

차세대 인터넷의 ITS 유선망에서의 기존 인터넷 서비스 게이트웨이 설계

Design of a Gateway for Existing Internet Service in ITS networks
with the next Generation Internet

최영지
(광운대, 석사과정)

진재경
(광운대, 석사과정)

임승욱
(전자부품연구원)

민상원
(광운대, 교수)

목 차

- I. 서론
- II. DSTM 개요
- III. DSTM 서버와 클라이언트 설계 및 구현
 - 1. DSTM 서버
 - 2. DSTM 클라이언트
 - 3. DSTM 테스트 환경 및 동작 시나리오

- IV. 결론 및 향후 과제
- V. 결론

I. 서론

ITS 망은 과거와 달리 통신기술의 급진적인 발달과 인터넷 사용자 수의 기하급수적인 증가, 그리고 사용자들의 새롭고 다양한 서비스에 대한 요구로 인해 많은 변화를 초래하게 되었다. 이러한 인터넷과 ITS 망의 중추적인 역할을 담당하는 IP(Internet Protocol)는 사용자 증가로 인한 주소 부족, 전자상거래 활성화로 인한 보안 강화의 필요성, 음성 및 비디오와 같은 대용량 데이터 전송의 QoS 보장, 이동성 지원 및 여러 가지 문제에 직면하고 있다. 이러한 문제를 해결하기 위한 방안으로 IETF(Internet Engineering Task Force)는 128bit 주소 체계를 가진 IPv6를 제안하고 있다[1][4].

그러나 ITS 망에 IPv6의 광범위한 사용 및 완전한 도입 이전까지는 장시간 IPv4와의 혼용이 예상되고 IPv6 초기 도입단계에서 IPv6네트워크는 기존 IPv4와의 연동 및 호환을 고려하여 구축하여야 한다. 이와 함께 두 네트워크간의 연동을 위해 IPv4-to-IPv6의 트랜지션(Transition)과 상호공존(Coexistence) 기술들이 운영되어야 한다. 이러한 변환기술은 IETF의 NGTrans(Next Generation Translation) working group에서 활발한 연구 및 개발이 진행되면서 다양한 기술들이 제안되고 있는데, 이 기술들은 크게 터널링과 변환기(Translator)를 이용한 기술로 분류될 수 있다[1][4].

터널링 방식은 서로 다른 IP 버전 방식으로 운용되고 있는 두 네트워크에서 데이터 전송을 위해 임의의 구간에서 터널을 설정한 후, 터널링 구간의 네트워크에서 운용되는 IP 버전의 패킷으로 원래의 패킷을 캡슐화하여 전송하는 방식이며, 변환 방식은 IPv4와 IPv6간의 터널링 없이 변환기의 도움을 받아 자신의 패킷을 그대로 상대방 네트워크로 전송하는 방식이다. 이때의 변환기는 상대방의 IP 버전으로 변환시켜 주는 기능을 담당한다[1][2][3][4].

이러한 트랜지션 기술 중에서 터널링을 이용한 DSTM(Dual Stack Transition Mechanism)은 소규모 IPv6 native 망의 IPv4 응용과 IPv4망의 응용과의 상호접속을 위한 기술이다. 본 논문 2장에서는 ITS 망에서의 DSTM 기술의 일반적인 개요를 알아보고 3장에서는 DSTM 기술의 필수 요소인 DSTM Server와 Client의 설계와 구현에 대하여 기술한다. 4장에서는 결론과 향후연구 과제에 대하여 기술한다[1][2][3][4].

