

IPv6 용 Web Server 의 구현

조성표*, 박형배*, 이영희*
*한국정보통신대학원대학교 공학부

e-mail : spcho@icu.ac.kr

The Implementation of IPv6 Web Server

Sung-Pyo Cho*, Hyoung-Bae Park*, Young-Hee Lee*
*School of Engineering, Information and Communications University

요 약

인터넷 주소의 고갈문제 때문에 현재의 IPv4 는 IPv6 로 점진적으로 바뀌어 나갈 것이다. 그러나 이에 따른 IPv6 용 Application 에 대한 연구는 거의 없는 실정이다. 이 논문에서는 IPv4 용 Web Server 를 이용해서 IPv6 용 Web Server 를 구현함으로써 IPv4 code 를 IPv6 code 로 변경하는 과정을 설명하였고 IPv6 용 서버를 구축해서 IPv4 를 IPv6 로 변경했을 때의 성능변화를 분석하고 IPv6 code 에서 일어날 수 있는 성능 저하 요인을 찾아보았다.

1. 서론

최근 인터넷은 점차 그 망이 확대되고 많은 사용자를 가짐에 따라 보다 많은 요구를 충족시켜 주어야만 하는 상황에 처했다. 이런 변화하는 환경에서 인터넷 연동의 근간 프로토콜로 이용되던 IP 는 망의 발전 속도가 인구의 증가 속도를 넘어섬에 따른 주소 공간의 부족이 예상되고, 또 복잡한 망에 의한 경로 설정의 어려움, 보안 기능의 부족, 다양한 멀티미디어 서비스 부족, bandwidth 의 제한 등의 약점으로 인해 거의 수명이 다해가고 있는 상황이다.[1]

특히 주소 공간 부족 문제는, 현재 인터넷인 IPv4(Internet Protocol version 4)는 인터넷 주소를 네트워크 규모에 따라 A Class, B Class, C Class 로 나누고 특별한 용도로 예약해 둔 D Class 로 구분하는 (A Class 주소의 경우에는 약 16,777,214 대의 컴퓨터가 인터넷에 연결될 수 있으며, B Class 의 주소에는 약 65,534 대, C Class 에서는 약 254 대의 컴퓨터가 인터넷에 연결될 수 있다.) 32 비트로 구성된 네트워크 주소 체계를 가지고 있고 인터넷을 사용하는 기관에서는 할당 받은 주소를 모두 사용하는 경우가 거의 없고 사용하고 남은 주소를 예비용으로 보유하는 것이 대부분이어서, 인터넷의 상업화에 따른 사용 기관의 급격한 증가로 인하여 한정된 인터넷 주소의 고갈 우려와 함께 일부 기업을 중심으로 인터넷 주소 확보에 대한 가수요까지 발생하는 등, 가장 심각하고 중대한 문제가

다.[2] 그러나, 미국을 비롯한 선진국은 주소고갈 문제에 있어서 다소 시간적 여유가 있을 정도로 많은 주소를 확보하고 있는 반면, 우리나라를 비롯한 아시아 개발 도상국은 그렇지 못한 사정이기 때문에 이 문제는 더욱 심각할 수 있다.

이러한 여러 가지 문제를 해결하기 위해 IETF 는 최근 IPv6 를 중심으로 차세대 인터넷 (IPng: IP next generation)에 관한 연구를 하고 있다. IPng 는 IPv4 보다 4 배가 되는 128 비트의 주소 크기를 가지기 때문에 1m² 당 1000 개 정도의 IP 를 부여할 수 있게 되어 기존의 주소 부족 문제를 완전히 해결할 수 있게 되었다. 또 간단한 헤더 구조를 가지므로 해서 Packet 의 처리 효율을 높이고, 헤더가 차지하는 대역폭의 비용을 절감하는 효과를 가져오게 되었다. 이 밖에도 기본 헤더와 분리된 extension 헤더를 통하여 다양한 추가 기능을 제공하고, flow labeling 을 통하여 QoS 에 관한 문제를 보장하였고, 거의 전무하던 보안과 인증 기능을 더하였다.[3]

즉, 차세대 인터넷은 보다 다양한 멀티미디어 서비스는 물론, 멀티캐스트를 이용한 서비스 그리고 서비스 품질을 보장하는 보다 안정된 서비스 등 현재의 인터넷의 한계를 넘어서는 고차원적인 네트워크라 할 수 있다.

