PAPER *Special Section on Internet Technologies to Accelerate Smart Society*

Analysis of DNS TXT Record Usage and Consideration of Botnet Communication Detection*

Hikaru ICHISE[†a)], ***Nonmember***, **Yong JIN**[††b)], ***Member***, ***and*** **Katsuyoshi IIDA**[†††c)], ***Senior Member***

SUMMARY There have been several recent reports that botnet communication between bot-infected computers and Command and Control servers (C&C servers) using the Domain Name System (DNS) protocol has been used by many cyber attackers. In particular, botnet communication based on the DNS TXT record type has been observed in several kinds of botnet attack. Unfortunately, the DNS TXT record type has many forms of legitimate usage, such as hostname description. In this paper, in order to detect and block out botnet communication based on the DNS TXT record type, we first differentiate between legitimate and suspicious usages of the DNS TXT record type and then analyze real DNS TXT query data obtained from our campus network. We divide DNS queries sent out from an organization into three types — via-resolver, and indirect and direct outbound queries — and analyze the DNS TXT query data separately. We use a 99-day dataset for via-resolver DNS TXT queries and an 87-day dataset for indirect and direct outbound DNS TXT queries. The results of our analysis show that about 30%, 8% and 19% of DNS TXT queries in via-resolver, indirect and direct outbound queries, respectively, could be identified as suspicious DNS traffic. Based on our analysis, we also consider a comprehensive botnet detection system and have designed a prototype system.

key words: *botnet communication, DNS TXT record, via-resolver DNS query, direct outbound DNS query, and indirect outbound DNS query*

1. Introduction

Botnet, a malicious logical network of cyber attackers, has become a significant security threat in cyberspace [4], [5]. There are many well known botnets, such as Conficker, Storm, TDL4, and Zeus, each of which once infected millions of computers [6]. Once a computer is infected by a bot program, which is a kind of malware and also a core program of a botnet, it can attempt several kinds of cyber attack such as Advanced Persistent Threat (APT), Distributed Denial of Service (DDoS), spreading spam mails, and phishing [6], [7]. Fig. 1 shows a typical workflow of a botnet-based cyber attack. First, a computer within an organization somehow gets infected by a bot program such as through web browsing, spam mail, or clicking a phishing site by mistake. After that, the bot program sends probes to its corresponding Command and Control (C&C) server to identify its existence as well as to update its status. After it is finished collecting a number of bot-infected computers, the C&C server can instruct them to perform several kinds of cyber attack. Here, we refer to the communication between a bot-infected computer and the C&C server, which is the most important information transmission in a botnet-based cyber attack, as *botnet communication*. With regard to the above workflow, in this research we target the botnet communication as a means to analyze and detect botnet-based cyber attacks.

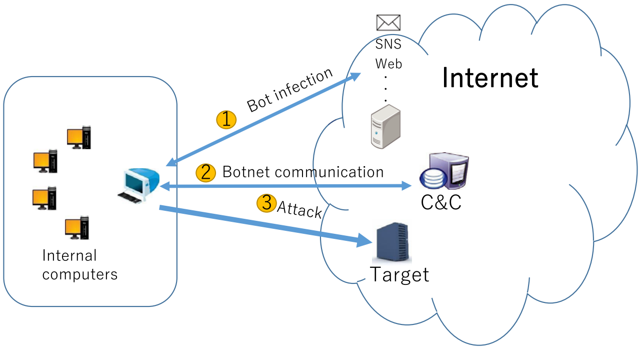


Fig. 1 Typical workflow in a botnet attack.

To date, Internet Relay Chat (IRC), Hypertext Transfer Protocol (HTTP) and Peer-to-Peer (P2P) protocols have been used in botnet communication [8]. In addition, many recent reports indicate that the Domain Name System (DNS) protocol [9], [10] is also used in botnet communication [11]–[13]. Basically, DNS protocol is mainly used for name resolution, which is translating the hostname to an IP address in the Internet, but the increasing popularity of Internet services has led to some minor records, such as DNS TXT, which is used in many Internet services. In [14], Xu et al. empirically show that cyber attackers can effectively hide botnet communication by using a DNS-based stealthy messaging system that uses hash functions to encode the contents. In [15], Anagnostopoulos et al. show how mobile botnets use DNS protocol in botnet communication and also evaluate the magnitude of DNS-based amplification attacks. Consequently, DNS packets, which are currently considered to be secure network traffic, have also become a target of monitored communication since network administrators cannot simply block all DNS traffic. Thus, an effective detection solution for botnet communication is needed.

In this research, as a preliminary step to develop a

Manuscript received February 20, 2017.
Manuscript revised May 22, 2017.
Manuscript publicized July 5, 2017.

[†] The author is with Technical Department, Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

[††] The author is with Global Scientific Information and Computing Center, Tokyo Institute of Technology, Tokyo, 152-8550 Japan.

[†††] The author is with Information Initiative Center, Hokkaido University, Sapporo-shi, 060-0811 Japan.

[*] Preliminary versions of this paper have been presented in [1]–[3].

a) E-mail: hichise@nap.gsic.titech.ac.jp
b) E-mail: yongj@gsic.titech.ac.jp
c) E-mail: iida@iic.hokudai.ac.jp

DOI: 10.1587/transcom.2017ITP0009

method for detecting DNS-based botnet communication, we analyze real DNS traffic from our campus network. There are three possible ways to use DNS traffic in botnet communication: through via-resolver, indirect outbound and direct outbound communication. The via-resolver type completely uses DNS full resolvers in botnet communication which is a typical usage in normal name resolution. The indirect outbound type partially uses DNS full resolvers only to obtain the IP addresses of C&C servers. After that, the bot-infected computers communicate with the C&C servers directly using DNS protocol. The direct outbound type does not use DNS full resolvers at all, but communicates with the C&C servers directly from the beginning. All three types have recently been detected in DNS-based botnet communication and reported in the literature [16], [17]. More importantly, recent detections also indicate that DNS TXT record, which has flexible usages compare to other DNS resource records, is increasingly being used in botnet communication.

Considering the above, in our analysis we focus on DNS TXT record and investigate the DNS traffic obtained from the DNS full resolvers of our campus network over a period of 99 days for via-resolver type botnet communication. For the other two types, we investigate all the DNS traffic, excluding the via-resolver DNS traffic, obtained at the gateway of the campus network to the Internet for 87 days. Based on our analysis, we also consider a detection system for the three types of DNS-based botnet communication and show the basic system architecture.