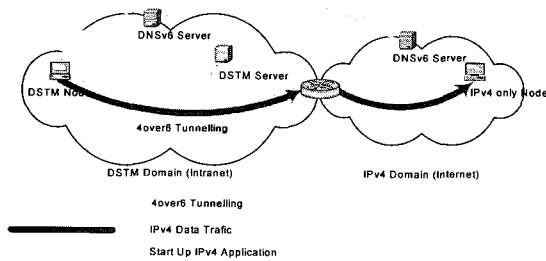
II. DSTM 개요

NGTrans working group은 DSTM의 동작을 위해서 표1과 같은 새로운 용어를 정의 하며 <그림 1>과 같은 Outbound 트래픽을 위한 기본적인 DSTM 동작을 수행한다[1][2][4].

*본 논문은 2002년도 전자부품연구원의 위탁과제의 결과입니다.

〈표 1〉 DSTM 용어 해설

Terminology	Description
DSTM Domain	DSTM Server가 운영되는 IPv6 망
DSTM Server	IPv4 주소 Pool을 유지하며 DSTM Node에 IPv4 주소를 할당하는 프로세스
TEP (Tunnel End Point)	IPv4 패킷을 포함하고 있는 IPv6 패킷의 목적지
4Over6 Tunneling	IPv4 패킷을 IPv6 패킷으로 캡슐화하는 인터페이스
DSTM Client	IPv4 주소를 받아오는 프로세스
DSTM Node	IPv4, IPv6 Dual Stack을 가진 Node



〈그림 1〉 DSTM 기본 동작

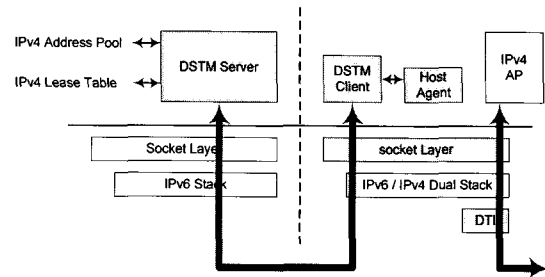
Dual stack의 DSTM Node에서 운영되고 있는 응용이 IPv4-only 노드에서 운영되고 있는 IPv4 응용과 세션을 설정하고자 할 때, DSTM Node는 DNS 네임의 "A" RR (Resource Record)에 대한 질의 메시지를 전송하고 이에 대한 응답을 받는다. DSTM Node는 IPv4 패킷을 전송하기 위해 IPv4 주소를 필요로 하므로 DSTM Server에서 임시 IPv4 주소를 할당받는다. DSTM Node는 4over6 Tunneling 인터페이스를 사용하여 IPv6로 캡슐화된 패킷을 TEP에 전송하고, TEP는 받은 패킷을 IPv4 패킷으로 복원하여 목적지 IPv4 Node에 전송한다[1][2][4].

현재 draft에 의하면 이러한 동작을 위해서는 확장된 DNSv6와 DHCPv6가 요구되며 Inbound 트래픽은 고려하지 않는다[3].

III. DSTM 서버와 클라이언트 설계 및 구현

DSTM Node가 IPv4 네트워크의 응용과 통신을 위해서는 세션을 설정하고자 하는 시점과 세션을 끝내는 시점을 파악하여 임시 IPv4 주소를 할당 및 해제할 수 있는 DSTM Client와 임시 IPv4 주소 Pool을 관리하며 DSTM Node의 요구에 따른 IPv4 주소 리스트를 유지하는 DSTM Server가 반드시 필요하다.

Draft 상에는 확장된 DHCPv6 서버와 클라이언트를 사용한다. 그러나, 이러한 구현이 이루어지지 않았기에 이를 대체할 수 있는 응용을 그림 2와 같이 구현하였다.



