

차세대 인터넷 서비스를 위한 DSTM의 IPv4 over IPv6 터널 구현

The Implementation of an IPv4 over IPv6 Tunnel of the DSTM for Next-Generation Internet Service

이승민*
(Seung-Min Lee)

민상원**
(Sang-Won Min)

김용진 ***
(Yong-Jin Kim)

요 약

본 논문은 DSTM (Dual Stack Transition Mechanism) 서비스를 제공하기 위해 필요한 구성 요소 중에서 DTI (Dynamic Tunnel Interface)를 통한 IPv4 over IPv6 터널에 대해 분석한 후 요소기술 및 기능들을 정의하여 적합한 모듈을 설계하여 구현하였다. 그리고 구현된 IPv4 over IPv6 터널이 실제 서비스될 수 있는지를 확인하기 위해 다양한 환경에서 기존의 6over4 서비스와 IPv4 트래픽과의 성능 비교 분석을 위한 실험 후 결과를 제시하였다. 제시된 결과는 IPv6 터널 헤더에서의 checksum 계산의 불필요성에 따른 지연 감소 등으로 6over4 서비스보다는 우수한 성능을 나타내었다.

Abstract

In this paper, we propose an efficient algorithm that assigns single temporary IPv4 address and port number to improve efficiency of IPv4 address that is allocated in DSTM service. And, we have analyzed the elementary functions for DSTM and have designed the functional modules. Also, we have implemented the DTI interface for encapsulation and decapsulation of IPv6 packets. The performance analysis and comparison are investigated whether the appropriate interworking service is possible or not. Our observation results show that the performance of IPv4 over IPv6 tunneling is suitable to DSTM service due to the reduction of delay by eliminating checksum calculations in the header of IPv6 tunneling.

Key Words : IPv4, IPv6, DSTM, DTI, tunneling

본 논문은 모다정보통신(주)와 2006년 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

* 주저자 : 삼성전자 무선사업부 연구원

** 공저자 : 광운대학교 전자통신공학과 부교수

*** 공저자 : 모다정보통신(주) 연구소장

† 논문접수일 : 2007년 12월 28일

I. 서론

현재 인터넷 환경에서 주소와 라우팅 기능을 담당하고 있는 IPv4 프로토콜은 32비트 클래스 기반의 주소 체계를 가지고 있다. 이러한 주소체계는 인터넷 주소 할당 과정에서 수요에 따른 정확한 주소 분배에 어려움이 있고, 인터넷의 기하급수적인 성장에 따른 IP 주소 수요의 급격한 증가 때문에 향후 이동 통신 및 홈 네트워킹, 그리고 정보가전 등에서 요구하는 주소 수요를 충족시킬 수 없다. 또한 기존의 IPv4 프로토콜은 최근 이슈가 되고 있는 보안 문제와 QoS (Quality of Service), 그리고 향후 서비스 예정인 이종 통신네트워크들과의 연동에 한계가 있다. 이러한 문제를 해결하기 위해 IETF (Internet Engineering Task Force)에서는 IPv4의 주소 고갈 문제를 해결하고 새로운 서비스들을 수용할 수 있도록 IPv6를 정의하고 있다 [1].

IPv6는 기존의 IPv4의 기능들을 그대로 수용하면서 발전하는 인터넷 통신 기술을 수용할 수 있도록 설계되었으며 향후 서비스될 것으로 예상되는 통신 기술들이 IPv6 주소체계를 수용할 수 있도록 호환성을 고려하여 개발되었다. 그러나 IPv6가 기존의 IPv4의 문제점들을 개선하여 더 나은 성능을 제공하는 프로토콜임에도 불구하고 기존의 IPv4 네트워크를 일시에 대체하기에는 어려움이 있다. 따라서 완전한 IPv6 네트워크 도입까지는 IPv4 네트워크와 IPv6 네트워크의 연동을 위해 트랜지션 메커니즘을 사용한 과도기적 솔루션을 필요로 한다. 이를 위해 현재 다양한 트랜지션 메커니즘을 위한 연구개발이 활발히 진행되고 있으며 이러한 트랜지션 메커니즘은 크게 터널링과 변환기를 이용한 메커니즘으로 구분될 수 있다 [2, 3].

변환기를 이용한 방식은 IPv4와 IPv6간의 터널링 없이 두 네트워크 사이의 변환기를 통해 자신의 패킷을 상대방 네트워크로 전송하는 방식이다. 이러한 변환방식으로는 RFC (Request For Comments) 표준으로 확정된 NAT-PT (Network Address Translation-Protocol Translation)와 SIIT (Stateless IP/ICMP Translation) 등이 제안되어 구현되고 있지만 변환방식의 특성 때문

에 여러 제한들이 존재한다 [4, 5]. 반면에 터널링 방식은 서로 다른 IP 버전이 운용되고 있는 두 네트워크간의 패킷 전송을 위해 터널링을 설정한 후 터널링 구간의 네트워크에서 운용되는 IP 버전의 패킷으로 원래의 패킷을 캡슐화하여 전송하는 방식이다. 현재 터널링 방식으로 운용되고 있는 서비스는 IPv4 네트워크를 통한 IPv6 호스트의 연동 시나리오에서 사용되는 6over4 등이 있다.

