IPv4와 IPv6의 연동과 호환을 위한 NAT-PT에 관한 연구

이승민[1], 민상원[1], 김용진[2], 박수홍[3]
1. 광 운 대 학 교 전 자 통 신 공 학 과
2. 한국전자통신연구원 선행표준연구팀
3. 오 피 콤 연구소
smlee@explore.kwangwoon.ac.kr

NAT-PT Operation for IPv4 and IPv6 Nodes to Coexist

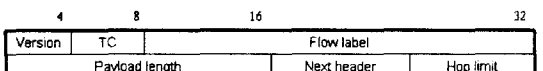
Seung-Min Lee[1], Sang-Won Min[1], Yong-Jin Kim[2], Soo-Hong Park[3]
1. Dept. of Telecommunications Engineering, Kwangwoon University
2. Standards Systems Research Teams, ETRI
3. OPICOM R&D Center

요 약

IPv4(Internet Ptorocol version 4)의 차세대 버전인 IPv6는 기존 IPv4의 문제점인 주소부족 및 새로운 부가기능 등의 필요성 때문에 IETF(Internet Engineering Task Force) 에서 IPv4를 대체하기 위해 개발한 프로토콜이다. 하지만 IPv4를 어느 한순간에 IPv6로 대체하는 것은 불가능하기 때문에 기존 IPv4와의 호환 및 연동을 위한 터널링 및 기타 여러 가지 transition mechanism들이 고려되고 있다. 이러한 mechanism들 중에서 최근 표준으로 발표된 NAT-PT(Network Address Translation □ Protocol Translation)는 DNS-ALG(Domain Name System □ Application Level Gateway)라는 개념을 도입하여 IPv4와 IPv6간의 양방향 주소변환 및 프로토콜 변환기능을 수행한다. 각각의 기능에서 주소 변환은 주소크기와 형식이 다른 IPv4와 Pv6간의 서로 상대 노드를 엑세스할 수 있도록 하고 프로토콜 변환은 다른 버전의 프로토콜을 사용하는 노드들이 상대노드가 보낸 패킷의 헤더정보를 이해할 수 있도록 한다. 그리고 이러한 기본적인 변환외에 헤더정보의 정확한 전달을 위해서는 기존의 path MTU(Maximum Transgfer Unit) discovery 그리고 pseudo header checksum 등도 고려되어야 한다.

1. 서론

현재 세계적으로 널리 쓰이고 있는 인터넷의 가장 기본적인 통신프로토콜인 IPv4는 기하급수적인 인터넷 성장에 따라 IP주소 고갈문제가 발생하고 새로운 부가기능들을 필요로 하게 되었다[1]. 이러한 여러가지 문제점 때문에 IETF에서는 기존의 IPv4를 대체하기 위해 IPv6[2]를 제안하였는데 프로토콜 설계는 기존의 IPv4의 기능들을 대부분 수용하면서 성능향상을 위해 몇 가지 변경사항들을 가지도록 의도되었다. 이런 변경사항들에 대해 간단히 설명하면 우선 그림 1에 표현된 40바이트의 IPv6 기본헤더 형식에서 주소길이가 기존의 32비트에서 128비트로 확장되어 IPv4의 주소고갈문제를 해결하였다.



TC : Traffic Class

그림 1. IPv6 기본헤더

그리고 IPv6는 기존 IPv4의 protocol 필드와 기본헤더의 fragment 기능을 담당하는 필드들을 next header에서 담당하도록 하여 융통성 있는 헤더형식을 가진다.

또 기존의 IPv4에서 수용하지 못했던 부가적인 기능들을 IPv6에서는 대부분 포함하고 있으며 traffic class 필드와 flow label 필드기능을 이용하여 현재 이슈가 되고 있는 QoS(Quality of Service) 구현이 IPv4에 비해 용이하도록 하였다. 그리고 IPv6에서 제공하고 있는 authentication 헤더는 수신측에서 전송된 패킷의 변형과 진위여부 확인을 통해 향상된 보안기능을 제공한다[2]. 하지만 IPv6가 비록 이전버전보다 향상된 기능들을 제공하고 있음에도 불구하고 현재 주소크기 문제와 기타 기능들의 호환성문제로 인해 광범위한 사용 및 완전한 도입 이전까지는 장시간 IPv4와의 혼용이 예상되고 있다. 따라서 현재 쓰이는 터널링(Tunneling)기법 외에 IPv4-to-IPv6의 transition과 coexistence mechanism 등이 필요하다[3]. 본 논문에서는 이러한 변환 기법중의 하나로서 최근 IETF에서 발표된 NAT-PT 표준에 대한 소개와 실제 변환에 필요한 기능 분석을 위해 2절에서 주소변환방법에 대해 설명[4]하고 3절에서는 프로토콜 변환 방법을 설명[5]한 후 4절 결론에서 이러한 변환방법들을 이용한 NAT-PT의 구현에 대해 분석하였다.

