PAPER

Cache Effect of Shared DNS Resolver

Kazunori FUJIWARA[†a)], **Akira SATO**[††b)], ***and*** **Kenichi YOSHIDA**[††c)], ***Members***

SUMMARY Recent discussions on increasing the efficiency of the Internet's infrastructure have centered on removing the shared Domain Name System (DNS) resolver and using a local resolver instead. In terms of the cache mechanism, this would involve removing the shared cache from the Internet. Although the removal of unnecessary parts tends to simplify the overall system, such a large configuration change would need to be analyzed before their actual removal. This paper presents our analysis on the effect of a shared DNS resolver based on campus network traffic. We found that (1) this removal can be expected to amplify the DNS traffic to the Internet by about 3.9 times, (2) the amplification ratio of the root DNS is much higher (about 6.3 times), and (3) removing all caching systems from the Internet is likely to amplify the DNS traffic by approximately 16.0 times. Thus, the removal of the shared DNS resolver is not a good idea. Our data analysis also revealed that (4) many clients without local caches generate queries repeatedly at short intervals and (5) deploying local caches is an attractive technique for easing DNS overhead because the amount of traffic from such clients is not small.

key words: *DNS, cache*

1. Introduction

The Domain Name System (DNS) [1], [2] forms a key part of the infrastructure of the Internet. Schomp et al. [3] recently discussed the benefits and adverse effects of a shared DNS resolver, which led to their subsequent proposal of removing the shared DNS resolver and using a full-service resolver at end clients instead. From the viewpoint of the cache mechanism, their proposal involves removing the shared cache (i.e., the shared DNS resolver) and using a local cache at the end clients (i.e., a local full-service resolver). Although the removal of unnecessary parts tends to simplify a system, such a large configuration change requires careful investigation before it can actually be deployed. In particular, the proposal will inevitably lead to the removal of the shared caching function from DNS.

Even though a careful and comprehensive analysis is needed on the effects of removing the shared DNS resolver, this matter has not yet been sufficiently investigated. Notably, the cache hit effect is one of the most important topics that needs to be analyzed. Because Schomp et al.'s proposal [3] depends on the assumption that a local cache can reduce DNS traffic to a reasonable extent, we focused on verifying this assumption. In practical terms, we analyzed the cache hit effect of the current shared DNS resolver based on the campus network traffic. We also estimated the effect of a local cache system based on the same traffic.

The remainder of this paper is organized as follows. Section 2 summarizes past research on DNS traffic. Section 3 explains the method used to estimate the effect of removing the shared DNS resolver. Our results are reported in Sect. 4, and Sect. 5 summarizes our findings.

2. Related Work

The importance of DNS has been recognized since the early days of the Internet. A number of researchers have studied DNS behavior [4]–[10].

For example, Jung et al. reported on DNS performance and the effectiveness of caching [4] and then proposed a model for time-to-live (TTL)-based Internet caches [5]. Zdrnja et al. [6] reported that their DNS response data included typo squatter domains, fast flux domains, and domains being used (and abused) by spammers. They also observed that the data locality of DNS requests diminishes because of the domains advertised in the spam. Callahan et al. passively monitored DNS and related traffic within a residential network to understand the impact of DNS server and client behaviors [8]. Schomp et al. characterized DNS clients with the aim of developing an analytic model of client interaction within the larger DNS ecosystem [9]. Chen et al. analyzed disposable domains that are likely generated automatically. These domains are characterized by a "one-time use" pattern and appear to be used as a way of "signaling" via DNS queries. Ishibashi et al. proposed the notion of hierarchical aggregate entropy and applied it to identify DNS client hosts that wastefully consume server resources.

These studies clarified the behavior of DNS. Especially, the DNS cache hit rate and its effects on shortening the response times and reducing the amount of traffic have been thoroughly studied. However, new issues related to DNS are appearing with the deployment of new services. The effects of CDNs and IPv6 as mentioned above are some examples of these new issues, with one of our previous papers [11] being an example of this kind of analysis.

Recently, the benefits and adverse effects of a shared DNS resolver have been discussed [3], [12], [13]. Schomp et al. [3] pointed out that shared DNS resolvers are complex,

Manuscript received June 22, 2018.
Manuscript revised October 5, 2018.
Manuscript publicized December 3, 2018.

[†] The author is with Japan Registry Services Co., Ltd., Tokyo, 101-0065 Japan.

[††] The authors are with University of Tsukuba, Tsukuba-shi, 305-8577 Japan.

a) E-mail: fujiwara@jprs.co.jp
b) E-mail: akira@cc.tsukuba.ac.jp
c) E-mail: yoshida.kenichi.ka@u.tsukuba.ac.jp

DOI: 10.1587/transcom.2018EBP3184

difficult to manage, and vulnerable to multiple forms of attacks, such as the injection of fraudulent records. They also claimed that the benefits of shared DNS resolvers are modest at best, and the use of a full-service resolver at end clients would provide a similar performance to the end user. On the other hand, other groups [12], [13] have tried to use shared DNS resolvers as a malware measure.

Although removing unnecessary parts tends to simplify a system, such a large system configuration change would have to be carefully analyzed before actual deployment. None of the previous works, including Schomp et al. [3], have supplied sufficient information to analyze the effect of this removal. Thus, in this work, we attempted to analyze the effect of the proposed removal in terms of DNS traffic.

The most important contribution of this work is that we found clear evidence against the removal of shared DNS resolvers. As shown in the following sections, such careless removal can be expected to increase the amount of DNS traffic, especially the root DNS traffic. Furthermore, the amount of DNS traffic is likely to increase by 3.9 times, and the root DNS traffic is likely to increase by 6.3 times.

An early version of this paper was presented at the 41st IEEE Computer Society Signature Conference on Computers, Software and Applications [14]. Although that version was based on the data of a single day, we improved the reliability for this paper by using continuous data over 13 months. We also report on the differences between weekdays, weekends, and holidays. Note that the stable operation of DNS service is indispensable for Internet operation. Thus, confirming the absence of changes in behavior through long-term observation of 13 months further supports the reliability of our study.

3. Model for Estimating Removal Effects

We constructed a DNS query model to evaluate the effects of a shared DNS resolver (i.e., a shared DNS cache) on the amount of DNS traffic. Our DNS query model comprises a (1) query generation model and (2) resolver behavior model. The purpose of the query generation model is to emulate the process according to which DNS queries are generated. The function of the resolver behavior model is to emulate how the resolver processes generated queries.

Section 3.1 explains the query generation model, and Sect. 3.2 explains the resolver behavior model. The proposed query model uses assumptions that we believe do not affect the characteristics we tried to analyze. The model does not consider CNAME chain following, out-of-bailiwick [15] name resolution, DNSSEC, the effect of the aggressive use of DNSSEC-validated cache (NSEC/NSEC3) [16], QNAME minimisation [17], and DNS cookies [18]. Section 3.3 explains the assumptions made by the resolver behavior model.

