6TALK : NAT-PT/SIIT 및 향상된 ALG의 구현

이주철* · 홍용근 · 신명기 · 김형준

한국전자통신연구원

6TALK : Implementation of NAT-PT/SIIT and enhanced ALG

Joo-Chul Lee* · Yong-Geun Hong · Myung-Ki Shin · Hyng-Jun Kim

Electronics and Telecommunications Research Institute

E-mail : {rune, yghong, mkshin, khj}@etri.re.kr

요 약

본 논문에서는 6TALK (IPv6 TrAansLator of Krv6) 프로젝트의 구현에 대한 내용과, 이를 이용하여 IPv6 망이 기존의 IPv4망이나, 다른 IPv6망과 자연스럽게 연동할 수 있는 시나리오에 관해 기술한다. 이미 잘 알려진 바와 같이 현재의 IPv4 인터넷은 인터넷 사용자의 폭발적인 급증과 더불어 한계를 맞이하고 있다. 이를 극복하기 위해 제안된 IPv6는 현재 여러 기술적인 표준화와 실험 망에서의 테스트 과정을 마무리 짓고 있는 단계이다. 앞으로의 IPv6의 보급에 있어 가장 중요한 과정은 기존에 널리 존재하고 있는 IPv4망과 자연스럽게 공존하며 그 적용 범위를 넓혀갈 수 있도록 하는 전이 기술을 개발, 보급하는 일이다. 이를 위해서 IETF에서는 NGTrans라는 WG을 별도로 만들어 IPv6로의 전환에 관련된 기술들을 표준화하고 있다. 현재 전 세계적으로 다양한 IPv6 관련 전환기술 프로젝트들이 진행되고 있으며, 본 논문에서 소개하는 6TALK 프로젝트도 NGTrans WG에서 소개하는 기술들을 바탕으로 망의 구성에 따라 IPv4망과 IPv6망이 자연스럽게 공존할 수 있도록 하는 솔루션들을 제공하는 것을 최종 목표로 하는 프로젝트이다. NGTrans WG에서 제안한 전환기술은 듀얼스택, 터널링, 변환기 등과 같은 기본 기술들과, 이 기본 기술을 확장 응용한 기술들로 이루어져 있다. 6TALK는 이러한 전환기술들을 복합적으로 사용하며, 네트워크의 가장자리(edge)에 존재하는 IPv6망이 기존 IPv4망이나, IPv4 망을 사이에 두고 존재하는 IPv6망과 연동할 수 있도록 하는 기능을 제공한다. 6TALK은 전이기술로 NAT-PT/SIIT와 몇몇 응용레벨 프로토콜에 대한 ALG, 구현환경으로는 커널 2.4.18버전이 탑재된 리눅스를 이용하고 있다.

ABSTRACT

This paper describes implementation of IPv6-IPv4 transition toolbox named as 6TALK(IPv6 TrAansLator of Krv6) and some scenarios using 6TALK which enables IPv6 island to connect other IPv6 island or IPv4 island seamlessly. 6TALK implements some transition mechanisms suggested in NGTrans Working Group of IETF. Those mechanisms are composed of basic mechanism, tunneling, and applied mechanism such as DSTM. 6TALK provides functions which enable IPv6 network at the edge of existing network to communicate with IPv4 network by using these transition mechanisms. As major transition mechanisms in 6TALK we adopt NAT-PT/SIIT and DSTM/DSTM options and as implementation environment we use Linux Kernel 2.4.18 and Netfilter framework. Software modules implemented in Linux kernel was ported to hardware box using Motorola MPC 8260 processor. The transition mechanisms used in 6TALK are the ones predicted to be used in initial transition step to IPv6.

