Anycast 기술을 통한 신뢰적 향상 기법의 DNS 서비스에 관한 연구

김 보 승* · 김 정 재** · 김 경 민*** · 박 찬 길**** · 신 용 태*****

A Reliability Improvement Technique of DNS Services Based on Anycast

Kim, Bo Seung · Kim, Jeong Jai · Kim, Kyung Min · Park, Chan Kil · Shin, Yong Tae

〈Abstract〉

DNS(Domain Name System) is a huge distributed database that converts host name to IP address. We are expecting the importance of DNS is more increased because many Internet application services appear according to the continuous increase of Internet users and nearly all the Internet application services use DNS. To prevent the interruption of DNS service, DNS server is configured with primary DNS server and a secondary DNS server which takes the place of primary DNS server in case of the service interruption. But this scheme is difficult for providing DNS service constantly in case of DDoS attack, which brings about much network load or network problems in DNS server group. Therefore, This paper proposed the scheme to locally distribute load of DNS server, and the use of address system to group the distributed DNS servers. Also, it proposed the authentication scheme of the correspondent server in case the server is changed in DNS server group having grouping address. In this paper, it is shown that the prosed scheme guarantees the improved service reliability with maintaining the present service performance through the evaluation. Through this, we can expect the high improved DNS service can be provided in the Internet environment in the future.

Key Words : Domain Name System, DNS, Anycast , DNS Authentication, Recursive DNS

Ⅰ. 서론

DNS(Domain Name System)란 도메인 네임과 그에 해당하는 IP 주소를 전환하고 변환하는 메커니즘으로써 웹, 이메일 등 다양한 응용 프로그램이 사용자에게 서비스가 이루어지기 위해서 핵심적인 역할을 담당하고 있는 거대한 분산 데이터베이스이다.

현재 IPv4 환경에서 32bit의 IP 주소는 산술적인 단순한 계산으로도 42억 개의 IP 주소가 존재한다. 이러한 IP 주소를 직접 사용하여 자신이 원하는 호스트로 접근한다

* 숭실대학교 컴퓨터학과(제 1저자, 교신저자)
** (주) RetailTech 수석 연구원
*** (주)디지캡 연구원
**** 한국사이버대학교 정보보안학과 교수
***** 숭실대학교 컴퓨터학부 교수

는 것은 불가능하다. 또한, IPv4 환경에 비해 월등히 늘어난 수의 주소를 가지는 128bit의 IPv6 환경[1]으로의 전이과정에 있는 앞으로의 상황에서 IP 주소와 도메인 네임간의 전환 서비스를 제공하는 DNS의 중요성은 더욱 부각되어진다. 도메인 네임 서비스의 중요성으로 인해 서비스 제공자는 DNS 서버가 최대의 가용성을 유지하도록 하기 위하여 일반적으로 1차(primary) DNS 서버와 함께 2차(secondary) DNS 서버를 서비스 제공자의 망에 구성한다. 이러한 구성을 통해 사용자는 일반적으로 자신의 PC에 설정한 1차 DNS 서버를 통해 서비스를 이용하게 되고, 1차 DNS 서버가 응답하지 않을 경우, 함께 설정한 2차 DNS 서버로 전환하여 DNS 쿼리를 전송함으로써 목적하는 서비스를 이용할 수 있게 된다. 하지만 1차, 2차 DNS 서버를 일반적으로 동일한 망에 구성하게 되는 기존의 방식은 악의적인 목적을 가진 공격자의 분산 서비스 거부공격(DDoS)이나 네트워크의 물리적인 장애에 대한 보안상의 약점이 드러나게 된다.

본 논문은 IPv6 환경에서 DNS 서버를 구성할 때 1차, 2차 DNS 서버로 이루어진 환경의 보안적인 문제점들을 극복하면서 일반 사용자가 DNS 서버 이상 시 별다른 조치 없이도 서비스 연속성을 유지할 수 있도록 하기 위해 Anycast 기술을 적용한 DNS 서버를 이용하여 그 해결책을 제안한다. 그리고 Anycast DNS 서버의 신뢰성을 높이기 위해 Anycast 라우팅에 의해 임의로 선택된 DNS 서버를 클라이언트가 CA를 통해 인증 할 수 있는 방법에 대해 제안한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 기존에 사용중인 DNS 서버의 구성 방식에 대한 문제점을 파악하고 Anycast 기법을 DNS 서버에 적용하였을 때 나타날 수 있는 장단점에 대해 알아본다. Ⅲ장에서는 제안사항인 Anycast 주소를 이용하여 DNS 서버를 구성하는 것과 Anycast DNS 서버를 인증하는 기법을 제시한다. Ⅳ장에서는 실제 테스트를 통해 기존의 기법과 제안 기법의 성능을 분석하고 마지막 Ⅴ장에서는 결론 및 향후 과제를 제시한다.

Ⅱ. 관련연구

2.1 DNS

DNS(Domain Name System)는 네트워크 계층이 이해하는 네트워크 주소와 사람이 이해하고 기억하기 쉬운 네임을 상호 변환하여 사용할 수 있도록 개발되었다. 초기에 소수의 호스트만이 연결되어 있던 환경에서는 네트워크 주소를 직접 사용하는 것에 불편함이 없었으나 점차 그 규모가 확대되면서 보다 이해하기 쉽고 기억하기 쉬운 호스트 네임의 사용이 도입되었고, 그로인해 현재의 인터넷 네임 체계인 DNS가 등장하였다[2].