이러한 시나리오 외에 IPv6 네트워크에서 IPv4 패킷을 전송하고자 하는 필요성이 제기됨에 따라 새롭게 DSTM (Dual Stack Transition Mechanism)이 제안되어 표준화가 진행되고 있다. DSTM은 6over4와는 반대로 IPv4 over IPv6 터널을 이용한 방식으로 native IPv6 네트워크에 존재하는 듀얼 스택 IPv6 호스트의 IPv4 애플리케이션과 IPv4 네트워크에 존재하는 IPv4 애플리케이션간의 통신 수단을 제공한다. 듀얼 스택 IPv6 호스트는 IPv4와 IPv6의 두 가지 프로토콜 스택을 사용할 수 있기 때문에 IPv4 호스트 또는 IPv6 호스트와 통신하기 위해서는 통신하고자 하는 호스트와 동일한 버전의 프로토콜 스택을 이용하여 통신한다.

DSTM 서비스에서 native IPv6 네트워크에 존재하는 듀얼 스택 IPv6 호스트의 IPv4 애플리케이션이 생성한 IPv4 패킷은 IPv6 터널 패킷으로 캡슐화되어 IPv6 터널을 통해 경계 라우터까지 전송된다. 그리고 전송된 IPv6 터널 패킷은 경계 라우터에서 다시 IPv4 패킷으로 복원되어 목적지 IPv4 호스트에게 전달된다. 그러나 듀얼 스택 IPv6 호스트는 IPv4 주소를 가지고 있지 않기 때문에 IPv4 패킷의 소스 주소를 설정하기 위해 DHCPv6를 이용한 다이내믹한 주소할당이 필요하다. 그리고 IPv4 패킷 전달을 위해서는 IPv4 패킷을 IPv6 패킷으로 캡슐화하고 복원하는 기능을 수행하는 DTI (Dynamic Tunnel Interface)가 구현되어야 하고 듀얼 스택 IPv6 호스트와 경계라우터까지 IPv4 over IPv6 터널이 구성되어야 한다.

본 논문은 이러한 시나리오에 적용시킬 수 있는 DSTM 서비스를 제공하기 위해 필요한 구성 요소 중에서 DTI를 통한 IPv4 over IPv6 터널에 대해 본

석한 후 요소기술 및 기능들을 정의하여 적합한 모듈을 설계하여 구현하였다. 그리고 구현된 IPv4 over IPv6 터널이 실제 서비스될 수 있는지를 확인하기 위해 다양한 환경에서 기존의 6over4 서비스와 IPv4 트래픽과의 성능 비교 분석을 위한 실험 후 결과를 제시하였다. 제시된 결과는 IPv6 터널 헤더에서의 checksum 계산의 불필요성에 따른 지연 감소 등으로 6over4 서비스보다는 우수한 성능을 나타내었다. 본 논문은 서론에 이어 2장에서 DSTM의 기본 알고리즘과 6over4 서비스에 대해 설명하고, 3장에서는 IPv4 over IPv6 터널 설계와 구현 방법에 대해 설명하였다. 그리고 4장에서는 IPv4 over IPv6 터널의 성능 측정 및 실험 결과를 제시하였으며 마지막 5장에서 결론을 맺었다.

II. DSTM

여기서는 소단원에 관한 내용을 간단히 살펴보겠습니다. 게다가 소소단원에 관한 내용도 간단히 살펴보겠습니다.

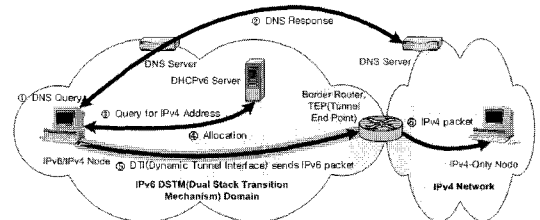
1. DSTM의 기본 동작

DSTM은 native IPv6 네트워크와 IPv4 네트워크간의 연동을 위한 트랜지션 메커니즘이다. 이를 위해 IPv6 주소만을 가지는 듀얼 스택 IPv6 호스트가 IPv4 네트워크의 IPv4-only 호스트와 연결될 수 있도록 IPv6 호스트에서 DSTM 도메인의 경계 라우터까지 IPv6 터널링을 구성한다. 그리고 IPv4 패킷 전송시 IPv6 호스트가 IPv4 헤더의 IPv4 소스 주소를 구성하는데 필요한 임시 global IPv4 주소를 DHCPv6 서버로부터 할당 받는 메커니즘이 추가로 구성되어야 한다. 이렇게 구성된 DSTM은 라우터가 IPv6 네트워크를 거쳐가는 IPv4 패킷들에 대해 IPv6 라우팅 테이블만을 유지하기 때문에 IPv6 도입에 따른 네트워크 관리를 단순화시킬 수 있다는 장점이 있다 [5, 6].

