing model of DNS name resolution

each cache server C_i is a queuing process of limited sojourn time η_i (i =1, 2, …, N).

3.2 Queuing Model for Authoritative Servers

The authoritative server receives queries from a set of cache servers and provides them the resolution service with the first come, first serve discipline. Thus all queries of different sources of cache servers are queued indiscriminately in the authoritative server. The authoritative server usually does not explicitly adopt a time out mechanism. So we do not pose any waiting time limits on the queries in the queue. Note that not all queries served by the authoritative server are useful for reducing the length of waiting queue in the cache server. This is due to the fact that some of them may already be abandoned by the cache server following the time out mechanism.

Let the aggregated query rate arriving at the authoritative server be λ, and it is given by

$$\lambda = \sum_{i=1}^{N} \lambda_i \tag{1}$$

The authoritative server has no capacity constraints, and its average service rate is μ requests per second with an arbitrary interservice time distribution $F_A(\xi)$. Once the request finishes its service in the authoritative server and leaves it, we assume that the same request also leaves its source cache server at exactly the same time if it does not time out. This is based on the above mentioned neglection of the propagation delay of the response from the authoritative server to the cache server. Note that the authoritative server is unware of whether the requests residing in it has timed out in the cache servers, so its service is independent of the time out mechanism of the cache servers.

3.3 Queuing Model for DNS Resolution

In this section, we provide the analytical solution for the proposed queuing model in Section 4. To simplify our discussion, we assume that the query arrival process in cache servers and the

service process of authoritative server are both Poisson process, thus $F_i(\xi)$ (i =1, 2, …, N) yields

$$F_i(\xi) = \lambda_i e^{-\lambda_i \xi}, \ \xi > 0 \tag{2}$$

$F_A(\xi)$ yields

$$F_A(\xi) = \mu e^{-\mu \xi}, \ \xi > 0 \tag{3}$$

Lemma 1. The aggregated query arrival process at the authoritative server is a Poisson process with the mean arrival rate as λ.

Lemma 2. The equivalent service process for the queries from cache server C_i is a Poisson process with the mean service rate as μ_i (i=1,2,…, N)

$$\mu_i = \frac{\lambda_i}{\lambda} * \mu \tag{4}$$

And the N equivalent service processes for the queries from cache server C_1, C_2, …, C_N are independent.

Lemma 3. Let the sojourn time of queries from cache server C_i in the authoritative server be S_i (i =1,2,…, N). The distribution of S_i yields

$$P(S_i > t) = e^{-\mu_i(1-\rho_i)t}, \quad t \geq 0 \tag{5}$$

Where ρ_i is the ratio of the arrival and service rate

$$\rho_i = \lambda_i / \mu_i \tag{6}$$

Proof: According to Lemma 2, the equivalent service process for queries from cache server C_i in the authoritative server is μ_i (i=1, 2,…, N). And the arrival process is also a Possion process with rate λ_i (i=1, 2,…, N). Therefore the queuing process for queries from cache server C_i in the authoritative server is viewed as an equivalent M/M/1 process (i=1, 2,…, N). The solution of this M/M/1 process holds (5) [27]. □

Theorem 1. Let the sojourn time of queries in cache server C_i be $S_i^{'}$ (i =1, 2,…, N). The mean of $S_i^{'}$ is given by

$$E(S_i^{'}) = \frac{1 - e^{-\mu_i(1-\rho_i)\eta_i}}{\mu_i(1-\rho_i)} \tag{7}$$

Proof: According to Lemma 3, we have the proportion of time out queries in cache server C_i (i =1, 2, …, N)

$$P(S_i > \eta_i) = e^{-\mu_i(1-\rho_i)\eta_i} \tag{8}$$

The probability density function of $S_i^{'}$ yields

$$p_i^{'}(t) = \begin{cases} p_i(t) & 0 \leq t < \eta_i \\ P(S_i > \eta_i) & t = \eta_i \end{cases} \tag{9}$$

Where $p_i(t)$ is the probability density function of S_i, namely, the sojourn time of queries from cache server C_i in the authoritative server.