2. Related Work

Hiding information in DNS traffic is not a new technology. The first implementation, called DNS tunneling, appeared in 1998 [8] and the target information embedded in FQDN and CNAME was used. After that, in 2004, Kaminsky presented his implementation to tunnel arbitrary data over DNS traffic to the security community [18]. A few years later, DNS-based VPN was created [19].

There are some conventional ways to detect "abnormal" DNS traffic. In 2011, several reports discussed the existence of DNS-based botnet communication and detection approaches [12], [16], [17]. So far as we know, [12] is the earliest one about DNS-based botnet communication. It discussed ways to detect DNS abnormal usage attempts and also provided recommendations to mitigate exposure. In [16], the authors analyzed 14 million DNS transactions of DNS TXT records and found instances of botnet communication used by "Feederbot". In Feederbot, the IP addresses of C&C servers are embedded in the bot program and the botnet communication is performed directly with the C&C servers using DNS TXT record without going through DNS full resolvers. In [17], another botnet named Morto has been reported, where DNS TXT record was also used in botnet communication. In Morto, a bot-infected computer works in collaboration with Domain Flux [20] to find C&C servers by querying the IP address of the generated domain names using DNS full resolvers. After identifying the C&C servers, the bot-infected computer performs botnet communication using DNS TXT record with them directly. In [21], the authors introduced several utility software applications for creating and detecting DNS tunnels. In [22], the authors analyzed an archived malicious dataset covering one year and a 30-day real DNS traffic obtained from a DNS full resolver, and they confirmed that a high ratio of DNS TXT record transactions might increase the risk of infection by botnet.

As we can see from the existing research, although the usage of DNS protocol and DNS TXT record in botnet communication has not gone unexamined, no clear conclusions regarding the official and malicious usages of DNS TXT record or the architecture of DNS-based botnet communication (via-resolver, indirect and direct outbound) have been provided. Since DNS TXT record is also used for legitimate purposes, such as Sender Policy Framework (SPF) [23] and domainkeys [24], which are used for email sender authentication, the proper usages of DNS TXT record need to be identified in order to detect botnet communication. More importantly, in addition to the traffic of DNS TXT record, which is a medium for botnet communication, the botnet communication architecture also needs to be considered since new DNS resource records can also appear as new media.

3. DNS TXT Record Type and Scope of this Paper

While there are three types of DNS traffic — via-resolver, indirect outbound and direct outbound — the objective of this research is to devise a way to detect any DNS-based botnet communication using DNS TXT records. In this section, we start by examining the DNS TXT resource record and the three types of DNS traffic.

3.1 DNS TXT Record Type

Although the major objective of DNS protocol is to provide name resolution between the hostname and IP address, it also provides supplementary functions using TXT (Text), MX (eMail eXchange), SRV (Service), and other resource record types. Of these, the DNS TXT record type is relatively flexible to use and provides a field to store some short text descriptions. Conventionally, the original RFC1035 only provided 512 octets for each UDP DNS packet[†], but now it becomes possible that each UDP DNS packet can provide more than 4000 octets through "Extension mechanisms for DNS (EDNS0)" [25]. This standard allows us to store a large amount of information in DNS TXT records, such as SPF records and DomainKeys which are used to deal with spam mail protections[††]. Besides SPF and DomainKeys records, there are other legitimate ways to use the DNS TXT record type based on application requirements, but we will not describe them in detail here. Unfortunately, EDNS0 also provides possibilities for some malicious parties to use the DNS

[†] DNS also supports TCP, but we focus on the majority part.

[††] SPF and DomainKeys resources records use the DNS TXT record type in DNS protocol.

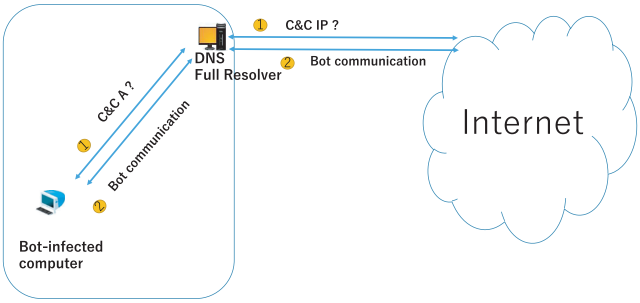


Fig. 2 An example of via-resolver DNS query.

TXT record type to transport malicious informati on useful for cyber attacks depending on its flexibility. Therefore, we need to establish a method to differentiate the legitimate and malicious DNS traffic, especially that using the DNS TXT record type, which has appeared in botnet communication.

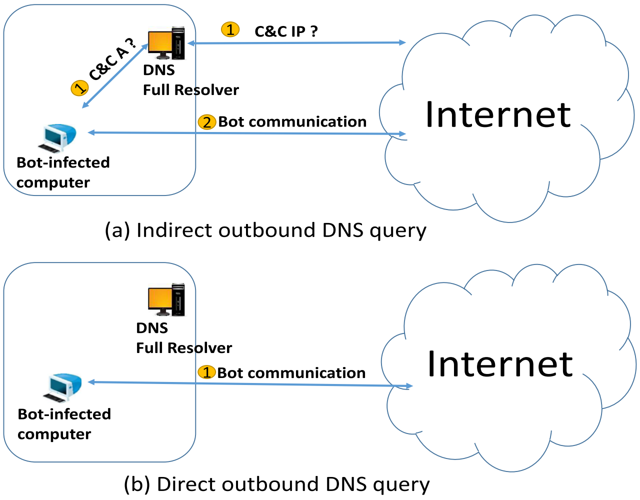
3.2 Via-Resolver DNS Query

In general, most organizations set up multiple DNS full resolvers to provide name resolution service to their internal computers. In this case, the internal computers send name resolution requests to one of the DNS full resolvers, which performs the name resolution on their behalf. We define such a name resolution request as a via-resolver DNS query in this research and Fig. 2 shows an example of via-resolver name resolution. The via-resolver name resolution relies completely on DNS full resolvers, so we need to monitor all DNS traffic on the DNS full resolvers in order to analyze a via-resolver DNS query. Several researchers have reported the usage of a via-resolver DNS TXT query in botnet communication [8], [22], so the via-resolver DNS TXT query is one of our monitoring and analysis targets when we design a botnet communication detection system.

