Oracle® Solaris 11.1 네트워크 구성 및 관리

부품 번호: E36459–02
2012년 10월

ORACLE

121210@25097

목차

머리말

Oracle Solaris 11.1 네트워크 구성 및 관리를 시작합니다. 이 설명서는 Oracle Solaris 네트워크 구성을 위한 기본 항목 및 절차가 포함된 **Establishing an Oracle Solaris 11.1 Network** 시리즈의 일부입니다. 이 설명서에서는 Oracle Solaris를 이미 설치했다고 가정합니다. 따라서 네트워크를 구성하거나 네트워크에 필요한 네트워킹 소프트웨어를 구성할 준비가 되어 있어야 합니다.

이 설명서의 대상

이 설명서는 네트워크에 구성된 Oracle Solaris 실행 시스템을 관리하는 모든 사용자를 대상으로 합니다. 이 설명서를 사용하려면 적어도 1~2년의 UNIX 시스템 관리 경험이 있어야 합니다. UNIX 시스템 관리 교육 과정에 참석하는 것도 도움이 될 수 있습니다.

Oracle Support에 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info`를 참조하거나, 청각 장애가 있는 경우 `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs`를 방문하십시오.

활자체 규약

다음 표는 이 설명서에서 사용되는 활자체 규약에 대해 설명합니다.

표 P–1 활자체 규약

활자체	설명	예
`AaBbCc123`	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	`.login` 파일을 편집하십시오. 모든 파일 목록을 보려면 `ls -a` 명령을 사용하십시오. `machine_name% you have mail.`

표 P–1 활자체 규약 *(계속)*

활자체	설명	예
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	`machine_name%` **`su`** `Password:`
AaBbCc123	위치 표시자: 실제 이름이나 값으로 바뀝니다.	`rm` *filename* 명령을 사용하여 파일을 제거합니다.
AaBbCc123	설명서 제목, 새 용어, 강조 표시할 용어입니다.	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하면 **안 됩니다**. **주:** 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 수퍼유저 프롬프트가 나와 있습니다. 명령 예제에 표시된 기본 시스템 프롬프트는 Oracle Solaris 릴리스에 따라 다릅니다.

표 P–2 셸 프롬프트

셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	`$`
수퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	`#`
C 셸	`machine_name%`
수퍼유저용 C 셸	`machine_name#`

◆ ◆ ◆ 1 장

1 네트워크 배치 계획

이 장에서는 네트워크 설정을 계획할 때 고려해야 할 몇 가지 사항에 대해 간략하게 설명합니다. 이러한 고려 사항은 계획적이며 비용 효율적인 방식으로 네트워크를 배치하는 데 유용합니다. 네트워크 계획에 대한 자세한 내용은 본 설명서에서 다루지 않습니다. 여기서는 일반적인 지침만 제공합니다.

본 설명서에서는 사용자가 기본적인 네트워킹 개념 및 용어에 친숙한 것으로 간주합니다. Oracle Solaris 11에서 TCP/IP 프로토콜 제품군을 구현하는 방법에 대한 설명은 **Oracle Solaris 11 네트워킹 소개**의 "Oracle Solaris의 네트워크 스택"을 참조하십시오.

네트워크 계획(작업 맵)

다음 표에서는 네트워크 구성 계획과 관련된 다양한 작업을 나열합니다.

작업	설명	정보
계획된 네트워크 토폴로지의 하드웨어 요구 사항을 식별합니다.	네트워크 사이트에 필요한 장비의 유형을 결정합니다.	12 페이지 "네트워크 하드웨어 결정" 특정 장비 유형에 대한 자세한 내용은 장비 제조업체 설명서를 참조하십시오.
등록된 IP 주소를 사용하고 얻는 데 필요한 IP 주소의 유형을 결정합니다.	IPv4 네트워크와 IPv6 네트워크 중 하나만 배치할지 아니면 두 유형의 IP 주소를 모두 사용하는 네트워크를 배치할지 선택합니다. 인터넷의 공용 네트워크와 통신할 고유한 IP 주소를 얻습니다.	13 페이지 "네트워크에 대한 IP 주소 지정 형식 결정" 15 페이지 "네트워크의 IP 번호 얻기"

작업	설명	정보
사용할 이름 서비스와 함께 네트워크의 호스트를 식별할 이름 지정 체계를 결정합니다.	네트워크의 시스템에 지정할 이름 목록을 만들고 NIS와 LDAP, DNS, 로컬 /etc 디렉토리의 네트워크 데이터베이스 중 사용할 데이터베이스를 결정합니다.	15 페이지 “호스트 이름 관리” 16 페이지 “이름 서비스 및 디렉토리 서비스 선택”
필요한 경우 관리 세분화를 설정하고 서브넷 전략을 설계합니다.	사이트에서 관리 세분화를 제공하기 위해 네트워크를 서브넷으로 구분해야 할지 여부를 결정합니다.	17 페이지 “서브넷 사용”
네트워크 설계 시 라우터를 배치할 위치를 결정합니다.	라우터가 필요한 만큼 네트워크가 큰 경우 라우터를 지원하는 네트워크 토폴로지를 만듭니다.	17 페이지 “네트워크의 라우터 계획”
전체 네트워크 구성 체계에 가상 네트워크를 만들지 여부를 결정합니다.	네트워크의 하드웨어 메모리 단위를 줄이기 위해 시스템에 가상 네트워크를 만들어야 할 수도 있습니다.	**Oracle Solaris 11.1에서 가상 네트워크 사용**

네트워크 하드웨어 결정

지원해야 할 시스템 수에 따라 네트워크 구성 방식이 달라집니다. 한 건물의 한 층에 수십 대의 독립형 시스템이 배치되는 작은 규모의 네트워크가 조직에 필요할 수도 있고, 여러 건물에 1,000대 이상의 시스템이 배치되는 네트워크를 설정해야 할 수도 있습니다. 이 설정에 따라 **서브넷**이라는 세분화로 네트워크를 추가로 구분해야 할 수 있습니다.

하드웨어에 대해 결정해야 할 몇 가지 계획 요소는 다음과 같습니다.

- 네트워크 토폴로지, 레이아웃 및 네트워크 하드웨어 연결
- 필요한 서버를 비롯하여 네트워크가 지원할 수 있는 호스트 시스템의 유형 및 수
- 이러한 시스템에 설치할 네트워크 장치
- 사용할 네트워크 매체의 유형(예: 이더넷 등)
- 이 매체를 확장하거나 로컬 네트워크를 외부 네트워크에 연결할 브릿지 또는 라우터가 필요한지 여부

주 – 라우터 작동 방법에 대한 설명은 17 페이지 “네트워크의 라우터 계획”을 참조하십시오. 브리지에 대한 개요는 **Oracle Solaris 11.1 네트워크 성능 관리**의 “브리징 개요”를 참조하십시오.

네트워크에 대한 IP 주소 지정 형식 결정

네트워크 주소 지정 체계를 계획할 때는 다음 요소를 고려하십시오.

- 사용할 IP 주소의 유형(IPv4 또는 IPv6)
- 네트워크의 잠재적 시스템 수
- 고유한 개별 IP 주소와 함께 여러 네트워크 인터페이스 카드(NIC)를 필요로 하는 멀티홈 또는 라우터 시스템 수
- 네트워크에서 개인 주소를 사용할지 여부
- IPv4 주소 풀을 관리하는 DHCP 서버를 사용할지 여부

다음은 IP 주소 유형을 요약한 것입니다.

IPv4 주소

이러한 32비트 주소는 TCP/IP에 대한 원래 IP 주소 지정 형식입니다. 이후에 IETF는 IPv4 주소 부족 및 전역 인터넷 경로 지정표의 제한적인 용량에 대한 중/단기적인 해결책으로 *CIDR(Classless Inter-Domain Routing)* 주소를 개발했습니다.

자세한 내용은 다음 자료를 참조하십시오.

- Internet Protocol DARPA Internet Program Protocol Specification (`http://tools.ietf.org/html/rfc791`)
- Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan (`http://tools.ietf.org/html/rfc4632`)

다음 표에서는 서브넷을 CIDR 표기법과 점으로 구분된 십진수 형식으로 제공합니다.

표 1–1 CIDR 접두어 및 이와 동등한 십진수

CIDR 네트워크 접두어	동등한 점으로 구분된 십진수 서브넷	사용 가능한 IP 주소
/19	255.255.224.0	8,192
/20	255.255.240.0	4,096
/21	255.255.248.0	2,048
/22	255.255.252.0	1,024
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64

표 1–1 CIDR 접두어 및 이와 동등한 십진수 (계속)

CIDR 네트워크 접두어	동등한 점으로 구분된 십진수 서브넷	사용 가능한 IP 주소
/27	255.255.255.224	32

DHCP 주소

DHCP(Dynamic Host Configuration Protocol) 프로토콜을 통해 시스템은 부트 프로세스의 일부로 DHCP 서버로부터 IP 주소 등의 구성 정보를 수신할 수 있습니다. DHCP 서버는 DHCP 클라이언트에 주소를 지정할 IP 주소의 풀을 유지 관리합니다. DHCP를 사용하는 사이트는 모든 클라이언트에 영구 IP 주소를 지정했을 때 필요한 것보다 작은 IP 주소 풀을 사용할 수 있습니다. DHCP 서비스를 설정하여 사이트의 IP 주소 또는 주소 일부를 관리할 수 있습니다. 자세한 내용은 **Oracle Solaris 11.1의 DHCP 작업**의 1 장, "DHCP 정보(개요)"를 참조하십시오.

IPv6 주소

128비트 IPv6 주소는 IPv4에서 사용할 수 있는 것보다 큰 주소 공간을 제공합니다. CIDR 형식의 IPv4 주소와 마찬가지로 IPv6 주소는 클래스가 없으며 접두어를 사용하여 사이트의 네트워크를 정의하는 주소 일부를 지정합니다. IPv6 주소 지정에 대한 자세한 내용은 Internet Protocol, Version 6 (IPv6) Specification (`http://tools.ietf.org/html/rfc2460`)을 참조하십시오.

개인 주소 및 설명서 접두어

IANA는 개인 네트워크에 사용하도록 IPv4 주소 블록 및 IPv6 사이트 접두어를 예약했습니다. 이러한 개인 주소는 개인 네트워크 내의 네트워크 트래픽에 사용되며, 설명서에서도 사용됩니다.

다음 표에서는 IPv4 주소 범위와 해당 넷마스크를 나열합니다.

IPv4 주소 범위	넷마스크
10.0.0.0 - 10.255.255.255	10.0.0.0
172.16.0.0 - 172.31.255.255	172.16.0.0
192.168.0.0 - 192.168.255.255	192.168.0.0

IPv6 주소의 경우 접두어 `2001:db8::/32`는 설명서 예에서 특별히 사용되는 특수한 IPv6 접두어입니다. 본 설명서의 예에서는 개인 IPv4 주소와 예약된 IPv6 설명서 접두어를 사용합니다.

네트워크의 IP 번호 얻기

IPv4 네트워크는 IPv4 네트워크 번호와 네트워크 마스크(**넷마스크**)의 조합으로 정의됩니다. IPv6 네트워크는 **사이트 접두어** 및 **서브넷 접두어**(서브넷으로 구분된 경우)로 정의됩니다.

개인 네트워크가 인터넷의 외부 네트워크와 통신할 수 있도록 하려면 해당 조직으로부터 네트워크에 대해 등록된 IP 번호를 얻어야 합니다. 이 주소가 IPv4 주소 지정 체계에 대한 네트워크 번호 또는 IPv6 주소 지정 체계에 대한 사이트 접두어로 사용됩니다.

