지구물리 제 4권 제 4호, p. 239-249 (2001년 12월)

TCP 세그멘트 정보를 이용한 NAT에 대한 연구

황재용 · 주기호
배재대학교 IT 공학부

요 약

NAT (Network Address Translation)은 임의의 사설 IP 주소를 인터넷의 정식 IP 주소로 변환하는 메커니즘으로써 네트워크 보안 및 인터넷 주소의 절약이 주요 기능이다. 일반적으로 NAT는 패킷 헤더의 IP정보를 이용하여 이러한 기능을 수행한다. 하지만 특정 응용 프로토콜들은 패킷의 헤더뿐만 아니라, 데이터 부분에도 종단간 통신에 필요한 정보가 존재하기 때문에, NAT는 이러한 정보를 적절하게 변환해 주어야 할 필요가 있다. 본 논문에서는 포트프락시(port proxy)서버를 이용하여, 패킷의 데이터에 있는 가상 IP정보를 실제 IP정보로 변환하여, 외부와 통신이 원활하게 이루어 질 수 있도록 하는 방안을 제시하고 실제로 구현하였다.

Hwang, Jae Yong Joo, Giho, A Study of an NAT using the TCP Segment Information

ABSTRACT : NAT (Network Address Translation) is an IP address modification protocol that translates private IP addresses into authentic Internet addresses. The main features of NAT are to improve network security and to save IP addresses. Generally speaking, in order to perform its functionality, NAT uses the address information in the packet header. Certain application protocols, however, use the information in the packet data as well as the information in the packet header to perform end-to-end communication. Therefore, to support these types of application protocols, NAT should be able to perform appropriate translation of protocol information in the packet data. In this thesis, we design and implement a method which translates virtual IP information in the packet date into real IP information by using port proxy server.

key words : network address translation, IP address, masquerading, port address translation

1. 서 론

TCP/IP 네트워크상의 모든 호스트들은 인터넷에 접속하기 위해 고유한 IP주소를 배정받아야 한다. 하지만 인터넷의 호스트의 급격한 팽창은 한정된 IP 주소공간의 고갈을 초래하였고, 그 결과 현재 IP 주소공간은 거의 소진된 상황이다. 이에 대한 해결책으로 IPv6가 제시되고 있으나 이것이 완전하게 사용되기까지는 얼마간의 시간이 필요할 전망이다. 따라서 한정된 IP 주소공간에서의 IP 주소의 적절한 사용을 위한 여러 가지 방안들이 제시되었으며, 그 중 하나가 NAT (Network Address Translation)이다[1].

NAT는 임의로 할당된 주소를 인터넷의 공인 IP주소로 변환하며, 네트워크 내부에서 사용되는 사설 네트워크 어드레스를 가진 패킷은 NAT기능을 가진 장치를 통해서 외부로 전송된다. NAT는 서버에 공인호스트 IP 주소를 할당하고 다수의 내부 호스트가 인터넷으로 트래픽을 송수신하게 하는 IP주소 공유 방법이다. NAT는 두 가지의 다른 형태로 구분할 수 있는데, 하나는 시작점 NAT(source NAT)이며 다은 하나는 목적지 NAT(destination NAT)이다. SNAT는 첫 패킷의 시작점 주소를 변경 할 때 사용되며, 주소 변환이 언제나 라우팅 후에 이루어지고 패킷이 바깥으로 나가기 직전에 이루어 진다. SNAT는 패킷의 헤더의 가상 IP에 대한 정보만을 공인 IP로 변경을 한다. 그러나, 일부 어플리케이션 들은 패킷의 헤더뿐만 아니라 데이터에도 가상 IP정보를 사용하기 때문에 SNAT를 이용하여 패킷의 헤더에 있는 가상 IP정보만을 변환할 경우, 이러한 애플리케이션을 원활하게 지원 할 수 없기 때문에 이러한 문제점을 해결 할 수 있는 포트프락시(port proxy) 방식이 개발되었다[2].