3.3 Indirect and Direct Outbound DNS Queries

Even though official DNS full resolvers are available, some internal computers may use their private DNS full resolvers or public DNS full resolvers (from the Internet) for purposes such as name resolution and independent configuration of a DNS full resolver. In this research, we define this kind of name resolution request without using official DNS full resolvers as a direct outbound DNS query. Several studies have also reported the use of direct outbound DNS queries in botnet communication [8], [16], [17].

Two types of outbound DNS query, which include indirect outbound query and direct outbound query, are involved in botnet communication. One is where the bot-infected computer uses the DNS full resolvers to query the IP address of the C&C server only and then sends DNS queries to the C&C server directly, as shown in Fig. 3(a). We refer to this as an indirect outbound DNS query. Note that the difference between via-resolver DNS query and indirect outbound DNS query is the way of bot communication, each of which is illustrated as arrow numbered two in Figs. 2 and 3(a), respectively. The second type is where the bot-infected computer never uses the DNS full resolvers, but instead sends DNS queries to the C&C server directly from the start, as shown in Fig. 3(b). We refer to this as a direct outbound DNS query. In this case, the IP address of the C&C server might have been embedded in the bot program at the installation stage, so the bot-infected computer can communicate with it directly without looking for its IP address via the DNS full resolvers.



(a) Indirect outbound DNS query

(b) Direct outbound DNS query

Fig. 3 Direct & Indirect outbound DNS queries.

As well as the DNS TXT record type, there are other exceptions where a direct outbound DNS query is a legitimate usage, such as updates of anti-virus software, the use of DNS black lists to check for spam mail, and the use of public DNS resolvers. Some anti-virus software use the DNS TXT record type to check the update status and some spam-mail checkers use the DNS TXT record type to check the domain name in DNS black lists. These usages of the DNS TXT record type are legitimate, so we need to differentiate them from malicious usages. On the other hand, several public DNS full resolvers have been launched to provide an effective name resolution service by using direct outbound DNS queries. The Google public DNS [26] is a popular public DNS resolver, which is announced with the IP addresses “8.8.8.8” and “8.8.4.4”, and some internal computers in many organizations may use these addresses for name resolution. Obviously, the DNS queries to these public DNS resolvers should be categorized as direct outbound DNS queries since they do not use the DNS full resolvers. Therefore, we also need to filter them out from malicious direct outbound DNS queries, but we have to include them into the analysis of DNS TXT usages.

4. Analysis of Via-Resolver DNS TXT Query

As explained in Sect. 3.2, the DNS TXT record type is used for botnet communication in various botnets. To detect botnet communication based on DNS TXT record type, legiti-

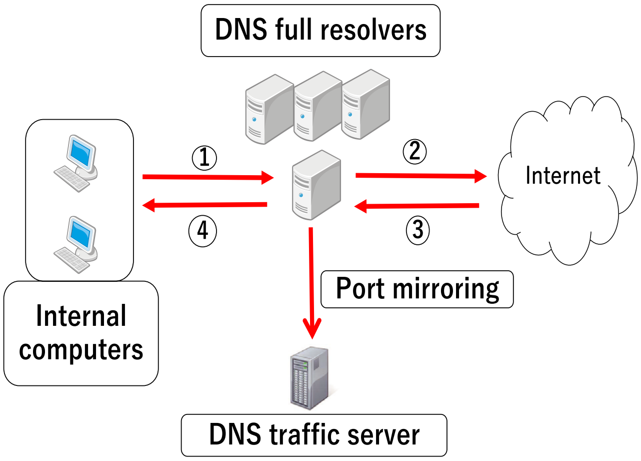


Fig. 4 Architecture to obtain via-resolver DNS TXT query data.

mate usages of DNS TXT record type must be distinguished from suspicious ones. In this section, we describe our analysis of via-resolver DNS TXT query data obtained from the DNS full resolvers of our campus network.

4.1 DNS Traffic Capturing and Analytical Methodology

Figure 4 depicts a simplified network topology of the DNS traffic capture from the DNS full resolvers set up in our campus network and the methodology that enabled us to obtain the DNS traffic. As shown in the topology, we only need to capture the DNS traffic launched by the DNS full resolvers (steps 2 and 3) and only store DNS TXT query type in the DNS traffic server. With such a network configuration, we captured and stored the DNS TXT query data for 99 days (from March 24, 2014 to June 30, 2014). Note that before the analysis we made the source IP addresses anonymous to respect the privacy of individual users. After we passively obtained the DNS TXT query data from the DNS full resolvers, we analyzed them through a two-step process:

1. Differentiate legitimate usages of the DNS TXT record type and filter out unconfirmed usages (Sect. 4.2)
2. Investigate the detection ratio calculated from the unconfirmed usages of DNS TXT record using a conventional method (Sect. 4.3)

4.2 Classify Usages of the DNS TXT Record Type

To identify legitimate usages of the DNS TXT record type and filter out unconfirmed usages, we statistically analyzed the obtained DNS traffic according to all published official usages of the DNS TXT record type. Table 1 lists some official usages of the DNS TXT record type as well as unconfirmed categories. These are limited to some typical examples we have found, so it is possible that new usages will appear with the introduction of new applications. Considering these possible usages of the DNS TXT record type specifically, we performed the following procedures to categorize the obtained DNS TXT query data.

Table 1 Usages of DNS TXT record.

Category	Basis	Usages
SPF	RFC7208	Formatted regular TXT record
DomainKeys	RFC6376	Formatted regular TXT record
DNSSD	RFC6763	Formatted regular TXT record
NFSv4	RFC 7530	-
Anti-virus	e.g., Sophos	Base64 encoded long TXT record
Spam and DNSBL	-	-
P2P tracker	-	
NTP	RFC5905	-
Misc	Various	Various long TXT record
Unconfirmed	Various	Base64 encoded long TXT record

