

IPv6 특징

- ▣ 128bit 주소 제공
- ▣ IP주소 지정 자동 구성
- ▣ 3가지 주소 타입
 - Unicast, Anycast, Multicast
- ▣ 확장된 헤더에 선택사항 기술 가능, 수신지에서만 검색되어 네트워크 속도 전반적 향상
 - Packet 출처 인증, 강화된 보안
- ▣ IHL이 40byte로 고정
 - 헤더 프로세싱의 SW적 최적화 용이
- ▣ 주소 지정시 NAT 불필요, Address Renumbering 지원, Plug&Play 제공

IPv4 & IPv6 주소 특징

구분	IPv4	IPv6
길이	32Bit	128Bit
IP 수	약 43억개	약 1조개
자동 구성	DHCP, Zero 구성	NonServer, Renumbering ...
보안	IPSec	IPSec 인증(기본)
이동성	수동	자동
Multicast	PIM/Multicast BGP	PIM/Multicast BGP
Header Checksum	필요	불필요
Fragment Infor.	전체	옵션
Flow Label	No	Yes
Class	Classfull & Classless	Classless Only
QoS	QoS 보장에 어려움	QoS 보장이 용이

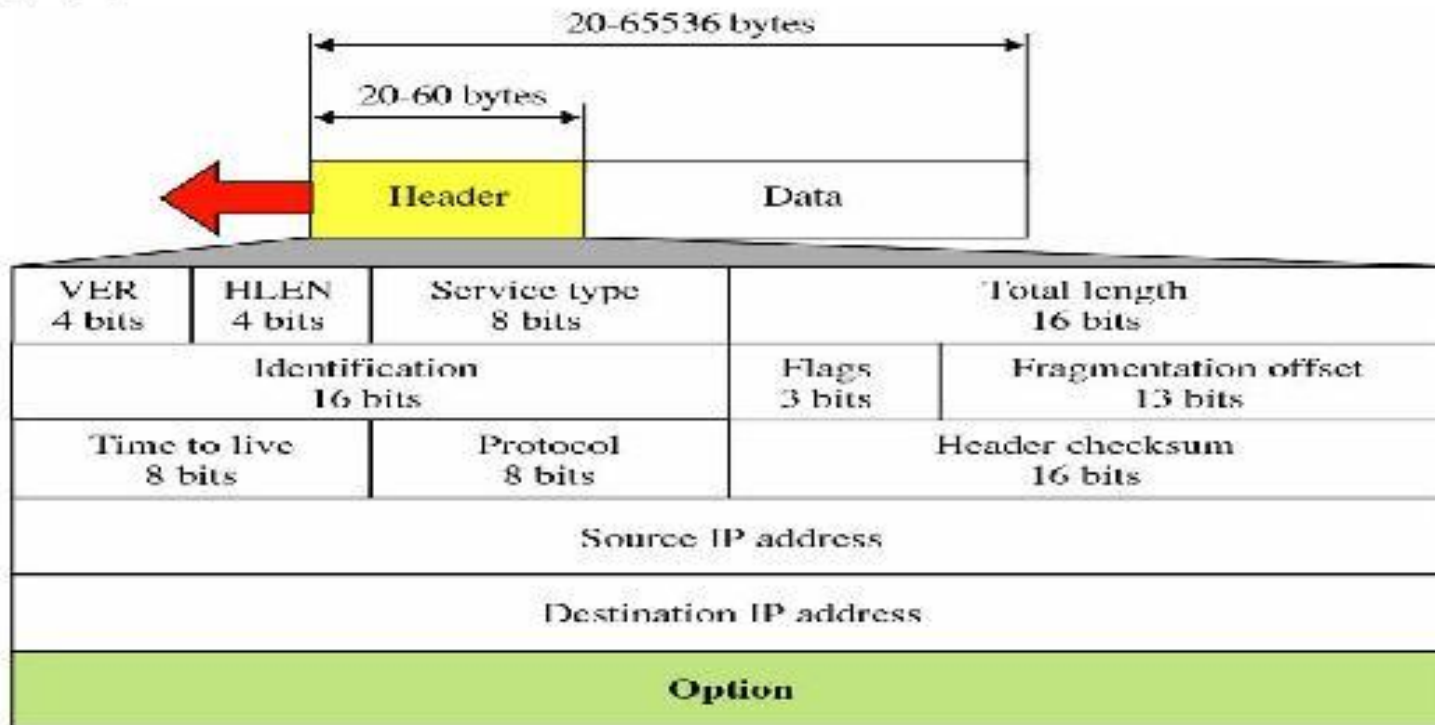


IPv6 주소 체계

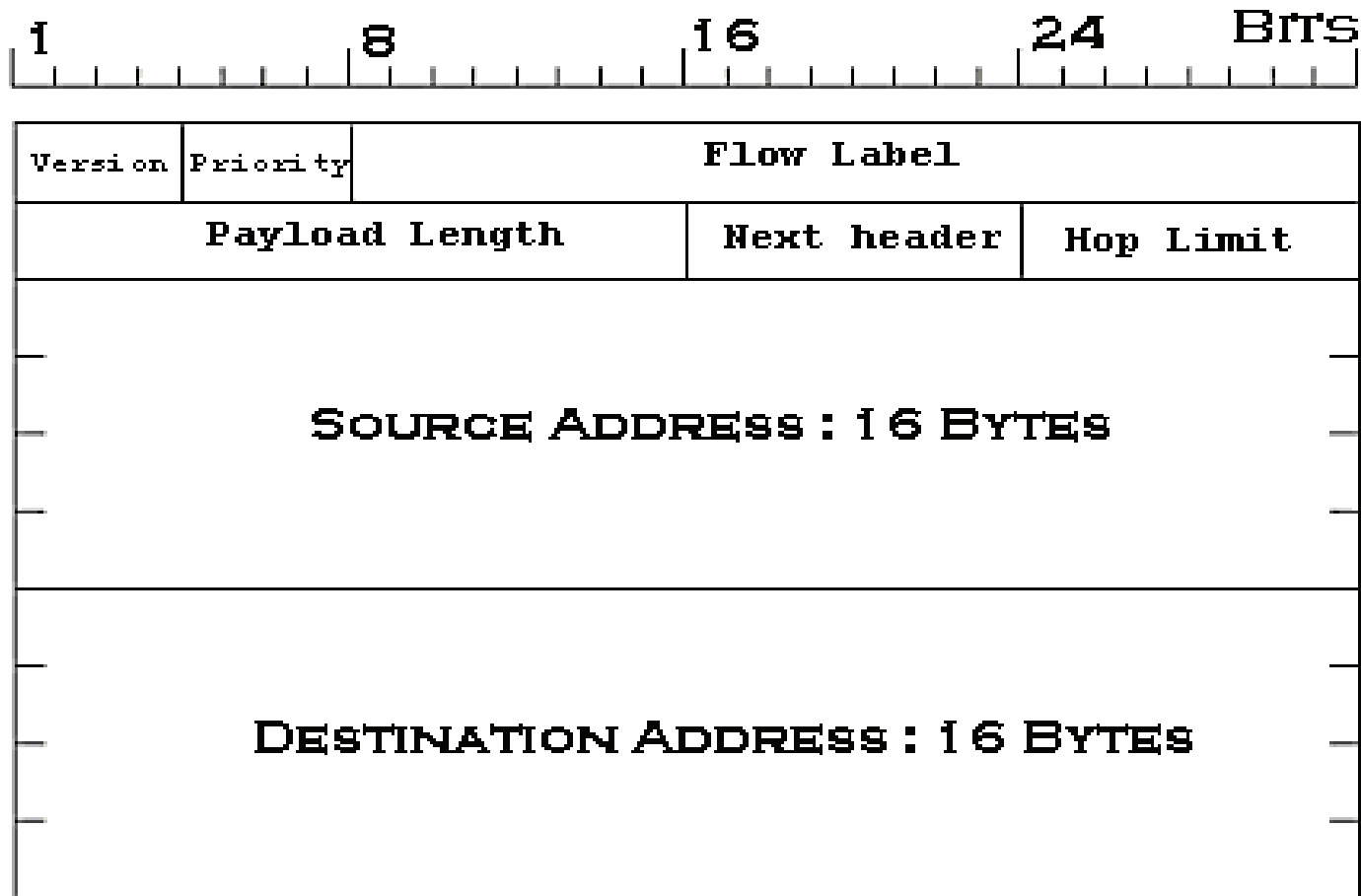
- ▣ 8구간 Part 구분
- ▣ 각 숫자는 0 - F사이 16진수로 표시
- ▣ Part는 점이 아닌 Colon으로 구분
- ▣ 최대 서브넷은 128Bit