NAT 포트프락시서버는 NAT환경에서 iptables의 포트 재직접(port redirect) 옵션을 적용하여 가상 IP의 데이터에 들어있는 특정 포트의 패킷을 서버로 전달한다. 그리고, 서버로 들어온 패킷의 데이터부분을 분석하여, 가상 IP가 포함되어 있으면 이를 공인 IP로 변경하고, 소켓을 이용하여 외부에 연결하려고 하는 서버와 재접속을 시도하게 된다. Iptables는 리눅스 커널 2.4 버전 이후부터 새로운 기능으로 패킷 여과기능(filtering)을 지원한다. 이 Iptables은 기존의 2.2 버전 커널의 Ipchains보다 더욱 확장된 기능을 가지고 있다[3]. 기본적인 패킷여과 방식은 변하지 않았지만, NAT라고 하는 라우팅을 지원하는 부분이 들어가 있다. 이러한 조합은 기존의 Ipchains 보다 훨씬 강력한 방화벽을 구축할 수 있게 해준다. 패킷여과 방화벽(packet filtering firewall)은 네트워크로 흘러 들어오는 패킷들의 헤더와 내용을 살펴보고 특정한 룰을 적용하고 이에 일치하는 패킷에 원하는 정책을 부여하는 방식의 방화벽이다. 이러한 방식의 방화벽은 현재 인터넷의 기반인 TCP/IP의 데이터 흐름 방식이 패킷 단위에 기초하고 있기 때문에 매우 유용하다.

본 논문에서는 리눅스 기반의 NAT환경에서 NAT 포트프락시 서버를 개발하여 패킷

의 데이터안에 있는 가상 IP를 공인 IP로 적절하게 변환하고, 프락시 서버와 외부 다른 통신 서버와 단절 없이 통신을 원활하게 유지시켜주는 방식을 고안하고, 실제 실험환경을 구축하여 시험한다. 본 논문의 구성은 다음과 같다. 2장에서는 NAT에 대한 일반적인 사항을 기술하고, 3장에서는 TCP/IP에 대한 방화벽 및 시험에 사용될 iptables에 대한 기술을 하며 4장에서는 시험환경 구축하고 실험하여 결과를 분석한다. 5장에서 결론 및 향후 과제를 기술한다.

2. 네트워크 주소변환 프로토콜

2.1 NAT의 정의

IPv4에 기반한 현재 인터넷에서 주소공간의 부족과 보안상의 문제로 많은 네트워크에서 IETF에서 권장하는사설 IP주소 공간(10.0.0.0/255.0.0.0, 172.16.0.0/255.240.0.0, 192.168.0.0/255.255.0.0)을 사용하고 있다. 사설 IP 주소를 이용하는 네트워크의 호스트가 인터넷에 접속을 하거나, 인터넷에서 이러한 네트워크 내부의 호스트에 접속하기 위해서 사설 IP 주소를 공인 IP 주소로 변환하는 기능이 필요하다[2]. NAT는 특정한 IP 주소를 한 그룹에서 다른 그룹으로 매핑하는 기능이다. 주소를 N-to-N 형태로 매핑하는 경우를 정적 NAT라 하고 M-to-N(M>N)를 동적 NAT라고 한다. 네트웍 주소 포트 변환은 기본적인 NAT의 확장기능으로 여러 가지 네트웍 주소와 TCP/UDP 포트를 단일의 네트웍 주소와 TCP/UDP 포트로 변환한다. 이러한 방식을 N-to-1 매핑이라고 하며 리눅스의 IP 마스커레이딩도 이러한 방식을 이용한다. 리눅스에서 NAT를 통한 가상 서버는 네트웍 주소 포트 변환을 통해 수행한다. NAT를 실행하는 링크는 패킷을 어떻게 조작했는지를 기억하며 응답 패킷이 지나갈 때 그 응답패킷을 원래대로 되돌려 모든 작동이 가능하도록 한다.

NAT를 사용하는 목적에는 2가지가 있는데, 첫째는 인터넷의 공인 IP 주소를 절약할 수 있다는 점이고 둘째는 인터넷이란 공공 망과 연결되는 사용자들의 고유한 사설망을 침입자들로부터 보호할 수 있다는 점이다.

