ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2012. No. 1(117)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 180 TELECOMMUNICATIONS ENGINEERING
TELEKOMUNIKACIJŲ INŽINERIJA

Adaptive TTL based Approach to Balance DNS Server Load

T. Mackus, T. Simonaitis, D. Tamulioniene
Information Technology Development Institute, Kaunas University of Technology, Studentų str. 48A, LT-51367 Kaunas, Lithuania, phone: +370 37 300606, e-mail: tomas@ktu.lt

crossref http://dx.doi.org/10.5755/j01.eee.117.1.1058

Introduction

Everyday internet users use Domain Name System (DNS) [1] to access internet resources using host names (domain names). As all internet resources can be accessed only by using Internet Protocol (IP) addresses which are hard to remember by humans, Domain Name System enables usage of easy to remember internet address names (domain names). When people want to reach internet resources, Domain Name system transparently translates entered internet address name to IP address before these resources can be reached.

The Domain Name System (DNS) uses caching mechanisms to reduce latency of the domain name resolution process and to minimize wide area network (WAN) traffic at the same time. DNS caching is deployed using policy, which states, that received DNS resource record (RR) is valid for specific period of time, called “time-to-live” (TTL). Every RR has his TTL value in Domain Name System. Administrator of domain name sets TTL for the resource records of domain he is administering.

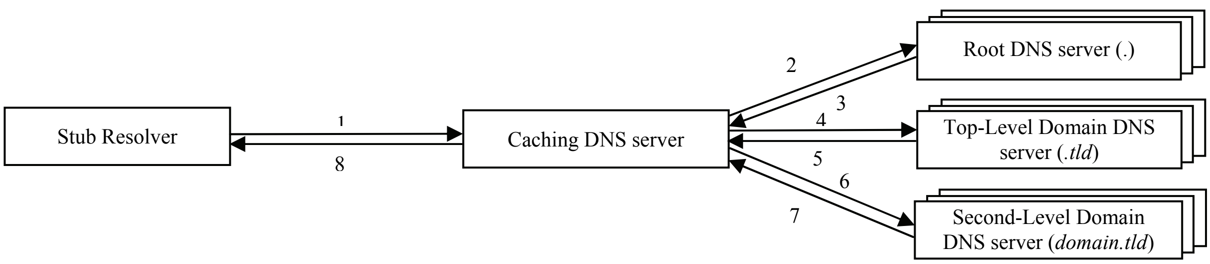

Fig. 1. Domain name resolution scheme

Typical host name resolution scheme (Fig. 1) using caching DNS server is presented. When client’s computer application needs access to internet resources, it uses clients computers stub resolver, which is usually configured to use caching DNS server to resolve host name to IP address. Example of domain name www.domain.tld DNS resolution, when caching DNS server has no cached records in its memory:

1. Stub resolver queries caching DNS server for IP address of www.domain.tld;
2. Caching DNS server queries one of Root servers for IP address of www.domain.tld;
3. Root server responds to caching DNS server with the list of Top-Level Domain .tld authoritative DNS servers. Answer is stored in caching DNS server memory (cache) for time period specified by TTL value;
4. Caching DNS server queries one of Top-Level Domain .tld authoritative DNS server for IP address of www.domain.tld;
5. Top-Level Domain .tld authoritative DNS server responds to caching DNS server with the list of domain.tld authoritative DNS servers. Answer is stored in caching DNS server cache for time period specified by TTL value;
6. Caching DNS server queries one of domain.tld authoritative DNS server for IP address of www.domain.tld;
7. Authoritative domain.tld DNS server responds to caching DNS server with IP address of www.domain.tld. Answer is stored in caching DNS server cache for time period specified by the TTL value;
8. Caching DNS server sends response to stub resolver with IP address of www.domain.tld.

If Caching DNS server receives DNS query to resolve host name mail.domain.tld when caching DNS server has cached information after resolving www.domain.tld address, it will send query directly to domain.tld authoritative DNS server, which will provide answer about mail.domain.tld IP address to caching DNS server. This answer will be sent to stub resolver.