- SPF (RFC4408) and DomainKeys (RFC4870): For this category, we filtered the responses for DNS TXT query which included “v=spf” and “domainkey” strings.
- DNS-based Service Discovery (RFC6763): For this category, we identified the DNS TXT queries which included “._dns-sd” in the FQDNs.
- NFSv4 (RFC3530): We identified and filtered out the DNS TXT queries which included “_nfsv4idmapdomain” in the FQDNs.
- Anti-Virus software: This category includes the update processes of anti-virus software which use the DNS TXT record type. We identified DNS TXT queries that included “.sophos.” [27], “immunet” [28], and AVG corporation’s domain [29] in the FQDNs.
- Spam email check and DNS black lists: This category includes specific domain names for spam email check and DNS black lists. We identified the DNS TXT queries which included “spamcop.net” [30], “spamhaus.org” [31], “rbl.maps.vix.com” and “sa-accredit.habeas.com” in the FQDNs.
- P2P tracker: This category represents P2P trackers used by BitTorrent. We identified the DNS TXT queries included “bttorent” and “tracker” strings in the FQDNs.
- NTP (RFC1305): Some corporations use the DNS TXT record type to obtain the IP addresses of NTP servers. We identified the DNS TXT queries which include “ntp minpool” and “time” in the FQDNs.
- Misc.: This category includes miscellaneous applications and campus internal communications. For miscellaneous applications, we identified the DNS TXT queries which include: “time.asia.apple.com”, “apple.com” (push notifications for mail deliveries by Apple), “planex.co.jp” (software updates for network devices manufactured by Planex Corporation), “gateway.com”, and “xmpp.org” [32] in the FQDNs. For internal communication, we identified the DNS TXT queries which included “titech.ac.jp” in the FQDNs.
- Unconfirmed: This is a group of usages of the DNS TXT record type which were not included in any of the above categories.

The last category, “Unconfirmed”, contains instances

Table 2 Usages and statistics of DNS TXT record.

Category	# of queries	Ratio [%]
SPF and domainkey	12,223	0.24
DNS-based service discovery	213,978	4.30
NFSv4	3,596,481	72.14
Anti-Virus	597,901	12.00
SPAM Check and DNS Blacklist	180,600	3.63
P2P Tracker	446	0.01
NTP	632	0.01
Misc	380,723	7.63
Unconfirmed	2,293	0.04
Total	4,985,277	100

that cannot be added into any other category. Those usages of the DNS TXT record type have not been announced officially by specific application vendors nor are they based on any standard protocols based on DNS. Therefore, these "Unconfirmed" usages possibly include suspicious information exchange, such as DNS-based botnet communication. The statistical results are shown in Table 2. Note that we only counted the DNS TXT queries that received responses since bot-infected computers need to exchange information with the C&C servers. Here, we call a query-and-response pair as a query having a response. Consequently, in our statistical analysis we obtained 2,293 "Unconfirmed" DNS TXT query-and-response pairs as suspicious.

4.3 Results of Via-Resolver DNS TXT Query Analysis

In Sect. 4.2, we investigated official usages, announced legitimate usages as well as unconfirmed usages of the DNS TXT record type and obtained 2,293 "Unconfirmed" usage of DNS TXT query from approximately five million query-and-response pairs of DNS TXT record. This means we can greatly reduce the number of query-and-response pairs required to be investigated from five millions to 2,293 through extracting only the unconfirmed usage. The next question is how is the detection ratio through such extractions of the unconfirmed usage. In this section, we therefore investigate the detection ratio calculated by the conventional method. The results are described in Table 3.

Before calculating the detection ratio, we first count the number of destination IP addresses in the DNS TXT queries. The number of total DNS TXT queries is of 4,985,277, which corresponds to 2,334 unique destination IP addresses, whereas that of unconfirmed TXT queries is of 2,293, which corresponds to 330 unique destination IP addresses, as shown in the column of "# of unique IP addresses" in Table 3. This equivalents that extractions of the unconfirmed usage reduced the number of destination IP addresses to 14.1%. However, if the extracted results do not include botnet communications, our method is not effective. To investigate the effectiveness, we use "Virustotal.com" [33], a free third-party security check web site. "Virustotal.com" provides a brief check of target IP addresses to see if they are involved in downloading suspicious files and hosting URLs to identify malware, infected files, malicious web sites, etc. Among them, "Searching for URL scan reports" and "Searching for IP address information" can be used for evaluation of our research. The former method, we call "URL detection" in this research, is based on the database of malicious URLs provided by URL scanners. And the latter method, we call "IP address detection" in this research, is based on the IP address database provided by antivirus solutions. Both detection methods can detect suspicious communications from the list of IP addresses. The experiment of "Virustotal.com" was performed at Dec. 12, 2016[†].

Table 3 shows the detection results of "Virustotal.com" by comparing with the "Unconfirmed" usages. From the results, first, we need to point out that the total detection ratio of our analysis (30.3%) is much higher than "Total" usages that do not use our analysis (9.77%). That is, among the 2,334 IP addresses in total, the "Virustotal.com" detected 228 IP addresses in URL or IP detection; and among the 330 IP addresses in Unconfirmed usages, 100 IP addresses were matched with the URL or IP detection. Note that "URL or IP detection" means the sum of "URL detection" and "IP address detection". This will give us the detection results with the widest coverage of suspicious IP addresses among detection methods we used. Details on each detection category can be referred to Table 3. Next, note that the cost effectiveness of the detection is much higher in our analysis. That is, in our analysis, we detected about 43.9% (100/228) of IP addresses which might involve in malicious communication only by investigating about 14.1% (330/2334) of that in the conventional method. Note that our method cannot detect 128 (= 228 − 100) IP addresses. This is because Virustotal.com is not specialized to detect only bot communications; there exist suspicious IP addresses not by bot communications. To detect such IP addresses, we have to use other methods by collaboration with our proposed method.

In all, in our analysis we extracted 330 unique destination IP addresses of DNS TXT record queries that might have been involved in malicious communication during the 99 days, which means an average of 3.3 IP addresses per day. Considering there are four DNS full resolvers set in our campus network and based on the network traffic conditions of our university, three or four times as many unique destination IP addresses will be detected as suspicious per day (about 13 IP addresses) from an organization with the same scale as that of our university. We consider this to be a reasonable number for handling by human operators.

5. Indirect/Direct Outbound DNS TXT Query Analysis

As described in Sect. 3.3, indirect and direct outbound DNS TXT queries are used for botnet communication in various botnets. To detect botnet communication using indirect and direct outbound DNS TXT queries, it is important to distinguish between legitimate and suspicious usages of the

[†] The results described in [1], [2] are different from this paper. This is because the experiment in [1], [2] was performed at Apr. 18, 2015, which is different date with this paper. Even if the same list of IP addresses sent to Virustotal.com, the statistical results will be changed if we tested at different date.

Table 3 “Virustotal.com” check results.