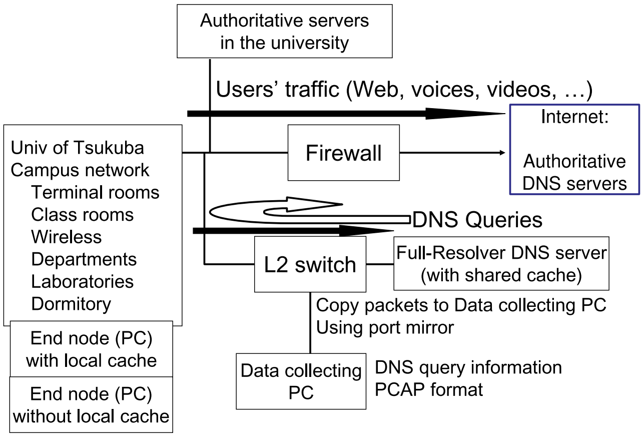

Authoritative servers in the university
Users' traffic (Web, voices, videos, ...)
Internet:
Authoritative DNS servers
Univ of Tsukuba
Campus network
Terminal rooms
Class rooms
Wireless
Departments
Laboratories
Dormitory
Firewall
DNS Queries
L2 switch
Full-Resolver DNS server (with shared cache)
Copy packets to Data collecting PC
Using port mirror
End node (PC) with local cache
End node (PC) without local cache
Data collecting PC
DNS query information
PCAP format

Fig. 1 DNS traffic data collecting environment.

3.1 Query Generation Model

For the query generation model, we simply used the recorded packet capture (pcap) file as the generator of the DNS query sequence. Thus, the pcap file is an indispensable part of our DNS query model.

We extract query sequences from pcap files by collecting DNS queries at the shared DNS full-resolver operated by the Academic Computing & Communication Center of the University of Tsukuba[†]. The center provided the shared DNS full-resolver for the university. The data collection topology is shown in Fig. 1. The campus network offered only IPv4 connectivity in 2016 – 2017.

End nodes in the campus network send DNS queries to the DNS full-resolver with a shared cache. The full-resolver sends queries to authoritative DNS servers in the Internet. At the collection point, all incoming and outgoing DNS packets of the full-resolver are captured. The captured packets include (1) packets between stub resolvers and the full-resolver and (2) packets between the full-resolver and authoritative DNS servers.

Authoritative DNS servers in the university are located in another part of the campus network. The captured data did not contain queries between the authoritative servers and the Internet. Thus the full-resolver data used in this study has information between campus network users and authoritative servers. There were about 22,000 campus network users (students, faculty members, and staffs) in May 2017.

This is the data collection system that we previously used [11]. Table 1 presents the main characteristics of the system. For the present work, we mainly analyzed the data collected by this system on May 31, 2017 (See Table 2 for a summary). This was the last weekday on which we collected DNS data. We used the data of this single day to

[†] Note that DNS query information contains private information of end users. We preserved the privacy inside this information as follows: the captured data were stored in one machine that only a limited number of researchers could use to access and analyze data, and the machine could only export the analyzed statistics.

Table 1 Cache hit rate and effect on authoritative DNS servers.

Cache hit rate [%]	Authoritative queries Divided by stub queries			Average Latency [ms]
	Root	TLDs	All	
75.1	0.00079	0.025	0.31	28.0

Table 2 Summary of collected DNS traffic.

Capture point	Full-resolver of campus network			
Capturing Date	May 1, 2016 – May 31, 2017 (13month)	May 31, 2017 0:00–24:00 (Weekday)	May 28, 2017 0:00–24:00 (Weekend)	Jul 31, 2016 0:00–24:00 (Holiday)
# of Clients	24,462	11,351	7,521	8,753
# of QNAMEs	82,147,609	503,844	210,002	368,007
Average # of queries/second	359	257.2	110.0	199.0
Cache hit rate		82.5%	80.8%	77.1%

carefully analyze the detailed behavior of the DNS system. We also confirmed the main characteristics by using data from a weekend (May 28, 2017), a holiday (July 31, 2016), and an extended period of 13 months (May 1, 2016–May 31, 2017). The collected data were stored in pcap format, and access information such as the contents of the DNS query, client that issued the query, and time when the query was issued were used to build the DNS query model.

3.2 Resolver Behavior Model

After the query generation model generates a DNS query sequence, the resolver behavior model emulates how the queries are to be processed. The model also assumes a three-layer hierarchy: the root, top-level domain (TLD), and second-level domains (SLDs) such as organizations. For example, if the pcap file contains a query for an A record (i.e., the IPv4 address) of "www.example.jp," the model assumes that the client without a cache first sends the query `www.example.jp A` to the root DNS servers and receives the NS record of `jp`. Then, the client sends the same query `www.example.jp A` to the TLD DNS server "`jp`" and receives the NS record of `example.jp`. After the client sends the query to the SLD DNS server, the client receives the A record (i.e., the IPv4 address `www.example.jp`). Thus, the resolver behavior model may generate three DNS queries from a single log record of the resolver.

Suppose that the pcap file contains:

```
Client X asks A of www.example.jp at 1:00
Client X asks A of www.example1.jp at 1:01
Client Y asks A of www.example.jp at 1:01
```

and the TTLs of the SLD are 300 (i.e., 5 minutes). Then,

Without cache

Client X sends queries to the root, TLD, and SLD DNS servers at 1:00 and receives three replies: two NS records and one A record. Both clients X and Y also send three queries and receive three replies at 1:01. Thus, there are nine queries sent and replies received in total.

This seems to be a harsh scenario. However, we estimate this situation to be comparable with other cases.

With a local cache

At 1:01, client X uses the cached information for the NS record of jp rather than send a query to the root. Thus, there are eight queries sent in total.

With the shared resolver cache

At 1:01, the shared resolver uses the cached information of the NS record of jp and example.jp. It also uses the cached information of the A record of www.example.jp. Thus, there are five external queries in total.

