IPv6 지원 FTP-ALG 설계 및 구현

홍용근* · 이주철* · 신명기* · 김형준*

*한국전자통신연구원

Design and Implementation of IPv6-support FTP-ALG

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

*Electronics and Telecommunications Research Institute

E-mail : yghong@etri.re.kr

요 약

인터넷의 주소 문제를 해결하고자 등장한 차세대 인터넷 프로토콜인 IPv6는 현재 많은 곳에서 실험적 또는 상업적으로 사용되고 있다. 성공적으로 IPv6로 전환하기 위해서 무엇보다도 필요한 것이 기존에 동작하고 있는 IPv4 호스트 및 라우터와의 호환성과 IPv4에서 사용되고 있는 많은 응용프로그램들과의 호환성이다.

IPv6 전환 메커니즘 중에 하나인 NAT-PT (Network Address Translation-Protocol Translation)는 IPv4 주소와 IPv6 주소를 자연스럽게 변환하여 IPv4 호스트와 IPv6 호스트간에 통신을 가능하게 한다. 그러나 DNS와 FTP같이 인터넷 주소가 패킷의 데이터 영역에 포함이 된 경우에는 별도로 처리해주는 메커니즘이 필요하다.

본 논문에서는 NAT-PT에서 IPv4 호스트와 IPv6 호스트 간에 FTP 서비스를 지원해 주기 위해서 필요한 FTP-ALG (Application Layer Gateway)를 설계하고 구현하는 방법에 관하여 기술한다.

ABSTRACT

The next generation Internet Protocol IPv6 that is appeared to solve the exhausting problem is now widely deployed in a testbed or commercial site. To successfully deploy IPv6, interoperation with exist IPv4 hosts and routers and interactions with many IPv4 applications are more important. The NAT-PT (Network Address Translation-Protocol Translation) among IPv6 transition mechanism enables the communication between IPv4 and IPv6 host with translating IPv4 address and IPv6 address. But for DNS and FTP, another specific mechanism is needed when internet address is included in packet payload area. This paper describes the design and implementation of IPv6-support FTP-ALG (Application Layer Gateway) to enable FTP service between IPv4 node and IPv6 node.

키워드

IPv6, FTP-ALG, NAT-PT, Netfilter

Ⅰ. 서 론

IPv4 기반의 인터넷에 IPv6가 소개된 이후 IPv4 인터넷 세계에서 IPv6 라우팅이 가능하게 하기 위해 IPv6 라우팅, IPv6 주소, DNS 등 여러 이슈들이 논의되고 있다. 성공적으로 IPv6로 전환하기 위해서는 무엇보다 필요한 것이 IPv4 호스트와 라우터와의 호환성과 기존 IPv4에서 IPv6로의 자연스러운 이전을 지원해 주는 IPv6 전환 메커니즘이다. IPv6 전환 메커니즘 중의 하나인 NAT-PT는 SIIT 프로토콜 변환과 NAT 동적 주소 변환을 조합하여 IPv6 전용 노드와 IPv4 전용 노드 사이에서 상호 통신을 가능하도록 한다. 그러나 DNS와 FTP 같이 인터넷 주소가 패킷의 데이터 영역에 포함된 경우에는 별도로 처리해 주는 메커니즘이 필요하다. IPv4 노드와 IPv6 노드간에 FTP 서비스가 가능하게 하기 위해서 필요한 것은 IPv4에서 사용하는 PORT, PASV 명령어(command)와 IPv6에서 사용하는 EPRT, EPSV 명령어간의 변환과 패킷의 데이터 영역에 포함된 IPv4 주소와 IPv6 주소간의 변환이다.

이러한 명령어 변환과 주소간의 변환을 가능하게 하기 위해 Linux에서 제공되는 Netfilter framework를 사용하여 FTP-ALG를 설계하고 구현하였다.

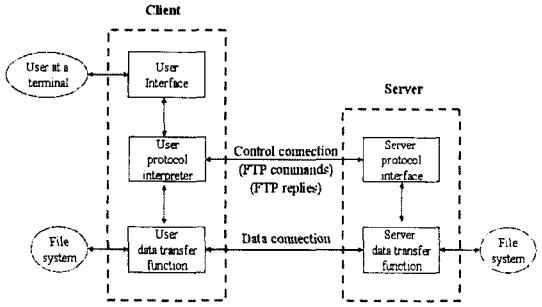
본 논문에서는 이러한 FTP-ALG를 어떻게 디자인하고 구현하였는지에 대하여 알아본다. 제 2장에서는 본 논문에서 제공하려는 FTP 서비스에 대하여 알아보고, 제 3장에서는 FTP-ALG와 함께 동작하는 NAT-PT 메커니즘에 대하여 알아본다. 제 4장에서는 Linux Netfilter framework를 이용하여 FTP-ALG를 설계한 방법에 대하여, 제 5장에서는 실제로 Linux에서 FTP-ALG를 구현한 것에 대하여 설명한다. 제 6장에서는 요약 및 결론을 맺는다.

