

## 리눅스 기반 ISATAP 연동 메커니즘 설계 및 구현

이상도\*, 신명기\*, 김형준\*  
\*한국전자통신연구원 표준연구센터  
e-mail: {sdlee,mkshin,hjkim}@etri.re.kr

### Implementaion of ISTAP Mechanism based on linux

SangDo Lee\*, Myung-Ki Shin\*, Hyoung-Jun Kim\*  
\*Electronics and Telecommunications Research Institutue  
Protocol Engineering Center

#### 요 약

ISATAP 은 IPv4 기반의 거대한 인터넷이 주류를 이루고 있는 IPv6 초기 도입기에 IPv4 기반 망 내의 IPv6 고립된 단말이 ISATAP 라우터와의 ISATAP 터널링을 통해서 IPv6 망으로 쉽게 접속할 수 있도록 지원함으로써 IPv4 망의 점진적인 IPv6 망으로의 전환을 지원할 수 있는 기술로 활용될 것으로 기대되는 메커니즘 중에 하나이다. 이 기술은 국제 민간 표준화 기구인 IETF 의 NGTrans 워킹 그룹에 의하여 표준화중인 IPv6 전환 기술 중에 하나이며 독립적으로 구성된 IPv6 호스트 IPv4 망에서 통신을 하는 경우에 널리 사용될 것으로 예상된다. 본 논문은 ISATAP 기술을 리눅스 환경에서 구현한 소프트웨어 모듈에 관한 설계 및 운용에 관한 것이다.

#### I. 서론

인터넷은 사용자의 급속한 증가로 인해 주소고갈 문제에 직면하고 있으며, 새롭게 등장하고 있는 휴대 인터넷, 홈 네트워크 등의 신규 서비스들은 인터넷 주소 고갈을 더욱 앞당길 것으로 예상된다. 이러한 주소 부족문제를 해결하기위해서 IETF 는 IPv6 (Internet Protocol version 6)를 개발하였고 IPv6 이제 그 표준화가 완료되어 도입기에 들어서고 있다[1].

IPv6 는 128 비트 주소 체계를 제공하여 거의 무한한 주소공간을 제공할 수 있다. 또한 단순화된 헤더, 주소 자동설정, 효율적인 이동성 지원 등 다양한 장점들로 인하여하여 기존 인터넷의 기반이 되는 인터넷 프로토콜인 IPv4 를 대체할 차세대 인터넷 프로토콜로 인정받고 있다. 또한, IPv6 는 인터넷의 영역을 사물들에까지 확대하는 차세대 통신패러다임으로 대두되고 있는 유비쿼터스 네트워크를 실현 시킬 기반 기술로서도 그 중요성을 가진다. 그러나 IPv6 는 IPv4 와 호환되지 않고 또한 이미 전 세계를 아우르는 거대한 인터넷이 IPv4 에 기반을 두어 운영되고 있어 점진적

인 IPv6 로의 전환이 불가피하다. 이는 상당 기간 동안 IPv4 망과 IPv6 망이 공존하게 됨을 의미한다. 따라서 IPv6 망과 IPv4 망사이의 투명한 연동을 지원하는 IPv6 전환기술의 개발 및 지원이 주요한 이슈로 대두되었고 IETF 의 Ngrans WG 을 중심으로 다양한 연동기술이 개발되었다[2][3][4]. 이 중에서 기존의 IPv4 네트워크를 이용하여 새로 구성된 IPv6 사이트간의 통신이 필요하게 될 때 이를 자동 터널링 방식을 이용하여 쉽게 통신할 수 있는 ISATAP 메커니즘에 대한 연구 및 ISATAP 메커니즘을 기반으로 하여 리눅스 상에서 구현한 설계 구조 및 운용 방식에 대해서 기술하고자 한다.

