Mitigating DoS Attacks against the DNS with Dynamic TTL Values

Jarmo Mölsä
Networking Laboratory, Helsinki University of Technology
email: jarmo.molsa@hut.fi

***Abstract*— This paper describes and analyzes a new mechanism to mitigate flooding Denial of Service (DoS) attacks against the Domain Name System (DNS). This mechanism is based on increasing the Time To Live (TTL) value of end-host IP addresses (DNS A records) when a name server is being overloaded with DoS attack traffic. This mechanism is most suitable for popular name servers providing authoritative DNS A records with short TTL values. According to the simulation results, both the average delay and the percentage of failed DNS lookups decrease clearly during a flooding DoS attack. For example, increasing the TTL of DNS A records from 10 minutes to 2 hours decreases the average percentage of failed DNS lookups from 16% to less than 3%, when 90% of the DNS requests are lost due to a DoS attack.**

***Index Terms*— Network Security, Denial of Service, Domain Name System, Time To Live.**

I. Introduction

The Domain Name System (DNS) represents an effective target for Denial of Service (DoS) attacks [1]. In a flooding DoS attack [2][3] a continuous flow of valid-looking DNS requests overloads a network link, a router, a firewall, or a name server. As a result, a legitimate DNS request has problems in reaching a name server and getting an answer. By disabling part of the DNS an attacker is able to prevent or delay access to many services in the Internet. Users typically have only the textual name of a server they are trying to connect to. If the DNS is not available for mapping a textual host name to a numerical IP address, the corresponding server cannot be contacted regardless of the availability of this server. A numerical IP address is always required before it is possible to create a connection to a server.

The objective of this paper is to study how flooding DoS attacks against name servers can be mitigated by modifying the Time To Live (TTL) value of IP addresses (DNS A records). This is an important subject due to the necessary role of DNS in accessing services, due to the prevalence of flooding DoS attacks against name servers [4][5][6], and due to the lack of effective defense mechanisms against these attacks.

The scope of this paper is limited to those name servers providing a final DNS A record (IP address) for a DNS lookup. These name servers are typically the responsibility of the owner of the corresponding *zone* ([7], p. 21). A zone is a non-overlapping part of the DNS. In February, 2003, approximately 68% of the zones in the *com.*-domain were found to be misconfigured [8]. Thus, the level of expertise in operating these name servers is not always high, and a flooding DoS attack against them can easily be successful. Root and Top Level Domain (TLD) name servers, on the other hand, have proved to be very resistant against flooding DoS attacks [9] due to required overprovisioning [10], so they are not considered in this paper.

The main contribution of this paper is to analyze how the total DNS lookup delay and the percentage of failed DNS lookups change when the TTL value of a DNS A record is modified during a flooding DoS attack. The research methodology is based on simulations with the ns-2 network simulator. As another contribution this paper suggests the *dynamic TTL mechanism* to mitigate flooding DoS attacks against name servers of a zone. This mechanism is based on increasing the TTL value of a DNS A record during a flooding DoS attack. The main goal is to increase the cache hit rate at local name servers which reduces the amount of legitimate DNS requests at an overloaded name server. Simulation results show that the mechanism is able to reduce both the average delay associated with the DNS lookup and the amount of completely failed DNS lookups.

At the moment there are no effective defense mechanisms that an organization could use to mitigate flooding DoS attacks against its name servers. Name servers can prevent DNS queries from specific source addresses, but for public services this kind of prevention is not possible. Ingress and egress filtering have been suggested for mitigating flooding DoS attacks using spoofed source IP addresses [4], but these defense mechanisms require extensive deployment in the Internet.

The effect of the length of the TTL on the DNS performance has been studied in [11], in which it was found that the client latency is not as dependent on the use of long TTL values for DNS A records as is commonly believed.

DNS-based load balancing [11][12] and DoS attack resistance require opposite kind of changes to TTL values. This problem manifests itself only during DoS attacks when accurate load balancing must be traded off for the increased availability of a service.

The rest of this paper is structured in the following way. First this paper gives an overview about the operation of the DNS. Then, the dynamic TTL mechanism is described. The next section describes the simulator used to validate the idea. After that the simulation results are shown and explained. The final section concludes the paper.

II. An Overview about the DNS

The Domain Name System (DNS) is used to map domain names in the domain name space into resource records [13][14]. A typical example is to map a textual host name (domain name) into a numerical IP address (type A record).