네트워크 주소는 전 세계에 산재한 통신망 속에서 단일한 대상을 지정하고 해당통신 대상에 접근할 수 있는 수단을 제공한다. 네트워크 주소를 사용하는 통신 매체가 급속하게 증가함에 따라 DNS가 현재에 이르러서는 인터넷에 필수적인 네임체계로서 자리 잡고 있다.

2.1.1 DNS 서버의 유형

(1) Recursive DNS : 클라이언트의 요청에 대해 이전에 Cache에 저장되어 있는 결과를 돌려주거나, 상위 DNS로의 계층적인 질의를 통해 결과를 제공해 주는 역할을 하는 DNS 서버로 ‘recursive mode’가 활성화되어 있는 DNS 서버이다. 일반적으로 클라이언트에 DNS 서버 리스트로 설정되어 1차적으로 질의를 받게 되는 DNS 서버를 뜻한다. 본 논문에서 제안하는 기법들은 모두 Recursive DNS 서버를 대상으로 하였다.

(2) Authoritative DNS : Root부터 Top-Level Domain, Second-Level Domain 등 계층적으로 구성된 DNS 체계에서 그 하위 영역에 대한 위치 정보의 권한을 가진 DNS 서버를 뜻한다. 일반적으로 DNS 질의를 통해 알고자 하는 대상이 속한 네트워크에 대한 권한을 가진 DNS 서버를 말한다.

2.2 기존의 DNS 환경

2.2.1 Object Impermanence

기존의 메인 DNS 서버에 대체 DNS 서버를 두는 기술들을 'Object Impermanence' 방식이라 부른다[3]. 이 방식은 다음과 같은 특징이 있다. 클라이언트는 설정된 DNS 서버 리스트로부터 Primary로 설정된 DNS 서버로 요청을 보내고 예정된 시간(timeout : 0-5초)동안 응답을 기다린다. 만약 예정된 시간 내에 'no response' 메시지를 받는다면 클라이언트는 서버 리스트의 다음 서버로 요청이 전달된다. 그리고 예정된 시간을 다시 기다린다. 받은 응답없이 서버 리스트의 서버를 다 사용해 버리면, 클라이언트는 Internal table('hosts' file)을 참고하거나 에러 정보를 반환한다.

2.2.2 Object Impermanence 방식의 문제점

기존의 Object Impermanence 방식의 문제점은 대체 서버로 선택되어진 Secondary DNS 서버 또는 백업 DNS 서버가 Primary DNS 서버와 같은 네트워크에 설치되고 클라이언트에게 권장되는 점이다. 설정 시 DNS1 서버를 Primary로 설정하고, DNS1 서버가 서비스를 제공할 수 없는 상황에 대처하기 위해 DNS2 서버를 Secondary 서버로 설정하였을 때, DNS1 서버가 서비스를 하지 못할 경우 DNS2 서버로 충분이 대처가 가능하다. 하지만 DDoS 공격 등의 네트워크 트래픽을 급증시키는 공격이나, 네트워크 자체에 대한 공격 그리고 DNS 서버가 위치한 네트워크의 물리적인 이상이 발생한 상황에는 DNS2 서버 역시 그 역할을 수행할 수 없다.

2.3 Anycast 전송방식

2.3.1 Anycast

Anycast는 단일 송신자와 다중 수신자 사이의 통신인 Multicast, 그리고 단일 송신자와 단일 수신자 사이의 통신인 Unicast와 대비하여 정의되었다. Anycast는 Multicast와 같이 일-대-다 전송을 지원한다. 그러나 그룹 내의 모든 수신자에게 보내어지는 것이 아니라 가장 가까운 서버 또는 사용자에게 서비스 할 수 있는 최선의 한 노드로만 전송하므로 결과적으로는 일-대-일 전송방식이라고도 볼 수 있다[4, 5].

하나의 Anycast 주소는 다수의 호스트에 할당되며, 발신 노드가 해당 Anycast 주소를 목적지로 하여 패킷을 전송하게 되면, 라우터가 라우팅 테이블에서 같은 Anycast 주소를 갖는 호스트 중 가장 근접한 호스트로 라우팅하게 된다. 이때 라우팅 거리는 설정되어 있는 라우팅 프로토콜을 따른다. 사용자는 가장 가까운 서비스 호스트로부터 서비스를 제공 받을 수 있음으로써 서비스의 품질 향상을, Anycast 서비스 호스트는 부하분산과 장애 시 서비스의 연속성 효과를 기대할 수 있다. Anycast 주소는 Unicast 주소의 구조를 그대로 사용하며, Unicast 주소 공간으로부터 할당되어진다. 따라서 Anycast 주소는 Unicast 주소와 구문적으로 구분되지 않는다.

2.3.2 Anycast 주소의 적용 범위

Anycast 주소는 UDP와 같은 Connectionless 응용서비스에 적합하다. 단순한 질의/응답과 같이 질의 1 패킷, 응답 1 패킷으로 구성된 응용서비스가 Anycast 방식에 의해 연결 호스트가 변경되어도 사용자는 아무런 문제없이 서비스를 받을 수 있다. 해당 서비스에는 DNS, Syslog, SNTP 등이 있다.