The biggest impact of DNS caching is for the root DNS servers, as all DNS queries will use Root DNS servers in hostname resolution process. DNS servers authoritative for lower lever DNS hierarchy domain names have smaller impact of DNS caching mechanisms compared to higher level domain names authoritative DNS servers.

Query rate received by DNS server is influenced by TTL value for the resource records of domain. The smaller TTL values are used, the more often queries are sent to authoritative DNS server, the more coherent information of domain name is. Used bigger TTLs reduces DNS query traffic to DNS server, but after domain name resource record information is updated it takes more time (up to TTL value in seconds) to propagate new information across the network. Typical recommended TTL value for resource records is from 1 to 5 days [2]. But in nowadays, when mobile networking [3] which uses dynamic DNS and low TTL's is in wide use and content distribution networks uses DNS for load balancing, TTL values more often are kept low.

In our work we will concentrate on DNS servers, serving lower DNS hierarchy level domain names, with focus on regular Internet Service provider's authoritative DNS servers which mainly holds second and lower level domain names information.

Because query rate DNS server receives can differ, depending on time of the day, day of the week or even time of the year, regular tracking of DNS query rate is needed to keep DNS server and internet traffic resources on the similar desired level. Our idea is to make real world DNS server query rate analysis and employ adaptive TTL mechanisms to keep DNS query rate on the desired level.

Related work

J. Jung, E. Sit, H. Balakrishnan, and R. Morris made analysis of performance and behavior of DNS and effectiveness of caching from client side perspective [4]. Author's one statement was that low TTL A record binding should not degrade DNS performance. But our work main focus is on performance of caching from the DNS server point of view.

Hitesh Balani and Paul Francis suggest changing caching server behavior on DNS TTL values in some cases [5]. This can impact on authoritative DNS server query load behavior in an unforeseen way.

Proposed renewal policies [6] helps reduce DNS resolution time for end-user, but requires caching DNS server to "renew" DNS records stored in cache by sending premature DNS queries, before stored in cache records are stale. Widely adopted renewal policy generates predictable and constant DNS query rate to the authoritative DNS servers.

Previous works on TTL-based internet cache modeling [7] concluded explained empirical finding of Jung et al. [4] that for DNS accesses, the cache hit rate rapidly increases as a function of TTL, quickly reaches over 80% for the 15 minute TTL, leaving room for only a modest increase in the hit rate for larger TTLs.

Idea to use Adaptive TTL is not new. There was the idea to use adaptive cache TTL adjustment to make consistent records in DNS cache after resource records update in authoritative DNS server [8]. According to the simulation results [9] the modification of the TTL of A records is a useful mechanism for mitigating flooding DoS attack against the DNS. But simulation was focused on authoritative DNS server responsiveness from the caching DNS server perspective. Adaptive-TTL approach was proposed to use for web server load balancing [10], but proposed approach does not take attention to DNS server load and examines web resources access performance.

In our work we will focus on authoritative DNS server, who has limited resources and needs to be kept on the desired DNS query load level.

Data collection and analysis

For our research we used 3 DNS servers, which were authoritative for the same 5038 domains. First of all we turned off DNS caching mechanism on all 3 DNS servers by setting TTL values on all resource records to 0. Negative caching TTL value was set to 0 also. Negative caching means time, how long answer about non existing DNS resource record (RR) should be stored by caching resolver before sending the same DNS query again.

When resource records with bigger TTL in DNS cache servers expired, we captured all DNS queries that arrived to these 3 DNS servers for a one week period. Collected data show, that all 3 DNS servers had very similar DNS query rates. This is predictable, as DNS resolver query sending behavior in most cases is defined to be random-cyclic (round-robin - a random selection of the initial order thereafter cyclic) and all 3 DNS servers are in the same network distance from clients.