3.3 Assumptions of the DNS Query Model

Note that our three-layer DNS query model has the following defects and assumptions:

- Despite the existence of third- and higher-level authoritative DNS servers, most second-level DNS servers are the authoritative DNS servers for higher-level domain names. Thus, we assumed a three-layer hierarchy.
  Note that "." in the domain name mostly indicates zone cuts. However, many of the zone cuts related to only "jp" resulted in many entries in our pcap file, which prompted us to handle all "{ac,ad,co,ed,go,gr,lg,ne,or}.jp" as TLDs. We ignored other exceptions.
- Some operating systems such as Windows and Mac OS X have a DNS caching mechanism. They hide the true occurrence of DNS queries. This causes some of the real occurrences of a DNS query to be ignored.
  However, our campus network has a sufficient number of (old) Linux-based systems that lack a DNS caching mechanism[†]. For example, there were 6,475 DNS client computers, and over 95.0% of the DNS traffic originated from computers without a local cache (See Sect. 4 for details).
  As shown in Sect. 4, the experimental results indicated the existence of a statistical multiplexing effect with multiple clients (6,475 in this case). This statistical multiplexing effect differentiates the effects of local and shared caches on DNS traffic. Thus, we assumed this simple model to be a reasonable base from which to estimate the effects of shared cache removal.
- The assumption of a uniform TTL is the third defect of our resolver behavior model. We assumed that all SLDs use the same TTL and that the TTL of a TLD is 1 day.
  We compensated for this defect by analyzing the experimental results that were obtained with different TTLs (from 30 s to 1800 s) for SLDs. Because the results with different TTLs were consistent, we believe that we have sufficient evidence to analyze the difference between local and shared caches.

[†] Very recent versions of Linux-based systems also have a DNS caching mechanism. However, most Linux-based systems in the university are not new and lack a DNS caching mechanism.

- The resolver behavior model assumes that the cache system of resolvers has sufficient storage to enable resolvers to retain all information before their TTLs expire.
- Although real DNS resolvers sometimes issue multiple queries simultaneously, our resolver behavior model assumes that the resolver handles queries one by one. For example, if a client requests A and AAAA records of example.jp, its resolver issues the two corresponding queries simultaneously. If it is the first query about example.jp, the resolver may send two queries to the root DNS servers. However, our resolver behavior model assumes that the resolver sends only one query to the root DNS servers because it uses cached information.
- The resolver behavior model does not consider query name existence. Because the selected TTL value of 300 is similar to or less than the negative cache TTL value, this defect does not affect the conclusions presented below.
- The resolver behavior model does not consider CNAME chain following behavior and out-of-bailiwick name resolution. Because the cache mechanism tends to reduce both CNAME and out-of-bailiwick related traffic, this lack of consideration underestimates the importance of a shared cache and does not affect our conclusions.
- The resolver behavior model does not consider Domain Name System Security Extensions (DNSSEC) and the effect of the aggressive use of DNSSEC-validated cache (NSEC/NSEC3). Because DNSSEC validation increases DNSKEY queries at the same rate for the local and shared cache scenarios, this lack of consideration of DNSSEC also underestimates the importance of the shared cache and does not affect our conclusions.
  The aggressive use of DNSSEC-validated cache decreases nonexistent domain name queries at a similar ratio for the local and shared cache scenarios. Thus, the lack of consideration of the aggressive use of a DNSSEC-validated cache can be neglected.
- We also employed other assumptions about QNAME minimisation and DNS cookies. Because the number of queries did not change with these DNS functions, these assumptions did not affect the conclusions.
- The resolver behavior model does not consider response times because we mainly analyzed the number of query packets, except for the data presented in Sects. 4.3 and 4.4. Note that the discussions in these sections are based on actual data and do not depend on the model.

4. Experimental Results

4.1 Main Results

Figure 2 shows the number of outgoing DNS queries to authoritative DNS servers. The measured number of queries from the full-resolver to authoritative DNS servers and estimated number of DNS queries sent out from the network were compared under the assumption of an SLD TTL of 300 (i.e., 5 min). We used data collected on May 31, 2017. The x- and y-axes represent the time of day and number of queries sent out at that time, respectively. The number of queries per second is shown. The number of estimated queries with a shared cache was similar to the number of measured queries. Although we made many assumptions as described in Sect. 3.3, the model simulated the actual results reasonably well.

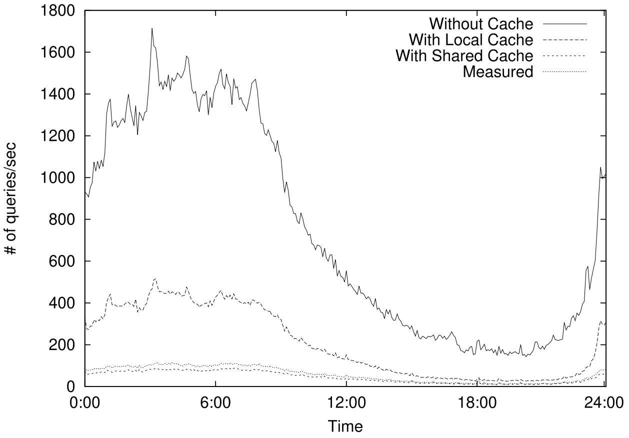


Fig. 2 Number of outgoing DNS queries per second (May 31, 2017, Wednesday).

Although the amplification rate slightly fluctuated during the day, the case without the cache amplified the DNS traffic 16.0 times on average. The case with the local cache amplified the DNS traffic 3.9 times. In other words, removing the shared DNS resolver as proposed by Schomp et al. [3] would increase the DNS traffic by 3.9 times. This result led us to conclude that removing the shared DNS resolver is not a good idea.

Figure 3 shows the fraction of the amplification ratio every 5 min under the assumption of DNS traffic with the shared resolver cache as the base. The shared resolver reduced the traffic by using the statistical multiplexing effect of multiple clients. The removal of shared cache should be avoided to prevent every attempt amplifying the amount of DNS traffic by 3.9 times.

The amplification ratio of 16.0 was larger than we expected based on the previous study which reported the cache hit rate as being 75.1%. Our first guess before the experiment was that the amplification ratio would be about 4–5. However, without any cache mechanism in the system, we observed a huge number of queries to the root DNS servers that replied an NS RRSet of TLDs. Because most TTLs of TLD are greater than 1 day, the actual pcap file does not contain the corresponding records. However, the simulation of scenario without the cache was flooded with such queries.

Figure 4 shows the estimated number of DNS queries sent out from the network under the assumption of an SLD TTL of 300. The daily data are shown from May 1, 2016 to May 31, 2017. The amplification ratios in Fig. 4 look similar to those in Fig. 2. This indicates the stability of the DNS

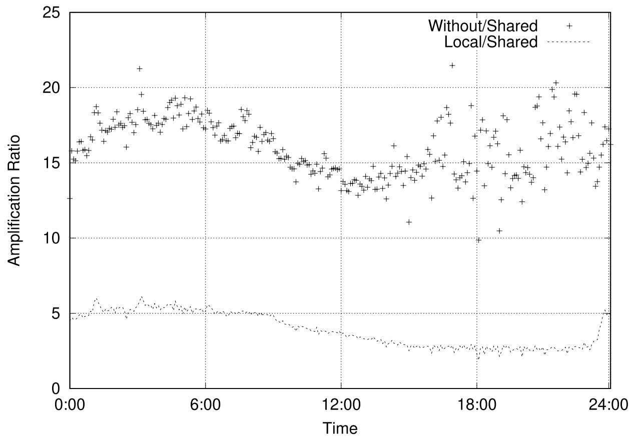


Fig. 3 Access amplification ratio (May 31, 2017, Wednesday).

