# Fast and Accurate Identification of Active Recursive Domain Name Servers in high-speed Network

Xiaomei Liu
Institute of Information Engineering, Chinese Academy of Sciences
No.91 Minzhuang Road
Beijing, China
liuxiaomei@iie.ac.cn

Yong Sun
Institute of Information Engineering, Chinese Academy of Sciences
No.91 Minzhuang Road
Beijing, China
sunyong@iie.ac.cn

Caiyun Huang
Institute of Information Engineering, Chinese Academy of Sciences
No.91 Minzhuang Road
Beijing, China
huangcaiyun@iie.ac.cn

Xueqiang Zou
National Computer Network Emergency Response Technical Team/Coordination of China
NO.3 Middle Yumin Road
Beijing, China
zouxueqiang@iie.ac.cn

Zhiguang Qin
University of Electronic Science and Technology of China
No.4,North Jianshe Road
Chengdu, China
qinzg@uestc.edu.cn

## ABSTRACT

Fast and accurate identification of active recursive domain name servers (RDNS) is a fundamental step to evaluate security risk degrees of DNS systems. Much identification work have been proposed based on network traffic measurement technology. Even though identifying RDNS accurately, they waste huge network resources, and fail to obtain host activity and distinguish between direct and indirect RDNS. In this paper, we proposed an approach to identify direct and forward RDNS based on our three key insights on their request-response behaviors, and proposed an approach to identify indirect RDNS based on CNAME redirect behaviors. To work in high-speed backbone networks, we further proposed an online connectivity estimation algorithm to obtain estimated values used in our identification approaches. According to our experiments, we can identify RDNS with a high accuracy by selecting the reasonable thresholds. The accuracy of identifying direct and forward RDNS can reach 89%. The accuracy of identifying indirect RDNS can reach 90%. Moreover, our work is capable of real-time analyzing high speed backbone traffics.

## Keywords

evaluate security risk degrees, recursive nameservers, connectivity estimation



*WTMC'16, May 30 2016, Xi'an, China*



DOI: http://dx.doi.org/10.1145/2903185.2903190

## 1. INTRODUCTION

Domain Name System (DNS) is the basic infrastructure of the Internet.It maps domains to IP addresses through nameservers, including root nameservers (Root-DNS), authoritative nameservers (ADNS) and recursive nameservers (RDNS).RDNS plays important role in the DNS because it connects clients and DNS directly. It can be classified into three different types: forward RDNS (FRDNS), direct DNS (DRDNS) and indirect DNS (IRDNS). As the relationship between RDNS and clients are one-to-many, it may cause serious network failures and make it impossible to access to the Internet normally when RDNS is crashed. More clients RDNS provides services for, greater losses will be made. FRDNS and DRDNS used by clients cannot change constantly because they allocated by network operator initally. However, IRDNS used by FRDNS or DRDNS can change constantly by using load balancing technology. Crashes on IRDNS has smaller influences than those on other two type of RDNS.

It is of great significance to evaluate the security level of RDNS for maintaining the safety of DNS. In the specific network environment, we can assess security levels of RDNS by their activity or linking relation with clients. An accurate and timely identification of active RDNS can provide a good foundation for their security assessments. According to whether to construct a specific domain request or not, there are two kinds of RDNS identification methods: active identification and passive identification.

Active identification approaches achieve the goal by analyzing responses of well-constructed domain requests [1]. Though achieving high identification rate[1,2], they need a large IP address list and proxies. Moreover, security measures on RDNS such as firewall may filter out domain request with random host prefixes. In addition, active approaches can only get IP addresses of RDNS and cannot get activity

[1]http://openresolverproject.org/
[2]https://dnsscan.shadowserver.org/

of RDNS. Active approaches are also unable to separate the DRDNS and FRDNS from IRDNS. Therefore, they cannot be used to assess security degrees adequately.

Passive approaches identify RDNS according to some distinguishable features in network traffic. This approach can overcome shortcomings of active methods effectively. For example, Cranor *et al.* identify active RDNS by constructing a offline DNS traffic graph and analyzing patterns of nodes in the graph [2]. However, it takes huge transmission bandwidth and storage spaces. In addition, it cannot reflect network status and evaluate security degrees of active RDNS timely. How to identify active RDNS accurately and timely from online DNS traffic is an open question that should be addressed urgently. However, online traffic analysis faces some great challenges because of high concurrency and bandwidth in high-speed networks.

In this paper, we propose a passive online RDNS identification approach based on connectivity estimation and C-NAME redirect behavior. Our contributions are as follows:

- We design and implement an online RDNS identification framework.
- We get three related features for identifying direct and forward RDNS from the DNS traffic: host connectivity, domain connectivity and host frequency. We proposed an online analysis method to identify indirect RDNS based on CNAME redirect behavior.
- We implement an effective connectivity estimation algorithm for calculating features online and select a reasonable threshold to identify active direct and forward recursive RDNS timely and accurately.
- We deploy online RDNS identification framework in China Unicom gateway (CUG) of a region. The bandwidth of CUG is about 5Gbps. We monitor reginal China Unicom network and identify active RDNS online.We use active approaches to verify identification results. By sending recursive requests of legal domains actively to identification RDNS and analyzing corresponding responses, we can get the recognition accuracy. High recognition accuracy has been proved by lots of experiments: direct and forward RDNS recognition accuracy can reach 89%, indirect RDNS recognition accuracy can reach 90%.

# 2. RELATED WORKS

## 2.1 DNS Background Knowledge

We first give the core name servers in DNS, and then show the DNS query process. Root-DNS is the top-level name severs which are responsible for returning the addresses of authoritative DNS (ADNS) of top-level domains.

ADNS is a name server that gives answers in response to questions asked about names in a zone. An authoritative-only name server returns answers only to queries about domain names that have been specifically configured by the administrator.