IPv4 Header

IP 헤더



IPv6 Header



IPv6 Address 타입

- **Unicast**
 - 1:1 방식

- **Multicast**
 - 1:多 방식

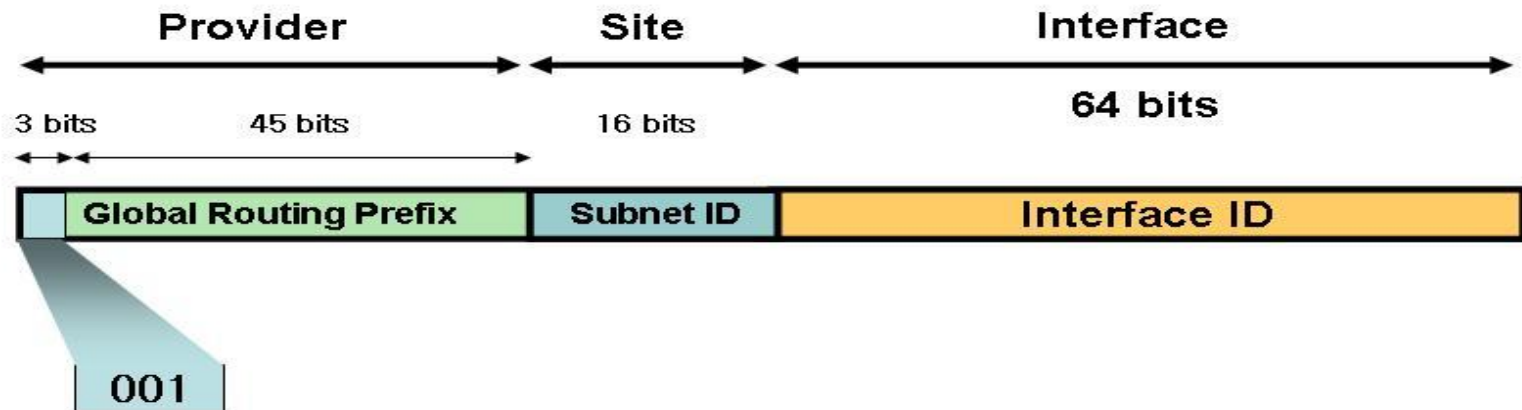
- **Anycast**
 - 1:近 방식
 - 여러 개의 네트워크 장비가 동일한 한 개의 주소를 공유
 - 라우터는 목적지로부터 가장 가까운 장비를 선택하여 부하분산을 도모하는 구조

IPv6 Address 종류

- **Global Address**
 - 2001::/16, 2002::/16, 2003::/18, 3FFE::/16
 - 자동/수동 설정, 인터넷 통신 가능한 가장 일반적 IPv6주소
- **Link Local Address**
 - FE80::/10
 - 자동 설정, Local Link에서만 사용
- **Site Local Address**
 - FEC0::/10
 - IPv4의 사설 IP와 유사
- **Multicast Address**
 - FF02::1
 - 항상 자동 설정, 모든 Host를 대상으로 하는 멀티캐스트 주소
- **Multicast Address**
 - FF02::1:FFxx:xxxx
 - 항상 자동 설정, 자신에 대한 멀티캐스트 주소(Solicited-Node)

Global Unicast Address

- 인터넷 사용을 위한 주소
- 3부분
 - Provider 영역: 48Bit
 - Site 영역: 16Bit



IPv6 Address 규칙

- **기본 표기 방식**
 - 16진수 4개씩 8번, 중간에 콜론 7개 찍어 주소 표기
 - XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
- **규칙 1**
 - A~F 알파벳은 대소문자 구분 없다.
 - 2001:0DB8:010F:0001:0000:0000:0000:0D0C
- **규칙 2**
 - 앞쪽에 오는 '0'은 쓰지 않아도 된다.
 - 2001:DB8:10F:1:0:0:0:D0C
- **규칙 3**
 - 0이 연속으로 나올 때는 ::로 표시할 수 있다.
 - 2001:DB8:10F:1::D0C

IPv6 Address 입력/확인

□ 주소 입력

- IPv4 주소 입력과 유사(단 Prefix는 '/64'와 같은 형식)
- 수동 입력 방식
 - R(config-if)# **ipv6 address** 2003:1:1:12::1/64
- EUI-64 방식
 - R(config-if)# **ipv6 address** 2003:1:1:12::/64 **eui-64**

□ 주소 확인

- 주소 입력 확인
 - R# **show ipv6 interface brief**
- 자동 입력
 - R# ping 상대방_IPv6_주소

자동 주소 부여

▣ Stateless Auto Configuration

- DHCP와 같은 별도의 설정이 없더라도 라우터와의 상호작용으로 자동으로 글로벌 주소를 획득할 수 있도록 하는 과정

▣ 자동 주소 부여 설정

- R(config-if)# **ipv6 address autoconfig**
 - ▣ 해당 인터페이스는 호스트에서의 인터페이스 특성을 가지게 된다.
 - ▣ 자동으로 링크로컬주소를 생성, 이를 바탕으로 라우터와 통신을 시도한다.

자동주소부여 3가지 과정

▣ DAD(Duplicate Address Detection)

- 처음 인터페이스가 활성화된 호스트는 자신과 인접한 라우터의 존재 유무를 확인하고 해당 라우터와 통신하기 위해 스스로 링크로컬주소를 생성한다.
- 이때 호스트는 링크상에서 자신이 사용할 주소를 이미 사용하고 있는 장비가 존재하는 여부를 확인한다.

▣ RS(Router Solicitation) 메시지 전송

- 자신이 사용할 글로벌주소를 생성하기 위해 호스트는 현재 라우터가 사용하고 있는 글로벌 주소의 네트워크 대역이 무엇인지 알려줄 것을 요청한다.

▣ RA(Router Advertisement) 메시지 전송

- RS 메시지를 받은 라우터는 자신의 네트워크 대역을 포함한 여러 정보를 알려준다.