#### II. 관련연구

##### 2.1 ISATAP 의 개요

ISATAP(Intra-Site Automatic Tunnel Addressing Protocol)은 주로 IPv4 기반의 인터넷에서 IPv6 노드를 점증적으로 배치할 수 있는 간단하고 확장성 있는

방법을 제공한다. ISATAP 은 64 비트 EUI-64 인터페이스 식별자와 표준 64 비트 IPv6 주소 프리픽스를 포함하는 통합 가능 글로벌 유니캐스트 주소 포맷을 기반으로 한다. 이 접근법은 IPv6 게이트웨이와 공통 데이터 링크를 공유하지 않는 듀얼 스택 노드가 사이트 내에서 IPv4 라우팅 인프라를 통해 IPv6 메시지를 자동으로 터널링함으로써, 글로벌 IPv6 네트워크에 결합할 수 있도록 한다. ISATAP 주소 자동설정을 위하여 두 가지 오프링크 IPv6 게이트웨이의 자동 탐색 방법이 제공된다. 이 접근법에서는 전체 사이트에서 하나의 IPv6 주소 프리픽스가 사용되므로, 보다 게이트웨이에서 통합 확장 문제없이 대규모 사이트 내 배치가 가능하다[5].

64bits	32bits	32bits
prefix	000(0/2):5EFE	IPv4 Address

(그림 1) ISATAP 의 주소 구조

(\*참고: ISATAP 주소 인터페이스 식별자는 /64 ISATAP 프리픽스의 로컬 범주 내에서만 해석되므로, 로컬 범주를 나타내기 위해 가장 중요도가 낮은 OUI 옥텟의 u/1 비트를 '0'으로 설정한다.)

ISATAP 에 의한 IPv6-in-IPv4 터널링에 의해 IPv4 망내의 IPv6 호스트에서 다른 IPv6 망으로 향하는 IPv6 패킷들은 ISATAP 라우터에서 IPv4 프로토콜 타입 41 을 가지는 IPv4 패킷들(RFC 791) 패킷들로 전송되며, IPv4 헤더는 목적지 주소와 송신측 주소를 포함한다. 이들 중 하나는 위에서 언급한 것처럼 구성된 IPv6 프리픽스의 V4ADDR 필드와 동일할 것이다. IPv4 패킷의 Payload 는 IPv6 헤더와 Payload 를 포함한다. (그림 2)는 ISATAP 에 의한 IPv6-in-IPv4 터널패킷의 형식을 보여준다. IPv4 TTL 은 캡슐화된 IPv6 Hop limit 이 될 일반값으로 설정될 것이다.

Version	HL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live	Protocol 41		Header Checksum	
Source Address				
Destination Address				
Options			Padding	
IPv6 header and Payload .....				

(그림 2) ISATAP 에 의한 IPv6 in IPv4 Tunnel 패킷 형식

이러한 ISATAP 은 IPv4 기반의 거대한 인터넷이 주류를 이루고 있는 IPv6 초기 도입기에 IPv4 기반 망내의 IPv6 고립된 단말이 ISATAP 라우터와의 ISATAP 터널링을 통해서 IPv6 망으로 쉽게 접속할 수 있도록

지원함으로써 IPv4 망의 점진적인 IPv6 망으로의 전환을 지원할 수 있는 기술로 활용될 것으로 기대된다.

## 2.2 6TALK 에서의 ISATAP 의 구현구조 및 운용방법

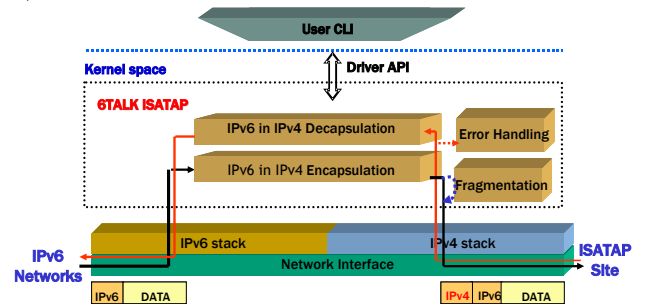
본 장에서는 리눅스 커널 하에서 구현된 ISATAP 의 구현 구조 및 ISATAP 의 운용방법에 관한 것이다.