The *domain name space* is represented as a tree where every node is associated with a *label*. The domain name tree has one leaf node for every end-host accessible from the public Internet (one leaf per end-host name). The internal nodes of the domain name tree reflect hierarchical network domains managed by different organizations. A *domain name* of a node of the tree is written as a sequence of labels from this node towards the root of the tree (e.g. *www.e-service.com.*). If a domain name ends with a dot, it is called a *Fully Qualified Domain Name* (FQDN) which is an absolute and unambiguous name. The last dot in an FQDN marks the root node, i.e. the root node has a null-label.

The DNS is implemented as a distributed data base, where different *name servers* are *authoritative* (responsible) for different non-overlapping *zones* (parts) of the domain name tree. For reliability and performance reasons there are several redundant name servers providing information about a zone.

In the DNS each domain name can be associated with different types of information called *resource records*. From this paper's point of view the most important resource record types are *host address records* (A records) and *authoritative name server records* (NS records). An A record contains a numerical IP address, and an NS record contains a reference to another name server having more detailed information about a domain name to be mapped.

The whole process of translating a domain name into a resource record is called a *name resolution* or a *DNS lookup*. A DNS lookup may require several DNS *requests* and DNS *responses* to be sent. The response messages are either *referrals* or *answers*. A referral contains a list of those name servers having more accurate information. An answer contains the final information requested by an end-host. Naturally a DNS response may indicate an error, like for a non-existent domain name. The result of a complete DNS lookup can be a success (the requested resource record returned), a failure (the requested information not found), or a timeout (no answer within a specified time).

Any name server may cache all received resource records for a period of time defined by the *Time To Live* (TTL) field of a resource record. The TTL is expressed as seconds. The use of caches enhances the performance of DNS.

Without caches every DNS lookup must involve a DNS request to one of the 13 *root name servers* (identified by letters *A* to *M*). In case of a target domain name like *www.e-commerce.com.*, a root name server would return referrals to the *generic Top Level Domain* (gTLD) name servers (identified by letters *A* to *M*) providing authoritative information about the *com.*-domain. A gTLD name server would in turn return referrals to the name servers of the *e-commerce.com.*-subdomain. One of these name servers will then provide the final answer. All these three DNS responses can be stored in caches.

The NS records for the root of the domain name space (IP addresses of the root name servers) are permanently configured into each name server. This guarantees that every name server knows the root of the domain name tree.

A name server can be *iterative* or *recursive*. An iterative name server never sends further DNS requests to other name servers. It simply responds with a referral if it does not know the final answer. A recursive name server will always return the final answer. A recursive name server must typically send DNS requests to several name servers to gradually get closer to the final authoritative answer.

Each end-host must have a *resolver* to access information in the DNS. A resolver is typically implemented as a stub resolver which is simply configured with the IP addresses of the *the local name servers*. Local name servers are recursive and use caches to enhance the DNS performance.

A simple DNS scenario is shown in Fig. 1, which includes a resolver, a set of local name servers, a set of root name servers, a set of gTLD name servers, and a set of name servers in the subdomain of the WWW server. Only local name servers are recursive. All other name servers are expected to be iterative. In this example scenario it does not matter, how many servers there are in any specific name server set. Messages are sent to and handled by one name server in the set. The exact name server can be selected randomly, according to the shortest Round-Trip Time (RTT), etc.

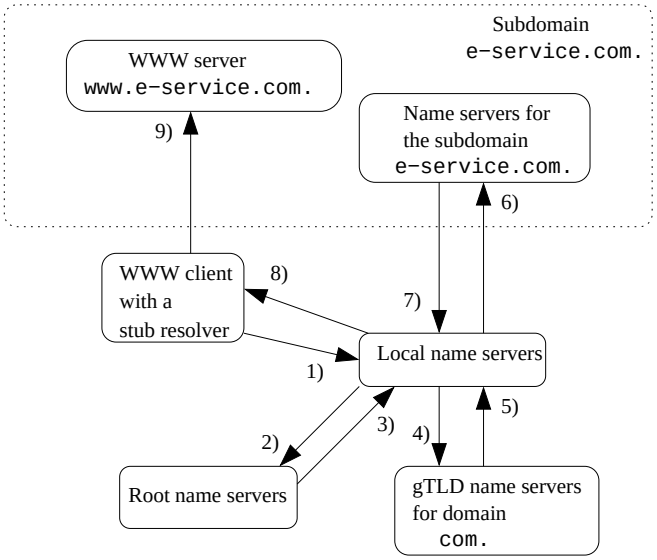

Fig. 1. An example of a DNS lookup.