인터넷 서비스 제공업체가 다양한 서비스 레벨을 기반으로 한 가격에 따라 네트워크에 대한 IP 주소를 제공합니다. 여러 ISP를 조사하여 네트워크에 가장 적합한 서비스를 제공하는 ISP를 결정하십시오. 일반적으로 ISP는 기업에 동적으로 할당되는 주소 또는 정적 IP 주소를 제공합니다. IPv4 주소와 IPv6 주소를 모두 제공하는 ISP도 있습니다.

사이트가 ISP인 경우 로케일에 적합한 인터넷 레지스트리(IR)로부터 고객의 IP 주소 블록을 얻습니다. 궁극적으로 IANA(Internet Assigned Numbers Authority)에서 등록된 IP 주소를 전세계의 IR로 위임합니다. 각 IR에는 IR이 제공하는 로케일에 적합한 템플리트와 등록 정보가 있습니다. IANA 및 IR에 대한 자세한 내용은 IANA's IP Address Service 페이지 (`http://www.iana.org/ipaddress/ip-addresses.htm`)를 참조하십시오.

네트워크의 이름 지정 엔티티

TCP/IP 프로토콜은 IP 주소를 사용하여 네트워크에서 시스템을 찾습니다. 하지만 호스트 이름을 사용하면 IP 주소보다 간편하게 시스템을 식별할 수 있습니다. TCP/IP 프로토콜(및 Oracle Solaris)의 경우 시스템을 고유하게 식별하는 데 IP 주소와 호스트 이름이 모두 필요합니다.

TCP/IP 관점에서 네트워크는 일련의 이름이 지정된 엔티티입니다. 호스트는 이름이 있는 엔티티입니다. 라우터도 이름이 있는 엔티티이며, 네트워크도 이름이 있는 엔티티입니다. 네트워크가 설치된 그룹 또는 부서가 사업부, 지역 또는 회사일 수 있으므로 해당 그룹 또는 부서에도 이름이 지정될 수 있습니다. 이론상 네트워크 식별에 사용될 수 있는 이름의 계층은 거의 제한이 없습니다. 도메인 이름은 **도메인**을 식별합니다.

호스트 이름 관리

네트워크를 구성할 시스템에 대한 이름 지정 체계를 계획합니다. 서버로 작동하며 NIC가 여러 개인 시스템의 경우 기본 네트워크 인터페이스의 IP 주소와 연관된 호스트 이름을 하나 이상 제공해야 합니다.

네트워크에 있는 두 시스템이 동일한 호스트 이름을 가질 수 없습니다. 따라서 각 호스트 이름은 각 시스템에 대해 고유해야 합니다. 하지만 고유한 이름이 지정된 호스트 또는 시스템의 IP 주소는 여러 개일 수 있습니다.

네트워크를 계획할 때는 설정 프로세스 중 간편하게 액세스할 수 있도록 IP 주소 및 연관된 호스트 이름 목록을 만드십시오. 이 목록을 통해 모든 호스트 이름이 고유한지 확인할 수 있습니다.

이름 서비스 및 디렉토리 서비스 선택

Oracle Solaris에서는 세 가지 유형의 이름 서비스(로컬 파일, NIS 및 DNS) 중에서 선택할 수 있습니다. 이름 서비스는 네트워크의 시스템에 대한 중요한 정보(예: 호스트 이름, IP 주소, 이더넷 주소 등)를 유지 관리합니다. 이름 서비스와 함께, 또는 이름 서비스 대신 LDAP 디렉토리 서비스를 사용할 수도 있습니다. Oracle Solaris의 이름 서비스 소개는 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**의 제I부, "이름 지정 및 디렉토리 서비스 정보"를 참조하십시오.

OS 설치 중 서버, 클라이언트 또는 독립형 시스템의 호스트 이름과 IP 주소를 제공합니다. 설치 프로그램이 네트워크를 제공할 때 네트워크 서비스에 사용될 `hosts` 데이터베이스에 이 정보를 추가합니다.

네트워크 데이터베이스의 구성은 중요합니다. 따라서 네트워크 계획 프로세스의 일부로 사용할 이름 서비스를 결정해야 합니다. 또한 이름 서비스 사용 여부 결정에 따라 조직에서 네트워크를 관리 도메인으로 구성할지 여부가 달라집니다.

이름 서비스로 다음 중 하나를 선택할 수 있습니다.

- NIS 또는 DNS – NIS 및 DNS 이름 서비스는 네트워크에 있는 여러 서버의 네트워크 데이터베이스를 유지 관리합니다. **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**에서는 해당 이름 서비스 및 데이터베이스 구성 방법에 대해 설명합니다. "이름 공간" 및 "관리 도메인" 개념에 대해서도 자세히 설명합니다.
- 로컬 파일 – NIS, LDAP 또는 DNS를 구현하지 않을 경우 네트워크는 **로컬 파일**을 사용하여 이름 서비스를 제공합니다. "로컬 파일"이라는 용어는 네트워크 데이터베이스에 사용되는 `/etc` 디렉토리의 일련의 파일을 의미합니다. 본 설명서의 절차에서는 별도로 지정되지 않은 경우 로컬 파일을 이름 서비스로 사용 중인 것으로 간주합니다.

주 – 네트워크에 대한 이름 서비스로 로컬 파일을 사용하기로 결정할 경우 나중에 다른 이름 서비스를 설정할 수 있습니다.

도메인 이름

여러 네트워크는 호스트 및 라우터를 관리 도메인의 계층으로 구성합니다. NIS 또는 DNS 이름 서비스를 사용 중인 경우 조직에 대해 전세계적으로 고유한 도메인 이름을 선택해야 합니다. 도메인 이름이 고유하도록 하려면 InterNIC에 도메인 이름을 등록해야 합니다. DNS를 사용하려는 경우에도 InterNIC에 도메인 이름을 등록해야 합니다.

도메인 이름 구조는 계층 구조입니다. 일반적으로 새 도메인은 기존의 관련 도메인 아래에 배치됩니다. 예를 들어, 자회사의 도메인 이름은 모회사의 도메인 아래에 배치될 수 있습니다. 도메인 이름에 다른 관계가 없을 경우 조직에서는 기존의 최상위 레벨 도메인(예: `.com`, `.org`, `.edu`, `.gov` 등) 중 하나의 바로 아래에 도메인 이름을 배치할 수 있습니다.

서브넷 사용

서브넷 사용은 크기 및 제어 문제를 해결하기 위해 관리 세분화를 사용해야 하는 것과 관련이 있습니다. 네트워크에 있는 호스트 및 서버가 많을수록 관리 작업이 복잡해집니다. 관리 세분화를 만들고 서브넷을 사용하면 복잡한 네트워크 관리가 간편해집니다. 네트워크에 대한 관리 세분화를 설정하는 것은 다음 요소에 따라 결정됩니다.

- **네트워크 크기**

  서브넷은 광대한 지역에 세분화가 배치된 비교적 작은 네트워크에서도 유용합니다.

- **사용자 그룹의 공통 요구 사항**

  예를 들어, 한 건물에 국한되며 비교적 적은 수의 시스템을 지원하는 네트워크가 있을 수 있습니다. 이러한 시스템은 여러 하위 네트워크로 구분됩니다. 각 하위 네트워크는 요구 사항이 다른 사용자 그룹을 지원합니다. 이 예에서는 각 서브넷에 대해 관리 세분화를 사용할 수 있습니다.

네트워크의 라우터 계획

TCP/IP에서는 호스트 및 라우터의 두 가지 유형의 엔티티가 네트워크에 존재합니다. 모든 네트워크에는 호스트가 있어야 하며, 라우터는 네트워크에 따라 필요합니다. 네트워크의 물리적 토폴로지에 따라 라우터가 필요한지 여부가 결정됩니다. 이 절에서는 네트워크 토폴로지 및 경로 지정의 개념에 대해 소개합니다. 이러한 개념은 기존 네트워크 환경에 다른 네트워크를 추가하려는 경우에 중요합니다.

주 – IPv4 네트워크의 라우터 구성을 위한 자세한 내용 및 작업을 보려면 35 페이지 “네트워크의 구성 요소 시스템 구성”을 참조하십시오. IPv6 네트워크의 라우터 구성을 위한 자세한 내용 및 작업을 보려면 64 페이지 “IPv6 라우터 구성”을 참조하십시오.

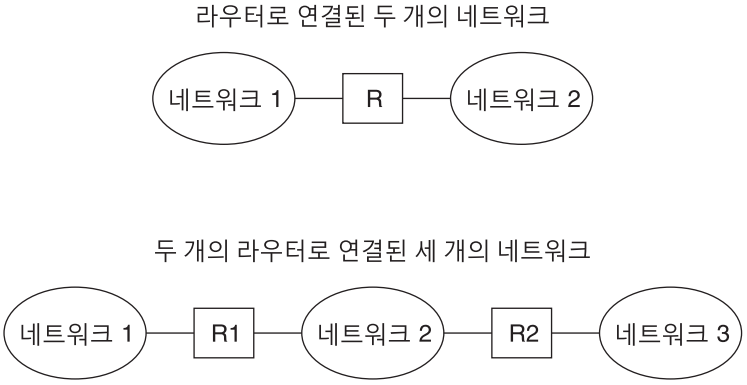
네트워크 토폴로지 개요

네트워크 토폴로지는 네트워크의 연결 방식을 기술합니다. 라우터는 네트워크를 서로 연결해주는 엔티티입니다. 라우터는 두 개 이상의 네트워크 인터페이스를 포함하고 IP 전달을 구현하는 시스템입니다. 하지만 시스템은 43 페이지 “IPv4 라우터 구성”에 설명된 대로 올바르게 구성하지 않으면 라우터로 작동할 수 없습니다.

라우터는 두 개 이상의 네트워크를 연결하여 보다 큰 인터네트워크를 형성합니다. 라우터는 두 개의 인접한 네트워크 간에 패킷을 전달하도록 구성되어야 합니다. 라우터는 또한 인접한 네트워크 외부에 있는 네트워크에 패킷을 전달할 수 있어야 합니다.

다음 그림에서는 네트워크 토폴로지의 기본 요소를 보여줍니다. 첫번째 그림은 단일 라우터로 연결된 두 네트워크의 간단한 구성을 보여줍니다. 두번째 그림은 두 개의 라우터로 연결된 세 네트워크의 구성을 보여줍니다. 첫번째 예제에서 라우터 R은 네트워크 1 및 네트워크 2를 보다 큰 인터네트워크로 결합합니다. 두번째 예제에서 라우터 R1은 네트워크 1과 2를 연결합니다. 라우터 R2는 네트워크 2와 3을 연결합니다. 이러한 연결로 네트워크 1, 2, 3이 포함된 네트워크가 형성됩니다.

