論文 95-12-12-4

투명한 포트-주소 변환기를 통한 IP 주소의 재사용

正會員 金 逸 煥*, 廉 憲 榮*

IP Address Reuse Through Transparent Port-Address Translator

Il Hwan Kim*, Heon Young Yeom** Regular Members

要 約

인터넷 표준 규약으로 채택된 IP의 주소공간은 폭발적으로 증가하는 수요와 클래스 분할에 따른 낭비에 의해 포화상태에 이르고 있다. 지금까지 제시되어온 단기적 해결책들에 대해 살펴 보면 지역 주소-전역 주소의 자동변환에 의한 IP 주소의 중복 재사용이 새로운 프로토콜의 채택 시점 전후까지의 적절한 대안이 될 수 있음을 알 수 있다.

본 논문에서는 지역 주소-전역 주소간의 일대일 변환에서 나아가, 다수의 지역 노드가 하나의 전역 주소를 공유하여 사용하게 해 주는 포트-주소 변환기를 제시하고, 그 설계 및 구현상의 문제점들과 해결 방법을 설명한다. 또, 주소의 자동 변환에 의한 IP주소의 재사용에 있어 야기될 수 있는 문제점과 효용성에 대해 고찰한다.

ABSTRACT

The address space of IP, a standard Internet Protocol, is being exhausted by explosively increasing number of demands and the inefficient address allocation scheme based on the network class partitions. Among the short term solutions that have been suggested so far, IP address reuse through automatic translation between local and global addresses is considered as an appropriate solution prior to a new protocol is adapted.

In this paper, we suggest a port-address translator which improves the one-to-one translation of local-global address translator by enabling several local nodes to share a globally unique address, and discuss various problems and their solutions we have encountered in designing and implementing the translator. Also, the problems and the effectiveness of address reuse by automaticaddress-address or port-address translator is investigated.

* 서울대학교 자연과학대학 계산통계학과
論文番號 : 95139-0407
接受日字 : 1995年 4月 7日

Ⅰ. 개 요

인터넷에서의 가장 큰 문제점 중 하나는 IP 주소의 부족이다. 4 바이트 길이의 IP 주소는 클래스 단위의 도메인간 라우팅(Inter-Domain routing)의 사용과 매우 빠른 인터넷의 확장에 따라 설계시의 예상 이상으로 빨리 소모되고 있다[TE93]. 이미 국민학교까지 IP 주소를 할당받은 미국에 비해 우리나라의 IP 주소 할당량은 매우 부족한 현실이며, 부족한 IP 공간을 효율적으로 사용하는 방법이 절실히 요구되고 있다.

가장 근본적이고 장기적인 해결책으로써 IP를 대체할 새로운 통신규약의 제정이 진행되고 있으나 상용 인터넷의 보급, 교육망 확장 사업등에 따른 주소공간 소비속도의 증가는 IP 주소의 부족을 당장 해결되어야 할 절실한 문제로 만들고 있다. 이에 장기적인 해결책의 연구와 병행하여 구현이 쉽고 적은 비용으로 기존 인터넷에 채택될 수 있는 몇가지의 단기적 해결책이 제시되고 있다.

앞으로 제 2 절에서 IP 주소 부족의 구조적인 문제점과 기존의 주소 재사용을 통한 해결책들을 소개하고, 제 3 절에서는 포트-주소 변환을 통한 주소 공간의 재사용에 대해 논한다. 여기서 제시된 포트-주소 변환기는 1차 시제품 수준으로 구현이 완료되었으며, 제 4 절에서 설계 및 구현상의 문제점과 그 해결책에 대해 설명한다. 마지막으로 제 5 절에서 여기서 제시된 방법의 타당성과 특징을 요약 서술한다.

Ⅱ. IP 주소할당의 문제점과 단기적 해결책들

1. IP 주소 공간의 낭비

IP 주소는 4바이트로 이루어지므로 이론상 2^{32}개의 서로 다른 노드가 접속될 수 있다. 그러나, 이 주소공간은 다음과 같은 두가지 이유때문에 낭비되는 경향이 있다.

1. IP 주소공간은 클래스 단위로 분할되며, 하나의 지역망은 하나 이상의 A/B/C 클래스 망 주소를 가진다. 서로 다른 지역망이 같은 망 주소를 공유하는 경우는 없다.
2. 한 지역망 내의 노드 중 외부망과 동시에 교신하는 수는 그 지역망의 크기에 비해 상대적으로 적다.

예를 들어, 한 C 클래스 지역망 내의 255개 주소 중 150개의 주소가 사용되며 그 중 50개의 주소만이 외부망 회선을 통해 교신하는 경우, 인터넷 광역망 입장에서 보면 이 지역망이 꼭 가져야 하는 IP 주소는 50개에 불과한 것이다. 외부와 교신하는 일이 없는 100개의 주소는 단지 그 지역망 내에서만 유일하게 식별이 가능하면 된다. 이때 255개 중 150개를 사용하고 남은 100여개의 주소는 다른 지역망에서 사용될 수 없기 때문에 불필요하게 낭비되는 셈이 된다.

이처럼 클래스단위 망 주소 분할과 불필요한 전역주소 할당은 IP 주소 공간 낭비의 2대 요인으로 간주될 수 있다.

2. 주소 재사용에 대한 고찰

기업망 등 사설 TCP/IP 지역망을 구축함에 있어서 이 지역망에 대해 전역적으로 유일하지 않은 중복된 주소를 할당하는 방법이 제시된 바 있다.

