A study of efficient avoidance in the event of DNS(Domain Name System) failure

YANG-WON LIM, HYON-A HWANG, HANKYU LIM
Department of Multimedia Engineering
Andong National University
388 Songchun-dong , Andong City, KyungBuk 760-749
The Republic of KOREA
http://multl.andong.ac.kr

Abstract: - The Domain Name System (DNS) is the core system for managing Internet address resources, providing the most fundamental naming service. Currently, the DNS is classified into a tree structure. In this structure, normal access to the lower DNS is difficult when there is an error in the upper DNS. Such risk still remains even when a supplementary DNS is operated. However, due to the merit of the DNS enabling fast searches, it is impracticable to abandon the current tree structure. To efficiently correspond to DNS errors, this study suggests a method where the merit of the current tree structure is kept, while a temporary operation of the local DNS is available when errors occur by adding a horizontal and independent DNS structure.

Key-Words: - DNS, Domain Name System, independent DNS, local DNS, DNS failure, host name solution

1 Introduction

Due to the recent development of communication and information technology, Internet service has been widely supplied and used in companies, public offices and ordinary households. Every day, more and more people are using super-speed Internet services to exchange information, enjoy financial services, use e-commerce, play games, etc [1]. Before, users just had to input the name of the host connected to the network and the corresponding IP address to use the service. However, as the network drastically expanded, serious problems started to occur when all hosts received host files by accessing the NIC(Network Information Center) which provides registration and management of IP address. Moreover, it became impossible to manage host files in NIC. DNS was developed to resolve such problems. The current DNS is very efficient because it provides expandability for networks using hierarchical name spaces and supports distributed management [2]. NIC is part of the Domain Name System (DNS) of the Internet which converts domain names to IP addresses. It is an organization which manages the registration of Domain names within the top-level domains for which it is responsible, controls the policies of domain name allocation, and technically operates its top-level domain. As the network is further developed, however, damage caused by cyber attacks is also greatly increasing, and a DNS server is especially vulnerable to such attacks. When the DNS server receives damage from these attacks, the Internet becomes unavailable, causing major chaos [3].

Although the vertical and hierarchical structure of the current DNS has many merits, such as searching data fast and efficiently managing them, it also has demerits, such as an unstable connection ring and a risk of not being able to interpret IP addresses when there is an error in a major node [4].

The current DNS system is exposed to danger and risk all the time. Its demerits and errors can create the worst situation, a total paralysis of all Internet accesses. If a safety measure can be prepared for such a situation, damages can be minimized.

As a resolution to such problems with the DNS, this study suggests a corresponding measure where the merits of the current hierarchical structure can be fully utilized, whereby an independent DNS and a horizontal structure are also added on to provide mutual operation in the event of errors.

This study is organized as follows. In Section 2, the study outlines the structure of the current DNS and its principles of operation. Then, it discusses the attacks that might cause errors in a DNS, as well as the merits and demerits of a hierarchical structure. In Section 3, the study describes a system that corresponds to DNS errors, and minimizes damage on the Internet network. In Section 4, the study explains the system's principle

of realization and operation and, finally, in Section 5, it draws a conclusion.

2 Related research

2.1 Summary of DNS (Domain Name System)

When computers communicate with each other through the Internet, an IP address must first be allocated to each of them logically. Such IP address must be given to clients requesting the information, as well as servers providing the information. IP addresses are expressed as 32-bit numbers, and for this reason, it is difficult for users (clients) to remember all of them. Thus, a name is given to allow clients to easily remember IP addresses, and the device which maps the names used by the clients with the IP addresses used by the computers is called DNS [5].
Fig.1 shows the basic structure of IP address interpretation.

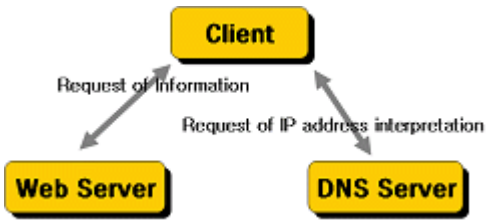

Fig.1 The basic structure of IP address interpretation

The names of any new hosts on the Internet were registered, and the IP address of the central host file was maintained by the Network Information Center (NIC). Then, all hosts on the Internet mutually shared these concentrated host files with each other. As the Internet expanded, however, the following problems were caused by the concentrated host files.
- The expansion of the Internet triggered renewal on a daily basis.
- The network of the Stanford Research Institute, which managed the host files, caused many Internet errors .
- Host names could not be copied freely over the Internet due to the monotonous characteristic of nominal space.
- The renewal of names took a lot of time to be fully displayed on the Internet.