RDNS (Recursive DNS) sends a domain query to ADNS instead of users and responds responses that are from ADNS to users. Specifically, there are three types of RDNS, namely forward RDNS (FRDNS), direct DNS (DRDNS) and indirect DNS (IRDNS)[3]. FRDNS forwards domain queries

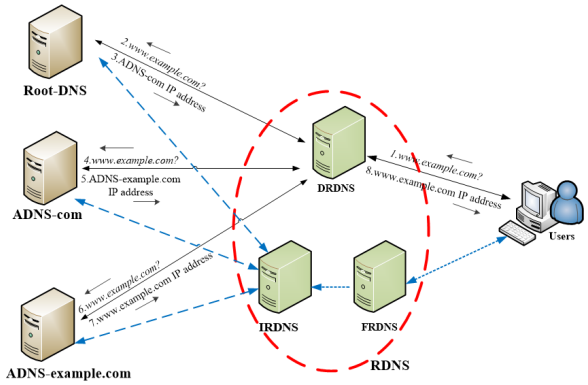


**Figure 1: the work process of querying domain www.example.com**

from users to IRDNS without parsing requests. DRDNS receives domain queries from users directly and sends them to ADNS instead of users. IRDNS indirectly receives domain queries from FRDNS and send them to ADNS instead of FRDNS.

Figure 1 shows the query process using `www.example.com` as an example.

1. Users send a domain query of `www.example.com` to DRDNS or FRDNS (which forwards the query to IRDNS).
2. DRDNS (IRDNS) sends the query to Root-DNS if they have no query result in their caches.
3. Root-DNS responds the IP address of ADNS-`com` to DRDNS (IRDNS).
4. DRDNS (IRDNS) sends the query to ADNS-`com`.
5. The IP address of ADNS-`example.com` is responed to DRDNS (IRDNS).
6. DRDNS (IRDNS) sends the domain request to ADNS-`example.com`.
7. ADNS-`example.com` responds the IP address of domain `www.example.com` to DRDNS (IRDNS).
8. DRDNS (IRDNS) caches this response. If users send this request to DRDNS directly, DRDNS responds the IP address to users directly. If users send this request to FRDNS directly, IRDNS responds the IP address to FRDNS, then FRDNS responds the IP address to users.

## 2.2 Identification of RDNS

Prior work on identifying whether a IP address is a RDNS or not can classified into two categories: active approaches and passive approaches. Prior active approaches can be further divided into two categories. (1) Query Scanning: sending some domain requests to the pre-defined IP addresses and making a determination based on the responses. (2) ADNS Listening: as RDNS only interacts with ADNS directly, any IP address that sends domain requests to ADNS can be identified as a RDNS. By deploying a ADNS in advance [1, 5, 6], D. Dagon *et al.* collect the IP addresses that send domain requests with random hosts in the same secondary domain to the ADNS. Domain requests with random hosts in the same secondary domain are sent by PlantLab nodes [7]. Active approaches have the following three limitations: First, they need a large pre-defined IP addresses list;

Second, they waste a lot of network resources detecting the IP addresses of no responses because many RDNS firewall

may filter out domain requests of random hosts in a secondary domain; Third, they can only identify a open RDNS while not get its activity, and cannot get linking relations between RDNS and clients, as a result they cannot help us to assess RDNS security degrees adequately.

To overcome these limitations, some passive approaches are proposed by analyzing related features of RDNS in DNS traffic. Prior passive approaches identify RDNS by analyzing offline DNS traffics [2]. However, storing offline traffics consumes large storage space. What is worse, analyzing offline DNS traffics cannot reflect activity of RDNS and cannot evaluate security degree of RDNS in time.

Network traffic measurement (NTM) technology is the core to make online traffic analysis.

There are two major NTM approaches: sampling and data streams. Sampling approaches include packet sampling and flow sampling. Packet sampling approaches can be further divided into systematic sampling, random sampling and stratified sampling [8]. He *et al.* reduced the system overhead of systematic sampling by using the features of self-similarity in the traffic. But they don't take the various sizes of packets into account [9]. Based on the efficient byte sampling, Raspall [10] proposed an improved method which made the measuring accuracy more independent on the flow characteristics. However, the packet size is restricted by transmission technology. In order to eliminate this restriction, researchers presented the flow sampling [11]. Even the flow sampling method can reflect plenty of traffic features, it still work on partial traffic that may lead to a result deviation.

Aim at obtaining more accurate original flow characteristics, researchers proposed the data flow technology. Data flow technology uses limited computation and memory resources to calculate network flow only once. It is an important method for measuring high-speed network traffic and is widely used to approximately measure statistics of traffic in high-speed link such as entropy estimation and connectivity estimation. Based on the traditional sampling technology, Zhao *et al.* proposed a packets storage method using the Bitmap [12]. their work decreased the memory consumption while increased the connectivity estimation accuracy. To further decrease the memory consumption, Li *et al.* presented a connectivity estimation algorithm based on optimal dynamic bit sharing [13]. In their scheme, a single bit in the Bitmap is shared by multiple hosts rather than only a single host. However, the multiple sharing mechanism in Bitmap may cause the host mapping conflicts. Yoon *et al.* solved the problem by setting up a visual vector for every host [14].

# 3. IDENTIFICATION FRAMEWORK

As shown in Figure 2, our online RDNS identification architecture consists of four modules: DNS traffic capture module, DNS traffic parsing module, Identification features calculating module and CNAME chains analysis module.

**DNS traffic capture module**: Using network devices to capture network traffic. It gets DNS traffic from the captured network traffic through analyzing protocol type and port number. Then it output the DNS traffic to DNS traffic parsing module.

**DNS traffic parsing module**: Parsing the basic attributes that include packets' type, the source IP address, the destination IP address, domain and CNAME etc. These attributes are closely related to the behavior of RDNS and can be used to analyze RDNS identification features.

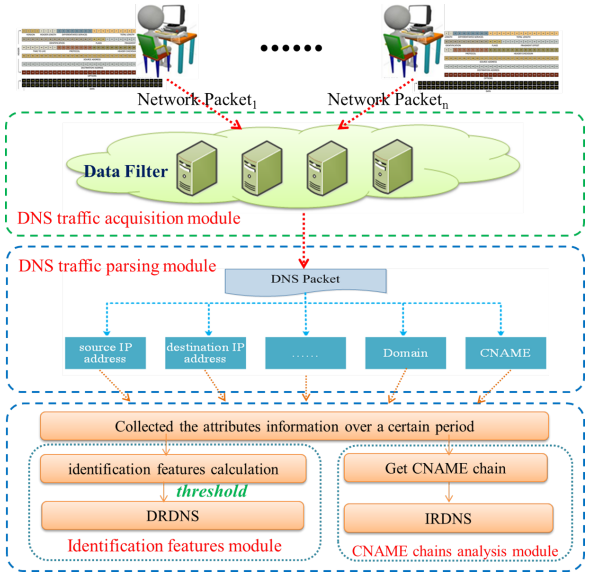


**Figure 2: Online RDNS identification framework**

**Identification features calculating module**: Calculating identification features in a certain time period. Firstly, we get DRDNS identification features by calculation method which is efficient and suitable for high speed backbone traffic. Secondly, we analyze and select the thresholds of identification features for identifying the active DRDNS.