IPv6 Routing Protocol

IPv4	IPv6
RIP	RIPng
OSPF(v2)	OSPFv3
ISIS	ISIS
BGPv4	BGPv4+

- ▣ **IPv6 Routing 실시**
 - R(config)# **ipv6 unicast-routing**

IPv6 Static Route

▣ Static Route 설정

- R(config)# **ipv6 route** 네트워크/프리픽스 [Next-Hop-IP | 인터페이스]
- Next-Hop-IP는 Link Local 주소를 설정하는 게 유리

▣ Default Route 설정

- R(cfg)# **ipv6 route ::/0** [출구인터페이스 | Next-Hop-IP]

▣ Routing Table 확인

- R# **show ipv6 route static**

IPv6 ACL

□ IPv6 ACL 특징

- Source와 Destination 주소 기반의 필터링
- 인터페이스에서 Inbound 또는 Outbound 필터링 적용
- 명시적 조건이 없을 때는 "Deny Any Any"
- Prefix 단위의 필터링

□ IPv6 ACL 예

- R(config)# **ipv6 access-list** BLK **deny** fec0:0:0:2::/64 any
- R(config)# **ipv6 access-list** BLK **permit any any**
- R(config)# interface f0/0
- R(config-if)# **ipv6 traffic-filter** BLK [out | in]

IPv6 Prefix List & Route Map

□ IPv6 Prefix List

- IPv4에서와 동일한 방식으로 설정
- R(config)# **ipv6 prefix-list** PRE **permit** 2001:1:1::/48

□ IPv6 Route Map

- IPv4 프로토콜과 IPv6 프로토콜에 대한 Route Map 구분이 없다.
- R(config)# **route-map** MYMAP
- R(config-route-map)# **match** ipv6 address V6ACL
- R(config-route-map)# **set** ipv6 next-hop fec0:1:1:12::2



RIPng

- **RIPng**
 - RIP Next Generation 약자
 - IPv6용으로 개발된 RIP 새 버전

- **IPv6 RIP 설정**
 - R(config-if)# **ipv6 router rip** 이름
 - RIPng는 따로 프로세스를 만들지 않아도 됨
 - 인터페이스에 RIPng 설정 순간 설정, RIPng 프로세스가 자동 생성
 - RIPng 프로세스명은 숫자나 알파벳 모두 가능
 - 모든 인터페이스에 적용된 RIPng 프로세스명은 동일해야 한다.

- **IPv6 RIP 설정 상태 확인**
 - R# **show ipv6 rip**

RIPng Default & Summary 설정

- **Default Route 설정**
 - R(config-if)# ipv6 rip CISCO **default-information originate**
- **Summary 설정**
 - Auto Summary 지원하지 않음
 - R(config-if)# ipv6 rip CISCO **summary-address** 2001::/16
- **IPv6 RIP Routing Table**
 - R# **show ipv6 route rip**
- **Debugging**
 - R# **debug ipv6 rip** 인터페이스_명

Distribute & Offset List 설정

□ Distribute List

- 경로 필터링 용도
- 기존 IPv4와 동일
- IPv6 Prefix List와 함께 사용

□ Offset List

- 경로 조정 용도
- R(config-if)# ipv6 rip CISCO **metric-offset** 3
- Inbound 정책으로만 사용

OSPFv3

- ▣ OSPFv2와 OSPFv3는 기본적으로 원리, 설정이 유사
- ▣ OSPFv3에서 router-id는 반드시 IPv4 포맷으로 설정
 - R(config)# **ipv6 router ospf 1**
 - R(config-rtr)# **router-id 1.1.1.1**
- ▣ OSPFv3도 네트워크 타입이 중요
- ▣ 네트워크 광고는 RIPng와 같이 인터페이스 광고
 - R(config-if)# **ipv6 ospf 1 area 0**

OSPFv3 Default & Summary 설정

- **Default Route**
 - R(config-rtr)# **default-information originate always**
- **ABR Summary**
 - R(config-rtr)# **area N range 2001::/16**
- **ASBR Summary**
 - R(config-rtr)# **summary-prefix 2005::/16**

NBMA Hub & Spoke

▣ F/R Mapping

- IPv6 Global Address뿐만 아니라 Link Local Address도 반드시 동일하고 매핑해줘야 한다.

▣ Hub Router: Serial 인터페이스 설정

- R(cfg)# int s0/0
- R(config-if)# ipv6 address ...
- R(config-if)# **frame-relay map ipv6 ... DLCI_값**
- R(config-if)# **frame-relay map ipv6 FE80::... DLCI_값 broadcast**
- R(config-if)# **ipv6 ospf neighbor FE80::...**
- R(config-if)# ipv6 ospf 1 area 0

Redistribution

□ 기본 명령

- R(config-rtr)# **redistribute** 프로토콜 ...
- 기존 IPv4에서 사용하던 재분배와 비교하면 주소 형식만 차이날 뿐 크게 다르지 않다.
- RIP 프로세스끼리도 재분배할 수 있다.