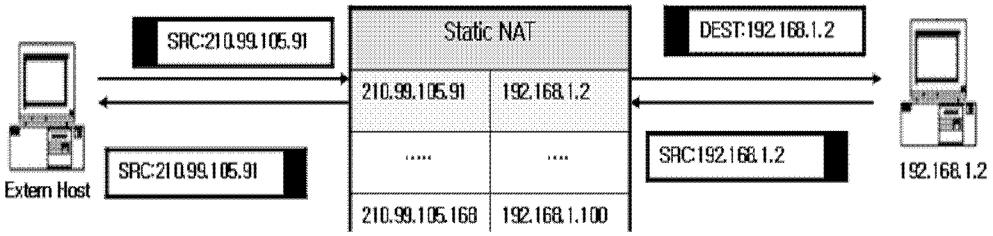
인터넷의 공인 IP 주소는 한정되어 있기 때문에 가급적 이를 공유할 수 있도록 하는 것이 필요한데 NAT를 이용하면 사설 IP 주소를 사용하면서 이를 공인 IP 주소와 상호 변환할 수 있도록 하여 공인 IP 주소를 다수가 함께 사용할 수 있도록 함으로써 이를 절약할 수 있는 것이다.

공개된 인터넷과 사설 망 사이에 방화벽을 설치하여 외부 공격으로부터 사용자의 통신망을 보호하는 기본적인 수단으로 활용할 수 있다. 이 때 외부 통신망 즉 인터넷망과 연결하는 장비인 라우터에 NAT를 설정할 경우 라우터는 자신에게 할당된 공인 IP 주소만 외부로 알려지게 하고, 내부에서는 사설 IP 주소만 사용하도록 하여 필요시에 이를 서로 변환시켜 준다. 따라서 외부 침입자가 공격하기 위해서는 사설망의 내부 사설 IP 주소를 알아야 하기 때문에 공격이 불가능해지므로 내부 네트워크를 보호할 수 있다.

2.1 NAT의 방식

정적 NAT(Static NAT)

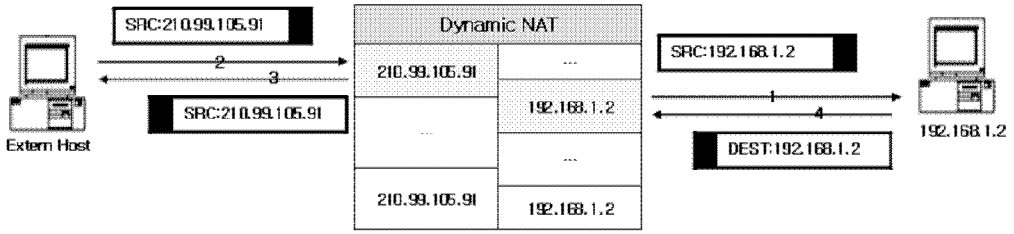
공인 IP 주소 와 사설 IP 주소가 1:1로 정적으로 고정된 룰에 의해서 변환되는 방식을 정적 NAT라 한다.. [그림 1]에서 외부 호스트가 송신한 패킷의 목적지 주소가 사설 네트워크로 전달되면서 정적 NAT에 의해서 사설주소로 변환되고 있다. 마찬가지로 내부 호스트 (192.168.1.2)가 송신한 패킷의 소스주소가 사설주소에서 공인주소로 변환되어서 인터넷으로 전달되고 있다[4].