The domain name space takes a tree structure, which shows all domains organizing the name space on the Internet. The root domain is on top of the tree. There is no actual text label in the root domain. Instead, it is expressed by using a period (.). The uppermost layers of domains are located under the root domain. The characteristic of these uppermost layers of domains can be divided into two. The first characteristic expresses the type of work, while the second expresses the region of the group (nation) using a two-digit code. The ordinary uppermost layers of domains were added to expand the domain name space. The second layers of domains, including the hosts and sub-domains, are located under the uppermost layers of domains. Fig.2 shows the example of groups registered as the second layers of domains under the uppermost layers of domains.

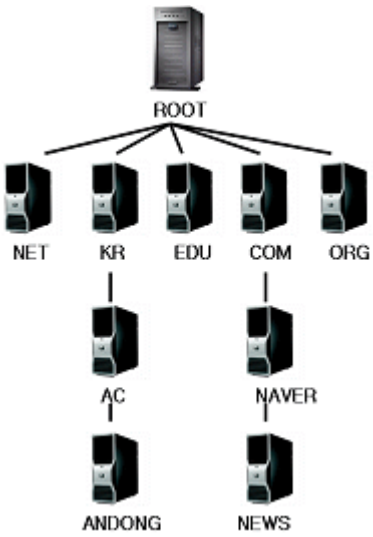

Fig.2 The hierarchical DNS structure

2.2 Operational principles of DNS

A computer changes a host name into an IP address by following the process below.

① Judge if the domain subjected for change is its own host currently in operation
② Judge if the domain in question exists in the host files
③ Judge if the remote DNS server has information on the domain

If none of these interpretation methods can find an IP address for the subject host name, the application

program returns an error message saying that the host name cannot be found [6].

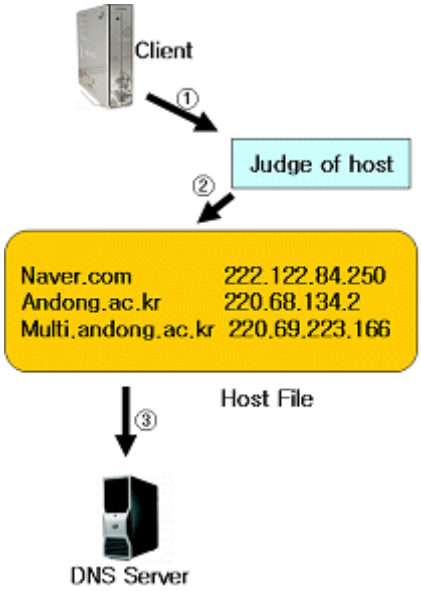

Fig.3 Process of IP address interpretation

2.3 Merits and demerits of a hierarchical DNS structure and cases of damage

In the uppermost level of the Internet, there are root domains (expressed with '.'), and these root domains are composed of numerous top level domains (TLD). Under these top level domains come sub-domains, of which the layers are not fixed. Each domain has a server which stores information on the domain name and responds to inquiries in real-time. This server is called the DNS (Domain Name System) server. Through the DNS server, domain names can be changed into IP addresses. As mentioned above, a hierarchical DNS structure is used today, and this structure naturally sends many requests to the uppermost DNS for interpreting domains, thereby putting an excessive burden on the system.

Around the world, there are currently 13 root DNS (10 in the United States, 2 in Europe and 1 in Japan), serving the central role of the Internet. If an error is caused in one of these 13 root DNS, a cyber disaster that could paralyze the Internet will break out. In fact, there was an incident of a major cyber disaster that took place in Korea on January 25, 2003. Korea uses two backbone networks and related DNS are operated in two places, one in the Hyehwa telephone office and the other in the Guro telephone office of KT Corporation. On January 25, 2003, the DNS server located in the Hyehwa telephone office caused errors. The access signals processed in these two offices take up about 80% of all domestic Internet accesses. As the DNS server in Hyehwa telephone office went down, access signals stormed the DNS server in the Guro telephone office, overloading the server there as well. Websites were shut down, Internet reservations were unavailable, email and Internet banking services were stopped, and electronic commerce was paralyzed, causing major chaos. The economic damage caused by this disaster was so great it was converted into monetary values [7].