Ⅲ. Anycast 기반의 DNS 서비스 신뢰성 향상 기법

본 논문은 인터넷 사용의 필수적인 요소 중 하나인 DNS 중 사용자와 직접적으로 통신하게 되는 Recursive DNS 서버의 서비스 가용성을 높이기 위해 Anycast IP 주소를 Recursive DNS 서버에 적용하여 구성하는 방법을 제안한다. 또한, 2차로 선택되는 Anycast DNS 서버의 신뢰성을 확보하기 위하여 Anycast DNS 서버를 인증하는 기법을 제안한다.

3.1 Anycast IP 주소를 활용한 Recursive DNS 구성

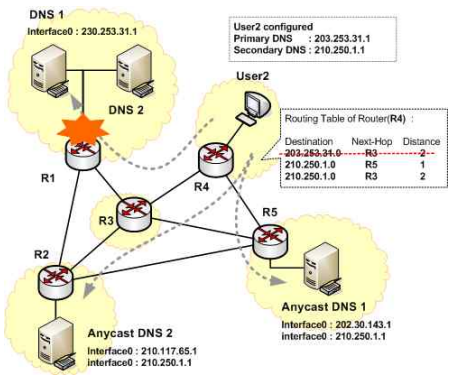
DNS 서비스를 제공받는 클라이언트는 DNS 서비스에 어떠한 문제가 발생했을 때, 스스로가 그 문제에 대해 인지하고 적절히 대체하기는 어렵다. 제안하는 DNS의 서비스 신뢰성 향상 기법은 1차 DNS 서버가 위치하고 있는 네트워크에 문제가 발생하였을 경우, Anycast IP 주소를 적용한 DNS 서버를 이용하여 자동으로 최선의 접근성을 가지고 있는 다른 네트워크의 DNS 서버를 2차 DNS 서버로 전환하여 서비스 가용성을 유지한다. 이를 통해 클라이언트는 부수적인 설정없이 DNS 서버 장애 발생 시에도 서비스를 유지할 수 있다.

3.1.1 Recursive DNS 서버의 Anycast IP 할당

본 논문에서 제안하고자하는 안전한 DNS 서버의 구성을 위해서 Recursive DNS 서버가 Anycast DNS 서버로 사용될 수 있도록 한다. 이를 위해, 해당 DNS 서버의 네트워크 인터페이스에는 기존의 IP 주소를 유지한 상태로 Anycast IP 주소를 추가 할당한다. Anycast IP 주소가 할당된 Recursive DNS 서버는 결국 두 개의 IP 주소를 가진다. 이 중 Global Unicast 주소는 1차(Primary) DNS 서버의 역할을 가지는 IP 주소로 기존 DNS 서버의 서비스를 유지하는데 사용하며, Global Anycast 주소는 2차(Secondary) DNS 서버로의 이용을 위한 IP 주소로 Anycast DNS 서버로써의 서비스를 위해 사용한다. 이때, 할당되는 Anycast 주소는 예약된 주소 범위 내에서 선택하게 되는데, Anycast 서비스를 위해 예약된 주소 범위를 사용하기 위해서는 IP 주소의 관리 기관인 ICANN을 통해 해당 주소 범위의 예약이 필요하다. 이것은 앞으로 정의될 수 있는 Anycast 주소를 활용한 다양한 응용 프로그램 및 서비스를 위한 목적으로 예약될 수 있을 것이다. 하나의 인터페이스에 두 개의 IP 주소를 할당하는 방법은 IP 주소체계에 따라 다르다. IPv4 환경에서는 IP Aliasing('multi-homing') 기술[6]을 통해 각각의 IP 주소를 할당할 수 있고, IPv6 환경에서는 'IP Version 6 Addressing Architecture'[7]을 통해 다수 IP 주소의 사용을 허용하고 있다.

3.1.2 Anycast DNS 서버를 이용한 DNS 서버 구성

1차 DNS 서버 설정에 기존의 Global Unicast 주소 설정을 유지하고 2차 DNS 서버 설정에 예약된 범위의 Anycast IP 주소를 모두 할당한 Recursive DNS 서버들을 적용한 DNS 서버 구성은 기존의 DNS 환경인 'Object Impermanence' 방식의 단점으로 지적되었던 네트워크 문제를 해결한다.



<그림 1> Anycast DNS 서버를 이용한 DNS 서버 구성

<그림 1>과 같이 User2에 1차 DNS 서버로 설정되어 있던 DNS1 서버에 직접적인 문제가 발생하거나, DNS1 서버가 속한 R1의 네트워크에 문제가 발생한다면 User2는 2차 DNS 서버로 설정되어 있는 Anycast IP 주소로 DNS 패킷을 전송한다. 이 DNS 패킷은 Anycast 라우팅에 의해 R4 라우터의 라우팅 프로토콜상 가장 근접한 Anycast DNS1 서버(210.250.1.1)로 서비스가 전환된다.