그러나, 기존의 IPv4 에 대한 여러 가지 문제점을 해결할 수 있는 대안인 차세대 인터넷으로 가는 길은 그리 쉽지 않다. 그 이유는 현재, 전세계에 퍼져있는

수 많은 IPv4 의 망 사용자들이 한 순간에 차세대 인터넷 망 혹은 IPv6 망으로 옮겨가지는 않을 것이며, 또한 그러한 작업이 그리 쉽게 이루어 지지는 않을 것이기 때문이다. 즉, 기존의 틈으로 IPv6 가 들어갈 자리를 마련하고 있지 못하며, 또한 IPv6 가 IPv4 에 비해 차등화 된 서비스를 어떻게 제공해 줄 것인지에 대한 문제인 IPv6 의 Killer Application 개발이 시급히 요구되는 상황에 처해 있다. 따라서 IPv6 를 위시한 차세대 인터넷의 성공 여부는 현재 불투명한 상태이다.

일반적인 관점에서는, IPv4 망이 진화적으로 IPv6 망으로 대체되어가는 시나리오를 가정하고 있는데, 이러한 과정에서 필요한 IPv6 용 Application 에 대한 연구는 거의 없는 실정이다.

이 논문에서는 IPv4 용 Web Server 의 IPv6 용 Web Server 로의 전환 과정을 보임으로 해서 앞으로 있을 IPv6 용 Application 의 구현에 도움을 주고 성능 평가를 통해 IPv4 용 Application 과 IPv6 용 Application 의 성능의 차이를 비교해보았다. 이해를 쉽게 하기 위해 2 장에서는 각각의 연구방향을, 3 장에서는 IPv4 Web Server 를 IPv6 Web Server 로 바꾸는 과정을 단계별로 설명하였고, 4 장에서는 구현된 IPv6 Web Server 와 원래 IPv4 Web Server 사이의 성능차이를 비교해보았고 마지막 5 장에서 결론을 맺었다.

2. 관련 연구

앞에서 언급했던 IPv4 의 문제를 해결하기 위해 각국은 차세대 인터넷 망에 대해 활발히 연구를 하고 있다. [4]

1996 년 10 월에 시작된 미국의 차세대 인터넷 개발 전략은 2 단계로 나뉜다. 1 단계는 미국의 주요 대학들의 네트워크 접속환경을 개선함으로써 대학에서의 차세대 인터넷 기술개발에 박차를 가하여 에블린(Abilene) 불리는 새로운 인터넷 망을 구성하는 Internet2 프로젝트이고, 2 단계는 클린턴 행정부가 미국 내 정보고속도로의 조기 구축이라는 강력한 의지를 갖고 추진하겠다는 차세대 인터넷 프로그램(NGI)이다. 현재 Internet2 프로젝트에는 122 개 대학 및 연구소, 25 개의 파트너 기업, 12 개의 비영리 연구단체, 그리고 미정부가 참여하고 있다. 미국정부의 구상은 에블린을 완성한 뒤 NGI 구축에 본격적으로 나선다는 것이다. 이를 위해 우선 1998 년에 8 천만 달러를 지원한 뒤 향후 5 억 달러를 더 투입할 계획으로 있다. 이외에도 미국은 국회의원 등을 위하여 차세대 인터넷의 다양한 응용 서비스들을 보여주고, 정책결정이나 Funding 을 얻기 위해 Highway1 이라는 데모 설비를 운영하고 있다.

캐나다에서는 차세대 인터넷을 위해 CANARIE(Canadian Network for the Advancement of Research, Industry and Education)가 1997 년 6 월 CA*netII 를 구축하여 여러 가지 응용 서비스들과 기술들을 테스트하고 있다. 기존의 인터넷 백본망은 CA*net 이며, CA*netII 는 차세대 상용 인터넷 서비스

들의 개발을 촉진하고, 대학과 정부 연구소 및 CANARIE 로 부터 허가받은 업체들의 연구를 돕기 위한 망 서비스를 제공한다. 특히, CA*netII 는 오디오 비디오와 같은 멀티미디어 정보를 전송가능 하도록 해 주고, 특정 응용 서비스들에 서비스 품질을 보장해 줄 수 있다. CANARIE 는 1993 년에 구성된 컨소시엄이며, 현재는 140 여 멤버가 가입되어 있고 정부로부터 지원을 받고 있다. 여기에는 15 개의 RAN(Regional Advanced Network) 지역망이 GigaPOP 을 통해 CA*netII 에 접속된다. GigaPOP 은 BGP 라우팅, caching 서버, Mbone, 보안, IP 전화 등을 제공한다. 이상과 같은 CA*netII 이외에도 CANARIE 에서는 통신망 기반 시설로써 CA*netIII 를 1999 년에 구축할 예정으로 있다.