### (1) 구현 구조

리눅스 기반 운영환경에서 ISATAP 라우터의 동작을 수행하는 커널레벨의 터널링 모듈 형태로 개발이 되었으며, 또한 개발된 모듈을 통해 ISATAP 터널링을 제어할 수 있는 텍스트 기반 CLI 를 제공하였다. 개발된 ISATAP 터널링 모듈은 리눅스 커널 2.4.20 기반 환경에서 독립모듈로 개발되며 텍스트기반 CLI 를 통해 제어되도록 구현하였으며 동적인 모듈 설치 제거가 가능하도록 설계하였다.

### (2) 전체 블록도

(그림 3)은 리눅스 커널 모듈로 구현된 ISATAP 의 구현 구조를 나타낸 그림이다. IPv4 망내의 IPv6 단말에서 ISATAP 라우터를 통하여 다른 IPv6 망으로 향하는 IPv6 패킷은 단말의 ISATAP 터널인터페이스를 통해서 IPv6-in-IPv4 터널 패킷으로 터널링되어 ISATAP 라우터로 전송된다. 이를 수행한 ISATAP 처리과정이 오른쪽에서 왼쪽으로 향하는 화살표로 나타나 있다. 또한 IPv6 망을 통해 수신되는 IPv6 패킷의 캡슐화 과정 및 ISATAP 터널을 통한 IPv4 망내의 IPv6 단말로의 전달과정을 왼쪽에서 오른쪽으로 향하는 화살표의 흐름으로 표시하고 있다. 이때 IPv6 패킷에 IPv4 터널 헤더가 추가되게 됨으로 패킷 크기가 증가되어 프래그멘테이션(Fragmentation)이 발생할 수 있다. 이러한 처리를 해당 처리 루틴이 담당하게 된다. ISATAP 은 리눅스 커널로 구현되어 있으며 관리자가 CLI(Command Line Interface)를 통한 터널생성 및 제거를 수행할 수 있도록 텍스트 기반의 CLI 를 제공하고 있다. 이러한 구현구조는 공개 리눅스 환경의 커널에서 이미 지원되는 기존의 SIT 인터페이스와 중첩됨으로 이 터널인터페이스를 사용하는 6to4 와 중첩되어 이를 방지하기 위해서는 linux/src/net/ipv6/af\_inet6.c 에서 sit\_init()와 sit\_cleanup() 모듈 부분을 코멘트 처리하여 커널을 재 컴파일 하는 것으로 이 문제를 해결하였다.



(그림 3) 리눅스 커널상의 6TALK 6to4 의 구현구조

(3) ISATAP 에서의 IPv6-in-IPv4 터널 패킷 수신 및 처리

```
ip6inip0 Link encap:IPIP Tunnel HWaddr
inet6 addr: 2001:230:1:4:0:5efe:81fe:90b1/64 Scope:Global
inet6 addr: fe80::5efe:81fe:90b1/64 Scope:Link
UP RUNNING NOARP MTU:1480 Metric:1
RX packets:4 errors:0 dropped:0 overruns:0 frame:0
TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:288 (288.0 b) TX bytes:300 (300.0 b)
```