In the example scenario of Fig. 1 a WWW client wants to connect to the WWW server *www.e-service.com.* The client does not know the IP address of this WWW server so it must use the resolver to send a DNS request to a local name server (message number 1).

The local name server tries to find the following resource records in the following order from the local cache ([13], p. 34): an A record for the *www.e-service.com.*-host, NS records for *e-service.com.*-subdomain, NS records for *com.*-domain and finally the NS records for the root name servers. The first record found defines how the local name server continues.

Either it returns the answer immediately or contacts the closest name server known. The NS records for the root name servers are guaranteed to be found from any cache due to permanent caching of these records from a pre-defined configuration file (hint file).

In this example scenario it is expected that initially the cache contains only the NS records for the root name servers. The local name server must query a root name server (message numbers 2 and 3), a gTLD name server (message numbers 4 and 5) and finally a name server in the subdomain of the WWW server (messages 6 and 7). Messages 7 and 8 include the A record for the WWW server. All DNS responses are cached by the local name server. At the end the WWW client can contact the WWW server by using the numerical IP address in the received DNS answer (message 9).

It should be noted that caching of DNS information is done in other places also. For example, browsers and Java have their own caches. The effect of these kind of separate caches is not included in this paper.

III. The Dynamic TTL Mechanism

This section describes the dynamic TTL mechanism which mitigates flooding DoS attacks against name servers. As a reaction mechanism [15] it is used after a DoS attack is detected manually or automatically (e.g. by inspecting log files or by measuring DNS performance from a remote site). The detection mechanism, however, is not the subject of this paper.

The dynamic TTL mechanism is based on using two different TTL values for each A record: a lower value for normal operation (*default TTL*) and a higher value during a detected DoS attack (*TTL during attack*).

A longer TTL value makes it possible to have higher cache hit rates at remote name servers. When the IP address of a destination host is found from the cache, the overloaded name servers need not be contacted. If the attack is targeted only at the name servers, the final destination can be contacted without problems.

The dynamic TTL mechanism is supposed to be used for A records. NS records are fairly static, and they have a much longer average TTL value (up to several days) than A records.

A major benefit of the dynamic TTL mechanism is that it is easy to implement and does not depend on any third parties. Only local operations are required to mitigate an attack and increase the availability of local services to the public Internet.

IV. The DNS Simulator

The effect of dynamic TTL values in mitigating DoS attacks against name servers was simulated with an OTcl program under the ns-2 network simulator. The setup of client groups, WWW servers, and name servers is shown in Fig. 2. The simulator implements all the basic DNS functions as described earlier in this paper.

A. Clients, WWW Servers and Name Servers

In the simulator there are 200 independent groups of clients. Each client group would reflect, for example, the users of an organization or the customers of a small Internet Service Provider (ISP). Each client group is expected to have one local caching name server. The exact nature of the arrival process of DNS requests at a local name server is not known. Here it is expected that each client group is initiating DNS lookups to the local name server with an exponentially distributed inter-arrival time. Two different average values for the exponentially distributed inter-arrival times were used in the simulations: 120 and 7200 seconds. This makes it possible to study the approximate effect of inter-arrival time on the usefulness of the dynamic TTL mechanism.