〈그림 2〉 구현된 DSTM Server와 Client 구조도

1. DSTM 서버

DSTM Server는 DHCPv6의 옵션 필드를 확장하여 사용할 수 있다. 그러나 현재 구현되지 않은 상태이므로 DSTM Node의 요구에 따라 IPv4 주소를 동적으로 할당할 수 있는 DSTM Server를 구현하였다. 구현된 DSTM 서버는 〈그림 2〉와 같이 임시 IPv4 주소를 유지하는 IPv4 Address Pool과 할당된 IPv4 주소를 관리하는 〈그림 3〉의 IPv4 Lease Table로 구성된다.

```

/* IPv4 Address Lease Table */
struct IPv4Lease_Table {
    struct in_addr host_ipv4address; /* IPv4 Address */
    struct in6_addr host_ipv6address; /* Host IPv6 Address */
    struct in_addr TEP_ipv4address; /* TEP IPv4 Address */
    struct in6_addr TEP_ipv6address; /* TEP IPv6 Address */
    unsigned short allocated_port; /* Resered port ranges */
    unsigned int expired_time_value; /* Leased Time(Microsecond) */
    struct IPv4Mapping_Table *next;
};
    
```

〈그림 3〉 IPv4 주소 관리를 위한 자료 구조

2. DSTM 클라이언트

DSTM Node는 그림 2와 같이 DSTM Client, Host Agent, 변경된 socket layer, DTI(Dynamic Tunneling Interface)로 구성된다. DSTM Client는 IPv4 주소를 요구하고, 네트워크 인터페이스에 할당, 해제 및 DTI의 활성화 및 비활성화를 책임진다. Host Agent는 실제적으로 첫째, DSTM Node의 응용이 IPv4 Node의 응용과 세션을 설정하고자 할 때와 둘째, 세션이 끝나고 IPv4 주소가 필요 없어지는 시점을 구분하여 준다.

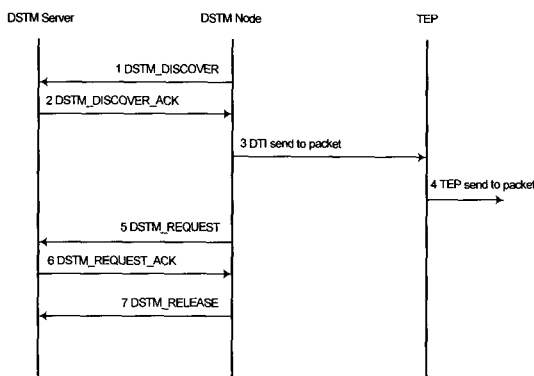
또한 리눅스 커널의 기존 socket layer를 변경하였다. 이는 Host Agent의 첫번째 작업을 수행하기 위해

socket layer의 sys_socket() 시스템 콜이 호출하는 inet_create() 함수를 수정하여 IPv4 응용과 IPv6 응용을 구분할 것이다. 마지막으로 DTI는 SUSE의 fourtun 코드를 수정 및 보안하여 4over6 Tunneling 인터페이스의 기능을 수행하도록 했다. Fourtun 코드는 아직 공개되지 않은 코드로서 리눅스 2.2 커널 기반으로 구성되어 있으며 현재 사용되고 있는 6over4 서비스의 IPv6 over IPv4 tunnel 기능을 수행하기 리눅스의 /usr/srv/linux/net/ipv6/sit.c에 정의되어 있는 SIT (Sample Internet Transition) 인터페이스와 거의 유사하게 구성되어 있다[5][6][7].

3. DSTM 테스트 환경 및 동작 시나리오

DSTM Server와 Client는 그림 4와 같은 5가지 메시지 타입을 정의 한다. DSTM Client는 처음 IPv4 주소 요구를 위한 DSTM_DISCOVER 메시지와 IPv4 주소의 재사용을 위한 DSTM_REQUEST 메시지, IPv4 주소 사용 해제 및 반환을 위한 DSTM_RELEASE 메시지가 있다.