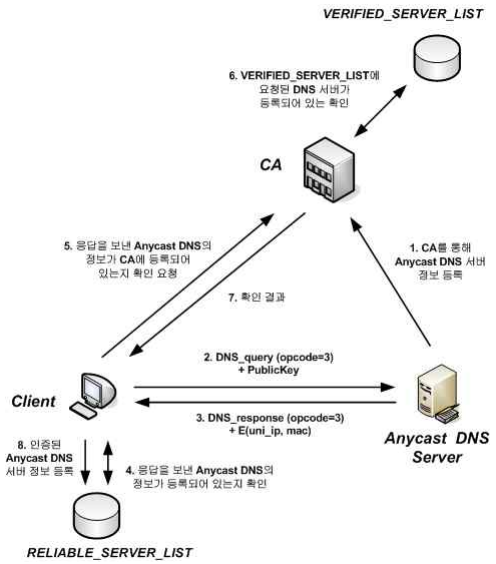
제안하는 DNS 서버 구성에서 Anycast DNS 서버를 1차 DNS로 설정한다면 2차 DNS 서버 설정과정 없이도 이용할 수 있고, 전체적으로 DNS 서버 설정을 간소화 할 수 있다. 하지만 클라이언트의 DNS 서버 설정을 Anycast DNS 서버의 주소로 할당하게 되면 클라이언트나 DNS 서비스를 제공자의 의도와 상관없이 클라이언트는 라우팅 경로상의 가장 가까운 DNS 서버의 자원을 사용하게 된다. 이는 DNS 서비스 제공자 입장에서는 의도하지 않은 클라이언트가 서비스 제공자의 DNS 시스템 자원을 어떠한 허가 절차도 없이 소모하게 되는 결과를 가져온다. 또한 서비스 제공자로부터 안정적이고 높은 성능의 서비스를 받아야 하는 클라이언트는 허가 절차 없이 접근한 다른 클라이언트들로 인해 DNS 서버와 서비스 제공자의 네트워크 자원을 나누어 사용하게 된다. 이러한 문제점은 많은 사용자를 보유하고 뛰어난 접근성과 질 높은 네트워크 서비스를 제공하는 서비스 제공자의 DNS 서비스 일수록 빈번하게 발생한다. 따라서 본 논문은 Anycast DNS 서버를 클라이언트의 2차 DNS 서버로 설정하여 위의 문제를 해결하였다.

3.2 서비스 전환 과정의 Anycast DNS 서버 인증 기법

서비스 전환 과정에서 선택된 Anycast DNS 서버는 라우팅 프로토콜상의 라우팅 거리 참조를 통해 임의적으로 선택된 서버이다. 그렇기 때문에 Anycast DNS 서버로 연결된 클라이언트는 선택되어진 서버가 정상적인 DNS 서비스를 제공하던 서버인지, 또는 악의적인 목적을 가진 사용자에 의해 설치된 서버인지를 구분할 수 없다. 이를 방지하기 위해 본 논문에서는 Anycast 주소 라우팅에 의해 선택된 DNS 서버에 대한 인증 기법을 제안한다. 또한, 제안하는 Anycast DNS 서버 인증 기법의 적용을 위하여 Anycast DNS 서버의 서비스를 받게 될 클라이언트에 Internal table('hosts' file) 외에 RELIABLE_SERVER_LIST를 추가하고, Anycast DNS 서버와의 최초 연결에 확장된 DNS 패킷을 사용할 것을 제안한다.

3.2.1 제안하는 Anycast DNS 서버 인증 구조

본 논문에서 제안하는 Anycast DNS 인증 기법의 전체적인 구조는 <그림 2>와 같다.



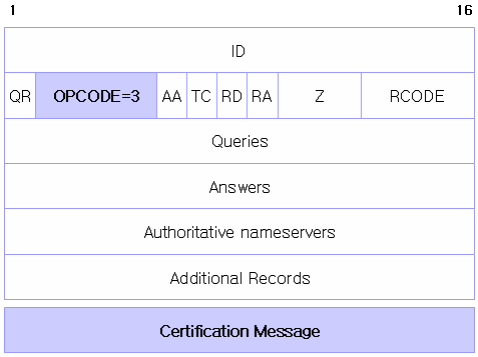
<그림 2> Anycast DNS 서버 인증 구조

Anycast DNS 서버로 사용될 서버는 서비스를 활성화하기 전에 CA에 서버의 정보를 등록하고 등록된 정보는 CA에 의해 관리된다. 클라이언트가 Anycast DNS 서버로 질의를 시도하고, 서버로부터 응답을 받는다. 이 과정에 클라이언트는 Anycast DNS 서버의 정보를 함께 받고 신뢰할

수 있는 서버인지 클라이언트에 저장되어 있는 리스트를 통해 확인한다. 클라이언트는 확인되지 않는 Anycast DNS 서버의 정보에 대해 CA로 문의하고, CA에 등록된 Anycast DNS 서버임이 확인되면 DNS 질의를 계속하게 된다.

3.2.2 DNS 패킷의 확장

제안 하는 인증 기법은 Anycast DNS 질의에 대해서만 적용되므로, 일반 DNS 질의와 Anycast DNS 질의를 구분해야 한다. 두 가지 질의를 구분하기 위해 기존 DNS 패킷에 정의되어 있는 OPCODE 필드를 '3' 값으로 설정한 패킷을 사용한다.



<그림 3> 확장된 DNS 패킷의 구조

위의 값은 현재 Reserved되어 있으며 일반 DNS 질의와 Anycast DNS 질의를 구분하는 목적으로만 사용한다. 그리고 클라이언트와 Anycast DNS 서버 사이의 최초 질의/응답 시 Anycast 기술에 의해 선택된 DNS 서버를 인증하기 위해 기존의 DNS 패킷의 마지막 위치에 Certification Message 필드를 <그림 3>과 같이 확장하여 사용한다.