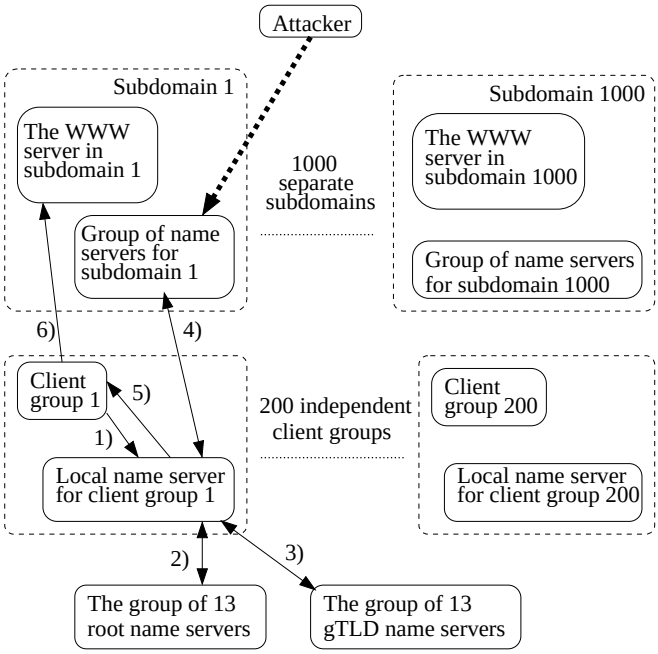

Fig. 2. The simulator setup. 200 independent client groups initiate DNS lookups for 1000 different WWW servers each in a separate subdomain. The WWW servers are ordered according to their Zipf-like popularity, the WWW server in subdomain 1 being the most popular. A flooding DoS attack is targeted against the name servers of subdomain 1 which is using the dynamic TTL mechanism.

The client groups are trying to resolve the name of a server, which is here expected to provide WWW services in a subdomain under the *com.*- or *net.*-domains. The number of WWW servers is 1000, each located in a different subdomain. There are thus 1000 different server subdomains with their own name servers. WWW servers and their corresponding subdomains are identified by their number from 1 to 1000. Each client group chooses the number of the destination WWW server from a Zipf-like distribution, with the parameter α = 0.8. In one study the distribution of web requests was found to be Zipf-like with the value of α being approximately 0.8 [16]. This kind of a distribution means that the WWW server in subdomain 1 is the most popular site, and the WWW server in subdomain 1000 is the least popular site [17].

Each client contains a stub resolver (see [7], p. 26), which is configured to always contact a single local name server. There is one local recursive name server for every client group. Every local name server saves any received resource record in its cache for the duration of the TTL.

Each WWW server subdomain is expected to have a set

of four name servers providing authoritative resource records for the subdomain. The number of these name servers has an effect on the retransmission schedule. Namely, a local name server retransmits in a round-robin fashion to every member of a name server group with an initial timeout length of two seconds ([18], and [7], p. 109).

B. TTL Values

In January, 2004, all root name servers used a TTL value of 172800 seconds (2 days) for the NS records of gTLD name servers. At the same time the gTLD name servers also used a TTL value of 172800 seconds (2 days) for the NS records of subdomain name servers. These TTL values were used in the simulator.

The IP addresses of WWW servers (the final end hosts) use a default TTL of 600 seconds (10 minutes). In one study it was found that the median TTL of A records was approximately 15 minutes when measured from a TTL distribution weighted by access counts [11]. Due to the tendency to use shorter TTL values for A records, the simulator uses 600 seconds as the TTL value for A records.

The dynamic TTL mechanism uses another higher TTL value during DoS attacks. A TTL value of 7200 seconds (2 hours) was chosen for this purpose. This temporary TTL should be in the order of the expected attack length.

C. Delay Distribution for Request-Response Times

The delay between the transmission of a DNS request and the reception of the corresponding response (request-response time) is expected to be normally distributed with a mean value of 92 milliseconds and a standard deviation of 15 milliseconds. In one study [19] it was found that generally no single distribution appears to give a consistently good fit to measured delays of real DNS traffic. A normal distribution with the above mentioned parameter values was found to match reasonably well the request-response times of the L root name server during one measurement period.

The simulator does not take into account the transmission delays between the resolver and the local name server.

D. Flooding DoS Attack

The victims of the DoS attack are the four name servers of the most popular WWW server subdomain having the number 1 in the Zipf-like distribution. The attacker is expected to flood all these name servers with excess traffic, like DNS, ICMP, UDP, or TCP SYN messages. All four name servers of the victim subdomain 1 are attacked in the same way. All remaining name servers (name servers for subdomains 2–1000, root name servers, gTLD name servers) experience no packet-loss and respond always to every DNS request.

The DoS *attack intensity* is defined to be the percentage of lost incoming DNS requests due to the excessive load. For example, only 10% of the DNS requests will be responded, if the attack intensity is 90%.

Random packet-loss typically found in networks is not included in the simulator.

E. Retransmission of Lost DNS Requests

During a flooding DoS attack, only the name servers for subdomain 1 will experience packet-loss. The simulator software includes support for the retransmission of lost DNS requests both at resolvers and local name servers.

