

가상 사설망을 통한 단일회선에서의 네트워크의 구현

류호, 황도삼

영남대학교 컴퓨터공학과

ryuho@nlp.yu.ac.kr, dshwang@yu.ac.kr

Realizing Networks in Virtual Private Networks on the Single Line

Ho Ryu, DoSam Hwang

Department of Computer Engineering, Yeungnam University

요약

현재, 네트워크는 컴퓨터 환경의 급속한 발달과 함께 빠른 속도로 확장되고 있다. 그러면서 거대한 인터넷의 경계도 끊임없이 확장되고 있다. 인터넷은 네트워크의 네트워크라고 일컬어진다. 즉, 인터넷은 사설 혹은 공용 네트워크로 이루어진 연결된 하나의 거대한 네트워크이다. 하지만 이러한 네트워크를 이용하기 위해서는 전체 인터넷에 연결되기 위한 공통된 약속된 주소를 가져야만 한다. 그러므로 전 세계의 모든 호스트를 연결하는 데는 물리적 한계가 있으며 현재 인터넷 주소체계의 한계가 나타나기 시작했다. 진정한 의미의 인터넷은 네트워크의 무한한 확장에 있다고 생각했을 때 현재의 방식은 인터넷의 진정한 의미에 부합하지 못한다.

본 논문에서는 IP의 공유를 통한 현재의 네트워크의 효율성 제고 및 서버 네트워크에서의 서버 구축이 네트워크의 개선을 이룰 수 있음을 증명한다. 그리고 TCP/IP 기반의 인터넷 통신의 확장, 이에 따른 네트워크 토폴로지를 제시해 단일회선 이하에서 네트워크를 구현하고 이의 이용에 관해 제시한다.

1. 서론

현재 인터넷에서의 주소체계는 IPv4를 쓰고 있다. 하지만 인터넷을 처음 사용할 당시는 오늘날의 인터넷의 증가를 전혀 상상하지 못하였다. 이러한 32비트 주소체계의 고갈문제를 해결하기 위해 1960년대 초 IETF는 IPv4프로토콜을 대신할 IPv6를 만들었다. 현재는 IPv4를 사용하고 있으나 문제는 IPv4 주소체계가 완전히 할당되어 더 이상의 주소가 할당되지 못할 시기가 문제가 된다. 지금 예상되는 IPv4의 예상 고갈 시기는 2008년에서 2018년으로 예상되고 있다[1]. 이미 1996년에 A 클래스의 주소 할당이 끝나버렸으며, B 클래스는 62%, C 클래스는 37% 분량이 끝난 상태이었다(America Registry for Internet Numbers). 이런 상황에서 현재의 네트워크 구조를 개선할 수 있는 무엇인가가 필요하게 된 것이다. 그렇다고 현재(2002년 기준) IPv6가 인터넷의 표준이 되지는 못할 것이다. 왜냐하면 현재의 IPv4가 CIDR(Classless Internetdomain Routing)을 표준화해 잘 사용되고 있기 때문이다. 이렇게 CIDR를 이용하여 서브네트워킹을 사용하면 IP의 불필요한 낭비를 막을 수 있다. 현재는 이렇게 CIDR를 이용한 IP의 효율성 제고를 통해 ISP들이 대처하고 있는 현실이다. 하지만 이것은 근본적인 해결책이 되지 못한다. 앞서 서술했듯이 IP의 고갈을 막을 수 있는 근본적인 대책이 있어야 할 것이다. 그런 점에서 현재의 많은 연구들은 IPv4에서 IPv6로의 구현에 많은 학자들은 관심을 두고 있다[2]. 사실 IPv6도 일종의 네트워크 계층 프로토콜로서 제시된 많은 네트워크 계층 프로토콜의 하나일 뿐이다. 그리고 사용 역시 제한적으로 이루어졌을 뿐이다. 그 원인으로는 기존 프로토콜의 호환성 문제로

인해 네트워크의 계층에 새로운 프로토콜을 도입하고 구조를 바꾸는 것이 쉽지 않기 때문이다.