II. FTP 서비스

FTP (File Transfer Protocol)는 TCP/IP 데이터 연결을 이용하여 데이터 전송을 가능하게 하는 프로토콜이며, 이때 FTP는 32 bits의 IPv4 주소를 사용하는 것을 가정한다[1]. 그러나 IPv6가 도입됨에 따라 더 이상의 32 bits의 인터넷 주소는 무의미하게 되었다. 즉 32 bits의 IPv4 주소와 128 bits의 IPv6 주소를 둘 다 지원하여야 한다. 요즘 나오는 FTP 서버와 클라이언트 중에는 IPv4 뿐만 아니라 IPv6 서비스도 지원하는 것들도 있다.

그림 1은 클라이언트와 서버간에 FTP 서비스를 보여 주고 있다.

그림 1. FTP 서비스 동작

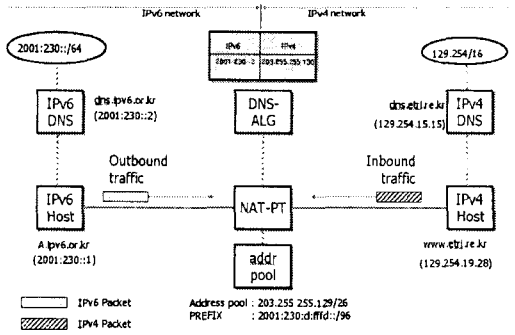


FTP 서비스는 control connection과 data connection을 통하여 이루어진다. Control connection은 일반적으로 서버-클라이언트 모델 형태로 이루어지며, 클라이언트가 서버의 포트번호 21번으로 접속을 요구하여 연결이 성립된다. Control connection은 FTP 서비스 전체 시간동안 계속 유지되며, 클라이언트가 서버에게 요청하는 명령어나 서버의 response용으로 사용된다. Data connection은 실제로 데이터가 전송되는 시점에 생성되었다가 데이터 전송이 끝나면 소멸된다.

III. NAT-PT

NAT-PT는 IPv6-IPv4 경계에서 SIIT 프로토콜 변환과 NAT 및 DNS ALG 등 적절한 ALG의 동적 주소 변환을 조합하고, IPv4 주소 풀을 사용하여 동적으로 IPv4 주소를 할당, IPv6 주소와 IPv4 주소를 서로 바인딩하여 IPv6 전용 노드와 IPv4 전용 노드 사이에서 상호 통신을 가능하도록 한다. 여기에서 각 네트워크 상의 단말 노드들은 변경할 필요가 없으며, IP 패킷 라우팅은 단말 노드에 있어서 완전히 투명하게 된다. 그러나 한 세션에서 들어오고 나가는 패킷은 동일한 NAT-PT 라우터를 거쳐야만 한다. NAT-PT 토폴로지 제약은 NAT와 동일하다[2]. 그림 2는 NAT-PT 동작 예를 보여 준다.

그림 2. NAT-PT 동작 예



IV. FTP-ALG 설계

FTP 는 패킷 데이터 영역에 IP 주소와 TCP 포트 번호를 실기 때문에 IPv4 노드와 IPv6 노드간에 FTP 서비스를 제공하기 위해서는 패킷 데이터 영역 안에 있는 IP 주소와 필요하다면 포트 번호를 변경하여야 한다. 또 IPv4 FTP에서는 PORT, PASV 명령어를 사용하지만, IPv6 FTP에서는 EPRT, EPSV 명령어를 사용하는 등 명령어에서도 IPv4와 IPv6간에 변경하는 일이 필요하다. 한편 패킷 데이터 영역에 포함된 명령어나 IP 주소, 포트 번호를 변경하다 보면 IP 패킷의 길이가 변하기 때문에, FTP-ALG는 이러한 패킷 길이 변화까지 책임지고 다루어야 한다.

1. IPv4 노드에서 IPv6 노드 방향

변경하고자 하는 패킷의 payload 부분이 PORT나 PASV 명령을 포함하고 있으면 이를 각각 EPRT, EPSV 명령으로 형식에 맞게 번역한다. 그리고 IPv4 형식의 source 주소와 destination 주소를 IPv6 형식의 주소로 변경한다. 이 변환 과정에서 payload 길이가 변한다면 IP 헤더의 payload length 부분과 TCP 헤더의 sequence number 값을 갱신한다.