유럽의 경우 EC(European Commission)가 지원하고 영국의 DANTE 가 운영하는 TEN-34(Trans-European network interconnecting at 34 Mbps) 프로젝트가 유럽내의 각 연구 기관들을 초고속 망으로 연동시키기 위한 목적으로 진행되고 있다. 여기에는 오스트리아의 ACONET, 벨기에의 BELNET, 체코의 CESNET, 프랑스의 RENATER, 독일의 DFN, 그리스의 GSRT, 헝가리의 HUNGARNET, 룩셈부르크의 RESTENA, 네덜란드의 SURFnet, 포르투갈의 FCCN, 스페인의 RedIRIS, 스위스의 SWITCH, 영국의 UKERNA, 덴마크 등의 NORDUnet 등이 참여하고 있다. TEN-34 에서 시험되고 있는 기술들은 TCP/UDP over high-speed and long delay, SVC(Switched Virtual Connection), ARP(Address Resolution Protocol), NHRP(Next Hop Routing Protocol) 등의 테스트, 주소문제, 망관리, IP over VBR (Variable Bit Rate), RSVP, ATM 보안문제 등의 기술이 있다.

한국, 일본을 중심으로 한 아시아/태평양 지역에서 21 세기 차세대 인터넷 사업을 진행하고 있다. 여기에는 정부차원에서 진행하고 있는 APII(Asia-Pacific Information Infrastructure) 사업과 학계/연구 기관을 중심으로 추진중인 APAN(Asia-Pacific Advanced Network Consortium) 사업이 있다. 아/태지역 정보통신기반(APII) 협력사업은 아태경제협력체(APEC, Asia Pacific Economic Cooperation) 회원국들의 정보통신장관회의를 통하여 추진 중에 있으며, 이중에서도 초고속 선도 시험망(APII Test-bed) 사업은 우리나라가 주축이 되어 일본, 싱가포르, 중국 등과 추진 중에 있다. APAN 은 1997 년 6 월에 아태지역의 학계/연구기관을 중심으로 하여 여러 가지 응용 서비스 및 기술들을 시험하고 국제적인 협력사업을 위해 현재의 국제적인 인터넷 망보다 향상된 초고속 네트워크 환경을 제공할 목적으로 구성되었다.

현재 APAN 에서 시험하고 있는 기술들은 IPv6 를 위한 6bone, Mbone, RSVP, Security, Measurement, Design, Cache 등이 있으며, RSVP 의 경우 IPv4 및 IPv6 상에서 구현되고 있다. 또한, RSVP 는 ATM 서비스 클래스와의 매핑을 통해 이더넷 망(ethernet network)과 ATM 망과의 연동을 시도하고 있다.

응용 서비스로는 BioInformatics, Education, Engineering, Environment, Medical, Science, MultiMedia,

Immersive, Design and Manufacturing, Satellite 등이 있다. Education 의 경우 가상대학(virtual university)이 계획되고 있으며, Medical Informatics 의 경우 일본과 싱가포르 사이에 Telemedicine Collaboration 이 준비되고 있다.

APAN 과 관련하여 국내의 동향으로는 APAN-KR 조직이 1997년 4월부터 논의가 시작되어 1998년 2월에 공식적으로 출범하였으며, 이를 통해 여러 가지 국제적인 협력사업이 기술 및 응용 서비스들을 중심으로 학계/연구계에서 이루어지고 있다.

국내에 IPv6 망의 도입 단계로 아직 기존 IPv4 망과의 연동에 관한 실질적인 연구나 투자가 부족한 현실이다. 국내의 IPv6 도입에 대한 견해는 전문가들마다 서로 다르고, 그 사용시기에 대한 예상과 인식 부족으로 인해 기존의 인터넷과의 연동에 대한 고려보다는 IPv6 망의 시험망을 구축하는 데 편중되어 있다.

국내의 IPv6 의 도입은 6Bone KR 을 중심으로 이루어지고 있는데, 그 구축상황은 ETRI, KT 의 연구기관과 ICU, KAIST, 숭실대, 한양대와 같은 교육기관을 중심으로 연결되어 있으며, 국외 APAN 프로젝트에도 연결할 예정이다. 국내 모든 기관은 H/W 라우터가 아닌 S/W 라우터로 터널링이 되어 있다.