**CNAME chains analysis module**: Through analyzing CNAME redirect behavior to get the IRDNS list.

# 4. MINING IDENTIFICATION FEATURES

To identify RDNS precisely, some unique and representative features from DNS traffic should be selected. This section presents a identification feature mining method by analyzing several datasets from Network Operator (NO) and Multiple Internet Gateway Routers (MIGRs) of CUG.

## 4.1 Data Acquision

We collect thousands of known ADNS IP addresses (for `.com`, `.org`, `.net` and `.edu` *etc.*) and RDNS IP addresses that communicate with clients directly in the NO dataset. The MIGR dataset consists of captured network traffics lasting 24 hours on port 53. The dataset is more than 50 TB, and has 1 billion packets in total.We collect this part of data from two sources. First, we collect a DNS dataset from 9:00 am to 9:00 pm which is named as *Data1*. In order to decrease the calculation error and improve the feature selection accuracy, we captured DNS traffic from another consecutive twelve hours which is defined as *Data2*. While DNS traffic provides us with valuable insights into DNS traffic features, it has several limitations. First, the gateway where we capture traffics does not guarantee having bidirectional DNS traffics. For example, we might see requests but not responses and vice versa. Second, the source and destination ports in DNS records may not be privileged port 53. The newer versions of bind use unprivileged ports that are larger than 1023.

DNS Response packets are closer related to DNS servers because source IP addresses of response packets are DNS

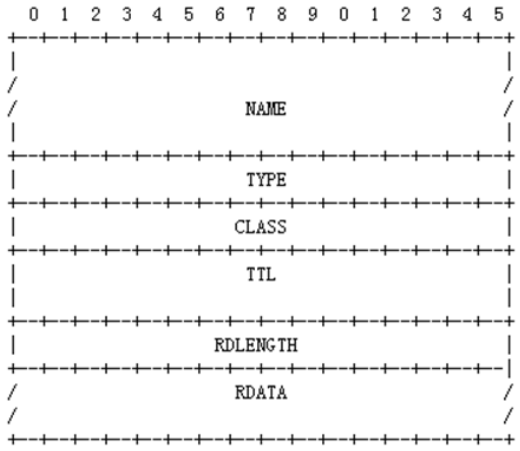


**Figure 3: DNS Response Records Format**

servers definitely. Therefore, we may get some obvious features for identifying RDNS by analyzing features of response packets.

RDNS consists of three sub-roles named DRDNS, FRDNS and IRDNS. Thus, we can select adequate features to identify DRDNS and FRDNS by analyzing above-mentioned datasets. However, we cannot get the known IRDNS list, so we take another way to identify IRDNS.

## 4.2 Mining Features of DRDNS and FRDNS

As shown in Figure 3, DNS response record is constructed with several segments such as Domain Name (DN), Query Type (QT), Record Time to Live (TTL) and Record Length (RL) and so on.

- Domain Name (**DN**, Indefinite length). The name of resources records is the same as the query name.
- Query Type (**QT**, Two bytes). The domain query types include A (IPv4 address), AAAA (IPv6 address), PTR (Reverse address resolution),CNAME ( alias records) and so on.
- Record Time to Live (**TTL**, Four bytes). The time to live of resources records in cache measured by seconds.
- Record Length (**RL**, Two bytes). The length of resource record is measured by bytes.

Besides, as Record Frequency (RF) and Destination IP (DIP) are also the basic attributes of the DNS information, we take them into consideration in our features selection method.

- **Record Frequency (RF):** The occurrence of the source IP address (DNS servers) exists in the response packets.
- **Destination IP (DIP):** In a DNS response packet, the source IP address has a corresponding destination IP address.

Because the IP addresses of Root-DNS are open knowledge, so we just need to analyze the difference between ADNS and RDNS in traffic. By analyzing the features of DNS information based on the known RDNS and ADNS list, we can get features to identify DRDNS and FRDNS. We mainly concern the accuracy of identification results and do not consider the recalling rate of identification results because of incomplete traffic. The results of the basic DNS traffic features analysis are shown below.

**DN:** the feature selection of DN is illustrated by analyzing the distribution of different domain names numbers. Figure 4 shows that the statistics of the percentages of DNS servers vary depending on the amount ranges of their corresponding different domain name. When different domain name number is larger than 100, the total percentages of known D/F RDNS are over 26.0% while the total percentages of known ADNS almost equal to 0.00%. Besides, the average number of different domain name for D/F RDNS and ADNS can help us to decide whether the DN is a reasonable feature. The average number of D/F RDNS and ADNS are 3284 and 25 in *Data1* while the numbers are 3436 and 28 in *Data2*. Thus, the average different domain name numbers of known D/F RDNS is much larger than the known ADNS. Moreover, the larger number range is, the higher total percentages of the known D/F RDNS have than the known ADNS. This leads us to a conclusion that large different domain name number can be used to distinguish D/F RDNS from ADNS. The average number can be used as a identification threshold.

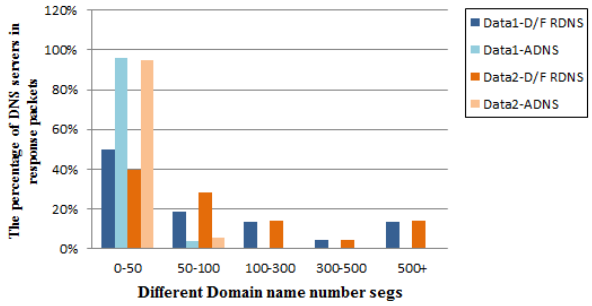


**Figure 4: Different Domain Name Amount Distribution of DNS Servers**

**QT:** the feature selection of DN is illustrated by analyzing the distribution of query type. The distribution of common query types of the DNS servers is shown in Figure 5. Each bar in the graph shows the percentage of DNS servers in different query type. In addition to type A and type CNAME, the average percentage gap of other query types between ADNS and D/F RDNS is quite small (less than 2%). The percentage of type A of ADNS is much larger than D/F RDNS. However, the percentage of type CNAME of ADNS is smaller than D/F RDNS. Because several network applications use CDN to accelerate their services, so D/F RDNS interacted with the clients directly have the high proportion of type CNAME. At the same time, the percentage of type CNAME of ADNS at around 20.0%. We cannot use type CNAME to identify D/F RDNS by ignoring this higher proportion of ADNS. We cannot identify D/F RDNS through analyzing the distribution of query type.