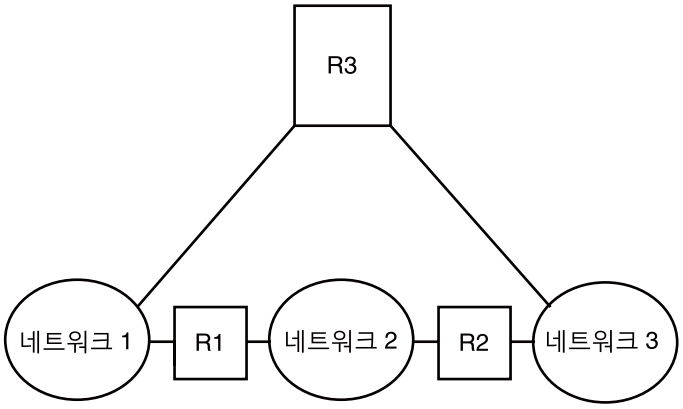
그림 1–1 기본 네트워크 토폴로지



네트워크를 인터네트워크로 연결하는 것 외에도 라우터는 대상 네트워크의 주소에 따라 네트워크 사이에 패킷 경로를 지정합니다. 인터네트워크가 점점 더 복잡해짐에 따라 각 라우터는 패킷 대상에 대해 더 많은 항목을 결정해야 합니다.

다음 그림은 보다 복잡한 경우를 보여줍니다. 라우터 R3은 네트워크 1과 3을 연결합니다. 중복성은 신뢰성을 향상해 줍니다. 네트워크 2가 작동 중지되면 라우터 R3이 네트워크 1과 3 사이의 경로를 제공할 수 있습니다. 여러 네트워크를 상호 연결시킬 수 있습니다. 하지만 네트워크는 동일한 네트워크 토폴로지를 사용해야 합니다.

그림 1–2 네트워크 사이에 추가 경로를 제공하는 네트워크 토폴로지



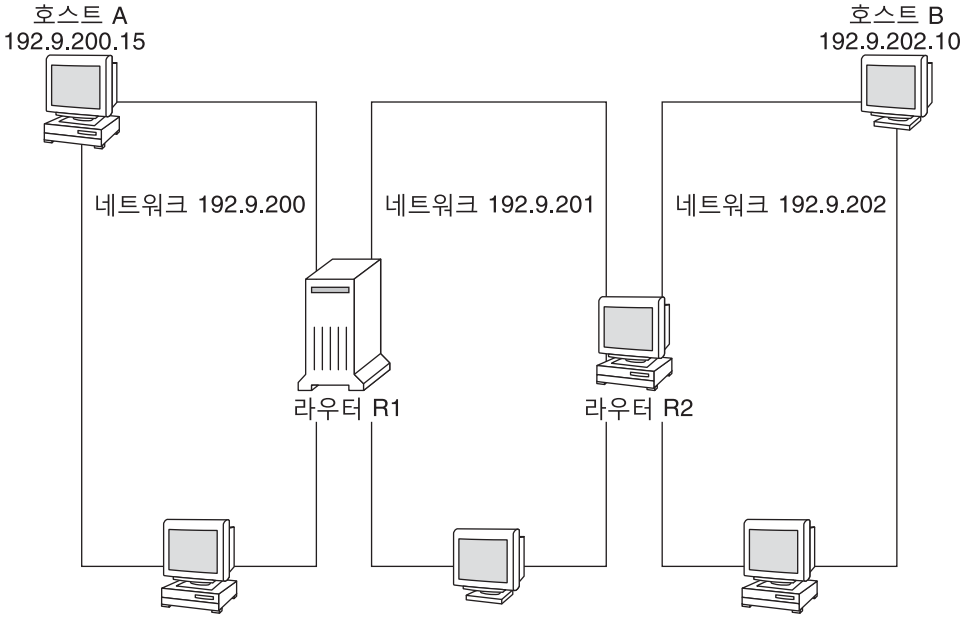
라우터가 패킷을 전송하는 방법

패킷 헤더에 포함되는 수신자의 IP 주소에 따라 패킷의 경로 지정 방식이 결정됩니다. 이 주소에 로컬 네트워크의 네트워크 번호가 포함된 경우 패킷이 해당 IP 주소의 호스트로 직접 이동합니다. 네트워크 번호가 로컬 네트워크가 아닌 경우 패킷이 로컬 네트워크의 라우터로 이동합니다.

라우터는 경로 지정 정보를 **경로 지정 테이블**에 유지 관리합니다. 이러한 테이블에는 라우터가 연결된 네트워크 상의 호스트 및 라우터의 IP 주소가 포함됩니다. 테이블에는 또한 이러한 네트워크에 대한 포인터도 포함됩니다. 라우터가 패킷을 수신하면 경로 지정 테이블에서 테이블의 헤더에 대상 주소가 나열되어 있는지 확인합니다. 테이블에 대상 주소가 포함되지 않았으면 라우터가 해당 경로 지정 테이블에 나열된 다른 라우터로 패킷을 전달합니다. 라우터에 대한 자세한 내용은 43 페이지 “IPv4 라우터 구성”을 참조하십시오.

다음 그림에서는 두 라우터로 연결된 세 네트워크의 네트워크 토폴로지를 보여줍니다.

그림 1–3 상호 연결된 세 네트워크의 네트워크 토폴로지



라우터 R1은 네트워크 192.9.200 및 192.9.201을 연결합니다. 라우터 R2는 네트워크 192.9.201 및 192.9.202를 연결합니다.

네트워크 192.9.200의 호스트 A가 네트워크 192.9.202의 호스트 B에 메시지를 전송하면 다음과 같은 이벤트가 발생합니다.

1. 호스트 A가 네트워크 192.9.200을 통해 패킷을 전송합니다. 패킷 헤더에는 수신자 호스트 B의 IPv4 주소인 192.9.202.10 이 포함됩니다.
2. 네트워크 192.9.200의 시스템에는 IPv4 주소 192.9.202.10이 포함되지 않습니다. 따라서 라우터 R1이 패킷을 수락합니다.
3. 라우터 R1은 해당 경로 지정 테이블을 검사합니다. 네트워크 192.9.201의 시스템에는 주소 192.9.202.10이 포함되지 않습니다. 하지만 경로 지정 테이블에는 라우터 R2가 나열되지 않습니다.
4. 그런 후 R1은 R2를 "다음 홉" 라우터로 선택합니다. R1은 패킷을 R2로 전송합니다.
5. R2는 네트워크 192.9.201을 192.9.202에 연결하기 때문에 R2는 호스트 B에 대한 경로 지정 정보를 포함합니다. 그런 후 라우터 R2는 패킷을 네트워크 192.9.202에 전달하고 여기에서 호스트 B가 패킷을 수락합니다.

가상 네트워크 배치

이 Oracle Solaris 릴리스에서는 가상 네트워크 카드(VNIC)와 영역을 구성하여 단일 네트워크에 여러 가상 네트워크를 만들 수 있도록 지원합니다. VNIC는 물리적 NIC 위에 만들어지는 네트워크 인터페이스입니다. 영역과 VNIC를 결합하면 많은 수의 물리적 시스템을 포함하는 거대한 데이터 센터를 적은 수의 시스템에 효과적으로 통합할 수 있습니다. 가상 네트워킹에 대한 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

◆ ◆ ◆ 2 장

2 IPv6 주소 사용 시 고려 사항

이 장은 1 장, “네트워크 배치 계획”의 내용을 보완하기 위해 네트워크에서 IPv6 주소를 사용하기로 결정한 경우 추가 고려 사항에 대해 설명합니다.

IPv4 주소와 IPv6 주소를 모두 사용하도록 계획한 경우 현재 ISP가 두 주소 유형을 모두 지원하는지 확인하십시오. 그렇지 않은 경우 IPv6 주소를 지원하는 별도의 ISP를 찾아야 합니다.

IPv6 개념에 대한 소개는 다음 리소스를 참조하고 Internet Protocol, Version 6 (IPv6) Specification (`http://www.ietf.org/rfc/rfc2460.txt`)을 참조하십시오.

IPv6 계획(작업 맵)

다음 표는 네트워크에서 IPv6을 구현하려고 계획한 경우 여러 고려 사항을 보여줍니다.

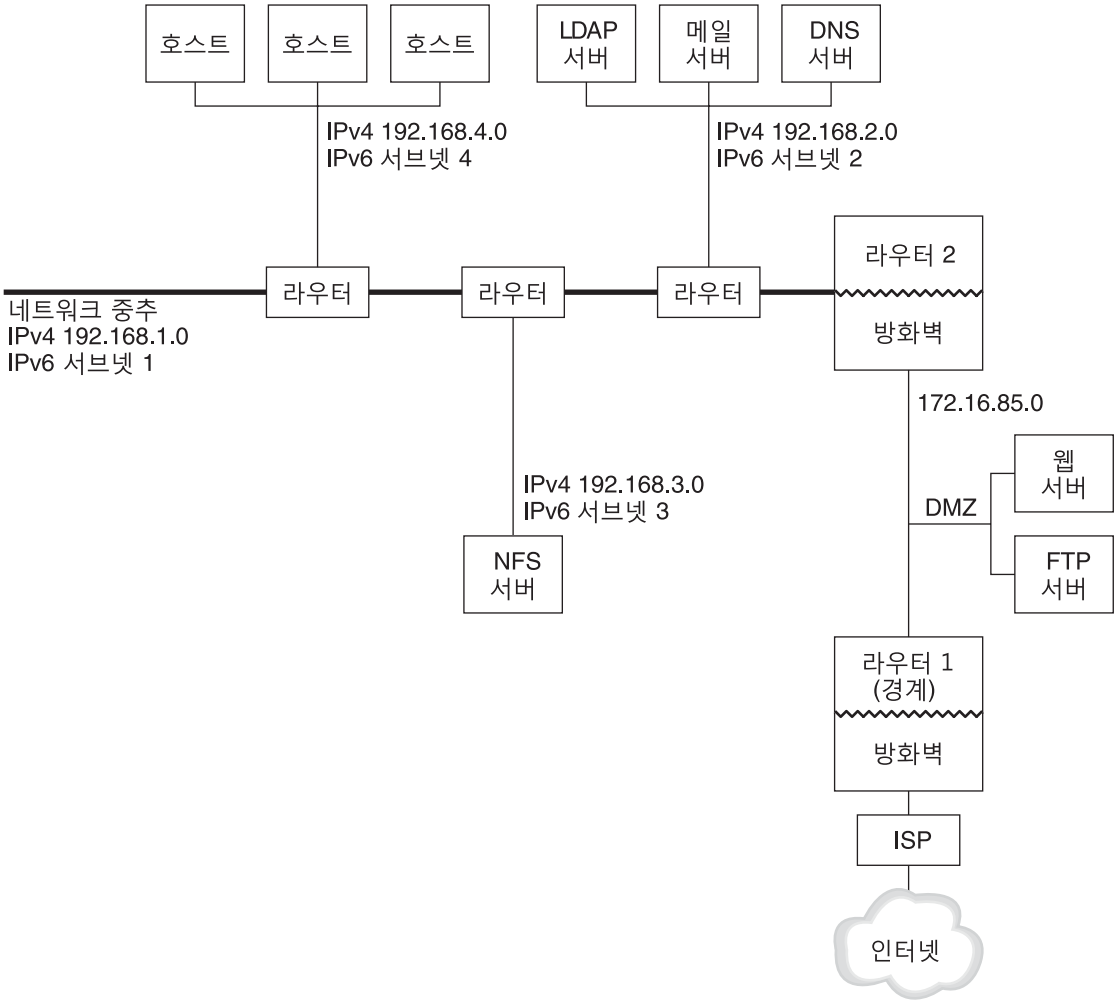
작업	설명	수행 방법
IPv6을 지원하도록 하드웨어 준비	하드웨어를 IPv6으로 업그레이드할 수 있는지 확인합니다.	26 페이지 “IPv6에 대한 하드웨어 지원 확인”
IPv6에서 응용 프로그램을 사용할 수 있는지 확인	IPv6 환경에서 응용 프로그램을 실행할 수 있는지 확인합니다.	28 페이지 “IPv6을 지원하도록 네트워크 서비스 구성”
터널 사용 계획 설계	다른 서브넷 또는 외부 네트워크에 대한 터널을 실행할 라우터를 결정합니다.	30 페이지 “네트워크에서 터널 사용 계획”

작업	설명	수행 방법
네트워크 보안을 설정하고 IPv6 보안 정책을 개발하는 방법 계획	보안을 위해 IPv6을 구성하기 전에 DMZ 및 해당 엔티티에 대한 주소 지정 계획이 필요합니다. 이 릴리스의 IP 필터, IP 보안 아키텍처(IPsec), IKE(Internet Key Exchange) 및 기타 보안 기능 사용과 같은 보안 구현 방식을 결정합니다.	31 페이지 “IPv6 구현에 대한 보안 고려 사항” **Oracle Solaris 11.1의 네트워크 보안**
네트워크 시스템에 대한 주소 지정 계획 만들기	IPv6을 구성하기 전에 서버, 라우터 및 호스트에 대한 계획을 세워야 합니다. 이 단계에는 네트워크에 대한 사이트 접두어 얻기 및 IPv6 서브넷 계획(필요한 경우) 작업이 포함됩니다.	27 페이지 “노드에 대한 IPv6 주소 지정 계획 만들기”

IPv6 네트워크 토폴로지 시나리오

일반적으로 IPv6은 다음 그림에 표시된 것과 같이 IPv4도 사용하는 혼합 네트워크 토폴로지에 사용됩니다. 이후 단원의 IPv6 구성 작업에 대한 설명에서 이 그림을 참조할 수 있습니다.