The DNS resolver in a client is expected to have a retransmission mechanism similar to the Berkeley Internet Name Domain (BIND) resolver, version 8.2.1 or later ([7], p. 110). The resolver will retransmit only once after a timeout of 5 seconds. If the retransmission is not answered within 10 seconds, the resolver will return an error to the calling software. A resolver will thus spend a maximum of 15 seconds for a DNS lookup.

Local name servers in the simulator have a similar retransmission mechanism as in BIND version 9.2.3 with the following exceptions: Round-Trip Time (RTT) is not calculated and the set of redundant name servers are cycled through only twice. The timeout length is always 2 seconds during these two cycles. A local name server will cease retransmitting after 7 trials.

V. Results of the Simulations

The goal of the simulations is to see how the DNS performance depends on the TTL value of DNS A records during a flooding DoS attack. The simulator provides information about the delay of successful DNS lookups and the proportion of completely failed DNS lookups.

The relevant parameter values in the simulations are the following:

- the length of one simulation is 1 000 000 seconds,
- the DoS attacks starts at the time of 300 000 seconds,
- the attack is carried out at the intensity of 90% (some tests also with the intensity of 50%),
- the default TTL value is 600 seconds (10 minutes),
- the TTL value during a detected DoS attack is 7200 seconds (2 hours), and
- the average time (from exponential distribution) between consecutive DNS lookups from a single client group is either 120 or 7200 seconds (*inter-arrival time*).

All simulation results are calculated from the DNS traffic from any client group (1–200) to subdomain 1, because only subdomain 1 is the target for a DoS attack.

In the simulations it is expected that the DoS attack is detected at the same time when the attack begins (detection is not the subject of this paper). In practice, however, there is always some delay associated with the attack detection. The dynamic TTL mechanism cannot increase the DNS performance until a DoS attack against the name servers is detected and the new TTL values have reached the local name servers.

A. The Delay of Successful DNS Lookups

The average DNS lookup delay is shown in Fig. 3. The X-axis indicates the time in seconds when a DNS lookup was initiated. The Y-axis indicates the delay of DNS lookups averaged over 10 000 second intervals. This figure shows the average delay when the dynamic TTL mechanism is used to

protect the subdomain 1 (*TTL = 600/7200)* and when this mechanism is not used (*TTL = 600*). The results are shown for inter-arrival values of 7200 seconds (thick lines) and 120 seconds (thin lines).

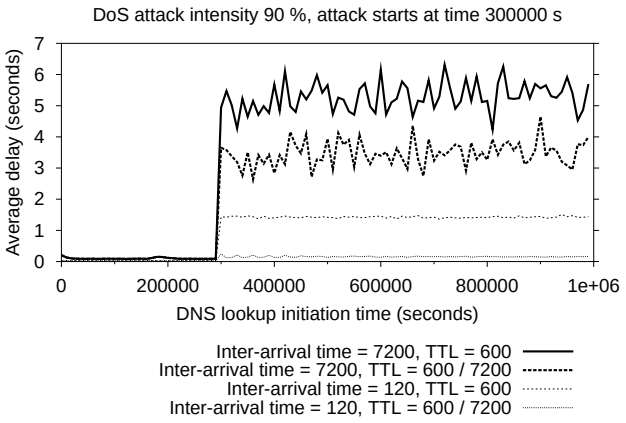

Fig. 3. The delay of successful DNS lookups averaged over 10 000 second intervals.

If the inter-arrival time is 120 seconds, the average delay is reduced almost 90% from 1.5 seconds to approximately 0.16 seconds. When the inter-arrival time is 7200 seconds, the average DNS lookup delay is much longer than when inter-arrival time is 120 seconds. The reason for this is very logical, because the more DNS lookups there are in a time unit, the more lookups will result in a cache hit at the local name server. This pulls down the average lookup delay.

The positive effect of the dynamic TTL mechanism is visible with both inter-arrival times. This effect is, however, stronger when inter-arrival time is shorter. The shorter the inter-arrival time is, the more cache hits will result at the local name server. If a subdomain is visited very seldom, the cached A record will time out before the next visit to it.

As expected, the dynamic TTL mechanism provides the best benefit for those client groups which have many clients referencing a similar set of popular destinations. This increases the possibility for a cache hit at the local name server.