$$p_i(t) = \mu_i(1-\rho_i) e^{-\mu_i(1-\rho_i)t}, \quad t \geq 0 \tag{10}$$

Thus the mean of $S_i^{'}$ is given by

$$E\left(S_i^{'}\right)=\int_0^{\eta_i} tp_i^{'}(t)dt \tag{11}$$

Plugging (8) and (10) into (9) and then (9) into (11), we get (7). □

Corollary 1. Let the number of queries maintained in cache server C_i be $L_i^{'}$ (i =1, 2, …, N). The mean of $L_i^{'}$ yields

$$E\left(L_i^{'}\right)=\frac{\rho_i\left(1-e^{-\mu_i(1-\rho_i)\eta_i}\right)}{(1-\rho_i)} \tag{12}$$

Proof: According to Little's law [27], a relation between $E\left(L_i^{'}\right)$, $E\left(S_i^{'}\right)$ and λ_i holds

$$E\left(L_i^{'}\right)=\lambda_i E\left(S_i^{'}\right) \tag{13}$$

Plugging (7) into (13), we obtain (12). □

Lemma 4. Let the waiting time of queries from cache server C_i in the authoritative server be w_i (i =1,2,…, N). The probability density function of w_i is given by

$$p_{w_i}(t)=\lambda_i(1-\rho_i)e^{-\mu_i(1-\rho_i)t},\ t>0 \tag{14}$$

Proof: According to Lemma 2, the equivalent service process for queries from cache server C_i in the authoritative server is μ_i (i=1, 2,…, N). And the arrival process is also a Possion process with rate λ_i (i=1, 2,…, N). Therefore the queuing process for queries from cache server C_i in the authoritative server is viewed as an equivalent M/M/1 process (i=1, 2,…, N). The solution of this M/M/1 process holds (14) [27]. □

Definition 2. *The service time of a query in the cache server is the time between the instant when it gets service in the authoritative server and time out or the instant when the service ends*.

Another important performance measure is the mean service time of queries in cache servers. Here since the cache server only forwards requests rather than serves requests, the service time is actually the virtual one observed from the server and it may be cut short by the limit of waiting time. The service time starts when a query gets service at the server and does not time out in the cache server. And it ends when the query times out at its cache server without being completely served, or when its service finishes at the server if its total sojourn time is under the limits of waiting in the cache server. Another special case in our analysis is that when a query times out before it gets service, its service time is zero.

Theorem 2. Let the service time of queries from cache server C_i in the authoritative server be $d_i^{'}$ (i =1, 2,…, N). The mean of $d_i^{'}$ is given by

$$E\left(d_i^{'}\right)=1-e^{-\eta_i\mu_i}-\rho_i e^{-\mu_i(1-\rho_i)\eta_i}\left(1-e^{-\eta_i\lambda_i}\right) \tag{15}$$

Proof: Let the service time of queries from cache server C_i in the authoritative server be d_i (i =1, 2,…, N). According to Lemma 3, the probability density function of d_i is given by

$$p_{d_i}(t)=\mu_i e^{-\mu_i t},\ t>0 \tag{16}$$

Due to the memoryless property of Poisson process, d_i and w_i are independent variables. The sojourn time S_i yields

$$S_i=d_i+w_i \tag{17}$$

The service time of queries from cache server C_i in the authoritative server $d_i^{'}$ is determined by the sojourn time S_i, the waiting time w_i and the service time d_i (i =1, 2,…, N). As

discussed above, the function of $d_i^{'}$ varies under three kinds of conditions, as formulated as follows:

• Condition I: $0 < S_i \le \eta_i$. The queries in cache servers can be completely served by the authoritative server if their sojourn times are so small as to not exceed the limit of sojourn time. Therefore $d_i^{'}$ is exactly the service time in the authoritative server d_i. The conditional distribution of $d_i^{'}$ holds

$$p\left(d_i^{'} \middle| 0 < S_i \le \eta_i\right) = p\left(d_i \middle| 0 < S_i \le \eta_i\right) \tag{18}$$