[RM94]에서는 앞 단락에서 살펴본 바와 같이 단위 지역망 내에서 외부와 접속이 필요한, 전역적으로 유일하게 할당되어야 하는 노드 수가 상대적으로 적다는 점을 지적하고 있다. 이 문서에서는 사설 TCP/IP 망 구축시에 사용될 수 있는 중복사용을 위한 IP 주소가 인터넷 관리 기구에 의해 다음과 같이 예약되어 있음을 명시하고 있다.

A Class 10.0.0.0 - 10.255.255.255, 1개
B Class 172.16.0.0 - 172.31.255.255, 16개
C Class 192.168.0.0 - 192.168.255.255, 255개

또, [FLYV93]에서는 비 클래스 도메인간 라우팅(CIDR: Classless Inter-Domain Routing)을 실현함으로써 클래스단위 망 분할에 의한 주소공간의 낭비를 막을 수 있음을 제시하였다.

3. Proxy 서버

보안이 엄격히 요구되는 사설망의 운용시 방화벽(firewall)을 구축하여 외부로부터의 접속을 근본적으로 차단하는 경우가 있다. 대개는 Packet Filtering Gateway를 통해 내부 망으로의 통신을 차단하는데, 이는 패킷의 선택적인 여과를 통해 이루어진다. 이때 사설망 내부의 사용자들은 여러 외부 서버로부터 인터넷 서비스를 제공받고자 하므로, 내부로부터 외부로의 접속

방법을 제공하는 것은 외부로부터의 통신을 차단하는 것 다음으로 중요한 사안이 된다.

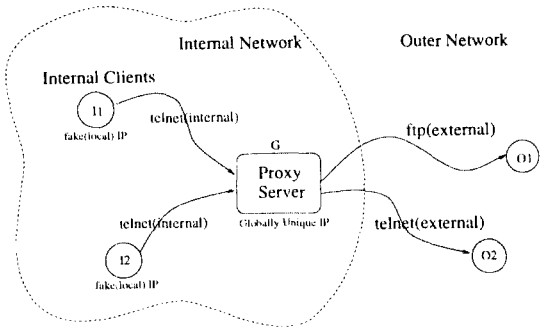


그림 1. Proxy 서버를 통한 외부 접속

가장 단순한 방법으로는 Proxy 서버를 통하여 외부로의 접속을 제공하는 것이라 할 수 있다. 이는 보안이 충분히 유지될 수 있는 Proxy 서버에 전역 주소를 할당해 주어 외부망 접속이 필요한 경우 일단 Proxy 서버를 거치도록 하는 방법이다(그림 1 참조).

이 경우 내부망 사용자는 자신이 사용하는 지역망의 환경 설정을 알아야 하며 Proxy 서버는 내부망 사용자를 위한 계정을 준비해야 한다. 또, Proxy 서버는 내부망에서 외부로 가는 다리 역할을 하므로, 자원과 사용자의 집중화를 야기하게 된다.

4. 소켓 서버

대개의 UNIX 운영체계에서는 응용 프로그램의 네트웍 접근이 소켓을 통해 이루어진다. 이 소켓은 보통 응용 프로그램이 수행되는 호스트의 커널이 관리하나, 전역망과의 통신에 사용되는 소켓을 별도의 서버가 관리/할당함으로써 사용자에게 투명한 외부망 서비스를 제공하는 방법이 연구된 바 있다.

[KK92]에서는, 한 지역망 내에서 외부와 직접 접속이 가능한 노드를 몇 개로 제한한 경우 원격 소켓의 관리를 통해 외부망으로의 중계를 제공하는 소켓 서버를 소개하고 있다. 외부와 통신할 수 있는 소켓은 모두 외부 주소를 가지는 소켓 서버로부터 Rconnect()를 통해 할당받아야 하며 직접 접속이 불가능한 내부 노드들은 이렇게 할당받은 외부 소켓을 통해서만 외부와 접속하게 한다. 소켓 서버는 각 소켓 Client에게 자신의 IP 포트 번호를 할당해 주며 이를 통해 외부와 내부 노드 간의 교신을 중계해 준다.

이 소켓 서버를 이용하면 UDP/TCP를 막론하고 지역망과 전역망사이의 패킷중계가 가능하다. 소켓 서버 방식은 사용자에게는 투명한 환경을 제공하나 Client 프로그램의 수정이 불가피하여, UNIX운영체계가 아닌 경우 적용이 어렵다는 단점이 있다.

5. 망 주소 변환기(Network Address Translator)

앞서 살펴 본 바와 같이 중복된 주소를 사용하는 지역망에서는, 특수한 방법을 쓰지 않는 한 전역망으로의 접속이 불가능하다. 전역망으로의 접속을 허용하기 위해 제시된 위의 방법들은-투명한 방식이든 아니든 간에-중복 재사용된 지역 주소를 전역적으로 유일한 주소(Globally unique)로 변환하는 절차를 필연적으로 수반한다.[1]

Bellcore의 P. Francis는 DNS(Domain Name System)와 결합된 형태의 투명한 양방향 망 주소 변환기를 통해 IP 주소의 재사용과 비 클래스단위 라우팅을 실현하는 방법에 대해 논한 바 있다[TE93][EF94]. 중복된 주소를 가지는 지역망들이 주소 변환기에 의해 전역망과 연결되는 개념적인 망 구성은 그림 2에서 보는 바와 같다.