그런 의미에서 인터넷의 회선공유는 하나의 대안이 될 수 있다. IP의 공유는 현재의 기술력에서 쉽게 이루어질 수 있으며 IP의 공유로 인한 인터넷의 확장은 프로토콜에 의한 확장이 아니라 응용층에서의 확장으로 구현할 수 있다. 그러므로 제약이 적으며 매력적인 동시에 상대적으로 간단하다. 위에서 IP의 공유에 의한 네트워크의 확장의 예시 및 서브네트워크에서의 구현, 그리고 IP공유로 인한 네트워크의 제약에 대한 해결방법의 제시, TCP/IP기반의 인터넷의 응용에 의한 네트워크의 구현, 이에 따른 네트워크의 새로운 토폴로지의 제안에 대해 기술한다.

2. 기존의 인터넷 회선 부족에 대한 대안 및 연구현황

현재 인터넷의 회선 부족에 대한 구체적인 연구 및 대안으로는 IPv4와 IPv6의 통합에 큰 관심을 두고 있다. 이는 NAT-PT(Network address translation-Protocol translation)에 관한 연구에 나타나 있다[2]. 현재의 NAT에 대한 실제적인 구현은 이미 OS내에서도 기본적으로 이루어질만큼 보편화되었다. 그리고 PT 역시 IP의 부족을 해결해 줄 대안으로서 상용화되진 않았으나 이미 그 틀이 잡혀 있다. 그러나 무엇보다도 인터넷의 효율적 이용 방법으로는 CIDR를 이용한 사례가 현재로서는 가장 대표적이다. 이로 인해 IPv4는 그 수명이 연장되었다고 말할 수 있다. 회사나 대학교에서는 일반적으로 클래스를 받아서 호스트의 주소로 사용한다. 이 경우는 주소의 특정한 지정이 아니라 어느 범위부터 어느 범위까지의 주소를 사용

한다는 범위가 지정되는 것이다. 거기에 맞춰 그 네트워크내의 호스트는 고유한 IP를 부여받게 되는 것이다.

우리나라 대학의 경우 B 혹은 C클래스의 주소범위를 받아서 사용하고 있다. 하지만 C 클래스의 경우, 254개의 호스트만을 제공하므로 많은 조직을 위해서는 턱없이 부족하다. 즉 규모가 큰 대학의 경우 C 클래스 하나로로는 감당하기가 어려울 것이다. 반면에 B클래스를 받은 조직은 C 클래스만큼의 서버 네트워크를 둘 수 있으므로 무려 65,634개의 호스트를 부여받는다. 이렇게 비효율적인 인터넷의 주소체계를 CIDR이 네트워크의 주소범위를 지정함으로 효과적으로 나누었다. 하지만 IP의 절대 부족현상이 나타나면 이 역시 쓸모없어질 것이다.

반면 소규모 영업을 하는 회사나 가정에서는 오히려 이러한 문제가 더욱 절실할 것이다. 가정이나 영세한 규모의 회사에서 인터넷에 연결하기 위해서는 ISP에 대해 회선을 하나씩 비용을 지불하고 받아야 한다. 더군다나 이러한 단순한 연결이 아닌 회사를 대외적으로 알리기 위한 서버를 구축하기 위해서는 항상 일정한 IP 하나를 회사에서 가지고 있어야 한다. 이런 경우 하나의 회선으로 다수의 서버가 작동할 수 있다면 경제적인 것이며, 하나의 회선일지라도 그 회선을 공유하는 호스트가 각각 동작을 함으로써 부하가 분산될 수 있다면 또한 이상적이다[3]. 하지만 현재의 각종 서버들은 하나의 주소에 하나의 호스트만이 작동하고 있는 경우가 대부분이며 더 나아가 단순한 클라이언트로서의 인터넷의 연결에 하나의 회선을 할당한다는 것은 회선의 부족을 겪고 있는 현실에서 보면 매우 큰 낭비이다.