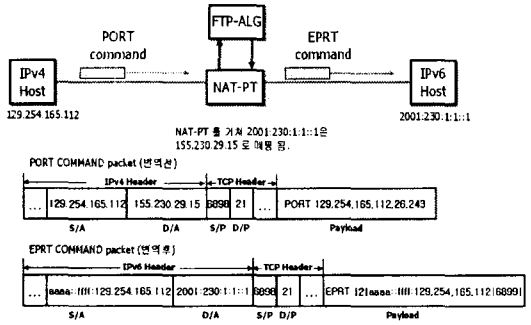


그림 3은 이러한 변환 과정을 나타낸 것이다.

그림 3. IPv4 노드에서 IPv6 노드로의 FTP

2. IPv6 노드에서 IPv4 노드 방향

번역하고자 하는 패킷의 payload 부분이 EPRT나 EPSV 명령을 포함하고 있으면 이를 각각 PORT, PASV 명령으로 형식에 맞게 번역한다. 그리고 IPv6 형식의 source 주소와 destination 주소를 IPv4 형식의 주소로 변경한다. 이 변환 과정에서 payload 길이가 변한다면 IP 헤더의 payload length 부분과 TCP 헤더의 sequence number 값을 갱신한다.

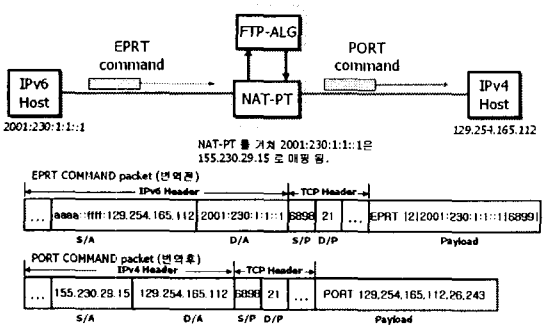


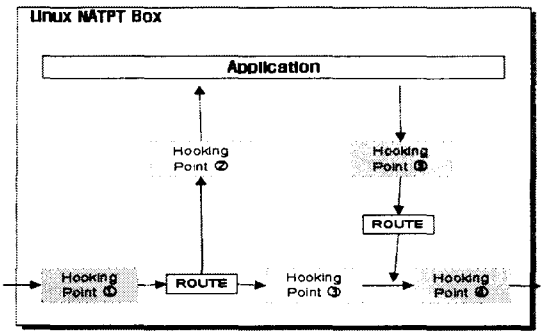
그림 4는 이러한 변환 과정을 나타낸 것이다.

그림 4. IPv6 노드에서 IPv4 노드로의 FTP

V. FTP-ALG 구현

1. Linux Netfilter Framework

Linux 커널 2.4.x에는 이전의 패킷 필터링 관련 모듈들이 새롭게 구현된 Netfilter 라는 framework으로 통합되었다. Netfilter는 커널 내에 5개의 훅킹 포인터를 두어서 응용에 맞게끔 필요한 처리 모듈을 각 포인터에 등록하여 원하는 처리를 할 수 있도록 디자인되었다. 그림 5는 이러한 훅킹 포인트들을 보여 주고 있다. 각 훅킹 포인터에 등록된 모듈들은 우선 순위를 부여할 수 있으며, 이 우선 순위에 따라 같은 훅킹 포인터에 등록된 모듈들의 우선 순위가 결정된다. 처리 모듈에서는 패킷을 처리할 때 대상이 되는 패킷을 선별



하는 조건을 담고 있는 테이블(ip_table)을 참조하여 패킷을 처리한다[3,4].

그림 5. Netfilter 훅킹 포인터

2. Connection 관리

Netfilter에서 제공하는 Packet-Filtering이나 Mangling과는 달리 FTP-ALG를 포함한 NAT-PT 류의 Netfilter 응용에서는 변환시 이용되는 주소 풀이 중요한 자원이므로, 가능한 신속히 주소를 재활용할 수 있도록 Netfilter box를 통과하는 connection들에 대한 관리 모듈이 필요하다. Netfilter에서는 connection tracking 메커니즘이 이러한 역할을 담당한다.

Connection 관리를 위해서 FTP-ALG는 control connection과 data connection 등 복수 개의 네트워크 connection을 사용할 수 있도록 보장하여야 하고, 데이터 스트림에서 관련된 IP 주소를 파악하여 이전 connection과 관련이 있는 "child" connection인지 확인하는 절차가 필요하다.