중복된 주소를 사용하는 지역망과 전역망의 경계에는 몇 개의 전역 주소를 예약해서 가지고 있는 주소변환기(NAT: Network Address Translator)가 있다. 이 주소변환기는 자신이 관리하는 지역망 내의 노드가 외부와 교신을 위해 전역적으로 유일한 주소를 필요로 할 때마다 전역 주소를 동적으로 할당해 준다. 한번 할당된 주소는 그 노드의 외부 접속이 모두 종료될 때까지 사용되며 접속이 유지되는 동안 주소변환기는 자동적으로 지역 주소와 전역 주소간의 변환을 행하게 된다. 이 변환은 외부망과의 경계를 통과하는 모든 패킷의 TCP/IP 헤더 정보를 검색한 후 전역 주소와 지역 주소간의 매핑 테이블을 참조하여 헤더를 수정함으로써 이루어진다.

1) 사실상 도메인간 라우팅은 주소 변환에 의해 달성된다. 라우팅에 따른 Ethernet interface 주소의 변환을 생각해 보라.

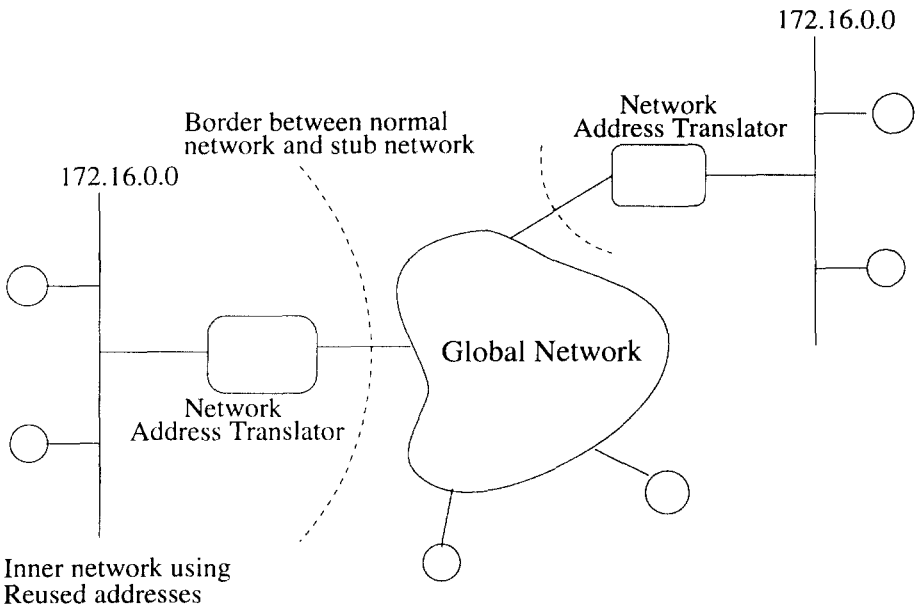


그림 2. 주소가 중복된 네트웍 구성도

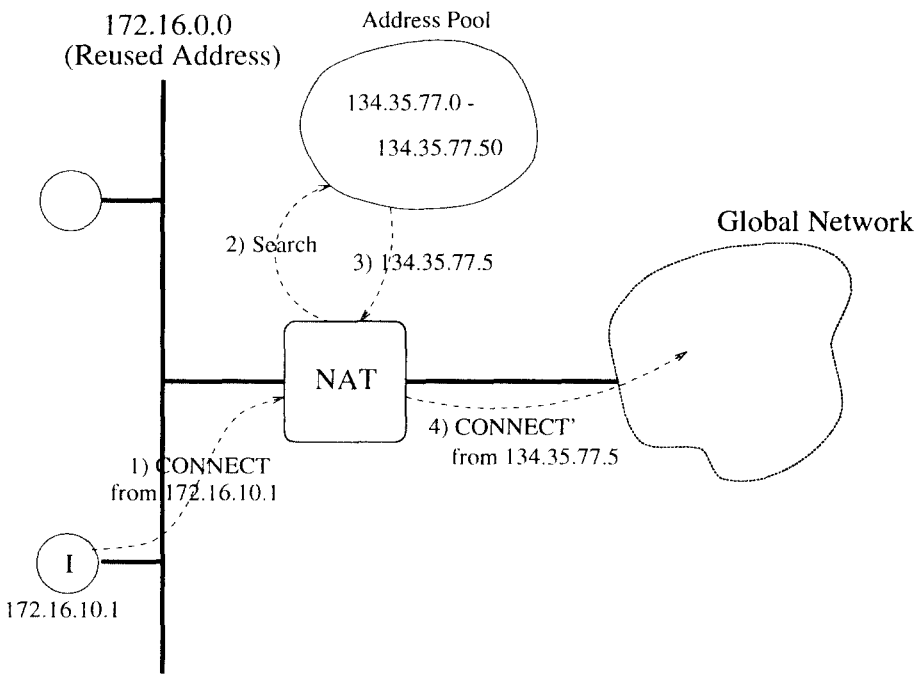


그림 3. 투명한 주소 변환(내부에서 외부로의 접속)

그림 3에서 보는 바와 같이 주소변환기는 DNS 및 도메인간 라우터/게이트웨이와 밀접한 연관을 가진다. 이 그림을 통해 주소변환기의 동작 과정을 살펴 보자.