	DNS TXT queries	unique IP addresses	IP detection	URL detection	IP and URL	IP or URL	Ratio of IP or URL
Total	4,985,277	2,334	161	190	123	228	9.77% (228/2334)
Unconfirmed	2,293	330	55	87	42	100	30.3% (100/330)
Unconfirmed / Total	0.0460%	14.1%	-	-	-	43.9%	-

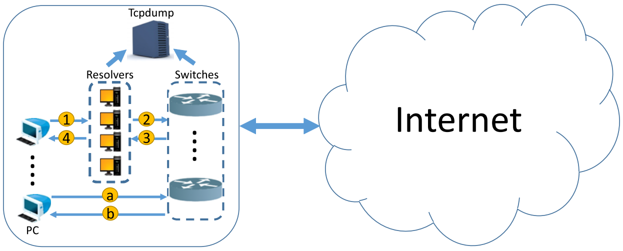


Fig. 5 System topology for DNS traffic capture.

DNS TXT record in indirect and direct outbound DNS TXT queries. In this section, we describe our analysis of indirect and direct outbound DNS TXT query data obtained from our campus network. Note that the query data we analyze in this section do not include the via-resolver DNS TXT query data.

5.1 DNS Traffic Capturing and Analytical Methodology

We obtained real DNS traffic from our campus network over a period of about three months (87 days) and analyzed all indirect and direct outbound DNS TXT query data. The system topology we used to obtain the DNS traffic is shown in Fig. 5. There are four DNS full resolvers set up in our campus network and we assume that some users use them while others send DNS queries to the outside DNS servers in the Internet directly. Thus, we captured all DNS traffic in the border routers and investigated the possibility of DNS TXT record usage in botnet communication. To respect the privacy of internal users, we made the IP address of the internal computers in the captured DNS packets anonymous in a one-to-one manner in order to trace the behavior in the corresponding name resolution (steps 1, 4, a and b in Fig. 5) after they had been made anonymous (except 2 and 3 due to they belong to the official full DNS resolvers) We captured the DNS traffic in our campus network from Nov. 1, 2014, to Jan. 26, 2015, and analyzed the DNS TXT query data through the following steps.

1. Capture all DNS traffic by mirroring from the border routers and filter out valid query-response pairs. Here, “valid” means that in a query-response pair the query is initialized from an internal computer and the corresponding response is returned.
2. Filter out NS records from the query-response pairs obtained in step 1 and map corresponding glue A records that can also be obtained from the same pairs or successive queries for the glue A record in case of out-of-bailiwick NS record [34]. After that, store them in a database called NS_DB.
3. Capture all DNS TXT queries sent directly from the internal computer to the outside and obtain their query destination IP addresses, and then check if NS_DB created in step 2 has these addresses.
4. If the destination IP address of a direct outbound DNS TXT query is in NS_DB, go to check the next record; otherwise, log it for further investigation.
5. Based on the check results of step 3, we create two unique destination IP address lists: one is for indirect outbound DNS TXT queries and the other is for direct outbound DNS TXT queries.
6. To confirm the results of our analysis, we also checked the IP addresses in the two IP address lists through a security check site, “Virustotal.com”.

In general, we believe that any direct outbound DNS query can be involved in some kind of malicious communication and it depends on the convenience of the DNS resource record type. Since use of the DNS TXT record type in botnet communication has been reported, however, we only analyzed DNS TXT record types that were involved in direct outbound DNS queries.

5.2 Results of Indirect/Direct DNS TXT Query Analysis

The number of unique destination IP addresses captured per day in indirect and direct outbound DNS TXT queries and the corresponding results from the “Virustotal.com” check are shown in Figs. 6 and 7, respectively. As shown in Fig. 6, the total number of unique destination IP addresses captured in direct outbound DNS TXT queries varies approximately from 15 to 120, and the average number of unique IP addresses per day is 62. On the other hand, as shown in Fig. 7, the total number of unique destination IP addresses in indirect outbound DNS TXT queries varies approximately from 160 to 305 and the average number of unique IP addresses per day is 235. These numbers may include duplicated IP addresses, though, since we only divided captured DNS traffic on a daily basis without considering the TTL-based cache in this analysis. Therefore, in actual operation the number of IP addresses that we would need to process is less than 300 per day, which is a reasonable number that will become smaller if we can remove the duplicated IP addresses.

Next, we checked the destination IP addresses captured from the indirect and direct outbound DNS TXT queries on “Virustotal.com”. As Fig. 6 shows, the hit rate of the IP addresses captured in direct outbound DNS TXT queries is comparatively stable at about 5 per day (the average hit rate is about 8%). This number includes the IP addresses that saw hits in URL-based detection or IP-based detection or both. This result indicates that for direct outbound DNS queries, we need to investigate in detail about 5 IP addresses per day, which seems not to be large for the network administrators.

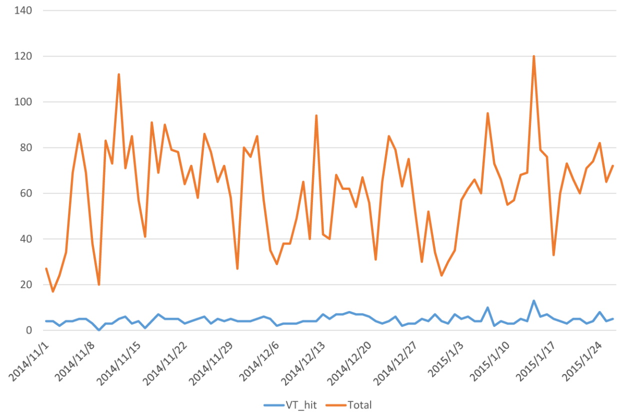


Fig. 6 VirusTotal results for direct outbound DNS query.

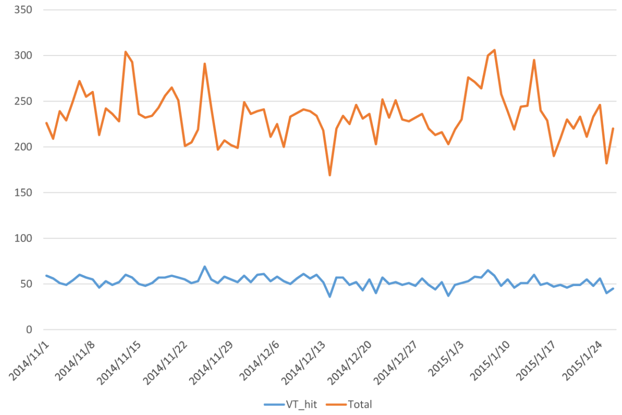


Fig. 7 VirusTotal results for indirect outbound DNS query.