**TTL:** The TTL distribution is an important feature of DNS traffic. As shown in Figure 6, the percentage of DNS servers vary depending on the different TTL range segs in *Data1* and *Data2*. The maximum average percentage gap of TTL between ADNS and D/F RDNS is located in interval [0, 300] (which is nearly 17%). Meanwhile, the average

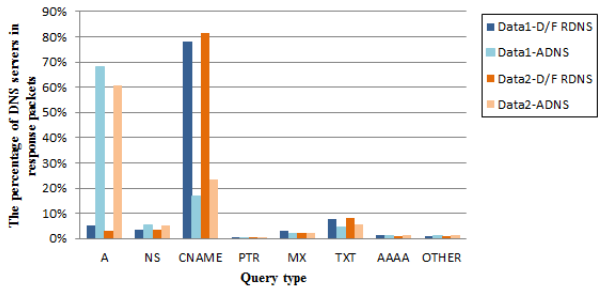


**Figure 5: Query Type Distribution of DNS Servers**

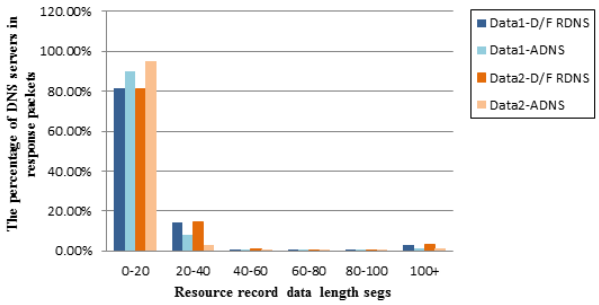


**Figure 7: Record Data Length Section Distribution of DNS Servers**

percentage gap in interval [300, 3600] and interval [10800, 86400] are all over 10%. In interval [0, 300], the percentage of the known ADNS is larger than the known D/F RDNS. The percentage gap between ADNS and D/F RDNS is little more than the percentage of D/F RDNS. While in intervals [300, 3600] and [10800, 86400], the percentage of known ADNS is smaller than that of known D/F RDNS. The percentage of ADNS is larger than the percentage gap between ADNS and D/F RDNS. Thus, there is no obvious gap between ADNS and D/F RDNS. Note that ADNS and D/F RDNS in other TTL ranges have similar percentages. So we cannot separate D/F RDNS with ADNS by using a reasonable range of TTL.

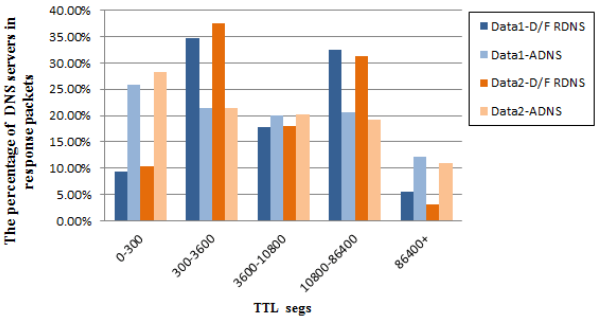


**Figure 6: TTL Range Section Distribution of DNS Servers**

**RL:** Another important feature of the DNS response packet is RL. The distribution of RL is closely related to DNS servers. Figure 7 shows the percentage of DNS servers in the different record length range. Servers with short record data length take the majority percentages.In range of [0, 20], both ADNS and D/F RDNS are accounted for the largest proportion (which is more than 80%). In bigger ranges, the distributions are similar. The average percentage difference between D/F RDNS and ADNS is less than 2%.We cannot see obvious differences between RL of RDNS and ADNS. Thus, we cannot identify D/F RDNS by selecting a reasonable record length range.

**RF:** In the described DNS information, we have mentioned the RF that the occurrences of the source IP address exists in the response packets. The frequency distribution of DNS servers is illustrated in Figure 8. In range of [0, 100], the percentage of ADNS is almost 100% and the percentage of D/F RDNS can reach 65%. In other ranges, the total percentage of D/F RDNS is up to 35% and far more than it in ADNS. When the frequency is larger than 100, the percentages of D/F RDNS in the two datasets have similar distribution. For example, in [100, 300] and [900, $\infty$], the percentages of D/F RDNS are more than 10%. The higher the frequency, the greater percentage of D/F RDNS is than ADNS. At the same time, the average frequency of D/F RDNS is 10430 in *Data1* and 10907 in *Data2* while the average frequency of ADNS is 26 and 29 respectively. Several D/F RDNS has larger record frequency than ADNS. Thus, we can identify active D/F RDNS from DNS servers by higher frequency.

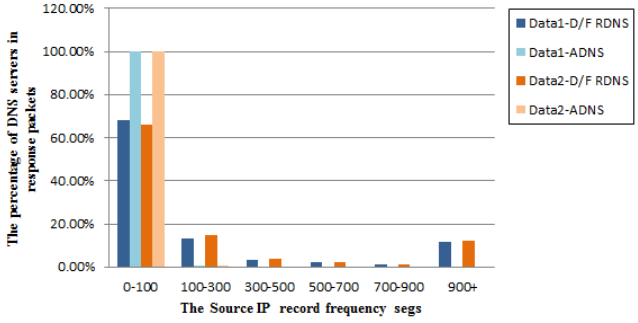


**Figure 8: Record Frequency Distribution of DNS Servers**

**DIP:** A better look at destination IP addresses inside DNS packets can be shown by analyzing the distribution of different destination IP addresses. The result is shown in Figure 9. When the different destination IP addresses number of DNS servers is more than 40, the percentage of D/F RDNS is far larger than ADNS. Moreover, in range [40, 60], the average percentage of ADNS is fairly small (which is less than 4%).When the different destination IP address number is more than 60, there is only D/F RDNS. In the higher amount, D/F RDNS occupy a higher proportion than ADNS. The average value of D/F RDNS is 171 in *Data1* and 193 in *Data2* while the average value of ADNS is 14 and 17 respectively. Several D/F RDNS has larger amount of destination IP addresses than ADNS. Thus, we can select suitable number of different DIP to separate D/F RDNS with DNS servers.