외부로의 접속을 원하는 노드 I가 패킷을 외부로 내보내고자 하면 주소변환기가 이 패킷을 가로채 외부 주소를 할당해 준다. 모든 패킷은 주소변환기를 거치며, 외부와 내부의 경계를 패킷이 통과할 때마다 주소변환기는 노드 I의 지역주소와 앞서 할당된 전역주소간의 변환을 행하고 이 수정된 패킷을 중계한다. 또, 외부에서 이 지역망 내 노드 I로의의 접속요청이 있는 경우, 먼저 DNS가 상황을 주소변환기에 알려 주소변환기가 I를 위해 전역주소를 준비하도록 한다. 따라서 투명한 양방향 주소변환이 이루어지게 된다. 이때, 동시에 외부와 접속이 가능한 지역망 노드의 최대수는 주소변환기에 미리 예비된 전역주소의 갯수이다.

이 방법은 다음과 같은 단점을 가지고 있다.

- DNS를 사용하지 않는 응용은 적용되지 않는다.
- UDP등의 connectionless 통신에는 적용되기 어렵다.
- 한 지역망 내에서 많은 노드가 외부와 접속될 경우 전역주소가 부족해 질 수 있다.
- 패킷 내에 IP 주소를 포함하는 응용에 대한 별도의 고려가 필요하다.(FTP, ICMP등)

이 방법은 지역망 내부 노드에 대한 별다른 수정 없이 망간 경계에 주소변환기를 설치함으로써 투명한 주소 변환을 가능하게 한다. 또, DNS와의 결합을 통해 내부로의 요청 (inbound request) 문제도 해결될 수 있다.

이 논문은 후에 인터넷 표준화 문서인 RFC 형식으로 다시 발표된다[EF94].

Ⅲ. 투명한 포트-주소 변환기

이 글에서는 새로운 방식의 망 주소 변환기인 포트-주소 변환기를 이용한 주소 재사용에 대해 설명한다. 한 노드에서 동시에 필요로 하는 소켓 수에 비해 실제 UDP, TCP포트 수가 월등히 많은 것에 착안하면 여러 개의 지역 소켓(local socket)을 하나의 전역 주소와 미사용 포트 번호로 변환함으로써 한 전역 주소로 복수의 지역 노드에 외부망 접속을 제공할 수있다.

앞으로 각 노드에서의 소켓은

(*IP address, TCP port number*)

로 표기하기로 하자. 또, 모든 TCP 패킷을

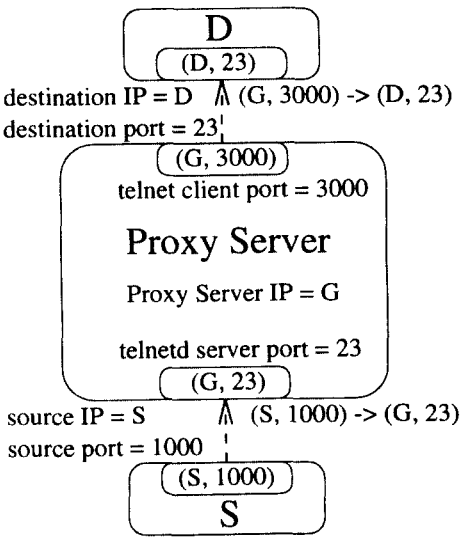
(*srcIP, srcPORT, dstIP, dstPORT*)

로 나타내기로 하자.

패킷이 지나가는 모든 경로 내에 같은 IP 주소를 가지는 노드가 없다고 가정하면, 각 노드내에 존재하는 통신 주체는 (*srcIP, srcPORT*)의 소켓에 의해 단위 지역망내에서—IP 주소의 중복이 전혀 없다면 전체 인터넷 내에서—유일하게 식별될 수 있다. 따라서 ((*srcIP, srcPORT*), (*dstIP, dstPORT*))의 소켓 쌍은 송신자와 수신자를 유일하게 식별할 수 있게 해 준다[TCP81].

1. proxy 서버 사용시의 일대일 패킷 전송

우선, Proxy 서버를 이용한 TCP connection의 peer entity 간 패킷 전달과정에 꼭 필요한 최소한의 정보가 무엇인지 살펴 보자.



지역주소를 가지는 S에서 telnet G를 사용해 G로 원격 로그인 한 후 다시 G에서 telnet D 하여 S와 D간의 간접적인 telnet connection이 생겼다. S는 전역적으로 유일한 주소를 가지지 않고도 외부 전역망과 접속이 된 상태이고, 외부의 정상적인 노드 D와 주고받는 패킷은 proxy 서버 G를 통해 중계되고 있다.

그림 4. proxy 서버를 통한 telnet session

그림 4를 살펴 보면, (*S*, 1000)에서 (*D*, 23)로 가는 패킷은 G가 중계하여 (*G*, 3000)로 바뀌어 (D, 23)로 보내고, (*D*, 23)에서 (*G*, 3000)으로 보내어진 패킷은 G가 (*S*, 1000)로 중계함으로써 S와 D 사이의 일대일 접속이 가능하게 됨을 알 수 있다.

이 과정을 자동화하여 투명한 변환과 중계가 이루어지게 하는 것이 포트-주소 변환기의 역할이다.

2. 포트-주소 자동변환에 의한 중계

앞 절에서 살펴본 바와 같이, D가 받는 모든 패킷은 G가 (*S*, 1000, G, 23)라는 패킷을 (*G*, 3000, D, 23)로 바꾸어 내보낸 것이며, S가 받는 모든 패킷은 G에 의해 (*D*, 23, G, 3000)에서 (*G*, 23, S, 1000)로 바뀌어 진 것이다. 이 변환은 사용자가 명시적으로 Proxy 서버를 이용한 결과이며 투명성을 제공하지 못함은 앞서 언급한 바와 같다. 여기서 사용자에게 완벽한 투명성을 보장해 주면서 기존 시스템의 최소한의 수정으로 같은 효과를 내기 위해서는 G가 위의 변환을 자동으로 해 주는 방법을 취해야 한다는 것을 알 수 있다.