On the other hand, the results for the indirect outbound DNS TXT queries are not so good, as we can see from Fig. 7. The average hit number of IP addresses per day reached about 53 (the average hit rate is about 22%), which means about 19% IP addresses in total had hits in “Virustotal.com” check. We also checked the destination IP address lists to see which public DNS resolvers were used in the campus network and found only Google public DNS resolvers were involved; i.e., 8.8.8.8 and 8.8.4.4. Since public DNS resolvers can also be used by bot-infected computers to search for the C&C servers belonging to indirect outbound DNS TXT queries, we have to include these as part of a target. Regular name resolution must retrieve the corresponding authoritative NS records and their IP addresses during its process. However, the destination IP address of a direct outbound DNS query cannot be traced in NS_DB, and in an indirect outbound DNS query it is not usual to use the DNS full resolver partially in one single name resolution process. Finally, in DNS-based botnet communication, even if an infected computer repeatedly sends a direct outbound DNS query to the same IP address (a C&C server), a method with NS_DB can successfully detect it since there is no valid NS record corresponding to the destination IP address in NS_DB.

Our results indicate that it is possible that the destination IP addresses captured in indirect and direct outbound DNS TXT queries may have been infected by some kind of malware, especially those in direct outbound DNS queries. This is because direct outbound DNS query requires hard coding of IP addresses for servers, which seem to be suspicious compared with indirect outbound one. Therefore, it is appropriate to detect botnet attacks in the early stage (while searching for C&C server and performing botnet communication) by monitoring indirect and direct outbound DNS TXT queries. Note that DNS protocol is continuously used in botnet communication after C&C server has been identified because it is easy to avoid being monitored. In this evaluation, we used “Virustotal.com” for further investigation and the main purpose of that was to confirm the effectiveness of our analytical method. Thus, in actual operation we can take action based just on the monitoring results. Considering over blocking and misdetection, we believe it can be mitigated by prerequisite leaning for legitimate IP address of name servers. In addition, like other security solutions, our analysis also has a zero-day vulnerability [35], but this is a challenge with respect to all anti-virus and security-related solutions. Therefore, we need to be very careful when creating policies for the destination IP addresses obtained in the indirect and direct outbound DNS TXT queries.

6. Consideration of Botnet Communication Detection

In Sect. 4, we investigated DNS TXT record type to find via-resolver-based botnet communications. In Sect. 5, we examined botnet communication based on either indirect or direct outbound DNS TXT queries. In this section, we discuss the possibility of a comprehensive DNS-based botnet detection system that would cover all the three query types (via-resolver and indirect/direct outbound DNS query) and look at the design of such a system. Via-resolver and indirect/direct outbound DNS queries based bot communications are different. We do not tend to compare the detection methods of via-resolver DNS query and indirect/direct DNS query. We tend to categorize legitimate usages of DNS TXT record type in via-resolver DNS queries and block the malicious destination IP addresses detected in indirect/direct outbound DNS queries. This system is based on botnet communication detection by monitoring DNS TXT queries. The basic idea of the system is as follows.

1. Via-resolver DNS TXT queries need to be checked to confirm legitimate usages.
2. All received NS (Name Server) information, including its FQDN and glue A record, needed in an organization must be stored in a database and updated periodically.
3. The destination IP addresses of all indirect and direct outbound DNS queries must be checked if they are included in the database.

Based on the above points, we introduce two databases into the system, as shown in Fig. 8. The first one (TXT_DB) is for storing information regarding legitimate usages of the DNS TXT record. The second one (NS_DB) is for registering valid NS information. These two databases are used to fil-

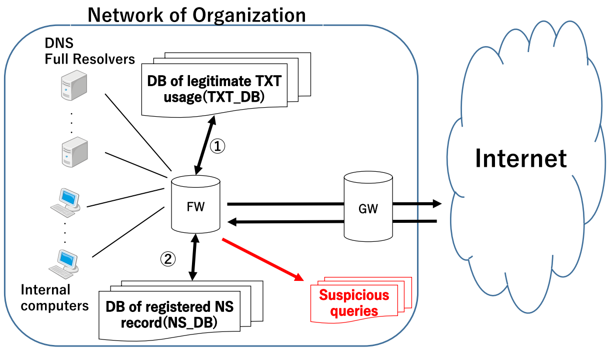


Fig. 8 Architecture of a DNS-based botnet detection system (covers via-resolver, and indirect and indirect outbound DNS TXT queries).

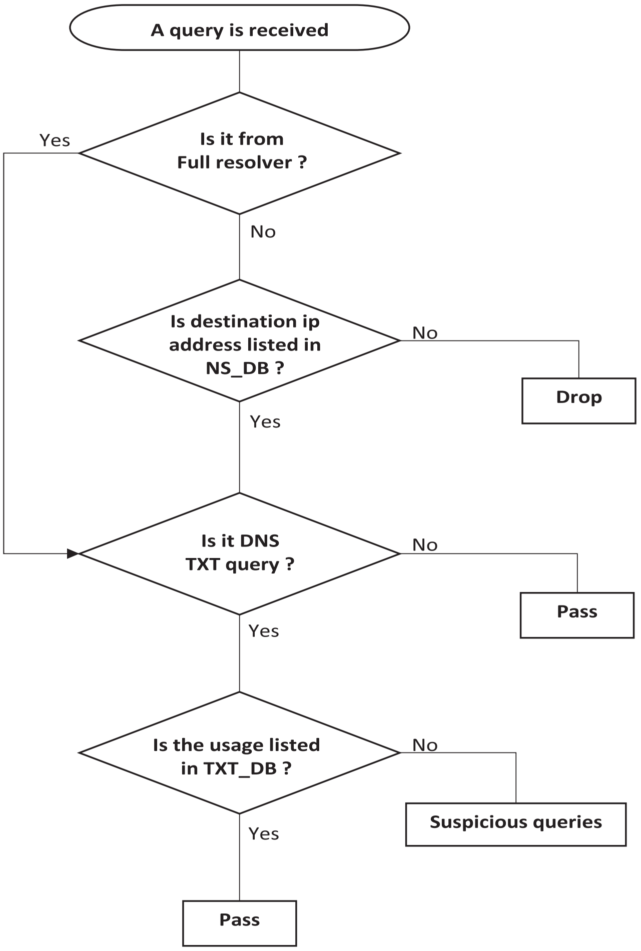


Fig. 9 Flow chart of comprehensive detection system.

ter unconfirmed DNS TXT queries and check indirect/direct outbound DNS TXT queries which can probably be used for detecting botnet communication.