• Condition II: $S_i > \eta_i$ and $0 < w_i < \eta_i$. The sojourn time is cut short, but the query still has chance to be served while the service is terminated by the limit of sojourn time before it finishes. The service time in cache servers is the residual time of η_i subtracting the waiting time. The conditional distribution of $d_i^{'}$ is given by

$$p\left(d_i^{'} \middle| S_i > \eta_i, 0 < w_i < \eta_i\right) = p\left(\eta_i - w_i \middle| S_i > \eta_i, 0 < w_i < \eta_i\right) \tag{19}$$

• Condition III: $S_i > \eta_i$ and $w_i \ge \eta_i$. The query times out in cache servers before they are served by the authoritative server. So its service time is zero in cache servers. Note $S_i > w_i$ always holds if $d_i > 0$. So this condition can be simplified as $w_i \ge \eta_i$. We have

$$p\left(d_i^{'} \middle| w_i \ge \eta_i\right) = \begin{cases} 1 & d_i^{'} = 0 \\ 0 & else \end{cases} \tag{20}$$

Thus the mean of $d_i^{'}$ is given by

$$\begin{aligned} E\left(d_i^{'}\right) = {} & E\left(d_i^{'} \middle| 0 < S_i \le \eta_i\right) P\left(0 < S_i \le \eta_i\right) + E\left(d_i^{'} \middle| S_i > \eta_i, 0 < w_i < \eta_i\right) \\ & P\left(S_i > \eta_i, 0 < w_i < \eta_i\right) + E\left(d_i^{'} \middle| w_i \ge \eta_i\right) P\left(w_i \ge \eta_i\right) \end{aligned} \tag{21}$$

Substituting (18), (19) and (20) into (21), we get

$$\begin{aligned} E\left(d_i^{'}\right) = {} & E\left(d_i \middle| 0 < S_i \le \eta_i\right) P\left(0 < S_i \le \eta_i\right) \\ & + E\left(\eta_i - w_i \middle| S_i > \eta_i, 0 < w_i < \eta_i\right) P\left(S_i > \eta_i, 0 < w_i < \eta_i\right) \end{aligned} \tag{22}$$

Plugging (5) in Lemma 3, (14) in Lemma 4, and (16) into the conditional probability $P\left(d_i = t \middle| 0 < S_i \le \eta_i\right)$, we have

$$\begin{aligned} P\left(d_i = t \middle| 0 < S_i \le \eta_i\right) &= \frac{P\left(d_i = t, 0 < S_i \le \eta_i\right)}{P\left(0 < S_i \le \eta_i\right)} \\ &= \frac{P\left(d_i = t, 0 < d_i + w_i \le \eta_i\right)}{P\left(0 < S_i \le \eta_i\right)} \\ &= \frac{P\left(d_i = t, 0 < t + w_i \le \eta_i\right)}{P\left(0 < S_i \le \eta_i\right)} \quad (d_i \text{ and } w_i \text{ are independent}) \\ &= \frac{P\left(d_i = t\right) P\left(0 \le w_i \le \eta_i - t\right)}{P\left(0 < S_i \le \eta_i\right)} \end{aligned}$$

$$= \frac{\mu_i e^{-\mu_i t}\left(1-\rho_i e^{-\mu_i(1-\rho_i)(\eta_i - t)}\right)}{P\left(0 < S_i \le \eta_i\right)}, \quad 0 < t \le \eta_i \tag{23}$$

Plugging (5) in Lemma 3 and (14) in Lemma 4 into the conditional probability

Table. 1 Prameters used in the scenarios

Scenario/ Parameters	*Number of Cache Servers*	*Query Traffic Profile*	*Time Out (s)*	*Service Capability (qps)*
1	1,000	*uniform distribution*	2	1,000
2	5,000	*uniform distribution*	2	1,000
3	1,000	*uniform distribution with heavy hitters*	2	1,000
4	1,000	*uniform distribution with heavy hitters*	2	3,000
5	1,000	*uniform distribution*	2	3,000
6	1,000	*uniform distribution with heavy hitters*	2, 0.5 *for heavy hitters*	1,000