전역적으로 유일하지 않은 주소를 사용하는 지역망의 노드가 외부로 패킷을 내보내기를 원하는 경우, 송신자 주소를 주소 변환기 G를 통해 전역적으로 유일한 주소로 바꾸어 주는 방법은 이미 [TE93]에서 논의된 바 있다. 이때 지역망과 전역망의 경계에서 발생하는 전역 주소와 지역 주소간의 변환은 각 단위 네트웍 내에서 (dstIP, dstPORT)쌍이 유일함을 보장하여 전역망내에서 패킷의 수취인을 유일하게 식별할 수 있게 한다.

TCP패킷의 포트 번호는 16비트로 이루어져 있으므로, 한 IP 노드에 2^{16}개의 서로 다른 소켓이 존재할 수 있다. 이는 하나의 IP 주소에 서로 다른 수취인이 2^{16}만큼 존재할 수 있는 것을 의미한다. 각종 응용에서 사용되는 대부분의 well-known 포트는 10000번 이하로 예약되어 있고, 한 노드에서 사용되는 포트의 갯수는 2^{16}보다도 훨씬 적은 경우가 많다. 따라서, 미사용인 포트에 다른 노드의 주소를 대응시킴으로써 전역 주소를 가지는 노드가 지역 주소를 가지는 노드의, 외부 전역망으로의 패킷을 대신 취급해 주는 방법을 생각해 볼 수 있다.

표 1. 포트-주소 변환 테이블의 예

G.PORT	(I.IPaddress, I.PORT)	STATUS
30000	(172.16.10.1 , 1234)	SYN-SENT
30001	(172.16.10.1 , 3487)	ESTABLISHED
30002	(172.16.30.27, 12039)	CLOSE-WAIT
30003	(NULL , NULL)	CLOSED
30004	(NULL , NULL)	CLOSED
⋮	(⋮ , ⋮)	⋮
40000	(172.16.68.99, 3465)	ESTABLISHED

표 1을 보자. 이 표는 172.16.0.0 주소를 가지는 stub B class network 내의 노드 I(Inner nodes)에서 생성된 소켓 (IPaddr, PORT)들이 전역 주소를 가지는 G(Gateway node)의 포트 번호로 대응되는 것을 나타낸다. 이 대응은

1. [TE93]에서와 같이 DNS를 통해
2. TCP 헤더 정보 중 SYN 플래그 검출을 통해

동적으로 할당(allocate)되며, STATUS가 CLOSED로 되면 해당 포트의 할당을취소한다(deallocate).

G는 주소의 변환이 필요한 패킷을 발견할 때마다 이 표를 참조하여 패킷의 헤더정보를 수정한 후 중계한다. 이 중계는 내부로 향하는 패킷(inbound packet)과 외부로 향하는 패킷(outbound packet)을 감시하여 이루어진다.

외부로 향하는 패킷의 송신자 헤더는 포트-주소 변환 테이블에 따라 (*I.Addr, I.PORT*)에서 (*G.Addr, G.PORT*)로 수정된 후 외부 전역망으로 중계된다. 또, 외부에서 G로 수신된 패킷은 (*G.Addr, G.PORT*)에서 (*I.Addr, I.PORT*)로 수신자 헤더가 수정되어 내부 지역망으로 전달된다.

외부로의 TCP 패킷은 중복 재사용되는 IP를 가지고 있으므로, 수정 없이는 지역망 밖으로 노출되지 않게 해야 한다. 여기서는 외부로의 요청에 따른 동적 할당만을 언급하였으나, 후술하게 될 정적 할당 방법에 의해 내부로의 접속 요청도 중계가 가능하다.

3. 주소 재사용의 효용성

포트-주소 변환기는 IP 주소공간을 효율적으로 사용할 수 있게 한다.

예를 들어 전역적으로 사용가능한 B Class 망 주소

를 가지고 C Class 지역망들로 재구성하는 경우, 한 지역망 내에서 동시에 요구되는 전역 소켓의 갯수가 10,000개라고 가정해보자. (이 숫자는 전혀 근거가 없는 것은 아니다. C Class내에는 최대 255개의 노드가 존재할 수 있으나 외부와 접속을 가지는 노드의 수가 동시에 50개를 넘지 않을 것이라 보면 한 노드당 최대 200개의 외부와 접속된 소켓이 동시에 존재할 수 있다.[2]) 한 노드에서 TCP Port는 2^{16}개가 사용될 수 있으므로 단 하나의 전역주소를 가지고 모든 전역 소켓을 처리할 수 있다. 따라서 [RM94]에서 제시된 재사용 전용 주소를 일반 노드에게 할당해 주고 각 C Class 지역망은 하나의 전역 주소만을 할당해 주면 된다.

이처럼 주소의 재사용은 IP 주소공간의 낭비를 줄이고 IP 주소 부족을 어느 정도 해소할 수 있다. 이때 포트-주소 변환기는 매우 적은 비용으로 기존 응용이나 망 구조의 수정 없이 사용자에게 투명한 서비스를 제공하기 위해 사용될 수 있다.