Averaging DNS lookup delays over 10 000 second intervals hides some details of shorter time scale. For this reason the time range from 290 000 seconds to 350 000 seconds of Fig. 3 is magnified in Fig. 4 which shows the delay of DNS lookups averaged over 200 second intervals. Only results for the inter-arrival time of 120 seconds are shown in this figure. Figure 4 shows well that all client groups are practically synchronized due to the short TTL value (600 seconds) after the DoS attack starts at the time of 300 000 seconds. The client groups gradually desynchronize due to the randomness in both the DNS lookup initiation process and the request-response times. Figure 4 also shows the delay until the dynamic TTL mechanism begins to enhance the DNS performance after the attack is detected (the high peak at the time of 300 000 seconds).

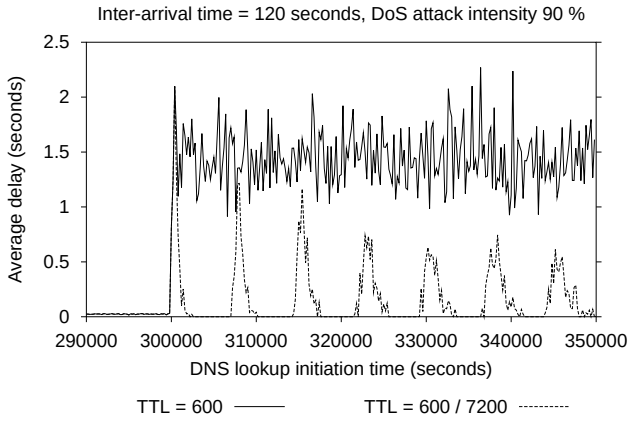

Fig. 4. The delay of successful DNS lookups averaged over 200 second intervals during the time range from 290 000 to 350 000 seconds.

B. The Percentage of Failed DNS Lookups

The average percentage of failed DNS lookups is shown in Fig. 5. The X-axis indicates the time in seconds when a DNS lookup was initiated. The Y-axis indicates the percentage of failed DNS lookups averaged over 10 000 second intervals. A DNS lookup fails if it times out completely at a resolver without an answer. This figure shows the average percentage of failed DNS lookups when the dynamic TTL mechanism is used to protect the subdomain 1 (*TTL = 600/7200)* and when this mechanism is not used (*TTL = 600*). The results are shown for inter-arrival values of 7200 seconds (thick lines) and 120 seconds (thin lines).

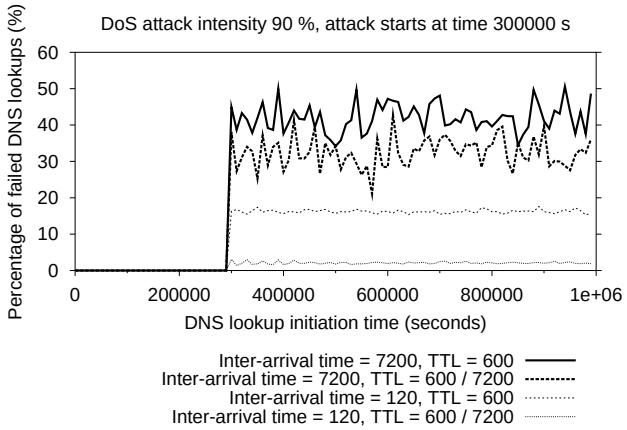

Fig. 5. The percentage of failed DNS lookups averaged over 10 000 second intervals.

The positive effect of the dynamic TTL mechanism is visible with both inter-arrival times. The shorter inter-arrival time (120 seconds) results in more cache hits at the local name server, which decreases the need to send any DNS requests to the overloaded name servers. This reduces the average DNS lookup failure percentage. Increasing the TTL value of A records from 10 minutes to 2 hours decreases the average percentage of failed DNS lookups from 16% to less than 3%, when 90% of the DNS requests are lost due to a DoS attack (inter-arrival time being 120 seconds).

C. Cumulative Distribution Functions for the DNS Lookup Delay

The Cumulative Distribution Functions (CDF) were calculated for several cases with different parameter combinations. The CDF is defined as follows: $CDF(X) = Probability(delay \leq X)$. These CDFs are shown in Fig. 6, where the inter-arrival time is 7200 seconds. Only successful DNS lookups are included here. Because a resolver will timeout completely after 15 seconds, the delay for all successful DNS lookups is less than 15 seconds.