DSTM의 세부 동작 설명을 위해 <그림 1>과 같이 듀얼 스택 IPv6 호스트와 IPv4-only 호스트간 아

웃바운드 (Outbound) 세션의 패킷 전송을 <그림 1>에서 단계별로 고찰하였다 [7]. 본 논문에서 설명되는 세션의 방향은 아웃바운드와 인바운드 세션으로 구별되는데 아웃바운드 세션은 IPv6 네트워크에서 IPv4 네트워크로 패킷을 전송하는 방향이고 인바운드 세션은 반대 방향을 나타낸다 [5]. IPv4/IPv6 듀얼 스택 IPv6 호스트가 IPv4-only 목적지 호스트의 IPv4 주소를 획득하기 위해 DNS 네임의 "A" RR (Resource Record)에 대한 query 메시지를 전송하면 IPv4-only 호스트의 네임을 관리하는 DNSv4 서버는 query 메시지에서 요구하는 IPv4 주소로 응답한다. 이렇게 목적지 주소를 획득한 애플리케이션은 첫 IPv4 패킷을 DTI 인터페이스로 전송하는데 처음 패킷을 보내는 경우 IPv4 헤더 구성을 위해 IPv4 소스 주소를 필요로 한다.

듀얼 스택 IPv6 호스트는 IPv4 주소를 획득하기 위해 DHCPv6 메시지를 사용하여 DHCPv6 서버에게 request 메시지를 전송하고 이 메시지를 수신한 서버는 클라이언트의 위치를 파악하여 임시 IPv4 global 주소를 제공하여야 한다. 구성된 IPv4 패킷을 전달 받은 DTI 인터페이스는 IPv4 패킷을 IPv6 패킷으로 캡슐화하여 TEP(Tunnel End Point)에 전송하는데 TEP는 IPv4 패킷을 포함하고 있는 IPv6 패킷의 목적지로서 본 논문의 경우 TEP는 듀얼 스택 경계 라우터로 설정된다. IPv6 패킷을 전송 받은 경계 라우터는 패킷을 IPv4 패킷으로 복원하여 목적지로 포워딩한 후 듀얼 스택 호스트의 IPv4 주소와 IPv6 주소정보를 저장한다 [7].

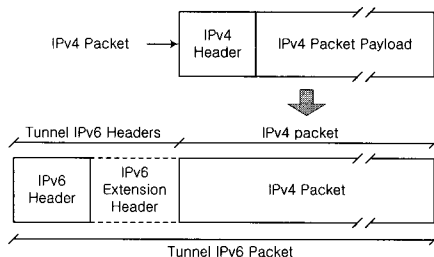


<그림 1> DSTM 알고리즘 동작
<Fig. 1> Operation of the DSTM algorithm

DSTM은 IPv6 도메인 내에서 IPv6 패킷만을 운용하기 때문에 IPv4 패킷 전달을 위한 IPv4 라우팅을 사용할 수 없다. 듀얼 스택 IPv6 호스트는 IPv4 패킷을 IPv6 패킷으로 캡슐화하여 TEP로 전송하고 TEP에서는 IPv6 패킷을 다시 IPv4 패킷으로 복원하여 IPv4 네트워크로 전송한다. IPv4 패킷의 캡슐화 및 복원은 각 노드의 DTI 인터페이스에서 수행된다. 이 때 DTI 인터페이스는 듀얼 스택 IPv6 호스트에 대한 임시 IPv4 주소할당과 동일하게 터널 IPv6 헤더의 목적지 TEP IPv6 주소를 할당하여야 한다. 그러나 TEP 주소는 수동으로 구성될 수 있고 IPv6 노드가 DHCPv6 서버로부터 IPv4 compatible 주소를 획득하는 경우에는 다이내믹하게 구성될 수 있다.

2. IPv4 over IPv6 터널 요구사항

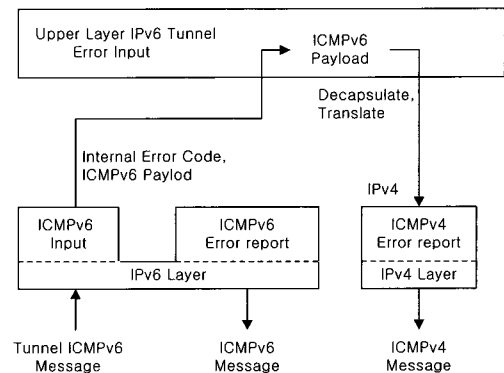
DSTM 도메인을 구성하고 있는 노드들의 DTI 인터페이스에서 수행되는 캡슐화 방식은 6over4의 IPv6 over IPv4 터널 방식과는 반대로 <그림 2>와 같이 IPv4 패킷을 IPv6 패킷으로 캡슐화하고 이러한 기술을 IPv6 터널링이라 한다. IPv6 터널링은 IPv4 패킷을 IPv6 패킷의 페이로드로 전송하기 위해 두 IPv6 노드 간에 가상링크를 설정하는 기술로서 터널 양단의 두 노드 관점에서 보면 IPv6 터널 즉, 가상링크는 IPv6가 링크계층 프로토콜로 동작하는 point-to-point 링크 기능을 수행한다. 그리고 패킷을 캡슐화하는 노드는 터널 entry-point 노드라 하고, 캡슐화된 패킷을 다시 원형 패킷으로 복원하는 노드는 터널 패킷의 목적지로서 터널 exit-point 노드라 한다 [8, 9].