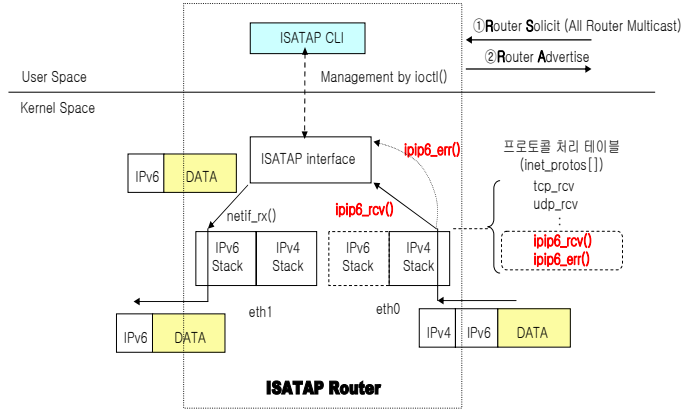
(그림 4) 생성된 ISATAP 인터페이스의 화면

(그림 4)는 리눅스 기반 ISATAP 인터페이스의 생성 정보를 보여준다. ISATAP 은 구동 시 ISATAP 단말들에 대한 터널링 서비스를 제공하는 IPv4 망 쪽 인터페이스에 설정된 IPv4 주소와 ISATAP 인터페이스에서 광고메시지(Router Advertisement)를 보내는 IPv6 Prefix 정보로 (그림 1)과 같은 형식의 ISATAP 주소를 ISATAP 인터페이스에 부여한다. 아래 그림에서는 ISATAP 인터페이스를 생성한 후의 시스템의 IPv6 라우팅 테이블의 예를 보여준다.

Destination	Next Hop	Flags	Metric	Ref	Use	Iface
::1/128	::	U	0	1	1	lo
::127.0.0.1/128	::	U	0	0	0	lo
::129.254.112.165/128	::	U	0	0	0	lo
::129.254.144.177/128	::	U	0	0	0	lo
::/96	::	U	256	0	0	ip6inip0
2001:230:1:1::/7/128	::	U	0	459	0	lo
2001:230:1:1::/64	::	UA	256	3	0	eth0
2001:230:1:4:0:5efe:81fe:90b1/128	::	U	0	0	0	lo
2001:230:1:4:201:2ff:fe96:cd3/128	2001:230:1:4:201:2ff:fe96:cd3	UAC	0	50	1	ip6inip0
2001:230:1:4::/64	::	UA	256	0	0	ip6inip0
3ffe::/16	2001:230:1:1::1	UG	1	0	0	eth0
2000::/3	2001:230:1:1::1	UG	1	576	0	eth0
fe80::5efe:81fe:90b1/128	::	U	0	0	0	lo
fe80::5efe:81fe:90b2/128	fe80::5efe:81fe:90b2	UAC	0	1	1	ip6inip0
fe80::204:75ff:fee2:2326/128	::	U	0	79	0	lo
fe80::250:daff:fe8e:2758/128	::	U	0	0	0	lo
fe80::/64	::	UA	256	0	0	ip6inip0
fe80::/10	::	UA	256	0	0	eth0
fe80::/10	::	UA	256	0	0	eth1
fe80::/10	::	UA	256	0	0	ip6inip0
ff00::/8	::	UA	256	0	0	eth0
ff00::/8	::	UA	256	0	0	eth1
ff00::/8	::	UA	256	0	0	ip6inip0
::/0	2001:230:1:1::1	UG	1	0	0	eth0

(그림 4) ISATAP 과 관련된 IPv6 라우팅 테이블의 예

위의 (그림 4)의 예에서는 ISATAP 인터페이스가 설정되어 ISATAP 라우터의 Forwarder 의 역할을 하는 시스템의 IPv6 라우팅 테이블의 내용을 보여주고 있다. eth0 인터페이스를 통해 IPv6 망으로 접속되어 있고 eth1 인터페이스가 ISATAP 서비스를 제공하는 망을 서비스하는 물리적 인터페이스이다. ISATAP 인터페이스인 ip6inip0 인터페이스 상으로 2001:230:1:4::/64 IPv6 Prefix 를 가지는 RA 가 뿌려지고 있다. 따라서 2001:230:1:4::/64 Prefix 를 가지는 트래픽의 출력 인터페이스가 ip6inip0 인터페이스로 지정되어 있음을 위의 라우팅 테이블의 예에서 확인할 수 있다.