Although most of the web servers were operating normally, the clients could not access these operating web servers because they could not receive the interpretation of the IP addresses. The fact that an attack from such a small virus can paralyze the entire Internet network has grave significance. The more serious problem is that such weakness of DNS servers is not fully resolved yet. In other words, the risk caused by such weakness is still out there, threatening us everyday. This problem cannot be fully resolved by simply reinforcing security measures. An alternative measure must be present [8].

The next chapter describes an improved measure where the merits of the current hierarchical DNS structure can be fully utilized, while a horizontal and independent DNS structure can be added on to supplement the current weakness.

3 Suggested measure to correspond to DNS errors

As shown in Fig.4 below, the current hierarchical DNS structure uses a multi-level structure to interpret IP addresses. In this structure, if the root or upper DNS does not function normally due to errors, the DNS in the lower level also cannot function normally because the connection ring is broken. Such malfunctioning of the lower DNS occurs even when the lower DNS is operating normally. Of course, a supplementary DNS can be operated for each DNS level, but the risk of malfunctioning still remains.

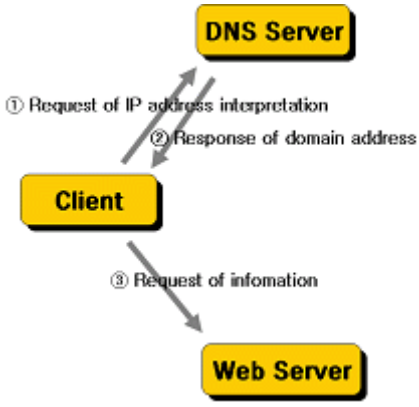

Fig.4 Structure of IP address interpretation

To resolve the problem above, this study suggests a measure that keeps the merits of the current hierarchical structure, while adding a horizontal and independent DNS structure for supplementary purposes. The suggested measure operates the current hierarchical structure and an independent (local) DNS structure together in normal times. When there is an error in the DNS, the proposed system operates the independent structure and the horizontal structure together. Also, by operating a local DNS exclusive to itself, it can minimize the load on the upper DNS. When using internal DNS information, the process of interpreting IP addresses is omitted to allow fast access. Fig.5 shows the current hierarchical structure combined horizontally with an exclusive local DNS and other local DNS.

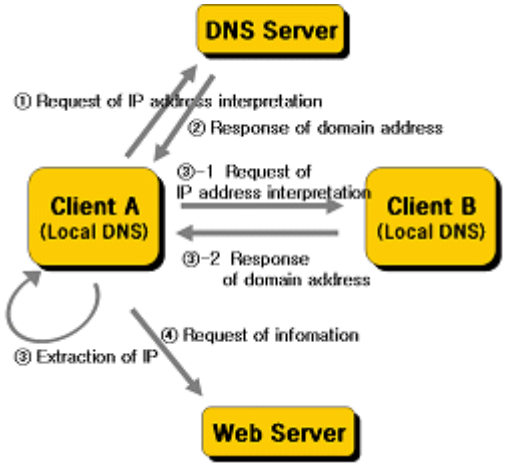

Fig.5 Structure of the suggested measure (hierarchical, horizontal and independent DNS structure)

The composition and the operational principles of the suggested measure are as follows. First, the application program attempts to search the address of the web server to be accessed in its local DNS. The function of the suggested measure dynamically and automatically organizes the host files upon the request of the application program. Such a dynamic function allows users to enjoy the Internet as they used to, without being aware of anything. If the IP address cannot be found in its local DNS, the application program requests other surrounding local DNS for domain interpretation. If these DNS do not have the information necessary, the application program repetitively requests other local DNS for the information until the requested IP address is found.

4 Implementation and Design

4.1 Structure of independent (local) DNS

As shown in Fig.6, the independent (local) DNS suggested in this study is operated on a local computer (user's computer), just as any application programs. The components manage the DNS data using a cache memory and file system, or a database. The principle of operation is as follows.

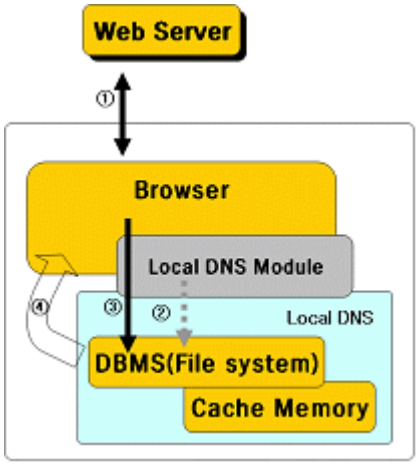

Fig.6 Structure of independent (local) DNS

① When the DNS is normal, access to the web server is normally made by the URL information inputted in the address window of the web browser. The IP of the domain is stored as local DNS information.