FTP-ALG는 다음과 같은 역할을 수행한다.

- 네트워크 인터페이스로 들어오는 패킷 중에 어떤 패킷을 관심을 가지고 처리하는지 Netfilter에게 알려 준다. (포트 번호 21번으로 들어오는 패킷은 FTP-ALG가 처리하도록 알려 줌)

- 특정 포트 번호(21번)로 들어오는 패킷을 처리하는 함수를 Netfilter에 등록한다.

- Control connection 연결 후에 생성되는 data connection을 같이 처리 할 수 있도록 data connection을 기다리도록 한다.

포트 번호 21번으로 들어오는 패킷에 대해서 FTP-ALG가 connection을 관리 할 수 있도록 Netfilter에서 시스템이 처음 초기화 할 때 이를 등록한다.

v4 인터페이스로 들어오는 경우 :
ip_conntrack_helper_register()
v6 인터페이스로 들어오는 경우 :
ip6_conntrack_helper_register()

Control connection 연결 후 생성되는 data connection 연결을 위해서, FTP-ALG는 control connection 연결 시 관심을 가지는 연결의 특성을 기록한 후 data connection 연결 시 이를 이용하도록 한다.

v4 인터페이스로 들어오는 경우 :
ip_conntrack_expect_related()
v6 인터페이스로 들어오는 경우 :
ip6_conntrack_expect_related()

3. 패킷 변환

Connection tracking이 생성되면 ip_table에서 패킷에 매치 되는 룰을 검색하여 변환 정보를 생성한 후 connection tracking에 저장한다. 이러한 과정이 끝나면 이 정보를 바탕으로 패킷을 변환한다. 패킷 변환은 먼저 IP 헤더의 변환을 시도하고, 그 다음 상위 계층 프로토콜에 따른 변환을 한다. FTP-ALG이 해당하는 부분이 바로 상위 계층 프로토콜(TCP, FTP)에 따른 변환을 하는 부분이다.

패킷 변환을 위해서 FTP-ALG가 하는 역할은 다음과 같다.

- Control connection과 data connection을 통하여 지나가는 패킷을 실시간으로 가로채서, FTP 명령어, IP 주소, 포트번호, payload 길이 부분, sequence number 값을 변경한다.

- Control connection과 data connection간의 상호 의존성을 확인하여 처리한다.

포트 번호 21번으로 들어오는 패킷에 대해서 FTP-ALG가 패킷을 변환 할 수 있도록 Netfilter에서 시스템이 처음 초기화 할 때 이를 등록한다.

v4 인터페이스로 들어오는 경우 :
ip_natpt_helper_register()
v6 인터페이스로 들어오는 경우 :
ip6_natpt_helper_register()

Control connection과 data connection간의 상호 의존성을 확인하여 처리하기 위해서 관심을 가지는 연결의 특성을 기록한 후 data connection 연결 시 이를 이용하도록 한다.

v4 인터페이스로 들어오는 경우 :
ftp_natpt_expect_related()
v6 인터페이스로 들어오는 경우 :
ftp_natpt_expect_related6()

VI. 결 론

NAT-PT는 IPv4 노드와 IPv6 노드의 변경 없이 IPv4 노드와 IPv6 노드간에 상호 통신을 가능하도록 하는 IPv6 전환 메커니즘 중의 하나이다. FTP 서비스는 IP 주소 및 FTP 명령어가 패킷 데이터 영역에 포함되기 때문에, FTP-ALG는 NAT-PT에서 FTP 서비스를 가능하게 하기 위해 별도의 변환 메커니즘을 제공한다. FTP-ALG는 IPv4 FTP 명령어와 IPv6 FTP 명령어를 서로 변경하여 주고, IP 주소와 포트 번호를 변경한 후, 패킷 길이 변화 및 sequence number 변화까지 책임지고 관리한다.

본 논문에서는 Linux Netfilter framework를 이용하여 IPv6를 지원하는 FTP-ALG를 설계하고 구현한 것을 설명하였다.

참고문헌

[1] RFC959 : FILE TRANSFER PROTOCOL (FTP)
[2] 차세대 인터넷 프로토콜:IPv6, IPv6 포럼코리아, 2002.3
[3] Rusty Russell, Linux Netfilter Hacking HOWTO , Linux-HOWTO Doc, Jul 2000
[4] 이주철, 이숙영, 신명기, 김용진, "6TALK: IPv6 망과 IPv4 인터넷망, 고립된 망 IPv6 망과 IPv6 망 사이의 연동 기술", 제 4회 차세대통신학회 논문집, 2001.