Based on the analysis results presented in this section we can now draw conclusions about the identification features of D/F RDNS. **DN**, **DIP** and **RF** in the DNS information

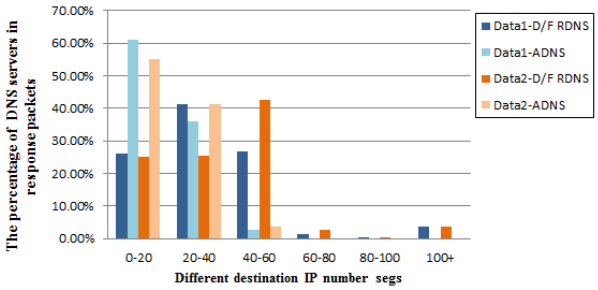


**Figure 9: Different Destination IP Amount Distribution of DNS Servers**

can be used to identify D/F RDNS. We can define specific identification features based on the three attributes.

- **dom-con:** This feature is related to DN. dom-con is the amount of different domains interacted with the source IP address in DNS response packets in a certain period. High dom-con that may be average amount can be used to identify D/F RDNS based on the above analysis.
- **src-con:** This feature is related to DIP. src-con is the amount of different destination IP addresses interacted with the source IP address in DNS response packets in a certain period. High src-con that may be average amount can be used to identify D/F RDNS based on the above analysis.
- **src-count:** This feature is related to RF. src-count is the frequency of source IP address in DNS response packets in a certain period. High src-count that may be average amount can be used to identify D/F RDNS based on the above analysis.

To get the active D/F RDNS lists more accurately, we integrated the three features for identification.

## 4.3 IRDNS Identification

FRDNS may choose different IRDNS to parse CNAME responses even for the same DNS request[15]. Thus we could identify IRDNS by analyzing redirect behavior of CNAME in network traffic. The process of CNAME redirect behavior is shown in Figure 10.

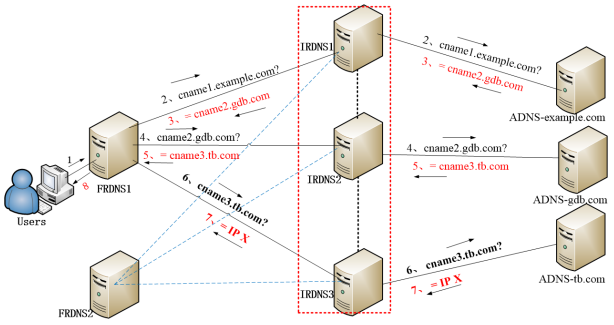


**Figure 10: Process of CNAME redirect behavior**

1. Users send the domain request (`cname1.example.com`) to FRDNS1.
2. FRDNS1 forwards the request to IRDNS1, then IRDNS1 sends the request to ADNS-`example.com`.
3. IRDNS1 receives a CNAME response (`cname2.gdb.com`) from ADNS-`example.com`, then IRDNS1 returns this response to FRDNS1.
4. FRDNS1 sends the CNAME request (`cname2.gdb.com`) to IRDNS2. IRDNS2 sends the CNAME request to ADNS-`gdb.com`.
5. IRDNS2 receives a CNAME response (`cname3.tb.com`) from ADNS-`gdb.com`, then IRDNS2 returns this response to FRDNS1.
6. FRDNS1 sends the CNAME request (`cname3.tb.com`) to IRDNS3. IRDNS3 sends the CNAME request to ADNS-`tb.com`.
7. ADNS-`tb.com` responds the IP X to IRDNS3, then IRDNS3 returns the the IP X to FRDNS1. At last, FRDNS1 returns this IP X to users. IRDNS1, IRDNS2 and IRDNS3 are associated through the CNAME redirect chain.

Our IRDNS identification is based on the CNAME redirect behaviors, as shown in Figure 11.

1. We acquire DNS traffic from network traffic through analyzing network protocol online.
2. We can get two types of DNS packets which are defined as response packets (RES for short) and request packets (REQ for short) by analyzing DNS traffic. If there is a CNAME response in T1, then we can keep a record of the source IP address $RES-SRC$ and the destination IP address $RES-DST$ and CNAME.
3. If there is a DNS request in T2, then we can keep a record of the source IP address $REQ-SRC$ and the destination IP address $REQ-DST$ and Domain.
4. We can compare and analyze these attributes got from REQ in [T1, T1+timeval] with the CNAME record in T1. If CNAME is equal to $Domain_i$ and $RES-DST$ is equal to $REQ-SRC_j$ and $REQ-DST_j$ is not equal to $RES-SRC$, then we save $RES-SRC$ and $REQ-DST_j$.
5. $RES-SRC$ and $REQ-DST_j$ saved are IRDNS that we want.

## 4.4 Features Threshold Analysis

We capture DNS traffic as the test dataset from China Unicom gateway whose bandwidth is 5 Gbps for 12h. We use this dataset to verify the accuracy of identification features of RDNS.Firstly, we calculate the identification features of src-con, dom-con and src-count for every source IP address in response packets with accurate connectivity measurement.Secondly, we take different thresholds referred to the average amount in Section 4.2 for each feature. Through sending legal domain requests to the RDNS identified under different thresholds actively, we verify the accuracy of identification results. This active verification method is reasonable because 1) the domain request is legal request such as the search engine's domain; 2) the number of detection IP

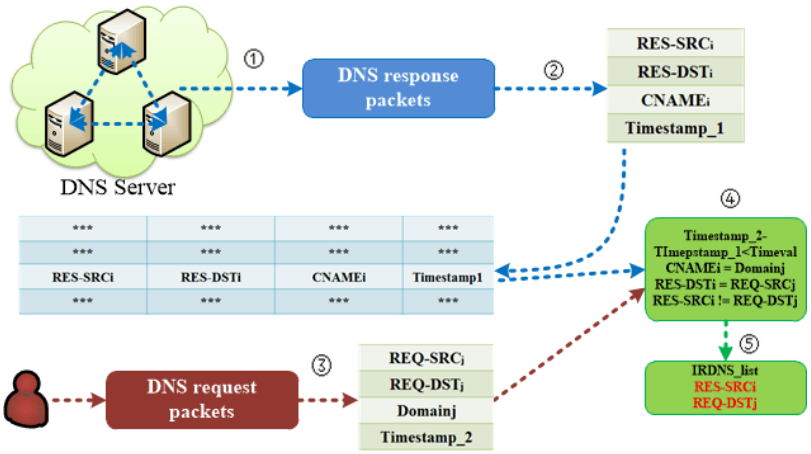


**Figure 11: Process of IRDNS identification**

addresses which are generated by programs automatically is limited.