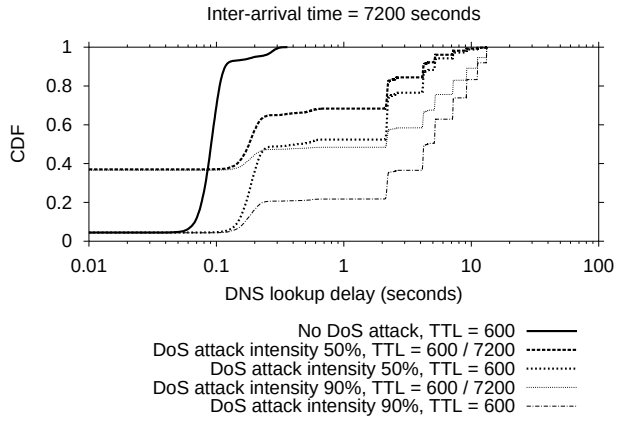

Fig. 6. The Cumulative Distribution Functions (CDF) for the DNS lookup delay, when the inter-arrival time of DNS requests at every local DNS server is 7200 seconds.

As can be seen from the Fig. 6 the dynamic TTL mechanism results in a better CDF, i.e. the mechanism increases the probability of low delays, and decreases the probability of longer delays.

The DNS retransmission policy is visible in these curves. The local name server will retransmit at times of 2 and 4 seconds. At the time of 5 seconds the resolver will timeout and retransmit. After that the local name server will again retransmit with a 2 second interval until the DNS lookup completely times out at the resolver at the time of 15 seconds.

D. The Effect of the Dynamic TTL Mechanism on the DNS Performance

The performance of the DNS during a flooding DoS attack depends on the TTL value. The longer the TTL value, the better the performance. The DNS performance as the function of the TTL value during an attack is shown in Fig. 7. The default TTL is 10 seconds when no DoS attack is present. The thick lines indicate the average delay of successful DNS lookups as a function of the TTL (left Y-axis). The thin lines indicate the percentage of failed DNS lookups as a function of the TTL (right Y-axis). Inter-arrival times of 120 and 7200 seconds were used in these simulations. The DoS attack intensity was 90%.

As can be seen from the Fig. 7 the shorter the inter-arrival time, the higher the performance gain from increasing the TTL. When the inter-arrival time at client groups is 120 seconds, a 50 second TTL (default TTL multiplied by 5) will increase the performance by approximately 10% at the most popular destination domain, and a 400 second TTL (default TTL multiplied by 40) will increase the performance by approximately 50%.

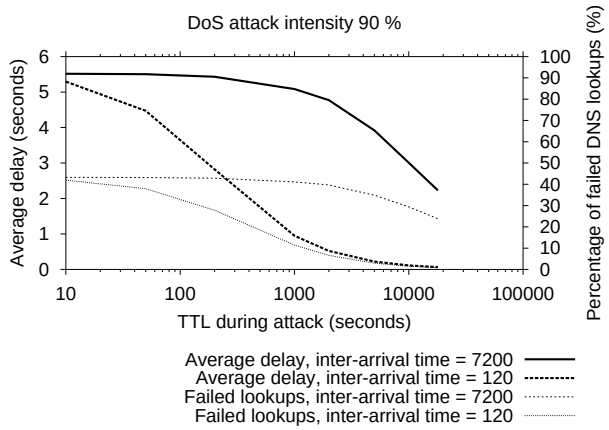

Fig. 7. The effect of the dynamic TTL mechanism on DNS performance. Thick lines indicate average delay of successful DNS lookups (left Y-axis). Thin lines indicate percentage of failed DNS lookups (right Y-axis). Attack intensity is 90%.

DNS-based load balancing depends on a small TTL value for A records. Even though the dynamic TTL mechanism requires relatively long TTL values during an attack, this mechanism can increase the availability of load balanced services. Without any TTL modification many requests for a popular service would fail at the DNS lookup phase, and even a perfect load balancing has no possibility for increasing the DNS performance. The dynamic TTL mechanism will increase the availability at the price of less effective load balancing. This should be seen as a good trade-off. In IPv6 networks one possibility to solve this problem with DNS-based load balancing (application-level anycasting) is to use network-level anycasting [20].

VI. Conclusion

The DNS is a necessary prerequisite for accessing practically any service in the Internet. This makes the DNS an attractive target for attackers who can, for example, disable part of the DNS by flooding a set of name servers with valid-looking but unnecessary DNS requests.