<그림 2> IPv4 over IPv6 Encapsulation

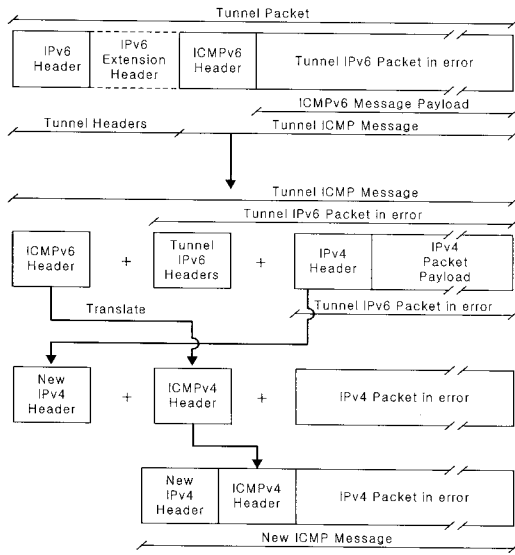
<Fig. 2> Structure of an IPv6 tunnel packet

IPv6 터널링은 IPv6 패킷의 처리과정 동안 탐지된 에러를 패킷의 송신측에 보고하여야 한다. IPv6 터널을 포함하는 포워딩 경로 상에서 터널에 존재하지 않는 노드에 의해 탐지된 에러는 IPv4 원형 패킷의 송신측에 전달되고 터널 내의 노드에 의해 탐지된 에러는 터널 패킷의 송신측 entry-point 노드에 전달된다. 그리고 터널 패킷 에러가 터널 패킷의 페이로드인 IPv4 패킷과 관련된 에러인 경우에는 IPv4 패킷의 송신측에 전달된다. 그러나 터널 내에서 탐지된 에러를 IPv4 패킷의 소스에 전달하기 위해서는 터널 entry-point 노드가 터널 내에서 전송 받은 ICMPv6 메시지를 IPv4 패킷의 소스에 전달해야 하기 때문에 <그림 3>과 <그림 4>와 같이 ICMPv6 메시지를 ICMPv4 메시지로 변환해 주어야 한다.



<그림 3> 듀얼스택 IPv6 호스트의 ICMP 에러 메시지 흐름도

<Fig. 3> ICMP error message flow of a host with dual stack protocols



<그림 4> ICMPv4 메시지 구성
 <Fig. 4> Construction of ICMPv4 message

<그림 3>과 같이 IPv6 터널 entry-point 노드가 터널 내부로부터 터널 ICMPv6 메시지를 수신하면 터널 entry-point 노드의 IPv6 계층은 수신된 ICMPv6 메시지를 ICMPv6 입력에 전달하여 미리 정의된 ICMPv6 타입과 코드에 따라 내부 에러 코드를 생성한다. 생성된 내부 에러 코드는 <그림 4>의 ICMPv6 메시지 페이로드와 함께 IPv6 터널 상위계층 에러 입력에 전달되어 ICMPv4 메시지의 타입과 코드 변환에 사용된다. 상위계층에서는 ICMPv6 메시지 페이로드인 터널 IPv6 패킷을 복원하여 IPv4 패킷을 획득한 후 IPv4 패킷의 헤더와 내부 에러 코드를 ICMPv4 에러 입력에 전달한다. ICMPv4 에러 입력에서는 복원된 IPv4 패킷과 내부 에러 코드로부터 ICMPv4 메시지의 타입과 코드를 생성한다. 그리고변환된 ICMPv4 메시지는 IPv4 헤더의 소스 주소에 터널 entry-point 노드의 IPv4 주소가 세팅되고 목적지 주소에 IPv4 패킷 소스 노드의 주소가 세팅된다.

IPv4 over IPv6 터널을 통해 IPv4 패킷을 전송하기 위해 IPv4 패킷에 IPv6 헤더를 추가하게 되면 패킷의 크기가 증가하기 때문에 터널 패킷은 각 링크의 MTU (Maximum Transfer Unit) 크기에 따라 분할

될 수도 있다. 그러나 IPv6 표준에서는 터널 내의 다른 노드에게 패킷을 포워딩하는 터널 내의 중간 노드는 포워딩되는 패킷을 분할 할 수 없다. 그러므로 캡슐화된 터널IPv6 패킷은 터널 entry-point 노드에서 생성한 IPv6 패킷으로 간주되기 때문에 터널 entry-point 노드는 터널 IPv6 패킷 분할을 수행할 수 있어야 한다. 따라서 IPv4 패킷이 터널에 전달될 때 IPv4 패킷의 크기가 터널 MTU (터널 헤더의 크기를 뺀 entry-point 노드와 exit-point 노드 사이의 Path MTU)보다 큰 경우에는 IPv4 패킷 헤더의 DF (Don't Fragment) 비트의 세팅 여부에 따라 다르게 처리되어야 한다.