As shown in Table 1, after a lot of experiments, we select representative thresholds and the corresponding identification results of DRDNS or FRDNS. It shows the identification accuracy of DRDNS or FRDNS under different thresholds of src-con = 100, 200, 300 dom-con = 4000, 5000, 6000 and src-count = 7000, 8000 respectively. With reasonable threshold for each feature, the accuracy of identifying FRDNS and DRDNS can reach 90%. The number of identification IRDNS is 9261. The number of verified IRDNS is 8523. By analyzing CNAME redirect chain, the accuracy of identifying IRDNS can reach 92%.

There are two factors influencing accuracy: 1) Several RDNS and their clients may behind a NAT, which exposes a common IP to the public Internet. Then the switched common IP address may have higher features. We can exclude the common IP address from identification results. 2) Because we cannot get the whole network traffic, so we cannot get the whole response packets for partial RDNS in measuring point.

## 5. CONNECTIVITY ESTIMATION MODEL

In order to identify active FRDNS or DRDNS in CUG backbone network whose bandwidth is 5Gbps, we need to calculate basic features for each source IP address. These calculation may take tremendous memory and computing resources consumption because of a large amount of traffic. For instance, when we calculate Src-con for each source, we should storage all different destinations for each source. Therefore we need a connectivity calculation method with high computational efficiency and low memory consumption. As it will cost less computing and memory resources, connectivity estimation model[14] is more suitable for analyzing backbone traffic online.

Then we gave out a detailed illustration of the connectivity estimation model by calculating Src-con. Table 2 shows the symbol definition used in model.

The principle of connectivity estimation model is shown as follow:

1. Let $A_j$ be the event that slot $j$ is empty in $S_i$ at the end of the measurement period and $1_{A_j}$ be the random variable of event $A_j$. If slot $j$ is empty in $S_i$, then $1_{A_j} = 1$. Otherwise, $1_{A_j} = 0$;
2. Let $A_i$ be the event that slot $j$ is empty in $B$ at the end of the measurement period and $1_{A_i}$ be the random variable of event $A_i$. If slot $i$ is empty in $B$, then $1_{A_i} = 1$. Otherwise, $1_{A_i} = 0$;
3. Let $n$ be the sum of connectivity of all different hosts and $k$ be the number of different destination IP addresses for a source in the measurement period.
4. Let $U_m$ be random variable for the number of '0' bits in $B$ and $V_m$ be random variable for the percentage of '0' bits in $B$.
5. Let $U_s$ be random variable for the number of '0' bits in $S_i$ and $V_s$ be random variable for the percentage of '0' bits in $S_i$.

Then, we can know $V_m = U_m/m$ and $V_s = U_s/s$ clearly.

$$U_s = \sum_{j=0}^{s-1} 1_{A_j} \tag{1}$$

$$\begin{aligned} E(V_s) =& \frac{1}{s}E(U_s) = \frac{1}{s}\sum_{j=0}^{s-1} E(1_{A_j}) = \frac{1}{s}\sum_{j=0}^{s-1} Prob(1_{A_j}) \\ &= (1-\frac{1}{m})^{n-k}(1-\frac{1}{s})^{k} \\ &\approx e^{-\frac{n-k}{m}} e^{-\frac{k}{s}} \ as\ n-k, m, k, s \to \infty \\ &\approx e^{-\frac{n}{m}-\frac{k}{s}} \ as\ k \ll m \end{aligned} \tag{2}$$

$$\widehat{k} \approx -s * \frac{n}{m} - s * ln(E(V_s)) \tag{3}$$

$$U_m = \sum_{i=0}^{m-1} 1_{A_j} \tag{4}$$

$$E(V_m) = \frac{1}{m}E(U_m) = \frac{1}{m}\sum_{i=0}^{m-1} E(1_{A_j}) \approx e^{-\frac{n}{m}} \tag{5}$$

$$\widehat{k} \approx s * ln(E(V_m)) - s * ln(E(V_s)) \tag{6}$$

**Table 1: Identification results and accuracy of DRDNS or FRDNS using accurate connectivity measurement**

| src-con Threshold | dom-con Threshold | src-count Threshold | Identification number | Verification number | accuracy (%) |
|---|---|---|---|---|---|
| 100 | 4000 | 7000 | 562 | 416 | 74.02 |
| | | 8000 | 519 | 387 | 74.57 |
| | 5000 | 7000 | 523 | 410 | 78.39 |
| | | 8000 | 486 | 385 | 79.22 |
| | 6000 | 7000 | 476 | 378 | 79.41 |
| | | 8000 | 443 | 353 | 79.68 |
| 200 | 4000 | 7000 | 495 | 399 | 80.61 |
| | | 8000 | 456 | 377 | 82.68 |
| | 5000 | 7000 | 470 | 395 | 84.04 |
| | | 8000 | 436 | 371 | 85.09 |
| | 6000 | 7000 | 437 | 375 | 85.81 |
| | | 8000 | 405 | 354 | 87.41 |
| 300 | 400 | 7000 | 461 | 382 | 82.86 |
| | | 8000 | 425 | 373 | 87.76 |
| | 500 | 7000 | 440 | 387 | 87.95 |
| | | 8000 | 407 | 357 | 87.71 |
| | 600 | 7000 | 411 | 369 | 89.78 |
| | | 8000 | 379 | 337 | 88.92 |