확장된 Certification Message 필드는 클라이언트에서 DNS 서버로 질의 시 클라이언트의 공개키가 저장되며, Anycast DNS 서버에서 클라이언트로 응답 시 클라이언트의 공개키로 암호화된 Anycast DNS 서버의 Unicast IP 주소와 MAC을 포함한다. 이 인증 과정은 <그림 4>와 같은 알고리즘이 사용되고, 클라이언트는 Certification Message 필드를 통해 받은 정보를 Anycast DNS 서버의 인증에 사용한다.

<표 1> 함수 정의

함수	정의
received	특정 메시지가 전송되어 옴
search	DNS 서버 리스트를 검색
generate	기본 패킷을 생성
set	해당 필드를 설정
encrypt	공개키를 이용하여 데이터를 암호화
associate	DNS 패킷을 조합
send	DNS 패킷을 전송
decrypt	공개키를 이용하여 데이터를 복호화

```
procedure Response_Host_Address (HostAddress, ServerMAC,
                                 DomainName, opcode, CertMessage)
input  DNS_query
output DNS_response

if received DNS_query from User_Host
begin
    search ServerInfo from Name_Server where DomainName
    generate DNS_header
    set DNS_header = ServerInfo
    if ( opcode == 3  && CertMessage )
        encrypt ECertMessage ( ServerMAC +
                                        ServerUnicastAddr )
        set opcode = 3 in DNS_header
        associate DNS_response ( DNS_header, ECertMessage
                        ( ServerMAC + ServerUnicastAddress ))
        send DNS_response to UserAddress
    else
        set opcode = 0 in DNS_header
        associate DNS_response ( DNS_header )
        send DNS_response to UserAddress
end
```

<그림 4> Anycast DNS 서버가 호스트의 질의에 응답

3.2.3 Anycast DNS 서버 인증을 위해 사용된 데이터베이스

Anycast DNS 서버를 통해 도메인 네임 서비스를 제공하고자 하는 사업자는 해당 Anycast DNS 서버의 정보를 CA의 데이터베이스(VERIFIED_SERVER_LIST)에 등록 한다. CA는 등록을 요청해온 Anycast DNS 서버들의 정보를 보관하고, 클라이언트로부터 해당 DNS 서버의 인증을 요청 받았을 때, 해당 서버가 신뢰할 수 있는

서버인지를 확인하여 응답한다. 이 과정에서 CA의 VERIFIED_SERVER_LIST가 사용되며 그 구조는 <그림 5>와 같다.

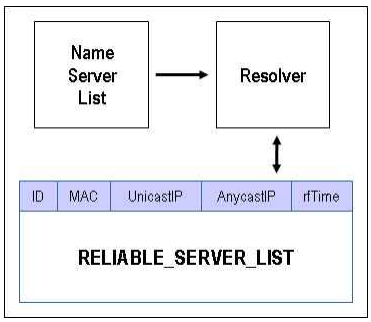
<그림 5> CA의 VERIFIED_SERVER_LIST

<표 2> 데이터베이스 필드 정의

함수	정의
ID	데이터베이스의 인덱스
MAC	Anycast DNS 서버 인터페이스의 MAC
UnicastIP	Anycast DNS 서버의 Unicast IP 주소
AnycastIP	Anycast DNS 서버의 Anycast IP 주소
AuthorityID	정상적인 서비스 제공자임을 확인하기 위한 코드
Date	Anycast DNS 서버가 CA에 등록된 시간
Reserved	추가 서비스를 위해 예약된 공간

서비스 전환 후 클라이언트는 Anycast DNS 서버와의 첫 번째 질의를 통해 서버의 인증 과정을 수행한다. Anycast DNS 서버로부터 획득한 응답 메시지 패킷에 포함된 서버의 정보를 통해, 해당 서버는 클라이언트가 유지하고 있는 RELIABLE_SERVER_LIST에 존재 하는지 여부를 확인한다.

RELIABLE_SERVER_LIST는 각각의 클라이언트가 유지하는 신뢰할 수 있는 서버들의 리스트로 <그림 6>과 같은 형식을 가지며, 리스트에 데이터가 존재한다면 연결된 Anycast DNS 서버는 신뢰할 수 있음을 의미한다. 클라이언트가 자체 데이터베이스를 통해 확인할 수 없는 서버라면, 클라이언트는 지정된 CA로 연결된 Anycast DNS 서버의 정보를 보내 인증을 요청한다. 인증 요청에 대한 CA의 응답을 통해 Anycast DNS 서버가 신뢰할 수 있음을 확인하면, 클라이언트는 RELIABLE_SERVER_ LIST에 인증된 Anycast DNS 서버의 정보를 기록하고 통신을 시작한다. CA를 통해서도 Anycast DNS 서버의 신뢰성을 확인할 수 없다면, 해당 패킷을 버리고 클라이언트에 설정된 Anycast IP 주소로 다시 DNS 서버의 연결을 시도한다.



<그림 6> Client의 RELIABLE_SERVER_LIST

Ⅳ. 성능 평가 결과

본 논문에서 제안하는 구성과 인증 기법을 평가하기 위하여 모의 실험을 통해 효율성과 신뢰성을 분석하고 기존의 DNS 서비스 구성 환경과 비교한다. 효율성은 DNS 패킷 교환의 최소 소요시간 증가로 DNS 서비스를 유지할 수 있는지 Ethereal을 통해 측정하였고, 신뢰성은 Anycast 라우팅으로 선택된 임의의 서버에 대해 인증할 수 있는지 구현물을 통해 평가한다.