$P\left(\eta_i - w_i = t \middle| S_i > \eta_i, 0 < w_i < \eta_i\right)$, we have

$$\begin{aligned} P\left(\eta_i - w_i = t \middle| S_i > \mu_i, 0 < w_i < \eta_i\right) &= \frac{P\left(\eta_i - w_i = t, S_i > \eta_i, 0 < w_i < \eta_i\right)}{P\left(S_i > \eta_i, 0 < w_i < \eta_i\right)} \\ &= \frac{P\left(\eta_i - w_i = t, d_i + w_i > \eta_i, 0 < w_i < \eta_i\right)}{P\left(S_i > \eta_i, 0 < w_i < \eta_i\right)} \\ &= \frac{P\left(w_i = \eta_i - t, d_i > t\right)}{P\left(S_i > \eta_i, 0 < w_i < \eta_i\right)} \ (d_i \text{ and } w_i \text{ are independent}) \\ &= \frac{P\left(w_i = \eta_i - t\right)P\left(d_i > t\right)}{P\left(S_i > \eta_i, 0 < w_i < \eta_i\right)} \\ &= \frac{\lambda_i\left(1-\rho_i\right)e^{-\mu_i(1-\rho_i)\eta_i - \rho\mu_i t}}{P\left(S_i > \eta_i, 0 < w_i < \eta_i\right)}, \quad 0 < t \le \eta_i \end{aligned} \tag{24}$$

Substituting (23) and (24) into (22), we get

$$\begin{aligned} E\left(d_i^{'}\right) &= \int_0^{\eta_i} P\left(d_i = t \middle| 0 < S_i \le \eta_i\right) P\left(0 < S_i \le \eta_i\right) dt \\ &+ \int_0^{\eta_i} P\left(\eta_i - w_i = t \middle| S_i > \mu_i, 0 < w_i < \eta_i\right) P\left(S_i > \eta_i, 0 < w_i < \eta_i\right) dt \\ &= \int_0^{\eta_i} \mu_i e^{-\mu_i t}\left(1-\rho_i e^{-\mu_i(1-\rho_i)(\eta_i - t)}\right) dt + \int_0^{\eta_i} \lambda_i\left(1-\rho_i\right) e^{-\mu_i(1-\rho_i)\eta_i - \rho\mu_i t} dt \\ &= 1 - e^{-\eta_i\mu_i} - \rho_i e^{-\mu_i(1-\rho_i)\eta_i}\left(1-e^{-\eta_i\lambda_i}\right) \end{aligned} \tag{25}$$ □

Theorem 3. Let the proportion of queries served by authoritative server but eventually abandoned by the cache servers be P_w, we have

$$P_w = \frac{\sum_{i=1}^{N} \lambda_i e^{-\mu_i(1-\rho_i)\eta_i}}{\lambda} \tag{26}$$

Proof: The total of queries abandoned by cache servers equals the summation of those queries time out in each cache server. Thus P_w holds