② Through a parsing method, the local DNS module extracts only the server domain address from the URL, and stores it with the IP.

③ When there is an error in the DNS and access to the web server is not available, the inputted URL is parsed to extract the information of the domain. Then, the related IP in DBMS and cache memory is searched.
④ The IP value searched in the local DNS is sent to the web browser.

When the upper DNS is operating normally, the IP searched in step ① is stored and renewed in the host file of the local DNS.

```
If isLiveDNS(Domain) Then
      ip = checkICMP(Domain)
      if isNotDomainInDBMS then
          saveDomain(Domain, ip)
      else
          updateDomain(Domain, ip)
      end if
else
      ip = searchDBMSandCache(Domain)
      if IPisNotExistinLocalDNS then
          searchOtherClientDNS()
      end if
      pushIPtoBrowser(Domain, ip, BrowserName)
end If
```

Fig.7 independent (local) DNS algorithm

Through the process above, local host files are dynamically organized without the manual work of users. Above all, in case there is an error in the DNS, the information allows users limited access to the web servers that they visited previously without the help of the upper DNS.

4.2 Designing and realizing DNS with a horizontal structure

A horizontal DNS structure supplements the demerits of an independent (local) DNS. As mentioned above, the biggest weakness of the local DNS is that all DNS information cannot be stored in the local DNS. This is because the local DNS cannot update renewed information in real-time, and has limits on managing large amounts of information. To overcome this problem, mutual supplementation can be made by requesting the missing DNS information from the local DNS of other computers. The principle of operation is as follows.

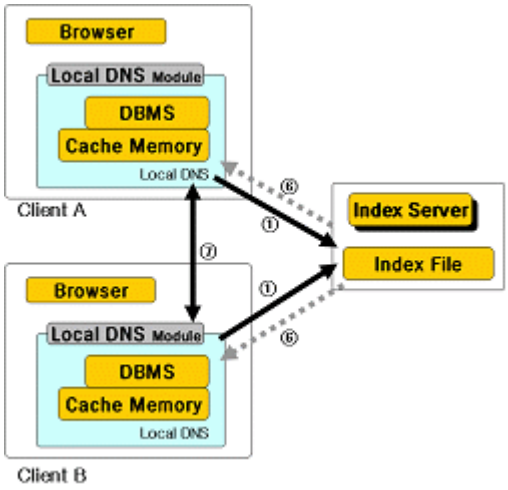

Fig.8 DNS connection with a horizontal structure

① When the local DNS of computer 'A' is operated, the IP of 'A' and the number of IP addresses stored in its host file are communicated to the index server. Using the host file, the index server stores the data sent from each computer to prepare for emergencies. Then, the index server communicates the IP address (link address) of computer 'B,' which has plenty of information, to computer 'A.' After receiving the IP address of computer 'B' from the index server, computer 'A' stores it in its memory as a variable to horizontally connect to computer 'B' in case of emergency.
② The URL information inputted in the address window of the web browser is sent to the local DNS.
③ Through a parsing method, the local DNS extracts only the server domain address excluding the directory address. With the extracted domain address, the local DNS attempts a sequential search on the cache memory where information is saved in an array. If the related IP address is not found, the local DNS goes to step ④. If the related IP address is found, the local DNS sends it to the web browser.
④ If the local DNS cannot find the related IP address in its cache memory, it searches the host file.
⑤ If the IP address is found, the local DNS sends it to the web browser.
⑥ If the related IP address is not found in the host file of computer 'A,' a request is made to the cache memory of the index server. If the address is still missing, a request for domain interpretation is made to the local DNS of computer 'B' by using the link address (the IP address of computer 'B') from step ①.

Upon the request of computer 'A,' computer 'B' searches for the requested IP address in its host file. If the related IP address is found, computer 'B' sends it to computer 'A.'

⑦ If computer 'B' does not have the related IP address, it sends the link address that it has to computer 'A.' Using the link address that computer 'B' sent, computer 'A' makes a request to another computer until the IP address is found. When computer 'A' finally receives the information on the related IP address, it informs the result to the cache memory of the index server, so that the index server can provide the information to other clients upon request. The optimal algorithm for searching DNS information is shown in the following Fig.9.

```
If isLiveDNS(Domain) Then
    call LocalDNSModule
    putLocalInfotoIndexServer(LocalDNSinfo)
else
    if IPisNotExistinLocalDNS then
        loop
            OtherInfo = getDNSInfofromIndexServer
            ArrayIP = OtherInfo.IPList
            ArrayDomain = OtherInfo.DomainList
            Search(Domain, ArrayDomain)
            if isExistDomain then
                ip = ArrayIP(Search)
                Exit loop
            end if
        end loop
    end if
    pushIPtoBrowser(Domain, ip, BrowserName)
end If
```

Fig.9 Optimal algorithm for searching DNS information

Through the repetitive data processing procedure described above, DNS information in other computers can be mutually used and shared.