3. 구현

본 논문에서는 아래와 같은 환경에서 IPv6 Web Server 를 구축하고 실험하였다.

OS 로는 FreeBSD 2.2.8 을 사용하였고 IPv6 Stack 으로는 KAME IPv6 Stack 2.2.8, 그리고 GNU C/C++ 을 사용하여 IPv4 Web Server 인 mathopd 1.4.2[5]를 IPv6 에 맞게 바꾸어주었다.

그림 3.1 은 IPv6 에 Porting 한 WWW 서버를 운영할 환경을 나타낸 것이다.

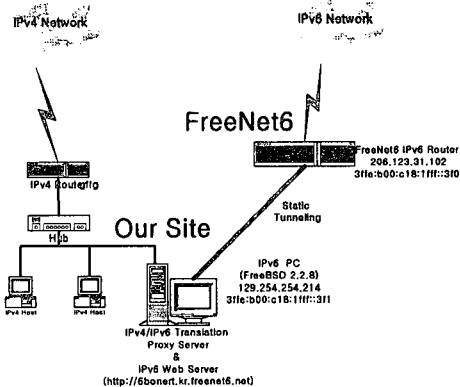


그림 3.1 IPv6 용 Web Server 운영 환경

그림 3.2 는 위의 환경을 시험하기 위한 기본 개념도이다.

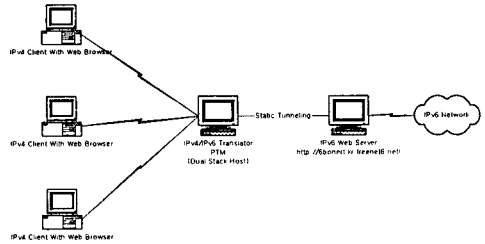


그림 3.2 실험 환경

다음은 IPv4 Web Server 를 IPv6 Web Server 로 변환시키는 과정을 나타낸 것이다.

먼저 첫번째로 Socket Structure 를 바꾸어주어야 하는데 그 방식은 다음과 같다.

```

in6_addr from in_addr
struct in6_addr
{
    u_int8_t s6_addr[16]; /* IPv6 address */
} IPv4 and IPv6 Socket structure 의 차이는 in_addr 와
sockaddr_in 라고 할 수 있다. In_addr structure 는 IPv6
address 때문에 in6_addr structure 로 바꾸어주어야 한다.
sockaddr_in6 from sockaddr_instruct sockaddr_in6
{
    u_char sin6_len; /* length of this structure */
    u_char sin6_family; /* AF_INET6 */
    u_int16m_t sin6_port; /* Transport layer
port # */
    u_int32m_t sin6_flowinfo; /* IPv6 flow information
*/
    struct in6_addr sin6_addr; /* IPv6 address */
};
Sockaddr_In structure 는 in6_addr structure 와 flowinfo field
때문에 sockaddr_in6 structure 로 바꾸어주어야 한다.
    
```

다음 두번째 단계로 address 와 protocol families 를 바꾸어주어야 하는데 그 예는 아래와 같다.

```

s = socket(PF_INET, SOCK_DGRAM, 0);
s = socket(PF_INET6, SOCK_DGRAM, 0); s =
socket(AF_INET, SOCK_STREAM, 0);
s = socket(AF_INET6, SOCK_STREAM, 0);
PF_INET 는 PF_INET6 로 AF_INET 은 AF_INET6 로
바꾸어주어야 한다. Address families 역시
sockaddr_int6 에 들어있으므로 역시 바꾸어주어야 한다.
    
```

세번째 단계로는 관련된 function 들을 바꾸어줘야 하는데 아래와 같은 것들이 있다.

- gethostbyname2 : 네트워크 호스트 엔트리를 이름과 address family 를 통해 지정.
 - getaddrinfo : specified service location 와 연관된 주소정보를 되돌림.
 - getnameinfo : 공급된 IP address 와 port number 에 연관된 text strings 을 되돌림.
 - inet_pton : text form 에 있는 specified address 를 binary equivalent 로 바꾼다.
 - inet_ntop : specified binary address 를 presentation 에 적합한 text equivalent 로 바꾼다.
- Example. struct hostent *h;
h=gethostbyname2("cyberpark.kr.freenet6.net",AF_INET6);

을 해 주었다. 첫째, 한 사용자만 접속했다. 둘째, 문서의 크기는 1k, 2k, 3k, 4k 이다. 셋째, 파일에서 디버깅 메시지에 대한 writing 시간은 무시한다.