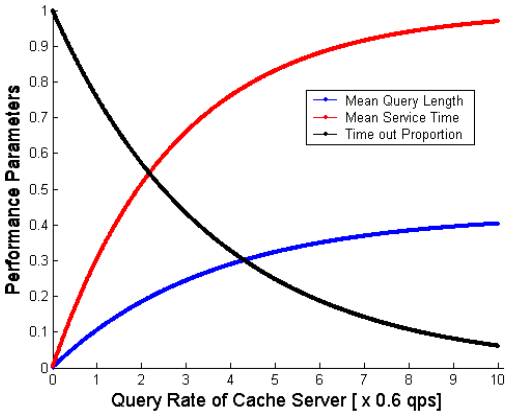

Fig. 2. Performance parameters for Scenario 1

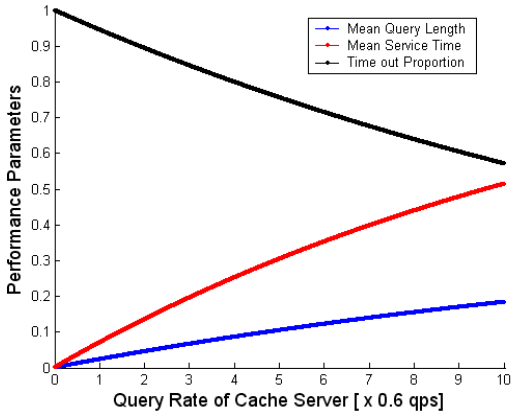

Fig. 3. Performance parameters for Scenario 2

$$P_w = \frac{\sum_{i=1}^{N} \lambda_i P\left(S_i > \eta_i\right)}{\lambda} \tag{27}$$

Plugging (8) derived by Lemma 3 into (27), we obtain (26). □

P_w actually expresses the invalid queries served by authoritative server, since the responses returned by authoritative server are no longer useful for them. For the efficiency of domain resolution service, P_w should stand at a low proportion.

4. Numerical Results

To demonstrate the impact of DDoS attacks on DNS cache server using queuing model, numerical results are provided for six very different scenarios of number of cache servers, query traffic profile, time out setting and service capability of authoritative server. For each set of parameters, we present the mean number of queries, the mean service time and the time out proportion in each cache server. The parameters that remain fixed across all scenarios are as follows: the aggregated query rate excluding the attacking heavy hitter cache server λ = 300 *qps*; the number of heavy hitter cache servers N_h = 3 and the query rate of each of them λ_h = 100 *qps*. The other parameter settings are given in **Table 1**.

(1) Scenario 1: Let the query traffic be distributed uniformly from 0 to the maximum among all cache servers. And the other parameters are shown in **Table 1**. The results are illustrated in **Fig. 2**. We see that cache servers generating high query rates receive more service time than those of lower rate. This explains the time out proportion bias against the light-loaded cache servers. Therefore we can conclude that under the best efforts service of authoritative server, the resources of authoritative server are preempted by the heavy-loaded cache servers, which account for their better system performance compared with their light-loaded counterparts.

Note that here the principle of more queries, more service does not lead to the fairness between cache servers under the free competition. The mean query length residing at cache servers grows with the increase of their query loads. This indicates resolution capability is demanding for busy cache servers.

(2) Scenario 2: We enlarge the amount of cache servers and remain the other parameters unchanged as given by **Table 1**. The results are shown in **Fig. 3**. The significant decrease in service time and increase in time out proportion even for heavy loaders pronounce the performance deterioration in this scenario. Recalling that the total query rate holds the same as Scenario 1, we can only owe this performance decline to increase in number of cache servers. The DNS servers authoritative for hot zones are widely and heavily requested by thousands of cache servers, therefore more efforts should be taken to ensure their resolution performance.