IPv4 패킷 헤더의 DF 비트가 세팅되어 있는 경우에는 entry-point 노드는 패킷을 버린 후 MTU size 필드를 터널 MTU 크기로 세팅하여 unreachable 타입과 packet too big 코드를 가지는 ICMP 메시지를 생성한다. 그리고 IPv4 헤더의 DF 비트 플래그가 세팅되어 있는 경우 전송되는 packet too big 메시지는 IPv4 패킷의 소스에 전송된다. 그러나 DF 비트가 세팅되어 있지 않은 경우에는 entry-point 노드는 IPv4 패킷을 캡슐화하고 캡슐화된 터널 패킷을 path MTU를 초과하지 않도록 분할하여야 한다 [9].

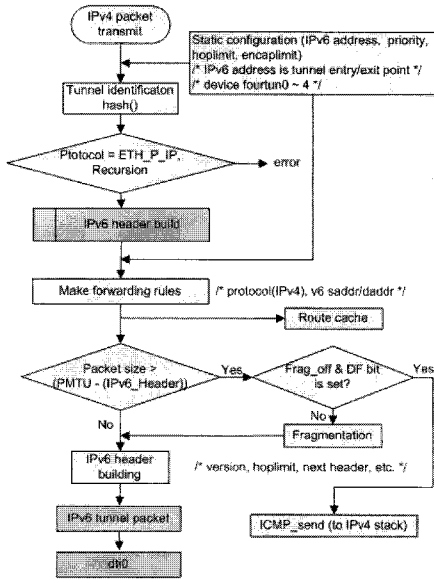
Ⅲ. IPv4 over IPv6 터널링 설계 및 구현

1. DTI 송수신 설계

IPv4 over IPv6 터널을 위한 DTI는 크게 <그림 5>의 구조를 가지는 터널 entry 노드에서의 송신부 기능과 <그림 6>의 구조를 가지는 터널 exit 노드에서의 수신부 기능으로 구분하여 가상 인터페이스 (Virtual Interface)로 설계하였다. 그러나 구현된 DTI의 주소할당 부분은 다이나믹하게 DHCPv6와 연동할 수 있도록 구성되어야 하지만, 아직 DHCPv6 표준화가 진행중이고 본 논문에서 사용할 수 있을 정도의 구현된 시스템이 없기 때문에 주소 설정과 다양한 옵션 설정은 static하게 구성하였다.

<그림 5>에서 송신부의 기능을 위해 상위 계층

에서 전달된 IPv4 패킷은 우선 터널식별을 위해 hash 함수를 이용해 터널 ID 값을 생성하여 캐쉬한 후 프로토콜이 IP를 위한 이더넷 코드인지와 중복 터널인지를 확인한다.



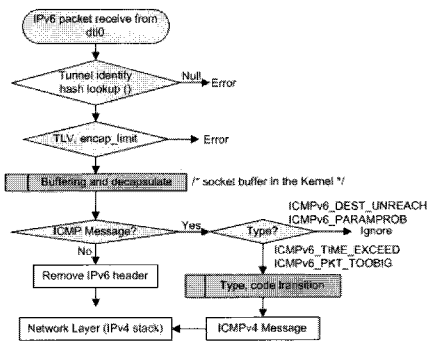
<그림 5> DTI 송신부의 플로우 차트
<Fig. 5> Flow chart of a DTI transmitting module

그리고 DTI 인터페이스의 핵심 기능인 IPv4 over IPv6 캡슐화를 위해 IPv6 터널 헤더를 생성하는데, 이 때 firewalling에서와 같은 포워딩 룰 기반의 플로우를 생성한 후 이 루트를 캐쉬하여 다음 터널 패킷의 루트 설정에 이용한다.

실제 인터넷 환경에서의 패킷 전송을 위해서는 분할도 고려해야 하는데 이를 위해 IPv4 패킷의 MTU 크기가 터널의 path MTU에서 IPv6 헤더를 빼 값보다 클 경우, 프래그 옵셋값과 DF 비트가 세팅되어 있으면 IPv4 스택에 에러 메시지를 전송하게 된다. 그리고 두 비트값이 세팅되어 있지 않는 경우에는 패킷을 분할하고 IPv6 헤더를 추가하여 IPv6 패킷을 생성한 후 dti0 인터페이스를 통해 전송한다 [10-12].

<그림 5>의 dti0 송신부에서 전송된 패킷은 IPv6 패킷을 수신하는 터널 exit 노드의 수신부에서 <그림 6>과 같이 hash_lookup 함수를 사용하여 터널 ID

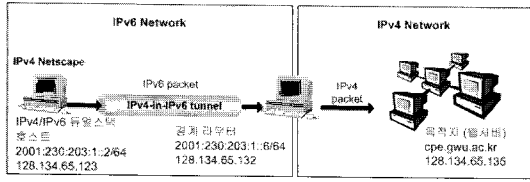
를 식별한 후 IPv6의 헤더의 type, length, value (TLV)형식의 이상 유무와 허용된 encapsulation limit 초과여부를 확인한다. 그리고 IPv6 터널 패킷을 IPv4 패킷으로 복원하기 위해 커널내의 소켓 버퍼를 이용해 패킷을 버퍼링한 후 복원과정을 수행하여 IPv4 스택으로 전달한다. 그러나 전달되는 패킷이 ICMP 메시지인 경우에는 터널 entry-point 노드와 터널exit-point 노드 모두에서 ICMP 메시지를 수신할 수 있다. 따라서 터널 exit-point 노드에서 수신하는 경우 ICMP 메시지의 타입을 검사하여 <그림 6>과 같이 현재 ICMPv6에서 정의되어 있는 두 타입만을 ICMPv4 형식으로 변환한 후 네트워크 계층의 IPv4 스택에 전달한다.