그림 2–1 IPv6 네트워크 토폴로지 시나리오



엔터프라이즈 네트워크 시나리오는 기존 IPv4 주소를 포함하는 5개의 서브넷으로 구성됩니다. 네트워크 링크는 관리 서브넷과 직접적으로 일치합니다. 네 개의 내부 네트워크는 RFC 1918 스타일의 개인 IPv4 주소로 표시되는데, 이는 IPv4 주소가 없는 경우의 일반적인 솔루션입니다. 이 내부 네트워크의 주소 지정 체계는 다음과 같습니다.

- 서브넷 1은 내부 네트워크 중추 `192.168.1`입니다.
- 서브넷 2는 LDAP `sendmail` 및 DNS 서버를 포함하는 내부 네트워크 `192.168.2`입니다.
- 서브넷 3은 엔터프라이즈의 NFS 서버를 포함하는 내부 네트워크 `192.168.3`입니다.
- 서브넷 4는 엔터프라이즈 직원에 대한 호스트를 포함하는 내부 네트워크 `192.168.4`입니다.

외부 공개 네트워크 172.16.85는 회사의 DMZ처럼 작동합니다. 이 네트워크에는 웹 서버, 익명 FTP 서버 및 엔터프라이즈가 외부에 제공하는 기타 리소스가 포함되어 있습니다. 라우터 2는 내부 중추와 구분된 공개 네트워크 172.16.85 및 방화벽을 실행합니다. DMZ의 다른 쪽 끝에 있는 라우터 1은 방화벽을 실행하며 엔터프라이즈의 경계 서버로 사용됩니다.

그림 2–1에서 공개 DMZ의 RFC 1918 전용 주소는 172.16.85입니다. 실제로 공개 DMZ에는 등록된 IPv4 주소가 있습니다. 대부분의 IPv4 사이트는 공개 주소 및 RFC 1918 개인 주소를 결합하여 사용합니다. 그러나 IPv6을 사용할 경우 공개 주소 및 개인 주소의 개념이 달라집니다. IPv6의 주소 공간은 훨씬 더 크므로 개인 네트워크 및 공개 네트워크 모두에서 공개 IPv6 주소를 사용하십시오.

Oracle Solaris 듀얼 프로토콜 스택은 동시 IPv4 및 IPv6 작업을 지원합니다. 네트워크에 IPv6을 배치하는 동안이나 배치한 후 IPv4 관련 작업을 성공적으로 실행할 수 있습니다. 이미 IPv4를 사용 중인 작동 중인 네트워크에 IPv6을 배치할 경우 진행 중인 작업이 중단되지 않습니다.

다음 절에서는 IPv6 구현을 준비할 때 고려해야 할 영역에 대해 설명합니다.

IPv6에 대한 하드웨어 지원 확인

하드웨어의 다음 클래스와 관련하여 IPv6이 사용 가능한지 제조업체의 설명서를 확인하십시오.

- 라우터
- 방화벽
- 서버
- 스위치

주 – 이 설명서의 모든 절차는 장비 특히 라우터를 IPv6으로 업그레이드할 수 있다고 가정합니다.

일부 라우터 모델은 IPv6으로 업그레이드할 수 없습니다. 자세한 내용 및 임시해결책은 **Troubleshooting Network Issues**의 "IPv4 Router Cannot Be Upgraded to IPv6"을 참조하십시오.

Neighbor Discovery 프로토콜을 사용하여 ID를 자동으로 얻는 대신, IPv6 서버의 NIC마다 IPv6 주소의 인터페이스 ID 부분을 수동으로 구성하십시오. 이 방식에서 NIC가 교체될 경우 동일한 인터페이스 ID를 대체 NIC에 적용할 수 있습니다. 다른 ID가 Neighbor Discovery 프로토콜을 통해 자동으로 생성될 경우 서버에서 예상치 않은 동작이 발생할 수 있습니다.

IPv6 주소 지정 계획 준비

IPv4에서 IPv6으로 전환하는 데 있어 중요한 부분은 주소 지정 계획을 개발하는 것입니다. 이 작업은 다음과 같은 준비 작업과 관련됩니다.

사이트 접두어 획득

IPv6을 구성하기 전에 사이트 접두어를 획득해야 합니다. 사이트 접두어는 IPv6 구현에서 모든 노드에 대한 IPv6 주소를 파생시키는 데 사용됩니다.

IPv6을 지원하는 ISP는 48비트 IPv6 사이트 접두어를 조직에 제공합니다. 현재 ISP가 IPv4만 지원할 경우 IPv4 지원을 위한 현재 ISP를 유지하면서 IPv6 지원을 위한 다른 ISP를 사용할 수 있습니다. 이 경우 여러 임시해결책 중 하나를 사용할 수 있습니다. 자세한 내용은 **Troubleshooting Network Issues**의 "Current ISP Does Not Support IPv6"을 참조하십시오.

소속된 조직이 ISP일 경우 적합한 인터넷 레지스트리에서 고객의 사이트 접두어를 획득합니다. 자세한 내용은 Internet Assigned Numbers Authority (IANA) (`http://www.iana.org`)를 참조하십시오.

IPv6 번호 지정 체계 만들기

제안된 IPv6 네트워크가 완전히 새로운 네트워크가 아니라면 기존 IPv4 토폴로지를 기반으로 IPv6 번호 지정 체계를 만드십시오.

노드에 대한 IPv6 주소 지정 계획 만들기

대부분의 호스트에서는 인터페이스에 대한 IPv6 주소의 Stateless 자동 구성이 적합한 시간 절약 전략입니다. 호스트가 가장 가까운 라우터로부터 사이트 접두어를 받으면 Neighbor Discovery가 호스트에 있는 각 인터페이스에 대한 IPv6 주소를 자동으로 생성합니다.

서버는 정적 IPv6 주소를 사용해야 합니다. 서버의 IPv6 주소를 수동으로 구성하지 않은 경우, 서버에서 NIC 카드가 교체될 때마다 새 IPv6 주소가 자동 구성됩니다. 서버 주소를 만들 때 다음 사항에 유의하십시오.

- 서버에 의미 있고 안정적인 인터페이스 ID를 제공합니다. 한 가지 전략은 인터페이스 ID에 순차적 번호 지정 체계를 사용하는 것입니다. 예를 들어 그림 2–1에 표시된 LDAP 서버의 내부 인터페이스는 `2001:db8:3c4d:2::2`가 될 수 있습니다.

- IPv4 네트워크의 번호를 정기적으로 재지정하지 않는 경우, 라우터 및 서버의 기존 IPv4 주소를 인터페이스 ID로 사용합니다. 그림 2–1에서 DMZ에 대한 라우터 1 인터페이스의 IPv4 주소는 `123.456.789.111`이라고 가정합니다. IPv4 주소를 16진수로 변환한 다음 그 결과를 인터페이스 ID로 사용할 수 있습니다. 새 인터페이스 ID는 `::7bc8:156F`입니다.

  ISP로부터 주소를 받은 것이 아니라 등록된 IPv4 주소를 소유한 경우에만 이 방법을 사용하십시오. ISP가 제공한 IPv4 주소를 사용하는 경우 종속성이 생기는데, 이 종속성으로 인해 ISP를 변경하면 문제가 발생할 수 있습니다.

IPv4 주소의 개수에는 제한이 있으므로 과거에는 네트워크 설계자가 등록된 전역 주소 및 개인 RFC 1918 주소를 사용할 위치를 고려해야 했습니다. 그러나 IPv6 주소에는 전역 및 개인 IPv4 주소의 개념이 적용되지 않습니다. 사이트 접두어를 포함하는 전역 유니캐스트 주소를 공개 DMZ를 비롯한 모든 네트워크 링크에 사용할 수 있습니다.

서브넷 번호 지정 체계 만들기

기존 IPv4 서브넷을 해당 IPv6 서브넷에 매핑하여 번호 지정 체계를 시작하십시오. 예를 들어 그림 2–1에 표시된 서브넷을 고려하십시오. 서브넷 1–4은 주소의 처음 16비트에 대해 RFC 1918 IPv4 개인 주소 지정을 사용합니다. 숫자 1–4는 서브넷을 나타냅니다. 설명을 위해 IPv6 접두어 refix `2001:db8:3c4d/48`가 사이트에 지정되었습니다.

다음 표는 개인 IPv4 접두어가 IPv6 접두어에 매핑되는 방식을 보여줍니다.

IPv4 서브넷 접두어	해당 IPv6 서브넷 접두어
`192.168.1.0/24`	`2001:db8:3c4d:1::/64`
`192.168.2.0/24`	`2001:db8:3c4d:2::/64`
`192.168.3.0/24`	`2001:db8:3c4d:3::/64`
`192.168.4.0/24`	`2001:db8:3c4d:4::/64`

IPv6을 지원하도록 네트워크 서비스 구성

현재 Oracle Solaris 릴리스에서 제공하는 다음과 같은 일반 IPv4 네트워크 서비스는 IPv6에서 사용할 수 있습니다.

- `sendmail`
- NFS
- HTTP(Apache 2 릴리스 또는 Orion)
- DNS
- LDAP

IMAP 메일 서버는 IPv4에서만 사용 가능합니다.

IPv6용으로 구성된 노드는 IPv4 서비스를 실행할 수 있습니다. IPv6을 설정할 경우 모든 서비스가 IPv6 연결을 수락하는 것은 아닙니다. IPv6으로 이식된 서비스만 연결을 수락합니다. IPv6으로 이식되지 않은 서비스는 계속 프로토콜 스택의 IPv4 절반에서 작동합니다.

서비스를 IPv6으로 업그레이드한 후 문제가 발생할 수 있습니다. 자세한 내용은 **Troubleshooting Network Issues**의 "Problems After Upgrading Services to IPv6"를 참조하십시오.