**Table 2: Symbol Definition**

| Symbol | Definitions |
|---|---|
| $B$ | A bit array shared by all sources. The number of elements is m.<br>Each element is '0' initially and occupies just one bit.<br>When slot i is hashed, then $B[i]$ ='1'. |
| $S_i$ | A virtual bit array owned by each source. The number of elements is s.<br>Each element is '0' initially and occupies just one bit.<br>When slot $j$ is hashed, then $S_i[j]$ ='1'. |
| $R$ | A array of random number. The number of elements is s. |
| $(src, dst)$ | The source IP address and the destination IP address in every packet. |
| $H_m$ | Hash function used in hash map.<br>The return value of the hash function is less than m.<br>The hash process: $B[H_m(src\ XOR\ R[H_m(dst)\ \ mod\ \ s])]$ ='1'. |

Although m and n are very large in fact, they are not infinite in ideal conditions. The error of estimated measurement caused by parameters can be measured as

$$\sigma = \frac{|s * ln(E(V_m)) - s * ln(E(V_s)) - k|}{k} = |s*ln(\frac{1-\frac{1}{m}}{1-\frac{1}{s}})-1| \tag{7}$$

For instance, when $m = 10^6$ and $s = 200$, this error is only 0.25%. We can accept this error caused by parameters.

# 6. EXPERIMENTAL RESULTS

## 6.1 Error Analysis for Connectivity Estimation

The error rate of connectivity estimation is not only related to parameters, but also related to hash confliction. A stable and independent hash algorithm can make the data distribute in the slots evenly to reduce hash confliction. Related researches have shown that BOB hash has good balance, high performance and small error rate. Therefore, we choose BOB hash to process DNS traffic [16].

We use the test dataset from CUG whose bandwidth is 5Gbps mentioned in Section 4.4. We calculate accurate src-con $S_{1i}$ and estimated src-con $S_{2i}$ for each source in the DNS traffic. When $S_{1i}$ of one source is very large, the percentage of '0' bits of its virtual array is 0 in connectivity estimation model. Then $S_{2i}$ of this source is inf according to formula(6). If $S_{1i}$ is in the top 1% of all sources' accurate value, the error between $S_{1i}$ and $S_{2i}$ is approximately equal to 0. Otherwise the error is approximately equal to $\infty$. Let $\lambda$ be the threshold of accurate value $S_1$ and $\delta$ be the average relative error of estimated value $S_2$. Then

$$\delta = \frac{1}{n} * \sum_{i=1}^{n} (\frac{|s_{1i} - s_{2i}|}{s_{1i}} * 100\%) \tag{8}$$

Let n be the number of hosts whose accurate value is greater than the threshold. The change of $\delta$ with $\lambda$ is shown in Figure 12.

From Figure 12, we know $\delta$ is reduced while $\lambda$ is increased. This is in accordance with the principle of connectivity estimation model that the higher accurate value, the smaller average relative error is. At the same time, there is no er-

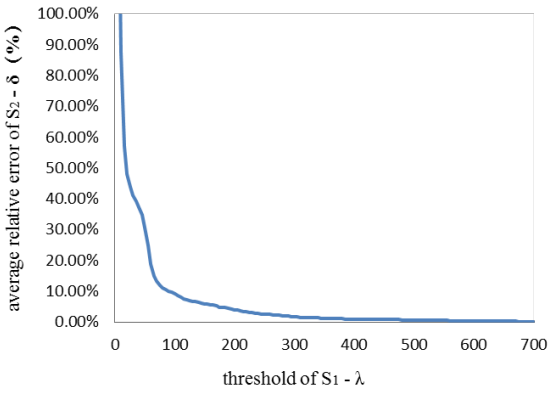


**Figure 12: The average relative error under different thresholds**

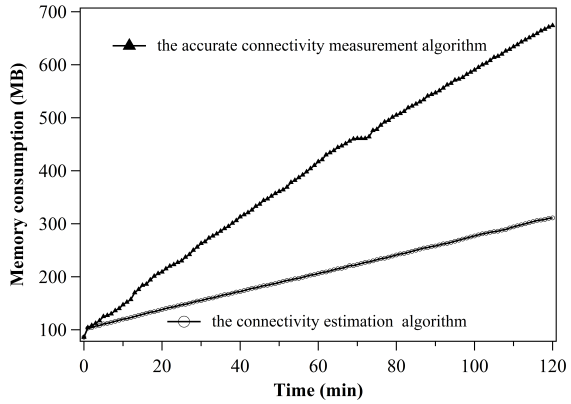


**Figure 13: Comparison of memory consumption**

ror equaling to $\infty$. This phenomenon shows if $S_{2i}$ is $\infty$, then its corresponding $S_{1i}$ is very large. When $\lambda = 50$, $\delta$ is 30.59%. When $\lambda = 100$, $\delta$ is similar to 10%. When $\lambda = 200$, $\delta$ is 3.88%. When $\lambda = 300$, $\delta$ is just 1.71%. Because active DRDNS or FRDNS has higher connectivity, so we just consider estimation error of high connectivity. When $\lambda > 100$, $\delta$ is less than 10%. This error rate is acceptable. However, this error is inevitable because connectivity cannot reach infinity in the process of approximate calculation.

## 6.2 Performance Evaluation for Connectivity Estimation

We ran the performance evaluation on both accurate connectivity measurement algorithm and connectivity estimation algorithm in CUG online. We took the records of memory consumption of these two algorithms per minute. Memory consumption is shown in Figure 13, the memory consumption of the accurate connectivity measurement algorithm grows faster. After 120 minutes, the accurate connectivity measurement algorithm consumes more than two times memory than the connectivity estimation algorithm. Therefore, the connectivity estimation algorithm can save 65.49% memory (1.17 GB) and is more suitable for backbone traffic analysis online.

The packet drop rate of accurate connectivity measurement algorithm reaches 0.3% at 108 minutes. However, even at 720 minutes the packet drop rate of our connectivity estimation algorithm does not reach the value. Therefore, the processing ability of connectivity estimation algorithm is stronger.