위와 같은 가정을 바탕으로 IPv6 Web Server 와 IPv4 Web Server 를 비교하였다.

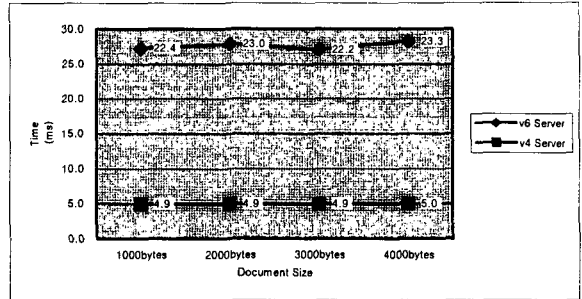


그림 4.2.1 query 당 프로세싱 타임

Data Structure 과 Declared variable 을 바꾸어주면 된다.

```

struct server
{
    int fd;
    int port;
    struct in6_addr addr;
    char *s_name;
    struct virtual *children;
    struct control *controls;
    struct server *next;
};
    
```

위의 4 단계를 따라서 수정을 해준다면 쉽게 IPv4 용 application 을 IPv6 용 Application 으로 바꿀 수 있을 것이다.

4. 구현 결과 및 성능 평가

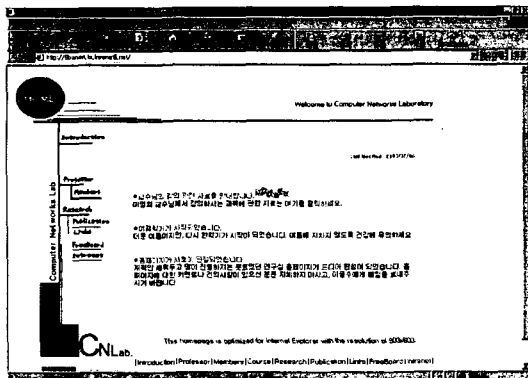


그림 4.1 웹 서비스 화면

위의 그림은 구현된 IPv6 Web Server 를 사용한 웹 서비스의 결과이다.

성능 평가를 위해 fork method 를 사용한 madhopdv6 를 사용하였다. 성능평가를 위하여 다음과 같은 가정

그림 4.2 는 1 개의 웹서비스 요구에 대한 IPv6 Web Server 와 IPv4 Web Server 의 프로세싱 타임을 비교한 것이다. 위의 그림에서 IPv6 Web Server 의 처리 시간은 22ms 정도였고 IPv4 Web Server 와 IPv6 Web Server 사이에 약 17~18ms 정도의 프로세싱 타임의 차이가 나는 것을 알 수가 있었다. 이는 IPv6 의 name lookup time 에 의한 것이다.

5. 결론

우리는 IPv6 환경에서 구동할 수 있도록 IPv4 용 Web Server 를 바꾸어 주었다. 이 IPv6 용 Web Server 는 IPv6 WWW 서비스를 제공해 줄 수 있다.

현재 Static Tunneling 된 6bone 이 native 6bone 으로 변경될 경우 name lookup time 이 충분히 IPv4 환경과 동일해질 것이고 성능 평가 시 드러난 IPv6 Web Server 에서의 lookup processing 에 의한 시간 소요문제도 해결될 것이다.

이 논문은 IPv4 용 Application 을 IPv6 용 Application 으로 바꾸는 과정을 설명하였고 IPv6 용 Application 을 구현함으로써 해서 IPv6 환경에서 웹 서비스를 제공할 수 있다.

이 논문의 내용을 바탕으로 Web Server 뿐만이 아니라 다른 IPv6 용 Application 도 구현을 해주고 각각의 성능 역시 비교평가 해 줄 수 있을 것이고 Chatting 프로그램 역시 구현이 가능할 수 있을 것이다.

참고문헌

[1] Stephen A. Thomas, "IPng and the TCP/IP protocol", John Wiley & Sons Inc, 1996.
 [2] Andrew S. Tanenbaum "Computer Networks", 1996
 [3] Deering, S., and T. Narten, "Internet Protocol, Version 6(IPv6) Specification", RFC 1883, December 1995
 [4] ICU 컴퓨터 네트워크 연구실 "IPv6/IPv4 Translation Router Protocol 에 관한 연구" May 1999
 [5] <http://www.boa.org>.