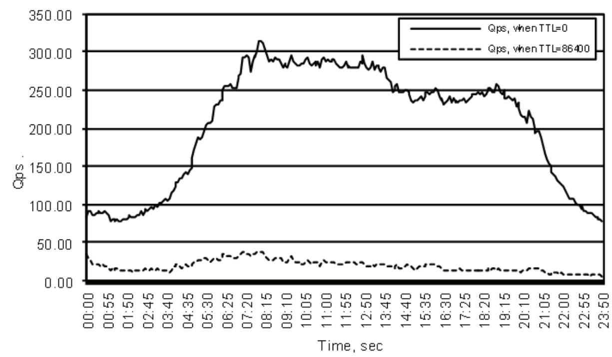

Fig. 2. 24 hour DNS query rate graph, with different TTL

Also we noticed, that query rate behavior on these DNS servers was related to the time of the day. Lowest query rate was at the night time, and highest – at the day time. We suggested that this query rate behavior on time of the day is because of clients geographical distribution in the same or similar time zone. Slight decrease in DNSquery rate on DNS servers was on weekend. This let us presume, that part of DNS queries was generated by live internet users, who use internet on weekends less, than on working days. Overall DNS query rate behavior from Monday to Friday was the same.

As DNS caching mechanism is effective only for the same queries from the same source address within specific period of time, we counted queries for 24 hour time slot, arriving from the same source IP address and for the same resource record and type. Query data from all 3 DNS servers was merged into one data set, which was analyzed. Our measurements showed, that only 5.6% of these queries were sent only once during 24 hour period of time. Average total query rate was 207 qps, with highest query rate of 314 qps and lowest query rate of 77 qps (Fig. 2).

After we calculated DNS query rate, that could be in case all DNS query sending resolvers would be well behaved and use given RR TTL values for caching properly. We counted DNS query rate if value for the TTL records would be 86400 (24 hours) (Fig. 2), i.e. if once received, same DNS query from the same IP address would not arrive on the same day again. Calculated average query rate in this case was 19 qps, or almost 11 times less than query rate without caching. By calculating this we found, that only 9.2% of all queries during this day were unique. 91.8% of received queries were the same and from the same source IP address, as already received during this day before. This let us state, that caching should be quite effective mechanism to regulate authoritative DNS servers load by regulating DNS query rate via regulating DNS resource records TTL value.

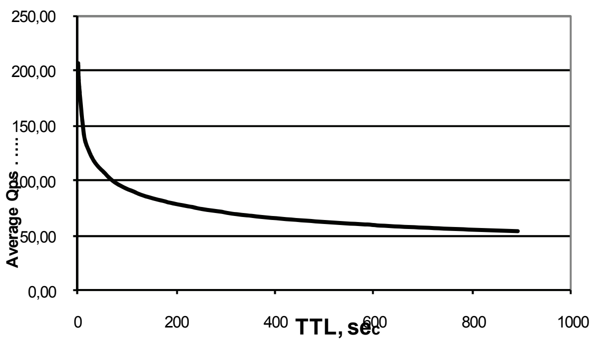

Fig. 3. Average query rate relation to the TTL value

Taking into account, that most effective changes in cache hit rate (for up to 80%) can be achieved by increasing TTL up to 900 [5], we counted query rate with the same query data, but different TTL, varying from 1 till 900 seconds. On presented chart (Fig. 3.) we can see average DNS query rate per second in relation to the DNS resource record caching TTL value.

Our calculations show, that by turning on DNS caching mechanism for TTL of 1 day (86400), we can reduce average DNS query rate by 91.8%. But it is enough to change TTL value to 15 minute (900), to reduce average query rate by 74.4%.

To confirm our suggestion, that most of DNS query sending clients resides in the same time zone, we made analysis of all IP addresses from where DNS queries were sent. Our analysis shows that 89% of all received queries were sent from Europe region. This clearly explains why DNS query rate dramatically increases at 7 a.m. and starts to decrease significantly at 7 p.m. in UTC (Universal Coordinated Time).