4.1 실험 결과 및 분석

4.1.1 Anycast DNS 서버 적용 환경

사용자 PC에 설정되어 있는 DNS 서버들이 정지되었을 때, 클라이언트에서 재질의 과정을 알아보기 위해 100회의 질의를 통해 그 패턴을 <표 3>으로 나타내었다. 이를 통해 클라이언트는 매 질의마다 설정된 DNS 서버 목록 내에서 서버를 변경한다는 것을 확인할 수 있다.

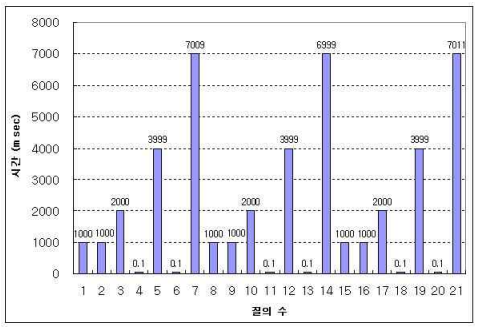
따라서 설정한 Anycast DNS 서버가 속한 Anycast 그

룹의 구성원의 수가 6 이상일 경우, 그룹 내에서 최대 6대의 다른 Anycast DNS 서버들에 임의로 질의를 전송하며 해당 그룹의 모든 Anycast DNS 서버가 사용할 수 없는 상황이라면 해당 질의는 실패임을 확인하게 된다.

<표 3> WindowsXP 환경에서 재질의 시 Timeout 값

n	Timeout(msec)	Domain	Destination
1	1000	www.ssu.ac.kr	Primary
2	1000	www.ssu.ac.kr	Secondary
3	2000	www.ssu.ac.kr	Primary
4	0.1	www.ssu.ac.kr	Primary
5	3999	www.ssu.ac.kr	Secondary
6	0.1	www.ssu.ac.kr	Primary
7	7009	www.ssu.ac.kr	Secondary

<표 3>를 그래프로 나타내면 <그림 7>과 같고, <표 4>를 통해 <표 3>의 7번째 질의에서의 Timeout 시간이 질의 구조를 바꿔서 재전송하는 데에 소비되는 시간임을 확인할 수 있다.



<그림 7> WindowsXP 환경에서 DNS 재질의 시 Timeout값 패턴

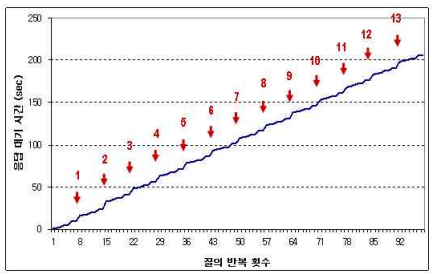
Object Impermanence 환경에서 클라이언트의 운영체제는 DNS 질의가 실패할 경우 운영체제가 질의의 형태를 <표 4>와 같은 형태로 바꾸어 재전송을 시작한다.

<표 4>에서 보는 바와 같이, 총 7가지 형태의 도메인명으로 IPv4/IPv6 도메인네임 포맷으로 질의를 바꾸어 가며 질의를 성공 할 때까지 시도한다. 하나의 질의 형태

<표 4> DNS 질의 실패 후 질의 형태의 변화

k	Domain	Format
1	www.ssu.ac.kr	A
2	www.ssu.ac.kr	AAAA
3	auto.search.msn.com	A
4	auto.search.msn.com	AAAA
5	www.www.ssu.ac.kr.co.kr	A
6	www.www.ssu.ac.kr.co.kr	AAAA
7	www.www.ssu.ac.kr.com	A
8	www.www.ssu.ac.kr.com	AAAA
9	www.www.ssu.ac.kr.org	A
10	www.www.ssu.ac.kr.org	AAAA
11	www.www.ssu.ac.kr.net	A
12	www.www.ssu.ac.kr.net	AAAA
13	auto.search.msn.com	A
14	auto.search.msn.com	AAAA

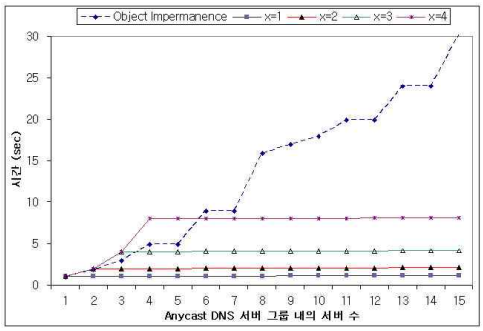
는 다시 <표 3>의 과정을 거치므로 클라이언트에 설정된 DNS 서버의 서비스가 정지되었을 때, 사용자는 최대 7 * 14 * Timeout의 질의에 대한 응답 대기시간을 가지게 된다. 그리고 이러한 대기시간은 <그림 8>을 통해 확인할 수 있듯이 약 200초의 응답 대기시간을 사용자에게 돌려준다.