□ 재분배 예

- R(config)# ipv6 router rip T1
- R(config-rtr)# redistribute rip T2 metric-3 route-map TAG

IPv4/IPv6 전이 방법

▣ Dual Stack

- IPv4 프로토콜과 IPv6 프로토콜을 동시에 사용하는 것

▣ Translation 기술

- IPv4 트래픽을 IPv6 트래픽으로 변환하는 기술
- 아직까지 완벽한 변환기술이 개발되지 않은 실정

▣ Tunneling

- IPv4 네트워크에 의해 서로 단절된 IPv6 네트워크간의 통신을 위해 사용되는 기술

IPv4/IPv6 Protocol 변환방식 종류

- **SITT(Stateless IP/ICMP Translation)**
 - IP 헤더와 ICMP 정보를 IPv4 혹은 IPv6 버전으로 변환하는 방식
 - 패킷의 헤더 즉, 네트워크 계층 정보를 직접 변환하기 때문에 프로토콜 변환 속도가 빠르고 간편
 - 데이터에 포함된 IP 주소 정보는 변환할 수 없다.
- **TRT(Transport Relay Translator)**
 - 네트워크 계층 정보를 변환하지 않고 상위 계층인 전송 계층의 정보를 변환하는 방식
 - 다른 IP 버전간의 통신을 중계할 수 있는 TRT 서버가 필요
 - DNS와 FTP 등의 통신시, 데이터 부분의 주소 정보는 여전히 변환 불능
- **ALG(Application Level Gateway)**
 - 데이터에 포함된 IP 주소 정보를 변환해주는 역할

NAT-PT

□ NAT-PT

- IPv4와 IPv6 네트워크간 통신을 위해 미리 설정해둔 매핑 정보를 기반으로 패킷의 IP 헤더와 ICMP 정보를 IPv4 혹은 IPv6 버전으로 변환
- **NAT-PT Prefix**: 변환하고자 하는 IPv6 패킷의 목적지 주소 범위

□ NAT vs. NAT-PT 차이점

- NAT: 패킷이 사설 네트워크에서 외부 인터넷으로 나갈 때 사용하는 출발지 주소만 공인 IPv4 주소로 변환하면 된다.
- NAT-PT: 패킷을 IPv4 네트워크에서 IPv6 네트워크로 전송하기 위해 헤더 정보를 모두 변환해야 하므로, 출발지와 목적지 주소에 대한 매핑 정보가 모두 전조해야 한다.

NAT-PT 종류

- ▣ **Static NAT-PT**
 - 변환되기 전 주소-변환될 주소를 1:1로 정한다.
- ▣ **Dynamic NAT-PT**
 - 변환되기 전 주소-변환될 주소 간의 매핑을 미리 지정된 주소 범위내에서 자동으로 이루어지게 한다.
- ▣ **PAT-PT**
 - 여러 개의 IPv6 주소들을 TCP와 UDP 포트번호를 통해 구분함으로써 최대 64000개의 IPv6 주소를 하나의 IPv4 주소로 변환할 수 있다.

NAT-PT 장단점

□ NAT-PT 장점

- IPv4와 IPv6간 통신을 위한 프로토콜 변환시 그 속도가 다른 방식들보다 빠르고 간편하다.
- IPv4 NAT의 확장이라 할 수 있는 기술이므로 익숙하다.

□ NAT-PT 단점

- IP 헤더 옵션과 ICMP 메시지 타입의 완벽한 호환 불가
 - IPv4와 IPv6 헤더간에는 서로 존재하지 않는 옵션들이 있으며, IPv6에는 확장 헤더라는 개념이 존재한다.
- 데이터에 존재하는 IP 주소 정보 변환 불가
 - 패킷의 헤더에 존재하는 주소 정보를 변환할 수 있지만, 데이터에 존재하는 IP 주소 정보는 변환하지 못한다.

Static NAT-PT 설정

▣ NAT-PT Prefix 지정

- R(config)# **ipv6 nat prefix** 2001:2:3:4:5:6::/96
- 항상 96비트로 정해져 있다.

▣ 출발지 주소에 대한 매핑

- R(config)# **ipv6 nat v6v4 source** 변환전_V6주소 변환후_V4주소

▣ 목적지 주소에 대한 매핑

- R(config)# **ipv6 nat v4v6 source** 변환전_V4주소 변환후_V6주소

▣ 인터페이스 설정

- R(config-if)# **ipv6 nat**

Dynamic NAT-PT 설정

- ▣ **NAT-PT Prefix 지정**
 - R(config)# **ipv6 nat prefix** 2001:2:3:4:5:6::/96
 - 항상 96비트로 정해져 있다.
- ▣ **IPv6 패킷의 출발지 주소에 대해 매핑**
 - R(config)# **ipv6 nat source list** 변환전_주소집합 **pool** 변환후_주소집합
- ▣ **변환전 IPv6 주소 집합**
 - R(config)# **ipv6 access-list** 변환전_집합
- ▣ **변환후 IPv4주소 집합**
 - R(config)# **ipv6 nat v6v4 pool** 변환후집합 시작 끝_주소 prefix-length 24
- ▣ **IPv6 패킷의 목적지 주소에 대한 매핑**
 - R(config)# **ipv6 nat v4v6 source** 버전4주소 버전6주소
- ▣ **인터페이스 설정**
 - R(config-if)# **ipv6 nat**

Dynamic NAT-PT 동작 확인

- ▣ **Debug 명령**
 - R# debug ipv6 nat

- ▣ **NAT-PT 매핑 테이블**
 - R# show ipv6 nat translation

- ▣ **NAT-PT 전반적 상태 확인**
 - R# show ipv6 nat statistics

V4-Mapped 옵션

- ▣ **V4-Mapped 옵션 사용할 경우**
 - NAT-PT 장비는 목적지 IPv6 주소 하위 32비트 부분에서 IPv4 주소를 추출하여 이를 주소 매핑에 사용하기 때문에 목적지 IPv6 주소에 대한 매핑을 할 필요가 없다.