[그림 1] Static NAT

동적 NAT

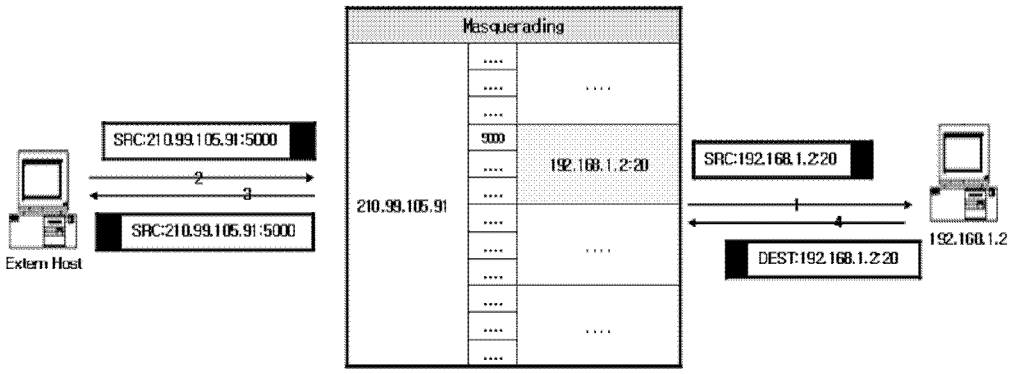
공인주소의 수가 가상주소의 수보다 적은 경우에도 적용될 수 있는 방법으로 내부 호스트 192.168.1.2에서 발생한 패킷이 외부 인터넷으로 전달되는 경우에 사설주소로 되어 있는 소스주소를 대치할 공인주소를 가용한 공인주소 풀(pool)에서 동적으로 할당하는 방식이다. 이렇게 동적으로 할당된 공인주소는 해당 접속이 유지될 때까지만 의미 있는 값이고, 접속이 끊어진 후에는 이 공인 주소를 통해서 내부 호스트 192.168.1.2를 접근할 수 없다. 이러한 특성이 보안 관련 기능에 사용되는 경우도 있다. 동적 NAT를 사용할 때 외부 호스트에서 내부 호스트 192.168.1.2를 접근 할 수 있게 하기 위해서 NAT 서버는 특정 공인 주소를 192.168.1.2와 연관시켜 기록해 놓아야 한다. [그림 2]는 이러한 과정을 보이고 있다. 내부 호스트 192.168.1.2에 응용서버를 운용할 때는 이러한 방식을 사용한다[4].



[그림 2] 동적 NAT

IP 마스커레이딩 (masquerading)

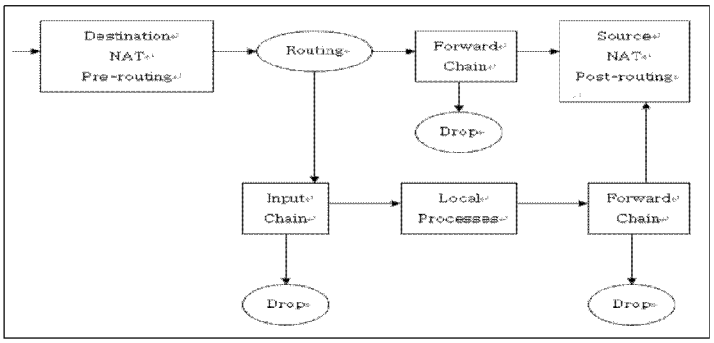
IP 마스커레이딩은 동적 NAT의 특별한 경우로 공인주소를 하나만 가지고 NAT를 운용하는 방법이다. [그림 3]에서 보인 것 같이 모든 사설주소가 하나의 공인주소로 변환되며, 포트 번호를 이용하여 여러 개의 사설주소와 매핑을 한다. 포트번호의 할당은 동적으로 이루어진다. 포트번호를 이용하여 NAT를 수행하므로 PAT(port address translation) 혹은 NAPT(network address port translation)등으로 부르기도 한다.



[그림 3] 포트 주소 변환(PAT)

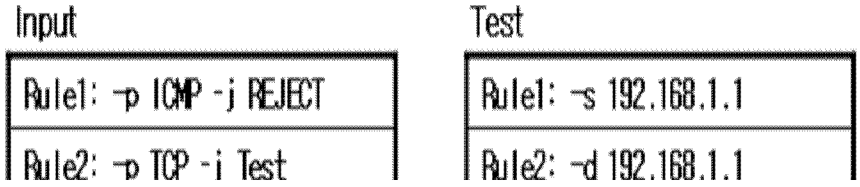
3. 패킷 필터링

iptables 유틸리티는 netfilter 필터링 룰을 설정할 때 사용되며, IP 필터링과 네트워크 주소 변환을 모두 설정 할 수 있다. Iptables 에는 기존의 2.2 대 버전 리눅스 커널의 ipchains 보다 더욱 확장된 기능이 추가되어 있다. 기본적인 패킷 필터링 방식은 변하지 않았지만, 네트워크주소변환 기능이 포함되어 있다. 이러한 조합은 기존의 ipchains을 이용한 것보다 훨씬 더 강력한 방화벽을 구축할 수 있게 해준다.[5]