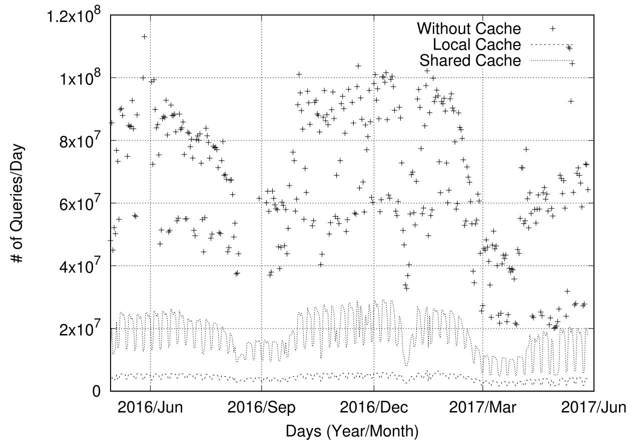
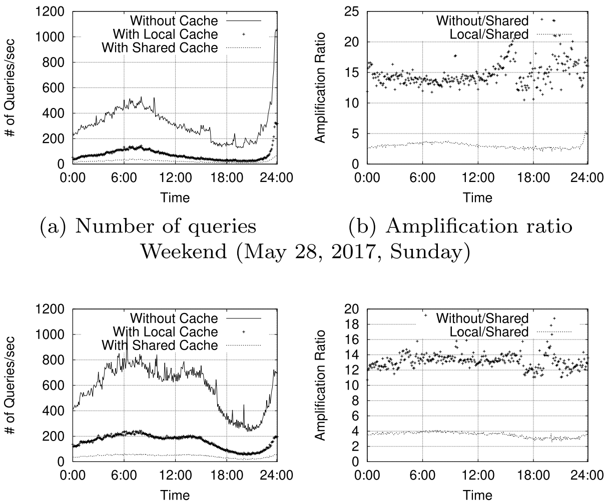


Fig. 4 Number of daily DNS queries (May 1, 2016–May 31,2017).



(a) Number of queries (b) Amplification ratio
Weekend (May 28, 2017, Sunday)

(c) Number of queries (d) Amplification ratio
Summer holiday (July 31, 2016)

Fig. 5 DNS queries and amplification ratios of holidays.

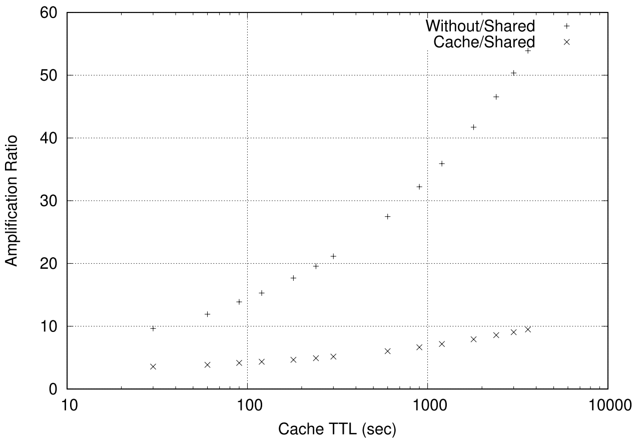


Fig. 6 Access amplification ratio per TTL (May 1, 2016–May 31, 2017).

cache effect and the seasonal behavior of DNS: (1) activity distinctly decreases during the summer and spring holidays, (2) activity also decreases slightly during the winter holidays, and (3) the activity differs on weekdays and weekends.

Although the activities of weekdays and weekends were different (e.g., the volume of traffic), their characteristics were similar. The amplification ratio was about 12–16, and the volume decreased at midnight. Figure 5 shows typical examples: the numbers of DNS queries and amplification ratios on the weekends (May 28, 2017) and holidays (July 31, 2016). As shown in Fig. 4, the DNS activity (i.e., number of DNS queries) decreased on both the weekends and holidays. However, the amplification ratios were similar for both periods.

Figure 6 shows how the amplification ratio changes with TTL. We assumed that the TTL value of RRSets from root servers was 86,400 (1 day) and changed the TTL values of RRSets from the TLD and SLD servers as 30, 60, 120, 180, 300, 600, 1200, and 1800. For this figure, we used data over 13 months to accurately estimate the amplification ratio. However, each day (e.g., May 31, 2017) had a similar amplification ratio. As expected, a longer TTL resulted in a larger amplification ratio. In other words, the cache system was more effective with a long TTL. However, even when we assumed very short TTLs (e.g., 30 s), the amplification ratio resulting from the removal was too large even when a local cache was used(e.g., 3.6 times for TTL of 30 s). Because the amplification affects the root DNS traffic more radically (see Sect. 4.2), we strongly oppose the proposal to remove the shared resolver.

As noted previously, the assumption of a uniform TTL is a defect of our resolver behavior model. In reality, each domain name has a different TTL. In other words, reality comprises a mixed situation, contrary to the uniform situation that we assumed. However, the statistical multiplexing effects were observed even when we changed the assumed TTL. We do not expect significant differences between the statistical multiplexing effects of the uniform and mixed situations. The number of redundant queries from different clients is the source of the statistical multiplexing effect and exists in both uniform and mixed situations. The statistical multiplexing effect observed with the shortest TTL (i.e., 3.6 times for 30 seconds) seems to be a reasonable estimation of the minimum difference between the scenarios with the shared resolver cache and with a local cache. The real situation would have more redundant queries.

4.2 Detailed Characteristics of the Amplification

Because the amplification ratio of 16.0 was much larger than expected, we carefully analyzed the estimation results by using a single day of data (i.e., May 31, 2017). Figure 7 shows the number of DNS queries of each level. It compares the numbers of queries without a cache, with a local cache, and with a shared cache (i.e., a shared DNS resolver). Interestingly, the number of SLD queries was larger than the number of TLD queries with the local cache and shared cache. This shows that the caching mechanism is effective with TLD queries. Because the SLD names (i.e., organizations) have much variety, this slightly negates the effect of the caching system at the SLD.

Figure 8 shows the amplification ratio of each level. An important feature is the amplification ratio of the scenario with the local cache for the root level. Although the root DNS servers are an indispensable component of the DNS service and the load to the root DNS servers should be minimized, these results show that the root DNS servers received a 6.3 times larger load if the shared resolver cache was replaced with a local cache. The greater traffic amplification effect on the root DNS servers is another important reason why we object to removing the shared DNS resolver.