Adaptive TTL adjustment

As authoritative DNS servers are the source of authoritative DNS records and TTL values associated with them, DNS resource records TTL value should be adjusted on these servers. Because of DNS server software diversity, DNS TTL value adjusting solution should be independent from the DNS server software.

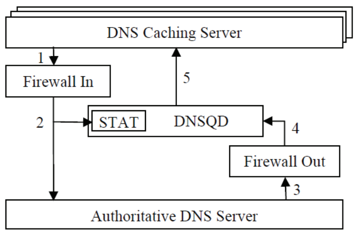

Fig. 4. Adaptive TTL adjustment program DNSQD scheme

We created program, called DNSQD. Our created program is independent from DNS server software. It listens and collects information about incoming DNS query packets to make it possible calculate incoming DNS query rate to the server. Program analyses outgoing DNS response packets and modifies them according to the policy, which depends on the calculated incoming DNS query rate (Fig. 4).

It was out of the scope in this work to find best policy for DNS server query load balancing. Simple policy was used: if last 10 seconds average qps rate is below the desired level, program reduces by 10 default TTL value in DNS response packets. If last 10 seconds average DNS qps rate is above desired level, DNSQD program increases default TTL value in DNS response packets by 10. If incoming DNS query rate is on desired level, then TTL value in outgoing DNS response packet stays the same.

Experiment

For our experiment we used the same 3 DNS servers, which were authoritative for the same 5038 domains. Desired DNS query rate per second was defined on 30 qps rate on each DNS server, or total rate of 90 queries per second. We took this value to be in between lowest and highest possible query rate as it was observed in our researches (Fig. 2.). TTL value range was defined to vary from 0 to 900, as our research show, that almost 75% of DNS query rate should be reduced by increasing TTL value from 0 till 900.

Results from our real world experiment (Fig. 5) shows that average qps rate on 24 hour period was 92,55. Calculated adaptive TTL query rate per second standard deviation 9,33 qps is significantly smaller than standard deviation 114,89 qps of measured query rate with TTL value 0 and standard deviation 26,54 qps of measured query rate with TTL value 900

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad (1)$$

where σ- standard deviation, N – amount of measured qps values, χ_i – measured qps value, μ – average qps value from the set of measured values $\chi_1 .. \chi_N$.

Average query rate with adaptive TTL was bigger than desired TTL value by 2,55 qps.

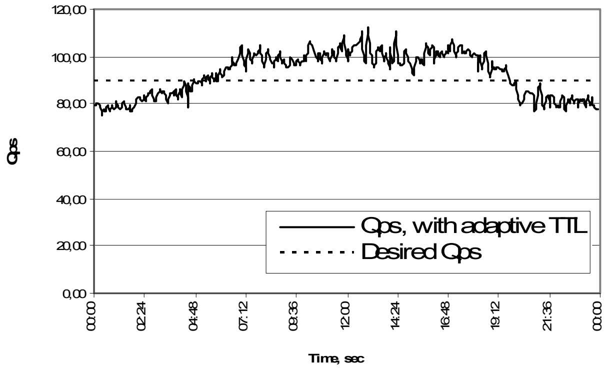

Fig. 5. DNS query rate after adaptive TTL adjustment

We also discovered that real world query rates could be bigger than expected because of the some domain names lame delegation. Query data analysis shows that 9,8% of all received queries were not valid and received DNS response packet with response code "Refused", which means, that these responses were not cached at all.

Conclusions

Our research confirmed our assumption, that DNS resource records TTL value can be used to regulate DNS server query load. But it should be noted, that query rate can be regulated only within determined ranges – lowest range is unique queries per period of time and max range is all DNS queries rate per period of time. It was about 9 % of unique queries in a one day time slot in our scenario, which means that up to 91% of valid incoming queries can be reduced by using DNS caching mechanism, i.e. by increasing DNS RR TTL values.

Another important aspect of this research was finding out that almost 75% of possible to regulate DNS query rate on the server can be regulated by regulating relatively small TTL value from 0 to 900 seconds.