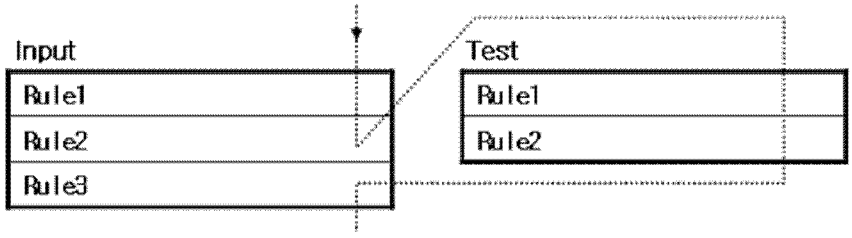
[그림 4] iptables의 모양

Iptables은 [그림 4]에서 도시한 바와 같이 NAT와 몇 개의 체인들로 구성되어 있다. 체인(chain)은 패킷을 여과하기 위한 필터링 룰들이 들어있는 곳으로써 이곳에서 패킷을 폐기할지 아니면 접수할 지가 결정된다. 체인 내부의 룰들은 사용자가 간단히 명령어를 이용하여 생성할 수 있다. 체인의 내부 내용을 살펴보면 다음과 같다.



[그림 5] 체인의 내부 내용

Input과 Test라는 체인을 간단하게 만들어 보았다. Input 체인은 3개의 룰로 구성되어 있고, Test 체인은 2개의 룰로 구성되어있다. [그림 5]에서 보이듯이 라우팅 단계를 거쳐 일단 내부로 들어온 패킷은 무조건 input이란 체인으로 유입된다.. Input 체인 룰에서 Rule 1은 ICMP 프로토콜로 들어오는 패킷은 무조건 REJECT 하라고 되어있다. Rule 2는 TCP 프로토콜로 들어오는 패킷은 Test라는 체인으로 넘겨서 처리하게 되어있다. Rule 3은 UDP 프로토콜로 들어오는 패킷은 DENY 하라고 되어있다. REJECT는 패킷을 버리지만, 이 패킷이 버려졌음을 상대방에게 알려준다. 이에 비해 DENY는 패킷을 버리면서 알리지 않는다. Test 체인의 Rule 1은 192.168.1.1로부터 들어온 패킷을 가리킨다. Rule 2는 목적지가 192.168.1.1 인 패킷을 가리킨다. 이러한 체인들에게 TCP 패킷이 1개 들어왔다고 가정해 보면 [그림 6]과 같이 흐르게 된다.



[그림 6] 패킷이 흘러가는 순서

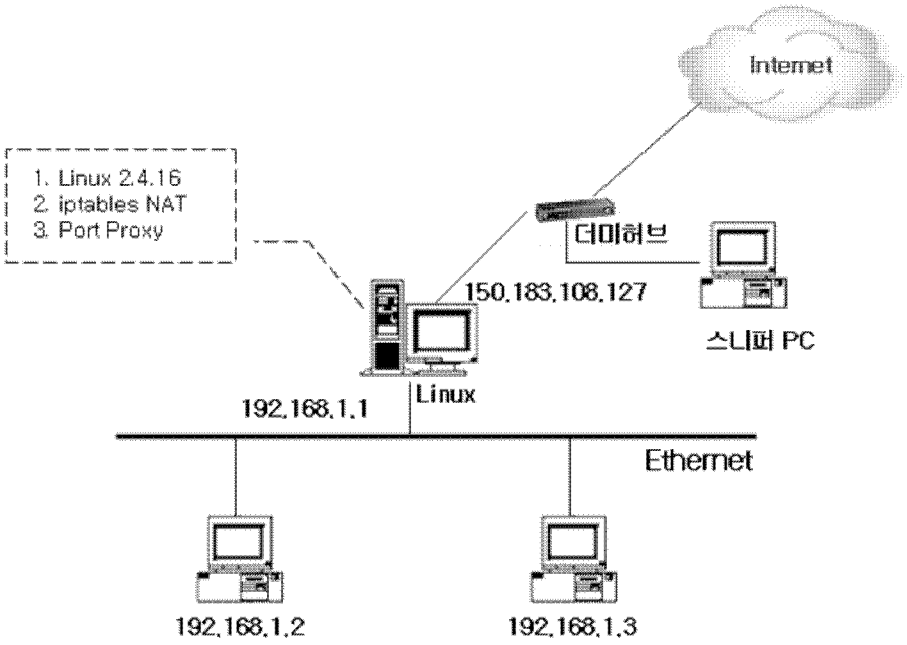
TCP프로토콜 패킷이므로 Input 체인의 Rule 1은 지나가고 Rule 2에서 걸리게 된다. 그렇게 되면 Test 체인으로 건너가게 되어 Rule 1과 Rule 2를 검사해보고 다시 Input으로 돌아와서 Rule 3을 통해 지나가게 된다. 위에서 본 것과 같이 체인 안에는 적절한 룰이 있고 이러한 룰들에 의해 패킷이 다루어짐을 알 수 있다. 이러한 Rule 들은 아주 간단한 명령어로 추가, 삭제, 유지, 보수가 가능하다.[3]