Ⅳ. 포트-주소 변환기의 설계 및 구현상의 고려사항

위에서 제시된 포트-주소 변환기는 IBM PC 호환기종과 packet driver상에서 구현되었다. 널리 사용되는 PC bridge에 비해 주소 변환등의 부차적 기능이 더 필요하므로 일반적인 bridge보다는 많은 하드웨어 자원을 요구한다. 386SX기종에서 실험해 본 바로는 기존 bridge에 비해 별다른 성능저하를 발견할 수 없었으나, 자세한 성능 평가는 차후 과제로써 수행되어야 할 것이다.

구현에 있어 가장 문제가 되는 사항으로는 다음과 같은 것들이 있다.

- TCP/IP 헤더의 수정에 따라 checksum의 적절한 수정이 필요하다.
- 내부 지역 소켓을 위한 포트 할당의 시점과 그 할당 취소의 시점은 어떻게 결정될 수 있는가.
- UDP를 지원하는 경우, 포트 할당/취소 시점의 결정이 어렵다.
- 지역 주소만을 가지는 서버로의 inbound request 중계를 위해 별도의 고려가 필요하다.
- FTP, ICMP등 응용 계층에서 IP 주소를 포함한 패킷을 발생시키는 경우 응용 계층에서의 처리가 불가피하다.

이러한 문제점들은 결국 포트-주소 변환기가 패킷으로부터 얻어낼 수 있는 정보가 한정되어 있다는데 기인한다. 이는 보안 강화를 위해 Packet Filtering Gateway를 구축할때의 문제점들과 유사한데[CB94]. 앞서 설명한 소켓 서버를 이용한 변환[KK92]의 경우 Rbind()에 의해 명시적인 정보의 제공이 이루어지는데 비해 투명한 주소 변환기는 패킷 헤더와 내용만으로 중계를 수행해야 하므로 상당한 어려움이 있다.

TCP/IP checksum의 계산은 매우 단순하므로, 헤더 checksum의 수정은 덧셈과 뺄셈의 조합을 통해 효율적으로 수행될 수 있다[EF94][TE93]. 본 포트-주소 변환기는 [TE93]에서 제시된 알고리듬을 구현하여 checksum문제를 해결하였다.

1. 포트-주소 변환 테이블의 할당/취소 시점

포트 할당/취소 시점에 대해서는 앞 절에서 두가지 방법이 언급되었다.

첫번째 DNS에서의 동적 주소 할당방식을 채택하면 주소 할당 시점이 명확해지며 내부로의 서비스를 제공할 수 있다는 장점이 있으나, 할당 해제 시점 결정을 위해 session을 감시하거나 명시적으로 주소를 반환하는 등의 방법을 별도로 도입해야 하므로 상당히 복잡해진다. 또한 이 방법은 모든 노드가 DNS에 미리 등록되어 있어야 하고 DNS query결과의 caching등에 따른 이름과 주소의 불일치를 야기할 수 있으므로 잇점이 적다.

두번째의 패킷 헤더 및 내용를 감시하는 방법을 사용할 경우 내부 지역 소켓의 TCP STATE 추적 알고리듬의 구현이 상당히 어려울 수 있다. 이 정보는 SYN, FIN등의 TCP 헤더 플래그를 통해 얻을 수 있다. 헤더 플래그 검출에 따른 TCP STATE추적을 상태 도표로 나타내면 그림 5와 같다. 본 포트-주소 변환기 구현에 사용된 알고리듬의 의사 코드 (pseudo code)는 부록 A.에 첨부되어 있다.

2) 실제로 주소변환기를 도입하려면 지역망 노드들의 활동특성에 대한 정량적인 분석이 반드시 수행되어야 한다.

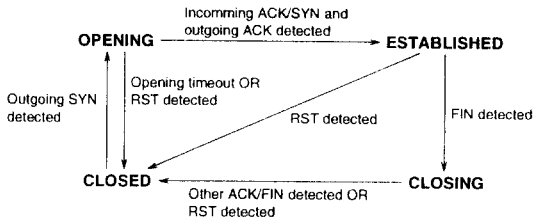


그림 5. 헤더 플래그 검출에 따른 TCP STATE transition diagram

2. UDP의 지원

UDP를 지원하기 위해 포트-주소 변환기를 확장하는 경우 할당/취소 시점을 결정하는 것은 TCP에 비해 훨씬 어렵다. UDP는 TCP와 달리 stateless 프로토콜이고, sequence number를 가지지 않기 때문에 정확한 session의 추적이 불가능하다. 이 경우 휴지기간의 측정에 의해 일정기간동안 같은 소켓에 대해 통신이 발생하지 않으면 UDP session의 종료로 보고 할당을 취소하는 휴지시간 기반 (idle time threshold based) 알고리듬을 사용할 수 있으나, 문턱치(threshold)의 선택이 어렵고 시작과 끝이 불명확한 UDP session의 특성상 TCP와 같은 정확한 동작은 기대할 수 없다.

현재 시제품 포트-주소 변환기는 문턱치를 2분으로 고정시킨 휴지시간 기반 알고리듬을 사용한다. 이 값은 NIS(Network Information System), NFS(Network File System)와 같이 응용 계층에서 자체의 time out을 가지는 경우 이들 응용의 time out 문턱치보다 충분히 길게 선택된 것이다.

time out을 두지 않거나 다른 문턱치를 가지는 응용을 지원하기 위해서는 별도의 고려가 필요하다.