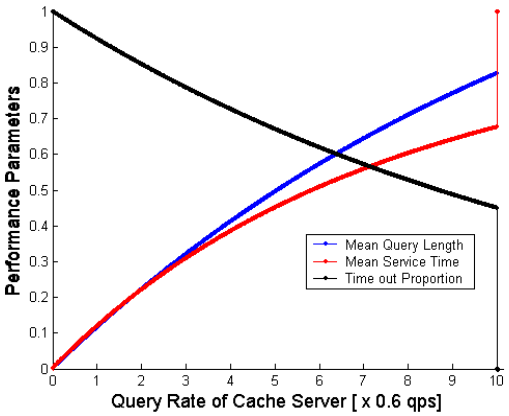

Fig. 4. Performance parameters for Scenario 3

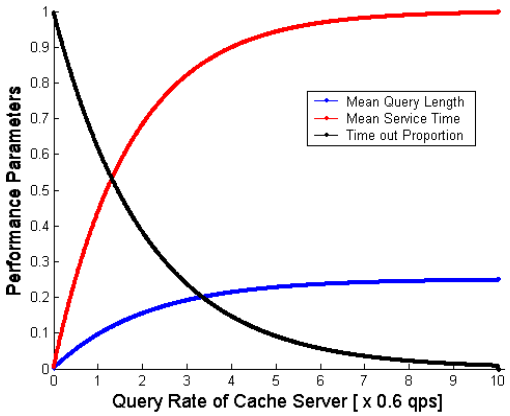

Fig. 5. Performance parameters for Scenario 4

(3) Scenario 3: To investigate the performance under DDoS attacks on cache servers, two heavy hitters are added to Scenario 1. We see in **Fig. 4** that the queries from the victim cache servers have little chance to time out while the strangling impacts on the performance of other caches servers under legitimate requests are apparent. Because the authoritative servers have been overloaded by the flooding traffic from the victim cache server pounded by DDoS attacks, share of resources for other cache servers are thus squeezed to sustain the quality of resolution service at a normal level.

(4) Scenario 4: As one of the usual solutions available for the protection of DDoS attacks is over provision, we evaluate the effects of such counter measure in this scenario by tripling the service rate of authoritative server. And the other parameters stay the same as Scenario 3. The results are shown in **Fig. 5**. As expected, both the time out ratio and the mean queue length are greatly improved in contrast to Scenario 3. Another observation tells that the flooded cache servers are no longer so clearly distinguished from others measured by performance parameters. The numerical results obtained by queuing model match closely with the practices of DNS operation.

(5) Scenario 5: To compare the results of over provision under legitimate traffic and DDoS attacks, we show the performance parameters for legitimate traffic in **Fig. 6**. The time out ratio and the mean queue length are better in **Fig. 6** than those in **Fig. 5**. The competitions of resources in authoritative servers among cache servers are mitigated due to the enhanced service capability and low level of service requests. Therefore the performance parameters stay relatively close among cache servers with varied loads. This explains the above mentioned concept that the root cause of unfairness is the resource competition.

(6) Scenario 6: Finally we study the effects of changing the setting of time out for the victim cache server under DDoS attacks. We reset the interval of time out as 0.5 second for victim cache servers. This is based on the hope that earlier time out would help ease the burden of flooding traffic on the victim cache servers. Unfortunately, the measure does not seem to take effects as illustrated in **Fig. 7**. The main reason lies in the fact that the time out proportion is already minor for even for the setting of 2 seconds(less than 0.001) , thereby lower value of time out setting brings negligible improvements by decreasing from less than 0.001 to less than 0.0001.

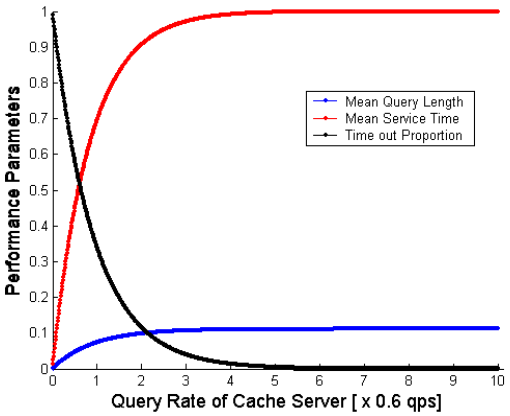

Fig. 6. Performance parameters for Scenario 5

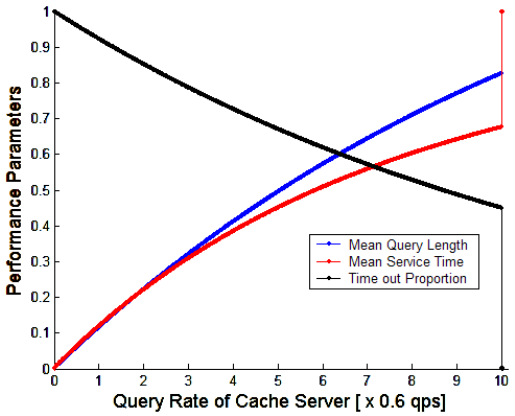

Fig. 7. Performance parameters for Scenario 6

5. Conclusion

This paper studies the impact of the flooding DNS query traffic on the DNS cache server by modelling the DNS query resolution service. The work provides the following major contributions. 1) The DNS query resolution service in cache server and authoritative server is modelled as a dual queuing process. 2) The queuing process is analytically solved and some important performance measures are provided. 3) Numerical results as well as further analysis are presented to evaluate the impacts of DDoS attacks on cache servers.