Note that Fig. 8 also shows that the root and TLD DNS servers received more severely amplified traffic (6.3 times for the root and 7.0 times for TLD) when the shared DNS resolvers were removed. Even if the effect on the SLD authoritative DNS servers is moderate (3.9 times), the increase in DNS traffic is undeniable.

4.3 Frequent Queries and Active Clients

During the experiments, we found the following: (1) a few clients repeatedly issued many of the same DNS queries in a short interval, and (2) a few QNAMEs (i.e., target domain name of the query) were repeatedly issued in a short interval. To analyze the mechanism behind these findings, we interviewed system mangers of related computers during September 2016.

Figure 9 shows the cumulative log number of clients per interval, and Fig. 10 shows the cumulative log number of QNAMEs per interval for the data on September 10, 2016. This was the date we chose the related computers for the interview. In Fig. 9, the x- and y-axes represent the average interval of the client by which the client issues DNS queries and the cumulative log number of queries and clients, respectively. In Fig. 10, the y-axis represents the cumulative log of queries and QNAMEs.

As shown in these figures, 65.2% of the clients issued DNS queries with an average interval of less than 60 seconds. The total number of DNS queries from these clients constituted 95.0% of all DNS queries, and 43.3% of QNAMEs were sent within an interval of less than 60 seconds. These made up 69.5% of the total number of queries.

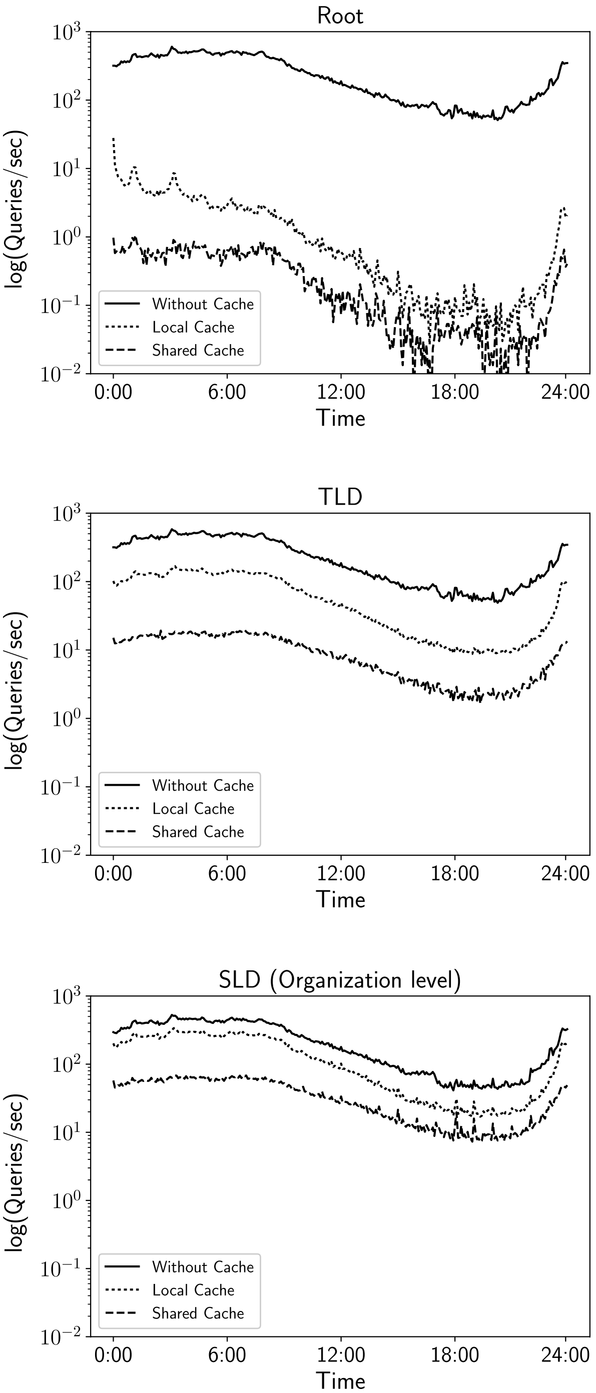


Fig. 7 Number of DNS queries (May 31, 2017).

Note that the DNS queries arrived in bursts. A DNS query is not a Poisson process, and the query interval does not follow a Gaussian distribution. Thus, Figs. 9 and 10 are not enough to analyze the behavior.

For deeper analysis of the data, Figs. 11 and 12 show the behaviors of the top five frequent QNAMEs and top five

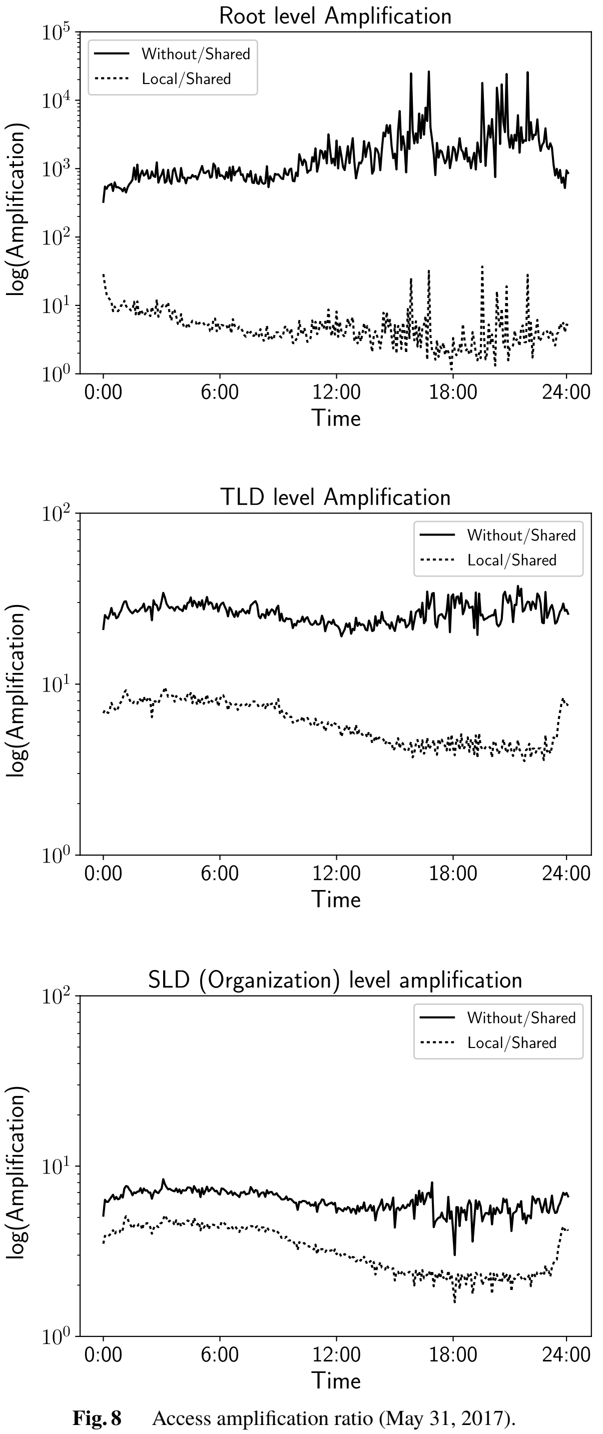


Fig. 8 Access amplification ratio (May 31, 2017).