3. 단일회선 네트워크의 설계 및 구현

본 연구에서는 하나의 회선에서의 네트워크의 설계에 초점을 맞춘다. 이는 기존의 네트워크에 대한 개선책이 일반적으로 클래스의 효율적 분배나 IPv6로의 변환을 통한 회선의 근본적 확장과는 그 시각을 달리하여 출발하는 것으로 가장 핵심적인 원칙으로는 꼭 필요한 곳에만 회선을 할당하는 데 있다.

3.1 IP 공유와 서브네트워크에서의 서버 구축

IP의 공유는 현재에도 쉽게 나타나고 있는 형태이다. 하나의 회선을 할당받은 호스트가 나머지 IP가 없는 호스트에 패킷을 전송하는 형태가 된다. 이러한 형태의 네트워크의 확장은 ISP와의 경계의 구분에 따라 다시 두가지 종류로 정의될 수 있는데 다음과 같다[4].

1. 라우터를 직접 네트워크의 설계자가 구축하는 경우로 라우터가 서브네트워크의 내부에 들어오게 된다.
2. 라우터를 ISP가 제공하는 네트워크에 속한다. 그러므로 DHCP에 의해 각각의 호스트에 전역주소가 역시 주어진다.

위의 두가지 경우에서 후자는 실질적으로 IP의 공유가 아니라 실제 네트워크의 확장이라 볼 수 있다. 첫 번째 경우가 개인이 구축한 인터넷에 하나의 회선을 넣은 IP의 공유라고 말할 수 있다. 즉 IP의 공유를 네트워크의 관점에서 정의하면, 하나의 호스트에 회선을 할당하는 것이 아니라 하나의 서브 네트워크에 하나의 회선을 할당하는 것이 된다[5]. 각각의 호스트에 회선이 각각 부여된 경우는 이 경우 관심 밖의 문제이다. 오히려 이런 경우 각각의 호스트에 다시 네트워크를 확장하는데 본 연구는 관심을 가진다. 다수의 호스트가 있음에도 불구하고 회선이 하나인 경우 어떻게 범용 통신망에 연결을 할 것이며, 이 경우 하나의 회선을 지니고 있는 네트워크에서 각각의 호스트가 인터넷망에서 어떻게 인식될 수 있는가

하는 것이 문제가 된다[5]. 본 연구에서는 이러한 문제점을 극복하기 위한 방법으로 하나의 회선을 가진 인터넷에서의 각각의 호스트들에 다수의 서버를 구현함으로써 이러한 네트워크의 효율성을 나타내었다.

<표1> 서브네트워크에서의 각 호스트별 네트워크 구축의 예시

서버의 종류	포트번호	작동 프로그램
telnet	23	win2000 telnet server 관리
FTP	21	Serv-U
HTTP	80	apache
입의 설계 서버 프로그램	1000	사용자 작성의 입의의 서버 프로그램

앞의 네트워크 설계는 하나의 전역 IP주소를 가진 165.229.192.171이라는 주소에서 각각의 호스트들을 인터넷 개념으로 연결한 후, 로컬 인터넷의 호스트에서 서버를 구현한 것이다. 이 경우 라우터 기능을 하는 호스트는 들어오는 포트 번호를 보고 사설 네트워크내의 호스트를 인지할 수가 있다. 이처럼 하나의 주소안에서 포트 번호로 세분화함으로써 인터넷에서도 인터넷을 통해 네트워크의 확장을 이룰 수 있다. 이러한 통신을 통해 peer-to-peer 서비스도 역시 가능하다.