(그림 5) ISATAP 에서의 IPv6-in-IPv4 터널 패킷의 수신 및 처리

(그림 5)는 리눅스 기반 ISATAP 모듈에서 수신된 IPv6-in-IPv4 터널 패킷의 처리과정을 보여준다. Ip6inip0 인터페이스는 IPv4 프로토콜로부터 IPv6-in-IPv4 터널 패킷을 전달받기 위해서 IPv4 프로토콜의 프로토콜처리 테이블에 IPv4 헤더의 프로토콜 타입이 IPv6 (41)일 경우 ipip6\_rcv() 함수로 전달되도록 함수 핸들러를 등록하고 IPv6-in-IPv4 터널 에러의 경우 처리함수로 ipip6\_err() 함수 핸들러를 등록하여 해당 이벤트의 처리가 이루어지도록 구현되었다. 이러한 프로토콜 처리 제공하는 inet\_add\_protocol()와 inet\_del\_protocol() API 함수를 통해서 이루어진다.

이러한 방식으로 시스템의 재실행 없이 isatap 인터페이스의 구동 및 제거가 가능하게 된다. IPv6-in-IPv4 터널 패킷이 수신되면 위에서 설명한 것처럼 IPv6-in-IPv4 터널패킷 처리함수인 ipip6\_rcv()로 전달되고 IPv6 in IPv4 터널헤더가 제거된 후 IPv6 프로토콜을 통해 IPv6 라우팅에 의해 IPv6 망의 목적지 단말로 전달된다. 이러한 방식으로 ISATAP 단말의 IPv6 망으로의 접속을 지원할 수 있게 된다.

ISATAP 터널링은 가장 널리 사용되고 있는 개인용 PC 의 운영체제인 윈도우 XP 에서 IPv6 프로토콜과 함께 지원된다. 따라서 IPv4 주소를 가지고 IPv4 인터넷을 사용하는 사용자 단말은 ISATAP Router 의 주소 정보와 함께 ISATAP 설정을 수행하면 ISATAP 터널링을 통해서 IPv6 망 접속을 할 수 있게 된다. ISATAP 단말로 동작하기 위해서 사용자는 먼저 IPv4 망 접속을 가능하게 하는 IPv4 주소가 설정되어 있어야 하며, "ipv6 rlu 2 129.254.144.177" 명령을 도스 명령어 창에서 수행하면 된다. 그러면 129.254.144.177 주소를 가지는 ISATAP 라우터로 ISATAP 터널된 Router Solicit 를 보내게 되고 이를 수신한 ISATAP 라우터는 해당 터널 패킷을 수신하여 해당단말로 IPv6 프리픽스를 지원하는 RA 메시지를 유니캐스트 통신으로 응답하게 된다. 위의 명령을 수행한 후의 윈도우 XP 단말 상의 IPv6 인터페이스상의 설정된 주소정보를 아래의 (그림 6)에서 보여주고 있다.

```
C:\Documents and Settings\Whclee>ip6 rlu 2 129.254.144.177
C:\Documents and Settings\Whclee>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix  . :
    IP Address. . . . . : 129.254.144.178
    Subnet Mask . . . . . : 255.255.255.240
    IP Address. . . . . : fe80::201:2ff:fe96:cdd3%4
    Default Gateway . . . . . : 129.254.144.177

Tunnel adapter 6to4 Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix  . :
    IP Address. . . . . : 2002:81fe:90b2::81fe:90b2
    Default Gateway . . . . . : 2002:836b:213c::836b:213c
                                2002:c058:6301::c058:6301

Tunnel adapter Automatic Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix  . :
    IP Address. . . . . : 2001:230:1:4:0:5efe:129.254.144.178
    IP Address. . . . . : fe80::5efe:129.254.144.178%2
    Default Gateway . . . . . : fe80::5efe:129.254.144.177%2

C:\Documents and Settings\Whclee>
```

(그림 6)ISATAP 윈도우 단말에서의 IPv6 주소설정

아래 (그림 7)은 ISATAP 단말에서 IPv6 인터넷상의 www.kame.net 사이트로의 ping 의 수행 예를 보여주고 있다. ISATAP 인터페이스에