키워드

IPv6, transition, NATPT, SIIT, DSTM, Linuxt

Ⅰ. 서 론

본 논문에서는 6talk (IPv6 TrAansLator of Krv6) 프로젝트의 구현에 대한 내용과, 이를 이용하여 IPv6 망이 기존의 IPv4망이나, 다른 IPv6망과 자연스럽게 연동할 수 있는 방법에 관해 기술한다. 이미 잘 알려진 바와 같이 현재의 IPv4 인터넷은 인터넷 사용자의 폭발적인 급증과 더불어 한계를 맞이하고 있다. 이를 극복하기 위해 제안된 IPv6는 현재 여러 기술적인 표준화와 실험 망에서의 테스트 과정을 마무리 짓고 있는 단계이다. 앞으로의 IPv6의 보급에 있어 가장 중요한 과정은 기존에 널리 존재하고 있는 IPv4망과 자연스럽게 공존하며 그 영역을 넓혀갈 수 있도록 해주는 전이 기술을 개발, 보급하는 일일 것이다. 이를 위해서 IETF에서는 NGTrans라는 WG을 별도로 만들어 IPv6로의 전이에 관련된 기술들을 표준화하고 있다. 현재 전 세계적으로 다양한 IPv6 관련 전이기술 프로젝트들이 진행되고 있으며, 본 논문에서 소개하는 6talk 프로젝트도 NGTrans WG에서 소개하는 몇몇 기술들을 바탕으로 망의 구성 상황에 따라 IPv4망과 IPv6망이 자연스럽게 공존할 수 있도록 하는 솔루션들을 제공하기 위한 취지로 만들어진 프로젝트이다. NGTrans WG에서 제안한 전이 기술은 듀얼스택, 터널링, 변환기등과 같은 기본 기술들과, 이 기본 기술을 확장 응용한 기술들로 이루어져 있다. 6talk는 이러한 전이 기술들을 복합적으로 사용하며, 네트워크의 가장자리(edge)에 존재하는 IPv6망이 기존 IPv4망이나, IPv4 망을 사이에 두고 존재하는 IPv6망과 연동할 수 있도록 하는 기능을 제공한다.

6talk의 구현 세부사항에 관해서는, 전이기술로 NAT-PT/SIIT와 DSTM이, 구현환경으로는 커널 2.4.18버전이 탑재된 리눅스를 이용하고 있다. 6talk는 커널레벨에서 동작하는 모듈로써 커널 2.4.x에서 지원하는 Netfilter Framework 를 기반으로 동작한다.

본 논문의 구성은 다음과 같다. 2장에서 몇몇 전이 기술에 대한 요약을, 3장에서는 구현에 대한 기술과 4장에서 그에 대한 테스트결과를 보여주고, 마지막 5장에서 결론과 향후 계획에 대해 기술한다.

1. 관련 연구

NGTrans Working Group에서 다루는 전이기술은 크게 다음 3가지 종류로 나누어 볼 수 있다.

가) 터널링 기술(Tunneling Mechanism)

나) 변환기 기술(Translator Mechanism)

다) i, ii 를 응용한 복합 전이 기술

가) 는 흔히 우리가 알고 있는 터널링 기법[1] 으로써 일반적으로 사용하고 있는 설정 터널(Configured Tunnel)과 IPv6 주소체계에서 지원하는 자동터널링(Automatic Tunnel)으로 나눌 수 있다. 자동 터널링기법은 터널 종단점에 해당하는 IPv4 주소를 포함하고 있는 IPv6 주소를 사용함으로써 별도의 터널링 종단점에 대한 정보를 설정하지 않아도 자동으로 터널링이 이루어지는 방법이다.

나)는 좀더 능동적인 변환기술로써 IPv6 망과 IPv4 망 사이에 IP 패킷의 헤더를 변환할 수 있는 일종의 라우터를 둠으로써 기존의 IPv4 망과 연동하는 방법이다. 대표적인 예로써 NAT-PT[2] 변환기술을 들 수 있는데, 이 방법은 IPv6주소를 IPv4 주소로 바꾸기 위한 IPv4 주소 풀을 관리하는 방법으로 기존의 NAT[4]에서 사용하는 방법을 사용한다. 그리고 프로토콜 변환 (PT: Protocol Translation)을 위해서는 IPv6와 IPv4 패킷 헤더의 상호 해당하는 필드를 맞추어 줌으로써 서로 변환하는 방법을 사용한다. 이 방법에 대한 자세한 명세는 SIIT (Stateless IP/ICMP Translation Algorithm)[3]에 상세히 기술되어 있다. NAT-PT는 NAT를 기반으로 하고 있는 기술이기 때문에 NAT의 특성과 한계점을 그대로 가지고 있다. 특히 메시지의 데이터 영역에 IP주소를 포함하고 있는 응용레벨의 프로토콜은 변환 시에 문제가 발생하게 된다. 이러한 프로토콜의 대표적인 예로써 FTP와 DNS을 들 수 있다. 이런 문제를 해결하기 위하여 특정 응용 프로토콜 별로 응용 프로토콜의 데이터 영역에 포함된 IP주소를 변환해 주는 모듈이 필요하게 된다. 이것을 ALG (Application Level protocol Gateway)라고 한다. 6talk에서는 현재 DNS ALG를 구현하고 있다.