3.2 단일회선이하 네트워크에서의 보안 및 유해사이드의 차단

단일회선의 네트워크의 장점으로는 강력한 보안 기능을 들 수 있다. 서브네트워크와 단일회선에서의 인터넷의 형성은 구조적으로는 동일하다[1]. 하지만 서브네팅은 클래스 내부에서의 주소의 분할인 반면 단일회선에서의 인터넷네트워킹은 하나의 회선을 가진 호스트가 인터넷에서 라우터 기능을 하게 된다. 여기에서의 라우터라는 용어도 네트워크에 실질적인 주소가 없는 상황에서 내부의 임의의 IP로 이루어지는 것이기 때문에 엄밀한 의미에서는 실제 라우터와는 기능상 비슷한 역할을 할 뿐 인터넷에서의 라우터는 아니다. 그러므로 인터넷에서의 라우터 기능을 하는 호스트로 인해 그 이하의 계층적인 호스트들은 전체 네트워크와의 연결에서 통제를 받게 된다. 이 때 라우터 기능을 하는 호스트는 전체 인터넷의 패킷이 통과하게 된다. 이 호스트를 통해 전체 네트워크를 조절할 수가 있게 된다.

유닉스 기반의 라우터 기능의 네트워크에서의 보안 및 통제의 작동변이는 다음과 같다[6][7][8][9][11].

- ① 자식 프로세스의 생성
- ② checksum algorithm을 통한 인터넷 패킷의 복사
- ③ ICMP패킷을 통한 제어
- ④ 자식 프로세스의 소멸
- ⑤ 제어 신호가 나올 때까지 프로세스를 생성하지 않음

이와 같은 보안 및 제어는 단일회선에서는 단 한 대의 호스트만 제어하면 되므로 대단히 용이하다. 또한 구조적으로 외부에서 인터넷으로의 접근이 불가능하므로 보안에 뛰어나다고 할 수 있다[12]. 다만 문제가 될 수 있는 것은 FTP와 같은 응용에서는 한번의 프로세스의 실행이 지속적인 경우 프로세스가 계속 늘어남으로 전역주소를 지닌 호스트에 큰 부담이 될 수 있다. 소규모 네트워크의 실험에서는 이러한 혼잡은 라우터에서는 크게 문제가 되

지 않았으나 중단 시스템에서 심각한 문제를 일으켰다. 특히 FTP서버의 경우 중단 시스템간의 통신 지속시간이 지속적으로 작동이 잘 이루어지지 않았다. 반면 HTTP 서버의 경우 한 번의 연결 후 바로 통신을 끊어버리기 때문에 지연에 따른 문제는 나타나지 않았다. 하지만 이 역시 네트워크가 커지면 상당한 지연이 나타날 것이다. 이에 대한 해결책으로는 프로세스의 재사용과 프로세스 대신 메모리가 공유되는 스레드를 사용하는 것이 해결방안으로 제시할 수 있다[7]. 스레드를 통한 프로그램의 실행은 각각이 독립적이다.

또한 위의 기술은 유닉스에만 볼 수 있는 알고리즘이다. 자식 프로세스를 생성함으로써 서버는 코드가 간단해지고 자식 프로세스는 매우 복잡해진다. 하지만 윈도우에서는 다르다. 유닉스에서는 일반적으로 프리포크 서버(Pre-fork server)를 사용하는데 이러한 말은 윈도우에서는 존재하지도 않는다. 그러므로 윈도우에서는 자식 프로세스를 생성한다는 것은 생각할 수도 없다.

윈도우에서는 스레드를 사용하며 각각의 스레드가 단일 프로세스 밑에서 수행되는 구조를 가지고 있어야 한다. 이 경우 위의 경우와는 달리 스레드에 일어 생겨 동작이 멈추는 것을 막기 위해 여분의 프로세스를 생성시켜야 한다[13].