The detection procedure of the comprehensive detection system works as shown in Fig. 9, which is a flow chart of the system. When a firewall (denoted by FW in the figure) detects a DNS packet, it first checks the source IP address of the packet. If the source IP address belongs to an official DNS full resolver and the record type is DNS TXT record, the system checks the usages of the DNS TXT record type in the TXT_DB. If the usage of the DNS TXT record type is listed in the TXT_DB then the DNS packet will be passed. On the other hand, if the usage of the DNS TXT query is not listed on the TXT_DB, information regarding the target DNS TXT query will be logged for further investigation.

Next, we consider the other case, when the source IP address does not belong to an official DNS full resolver, which is in the top branch of the flow chart. In this case, the system first checks whether the destination IP address of the DNS query is listed on NS_DB. Since direct outbound DNS queries without using an official DNS full resolver are unusual, we must drop the query if it is not listed on NS_DB. If the destination IP address of the DNS query is listed on NS_DB and the query is the DNS TXT record type, the system then checks TXT_DB. If the DNS TXT query is not listed on TXT_DB, we should log a suspicious queries entry similar to that for the DNS full resolver case. Note that for the special destination IP addresses such as official public DNS resolver, we will register them in the NS_DB in advance.

After the system has collected "suspicious queries," the network operators should investigate the IP addresses listed on the suspicious queries with collaboration of other security facilities to confirm the fact. Basically, we consider the collected queries are suspicious due to they use abnormal usages of DNS protocol. However, this botnet detection system also has some shortcomings. First, it can only detect botnet communication using the DNS TXT record type, which means we cannot find botnet communications using the A and/or CNAME record type. Second, the attacker may implement a DNS TXT-based application that is very similar to a "legitimate" usage, and this will make it more difficult to identify suspicious DNS traffic. Third, the system needs collaboration with other published security information and we may not able to detect highly suspicious botnet communication if there is no necessary information.

To overcome the first issue, we can extend the botnet detection system to support A and/or CNAME record types. For the second one, we need to find deeper characteristics of "true legitimate applications" that cannot be mimicked by attackers. For the last one, we need to develop an advanced method to find "highly suspicious" botnet communication — e.g., by using multiple third-party security check web sites in combination to ensure the necessary data is always available. Finally, we also need to consider about name resolution privacy. DNS over Transport Layer Security (TLS) [36] has been defined as a standard and we need to add a new feature to obtain DNS queries under its deployment. These solutions will be part of our future work.

7. Conclusion

Although the use of DNS TXT record type in botnet communication has been widely reported, the literature does not provide a comprehensive botnet detection system to prevent such botnet communication. Our goal in this research has been to detect three types of botnet communication based on

DNS TXT queries: through via-resolver as well as indirect and direct outbound DNS TXT queries. To deal with the via-resolver DNS TXT queries, we analyzed 99-day DNS traffic of TXT records obtained from DNS full resolvers of our campus network and categorized its usages. To deal with indirect and direct outbound DNS TXT queries, we analyzed 87-day DNS traffic of TXT records obtained from the border gateway of our campus network, which did not include the via-resolver DNS TXT query data.

In the first case, we significantly reduced the number of IP addresses needed to be investigated in detail (from 2,334 to 330) by extracting "Unconfirmed" usages of DNS TXT record. We then confirmed that among the "Unconfirmed" DNS TXT queries about 30.3% were identified as "highly suspicious" and it had about 43.9 % common detection rate with a conventional security check site "Virustotal.com.". In the latter case, through a similar analysis, we found that about 19% and 8% of destination IP addresses were identified as "highly suspicious" in indirect and direct outbound DNS TXT queries, respectively. Based on our analysis, we discussed the possibility of a comprehensive botnet detection system and proposed a design for such a system. Although this botnet detection system has shortcomings, we intend to find ways to overcome these in our future work.

References

[1] H. Ichise, Y. Jin, and K. Iida, "Analysis of via-resolver DNS TXT queries and detection possibility of botnet communications," Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM2015), pp.216–221, Aug. 2015. DOI: 10.1109/PACRIM.2015.7334837

[2] H. Ichise, Y. Jin, and K. Iida, "Analysis of via-resolver DNS TXT queries and detection possibility of botnet communications," IEICE Commun. Express, vol.5, no.3, pp.74–78, March 2016. DOI: 10.1587/comex.2015XBL0186

[3] Y. Jin, H. Ichise, and K. Iida, "Design of detecting botnet communication by monitoring direct outbound DNS queries," Proc. IEEE Int'l Confrerence on Cyber Security and Cloud Computing (CSCloud2015), pp.37–41, New York, NY, Nov. 2015. DOI: 10.1109/CSCloud.2015.53

[4] S. Khattak, N.R. Ramay, K.R. Khan, A.A. Syed, and S.A. Khayam, "A taxonomy of botnet behavior, detection, and defense," IEEE Commun. Surveys & Tuts., vol.12, no.2, pp.898–924, Oct. 2013. DOI: 10.1109/SURV.2013.091213.00134

[5] H. Binsalleeh, "Botnets: Analysis, detection, and mitigation," Network Security Technologies: Design and Applications, ed. A. Amine, O.A. Mohamed, and B. Benatallah, pp.204–223, IGI Global, Hershey, PA, Nov. 2013. DOI: 10.4018/978-1-4666-4789-3.ch012

[6] S. Soltani, S.A.H. Seno, M. Nezhadkamali, and R. Budiarto, "A survey on real world botnets and detection mechanisms," Int'l Journal of Information and Network Security, vol.3, no.2, pp.116–127, April 2014.

[7] McAfee, "McAfee labs threats report," http://www.mcafee.com/us/resources/reports/rp-quarterly-threats-mar-2016.pdf, March 2016.

[8] OpenDNS inc., "The role of DNS in botnet command and control," online http://info.opendns.com/rs/opendns/images/OpenDNS_SecurityWhitepaper-DNSRoleInBotnets.pdf, 2012.