2001:230:1:4:0:5efe:129.254.144.178 주소가 부여 되어 있다.

```
C:\Documents and Settings\Whclee>ping www.kame.net

Pinging orange.kame.net [2001:200:0:8002:203:47ff:fea5:3085] with 32 bytes of data:

Reply from 2001:200:0:8002:203:47ff:fea5:3085: time=70ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085: time=58ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085: time=76ms
Reply from 2001:200:0:8002:203:47ff:fea5:3085: time=55ms

Ping statistics for 2001:200:0:8002:203:47ff:fea5:3085:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 55ms, Maximum = 76ms, Average = 64ms

C:\Documents and Settings\Whclee>
```

(그림 7)ISATAP 윈도우 단말에서의 IPv6 사이트 접속통신 예

(4) ISATAP 에서의 IPv6 패킷 수신 및 IPv6-in-IPv4 터널링

ISATAP 인터페이스가 활성화되면 ip6inip0 라는 이름의 IPv6-in-IPv4 IPv6 터널인터페이스가 (그림 6)과 같이 생성된다. 이러한 인터페이스의 전송 함수로 ipip6\_tunnel\_xmit()함수가 등록되고 터널생성함수, 터널인터페이스에 대한 ioctl 를 수행하는 함수, 터널제거 함수등이 등록되어 각각의 이벤트시 호출되게 된다. IPv6 단말로부터 ISATAP 단말로 전송되어야 할 IPv6

패킷이 isatap 라우터에 수신되면 IPv6 라우팅에 의해서 ip6inip0 인터페이스로 전달되게 되고 ipip6\_tunnel\_xmit()함수가 이를 수신하여 IPv6 in IPv4 터널링을 수행후 IPv4 프로토콜에 의해 처리되도록 IPv4 프로토콜로 전달하게 된다

(5) ISATAP 의 사용자 인터페이스

ISATAP 은 관리자에게 웹 CGI 모드 기반 제어 인터페이스 및 텍스트 기반 CLI 를 제공한다. 이러한 인터페이스 상의 명령들에 대한 ISATAP 인터페이스로의 전달은 (그림 8)에서도 보여주듯이 ioctl 를 통해 이루어진다. ioctl 코드를 기반의 ISATAP 이 관리자에게 지원하는 명령어와 그 설명은 아래 테이블 참고

명령어	기능	구현방법
down	인터페이스삭제	'ip6inipctl ip6inip0 off' 'ifconfig ip6inip0 down' 'rmdir ip6inip'
nodown ip_addrv4 addrv6/prefix_len	인터페이스생성 및 구동	'insmod ip6inip.o' 'echo "1">/proc/sys/net/ipv4/conf/all/forwarding' 'ip6inipctl ip6inip0 isatap 129.254.144.177' 'ifconfig ip6inip0 up' 'ifconfig ip6inip0 add 2001:230:1:4:0:5efe:129.254.144.177/64' 'ifconfig ip6inip0 add fe80::5efe:129.254.144.177/64' 'echo "1">/proc/sys/net/ipv6/conf/all/forwarding'

(그림 8) ISATAP 제어 명령어와 그 구현 방식

III. 결론

본 논문에서는 ISATAP 연동 메커니즘을 구현하기 위한 리눅스 상에서 설계 및 구현구조에 대해서 살펴 보았으며, 그 동작 성을 확인하기 위한 테스트를 수행 하였다. 이러한 리눅스 기반의 네트워크 모듈 방식은 응용하여 터널 링을 기본으로 하는 여러 연동 메커니즘을 손쉽게 구현할 수 있을 것으로 기대한다. 향후 구현한 ISTAP 메커니즘을 이용하여 IPv6-in-IPv4 터널링 방식을 이용한 성능에 관한 연구를 진행할 예정이다.

[참 고 문 헌]

[1] Deering, S., and R. Hinden, Editors, "Internet Protocol Version 6 (IPv6) Specification", RFC 1883, December 1995.  
 [2] B. Carpenter, K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001  
 [3] G. Tsirtsis, P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000  
 [4] R. Gilligan, E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 2893, August 2000  
 [5] F. Templin, T. Gleeson, M. Talwar, D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", draft-ietf-ngtrans-isatap-24.txt