- ▣ **V4-Mapped 옵션 설정**
 - R(config)# ipv6 nat prefix ::ffff:0:0/96 mapped 변환전_주소집합
 - 상위 80비트는 모두 0, 다음의 16비트는 FFFF, 하위 32비트는 IPv4 주소를 포함

Tunneling 기술

▣ Tunneling

- 장비간에 가상의 경로를 구성하여 두 장비 사이에 여러 장비들이 존재 하더라도 마치 서로가 직접 연결되어 있는 것처럼 만들어 주는 기술

▣ Tunneling 기술의 단점

- 장비가 지원하는 프로토콜의 문제
- 보안상의 문제
- 패킷의 Fragment 문제
- PAT가 동작되는 네트워크 환경에서의 터널링 기술



Tunneling 기술

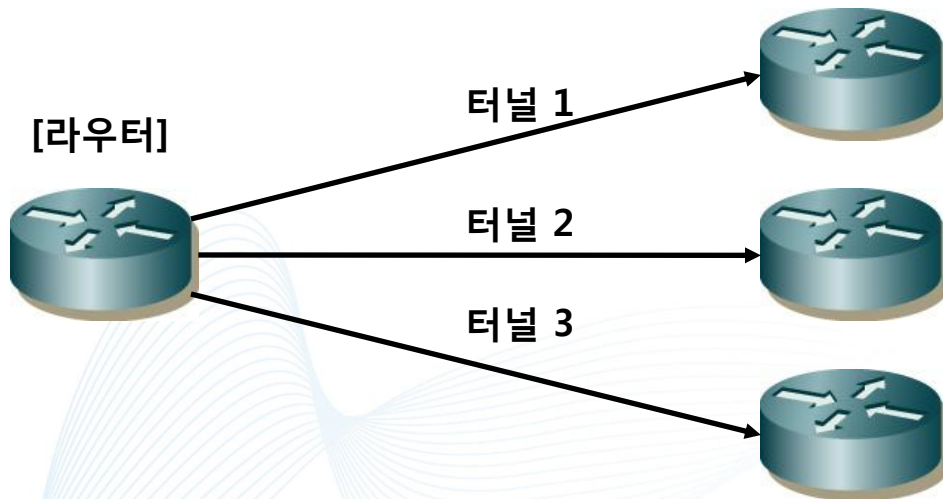
- ▣ **수동 터널링**
 - IPv6 over IPv4
 - GRE over IPv4

- ▣ **자동 터널링**
 - IPv4 호환 주소 기반 터널
 - 6to4 터널
 - ISATAP 터널
 - Teredo 터널

Tunneling 기술 동작

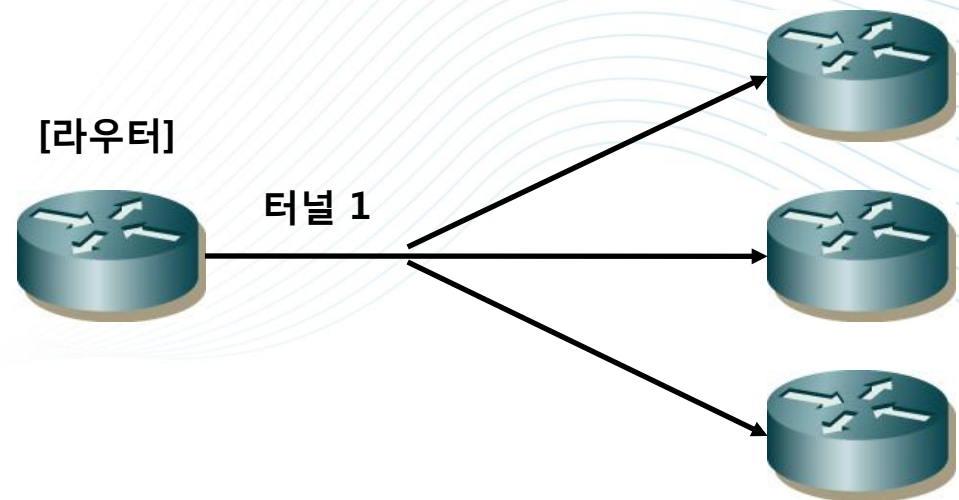
수동 터널링

- 터널 설정시 출발지와 목적지 주소를 일일이 지정해주는 기술



자동 터널링

- 하나의 터널 인터페이스만으로 여러 목적지와 통신이 가능





IPv4 호환주소 기반 터널

▣ IPv4 호환주소 기반 터널

- IPv4 네트워크에 의해 단절된 IPv6 장비간 통신을 위해 사용되는 자동 터널링 기술 중 하나

▣ 장점

- 간단한 설정만으로도 IPv4 네트워크 환경에서 자동 터널링 기술을 구현할 수 있다.