▼ IPv6을 지원하도록 네트워크 서비스를 준비하는 방법

1 **IPv6을 지원하도록 다음 네트워크 서비스를 업데이트합니다.**

- 메일 서버
- NIS 서버
- NFS

주 – LDAP은 IPv6 관련 구성 작업 없이 IPv6을 지원합니다.

---

2 **IPv6에서 방화벽 하드웨어를 사용할 수 있는지 확인합니다.**

지침은 해당 방화벽 관련 설명서를 참조하십시오.

3 **네트워크에 있는 다른 서비스가 IPv6으로 이식되었는지 확인합니다.**

자세한 내용은 소프트웨어의 마케팅 보조 자료 및 관련 설명서를 참조하십시오.

4 **사이트에서 다음 서비스를 배치하는 경우 이러한 서비스에 대해 적절한 조치를 취했는지 확인합니다.**

- 방화벽

  IPv6을 지원하기 위해 준비된 IPv4의 정책을 강화합니다. 보다 자세한 보안 고려 사항은 31 페이지 "IPv6 구현에 대한 보안 고려 사항"을 참조하십시오.

- 메일

  DNS용 MX 레코드의 경우 메일 서버의 IPv6 주소를 추가합니다.

- DNS

  DNS 관련 고려 사항은 30 페이지 "IPv6을 지원하도록 DNS를 준비하는 방법"을 참조하십시오.

- IPQoS

IPv4에 사용된 것과 동일한 Diffserv 정책을 호스트에 대해 사용합니다. 자세한 내용은 **Oracle Solaris 11.1에서 IP 서비스 품질 관리**의 "분류기 모듈"을 참조하십시오.

5 **해당 노드를 IPv6으로 변환하기 전에 노드에서 제공하는 네트워크 서비스를 감사합니다.**

▼ IPv6을 지원하도록 DNS를 준비하는 방법

현재 Oracle Solaris 릴리스는 클라이언트 측과 서버 측 모두에 대한 DNS 분석을 지원합니다. IPv6을 위해 DNS 서비스를 준비하려면 다음을 수행하십시오.

IPv6에 대한 DNS 지원과 관련된 자세한 내용은 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**을 참조하십시오.

1 **순환 이름 분석을 수행하는 DNS 서버가 듀얼 스택(IPv4 및 IPv6)인지 아니면 IPv4 전용인지 확인합니다.**

2 **DNS 서버에서 DNS 데이터베이스를 정방향 영역의 관련 IPv6 데이터베이스 AAAA 레코드로 채웁니다.**

주 – 중요한 서비스를 여러 개 실행하는 서버의 경우 특별한 주의가 필요합니다. 네트워크가 제대로 작동하는지 확인하십시오. 또한 중요한 서비스가 모두 IPv6으로 이식되었는지도 확인하십시오. 그런 다음 서버의 IPv6 주소를 DNS 데이터베이스에 추가하십시오.

---

3 **AAAA 레코드의 연관된 PTR 레코드를 역방향 영역에 추가합니다.**

4 **영역에 대해 설명하는 NS 레코드에 IPv4 전용 데이터 또는 IPv6 및 IPv4 데이터를 추가합니다.**

네트워크에서 터널 사용 계획

사용자의 네트워크가 IPv4 및 IPv6으로 마이그레이션되므로 IPv6 구현은 전환 방식으로 사용될 여러 터널 구성을 지원합니다. 터널을 통해 분리된 IPv6 네트워크가 통신할 수 있게 됩니다. 대부분의 인터넷은 IPv4를 실행하므로, 사용자 사이트의 IPv6 패킷은 인터넷에서 터널을 통과하여 대상 Pv6 네트워크로 이동합니다.

다음은 IPv6 네트워크 토폴로지에서 터널을 사용하기 위한 몇 가지 주요 시나리오입니다.

- IPv6 서비스를 구매한 ISP는 사이트의 경계 라우터에서 ISP 네트워크로 연결되는 터널을 만들 수 있도록 해줍니다. 그림 2–1는 이러한 터널을 보여줍니다. 이 경우 IPv4 터널을 통해 수동 IPv6을 실행합니다.

- IPv4 연결로 분산된 대형 네트워크를 관리합니다. IPv6을 사용하는 분산된 사이트를 연결하려면 각 서브넷의 에지 라우터에서 자동 6to4 터널을 실행하면 됩니다.
- 기반구조의 라우터를 IPv6으로 업그레이드할 수 없는 경우도 있습니다. 이 경우 두 개의 IPv6 라우터를 끝점으로 사용하여 IPv4 라우터를 통과하는 수동 터널을 만들 수 있습니다.

터널 구성 절차는 106 페이지 "터널 구성(작업 맵)"을 참조하십시오. 터널과 관련된 개념 정보는 97 페이지 "IP 터널 개요"를 참조하십시오.

IPv6 구현에 대한 보안 고려 사항

IPv6을 기존 네트워크에 사용할 경우 사이트의 보안이 손상되지 않도록 유의해야 합니다. IPv6 구현을 도입할 때 다음 보안 문제에 유의하십시오.

- IPv6 패킷과 IPv4 패킷 모두에 대해 동일한 양의 필터링이 필요합니다.
- IPv6 패킷은 대개 방화벽을 통해 터널링됩니다. 따라서 다음 시나리오 중 하나로 구현해야 합니다.
  - 방화벽이 터널 내에서 컨텐츠 검사를 수행하도록 합니다.
  - 반대쪽 터널 끝점에 동일한 규칙을 사용하는 IPv6 방화벽을 배치합니다.
- IPv6 - UDP - IPv4 터널을 사용하는 전환 방식이 존재합니다. 이러한 방식은 방화벽을 방해하므로 위험합니다.
- IPv6 노드는 엔터프라이즈 네트워크 외부에서 전역적으로 연결할 수 있습니다. 보안 정책이 공개 액세스를 금지하는 경우 방화벽에 대해 보다 엄격한 규칙을 설정해야 합니다. 예를 들어 Stateful 방화벽 구성을 고려하십시오.

이 설명서는 IPv6 구현 내에서 사용할 수 있는 보안 기능을 다룹니다.

- IP 보안 아키텍처(IPsec) 기능을 통해 IPv6 패킷에 대한 암호화된 보호를 제공할 수 있습니다. 자세한 내용은 **Oracle Solaris 11.1의 네트워크 보안**의 6 장, "IP 보안 아키텍처(개요)"를 참조하십시오.
- IKE(Internet Key Exchange) 기능을 통해 IPv6 패킷에 대한 공개 키 인증을 사용할 수 있습니다. 자세한 내용은 **Oracle Solaris 11.1의 네트워크 보안**의 9 장, "Internet Key Exchange(개요)"을 참조하십시오.

◆ ◆ ◆ 3 장

3 IPv4 네트워크 구성

네트워크 구성은 하드웨어 어셈블 단계와 데몬, 파일 및 TCP/IP 프로토콜을 구현하는 서비스에 대한 구성 단계로 진행됩니다.

이 장에서는 IPv4 주소 지정 및 서비스를 구현하는 네트워크에 대한 구성 방법을 설명합니다.

이 장에서 설명되는 대부분의 작업은 IPv4 전용 및 IPv6 사용 네트워크에 모두 적용됩니다. IPv6 네트워크에만 적용되는 작업은 4 장, "네트워크에서 IPv6 사용"에서 설명됩니다.

주 – TCP/IP를 구성하기 전에 1 장, "네트워크 배치 계획"에 나열되는 다양한 계획 작업을 검토하십시오. IPv6 주소를 사용하려면 2 장, "IPv6 주소 사용 시 고려 사항"도 참조하십시오.

이 장은 다음 정보를 포함합니다.

네트워크 구성(작업 맵)

다음 표에서는 서브넷이 없는 네트워크 구성에서 서브넷을 사용하는 네트워크로 변경한 후 수행할 추가 작업을 나열합니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
시스템의 IP 인터페이스를 구성합니다.	시스템의 IP 인터페이스에 IP 주소를 지정합니다.	**Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"
로컬 파일 모드에 대한 시스템을 구성합니다.	시스템의 /etc 디렉토리에 있는 특정 구성 파일을 편집하고 nis/domain SMF 서비스를 구성합니다.	40 페이지 "로컬 파일 모드에 대한 시스템 구성 방법"
네트워크 구성 서버를 설정합니다.	in.tftp 데몬을 사용으로 설정하고 시스템의 /etc 디렉토리에 있는 다른 구성 파일을 편집합니다.	42 페이지 "네트워크 구성 서버 설정 방법"
네트워크 클라이언트 모드에 대한 시스템을 구성합니다.	시스템의 /etc 디렉토리에 있는 구성 파일을 편집합니다.	41 페이지 "네트워크 클라이언트 모드에 대한 시스템 구성 방법"
네트워크 클라이언트에 대한 경로 지정 전략을 지정합니다.	정적 경로 지정 또는 동적 경로 지정을 사용하도록 시스템을 구성합니다.	51 페이지 "단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법" 및 53 페이지 "단일 인터페이스 시스템에서 동적 경로 지정을 사용으로 설정하는 방법".

네트워크 구성을 시작하기 전에

이 Oracle Solaris 릴리스에서 시스템의 네트워크 구성은 활성 **네트워크 구성 프로파일***(NCP)*을 통해 관리됩니다. 활성 NCP가 반응적인 경우(예: automatic NCP) 시스템의 네트워크 구성이 자동으로 수행됩니다. 활성 NCP가 DefaultFixed이면 시스템의 네트워크 구성 모드가 고정됩니다. 반응적 네트워크 구성의 시스템은 고정된 네트워크 구성의 시스템과 다르게 동작합니다.

수행하는 모든 구성은 활성 NCP에 적용됩니다. 따라서 구성 절차를 수행하기 전에 먼저 어느 NCP가 올바르게 활성 상태인지 확인해야 합니다. 따라서 시스템은 구성 절차를 완료한 후 예상한 대로 동작합니다. 시스템에서 활성 상태인 NCP를 확인하려면 다음 명령을 입력합니다.

```
# netadm list
TYPE        PROFILE        STATE
ncp         DefaultFixed   online
ncp         Automatic      disabled
loc         Automatic      offline
loc         NoNet          offline
loc         User           offline
loc         DefaultFixed   online
```

상태가 온라인으로 나열된 프로파일이 시스템에서 활성 NCP입니다.

시스템에서 NCP에 대한 자세한 내용을 보려면 `netadm` 명령에 `-x` 옵션을 사용합니다.

```
netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         DefaultFixed   online         active
ncp         Automatic      disabled       disabled by administrator
loc         Automatic      offline        conditions for activation are unmet
loc         NoNet          offline        conditions for activation are unmet
loc         User           offline        conditions for activation are unmet
loc         DefaultFixed   online         active
```

프로파일 유형 간에 전환하려면(예: 반응적 프로파일에서 고정된 프로파일로 전환) 다음 명령을 입력합니다.

```
# netadm enable -p ncp NCP-name
```

여기서 *NCP-name*은 NCP 유형의 이름입니다.

프로파일로 관리되는 네트워크 구성에 대한 소개 정보를 보려면 **Oracle Solaris 11 네트워킹 소개**의 "네트워크 구성 프로파일"을 참조하십시오. NCP에 대한 자세한 내용은 **Oracle Solaris 11.1에서 반응적 네트워크 구성을 사용하여 시스템 연결**을 참조하십시오.

네트워크의 구성 요소 시스템 구성

네트워크 시스템을 구성할 때는 다음 구성 정보가 필요합니다.