다)는 최근에 주로 연구되고 있는 기술로써 기존에 나와있던 변환기술을 한 개 이상 응용한 기술이다. 이 기술의 예로써 최근 활발히 연구되고 있는 기술은 DSTM[5], 6to4[6], BIS[7] 등이 있다.

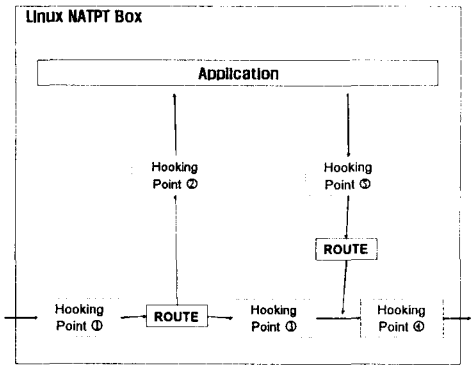
Ⅱ. 구 현

6talk는 앞서 언급한 변환 기술 중 변환기 기술과 터널링 기술을 모두 포함하고 있다. 이 중, 변환기 기술은 NAT(PT)-PT를 구현하고 있으며 현재 DSTM을 개발 마무리하고 있는 중이다.

1. 리눅스 Netfilter Framework

리눅스 커널 2.4.x에는 이전의 패킷 필터링 관련 모듈들이 새롭게 구현된 Netfilter[8]라는 프레임웍으로 통합되었다. Netfilter는 커널 내에 5개의 패킷 훅킹 포인트를 두어서 응용에 맞게끔 필요한 처리 모듈을 각 포인트에 등록하여 원하는 처리를 할 수 있도록 디자

인되어 있다. <그림 1>은 이러한 훅킹 포인트를 보여주고 있다. 각 훅킹 포인트에 등록되는 모듈들은 우선순위를 부여할 수 있으며, 이 우선순위에 따라 같은 훅킹포인트에 등록된 모듈들의 수행순서가 결정된다.



<그림 1> 커널 hooking points

2. 커널 iptable

각 훅킹 포인트에 등록된 모듈에서의 패킷 처리 여부는 커널 내 iptable 이란 자료구조에 저장된 패킷 매칭 룰에 의해서 판단된다. iptable의 각 엔트리는 패킷의 선택 조건을 명시하는 match라는 부분과 match에 의해 선택된 패킷에 대한 처리를 명시한 target으로 나누어져 있다. 특정 훅킹 포인트에 등록된 모듈에 도착한 패킷은 먼저 iptable을 검색하여 패킷을 만족하는 조건을 가진 엔트리가 있는지 확인하고, 그러한 엔트리가 있으면 그 엔트리가 가지고 있는 target에 따라 패킷의 처리를 수행한다. 이 iptable은 Netfilter 응용에 따라 한 개씩 유지하고 있다.

3. 커널 connection tracking 자료구조

Netfilter 응용의 종류에 따라서 각 connection 에 대한 정보를 유지해야 하는 경우가 있다. 그 대표적인 경우가 NAT인데 NAT는 주소 풀을 관리하므로 각 connection을 추적하여 connection이 더 이상 유효하지 않게 되면 사용하고 있던 주소를 회수하게 된다. 이외에 packet filtering이나 mangling은 connection 정보가 필요치 않다.