At the moment there are no effective mechanisms to mitigate flooding DoS attacks against name servers providing DNS A records. Existing misconfigurations in most of these name servers make them even more attractive for attackers. Root and gTLD name servers, on the other hand, have proved to be very resistant against these kind of attacks due to required overprovisioning and good expertise. New defense mechanisms are thus required especially for name servers providing DNS A records, i.e. final answers to DNS lookups.

This paper described how dynamic TTL values can be used to mitigate flooding DoS attacks against name servers providing DNS A records. When the name servers of a DNS zone are flooded with unnecessary traffic, increasing the TTL of address resource records increases the cache hit rate at legitimate clients and reduces the amount of DNS traffic against overloaded name servers.

The simulation results clearly show the benefits of this new simple mechanism. For example, when the inter-arrival time of DNS requests is 120 seconds and the attack intensity is 90%, increasing the TTL from 600 seconds to 7200 seconds during an attack reduces the average DNS lookup delay by 90% from approximately 1.5 seconds to 0.16 seconds. The average percentage of failed DNS lookups was also reduced approximately from 16% down to less than 3%. According to the simulation results the modification of the TTL of A records is a useful mechanism for mitigating flooding DoS attacks against the DNS.

REFERENCES

[1] A. Chakrabarti and G. Manimaran, "Internet infrastructure security: A taxonomy," *IEEE Network*, vol. 16, no. 6, pp. 13–21, 2002.

[2] R. K. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, Oct. 2002.

[3] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 3, July 2001.

[4] CERT Coordination Center, "CERT incident note IN-2000-04, denial of service attacks using nameservers," Apr. 2000.

[5] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," in *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., Aug. 2001.

[6] N. Brownlee, K. C. Claffy, and E. Nemeth, "DNS measurements at a root server," in *Proceedings of the IEEE GlobeCom*, San Antonio, USA, Nov. 2001.

[7] P. Albitz and C. Liu, *DNS and BIND*, 4th ed. Sebastopol, California, USA: O'Reilly & Associates, Inc., Apr. 2001.

[8] Men&Mice, "Domain health survey for .COM," Feb. 2003. [Online]. Available: http://www.menandmice.com/dnsplace/healthsurvey.html

[9] P. Vixie, G. Sneeringer, and M. Schleifer, "Events of 21-Oct-2002," ISC/UMD/Cogent, Tech. Rep., Nov. 2002.

[10] R. Bush, D. Karrenberg, M. Kosters, and R. Plzak, "Root name server operational requirements," Internet Engineering Task Force, Request for Comments RFC 2870, June 2000.

[11] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS performance and the effectiveness of caching," *IEEE/ACM Trans. Networking*, vol. 10, no. 5, pp. 589–603, Oct. 2002.

[12] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," in *Proceedings of the IEEE INFOCOM*, Anchorage, USA, Apr. 2001.

[13] P. Mockapetris, "Domain names - concepts and facilities," Internet Engineering Task Force, Request for Comments RFC 1034, Nov. 1987.

[14] ——, "Domain names - implementation and specification," Internet Engineering Task Force, Request for Comments RFC 1035, Nov. 1987.

[15] A. Householder, A. Manion, L. Pesante, G. M. Weaver, and R. Thomas, *Managing the Threat of Denial-of-Service Attacks*. CERT Coordination Center, Oct. 2001.

[16] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, USA, Mar. 1999.

[17] L. A. Adamic, "Zipf, Power-laws, and Pareto - a ranking tutorial," Xerox Palo Alto Research Center, Tech. Rep., 2000. [Online]. Available: http://ginger.hpl.hp.com/shl/papers/ranking

[18] A. Kumar, J. Postel, C. Neuman, P. Danzig, and S. Miller, "Common DNS implementation errors and suggested fixes," Internet Engineering Task Force, Request for Comments RFC 1536, Oct. 1993.

[19] N. Brownlee and I. Ziedins, "Response time distributions for global name servers," in *Proceedings of the Passive & Active Measurement Workshop*, Fort Collins, Colorado, USA, Mar. 2002.

[20] S. Weber and L. Cheng, "A survey of anycast in IPv6 networks," *IEEE Commun. Mag.*, vol. 42, no. 1, pp. 127–132, Jan. 2004.