3.3 TCP/IP 기반의 인트라넷 통신의 확장 및 네트워크의 구성도

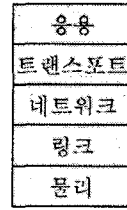
이러한 인트라넷의 구축은 인트라넷 안에서는 라우터만이 종적인 구조를 가질 뿐 나머지는 모두가 대등하다. 횡적 구조는 트래픽에서 상당한 이점을 보인다. 다만 다수의 인트라넷에서의 트래픽의 집중이 생길 때 정체가 생길 수 있으므로 이러한 경우 라우터를 두 군데로 분산하여 인트라넷과 인터넷을 연결할 수 있다[1][3][4][11]. 인터넷 및 인트라넷에 의한 호스트간의 1:1의 통신은 인터넷 프로토콜 스택의 5가지 중 제 3계층인 네트워크층을 이용하므로 호환성이 매우 뛰어나다. 네트워크 계층을 이용하므로 기존의 인트라넷의 구축시 나타나던 운영체제간의 호환성문제를 극복할 수가 있다. 이 경우 NIC의 효율은 다음과 같다.

<표2> 각 네트워크 연결환경에 따른 데이터의 효율 (전송속도 단위는 bps)

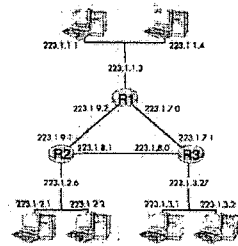
링크대역폭 \ 호스트기종	10Mbps	100Mbps
windows-solaris	1.88M(18.8%)	1.88M(1.88%)
windows-windows	1.88M(18.8%)	1.88M(1.88%)
기존의 windows네트워크	7M(70%)	61M(61%)

위의 표를 보면 링크의 대역폭이나 운영체제간의 차이에 관계없이 TCP/IP에 의한 각 호스트간의 통신속도는 1.88Mbps로 모두가 동일하다. 이것은 마치 인터넷에서 두 호스트간의 통신원리를 확장된 인트라넷에 도입하였기 때문에 네 가지 경우 모두 동일한 전송속도를 나타내는 것이다. 즉 TCP/IP기반의 통신에서는 각 호스트에서 동작하는 어플리케이션의 성능(대표적인 원인으로 버퍼가 있다.)이 통신에 가장 큰 영향을 끼친다. 위의 측정치 역시 어플리케이션에 따라 가변적일 수 있다.

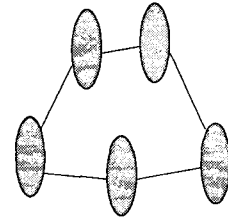
반면 기존의 윈도우 로컬 인트라넷의 경우는 두 호스트간의 링크 대역폭의 영향을 크게 받게 된다. 위의 표에 나타나듯 대역폭이 커짐에 따라 데이터의 전송률도 비례적으로 증가한다. 측정시 링크 효율은 일반적으로 대역폭의 70%에 육박할 정도로 뛰어난 성능을 보인다.



(그림1) 인터넷 프로토콜 스택



(그림2) 네트워크간의 횡적 구조의 예시



(그림3) 각 네트워크간의 횡적 연결

그렇다면 인트라넷에 의한 인터넷의 확장에 가장 좋은 네트워크의 구성도는 무엇일까? 그에 대한 해답으로 위의 (그림3)과 같은 링형의 인트라넷의 연결을 제시한다[13]. 각 네트워크는 하나의 회선만 들어오는 단일회선의 인트라넷이다. 여기에서는 다섯 개의 그룹지어진 인트라넷이 링형으로 연결되어 있으나 사실은 이러한 인트라넷의 결합이 인터넷으로 이어지는 사실상 봤을 때 거시적인 관점에서 링형으로 나타내었을 뿐 큰 의미는 부여하지 않는다. 다만 앞의 구성도에서 주목할 점은 연결이 호스트간의 연결이 아닌 네트워크간의 연결이라는 점에 주목할 수 있다. 이와 같은 네트워크의 구성도는 인터넷과 인트라넷의 통합을 나타내며 현재의 프로토콜 환경에서도 인터넷의 무한한 확장이 가능해짐을 나타낸다.