4. iptables 사용자 인터페이스

Netfilter 는 iptable에 대한 사용자의 룰을 입력받기 위해 iptables라는 사용자 인터페이스를 제공한다. iptables를 이용하여 사용자는 커널내의 테이블을 선택하고, 선택한 테이블에 사용자가 명시한 룰을 삽입할 수 있다. iptables 사용자 인터페이스의 문법은 다음과 같다.

iptables [테이블 명시] [패킷 선택조건 명시] [타겟 정보명시]

5. Netfilter의 한 응용관점에서의 NAT-PT

현재 6talk에서 지원하고 있는 변환기 기술인 NAT-PT는 Netfilter의 한 모듈로써 구현되어졌다. 다만 현재 버전의 Netfilter는 filtering에 관한 처리모듈만 구현되어 있는 상태이므로 여러 추가 구현이 필요하다. 즉 connection 관리에 관한 부분과 NAT-PT처리에 관한 부분이 주 구현 대상이 된다.

6. NAT-PT 훅킹 포인트

NAT-PT는 PRE-ROUTING 훅킹 포인트를 사용한다. NAT-PT는 IPv4/IPv6 두 개의 IP스택을 모두 사용하지만 실제 패킷의 라우팅은 변환된 후의 IP스택에 의존하므로 PRE-ROUTING 훅킹 포인트를 사용하여 connection tracking정보와 헤더변환을 하고 바로 다음 IP 스택으로 넘긴다.

7. 두 개의 connection tracking 자료구조

기존의 NAT는 어느 쪽이든지 하나의 IP버전의 connection tracking정보만 유지하면 되었지만 NAT-PT는 양쪽 IP스택 모두의 connection정보를 유지해야 한다. 더욱이 이 두 개의 connection tracking 정보가 같이 생성되고 삭제되어야 하기 때문에 동기화 문제가 발생하게 된다. 이러한 문제는 양쪽 connection tracking구조가 공유하는 자료구조와 이 자료구조에 양쪽 connection tracking에 대한 포인터를 두어 해결하였다.

8. Fragmentation 문제

NAT-PT는 IPv6/IPv4 양쪽 도메인에 모두 관여를 하므로 fragmentation문제가 발생할 수 있다. 즉, IPv6/IPv4 양쪽의 MTU값이 다를 수가 있기 때문에 fragment된 패킷이 들어오면 그 패킷을 그대로 전송할 수 없다. 이 문제는 fragment된 패킷이 들어오면 일단 조합을 한 후 변환처리를 하고 다른 IP 도메인으로 전송할 때 그 도메인에 해당하는 MTU크기에 맞추어 fragmentation을 수행하였다.

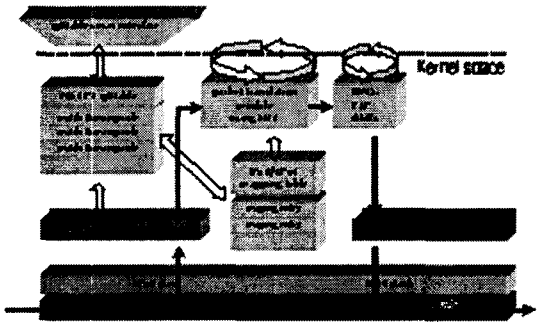
9. DNS ALG

DNS는 응용 레벨 프로토콜로서 데이터 영역에 IP 주소를 포함하고 있다. 따라서 질의 응답에 포함되어 있는 IPv4 주소를 IPv6 주소로 바꾸어주는 역할을 수행하는 ALG가 필요하다.

10. ip6tables 사용자 인터페이스의 확장

Netfilter의 사용자 인터페이스인 iptables 명령은 IPv4 용과 IPv6 용으로 나누어져 있다. 이중 IPv6에 해당하는 것이 ip6tables이다. iptables 는 확장을 위하여 특정 타겟에 대한 처리 모듈을 라이브러리 형식으로 추가할 수 있도록 하고 있다. 따라서 NAT-PT에 대한 모듈도 라이브러리로 추가하였다.