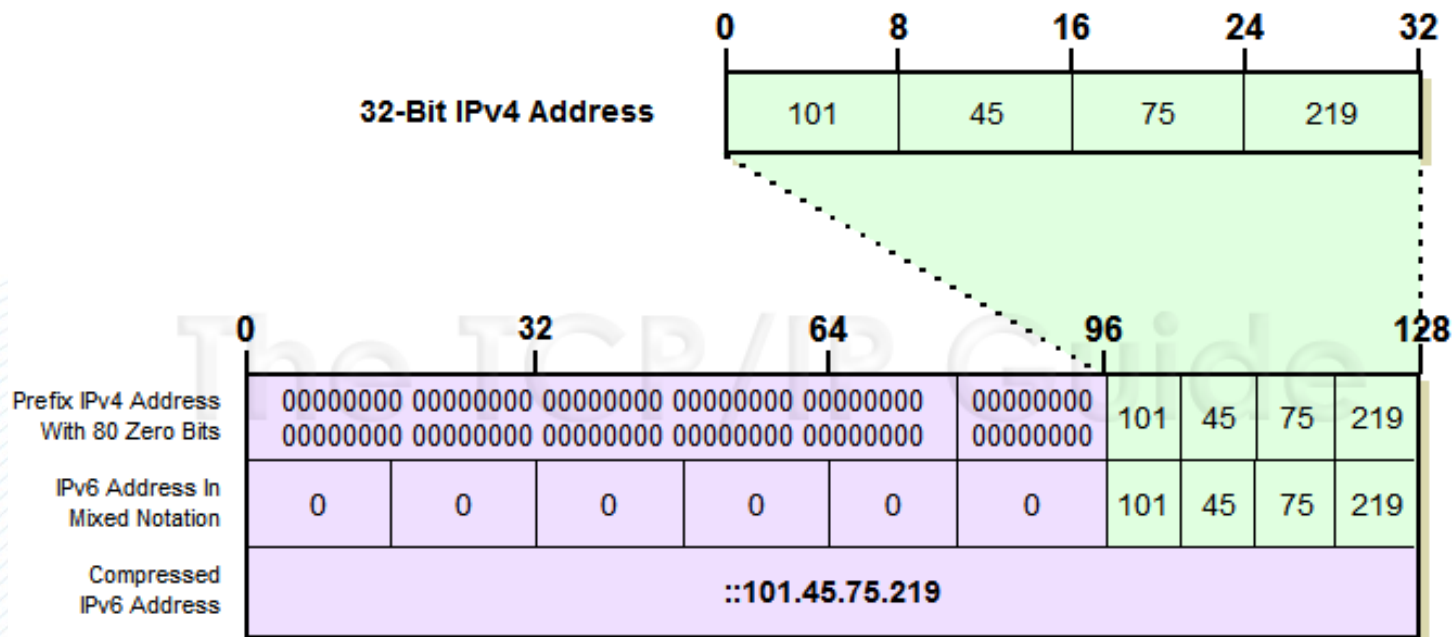
▣ 단점

- 인터넷 상의 장비와 IPv6 통신이 가능하려면 반드시 공인 IPv4 주소를 사용하여 터널을 설정해야 한다.

IPv4 호환주소

□ IPv4 호환주소 구성

- IPv4 주소를 바탕으로 생성
- 포맷 - ::IPv4 주소/96, 하위 32비트 제외한 나머지 비트들은 모두 0



IPv4 호환주소 기반 터널 설정

▣ 터널 인터페이스 설정

- R(config-if)# **tunnel mode ipv6ip auto-tunnel**
- R(config-if)# tunnel source lo0
- 간단한 설정만으로도 IPv4 네트워크 환경에서 자동 터널링 기술을 구현할 수 있다.
- 터널의 목적지 IPv4 주소는 지정하지 않는다.
 - IPv4 호환 주소 기반 터널은 목적지 IPv6 주소에서 터널의 목적지 IPv4 주소를 추출하여 패킷을 인캡슐레이션하기 때문

6to4 Tunnel

□ 6to4 Tunnel

- 6to4주소를 기반으로 동작되는 터널
- 목적지 IPv6 주소에서 터널의 목적지 IPv4 주소를 추출하여 패킷을 인캡슐레이션한다.
- 목적지별로 일일이 터널 인터페이스를 생성할 필요가 없다.

□ 장점

- 하나의 IPv4 주소를 통해 /48의 IPv6 네트워크 대역을 생성하고, 이를 바탕으로 내부 IPv6 네트워크를 구성할 수 있다.