3. Inbound request의 지원

본 주소 변환기의 포트-주소 변환 테이블은 패킷 플래그 검출을 통해 동적으로 할당/취소되나, 포트에 특정 주소를 항상 할당해 두는 정적 주소 할당 방식도 함께 채택하여 지역망 내부 서버에 대한 inbound connection request도 처리할 수 있게 하고 있다.

이 정적 할당은 포트-주소 변환기가 구동될때 지정되며 항상 유효하다. FTP, HTTP, TELNET등의 서버는 각자 well-known port로 알려져 있으므로, 외부망에 노출되어야 하는 서버의 주소를 해당 포트에 미리 할당해 두면 내부로의 connection request도 중계가 가능하다.

예를 들어 (www.our.domain, 80)에서 HTTP 서버가 대기하고 있다면, (gateway.our.domain, 80)으로 들어오는 모든 패킷을 (www.our.domain, 80)으로 변환하여 중계한다. 이때 서비스 요청자의 편의를 위해 외부에 서비스되는 DNS에는 www.out.domain을 gateway.our.domain의 주소로 등록해 둔다.

최근의 정보 검색 서비스는 서버의 부하 분산을 위해 동일한 서비스를 제공하는 복수의 서버 (server pool)로 구성되는 경향이 있다. 위의서비스 중계 방법을 확장하면 각 요청마다 이 서버들 중 한 곳을 선택하여 중계해 주는 동적 부하 균형기 (dynamic load balancing)로의 활용도 가능하다.[3]

4. 응용 프로토콜 별 고려사항

어떤 특정한 응용 프로토콜들은 포트-주소 변환기에 별도의 기능을 추가해야 중계가 가능하다. 발표된지 오래 되었으며 TCP/IP 망에서 가장 많이 사용되는 응용 중의 하나인 FTP의 경우를 살펴 보자.

FTP에서 data connection은 control connection과 별도로 생성된다. FTP client는 data connection의 생성이 필요할 때 control connection을 통해 자신의 IP 주소와 data connection을 위한 TCP 포트번호를 PORT 명령과 함께 server에게 알린 후 그 포트에서 대기(LISTEN)한다[PR85]. PORT 명령을 받은 FTP server는 명시된 주소와 포트 번호로 SYN을 보냄으로써 새로운 data connection을 생성하게 된다.

중복된 지역주소를 가지는 노드가 포트-주소 변환기를 통해 외부의 FTP서버로 접속한 경우, 이 노드의 ftp client는 자신의 주소—중복되어 있고 외부로 알려져서는 안되는—와 노드내의 미사용 포트번호를 하나 할당받아서 PORT명령과 함께 TCP 패킷으로 만들어 보낸후 그 포트에서 LISTEN상태로 들어간다. 따라서 올바른 data connection이 생성될 수 있게 하려면 포트-주소 변환기는 자신이 내부 노드를 대신하여 유지하고 있는

3) 분산형 WWW 시스템 개발에 있어 이 부하 균형기의 채택이 고려되고 있다.

connection이 FTP session임을 인식하고 PORT 명령을 발견할 때 마다 FTP 패킷내부에 적절한 수정을 가한 후 중계하여야 한다.

이때 PORT 명령에 따르는 IP주소와 포트번호는 ASCII문자로 표시되어 있다. 따라서 PORT 명령의 주소/포트 번호를 수정하게 되면 자릿수의 차이에 따라 TCP sequence number의 불일치가 생긴다. 이 불일치는 이후의 TCP 교신을 불가능하게 하므로, 각 connection마다 발생한 sequence number의 차이를 관리하여 이후 sequence number의 불일치가 발생하지 않게 수정한다. 또, 이와 함께 ACK 패킷의 acknowledgement number도 수정되어야 한다. 현재 구현된 시제품의 포트-주소 변환기는 원래의 sequence number와 수정된 패킷의 sequence number의 차이를 유지해 이후의 모든 패킷의 acknowledge number와 sequence number를 수정하므로써 FTP서비스를 원활히 중계해 준다.

SNMP등의 응용은 주소와 포트를 암호화하여 전송하는 경우가 있다. 이처럼 패킷 내용이 암호화되어 전송되는 경우는 포트-주소 변환기에서의 대응은 거의 불가능해진다. 그러나, 실제로는 SNMP패킷이 지역망 경계를 넘어 전달되어야 하는 경우는 거의 없다고 보는 것이 현실적이므로, 구현시의 고려 대상에서는 제외시켰다.

이와 같이, 투명한 포트-주소 변환기를 구현하기 위해서는 응용에 따라 패킷을 분류하고 응용별로 적절한 변환방법을 적용해야 하는데, 여기서 제시된 것 이외의 응용이 어떤 특수처리를 해야 하는지는 시제품의 시험과 적용을 통해 밝혀내어야 할 것이다.

Ⅴ. 결 론

포트-주소 변환기는 매우 적은 비용으로 기존 응용이나 망 구조의 수정 없이 포트-주소 변환기는 주소-주소 변환기, CIDR(Classless InterDomain Routing)과 함께 IP 주소의 재사용에 이용될 수 있으며, 이는 현재 진행중인 새 TCP/IP의 연구와 채택이 완료되기 전에 발생하는 IP 주소공간의 부족을 어느 정도 메꿀 수 있을 것으로 생각된다. 특히, 전역적으로 유일한 주소(Globally unique address)와 중복 사용되는 지역주소(Reused address)간의 일대일 대응을 원칙으로 하는 주소-주소 변환 방식에 비해 일대다 대응을 가능하게 하는 포트-주소 변환기는 더욱 높은 주소 재사용도를 제공한다. 이는 궁극적으로는 하나의 C Class 지역망이 단 하나의 전역주소를 할당받아 구축될 수 있음을 의미한다.