References

[1] P. Mockapetris, Domain names – concepts and facilities, *Internet Request for Comments* (*RFC 1034*), November 1987.
[2] P. Albitz and C. Liu, *DNS and BIND*, O'Reilly and Associates, 1998.
[3] H. Rood, "What is in a name, what is in a number: some characteristics of identifiers on electronic networks," *Telecommunications Policy*, vol.24, pp.533-552, 2000. Article (CrossRef Link)
[4] Lajos Takács, "A single-server queue with limited virtual waiting time," *Journal of Applied Probability*, vol.11, no.3, pp.612-617, 1974. Article (CrossRef Link)
[5] Do Le Minha, "The GI/G/1 queue with uniformly limited virtual waiting times; the finite dam," *Advances in Applied Probability*, vol.12, no.2, pp. 501-516, 1980.Article (CrossRef Link)
[6] Do Le Minha, "The single-server queue with uniformly limited actual waiting times," *Optimization*, vol.12, no.4, pp.607-621, 1981.
[7] J. Van Velthoven, B. Van Houdt, and C. Blondia, "On the probability of abandonment in queues with limited sojourn and waiting times," *Operations Research Letters*, vol.34, no.3, pp. 333-338, 2006. Article (CrossRef Link)

[8] H. Yang, H. Luo, Y. Yang, S. Lu, and L. Zhang. "HOURS: Achieving DoS resilience in an open service hierarchy," in *Proc. of the International Conference on Dependable Systems and Networks*, Palazzo dei Congressi, Florence, Italy, pp.83-93, 2004.
[9] K. Parka, V. Pai, L. Peterson, and Z. Wang. "CoDNS: Improving DNS performance and reliability via cooperative lookups," in *Proc. of the 6th conference on Symposium on Opearting Systems Design & Implementation*, San Francisco, CA, pp.14-14, 2004.
[10] Hitesh Ballani and Paul Francis. "Mitigating DNS DoS attacks," in *Proc. of the 15th ACM conference on Computer and communications security*, Alexandria, Virginia, pp.189-198, 2008.Article (CrossRef Link)
[11] D.M. Koppelman, "Congested Banyan network analysis using congested-queue states and neighboring queue effects," *IEEE/ACM Transactions on Networking*, vol.4, no.1, pp. 106-111, 2006. Article (CrossRef Link)
[12] H.C. Tijms, *Algorithmic Analysis of Queues*, Wiley, Chichester, 2003.
[13] V. Bhaskar, L. Joiner, "Modeling scheduled dataflow architecture – an open queuing network model approach," *Int. J. Pure Appl. Math.* vol. 18, no. 3, pp. 271-283, 2005.
[14] V. Bhaskar, K. Adjallah, "A hybrid open queuing network approach for multi-threaded dataflow architecture," *Int. J. Comput. Commun*. vol. 31, no. 17, pp. 4098-4106, 2008.Article (CrossRef Link)
[15] V. Bhaskar, K. Adjallah, "A hybrid closed queuing network approach to model dataflow in network distributed processors," *Int. J. Comput. Commun*. vol. 31, no. 1, pp. 119-128, 2008.Article (CrossRef Link)
[16] V. Bhaskar, "A hybrid closed queuing network model for multi-threaded dataflow architecture," *Int. J. Comput. Electric. Eng*. vol. 31, no. 8, pp. 556-571, November 2005.Article (CrossRef Link)
[17] V. Bhaskar, G. Lavanya, "Equivalent single-queue–single-server model for a Pentium processor," *Applied Mathematical Modelling*, vol. 34, no. 9, pp. 2531-2545, September 2010.Article (CrossRef Link)
[18] V. Paxson, S. Floyd, "Wide-area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226-244, June 1995.Article (CrossRef Link)
[19] R. H. Riedi, W. Willinger, "Towards an improved understanding of network traffic dynamics," *Selfsimilar Network Traffic and Performance Evaluation*, Wiley, 2000, chapter 20, pp. 507-530. Article (CrossRef Link)
[20] A. J. Marie, Y. Calas, and T. Alemu, "On the compromise between burstiness and frequency of events," *Performance Evaluation*, vol. 62, no. 1-4, pp. 382-399, 2005.Article (CrossRef Link)
[21] R. Jain, S. Routhier, "Packet Trains--Measurements and a New Model for Computer Network Traffic," *IEEE Journal on Selected Areas in Communications*, vol. 4, no. 6, pp. 986-995, 1986.Article (CrossRef Link)
[22] T. Karagiannis, M. Molle, and M. Faloutsos, "A Nonstationary Poisson View of Internet Traffic," in *Proc. of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies*, San Francisco, CA, pp. 1558-1569, 2004.
[23] W. Cleveland, D. Lin, and D. Sun, "Internet traffic tends toward Poisson and independent as the load increases," *Nonlinear Estimation and Classification*, D. Denison, M. Hansen, C. Holmes, B. Mallick, and B. Yu, Eds. New York, NY: Springer Verlag, Dec. 2002.
[24] D. Manjunath, B. Sikdar, "Input queued switches for variable length packets: analysis for Poisson and self-similar traffic," *Computer Communications*, vol.25, no.6, pp. 590-610, 2002.Article (CrossRef Link)
[25] A. Kamath, O. Palmon, and S. Plotkin, "Routing and Admission Control in General Topology Networks with Poisson Arrivals," *Journal of Algorithms*, vol. 27, no. 2, pp. 236-258, 1998.Article (CrossRef Link)
[26] T. Field, U. Harder, and P. Harrison, Network traffic behaviour in switched Ethernet systems, *Performance Evaluation*, vol. 58, no. 2-3, pp. 243-260, 2004.Article (CrossRef Link)
[27] S.M. Ross, *Introduction to Probability Models (6th ed.)*, Academic Press, London, 1997.