<그림 8> Object Impermanence 환경에서 사용자의 응답 대기시간

위와 같은 환경에서 Secondary DNS 서버를 Anycast DNS 서버로 적용하여 사용자의 응답 대기시간을 줄일 수 있다. Anycast 그룹의 주소로 설정된 Secondary DNS 서버는 Primary DNS 서버의 서비스가 정지되면 Secondary DNS 서버로 설정된 Anycast 주소를 할당받

은 서버 중 라우팅 거리상 가장 가까운 서버를 선택하여 서비스를 시도하며, 이 시도가 실패한다면 같은 Anycast 그룹의 주소를 가지는 다른 Anycast DNS 서버를 찾아 질의를 시도하게 된다.



<그림 9> Anycast DNS 서버 적용을 통해 단축되는 응답 대기시간

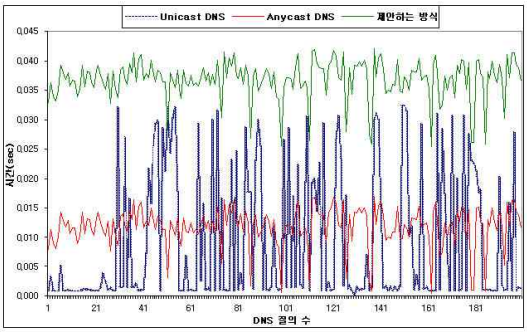
<그림 9>의 그래프를 통해, 클라이언트에 설정된 DNS 서버의 서비스가 정지하였을 때, Object Impermanence 환경보다 Anycast DNS 서버 적용 환경의 예상되는 응답 대기시간이 짧다는 것을 알 수 있다. x값은 서비스 가능한 Secondary Anycast DNS를 만나게 되는 홉 수이다. Secondary로 설정된 Anycast 주소를 가진 서버 중 선택된 첫 번째 서버가 서비스 가능하다면 x=1, 첫 번째로 선택된 서버 역시 서비스를 할 수 없는 상태라 두 번째로 서비스 전화된 서버를 이용하게 되는 상황의 그래프가 x=2의 그래프이다.

Object Impermanence 환경에서 설정된 DNS 서버들이 서비스를 할 수 없을 경우, 총 98회의 질의를 수행한 후 해당 서비스를 유지할 수 없음을 확인하게 되는 반면, Anycast DNS를 적용한 것은 서비스 중인 다른 Anycast DNS 서버를 만나는 시점까지만 질의를 하여 질의 횟수가 눈에 띄게 감소하며, 기존 환경처럼 설정된 소수의 DNS 서버를 대상으로 하는 질의가 아닌, 동일한 Anycast 주소를 사용하는 다수의 DNS 서버로 질의를 시도함에 따라 서비스 중인 DNS 서버를 찾을 확률을 높일 수 있다.

4.1.2 제안하는 인증 기법 실험

인증 기법 과정에 소요되는 시간의 평균값을 측정하여 25msec의 시간 증가를 측정하였다. Ethereal을 통해 600회 측정한 Anycast DNS 적용 환경의 질의 시간 그래프에 인증 기법에 소요되는 평균 시간을 더하여 그래프로 나타내어 보면 <그림 10>과 같은 패턴으로 나타나는 것을 알 수 있다.

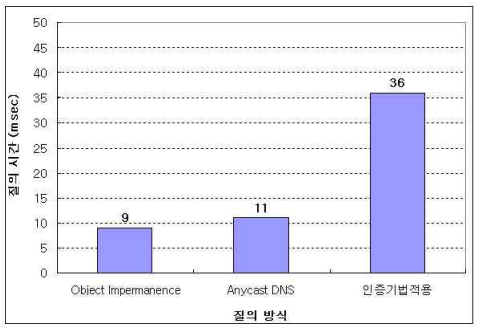
일반적으로 DNS 서비스에 문제가 없는 상황에서 비교하였을 때, Unicast DNS 서버를 사용하는 Object Impermanence 환경보다는 Anycast DNS 서버를 사용하는 방식이 평균 2msec의 늦은 반응속도를 보여주었지만 대체적으로 고른 질의 시간을 보여주었고, Unicast DNS 서버는 목적지가 되는 사이트에 따라 심한 편차를 보여주었다. 그리고 인증 기법이 적용된 그래프는 Unicast DNS 서버의 응답시간에 평균 27msec가 증가하였다.



<그림 10> DNS 서버 주소 방식에 따른 질의 시간

4.1.3 실험 결과 분석

실험 결과는 기존의 Object Impermanence 환경에서 외부 공격으로 인해 Primary, Secondary DNS 서버가 사용할 수 없는 상황이 되었을 때, DNS 질의 소요시간을 최소로 증가시키면서 DNS 서비스의 가용성과 선택된 DNS 서버의 신뢰성을 유지할 수 있는가를 측정하여 결과를 <그림 11>로 나타내었다.



<그림 11> 질의 방식에 따른 질의과정의 평균 소요 시간

인증기법이 적용되는 환경인 DNS 서비스의 유지에 문제가 있는 상황에서 Unicast DNS 서버가 200초 이상의 응답 대기시간을 가지게 된다는 점에 비추어 볼 때, 제안하는 기법이 27msec의 체감하기 어려운 시간의 증가로 DNS 서비스를 유지하면서 Anycast 라우팅으로 임의로 선택된 DNS 서버의 신뢰성을 확보할 수 있다는 것을 의미한다.