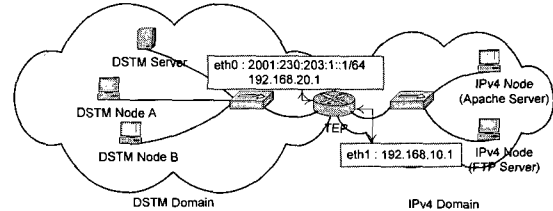
DSTM Server는 IPv4 주소를 처음 요구를 처리하는 DSTM_DISCOVER_ACK 메시지와 IPv4 주소 사용 연장을 요구하는 DSTM_REQUEST의 응답인 DSTM_REQUEST_ACK 메시지를 정의한다. 주소를 할당 받은 DSTM Node는 DTI를 통하여 TEP까지 캡슐화된 패킷을 보내고 TEP는 받은 패킷을 원래의 IPv4 패킷으로 변환하여 목적지까지 전송한다.



〈그림 4〉 DSTM 동작 과정 시나리오

구현물의 테스트 망은 〈그림 5〉와 같이 구성하여 DSTM Node A와 B에서 ping 테스트, IPv4 Domain내의 웹 서버와 FTP 서버에 각각 접속하는 테스트를 수행하였다. DSTM 동작에 맞게 확장된 DNSv6 서버 대신 /etc/hosts 파일에 요구되어지는 DNS 정보를 수정하였다. DSTM Node A는 DSTM 서버에 IPv4 주소를 요구하고 응답을 받아DSTM

IPv4 Address Pool range내의 192.168.20.2를 할당 받는다. DTI를 통하여 캡슐화된 패킷은 TEP에서 발신지 주소가 할당 받은 IPv4 주소인 192.168.20.2로 변형된 후 웹 서버 192.168.10.2에 접속한다.



DSTM IPv4 Address Pool Range : 192.168.20.2 ~ 192.168.20.254
DSTM Server : 2001:230:203:1::1/64
DSTM Node A : 2001:230:203:1::21/64
DSTM Node A : 2001:230:203:1::22/64
IPv4 Node (Apache Server) : 192.168.10.2
IPv4 Node (FTP Server) : 192.168.10.3

〈그림 5〉 DSTM 테스트 망

IV. 결론 및 향후 과제

본 논문은 ITS 망에 IPv6 네트워크의 완전한 도입 이전의 IPv4 네트워크와 혼용되는 과도기적인 시기에 사용되는 IPv4-to-IPv6의 트랜지션 기술인 DSTM 기술 중 DSTM Server와 Client의 설계 및 구현에 대하여 기술하였다. DSTM 기술은 소규모IPv6 native 망의 IPv4 응용과 IPv4망의 응용과의 상호접속을 위한 최적의 기술이다. 그러나 현재 제안된 draft 상에는 확장된 DHCPv6, DNSv6가 필요하며, 또한 Out-bound 트래픽만 고려하였다. 결과적으로 Inbound 트래픽에 대한 논의가 더 필요할 것으로 보인다. 구현된 DSTM Server와 Client역시socket layer의 수정과 메시지 보안에 대한 고려, DSTM Client의 User Interface, 세션의 설립과 해제의 명확한 구분 등이 충분치 않았다. 그러므로 보다 더 효율적인 DSTM 동작을 위해서는 추가적인 연구가 더 필요하겠다.

참고문헌

1. J. Bound, et al., "Dual Stack Transition Mechanism (DSTM) Overview", IETF darft-ietf-ngtrans-dstm-overview-00.txt, June 2002
2. J. Bound, et al., "Dual Stack Transition Mechanism (DSTM)", IETF darft-ietf-ngtrans-dstm-08.txt, January 2002
3. W. Biemolt, et al., "An Overview of the introduction of IPv6 in the Internet", IETF darft-ietf-ngtrans-introduction-to-ipv6-transition-08.txt, March 2002

4. 김용진 외 2인, 차세대 인터넷 프로토콜 IPv6, *IPv6 포럼 코리아*, 2002
5. W. Richard Steven, *Advanced Programming in the UNIX Environment*, Addison Wesley, 1998
6. W. Richard Steven, *UNIX Network Programming Volume 1*, Prentice Hall, 1997
7. Daniel P. Novet, et al., *Understanding the LINUX KERNEL*, O'reilly, 20