Created independent from DNS software program DNSQD is suitable program to use for DNS response packet TTL value changes, as it increase DNS query response time only for 0.2ms. Compared to regular DNS query resolution time [11] 0.2ms delay is not significant in total DNS query resolution round trip time.

References

1. **Mockapetris P., Dunlap K.** Development of the Domain Name System // in Proc. ACM SIGCOMM, Computer Communication Review. – Stanford, CA, 1988. – Vol. 18. – No. 4. – P. 123–133.
2. **Jung J., Sit E., Balakrishnan H., Morris R.** DNS Performance and the Effectiveness of Caching // IEEE/ACM Transactions on Networking. – IEEE/ACM, 2002. – Vol. 10. – Iss. 5. – P. 589–603.
3. **Barr D.** Common DNS Operational and Configuration Errors. – RFC1912, 1996. Online: http://www.rfc–editor.org/rfc/rfc1912.txt.
4. **Snoeren A. C., Balakrishnan H.** An End–to–End Approach to Host Mobility // in Proceedings of the 6th ACM MOBICOM. – ACM, New York, USA, 2000. – P. 155–166.
5. **Ballani H., Francis P.** Mitigating DNS DoS Attacks // ACM CCS 2008 – ACM, New York, 2008. – P. 189–198.
6. **Cohen E., Kaplan H.** Proactive caching of DNS records: addressing a performance bottleneck // Computer Networks. – Elsevier, 2003. – Vol. 41. – Iss. 6. – P. 707–726
7. **Jung J., Berger A. W., Balakrishnan H.** Modeling TTL–based Internet Caches // Twenty–Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'2003.). – IEEE Societies, 2003. – Vol. 1 – P. 417–426.
8. **Tomohiro I., Sekiya Y.** DNS cache–record coordination Using Adaptive Cache TTL Adjustment // (IJCNS) International Journal of Computer and Network Security. – IJCNS, 2010. – Volume 2, No. 1 – P. 66–69.
9. **Mölsä J.** Mitigating DoS attacks against the DNS with dynamic TTL values // in Proceedings of the Ninth Nordic Workshop on Secure IT Systems. – Espoo, Finland, 2004. – P. 118–124.
10. **Chatterjee D., Tari Z., Zomaya A.** A task–based adaptive TTL approach for Web server load balancing // In Proceedings of 10th IEEE Symposium on Computers and Communications. – IEEE Computer Society, 2005. – P. 877–884.
11. **Mackus T.** Investigation of Network Traffic Generated by DNS Service // in Proceedings of IT 2004 Conference. – Kaunas: Technologija, 2004. – P. 618–625.

Received 2011 06 21
Accepted after revision 2011 11 05

T. Mackus, T. Simonaitis, D. Tamulioniene. Adaptive TTL based Approach to Balance DNS Server Load // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 1(117). – P. 81–84.

This paper presents analysis of Domain Name System (DNS) resource records (RR) "time-to-live" (TTL) values at authoritative DNS server to balance DNS query rate at the desired level. According to the research and experiment results adaptive TTL value can be effectively used to manipulate incoming DNS query rate. Ill. 5, bibl. 11 (in English; abstracts in English and Lithuanian).

T. Mackus, T. Simonaitis, D. Tamulioniene. DNS serverio apkrovimo balansavimas naudojant adaptyviąsias TTL vertes // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 1(117). – P. 81–84.

Straipsnyje analizuojama interneto adresų sričių sistemos (DNS) išteklių įrašų saugojimo verčių (TTL) įtaka pripažintų DNS serverių užklausų apkrovimo reguliavimui. Tyrimo ir eksperimento rezultatai rodo, kad, naudojant adaptyvias TTL reikšmes galima efektyviai reguliuoti į DNS serverį ateinančių užklausų skaičių. Il. 5, bibl. 11 (anglų kalba; santraukos anglų ir lietuvių k.).