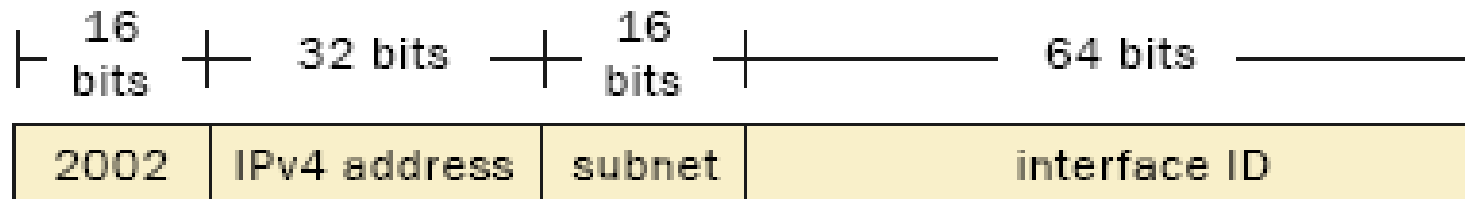
□ 단점

- Edge 라우터에 부여되어 있는 공인 IPv4 주소가 변경될 경우 6to4 주소 포켓도 함께 변경이 되어버린다.

6to4 주소

□ 6to4 주소 구성

- IPv4 주소를 바탕으로 생성
- 포맷
 - 6to4 Prefix: 최상위 16비트 부분 - 2002::로 시작
 - 다음 32비트: IPv4 주소
 - Subnet: 관리자가 임의로 할당
 - Interface ID: 내부 IPv6 네트워크장비들에게 할당





6to4 Tunnel 설정

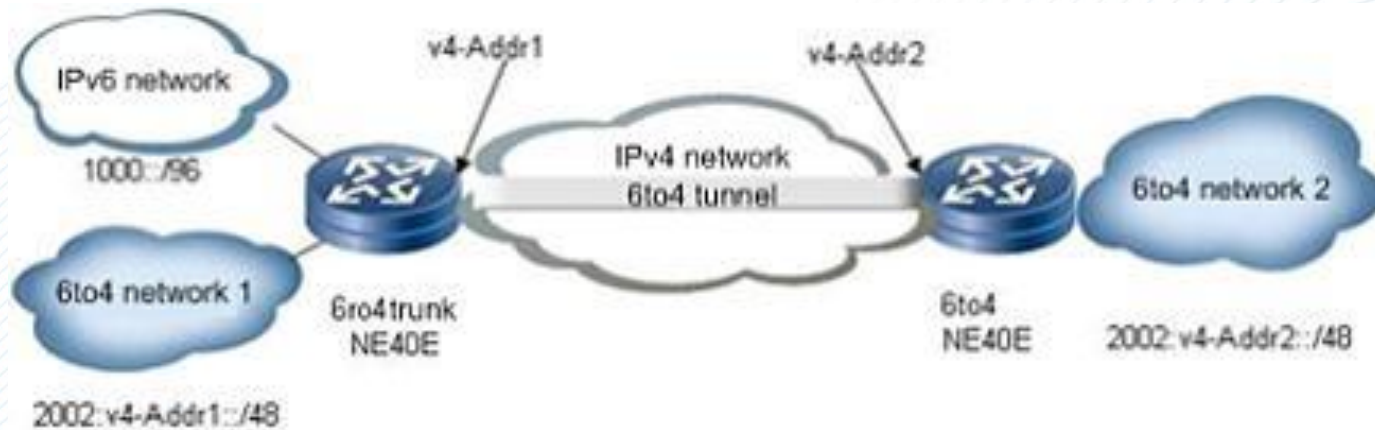
▣ 터널 인터페이스 설정

- R(config-if)# **tunnel mode ipv6ip 6to4**
- R(config-if)# **ipv6 enable**

- 터널의 목적지 IPv4 주소는 지정하지 않는다.
- 터널링 동작만을 위해서라면 주소를 부여하지 않아도 된다.
 - ▣ 이때는 ipv6 enable 명령어나 링크 로컬 주소를 부여함으로써 활성화

6to4 Relay

- ▣ **6to4 Relay 역할**
 - 목적지 주소의 대역이 2002::/16이 아닐 경우라도 6to4 터널을 통해 통신이 가능하도록 도와주는 것
- ▣ **공영 6to4 Relay 주소**
 - 192.88.99.0/24
 - IANA가 6to4 Relay를 위해 할당해놓은 IPv4 네트워크 대역



6to4 Relay 설정

▣ 6to4 터널 인터페이스 설정

- R(config)# int tunnel 0
- R(config-if)# **tunnel mode ipv6ip 6to4**
- R(config-if)# ipv6 enable
- R(config-if)# tunnel source lo0

- R(config)# ipv6 route 2002::/16 tunnel0
- R(config)# **ipv6 route ::/0 2002:101:303**
- Default Route의 다음 홉 주소정보는 존재하지 않는 IPv6 주소
 - ▣ 단순히 터널의 목적지 IPv4주소를 추출하는 용도로 사용
 - ▣ /48까지만 정확히 지정하면 된다.