<그림 6> DTI 수신부의 플로우 차트
<Fig. 6> Flow chart of a DTI receiving module

2. 개발환경 및 시험망 구축

DSTM 구성요소 중 본 논문에서 설계한 DTI의 IPv4 over IPv6 캡슐화 및 복원 기능의 성능 분석을 위해 <그림 7>과 같이 리눅스 기반의 구현 및 실험 환경을 구성하였다. <그림 7>의 IPv6 네트워크는 듀얼 스택 IPv6 호스트와 경계 라우터로 구성된 후 draft에서의 자동 주소 할당 대신에 각각 static하게 IPv6 주소와 IPv4 주소를 할당하였다. 그리고 IPv4 네트워크는 듀얼 스택 IPv6 호스트가 액세스할 수 있도록 IPv4 웹서버로 구성된 후 듀얼 스택 IPv6 호스트와 IPv4-only 호스트의 패킷 전송을 위한 디폴트 게이트웨이가 경계 라우터로 설정되도록 라우팅 테이블을 수정하였다.



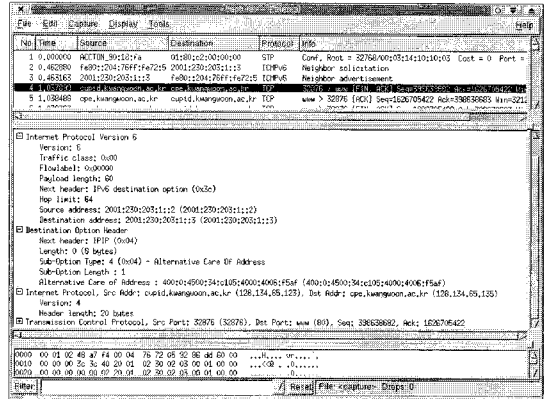
<그림 7> 실험 및 구현 환경
<Fig. 7> Test and implementation environment

<그림 8>은 실험 환경 구성 후 DTI의 동작 여부를 확인하기 위해 듀얼 스택 IPv6 호스트에서 경계 라우터까지 ICMP echo request/reply를 응용한 ping 실험 결과로서 dti0 인터페이스에서의 tcpdump를 이용한 패킷 필터링 결과이다 [13]. dti0 인터페이스는 DTI 기능이 구현된 가상 인터페이스로서 IPv4 패킷을 캡슐화하여 IPv6 터널 패킷을 실제 네트워크 인터페이스인 eth0에 전달하거나 eth0로부터 전달받은 IPv6 터널 패킷을 IPv4 패킷으로 복원한다. <그림 8>에서 나타난 바와 같이 IPv4 echo request 패킷이 dti0 인터페이스에서 IPv6 패킷으로 캡슐화되어 eth0 인터페이스를 통해 경계 라우터까지 전달된 후 다시 IPv4 패킷으로 복원되어 전달되는 것을 확인할 수 있다.

```
[root@cupid/root]#tcpdump -n -i dti0
user level filter, protocol ALL, datagram packet socket
tcpdump: listening on all devices
11:52:51.936737 eth0 M 802.1d config 8000:00:03:14:10:10:03.8018 root 8000.00:03
14:10:10:03 pathcost 0 age 0 max 20 hello 2 fdelay 15
11:52:51.937978 eth0 B arp who-has 128.134.65.6 tell 128.134.65.1
11:52:51.978742 dti0 > 128.134.65.123 > 128.134.65.132: icmp: eth0 request(DP)
11:52:51.978761 eth0 > 0:0:0:0:0:0:0:0:4:76:72:5:92 ipv6 146:
11:52:51.979261 eth0 < 0:0:0:24:c5:ef:33 0:0:0:0:0:1 ipv6 146:
11:52:51.979261 dti0 < 128.134.65.123 > 128.134.65.123: icmp: eth0 reply
```

<그림 8> dti0 인터페이스의 동작
<Fig. 8> Operation of a dti0 interface

<그림 9>는 실제로 듀얼 스택 IPv6 호스트에서 목적지 IPv4 웹서버를 액세스 할 수 있는지를 확인하기 위한 실험 결과로서 듀얼 스택 IPv6 호스트의 dti0 인터페이스에서 Ethereal 프로그램을 사용하여 패킷 필터링한 결과를 캡처하였다 [14]. 실험 결과에서와 같이 static하게 할당받은 128.134.65.123의 IPv4 주소를 가지는 듀얼 스택 IPv6 호스트가 128.134.65.135의 IPv4 주소를 가지는 목적지 웹서버에게 전송한 IPv4 패킷이 IPv6 패킷으로 캡슐화되는 것을 확인할 수 있다. 그리고 터널 구간이 듀얼