2. Network Address Translation (NAT)

IPv4와 IPv6간의 주소변환에서 고려되어야 할 점은 IP의 버전에 따른 다른 주소크기와 형식이다. 따라서 다른 버전을 사용하는 노드들은 상대 노드들을 엑세스할 수 있도록 dual stack 형식으로 다른 버전의 형식에 대한 호환성을 가지고 있거나 그림 2처럼 다른 버전의 주소형식을 자신의 IP 버전에 맞는 주소형식으로 변환하여 상대노드를 인식해야 한다[4].

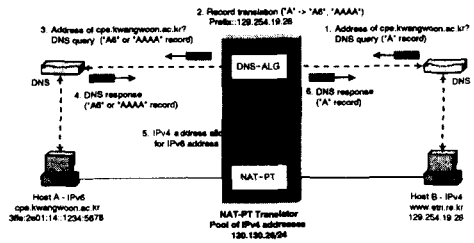


그림 2. DNS-ALG를 이용한 IPv6 주소의 변환

이 때 문제점은 전통적인 IPv4 NAT의 단방향 주소변환 문제점과 마찬가지로 IPv4 에서 IPv6로의 inbound traffic 에 대한 주소변환이다. 이것은 32비트의 IPv4노드가 128비트의 IPv6주소형식을 표현할 수가 없기 때문이다. 이러한 단방향 NAT의 단점을 보완하기 위해 그림 3의 구조를 가질 수 있는 변환기(Translator)에 포함된 DNS-ALG의 기능을 이용한다. 즉, DNS 메시지가 IP 주소정보를 이용하므로 DNS 메시지의 변환이 필요하게 되고 이 기능을 DNS-ALG가 담당하도록 함으로써 그림 2처럼 애플리케이션 레벨의 DNS를 이용한 실제 상대노드의 주소획득 방법을 제공한다.

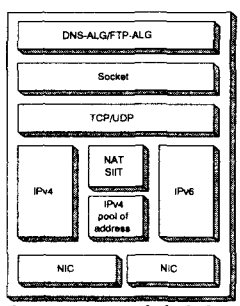


그림 3. NAT-PT 변환기의 구조

그림 2에서 IPv4 호스트 B는 IPv6 호스트 A에 대한 주소정보를 얻기 위해 IPv4 네트워크를 담당하는 DNS 서버가 DNS-ALG에 cpe.kwangwoon.ac.kr 호스트의 이름에 대한 IP 주소를 요청하는 DNS query 메시지를 보내게 된다. 이 메시지를 받은 DNS-ALG는 표 1처럼 이름과 IPv4 32비트 주소와의 매핑정보를 가지고 있는 " A" 레코드를 IPv6에서 운용될 수 있도록 하기 위해 " AAAA" 또는 " A6" 레코드로 변환한다.

표 1. DNS records

	IPv4	IPv6
Name to address	A	AAAA, A6
Address to name	IN-ADDR.ARPA	IP6.INT

DNS-ALG에서 변환된 DNS query는 최종적으로 IPv6 네트워크의 DNS 서버에 전달된 뒤 처리되어 호스트 A의 주소를 포함하는 DNS response 메시지로 되돌려진다.
DNS response 메시지가 DNS-ALG에 도착하면 DNS-ALG는 이 메시지가 IPv4 네트워크에서 운영될 수 있도록 " AAAA" 또는 " A6" 레코드를 " A" 레코드로 다시 변환한다. 이 때 IPv4 주소형식 자체가 IPv6 주소를 나타낼 수 있는 방법이 없기 때문에 IPv6 주소를 사용하는 노드를 IPv4 노드가 인식할 수 있도록 사전에 미리 할당된 pool of IPv4 주소들 중 하나의 주소를 IPv6 주소에 할당한다. 따라서 DNS-ALG는 DNS response 메시지에 포함된 호스트 A의 IPv6 주소를 pool of IPv4 주소로부터 할당받은 임의의 IPv4주소로 변환하여 실제 IPv4네트워크로 내보내게 된다.