4. 시험 환경 구축

4.1 실험 네트워크 설치

본 연구에서는 실험을 위해서 리눅스 커널 2.4.16을 사용하였다. NAT 환경을 위해서 iptables를 인스톨 한 뒤 커널에서 NAT를 가능하게 하는 모듈인 Netfilter의 조정옵션을 선택하여 리눅스 커널을 컴파일 하였다. 시험을 하기 위한 네트워크 응용은 스타크래프트 배틀넷을 사용하였다. NAT환경에서 스타크래프트 배틀넷을 사용할 때, 외부로 나가는 패킷의 데이터에 NAT내에 있는 IP정보(192.168.XXX.XXX)가 들어가 있기 때문에, 그 패킷을 받은 외부 서버가 이 정보를 이용할 경우에는 패킷을 전송해온 호스트와 통신을 할 수 없게 된다. 포트프락시는 패킷이 외부로 나갈 때, 패킷의 TCP 데이터에 들어있는 가상 IP정보를 NAT서버의 공인 IP정보로 변경하여 내보내는 역할을 한다.

[그림 7]은 시험 시스템을 위한 구성도이다. IP 주소가 각각 192.168.1.2, 192.168.1.3 인 두 개의 호스트를 설치하고, NAT 기능을 수행할 호스트는 내부 IP주소는 192.168.1.1 외부 공인 IP 주소는 150.183.108.127 로 설정하였다. 150.183.108.127 쪽에 더미허브를 설치하여 인터넷으로 나가고 들어오는 패킷을 스니핑(sniffing)할 수 있도록 하였다.

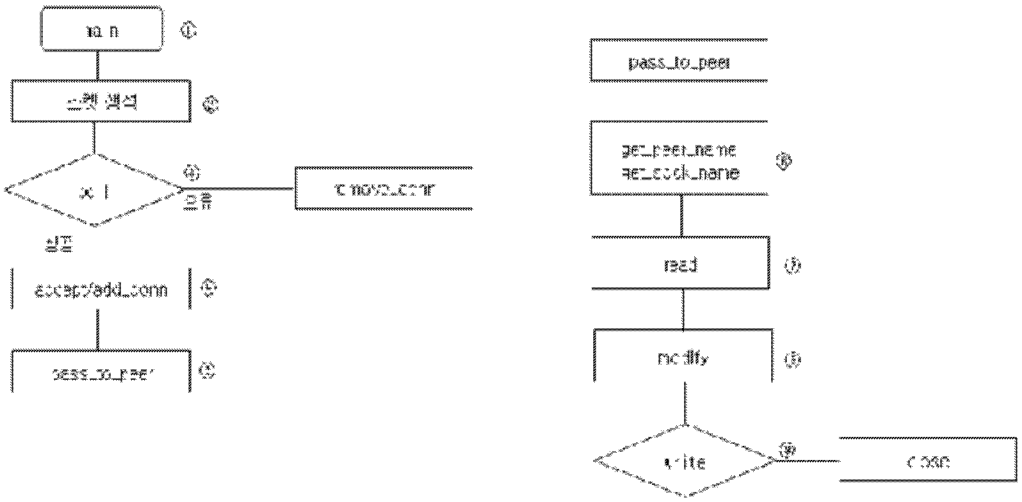


[그림 7] 실험 네트워크 구성도

4.2 포트프락시의 동작

포트프락시는 iptables의 포트 재집적룰에 의해 6112 포트로 유입되는 패킷들을 처리하기 위해 데몬으로 실행되며, 유입된 패킷을 분석하여 소스주소, 소스포트, 목적지주소, 목적지정보를 캐쉬에 보관한다.