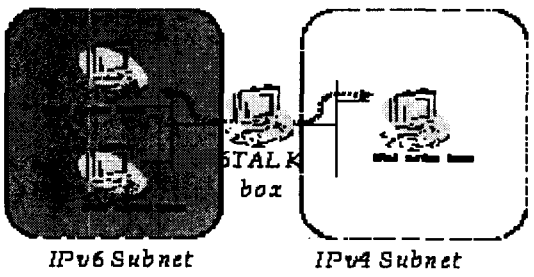
이상과 같은 NAT-PT의 구현은 <그림 2>와 같은 구조로 요약될 수 있다.



<그림 2> NAT-PT 의 커널 내 구성도

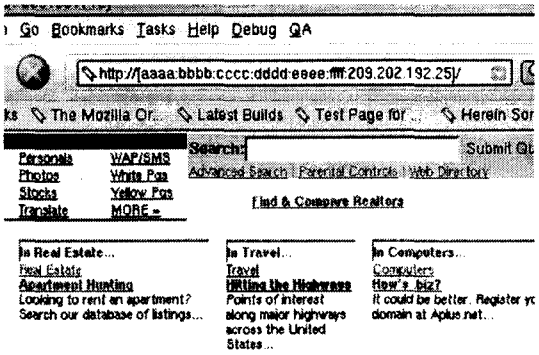
III. 실 험

본 구현을 테스트하기 위하여 다음과 같은 실험을 하였다. 몇몇 IPv6 전용 응용을 이용해 아무런 설정도 하지 않은 IPv6 리눅스 호스트에서 외부 IPv4 호스트와 통신을 하였다. 테스트 토폴로지는 <그림3>과 같다.

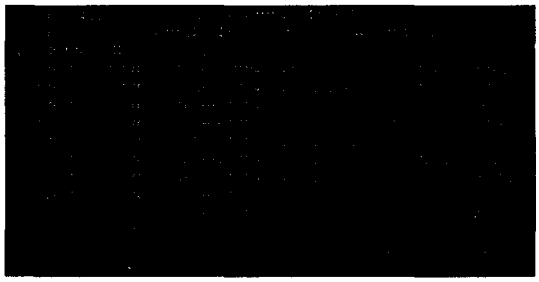


<그림 3> 실험 토폴로지

응용 프로그램으로는 telnet for IPv6, ping for IPv6, Mozilla web browser 등을 사용하였다. <그림 4> <그림 5>는 이러한 실험 결과를 캡쳐한 결과이다.



<그림 4> mozilla 를 이용하여 IPv4 사이트에 접속



<그림 5> IPv4 호스트에 대한 ping6 시도

IV. 결론 및 향후 계획

6talk 프로젝트의 최종 목표는 Smart Transition Box 이다. 즉, 네트워크의 구성상황에 따라 적절한 변환메커니즘을 적용하여 IPv6 망과 IPv4 망의 자연스러운 연동을 가능케 하는 edge device이다. 이를 위하여 오픈 플랫폼인 리눅스의 Netfilter구조를 사용하였으며 6talk 가 지원예정인 변환기술 중 NAT-PT/SIIT와 DNS-ALG를 구현하였다.

참고문헌

[1] R. Gilligan, "Transition Mechanisms for IPv6 Hosts and Routers", RFC-2893, Aug 2000.

[2] G. Tsirtsis, "Network Address Translation - Protocol Translation (NAT-PT)", RFC-2766, Feb 2000.

[3] E. Nordmark, "Stateless IP-ICMP Translation Algorithm (SIIT)" RFC-2765, Feb 2000.

[4] P. Srisuresh, "Traditional IP Network Address Translator (Traditional NAT)", RFC-3022, Jan 2001.

[5] Jim Bound, "Dual Stack Transition Mechanism

(DSTM)", Internet Draft, July 2001.
[6] B. Carpenter, "Connection of IPv6 Domains via IPv4 Clouds", RFC-3056, Feb 2001.
[7] K. Tsuchiya, "Dual Stack Hosts using the Bump-In-the-Stack Technique" (BIS), RFC-2767, Feb 2001.
[8] Rusty Russell, "Linux Netfilter Hacking HOWTO", Linux-HOWTO Doc, Jul 2000.