4.3 Analyzing the DNS structure suggested

The DNS structure suggested by this study operates the hierarchical structure and the independent DNS structure together in normal times. When the upper DNS causes errors, it uses the independent structure and the horizontal structure to operate the DNS under any circumstances. Also, by operating an exclusive local (independent) DNS, burdens on the upper DNS can be minimized. Fig.10 shows a simulation of the process of collecting independent DNS information. When the DNS is normally operated, the related IP is stored in the local database. When DNS is normally operating, the web surfing is possible after input of the domain in the address windows(①) of the browser. At this time, the IP address which is verified by the DNS server is stored as the local information of PC(②). If the address interpretation problems are occurring due to the DNS failure, the web surfing is not possible in the previous research because of failure of DNS server connection. However, the web surfing by the proposing idea in this research is normally operating because the IP address is given by the stored local information. Even if the DNS server has problems, the web server is normally operating. When the users access the web by the domain address, the normal connection is impossible due to DNS malfunction even if the web server is normal. The uses can access the web by the IP address in the local information regardless of the DNS server failure.

Fig.10 Simulation screen of independent (local) DNS

When there is a problem in interpreting addresses due to an error in the DNS, access is made by using the IP address stored in the local database. When an error occurs and the horizontal structure is used, access time might be extended, and in the worst case, the requested information might not be available. Nevertheless, the management of a certain amount of information collected through a simulation with a collectable format will not be a problem due to the rapid growth of recent computer performance. Furthermore, the problem of extended access time only happens when there is an error in the upper DNS. Thus, considering the merits of the suggested DNS structure where most of the Internet services are available without interruption, the problems above are rather trivial. Since the suggested DNS structure can minimize the

economic damage caused by Internet shutdowns, it is safe to say that the suggested structure is more stable than the previous DSN structure. Moreover, the structure suggested in this study can be realized at a low cost, since it does not require the alternation of the previous DNS network.

5 Conclusion

The current DNS is very efficient because it provides expandability for networks using hierarchical name spaces and supports distributed management. Due to the characteristic of a hierarchical structure, however, the current DNS structure has demerits, such as a weak connection structure, unsecured environment and exposure to malicious attacks. Such risks still remain even if a supplementary DNS is operated.

Since the current hierarchical structure has merits, such as dispersing information, allowing fast searches and managing data efficiently, it would be inappropriate to abandon this structure. However, attempts must be made to overcome the problems of the current structure. The measure suggested in this study follows the current hierarchical structure, while adding a horizontal and independent DNS structure. In normal times, the current hierarchical structure and the local DNS structure are operated together; and when there is an error in the upper DNS, the independent structure and the horizontal structure are operated together. As a result, the client can access the Internet normally by using the information stored in the client itself, even though there is an error in the upper DNS. The suggested measure can efficiently manage all DNS operations and, at the same time, provide more stable Internet access to clients. Also, by using the DNS information stored in the client itself, the process of interpreting an IP address can be omitted to allow faster access. The suggested measure can be used in systems to actively correspond to malicious DNS attacks.

References:

[1] Hiroshi Ogawa, Yasunari Goto, *Web 2.0 Book*, Impress Japan Corporation, 2006

[2] Unlyess Black, *Computer Networks-Protocol ,Standards and Interfaces*, Prentice Hall, 1993

[3] R.Lemos, *Net attack-how it was squashed*, CNET News.com, October 2002

[4] *A discussion meeting of information security* , http://news.naver.com/news/read.php?mode=LSD&office_id=030&article_id=0000057637&section_id=105&menu_id=105, ETNews, 2004

[5] Behroz A. Forouzan, *TCP/IP Protocol Suite*, McGraw Hill, 2001

[6] Brain Komar, "TCP/IP Network Administration", Sams, 1999

[7] HongSub Lee, *Cyber disaster – more frightening than a natural disaster*, http://www.etnews.co.kr/news/detail.html?id=200602240025, ETNews, 2006

[8] Chuck Easttom, *Computer security fundamentals*, Prentice Hall, 2006