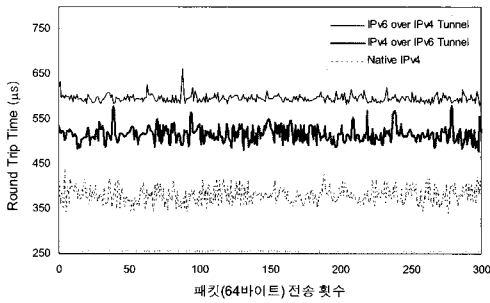
<그림 9> dti0 인터페이스 패킷 필터링 결과
<Fig. 9> Packet filtering result of the dti0 interface

스택 IPv6 호스트에서 경계 라우터까지이기 때문에 IPv6 터널 패킷의 IPv6 목적지 주소는 경계 라우터의 IPv6 주소인 2001:230:203:1::3으로 세팅되고, IPv6 헤더의 next header 필드가 IP-in-IP 캡슐화를 나타내기 위한 IPIP로 세팅되어 있는 것을 확인할 수 있다.

IV. 성능 측정 및 결과

구현된 DTI 인터페이스의 동작 확인 후 DSTM을 위한 IPv4 over IPv6 터널의 성능 여부를 확인하기 위해 본 논문에서는 현재 서비스되고 있는 6over4 서비스의 IPv6 over IPv4 터널에서의 패킷 전송 그리고 기존의 IPv4 패킷 전송과 각각 비교 분석하였다. 이를 위해 IPv4 패킷과 구현된 IPv4 over IPv6 터널 패킷 전송은 ping을 이용하여 실험하였고 6over4 서비스는 IPv6 애플리케이션간의 통신이기 때문에 ping의 IPv6 애플리케이션인 ping6를 이용하여 실험하였다.

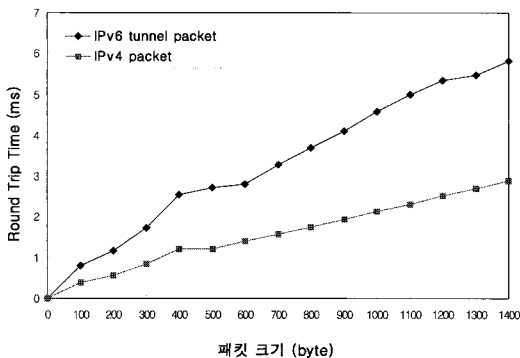
<그림 10>은 1홉 단위의 터널 구간에서 각 패킷 별로 전송횟수에 따른 RTT (Round Trip Time)을 비교하여 도식한 실측 결과이다.



<그림 10> 패킷별 전송횟수에 따른 RTT 비교
 <Fig. 10> RTT comparison by packet transmission number of times

<그림 10>에서 나타난 바와 같이 IPv4 패킷의 캡슐화 처리에 따른 지연 때문에 두 터널 패킷의 RTT가 IPv4 패킷의 전송속도에 비해 저하된 성능을 나타내었다. 그러나 본 논문에서 구현된 IPv4 over IPv6 터널 패킷은 IPv6 터널 헤더의 checksum 계산이 필요하지 않기 때문에 IPv4 터널 패킷의 checksum 계산이 필요한 IPv6 over IPv4 패킷의 전송속도에 비해 약 80 s정도의 우수한성능을 나타내었다. 따라서 기존 6over4 서비스 외에 DSTM 서비스가 가능함을 알 수 있었다.

<그림 11>은 순수한 IPv4 패킷과 구현된 IPv4 over IPv6 패킷을 크기별로 전송한 실측 결과를 도식한 그림이다.



<그림 11> 패킷 크기에 따른 RTT 비교
 <Fig. 11> RTT comparison by packet size

그림에서 나타난 대로 패킷의 크기가 증가할수록 상대적으로 터널 패킷의 RTT 증가율이 커지는 것을 확인할 수 있다.

이것은 상대적으로 IPv6 기본 헤더의 크기가 40 바이트로서 IPv4 기본헤더의 20바이트에 비해 크고 터널 패킷의 크기가 증가함에 따라 시스템 내부에서의 버퍼링에 따른 지연의 결과이다.

V. 결 론

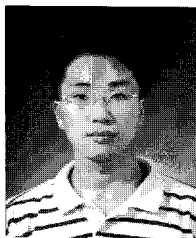
본 논문은 IPv6 초기 도입 단계에서 듀얼 스택 IPv6 호스트와 IPv4-only 호스트간의 통신 방법을 제안하고 있는 DSTM의 IPv4 over IPv6 터널에 필요한 요소 기술들을 분석하여 효율적인 알고리즘을 설계한 후 이를 위한 DTI 인터페이스를 구현하였다. IPv4 over IPv6 터널은 현재 IPv6 애플리케이션들간의 통신 수단을 제공하는 6over4 서비스의 IPv6 over IPv4 터널과는 반대의 기능을 수행한다. 이는 IPv6 네트워크가 점진적으로 도입되면서 통신 환경과 목적에 따라 다양한 트랜지션 메커니즘이 적용될 수 있기 때문이다. 그리고 DTI 인터페이스는 IPv4 패킷을 IPv6 패킷으로 캡슐화하고 다시 캡슐화된 패킷을 복원하는 기능을 수행하는데 이러한 터널 구간은 IPv4 패킷 전송이 불가능한 IPv6 네트워크내에 설정된다.