- 각 시스템의 호스트 이름
- 각 시스템의 IP 주소 및 넷마스크. 네트워크가 서브넷으로 세분화된 경우 개별 넷마스크를 비롯하여 서브넷 번호와 각 서브넷의 시스템에 적용할 IP 주소 스키마가 있어야 합니다.
- 각 시스템이 속한 도메인 이름
- 기본 라우터 주소

  각 네트워크에 연결된 라우터가 하나뿐인 간단한 네트워크 토폴로지를 사용하는 경우 이 정보를 제공합니다. 라우터가 RDISC(Router Discovery Server Protocol), RIP(Router Information Protocol) 등의 경로 지정 프로토콜을 실행하지 않는 경우에도 이 정보를 제공합니다. Oracle Solaris에서 지원하는 경로 지정 프로토콜 목록뿐만 아니라 라우터에 대한 자세한 내용은 121 페이지 "Oracle Solaris의 경로 지정 프로토콜"을 참조하십시오.

주 – Oracle Solaris를 설치하는 동안 네트워크를 구성할 수 있습니다. 지침은 **Oracle Solaris 11.1 시스템**를 참조하십시오.

본 설명서의 절차에서는 OS를 설치한 후 네트워크를 구성 중인 것으로 간주합니다.

네트워크의 구성 요소 시스템을 구성하려면 다음 절의 그림 3–1을 참조하십시오.

IPv4 자율 시스템 토폴로지

일반적으로 라우터와 네트워크가 여러 개인 사이트에서는 네트워크 토폴로지를 단일 경로 지정 도메인 또는 **자율 시스템***(AS)*으로 관리합니다.

그림 3–1 IPv4 라우터가 여러 개인 자율 시스템

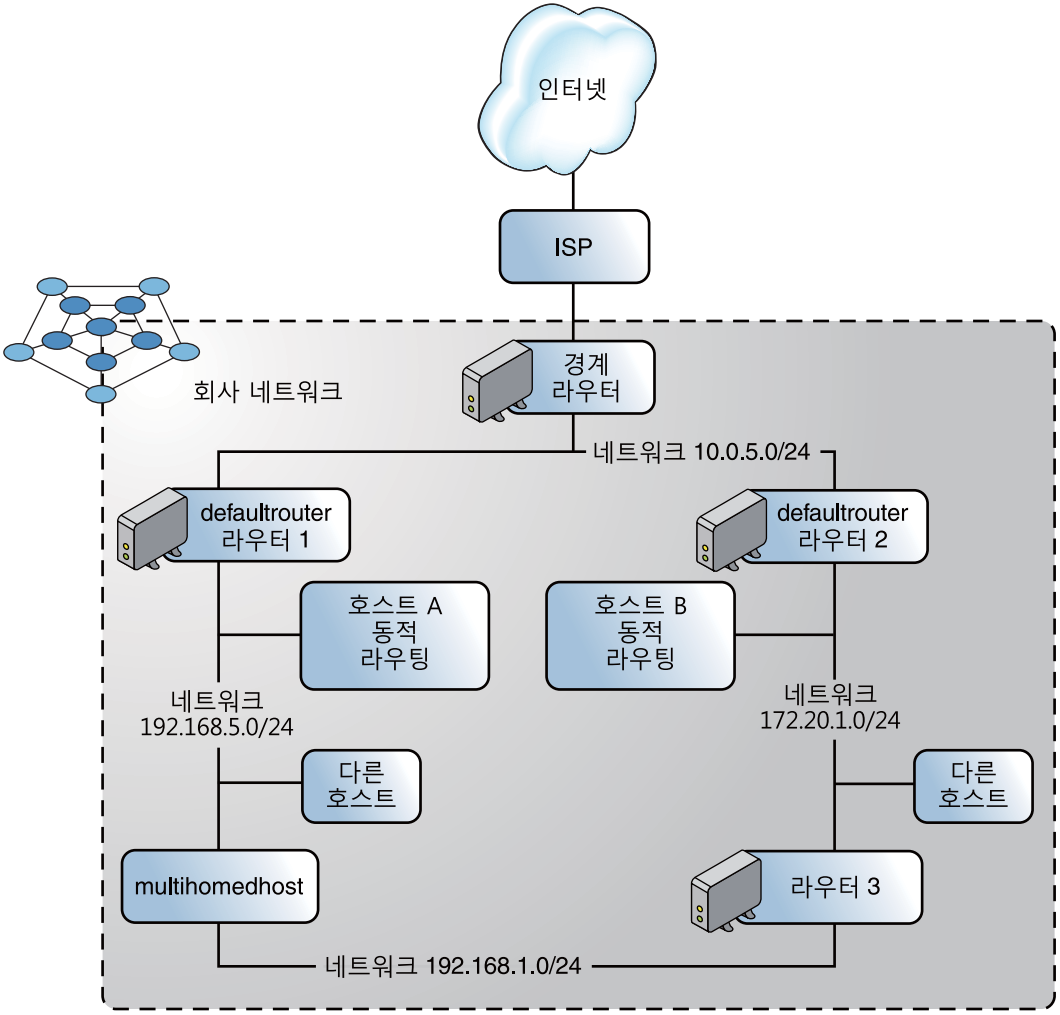


그림 3–1은 세 개의 로컬 네트워크(`10.0.5.0`, `172.16.1.0` 및 `192.168.5.0`)로 구분된 AS를 보여 줍니다. 네트워크는 다음 유형의 시스템으로 구성됩니다.

- 라우터는 경로 지정 프로토콜을 사용하여 네트워크 패킷이 소스에서 로컬 네트워크 내의 대상 또는 외부 네트워크로 지정되거나 경로 지정되는 방식을 관리합니다. Oracle Solaris에서 지원되는 경로 지정 프로토콜에 대한 자세한 내용은 122 페이지 “Oracle Solaris의 경로 지정 프로토콜 표”를 참조하십시오.

  라우터의 유형은 다음과 같습니다.

  - **경계 라우터**는 외부적으로 로컬 네트워크(예: `10.0.5.0`)를 서비스 공급자에 연결합니다.

  - **기본 라우터**는 여러 로컬 네트워크를 자체적으로 포함할 수 있는 로컬 네트워크에서 패킷 경로 지정을 관리합니다. 예를 들어, 그림 3–1에서 Router 1은 `192.168.5`에 대한 기본 라우터로 사용됩니다. 동시에 Router 1은 `10.0.5.0` 내부 네트워크에도 연결됩니다. Router 2의 인터페이스는 `10.0.5.0` 및 `172.16.1.0` 내부 네트워크에 연결됩니다.
  - **패킷 전달 라우터**는 내부 네트워크 간에 패킷을 전달하지만 경로 지정 프로토콜을 실행하지 않습니다. 그림 3–1에서 Router 3은 `172.16.1` 및 `192.168.5` 네트워크에 연결된 패킷 전달 라우터입니다.
- 클라이언트 시스템
  - 멀티홈 시스템 또는 NIC가 여러 개인 시스템 Oracle Solaris에서 이러한 시스템은 기본적으로 패킷을 동일한 네트워크 세그먼트 내 다른 시스템으로 전달할 수 있습니다.
  - 단일 인터페이스 시스템은 패킷 전달 및 수신 구성 정보에 로컬 라우터를 사용합니다.

시스템 구성 모드 설정

이 절에서는 **로컬 파일 모드** 또는 **네트워크 클라이언트 모드**에서 실행할 시스템을 설정하는 절차에 대해 설명합니다. 로컬 파일 모드에서 실행하는 경우 시스템은 로컬 디렉토리에 있는 파일에서 모든 TCP/IP 구성 정보를 가져옵니다. 네트워크 클라이언트 모드에서는 원격 네트워크 구성 서버가 네트워크의 모든 시스템에 구성 정보를 제공합니다.

일반적으로 다음과 같은 네트워크의 서버는 로컬 파일 모드에서 실행됩니다.

- 네트워크 구성 서버
- NFS 서버
- NIS, LDAP 또는 DNS 서비스를 제공하는 이름 서버
- 메일 서버
- 라우터

클라이언트는 두 모드 중 하나에서 실행할 수 있습니다. 따라서 네트워크에서는 다음 그림과 같이 구성된 다양한 시스템에서 이러한 모드의 조합이 사용될 수 있습니다.

그림 3–2 IPv4 네트워크 토폴로지 시나리오의 시스템

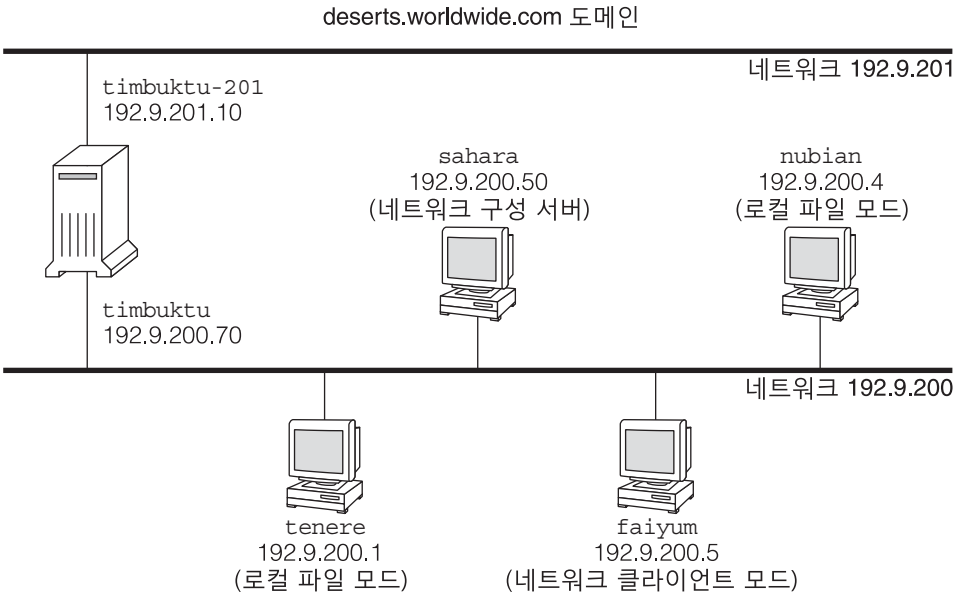


그림 3–2에서는 192.9.200 네트워크의 시스템을 보여 줍니다.

- 모든 시스템이 조직 도메인 deserts.worldwide.com에 속합니다.
- sahara는 구성 서버입니다. 서버로서, 시스템의 로컬 디스크에서 TCP/IP 구성 정보를 가져오는 로컬 파일 모드로 실행됩니다.

주 – 네트워크 클라이언트 모드에서 실행하도록 클라이언트를 구성할 경우 해당 클라이언트에 구성 정보를 제공할 네트워크 구성 서버를 하나 이상 구성해야 합니다.

- tenere, nubian 및 faiyum은 네트워크의 클라이언트입니다. tenere 및 nubian은 로컬 파일 모드에서 실행됩니다. faiyum의 로컬 디스크에 관계없이 시스템은 네트워크 클라이언트 모드에서 작동하도록 구성됩니다.
- timbuktu는 라우터로 구성되므로 로컬 파일 모드에서 작동합니다. 시스템에는 각각 고유하게 구성된 IP 인터페이스가 있는 두 개의 NIC가 포함되어 있습니다. 첫번째 IP 인터페이스는 이름이 timbuktu이며 192.9.200 네트워크에 연결됩니다. 두번째 IP 인터페이스는 이름이 timbuktu-201이며 192.9.201 네트워크에 연결됩니다.