Ⅴ. 결론

DNS(Domain Name System)는 인터넷 상의 도메인 네임을 그에 해당하는 IP 주소로 전환하는 역할의 기술로 현재 대부분의 인터넷 응용 프로그램을 통한 서비스가 이루어지기 위해 반드시 필요한 분산 데이터베이스로서 그 중요성이 커지고 있다. 그러나 현재 DNS 서버의 서비스 중단을 방지하기 위한 기술인 Object Impermanence 기술은 네트워크를 목적으로 하는 공격이나 DDoS 공격 등으로 백업 DNS 서버까지 서비스를 제공할 수 없는 상황에서는 적절하게 대응할 수 없다.

이런 문제를 해결하기 위해 본 논문에서는 Anycast 주소체계를 이용하여 DNS 서버를 지역적으로 분산하고 그룹화하며, 서버에 문제가 발생했을 경우 그룹 내의 다른 서버로 전환할 수 있도록 하였다. 그리고 그룹 내에서 임의로 선택되는 DNS 서버가 신뢰할 수 있지를 확인하기 위해 CA와 RSA 알고리즘을 이용한 인증을 위한 기법을 제안하였다. 그리고 제안한 기법의 성능분석을 위해 구현물을 이용한 테스트와 패킷 분석을 이용하여 기존의 방식에서 DNS 서비스의 성능을 최대한 유지하면서 생존력을 향상 시켰음을 증명하였다.

본 논문에서 제안하는 DNS 서버 구성과 인증 기법을 바탕으로 평상 시 DNS 서비스의 성능을 유지하면서 DNS 서비스의 위험 상황에서 더욱 안전한 DNS 서비스를 제공할 수 있을 것이다. 따라서 향후에는 본 제안에 대해 보다 확장된 범위의 다양한 환경에서 Anycast 기술을 이용한 신뢰적 향상 메커니즘을 적용하였을 경우의 성능에 대한 연구가 필요할 것이다.

참고문헌

1] 최영현 · 박민우 · 엄정호 · 정태명, “Inter-MAG이 고려된 PMIPv6 환경에서 전달자 정보를 이용한 경로 최적화 기법에 관한 연구,” 디지털산업정보학회 논문지, 제6권, 제3호, 2010, pp.58-68.

[2] Praveen Yalagandula, “A survey of DNS,” October 4, 2000.

[3] Kevin Miller, “Three Practical Ways to Improve Your Network,” in proc. Large Installation Systems Administration Conference San Diego, CA, USA October 26, 2003, pp.101-111.

[4] E. Zegura, M. Ammar, Z. Fei and S. Bhattacharjee, “Application-Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service,” ACM/IEEE Transactions on Networking, Vol.8, No.4, 2000, pp.455-466.

[5] C. Partridge, T. Mendez, and W. Milliken, “Host Anycasting Service,” RFC1546, November, 1993.

[6] Niall Mansfield, “IP aliasing (“multi-homing”) on Linux,” addison Wesley Professional 2003.

[7] R. Hinden, and S. Deering, “IP Version6

Addressing Architecture," RFC4291, February, 2006.

■ 저자소개 ■

김 보 승
Kim, Bo Seung

2005년~현재 숭실대학교 컴퓨터학과 박사과정
2004년 숭실대학교 컴퓨터학과 공학석사
2002년 영동대학교 컴퓨터공학과 공학사

관심분야 : 멀티캐스트, IPTV, 센서네트워크, IPv6, DNS, 홈네트워크
E-mail : kdwon2002@ssu.ac.kr

김 정 재
Kim, Jeong Jai

2010년 (주) RetailTech 수석 연구원
2005년 숭실대학교 컴퓨터학과 공학박사
1999년 숭실대학교 컴퓨터학과 공학석사
1995년 영동대학교 컴퓨터공학과 공학사

관심분야 : 멀티미디어 보안, 멀티미디어 데이터베이스, DRM, RFID, 멀티미디어 통신
E-mail : argniss@ssu.ac.kr

김 경 민
Kim, Kyung Min

2007년~현재 (주)디지캡 연구원
2007년 숭실대학교 컴퓨터학과 공학석사
2004년 한경대학교 컴퓨터공학과 공학사

관심분야 : IPTV, DRM, CAS, DNS, 멀티캐스트
E-mail : kmkim@digicaps.com

박 찬 길
Park, Chan Kil

2010년~현재 한국사이버대학교 정보보안학과 교수
2009년 한성디지털대학교 멀티미디어학과 교수
2006년 숭실대학교 컴퓨터학과 공학박사
1995년 서울과학기술대학교 컴퓨터공학과 공학석사
1991년 서울과학기술대학교 컴퓨터공학과 공학사
2004년~현재 (사)디지털산업정보학회 이사

관심분야 : 네트워크보안, 유비쿼터스, DRM,
E-mail : ckpark@mail.kcu.ac

신 용 태
Shin, Yong Tae

1995년~현재 숭실대학교 컴퓨터학부 교수
1994년 Univ. of Iowa 전산학과 공학박사
1990년 Univ. of Iowa 전산학과 공학석사
1985년 한양대학교 산업공학과 공학사

관심분야 : 멀티캐스트, RFID/USN, IPTV, DRM
E-mail : shin@ssu.ac.kr

논문접수일 : 2010년 11월 27일
수 정 일 : 2010년 12월 5일
게재확정일 : 2010년 12월 9일