포트 프락시는그리고, 패킷의 데이터를 분석하여, 가상 IP 정보가 있으면 공인 IP 정보로 변경한 후 캐쉬에 저장된 소스/목적지정보를 이용하여 재접속을 시켜주고 프로그램은 종료된다. 프로그램의 진행 순서는 [그림 8]와 같다.



[그림 8] 포트프락시의 수행 순서도

첫째, 함수가 시작되면 소켓을 생성하고 데몬으로 프로세스를 실행한다. 둘째, poll()함수를 이용하여 서버로 유입되는 패킷을 검사한다. 셋째, gettranssockname(), getpeername()함수를 이용하여, 연결하고자 하는 연결 목록을 생성하고, 연결된 목록이 있으면 외부에 연결된 공인 IP 및 포트정보를 을 읽어서 전달 해 준다. 넷째, getpeername()함수로, 소스/목적지의 IP및 포트 정보를 알아낸다. 다섯째, 현재 소켓을 읽어서 가상 IP정보가 있으면 공인 IP 정보로 변경한다. 여섯째, 공인 IP 정보로 변경된 후, 외부 인터넷과 연결된 소켓 번호를 찾아내 쓰고 프로그램이 종료된다.

5. 실험 및 결과 분석

5.1 포트프락시 통과하지 않은 경우

[그림 10]은 포트프락시를 통과하지 않았을 때, 패킷의 헤더를 보여준다. NAT를 거치면서 소스주소가 마스커레이딩 된 것을 볼 수 있다.

```
------------  IP Header  ------------
IP: Version = 4
IP: Header length = 20
IP: Differentiated Services (DS) Field = 0x00
IP:     0000 00.. DS Codepoint = Default PHB (0)
IP:     .... ..00 Unused
IP: Packet length = 90
IP: Id = ec61
IP: Fragmentation Info = 0x4000
IP:     .1.. ....  .... .... Don't Fragment Bit = TRUE
IP:     ..0. ....  .... .... More Fragments Bit = FALSE
IP:     ...0 0000  0000 0000 Fragment offset = 0
IP: Time to live = 127
IP: Protocol = TCP (6)
IP: Header checksum = CB42
IP: Source address = 150.183.108.127
IP: Destination address = 213.248.106.202
```

[그림 10] IP 패킷 헤더

[그림 11]은 포트프락시를 통과하지 않았을 때 TCP 데이터를 보여주고 있다. TCP 6112의 포트 정보를 보여주고 있다. 여기에서 c0 a8 01 02부분이 내부 네트워크에 있는 호스트의 IP 주소를 나타내고 있다. c0 a8 01 02는 16진수로서 10진수로 변환하면 192 168 1 2의 가상 IP가 된다.

```
00 e0 63 55 96 b5 00 05   ea 00 0f d9 08 00 45 00   ..cU.... ......E.
00 5a ec 61 40 00 7f 06   cb 42 96 b7 6c 7f d5 f8   .Z.a@... .B..l...
6a ca 12 01 17 e0 b5 87   f5 14 ce c4 d3 a1 50 18   j....... ......P.
fa f0 56 77 00 00 ff 50   32 00 00 00 00 00 36 38   ..Vw...P 2.....68
58 49 50 58 45 53 c7 00   00 00 00 00 00 00 c0 a8   XIPXES.. ........
```