▼ 로컬 파일 모드에 대한 시스템 구성 방법

이 절차에 따라 로컬 파일 모드로 실행할 시스템을 구성할 수 있습니다.

1 **지정된 IP 주소로 시스템의 IP 인터페이스를 구성합니다.**

절차는 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

2 **/etc/nodename 파일에서 올바른 호스트 이름이 설정되었는지 확인합니다.**

3 **/etc/inet/hosts 파일의 항목이 최신인지 확인합니다.**

Oracle Solaris 설치 프로그램이 기본 네트워크 인터페이스, 루프백 주소 및 설치 중 구성된 추가 인터페이스(해당하는 경우)에 대한 항목을 만듭니다.

파일에는 기본 라우터의 이름 및 라우터의 IP 주소도 포함되어야 합니다.

a. **(옵션) 설치 후 시스템에 추가된 네트워크 인터페이스에 대한 IP 주소 및 해당 이름을 추가합니다.**

b. **(옵션) /usr 파일 시스템이 NFS 마운트된 시스템인 경우 파일 서버의 IP 주소를 추가합니다.**

4 **nis/domain SMF 서비스의 등록 정보로 시스템의 정규화된 도메인을 지정합니다.**

예를 들어, 다음과 같이 deserts.worldwide.com을 nis/domain SMF 서비스의 domainname 등록 정보에 대한 값으로 지정합니다.

```
# domainname domainname
```

이 단계는 지속적인 변경 사항에 영향을 줍니다.

5 **/etc/defaultrouter 파일에 라우터의 이름을 입력합니다.**

6 **해당하는 경우 넷마스크 정보를 추가합니다.**

주 – DHCP 서비스를 사용 중인 경우 이 단계를 건너 뛰십시오.

---

a. **/etc/inet/netmasks 파일에 네트워크 번호 및 넷마스크를 입력합니다.**

항목을 만들려면 *network-number netmask* 형식을 사용합니다. 예를 들어, 클래스 C 네트워크 번호 192.168.83의 경우 다음과 같이 입력합니다.

```
192.168.83.0     255.255.255.0
```

CIDR 주소의 경우 네트워크 접두어를 동등한 점으로 구분된 십진수 표현으로 변환합니다. 네트워크 접두어 및 동등한 점으로 구분된 십진수 표현은 표 1–1에서 확인할 수 있습니다. 예를 들어, CIDR 네트워크 접두어 `192.168.3.0/22`를 표현하려면 다음을 사용합니다.

```
192.168.3.0     255.255.252.0
```

b. 로컬 파일이 먼저 검색되도록 스위치의 SMF 등록 정보에서 넷마스크에 대한 조회 순서를 변경한 다음 인스턴스를 새로 고칩니다.

```
# svccfg -s name-service/switch setprop config/host = astring: '"files nis"'
# svccfg -s name-service/switch:default refresh
```

7 시스템을 재부트합니다.

▼ 네트워크 클라이언트 모드에 대한 시스템 구성 방법

네트워크 클라이언트 모드에서 구성할 각 호스트에서 다음 절차를 수행합니다.

시작하기 전에 네트워크 클라이언트는 네트워크 구성 서버에서 구성 정보를 수신합니다. 따라서 시스템을 네트워크 클라이언트로 구성하기 전에 네트워크에 대해 하나 이상의 네트워크 구성 서버가 설정되었는지 확인해야 합니다.

1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2 지정된 IP 주소로 시스템의 IP 인터페이스를 구성합니다.

절차는 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

3 `/etc/inet/hosts` 파일에 루프백 네트워크 인터페이스의 `localhost` 이름 및 IP 주소만 포함되어 있는지 확인합니다.

```
# cat /etc/inet/hosts
# Internet host table
#
127.0.0.1       localhost
```

4 `nis/domain` SMF 서비스의 `domainname` 등록 정보에 지정된 값을 제거합니다.

```
# domainname ''
```

이 단계는 지속적인 변경 사항에 영향을 줍니다.

5 클라이언트 `name-service/switch` 서비스의 검색 경로가 네트워크에 대한 동일한 서비스 요구 사항을 반영하는지 확인합니다.

▼ 네트워크 구성 서버 설정 방법

설치 서버 및 부트 서버 설정 정보는 **Oracle Solaris 11.1 시스템**에서 확인할 수 있습니다.

1. **관리자로 전환합니다.**

   자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2. **다음과 같이 `in.tftpd` 데몬을 켭니다.**

   a. **지정된 네트워크 구성 서버의 루트(/) 디렉토리로 이동합니다.**

   b. **다음과 같이 `/tftpboot` 디렉토리를 만듭니다.**

      ```
      # mkdir /tftpboot
      ```

      이 명령은 시스템을 TFTP, bootparams 및 RARP 서버로 구성합니다.

   c. **디렉토리에 대한 심볼릭 링크를 만듭니다.**

      ```
      # ln -s /tftpboot/. /tftpboot/tftpboot
      ```

3. **`/etc/inetd.conf` 파일에서 `tftp` 행을 추가합니다.**

   행이 다음과 같이 표시됩니다.

   ```
   tftp dgram udp6 wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
   ```

   이 행은 `in.tftpd`가 `/tftpboot`에 있는 파일 이외의 다른 파일을 검색하지 않도록 합니다.

4. **`/etc/hosts` 데이터베이스에서 네트워크의 모든 클라이언트에 대한 호스트 이름 및 IP 주소를 추가합니다.**

5. **`/etc/ethers` 데이터베이스에서 네트워크 클라이언트 모드로 실행되는 네트워크의 모든 서버에 대한 항목을 만듭니다.**

   이 데이터베이스의 항목은 다음 형식을 사용합니다.

   MAC Address *host name* *#comment*

   자세한 내용은 `ethers(4)` 매뉴얼 페이지를 참조하십시오.

6. **`/etc/bootparams` 데이터베이스에서 네트워크 클라이언트 모드로 실행되는 네트워크의 모든 시스템에 대한 항목을 만듭니다.**

   이 데이터베이스 편집에 대한 자세한 내용은 `bootparams(4)` 매뉴얼 페이지를 참조하십시오.

7. **`/etc/inetd.conf` 항목을 SMF(서비스 관리 기능) 서비스 매니페스트로 변환하고 결과 서비스를 사용으로 설정합니다.**

   ```
   # /usr/sbin/inetconv
   ```

8 **in.tftpd가 제대로 작동 중인지 확인합니다.**

```
# svcs network/tftp/udp6
```

출력이 다음과 유사하게 표시됩니다.

```
STATE          STIME    FMRI
online         18:22:21 svc:/network/tftp/udp6:default
```

자세한 정보 **in.tftpd 데몬 관리**

in.tftpd 데몬은 서비스 관리 기능을 통해 관리됩니다. in.tftpd에 대한 관리 작업(예: 사용으로 설정, 사용 안함으로 설정 또는 다시 시작)은 svcadm 명령을 사용하여 수행할 수 있습니다. 이 서비스에 대한 시작 및 다시 시작 권한은 inetd로 위임됩니다. inetadm 명령을 사용하여 구성을 변경하고 in.tftpd에 대한 구성 정보를 볼 수 있습니다. svcs 명령을 사용하여 서비스 상태를 질의할 수 있습니다. 서비스 관리 기능의 개요는 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 1 장, "서비스 관리(개요)"를 참조하십시오.

IPv4 라우터 구성

라우터는 두 개 이상의 네트워크 간의 인터페이스를 제공합니다. 따라서 라우터의 물리적 네트워크 인터페이스 각각에 고유한 이름과 IP 주소를 지정해야 합니다. 즉, 각 라우터에는 기본 네트워크 인터페이스와 연관된 호스트 이름과 IP 주소를 비롯하여 추가 네트워크 인터페이스 각각에 대한 하나 이상의 고유한 이름과 IP 주소가 있는 것입니다.

다음 절차에 따라 물리적 인터페이스가 하나뿐인 시스템(기본적으로 호스트)을 라우터로 구성할 수도 있습니다. **Oracle Solaris 11.1에서 UUCP 및 PPP를 사용하여 직렬 네트워크 관리**의 "다이얼 업 PPP 링크 계획"에 설명된 대로 시스템이 PPP 링크에서 하나의 끝점으로 사용되는 경우 단일 인터페이스 시스템을 라우터로 구성할 수 있습니다.

▼ IPv4 라우터 구성 방법

다음 지침에서는 설치 후 라우터에 대한 인터페이스를 구성 중인 것으로 간주합니다.

시작하기 전에 라우터가 네트워크에 물리적으로 설치된 후 40 페이지 "로컬 파일 모드에 대한 시스템 구성 방법"에 설명된 대로 로컬 파일 모드에서 작동하도록 라우터를 구성합니다. 이 구성은 네트워크 구성 서버의 작동이 중지된 경우 라우터가 부트되도록 합니다.

1 **관리자로 전환합니다.**

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2 **시스템의 NIC에서 IP 인터페이스를 구성합니다.**

IP 인터페이스를 구성하기 위한 세부 단계는 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

각 IP 인터페이스는 시스템에서 패킷을 경로 지정할 네트워크의 IP 주소로 구성되어 있어야 합니다. 따라서 시스템이 `192.168.5.0` 및 `10.0.5.0` 네트워크를 제공하는 경우 각 네트워크에 대해 하나의 NIC를 구성해야 합니다.

주의 – DHCP를 사용하도록 IPv4 라우터를 구성하려면 DHCP 관리를 철저히 파악하고 있어야 합니다.

3 **`/etc/inet/hosts` 파일에 각 인터페이스의 호스트 이름 및 IP 주소를 추가합니다.**

예를 들어, Router 1의 두 인터페이스에 대해 지정된 이름이 각각 `krakatoa`와 `krakatoa-1`이라고 가정합니다. 이 경우 `/etc/inet/hosts` 파일의 항목은 다음과 같습니다.

```
192.168.5.1      krakatoa        #interface for network 192.168.5.0
10.0.5.1         krakatoa-1      #interface for network 10.0.5.0
```

4 **나머지 단계를 수행하여 로컬 파일 모드에서 실행되도록 이 라우터를 구성합니다.**

40 페이지 "로컬 파일 모드에 대한 시스템 구성 방법"을 참조하십시오.

5 **라우터가 서브넷 네트워크에 연결된 경우 `/etc/inet/netmasks` 파일에 네트워크 번호 및 넷마스크를 추가합니다.**

예를 들어, 추가 IPv4 주소 표기법(예: `192.168.5.0`)의 경우 다음과 같이 입력합니다.

```
192.168.5.0     255.255.255.0
```

6 **라우터에서 IPv4 패킷 전달을 사용으로 설정합니다.**

```
# ipadm set-prop -p forwarding=on ipv4
```

7 **(옵션) 경로 지정 프로토콜을 시작합니다.**

다음 명령 구문 중 하나를 사용합니다.

- `# routeadm -e ipv4-routing -u`
- `# svcadm enable route:default`

  `in.routed` 데몬과 연관된 SMF FMRI는 `svc:/network/routing/route`입니다.

경로 지정 프로토콜을 시작하면 경로 지정 데몬 `/usr/sbin/in.routed`가 자동으로 경로 지정 테이블을 업데이트합니다. 이 프로세스를 **동적 경로 지정**이라고 합니다. 경로 지정 유형에 대한 자세한 내용은 46 페이지 "경로 지정 테이블 및 경로 지정 유형"을 참조하십시오. `routeadm` 명령에 대한 자세한 내용은 `routeadm(1M)` 매뉴얼 페이지를 참조하십시오.