이러한 과정을 통해 호스트 B는 실제로는 호스트 A를 나타내는 변환기의 IPv4 주소를 이용하여 호스트 A에게 패킷을 보낼 수 있다.

IPv6에서 IPv4로의 outbound traffic에 대한 패킷의 주소변환은 inbound traffic 의 주소변환방법에 비해 간단하다. 즉, IPv6 호스트는 IPv6 네트워크에서 운용될 수 있도록 네트워크에서 운용되는 prefix를 name lookup 을 통해 얻어진 호스트B의 IPv4 주소에 붙여서 실제 목적지 주소로 사용한다. 패킷 전송시에 호스트 A는 IPv4 네트워크에서 운용될 수 있도록 자신의 주소로서 NAT-PT의 pool of IPv4 address 로부터 하나의 주소를 임시로 할당받아 사용한다.

3. Protocol Translation (PT)

IPv4와 IPv6는 버전별 기능의 차이점과 각 기능을 담당하는 헤더의 형식이 다르기 때문에 노드들이 서로 다른 버전의 IP를 사용하는 경우 상대 노드의 프로토콜을 이해할 수 있도록 주소변환과 마찬가지로 변환기가 이러한 패킷들의 헤더를 변환해주어야 한다[5]. 헤더변환이 필요한 프로토콜은 크게 IP와 ICMP 이렇게 둘로 나뉘어질 수 있는데 이러한 헤더 변환방법은 주소부분을 제외하고는 별도의 변환표준인 SIIT(Stateless IP/ICMP Translation)표준과 동일하다. SIIT의 " Stateless" 의 의미는 변환기가 각 패킷들에 대한 state를 유지하지 않기 때문에 변환이 패킷마다 독립적으로 운영된다는 것을 나타낸다. 따라서 별도의 변환기의 기능조정이 필요없고 주어진 TCP 연결은 다른 변환기를 거쳐가는 패킷들의 두 방향에서 이루질 수 있다.

표 2와 3은 일반적인 IPv4와 IPv6 간의 기본헤더변환시 각 필드들의 변환방법을 나타내고 있다. 서로 다른 두 버전의 IP헤더간 중요한 차이점들 중의 하나는 fragment 에 관련된 정보가 IPv4에서는 기본헤더에서 표현되지만 IPv6 에서는 next header 에서 표현된다는 점이다. 그리고 IPv6 의 payload length 필드와 IPv4의 header length 필드 및 total length 필드간의 변환시에는 fragment header 존재 여부를 고려하여 필드값이 재계산되어야 한다.

표 2. IPv4 헤더의 IPv6 헤더로의 변환

Traffic Class	• Copied from IP TOS - if identical • Ignore the IPv4 "TOS", set "0"	자동
Flow label	All bit zero	자동
Payload Length	Total length of IPv4 packet - (IPv4 header + IPv4 options)	(Total length of IPv4 + fragment header) - (IPv4 header + IPv4 options)
Next Header	Copied from IPv4 header	• Fragment header (44) • Fragment header field - Next header : copied from IPv4 - Fragment offset : " - M flag : " Identification : lower-order 16bits copied from IPv4, higher-order bits set to zero
Hop Limit	TTL value - since translator is a router	자동
Source Address	• Low-order 32 bits : IPv4 source address • High-order 96 bits : 모든 IPv4 통신을 위한 PREFIX - PREFIX를 가지는 모든 주소들은 NAT-PT를 경유함	자동
Destination address	• Destination 노드에 대한 IPv4와 IPv6 주소 매핑 유지 • IPv4 destination 주소는 매핑을 통해 IPv6 주소로 변환	자동