active clients, respectively. In Fig. 11, the subfigures on the right show histograms of the intervals at which each query was issued. The histograms were made at 1 second intervals. The top five QNAMEs were sent within an interval of less than 30 seconds. Note that the third QNAME (Fig. 11(c)) was sometimes issued with a longer interval. However, the y-axis of these subfigures uses log-scaling. Thus, most of the third QNAMEs were also issued with a short interval.

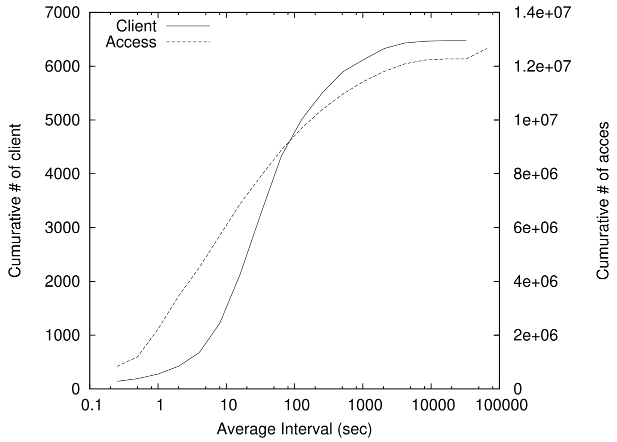


Fig. 9 Cumulative access of clients (Sep. 10, 2016).

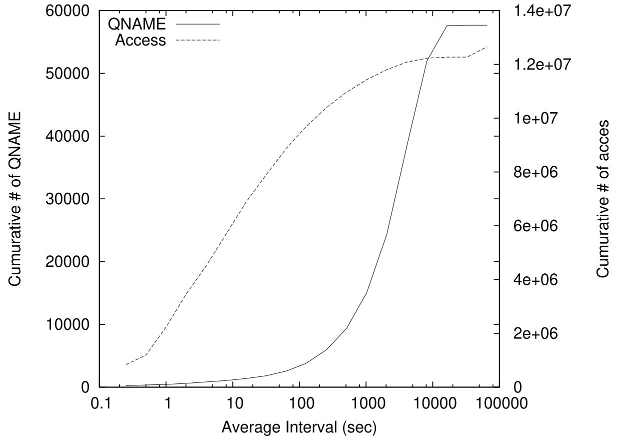
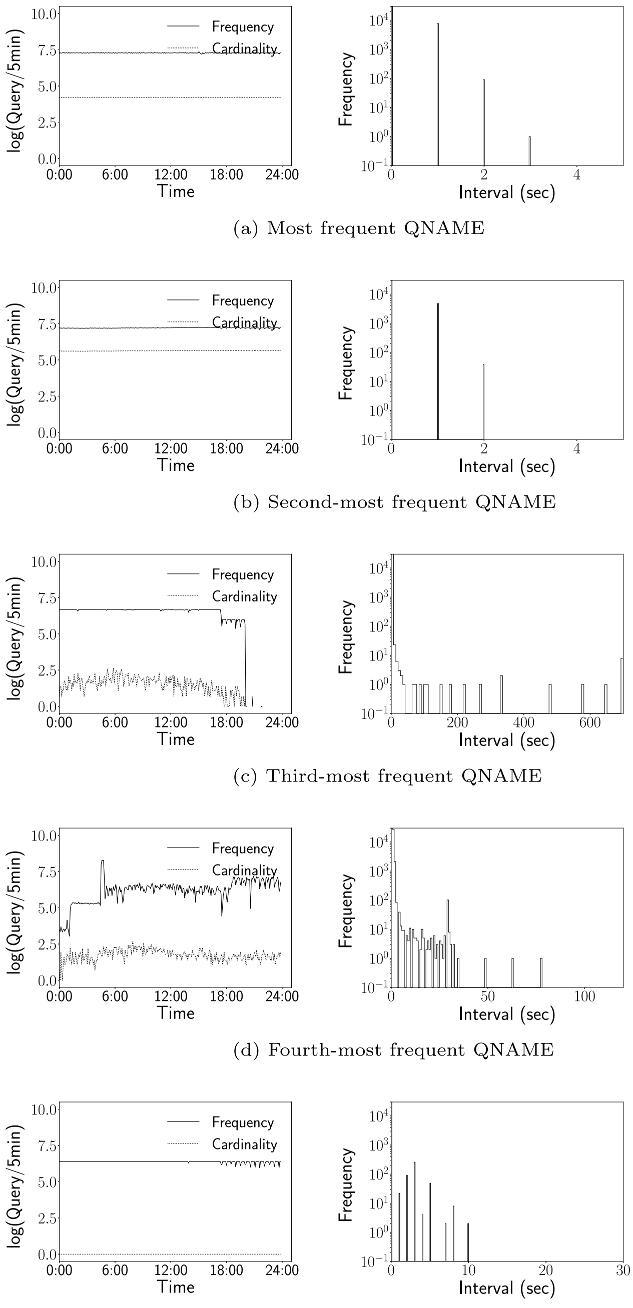


Fig. 10 Cumulative access of QNAMEs (Sep. 10, 2016).

The subfigures on the left in Fig. 11 show how the access behavior changed during the course of the day. The x- and y-axes represent the time of day and number of accesses, respectively. "Frequency" signifies the number of queries issued, and "cardinality" denotes the number of clients that issued a QNAME. For example, if client A issues query X twice and client B issues query X once, the frequency of Query X is 3. However, the cardinality is 2. As shown in the figure, the same client repeatedly issued the same QNAME. The most extreme case is shown in Fig. 11(e). Only one client sent the corresponding QNAME multiple times.

Although we object to the removal of the shared DNS resolver, we accept the importance of local cache installation. Such a local cache can eliminate multiple queries sent from the same client. For the case shown in Fig. 11(e), about 600 accesses per 5 minutes can be reduced to a single access by the local cache.