[28] Ruoyu Yan, Qinghua Zheng and Haifei Li, "Combining Adaptive Filtering and IF Flows to Detect DDoS Attacks within a Router," *KSII Transactions on Internet and Information Systems*, vol.4, no.3, pp. 428-451, June 2010.
[29] Zhu Jian-Qi, Fu Feng, Chong-kwon Kim, Yin Ke-xin and Liu Yan-Heng, "A DoS Detection Method Based on Composition Self-similarity," KSII Transactions on Internet and Information Systems, vol.6, no.5, pp. 1463-1478, May 2012.

Zheng Wang received the MS degree in Electrical Engineering from Institute of Acoustics, Chinese Academy of Sciences in 2006, and the PhD degree in Computer Science from Computer Network Information Center, Chinese Academy of Sciences in 2010. He is currently the director of Joint Labs in China Organizational Name Administration Center. His research interests are in the areas of network architecture, Domain Name System, and information systems.

Shian-Shyong Tseng received his Ph.D. degree in Computer Engineering from National Chiao Tung University (NCTU) in 1984. From Aug. 1983 to July 2009, he was on the faculty of the Department of Computer and Information Science at National Chiao Tung University. From Aug. 2009, he is with Asia University as a Chair Professor in the Department of Applied Informatics and Multimedia, and is currently a Vice President there. From 1988 to 1991, he was the Director of the Computer Center at NCTU. From 1991 to 1992 and 1996 to 1998, he acted as the Chairman of Department of Computer and Information Science at NCTU. From 1992 to 1996, he was the Director of the Computer Center at the Ministry of Education and the Chairman of Taiwan Academic Network (TANet) Management Committee. In December 1999, he founded Taiwan Network Information Center (TWNIC) and was the Chairman of the board of directors of TWNIC from 2000 to 2005, and from 2009 till now. He was the recipient of the Excellent Achievement Award for Computer Technology made by the Ministry of Transportation and Communications, Taiwan in 2010. His current research interests include data mining, expert systems, computer algorithms and Internet-based applications.