본 논문에서 구현된 DSTM의 DTI 인터페이스를 통해 실제로 듀얼 스택 IPv6 호스트와 IPv4 호스트와의 연결이 가능하였다. 그리고 성능 분석 결과 IPv6 터널 패킷의 전송 속도가 기존의 순수한 IPv4 패킷 전송 속도에 비해 캡슐화와 복원 과정에 따른 지연 때문에 감소하지만, 6over4의 IPv6 over IPv4 터널에 비해 우수한 성능을 나타내어 다양한 트랜지션 메커니즘의 솔루션으로 활용 가능함을 확인할 수 있었다. 그리고 이렇게 구현된 IPv4 over IPv6 터널은 터널의 특성상 터널 entry-point 호스트와 exit-point 노드 사이의 IPv6 경로가 가상링크로 설정됨으로써 보안성을 제공할 수 있는 장점이 있다. 또한 본 논문에서 구현한 사항들은 IPv4/v6 transition 방안으로 제시되는 여러 tunneling 기법을 구현할 때 일부분을 수정 변경한다면 충분히 활용될 것이다.

참고문헌

- [1] D. C. Lee, D. L. Lough, S. F. Midkiff, N. J. Davis IV, P. E. Benchoff, "The next generation of the internet: aspects of the internet protocol version 6," *IEEE Network Magazine*, vol. 12, no 1, pp. 28 ~ 33, Feb. 1998.
- [2] C. Huiteman, "Teredo: tunneling IPv6 over UDP through NATs," *IETF draft-huitema-v6ops-teredo-05*, April 2005.
- [3] 이승민, 진재경, 민상원, "IPv4/IPv6 프로토콜 및 주소변환 기능의 요소기술 분석 및 설계," *정보과학회논문지*, 제30권, 제1호, pp. 117 ~ 125, 2003. 2.
- [4] G. Tsirtsis and P. Srisuresh, "Network address translation - Protocol translation (NAT-PT)," *IETF RFC 2766*, Feb. 2000.
- [5] E. Nordmark, "Stateless IP/ICMP translation a Algorithm (SIIT)," *IETF RFC 2765*, Feb. 2000.
- [6] J. Bound, L. Toutain, and J. Richier, "Dual stack IPv6 dominant transition mechanism (DSTM)," *IETF draft-bound-dstm-exp-04.txt*, Oct. 2005.
- [7] D. Waddington and F. Chang, "Realizing the transition to IPv6," *IEEE Comm. Magazine*, vol. 40, no. 6, pp. 138 ~ 148, June 2002.
- [8] E. Nordmark and R. Gilligan, "Basic transition mechanism for IPv6 hosts and routers," *IETF RFC 4213*, Oct. 2005.
- [9] W. Stevens, *TCP/IP Illustrated vol. 1*, Addison-Wesley, Dec. 1993.
- [10] W. Stevens, *TCP/IP Illustrated vol. 2*, Addison-Wesley, Jan. 1995.
- [11] W. Stevens, *TCP/IP Illustrated vol. 3*, Addison-Wesley, Jan. 1996.
- [12] A. Miller, *Implementing IPv6*, M&T Books, March 2000.
- [13] Tcpdump, <http://www.tcpdump.org>
- [14] Ethereal, "Ethereal User's Guide," <http://www.ethereal.com/docs/user-guide>

저자소개



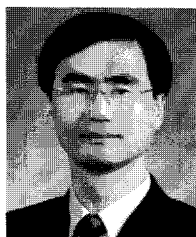
이 승 민 (Lee, Seung-Min)

2000년 : 광운대학교 전자통신공학과 학사
 2002년 : 광운대학교 전자통신공학과 석사
 2002년~현재 : 삼성전자 무선사업부 연구원
 관심분야 : 통신프로토콜, All-IP



민 상 원 (Min, Sang-Won)

1988년 : 광운대학교 전자통신공학과 학사
 1990년 : 한국과학기술원 전기및 전자공학과 석사
 1996년 : 한국과학기술원 전기및 전자공학과 박사
 1990년~1999년 : LG정보통신(현 LG-Nortel) 선임연구원
 1999년~현재 : 광운대학교 전자통신공학과 부교수
 관심분야 : 통신프로토콜, All-IP, 3G/4G 이동통신망, IPv6, ITS, 공학교육



김 용 진 (Kim, Yong-Jin)

1983년 : 연세대학교 전자공학과 학사
 1989년 : KAIST 전기 및 전자공학과 석사
 1997년 : KAIST 전기 및 전자공학과 박사
 1983년 ~ 2002년 : 한국전자통신연구원 팀장, 책임연구원
 2002년 ~ 현재 : 모다정보통신 CTO
 관심분야 : 통신 프로토콜, IPv6, All-IP, 3G/4G 이동통신망