Figure 12 shows the client-based version of Fig. 11. Here, "cardinality" is the variety of QNAMEs each client issues, and "frequency" is the number of queries issued. These top five active clients issued DNS queries within a very short interval. In approximate terms, the second- and fifth-most active clients only asked for a single QNAME (see "cardinality" of Figs. 12(b) and (e)). Thus, a similar traffic



(a) Most frequent QNAME

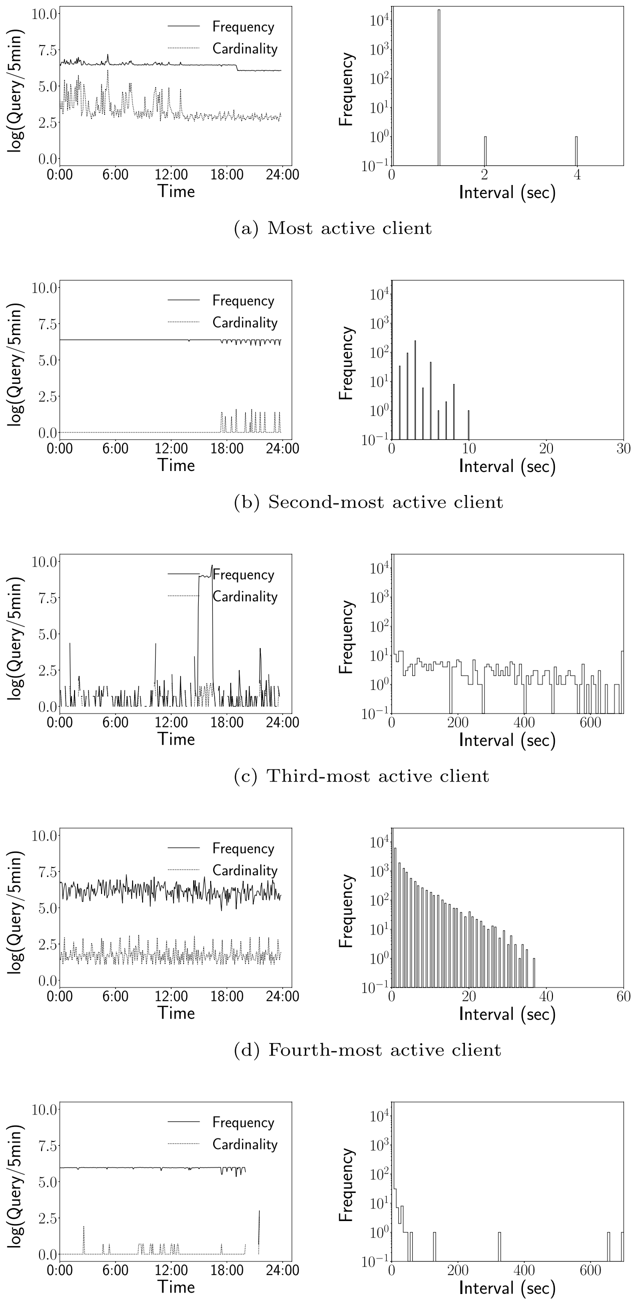
(b) Second-most frequent QNAME

(c) Third-most frequent QNAME

(d) Fourth-most frequent QNAME

(e) Fifth-most frequent QNAME

Fig. 11 Access pattern of QNAMEs (Sep. 10, 2016).



(a) Most active client

(b) Second-most active client

(c) Third-most active client

(d) Fourth-most active client

(e) Fifth-most active client

Fig. 12 Access pattern of clients (Sep. 10, 2016).

reduction (i.e., from 600 accesses down to a single access) can be expected.

Among the domain names of which the related queries are shown in Fig. 11, the second-, third-, and fourth-most frequently issued QNAMEs seemed to use the DNS-based server load balancing technique [19] with short TTLs (i.e., 30, 60, and 30 s, respectively). The most and fifth-most frequent QNAMEs used TTLs longer than 1 hour (24 hours and 1 hour, respectively). Thus, the reason for these repeated queries where the interval is less than 10 seconds is not on the server side but on the client side.

According to the system managers of the most and the fifth-most frequently issued domains, they both deployed keep-alive processes inside our campus network. Because their keep-alive processes periodically and frequently sent out packets, such packets caused frequent issues with DNS queries. Note that most of the Linux-based systems lacked a local DNS cache mechanism. The individual processes to generate keep-alive messages and the lack of a local cache were the cause of the frequent DNS queries.

We believe the same situation exists for other queries. Thus, the installation of a local cache seems to alleviate this

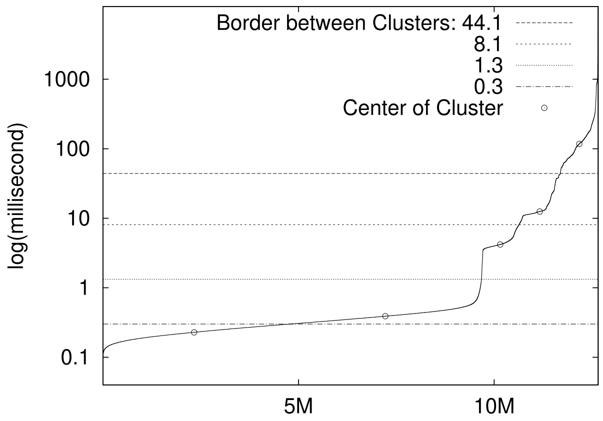


Fig. 13 Cumulative distribution of the response time.

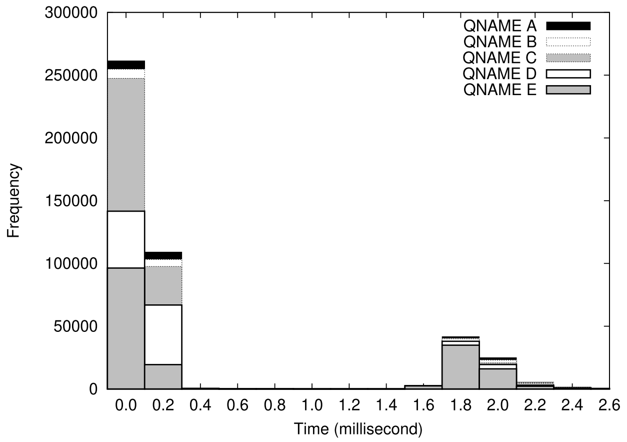


Fig. 14 Response times of five example query names.

situation.

4.4 Response Time

Although we are mostly against Schomp et al.'s proposal [3], we support the installation of a local cache, which inevitably leads from their arguments. The analysis of the response time clarified the appropriateness of the local cache.

Figure 13 shows the distribution of the current response time from the shared DNS resolver to clients. K-means analysis found five clusters of data with borders of 0.3, 1.3, 8.1, and 44.1 milliseconds between them. Each small circle (i.e., center) shows the median of each cluster.