[9] P. Mockapetris, "Domain names: Concepts and facilities," IETF RFC1034, Nov. 1987.

[10] P. Mockapetris, "Domain names: Implementation and specification," IETF RFC1035, Nov. 1987.

[11] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," Proc. IEEE Int'l Conference on Emerging Security Information, Systems and Technologies, pp.268–273, Glyfada, Greek, June 2009. DOI: 10.1109/SECURWARE.2009.48

[12] S. Bromberger, "DNS as a covert channel within protected networks," White paper of Department of Energy, http://energy.gov/oe/downloads/dns-covert-channel-within-protected-networks, Jan. 2011.

[13] N.M. Hands, B. Yang, and R.A. Hansen, "A study on botnets utilizing DNS," Proc. ACM Conference on Research in Information Technology (RIIT'15), pp.23–28, Chicago, IL, USA, Sept.-Oct. 2015. DOI: 10.1145/2808062.2808070

[14] K. Xu, P. Butler, S. Saha, and D. Yao, "DNS for massive-scale command and control," IEEE Trans. Dependable and Secure Computing, vol.10, no.3, pp.143–153, May/June 2013. DOI: 10.1109/TDSC.2013.10

[15] M. Anagnostopoulos, G. Kambourakis, and S. Gritzalis, "New facets of mobile botnet: Architecture and evaluation," Int'l Journal of Information Security, 19 pages, Dec. 2015. DOI: 10.1007/s10207-015-0310-0

[16] C.J. Dietrich, C. Rossow, F.C. Freiling, H. Bos, M. Steen, and N. Pohlmann, "On botnets that use DNS for command and control," Proc. IEEE European Conference on Computer Network Defence (EC2ND'11), pp.9–16, Gothenburg, Sweden, Sept. 2011. DOI: 10.1109/EC2ND.2011.16

[17] C. Mullaney, "Morto worm sets a (DNS) record," http://www.symantec.com/connect/blogs/morto-worm-sets-dns-record, Aug. 2011.

[18] D. Kaminsky, "The black ops of DNS," Presented in Black Hat USA 2004, Las Vegas, NV, July 2004. http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-kaminsky/bh-us-04-kaminsky.ppt

[19] AnalogBit, "tcp-over-dns," http://analogbit.com/software/tcp-over-dns, last accessed on July 6 2016.

[20] J. Riden, "How fast-flux service networks work," http://www.honeynet.org/node/132, Accessed on May 30 2016.

[21] G. Farnham, "Detecting DNS tunneling," White paper of SANS institute, https://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152, 32 pages, March 2013.

[22] A.M. Kara, H. Binsalleeh, M. Mannan, A. Youssef, and M. Debbabi, "Detection of malicious payload distribution channels in DNS," Proc. IEEE Int'l Conference on Communications (ICC2014), pp.853–858, Sydney, Australia, June 2014. DOI: 10.1109/ICC.2014.6883426

[23] S. Kitterman, "Sender policy framework (SPF) for Authorizing Use of Domains in Email, Version 1," IETF RFC7208, April 2014.

[24] D. Crocker, T. Hansen, and M. Kucherawy, "DomainKeys identified mail (DKIM) signatures," IETF RFC6376, Sept. 2011.

[25] J. Damas and P. Vixie, "Extension mechanisms for DNS (EDNS0)," IETF RFC6891, April 2013.

[26] Google, "Introduction to Google public DNS," https://developers.google.com/speed/public-dns/docs/intro, Accessed on May 30 2016.

[27] Sophos Ltd., "Overview of the Sophos live protection architecture in SESC 9.5+," https://www.sophos.com/en-us/support/knowledgebase/111334.aspx, Accessed on May 30 2016.

[28] Cisco, "Immunet 3.0," http://www.immunet.com/, Accessed on May 30 2016.

[29] AVG, "AVG internet security 2015," http://www.avg.com/us-en/internet-security, Accessed on May 30 2016.

[30] SpamCop, "How can I check if an IP is on the list?" https://www.spamcop.net/fom-serve/cache/351.html, Accessed on May 30 2016.

[31] Spamhaus project, "Spamhaus," https://www.spamhaus.org, Accessed on May 30 2016.

[32] J. Hildebrand, P. Saint-Andre, and L. Stout, "XEP-0156: Discovering alternative XMPP connection methods," http://xmpp.org/extensions/xep-0156.html, Accessed on May 30 2016.

[33] Google, "Virustotal," https://www.virustotal.com/en/, Accessed on May 30 2016.

[34] W. Hardaker, "Child-to-parent synchronization in DNS," IETF RFC7477, March 2015.

[35] Symantec, Inc., “What is a zero-day vulnerability?,” http://www.pctools.com/security-news/zero-day-vulnerability/, Accessed on May 30 2016.

[36] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, P. Hoffman, “Specification for DNS over transport layer security (TLS),” IETF RFC7858, May 2016.

Hikaru Ichise received the B.S. degree in mathematics from Kwansei Gakuin University, Sanda, Japan in 2008. Currently, he is a technical staff in Tokyo Institue of Technology, Tokyo, Japan. His research interest is of detecting botnet communciations using DNS protocol.

Yong Jin received his M.E. degree in electronic and information systems engineering and Ph.D. degree in Industrial Innovation Sciences from Okayama University, Japan in 2009 and 2012, respectively. In April 2012, he joined National Institute of Information and Communications Technology, Japan, as a researcher. From October 2013, he joined the Global Scientific Information and Computing Center of Tokyo Institute of Technology as an assistant professor. His research interests include network architecture, traffic engineering and Internet technology. He is a member of IEICE.

Katsuyoshi Iida received B.E., M.E. and Ph.D. degrees in, respectively, Computer Science and Systems Engineering from Kyushu Institute of Technology (KIT), Iizuka, Japan in 1996, Information Science from Nara Institute of Science and Technology (NAIST), Ikoma, Japan in 1998, and Computer Science and Systems Engineering from KIT in 2001. Currently, he is an Associate Professor in the Information Initiative Center, Hokkaido University, Sapporo,, Japan. His research interests include network systems engineering such as network architecture, performance evaluation, QoS, and mobile networks. He is a member of the WIDE project and IEEE. He received the 18th TELECOM System Technology Award and the Tokyo Tech Young Researcher’s Award in 2003 and 2010, respectively.