인트라넷으로 연결한 인터넷의 효율은

$$\text{efficiency} = \frac{1}{1 + 5t_{prop}/t_{trans}} \cdot n \quad [1]$$

으로 표현할 수 있다. (t_{prop} : 전파지연, t_{trans} : 프레임의 전송시간, n : 인트라넷 내부의 호스트의 대수)

4. 최적화 및 효율적인 네트워크의 구축과 평가

사실망은 C클래스로 설정을 하였다면 이론상 254대의 호스트까지 확장할 수 있다. 하지만 단일회선이하에서 254대의 호스트를 물리적으로 확장을 하게 된다면 트래픽이 심해 네트워크 외부로 패킷이 나가면 정체가 심하다. 그러므로 네트워크의 확장에는 한계를 정확히 파악해 경제적이면서도 좋은

성능을 나타낼 수 있도록 최적화할 필요가 있다. 특히 지연에 민감한 멀티미디어 응용, 비디오/오디오 스트리밍이 갈수록 많이 나타나는 추세에서 네트워크는 많은 대역폭을 필요로 한다. 그러므로 네트워크의 최적화는 멀티미디어 응용기술의 성능여부가 관건이다. 비디오/오디오 스트리밍은 공급자의 입장에서 300K/56K의 속도로 제공하는 것이 일반적이다. 고품질의 비디오/오디오 서비스는 300K로서 서비스하는 것이 일반적이며 단순 오디오 서비스는 56Kbps의 서비스로도 좋은 품질의 서비스가 가능하다(현재 통용의 AOD서비스 수준). 이러한 서비스는 공급자의 전송률에 완전히 의존하며 전송률의 차이 역시 매우 크다. 실시간 비디오/오디오 스트리밍의 전송률을 라우팅 호스트 및 종단 호스트에서 측정할 시 각 스트리밍의 전송률은 30K에서 1048Kbps까지 매우 다양하게 측정되었다(하지만 300Kbps가 일반적이며 이 경우 인터넷 패킷의 실측 시 더 많은 패킷이 라우터를 통과한다). 이러한 응용들은 단일회선에서 네트워크 설계에 가장 큰 요인이 되어야 한다. 앞서 언급되었듯이 이들은 데이터의 손실에는 둔감하나 지연에는 대단히 민감하므로 넓은 대역폭을 필요로 한다. 그러므로 사설망을 이용한 네트워크의 구현은 이러한 응용의 제약이 나타나지 않는 범위내에서 이루어져야 한다. 네트워크의 설계에는 이것에 의해 최적화 조건이 결정되게 되며 확장한 한계를 여기서 기술할 수 있다. 밑의 조건은 실시간 스트리밍의 서비스를 바탕으로 적절히 호스트의 확장을 결정하는데 사용한다.

- ① 단일 호스트에서 사설망의 최적화 대수
 $대역폭/n \geq S + p \times S$
 (S- 스트리밍서비스 제공속도, n- 호스트의 대수, p- 전체 패킷중 제어 패킷의 비율)
 - ② 전체 망의 관점에서 최적화 대수
 $대역폭 \geq \sum 데이터패킷 + \sum 제어패킷$
 혹은
 $대역폭 \geq \sum S + p \times \sum S$
- 으로 표시할 수 있다.

마지막으로 실험에 대한 측정치를 보이겠다.

- ① 실시간 오디오/비디오 스트림의 제어패킷 및 데이터 패킷의 분리

<표3> 오디오/비디오 스트림의 랜덤 표본조사 결과

데이터 패킷 전송률	비율(19)
256K	5%(1)
275K	5%(1)
307K	42%(8)
308K	16%(3)
508K	5%(1)
512K	5%(1)
694K	16%(3)
1048K	5%(1)

실시간 오디오/비디오 스트림을 재생하지 않고 저장한 후 디스크에서 파일형태로 변환하여 데이터 패킷과 제어 패킷을 분리한다. 분석시 인터넷 패킷과 데이터 패킷의 비율은 스트리밍마다 차이를 보이나 라우팅하는 호스트의 전송률이 항상 종단 호스트의 스트리밍 재생 속도보다는 높다. 실제로 307Kbps의 송신서버의 오디오/비디오 스트림을 받으면 종단 호스트는 355Kbps의 데이터 전송률을 나타내었다.