For the two clusters with the lowest and second-lowest response times, the information was retrieved from the cache of the shared resolver. Although a response time of less than 390 microseconds (i.e., median of the second-lowest response time cluster) is difficult to achieve by information retrieval through the Internet, a local cache can achieve a similar or shorter response time without accessing data via a network. Note that the amount of storage required by a local cache is less than that of a shared resolver cache. Thus, we do not expect any difficulty in preparing storage for a local cache to realize a similarly short DNS retrieval time.

Figure 14 shows the distribution of the DNS response time for five query names (A, B, C, D, E). Here, typical five query names with two peaks (corresponding to a cache hit and miss) were selected as the distribution of RTT values. The response times formed two clusters of less than 200 microseconds and about 2 ms. Figure 13, whereas the second cluster in Fig. 14 corresponds to the third cluster in Fig. 13. The examples in Fig. 14 clearly show that DNS retrieval via the Internet requires a response time that is 10 times greater. The use of a local cache is a reasonable countermeasure.

5. Conclusion

DNS is a key part of the Internet infrastructure. Removing the shared DNS resolver and installing a full-service resolver at the end client have recently been discussed. Although the removal of unnecessary parts tends to simplify a system, such a large system configuration change should be carefully analyzed before actual deployment.

This paper presents our analysis of the cache effect of the shared DNS resolver based on campus network traffic. Our findings are as follows:

- Such a removal of the shared cache and replacement with a local cache can be expected to amplify the DNS traffic by about 3.9 times.
- The amplification ratio is much worse for the root DNS (about 6.3 times).
- Removal of all caching systems from the Internet is likely to amplify the DNS traffic by about 16.0 times.

This led us to conclude that removing the shared DNS resolver is not a good idea.

Although the increase in traffic (i.e., 6.3 times amplified traffic to root DNS servers and 3.9 times amplified traffic to SLD DNS servers) can be processed by current Internet infrastructure, DNS is the backbone of the Internet. Trying to reduce unnecessary load on DNS is our basic stance as a DNS operator.

Our data analysis also showed that many clients without local caches generate queries repeatedly at short intervals (less than 1 minute). Because the amount of traffic from such computers was not small (about 95.0% of all network traffic), deploying local caches is an attractive technique for easing DNS overhead.

The DNS situation continues to change. Because DNS is a key part of the Internet infrastructure, we are continuously monitoring it and its performance. In a past study [11], we analyzed the effect of IPv6 deployment on DNS. In this study, we analyzed the effect of a shared DNS resolver. In the future, we are planning to analyze the effects of DNSSEC, DNSSEC with RFC 8198, misconfiguration, and malicious use of DNS. However, these were outside the scope of the current study and left as future research issues.

References

[1] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (INTERNET STANDARD), Nov. 1987. Updated by RFCs 1101,

1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020.

[2] P. Mockapetris, "Domain names - implementation and specification," RFC 1035 (INTERNET STANDARD), Nov. 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604, 7766.

[3] K. Schomp, M. Allman, and M. Rabinovich, "DNS resolvers considered harmful," Proc. 13th ACM Workshop on Hot Topics in Networks, HotNets-XIII, pp.16:1–16:7, ACM, New York, NY, USA, 2014.

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

[5] J. Jung, A.W. Berger, and H. Balakrishnan, "Modeling TTL-based Internet caches," IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. no.03CH37428), vol.1, pp.417–426, March 2003.

[6] B. Zdrnja, N. Brownlee, and D. Wessels, "Passive monitoring of dns anomalies," Detection of Intrusions and Malware, and Vulnerability Assessment, pp.129–139, 2007.

[7] Y. Chen, M. Antonakakis, R. Perdisci, Y. Nadji, D. Dagon, and W. Lee, "DNS noise: Measuring the pervasiveness of disposable domains in modern DNS traffic," 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp.598–609, June 2014.

[8] T. Callahan, M. Allman, and M. Rabinovich, "On modern DNS behavior and properties," SIGCOMM Comput. Commun. Rev., vol.43, no.3, pp.7–15, July 2013.

[9] K. Schomp, M. Rabinovich, and M. Allman, "Towards a model of DNS client behavior," Passive and Active Measurement, pp.263–275, Springer International Publishing, Cham, 2016.

[10] K. Ishibashi and K. Satoh, "Identifying DNS anomalous user by using hierarchical aggregate entropy," IEICE Trans. Commun., vol.E100-B, no.1, pp.140–147, Jan. 2017.

[11] K. Fujiwara, A. Sato, and K. Yoshida, "DNS traffic analysis: Issues of IPv6 and CDN," Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on, pp.129–137, IEEE, 2012.

[12] NTT Communications, "OCN Mulware Block Service," https://www.ntt.com/personal/ocn-security/info/malware.html

[13] CISCO, "Cisco Umbrella," https://www.cisco.com/c/m/ja_jp/solutions/cisco-start/product-umbrella.html

[14] K. Fujiwara, A. Sato, and K. Yoshida, "Cache effect of shared DNS resolver," Proc. 41th IEEE Computer Society Signature Conference on Computers, Software and Applications, IEEE Computer Society, 2017.

[15] P. Hoffman, A. Sullivan, and K. Fujiwara, "DNS terminology," RFC 7719 (Informational), Dec. 2015.

[16] K. Fujiwara, A. Kato, and W. Kumari, "Aggressive use of dnssec-validated cache," RFC8198, July 2017.

[17] S. Bortzmeyer, "DNS query name minimisation to improve privacy," RFC 7816 (Experimental), March 2016.

[18] D.E. 3rd and M. Andrews, "Domain name system (DNS) cookies," RFC 7873 (Proposed Standard), May 2016.

[19] A.J. Su, D.R. Choffnes, A. Kuzmanovic, and F.E. Bustamante, "Drafting behind Akamai: Inferring network conditions based on CDN redirections," IEEE/ACM Trans. Netw., vol.17, no.6, pp.1752–1765, Dec. 2009.

Kazunori Fujiwara received his M.E. from Waseda University in 1991 and Ph.D. from the University of Tsukuba in 2015. He worked as a research associate at Waseda University. Since 2002, he has been a researcher at Japan Registry Services Co., Ltd. His research interests are DNS and other Internet protocols. He is a member of the IPSJ and IEICE.

Akira Sato received his Ph.D. from the University of Tsukuba in 1998. He is an Associate Professor in the Department of Information Engineering, Academic Computing and Communications Center at the University of Tsukuba. His current research interest is the operation of academic networks. He is a member of the IEICE and IPSJ.

Kenichi Yoshida received his Ph.D. from Osaka University in 1992. In 1980, he joined Hitachi Ltd., and he has been working for the University of Tsukuba since 2002. His current research interests include applications of the Internet and machine learning techniques.