표3. IPv6 헤더의 IPv4 헤더로의 변환

Type of Service And Precedence	• Copied from the IPv6 Traffic class • Ignore the IPv6 traffic class • Set the IPv4 "TOS" to "0"	자동
Total Length	Payload length of IPv6 header + size of the IPv4 header	Payload length of IPv6 header + (size of the IPv4 header - fragment header)
Identification	All "0"	Copied from low-order 16bits in the identification field in fragment header
Flag	More fragments flag is set to "0"	More fragments flag is copied from M flag in the Fragment header
Fragment offset	변환기가 라우터인 경우 "0"을 확인하여 ICMPv4 6에시지전송	Copied from IPv6 header
Header Checksum	IPv4헤더 설정시 재계산	자동
Source Address	NAT-PT는 IPv6 source 주소와 pool of IPv4 주소의 가용한 IPv4 주소와의 매핑을 유지하여 IPv6 source 주소를 이 IPv4 주소로 변환	자동
Destination Address	Low-order 32bits가 IPv4 destination 주소로 복사됨	자동

그리고 IPv6 의 기본헤더에는 checksum 필드가 존재하지 않지만 IPv4 기본헤더에서는 존재하므로 IPv6에서 IPv4로의 헤더변환시에는 checksum값 계산이 필요하다.

또 ICMPv6, TCP, UDP 의 checksum 필드는 계산의 범위가 헤더 외에 데이터나 메시지 필드까지 포함하고 있기 때문에 16비트 연산을 위한 padding을 위해 pseudo header를 필요로 한다. 따라서 TCP나 UDP의 pseudo header가 데이터그램 전달의 신뢰성을 위해 IP 주소정보를 이용하므로 주소정보를 이용하는 호스트들을 위해 pseudo header도 재계산되어야 한다.

IPv4와 IPv6의 차이점중의 또 다른 하나는 IPv6에서는 path MTU(Maximum Transfer Unit) 가 필수이지만 IPv4에서는 선택사항이라는 점이다. 따라서 만일 IPv4 노드가 path MTU discovery를 하는 경우[6]에는 path MTU discovery 가 end-to-end로 운영되어야 한다. 하지만 IPv4 의 official minimum MTU가 68바이트인 반면에 IPv6 에서는 1280바이트이기 때문에 IPv4 또는 IPv6 라우터가 " datagram too big" 혹은 " packet too big " 의 ICMP error message를 보낼수 있다.

그림 4는 IPv4 호스트 B가 Don' t framgment(DF) 비트를 세팅한 다음 IPv6 호스트 사이에 존재하는 링크에 대한 path MTU discovery를 수행한 후 패킷을 path MTU 크기로 분할하여 전송하고 fragment 정보를 end-to-end로 운영하는 방법을 설명하고 있다[7].

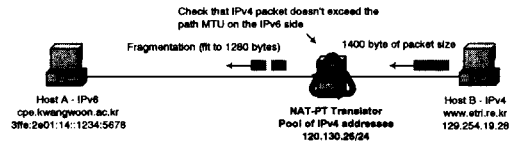


그림 4. IPv4 호스트의 path MTU discovery 수행

만약 IPv4 호스트 B가 1400 바이트의 패킷을 생성하여 IPv6 호스트 A로 전송을 하는 경우 path MTU discovery를 하는 과정에서 ICMP error message를 통해 전체 path MTU가 결정되고 이 크기에 따라 패킷을 분할하여 전송하게 된다. 이 때 NAT-PT는 분할된 패킷의 fragment header 즉, IPv4의 기본헤더의 분할 정보를 IPv6 의 next header 인 fragment header로 전달하는 프로토콜 변환을 담당한다.

그림 5 는 IPv4 호스트B가 DF 비트를 세팅하지 않고 path MTU discovery를 하지 않는 경우의 fragmentation을 설명하고 있다. 비록 IPv4 호스트가 path MTU discovery를 하지 않는다고 해도 IPv6 호스트의 path MTU discovery는 필수사항이기 때문에 NAT-PT는 도착한 IPv4 패킷이 IPv6 호스트측의 path MTU를 초과하는지를 확인하여 만약 초과할 경우 IPv6 측의 path MTU 크기에 맞게 분할한 다음 패킷을 전송하게 된다.

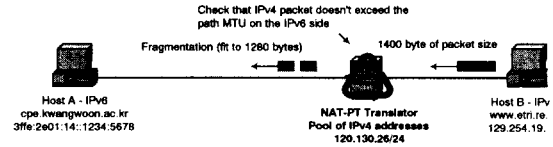


그림5. Path MTU를 수행하지 않는 경우

그리고 호스트들은 path MTU 를 결정하는 과정에서 ICMP 에러 메시지를 참조하여 그 기능을 수행하게 되는데 이러한 ICMP 에러 메시지들도 버전에 따라 그 기능이나 형식이 다르다. 따라서 이것 역시 프로토콜 변환을 통해 다른 버전의 ICMP를 사용하고 있는 호스트들이 이해할 수 있도록 변환되어야 한다. 현재 이러한 ICMP error 메시지들은 전부 변환되지 않고 일부 메시지들만 변환된다.