- ② 프로토콜에 따른 전송속도의 차이

TCP기반의 HTTP: 한계범위(실제시 NIC링크 환경에서 약 2%, DSL장비 다운로드시 12.5% 효율)내에서 호스트의 대수로 나누어진다. 즉 대역폭 2M인 DSL 모뎀에서 250K, 10M인 NIC에서 200K, 100M에서 2Mbps 정도의 효율을 나타낸다. 그리고 이러한 한계속도내에서 서버네트워크에 패킷을 배분한다.

UDP기반의 mms: 네트워크 링크 대역폭까지 네트워크 효율의 확장이 가능하다.

$$effency \leq width \text{ of link}$$

이를 다시 정리하면

$$(a+b+c)+p(a+b+c+...) \leq width \text{ of link}$$

로 정리된다(a,b,c는 임의의 multimedia stream, p : 전체 데이터그램 중 제어패킷).

5. 결론 및 향후 발전 과제

인트라넷과 인터넷의 융합은 기존에도 관심을 많이 가져온 분야이다. 97년 전반에 PC기반의 C/S 기술과 인터넷기술을 융합하여 인트라넷을 구축하는 도구의 출시가 있었으며, 이후 인트라넷은 인터넷과 클라이언트/서버방식의 혼합형이 되어가고 있다[3]. 그러면서 인트라넷이 거대해지고 인트라넷안에서의 다양한 응용 및 인트라넷안에서의 클라이언트/서버에 대한 연구도 다양해졌다. 이러한 것은 기업의 환경이 바뀌면서 산업구조적인 영향으로 인해 기업의 전산화에 가장 핵심적인 요소로 이 기술이 자리잡았기 때문이다. 그러므로 이 분야에서 앞으로 보안에 관련된 분야가 더욱 발전되어야 할 것이며 네트워크가 비대해집에 따른 효율성의 제고에 관심이 더욱 필요할 것이다.

6. 참고 문헌

- [1] James F Kurose, Keith W Ross / 강현욱, 신용태, 안상현, 최중현 역, "컴퓨터 네트워킹", 홍릉과학출판사, 2001.
- [2] ETRI 수행과제-IPv4/IPv6 차세대 인터넷 주소변환, 1999.
- [3] 서보현, 심규성, 이종하, 이경배, "클라이언트/서버와 인트라넷(1)", 시스템통합연구소, 1998.
- [4] David Garrett / 이만용 역 "인트라넷 언리쉬드", 정보문화사, 1997.
- [5] Merlin Hughes, Michel Shoffner, Derek Hamner, Umesh Bellur / 고려대학교 컴퓨터학과 시스템연구회 역, "Java network programming", 인포북, 1999.
- [6] Richard Stevens, "Unix network programming volume1 second edition", Prectice Hall, 1998.
- [7] 엘리엇 헤럴드 / 김한기, 유재민 역 "Java network programming", 한빛 미디어, 2001.
- [8] Avi Silberschatz, Peter Galvin, Greg Gagne, "Applied operating system concept", wiley 1999.
- [9] Barry holmes / 홍기형, 박현석 "자바 프로그래밍", 사이텍 미디어, 1998.
- [10] 김화중, "컴퓨터 네트워크 프로그래밍".
- [11] Behrouz a Forouzan, "Data Communication and Networking", McGRAW-HILL, 2000.
- [12] D Brent Chapman, Elizabeth D Zwicky / 채규혁 역, "인터넷 방화벽 구축하기", 한빛미디어, 1998.
- [13] Mohammed J. Kabir / 최송국 역, "아파치 웹 서버 구축", (주)사이버 출판사, 2001.