[그림 11] 포트프락시를 통과하지 않았을 경우의 TCP 데이터

5.2 포트프락시를 통과했을 경우

[그림 12]는 포트프락시를 통과했을 때 IP 헤더를 보여주고 있다. NAT 기능에 의하여 소스주소가 공인 IP로 변경되었음을 알 수 있다.

```
------------  IP Header  ------------
IP: Version = 4
IP: Header length = 20
IP: Differentiated Services (DS) Field = 0x00
IP:     0000 00.. DS Codepoint = Default PHB (0)
IP:     .... ..00 Unused
IP: Packet length = 103
IP: Id = 2cc9
IP: Fragmentation Info = 0x4000
IP:     .1.. ....  .... .... Don't Fragment Bit = TRUE
IP:     ..0. ....  .... .... More Fragments Bit = FALSE
IP:     ...0 0000  0000 0000 Fragment offset = 0
IP: Time to live = 64
IP: Protocol = TCP (6)
IP: Header checksum = C9CE
IP: Source address = 150.183.108.127
IP: Destination address = 213.248.106.202
```

[그림 12] IP 패킷 헤더

[그림 13]은 포트프락시를 통과하였을 때 TCP 6112의 포트 정보를 보여주고 있다. 여기에서 96 b7 6c 7f 부분이 내부 네트워크에 있는 호스트의 IP 주소를 나타내고 있다. 96 b7 6c 7f 는 16진수로서 10진수로 변환하면 150 183 108 127의 공인 IP로 바뀌어서 외부의 다른 네트워크와 연결을 하게 된다.

```
00 e0 63 55 96 b5 00 05    ea 00 0f d9 0b 00 45 00    ..cU.... ......E.
00 67 2c c9 40 00 40 06    c9 ce 96 b7 6c 7f d5 f8    .g,.@.@. ....l...
6a ca 0c 0c 17 e0 c6 47    02 69 c9 87 09 bc 80 18    j......G .i......
16 d0 4b 22 00 00 01 01    08 0a 00 02 2e cf 00 00    ..K".... ........
00 00 01 ff 50 32 00 00    00 00 00 36 38 58 49 50    ....P2.. ...68XIP
58 45 53 c7 00 00 00 00    00 00 00 96 b7 6c 7f e4    XES..... .....l..
```

[그림 13] 포트프락시를 통과하였을 때 TCP 데이터

6. 결론 및 향후 연구 과제

IP 어드레스의 고갈 현상은 이미 오래 전부터 예상되어온 문제로 그 해결 방안으로 여러 가지가 제시되었다. 네트워크 주소변환 프로토콜은 IP 어드레스 고갈 현상을 해결하는 한 방법은 사용된다. 일반적인 어플리케이션을 사용할 때 NAT는 문제가 되지 않지만, 네트워크 게임이나 인터넷폰등 패킷의 데이터에 가상 IP정보를 전송하는 특정 프로토콜을 지원하는 데 어려움이 있다.

본 연구에서는 리눅스를 사용하여 NAT를 구성하고, 포트프락시서버를 개발하여 시험을 하였다. 리눅스를 라우터로 동작하도록 설정을 하고, 포트프락시를 사용하여 시험한 어플리케이션은 인터넷 게임으로 많은 사용자가 이용하는 스타크래프트 베틀넷 v1.10을 사용하였다. 스타크래프트 베틀넷에 사용자가 로그인을 하면서 패킷을 인증서버에 요청한다. 인증서버로 요청할 때 패킷은 iptables의 마스커레이딩과 포트재집적룰을 이용해서 포트프락시로 보내지게 된다. 포트프락시 서버는 받은 패킷을 분석하여 이 중 TCP 데이터의 가상 IP정보를 공인 IP정보로 바꾸고, 패킷을 변경하기 전에 기억된 외부/내부 IP 주소와 포트번호를 이용해서 외부 인증서버와 원활한 통신을 수행하게 한다. 추후 UDP 데이터뿐만 아니라 다중 포트를 사용하는 패킷들에 대해서도 대응을 하고, 음성채팅, 화상채팅과 같은 VoIP, SIP를 사용하는 프로토콜의 TCP 가상 IP정보를 공인 IP정보로 변경할 수 있도록 확장하고자 한다.

참고문헌

1. K. Egevang. □□The IP Network Address Translator (NAT)□□, RFC 1631
2. G. Tsirtsis, et al.,□□Network Address Translation-Protocol Translation (NAT-PT),□□ IETF RFC 2766,
3. http://www.machi.pe.kr/system/security/document/report/kernel_nat.htm
4. http://www.hicore.com/doc/pdf/NAT_ Filtering.pdf
5. http://www.machi.pe.kr/system/security/document/report/kernel_nat.htm