표4와 5는 ICMPv4 메시지의 ICMPv6 메시지로의 변환방법을 설명하고 있는데 반대방향의 변환 방법도 유사하다.

표4. ICMPv4 query 메시지의 변환

Echo and Echo reply (Type 8 and Type 0)	• Type 8 → 128, 0 → 129 • ICMP checksum 고려, ICMPv6 pseudo-header 포함
Information Request/Reply	• Silently drop
Timestamp and Timestamp reply	• Silently drop
Address Mask Request/Reply	• Silently drop
ICMP Router Advertisement	• Silently drop
ICMP Router Solicitation	• Silently drop
Unknown ICMPv4 types	• Silently drop

표5. ICMPv4 에러 메시지의 변환

Destination Unreachable (Type 3)	Code 0, 1 (net, host unreachable)	Set code to 0 (no route to destination)
	Code 2 (protocol unreachable)	• ICMPv6 parameter problem (Type4, Code1) • Make the pointer point to the IPv6 next header
	Code 3 (port unreachable)	Set code to 4 (port unreachable)
	Code 4 (fragmentation needed, DF bit)	ICMPv6 Packet Too Big message with code 0 IPv4와 IPv6 헤더 차이를 위한 MTU field 필요
	Code 5 (source route failed)	Set code to 0 (no route destination)
	Code 6, 7, 8	Set code 0
	Code 9, 10	Set code 1
	Code 11, 12	Set code 0
Redirect (Type 5)		Single hop message, silently drop
Source Quench (Type 4)		Obsoleted in ICMPv6, silently drop
Time Exceeded (Type 11)		Set the type field to 3 (code field is unchanged)
Parameter Problem (Type 12)		Set the type field to 4 (pointer 필요)

4. 결론

NAT-PT를 구현함에 있어서 기존의 NAT와 관련된 모든 제한들은 NAT-PT에서도 똑같이 적용된다. 우선 세션에 참여하고 있는 모든 요청과 응답이 동일한 NAT-PT 라우터를 거쳐서 전송되어져야 하지만 이러한 망 구성에 대한 제한은 경계 라우터에 dual stack으로 NAT-PT를 구현함으로써 문제 해결이 가능하다. 하지만 프로토콜 변환시 완벽한 의미를 전달할 수 있는 프로토콜 변환에 대한 어려움이 존재하고 보안에 대한 중요성에도 불구하고 AH(Authentification Header)의 identification field가 항상 보존되지 않기 때문에 network layer에서의 보안기능이 현재 보장되지 않는다. 따라서 보안기능을 유지하기 위해 ESP transport 모드나 tunnel 모드를 이용한 방법을 적용하여 이러한 보안에 대한 문제점을 해결할 수 있다. 그리고 변환의 성능향상을 위해서는 NAT-PT 구현시 최소한의 pool of IPv4 address를 유지하고 대신에 기존의 NAPT(Network Address Port Translation)처럼 포트 넘버를 이용한 변환을 적절하게 사용하고 각 버전별로 운영되는 라우터들과의 호환성도 고려해야 한다.

참고문헌

[1] Stephen A. Thomas, IPng and the TCP/IP Protocols, Wiley, 1996

[2] S.Deering, R. Hinden, " Internet Protocol,Version 6 (IPv6) Specification," IETF RFC 2460, December 1998

[3] R. Gilligan, E. Nordmark, "Transition mechanisms for IPv6 Hosts and Routers," IETF RFC 1933, April 1996

[4] G. Tsirtsis, P. Srisuresh "Network Address Translation - Protocol Translation (NAT -PT)," IETF RFC 2766, February 2000

[5] E. Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT)," IETF RFC 2765, February. 2000.

[6] J. Mogul *et al.*, " IP MTU Discovery Options ", IETF RFC 1063, July 1988

[7] J. McCann *et al.*, "Path MTU Discovery for IPv6 ". IETF RFC 1981, August 1996