포트-주소 변환기에 의한 IP주소의 재사용은 다음과 같은 점에서 현실적인 안으로 여겨지며, 이는 IP주소 부족을 단기적으로 해결해 줄 수 있을 것으로 보인다.

- 포트-주소 변환기의 구현 및 망 구조의 재편성은 Packet Filtering Gateway의 설치와 같은 정도의 비용으로 이루어질 수 있다.
- 포트-주소 변환기는 기존 응용프로그램 및 망 구조의 수정없이 사용자에게 투명한 서비스를 제공한다.
- 적절한 시간 기반 알고리듬의 도입을 통해 UDP응용의 중계가 가능하다.
- 포트의 정적 예약 및 server pool방식의 도입을 통해 내부로의 서비스 요청도 중계가 가능하다.
- FTP등 많이 이용되는 서비스의 대부분은 기능상의 추가로 중계가 가능하다.
- 내부로의 요청을 근본적으로 제어하므로 방화벽 시스템이 제공하는 보안 효과를 얻을 수 있다.

또, 이 글에서 제시한 포트-주소 변환기는 다음과 같은 한계를 가지고 있으며, 향후 다음과 같은 문제점을 해결하기 전에는 전면적이고 장기적인 채택은 어려울 것으로 보인다.

- 어떤 응용들은 특별한 취급을 받아야 하며, 몇몇 응용은 포트-주소 변환기법을 적용할 수 없다.
- ICMP, SNMP, RIP등 다양한 프로토콜에 대한 고려가 필요하다.
- 기반이 되는 하드웨어에 따른 단위시간당 중계 능력의 평가가 필요하다.

부 록

A. 포트-주소 변환기 TCP STATE 추적 알고리듬

```
main()
{
    do {
        packet = get_next_packet();
```

```
        if (packet is from inner network) {
            newpacket = i_to_o(packet);
            send_packet(newpacket, outer network);
        } else {
            newpacket = o_to_i(packet);
            send_packet(newpacket, inner network);
        }
    } while(end);
}

i_to_o(packet)
{
    if (packet.flag = SYN) {
        alloc_port(packet);
    }

    newpacket.src_ip = gateway_ip;
    newpacket.src_port = get_gatewayport
    (packet.src_ip, packet.src_port);

    /* we should carefully design an algorithm to handle
        graceful shutdown of TCP connection..TBD */
    if ((packet.flag = FIN) || (packet.flag = RST)) {
        prepare_release_port(packet);
    }
    if (packet.flag = ACK) {
        if (the packet is ACK to previous FIN..etc) {
            .....
            release port w.r.t. packet flags;
        }
    }

     return newpacket;
}

o_to_i(packet)
{
    if (port_table[packet.dst_port] = NULL) {
        /* no entry in table, which means there is no TCP client
            in inner network, so send back reset to source */
            newpacket.dst_ip = packet.src_ip;
            newpacket.dst_port = packet.src_port;
            newpacket.src_ip = packet.dst_ip;
            newpacket.src_port = packet.dst_port;
            newpacket.flag |= RST;
            return newpacket;
    }

    newpacket.dst_ip = port_table[packet.dst_port].inner_ip;
    newpacket.dst_port = port_table[packet.dst_port].inner_port;

    /* again, we should redesign following algorithm to handle
        graceful close of TCP connection */
    if ((packet.flag = FIN) || (packet.flag = RST)) {
        prepare_release_port(packet);
    }

    return newpacket;
}
```

參考文獻

[CB94] W. R. Ceswick and S. M. Bellovin. Firewalls and Internet Security, section 3, pages 49-83. Addison-Wesley, 1994.

[EF94] K. egavang and P. Francis. The ip network address translator(NAT). RFC1631, May. 1994.

[FLYV93] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless inter-domain routing(CIRD): An address assignment and aggregation strategy. RFC1519, Sep. 1993.

[KK92] D. Koblas and M. R. Koblas. SOCKS. In USENIX security symposium proceedings III, Sep. 1992.

[PR85] J. Postel and J. Reynolds. File transfer protocol. RFC959, Oct. 1985.

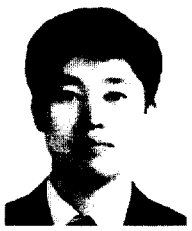
[RM94] Y. Rekhter and B. Moskowitz. Address allocation for private internets. RFC1597, Mar. 1994.

[TCP81] Transmission Control Protocol. RFC793, Sep. 1981.

[TE93] P. F. Tsuchiya and Tony Eng. Extending the ip internet through address reuse. ACM Computer Communication Review, Jan. 1993.

金逸煥(Il Hwan Kim) 정회원

1971년생
1994년 2월: 서울대학교 계산통계학과 졸업(학사)
1996년 2월: 서울대학교 대학원 계산통계학과 졸업 예정(석사)



廉憲榮(Heon Young Yeom)정회원

1961년생
1984년 2월 : 서울대학교 계산통계학과 졸업(학사)
1986년 5월 : 美 Texas A&M Univ.(석사)
1992년 12월 : 美 Texas A&M Univ.(박사)
1992년 10월~1993년 8월 : 삼성 데이타 시스템 선임연구원
1993년 9월~현재 : 서울대학교 자연과학대학 전산과학과, 조교수