## 6.3 Online Identification Results Analysis

We deploy online RDNS identification framework in a reginal CUG and get identification results through monitoring the reginal CUG. Online passive RDNS identification researches is few in the past. We compare the online RDNS identification results between features calculated by accurate connectivity measurement and features calculated by connectivity estimation. According to the thresholds analysis in Section 4.4, we take the same thresholds for identifying RDNS online and the same method to verify the accuracy of online identification results.As shown in Table 3, the highest accuracy of identifying FRDNS and DRDNS can reach nearly 89%. The accuracy of identification online is similar to the analysis results in 4.4. The higher threshold of features is, the smaller estimation error has. Then reasonable thresholds have little impact on the error caused by the connectivity estimation algorithm. The number of online identification IRDNS is 8996. The number of verified IRDNS is 8105. The highest accuracy of identifying IRDNS online can reach 90%.

# 7. CONCLUSION

Identifying the active RDNS quickly and accurately is the basis of RDNS security assessments in a network environment.In this paper, we proposed a online RDNS identification framework based on connectivity estimation and CNAME redirect behavior. We implemented an efficient online calculation algorithm for both host connectivity and domain connectivity estimation. Experimental evaluations with real data validate the result that our method can achieve high accuracy of active RDNS identification by choosing a suitable threshold value. Meanwhile, compared with the traditional active and passive schemes, our scheme has advantages such as less resources consumption, less storage consumption and better timeliness.

However, we cannot get the whole active RDNS and didn't guarantee the recall rate of our results in the proposed scheme. As a future research direction, we plan to measure the recall rate of active RDNS identification results while maintaining high accuracy and low consumption.

# 8. ACKNOWLEDGEMENT

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences No.XDA06030200. the National Natural Science Foundation of China No.61402464.

# 9. REFERENCES

[1] Dagon D, Provos N, Lee C.P., Lee W.:Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority. In: Proceedings of the Network and Distributed System Security Symposium, pp: San Diego, California, USA (2008)

[2] Cranor C. D, Gansner E, Krishnamurthy B, Spatscheck O.: Characterizing Large DNS Traces Using Graphs. In: Proceedings of the 1st ACM SIGCOMM Internet Measurement Workshop,PP.55–67, San Francisco, California, USA (2001)

**Table 3: Identification results and accuracy of DRDNS or FRDNS using connectivity estimation**

| src-con Threshold | dom-con Threshold | src-count Threshold | Identification number | Verification number | accuracy (%) |
|---|---|---|---|---|---|
| 100 | 4000 | 7000 | 642 | 457 | 71.18 |
| | | 8000 | 606 | 442 | 72.94 |
| | 5000 | 7000 | 634 | 454 | 71.61 |
| | | 8000 | 598 | 437 | 73.08 |
| | 6000 | 7000 | 577 | 426 | 73.83 |
| | | 8000 | 545 | 419 | 76.88 |
| 200 | 4000 | 7000 | 499 | 413 | 82.77 |
| | | 8000 | 476 | 402 | 84.45 |
| | 5000 | 7000 | 497 | 421 | 84.71 |
| | | 8000 | 474 | 401 | 84.60 |
| | 6000 | 7000 | 470 | 403 | 85.74 |
| | | 8000 | 448 | 388 | 86.61 |
| 300 | 400 | 7000 | 447 | 386 | 86.35 |
| | | 8000 | 425 | 378 | 88.94 |
| | 500 | 7000 | 446 | 391 | 87.67 |
| | | 8000 | 424 | 378 | 89.15 |
| | 600 | 7000 | 431 | 382 | 88.63 |
| | | 8000 | 409 | 363 | 88.75 |


[3] Schomp K.čň Callahan T, Rabinovich M, Allman M.: On Measuring the Client-Side DNS Infrastructure. In: Proceedings of the 2013 Internet Measurement Conference, pp.77–90, Barcelona, Spain (2013)

[4] DESIGN AND IMPLEMENTION FOR DNS INFORMATION DETECTION SYSTEM BASED ON DISTRIBUTED PLATFORM. Sun Rui, Harbin (2013)

[5] Huang C. Maltz D. A.: Public DNS System and Global Traffic Management. In: Proceedings of 30th International Conference on Computer Communications. pp.2615–2623.Shanghai, China (2011)

[6] Schomp K., Callahan T., Rabinovich M., Allman M.: Assessing DNS Vulnerability to Record Injection. In Proceedings of 15th International Conference on Passive and Active Measurement, pp.214–223. Los Angeles, CA, USA (2014)

[7] Chun B., Culler D.,Roscoe T.:Planetlab: an overlay testbed for broad-coverage services. J. Computer Communication Review. 33(3), 3–12 (2003).

[8] Dhandere K., Kim H., Pan T. J.:The Application and Effect of Sampling Methods on Collecting Network Traffic Statistics. J.Citeseer, (2001)

[9] He G., Hou J. C.: On sampling self-similar Internet traffic. J. Comput. Networks, 50, 2919–2936 (2006)

[10] Raspall F. :Efficient packet sampling for accurate traffic measurements. J. Comput. Networks, 56, 1667–1684 (2012)

[11] Song D., Gibbons P. B.: New Streaming Algorithms for Fast Detection of Superspreaders. In Proceedings of Network and Distributed System Security Symposium, San Diego, California, USA (2005)

[12] Zhao Q.,Xu J.,Kumar A.: Detection of super sources and destinations in high-speed networks: Algorithms, analysis and evaluation. IEEE J. Sel. Areas Commun., 24, 1840-1852, 2006.

[13] Li T.,Chen S.,Luo W.,Zhang M.: Scan detection in high-speed networks based on optimal dynamic bit sharing. In: Proceeding of 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, pp. 3200–3208,Shanghai, China (2011)

[14] Yoon M. K., Li T., Chen S., Peir J. K.: Fit a compact spread estimator in small high-speed memory. J. IEEE/ACM Trans. Netw., 19, 1253-1264 (2011)

[15] Alzoubi A.H,Rabinovich M.,Spatscheck O.: The Anatomy of LDNS Clusters:Findings and Implications for Web Content Delivery.In: Proceedings of the 22nd International World Wide Web Conference, pp:83–94,Rio de Janeiro, Brazil (2013)

[16] Henke C., Schmoll C.,Zseby T.: Empirical Evaluation of Hash Functions for PacketID Generation in Sampled Multipoint State of Art. In: Proceedings of 10th International Conference on Passive and Active Network Measurement, pp. 197–206,Seoul, Korea (2009)