예 3–1 네트워크에 대한 기본 라우터 구성

이 예는 그림 3–1을 기반으로 합니다. Router 2에는 두 개의 유선 네트워크 연결(172.16.1.0 네트워크에 대한 연결과 10.0.5.0 네트워크에 대한 연결)이 포함되어 있습니다. 예에서는 172.16.1.0 네트워크의 기본 라우터가 되도록 Router 2를 구성하는 방법을 보여 줍니다. 또한 예에서는 40 페이지 "로컬 파일 모드에 대한 시스템 구성 방법"에 설명된 대로 Router 2가 로컬 파일 모드에서 작동하도록 구성되었다고 간주합니다.

수퍼유저 또는 동등한 역할의 사용자로 로그인한 후 시스템 인터페이스의 상태를 확인합니다.

```
# dladm show-link
LINK     CLASS     MTU     STATE   BRIDGE   OVER
net0     phys      1500    up      --       --
net1     phys      1500    up      --       --
net2     phys      1500    up      --       --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           172.16.1.10/24
```

net0만 IP 주소로 구성되었습니다. Router 2를 기본 라우터로 설정하려면 물리적으로 net1 인터페이스를 10.0.5.0 네트워크에 연결합니다.

```
# ipadm create-ip net1
# ipadm create-addr -a 10.0.5.10/24 net1
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           172.16.1.10/24
net1/v4           static   ok           10.0.5.10/24
```

그런 다음 새로 구성된 인터페이스 및 연결된 네트워크에 대한 정보로 다음 네트워크 데이터베이스를 업데이트합니다.

```
# vi /etc/inet/hosts
127.0.0.1       localhost
172.16.1.10       router2       #interface for network 172.16.1
10.0.5.10         router2-out   #interface for network 10.0.5
# vi /etc/inet/netmasks
172.16.1.0    255.255.255.0
10.0.5.0      255.255.255.0
```

마지막으로 패킷 전달 및 in.routed 경로 지정 데몬을 사용으로 설정합니다.

```
# ipadm set-prop -p forwarding=on ipv4
# svcadm enable route:default
```

그러면 RIP를 통한 IPv4 패킷 전달 및 동적 경로 지정이 Router 2에서 사용으로 설정되었지만, 172.16.1.0 네트워크에 대한 기본 라우터 구성은 아직 완료되지 않은 것입니다. 다음 작업을 수행해야 합니다.

- 호스트가 새 기본 라우터에서 경로 지정 정보를 가져오도록 172.16.1.0 네트워크에서 각 호스트를 수정합니다. 자세한 내용은 51 페이지 "단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법"을 참조하십시오.
- Router 2의 경로 지정 테이블에서 경계 라우터에 대한 정적 경로 지정을 정의합니다. 자세한 내용은 46 페이지 "경로 지정 테이블 및 경로 지정 유형"을 참조하십시오.

경로 지정 테이블 및 경로 지정 유형

라우터와 호스트는 모두 **경로 지정 테이블**에서 유지 관리됩니다. 경로 지정 테이블에는 시스템의 로컬 기본 네트워크를 비롯하여 시스템에서 인식한 네트워크의 IP 주소가 나열됩니다. 알려진 각 네트워크에 대한 게이트웨이 시스템의 IP 주소도 나열됩니다. **게이트웨이**는 송신 패킷을 수신하여 로컬 네트워크 외부의 한 홉으로 전달할 수 있습니다.

다음은 IPv4 전용 네트워크의 시스템에 대한 간단한 경로 지정 테이블입니다.

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.16.1.10          UG        1    532   net0
224.0.0.0            10.0.5.100           U         1      0   net1
10.0.0.0             10.0.5.100           U         1      0   net1
127.0.0.1            127.0.0.1            UH        1     57   lo0
```

Oracle Solaris 시스템에서는 두 가지 유형(정적 및 동적)의 경로 지정을 구성할 수 있습니다. 단일 시스템에서 경로 지정 유형 중 하나 또는 두 가지 모두를 구성할 수 있습니다. **동적 경로 지정**을 구현하는 시스템은 경로 지정 프로토콜(IPv4 네트워크의 경우 RIP, IPv6 네트워크의 경우 RIPng)을 사용하여 네트워크 트래픽을 경로 지정하고 테이블의 경로 지정 정보를 업데이트합니다. **정적 경로 지정**을 사용하는 경우 route 명령을 사용하여 수동으로 경로 지정 정보를 유지 관리합니다. 자세한 내용은 route(1M) 매뉴얼 페이지를 참조하십시오.

로컬 네트워크 또는 자율 시스템에 대한 경로 지정을 구성할 때는 특정 라우터 및 호스트에서 지원할 경로 지정 유형을 고려하십시오.

다음 표에서는 다양한 경로 지정 유형과 각 경로 지정 유형이 최적으로 적용되는 네트워킹 시나리오를 보여 줍니다.

경로 지정 유형	최적 사용 사례
정적	작은 규모의 네트워크, 기본 라우터에서 경로를 가져오는 호스트, 다음 홉에서 하나 또는 두 개의 라우터에 대해서만 인식해야 할 기본 라우터
동적	보다 큰 규모의 인터네트워크, 호스트가 여러 개인 로컬 네트워크의 라우터, 큰 자율 시스템의 호스트. 거의 모든 네트워크의 시스템에 동적 경로 지정을 선택하는 것이 좋습니다.

경로 지정 유형	최적 사용 사례
정적과 동적 결합	정적으로 경로 지정된 네트워크와 동적으로 경로 지정된 네트워크를 연결하는 라우터, 내부 자율 시스템을 외부 네트워크와 연결하는 경계 라우터. 시스템에서 정적 경로 지정과 동적 경로 지정을 결합하여 사용하는 것이 일반적입니다.

그림 3–1에 표시된 AS는 정적 경로 지정과 동적 경로 지정을 결합한 것입니다.

주 – 시스템에서는 동일한 대상에 대한 두 경로를 통해 자동으로 로드 균형 조정 또는 페일오버를 수행하지 않습니다. 이러한 기능이 필요하면 **Oracle Solaris 11.1 네트워크 성능 관리**의 5 장, “IPMP 소개”에 설명된 대로 IPMP를 사용합니다.

▼ 경로 지정 테이블에 정적 경로 지정을 추가하는 방법

1 경로 지정 테이블의 현재 상태를 확인합니다.

일반 사용자 계정으로 다음 형식의 `netstat` 명령을 실행합니다.

```
% netstat -rn
```

출력이 다음과 유사하게 표시됩니다.

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
192.168.5.125        192.168.5.10          U      1   5879   net0
224.0.0.0            198.168.5.10          U      1  0       net0
default              192.168.5.10          UG     1  91908
127.0.0.1            127.0.0.1             UH     1  811302   lo0
```

2 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.

3 (옵션) 경로 지정 테이블의 기존 항목을 비웁니다.

```
# route flush
```

4 시스템 재부트 시 지속되는 경로를 추가합니다.

```
# route -p add -net network-address -gateway gateway-address
```

`-p`	시스템 재부트 시 지속되어야 할 경로를 만듭니다. 경로를 현재 세션에만 적용하려면 `-p` 옵션을 사용하지 마십시오.
`-net` *network-address*	경로가 *network-address*의 주소를 사용하는 네트워크로 이동하도록 지정합니다.
`-gateway` *gateway-address*	지정된 경로에 대한 게이트웨이 시스템의 IP 주소가 *gateway-address*임을 나타냅니다.

예 3–2 경로 지정 테이블에 정적 경로 지정 추가

다음 예에서는 그림 3–1의 Router 2에 정적 경로 지정을 추가하는 방법을 보여 줍니다. 정적 경로 지정은 AS의 경계 라우터 10.0.5.150에 필요합니다.

Router 2의 경로 지정 테이블을 보려면 다음을 입력합니다.

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.16.1.10          UG        1    249 ce0
224.0.0.0            172.16.1.10          U         1      0 ce0
10.0.5.0             10.0.5.20            U         1     78 bge0
127.0.0.1            127.0.0.1            UH        1     57 lo0
```

경로 지정 테이블은 Router 2가 인식하는 두 경로를 나타냅니다. 기본 경로는 Router 2의 172.16.1.10 인터페이스를 게이트웨이로 사용합니다. 두번째 경로 10.0.5.0은 Router 2에서 실행되는 in.routed 데몬을 통해 검색되었습니다. 이 경로에 대한 게이트웨이는 IP 주소가 10.0.5.20인 Router 1입니다.

게이트웨이가 경계 라우터인 10.0.5.0 네트워크에 두번째 경로를 추가하려면 다음을 입력합니다.

```
# route -p add -net 10.0.5.0/24 -gateway 10.0.5.150
add net 10.0.5.0: gateway 10.0.5.150
```

그러면 경로 지정 테이블에 IP 주소가 10.0.5.150/24인 경계 라우터에 대한 경로가 포함됩니다.

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.16.1.10          UG        1    249 ce0
224.0.0.0            172.16.1.10          U         1      0 ce0
10.0.5.0             10.0.5.20            U         1     78 bge0
10.0.5.0             10.0.5.150           U         1    375 bge0
127.0.0.1            127.0.0.1            UH        1     57 lo0
```

멀티홈 호스트 구성

Oracle Solaris에서는 인터페이스가 두 개 이상인 시스템을 **멀티홈 호스트**로 간주합니다. 멀티홈 호스트의 인터페이스는 다른 물리적 네트워크 또는 동일한 물리적 네트워크의 서로 다른 서브넷에 연결됩니다.

여러 인터페이스가 동일한 서브넷에 연결되는 시스템에서는 먼저 인터페이스를 하나의 IPMP 그룹으로 구성해야 합니다. 그렇지 않으면 시스템이 멀티홈 호스트가 될 수 없습니다. IPMP에 대한 자세한 내용은 **Oracle Solaris 11.1 네트워크 성능 관리**의 5 장, "IPMP 소개"를 참조하십시오.

멀티홈 호스트는 IP 패킷을 전달하지 않지만 경로 지정 프로토콜을 실행하도록 구성될 수 있습니다. 일반적으로 다음 유형의 시스템을 멀티홈 호스트로 구성합니다.

- 대규모 사용자 풀에서 파일을 공유하기 위해 NFS 서버, 특히 큰 데이터 센터로 작동하는 서버를 두 개 이상의 네트워크에 연결할 수 있습니다. 이러한 서버는 경로 지정 테이블을 유지 관리할 필요가 없습니다.
- NFS 서버와 마찬가지로 데이터베이스 서버는 대규모 사용자 풀에 리소스를 제공할 네트워크 인터페이스를 여러 개 포함할 수 있습니다.
- 방화벽 게이트웨이는 회사 네트워크와 공용 네트워크(예: 인터넷) 간의 연결을 제공하는 시스템입니다. 관리자는 방화벽을 보안 조치로 설정합니다. 방화벽으로 구성된 호스트는 호스트의 인터페이스에 연결된 네트워크 간에 패킷을 전달하지 않습니다. 단, 이 경우에도 호스트는 권한이 부여된 사용자에게 표준 TCP/IP 서비스(예: `ssh`)를 제공할 수 있습니다.

주 – 멀티홈 호스트의 인터페이스에 여러 유형의 방화벽이 있을 경우 의도치 않게 호스트의 패킷이 중단되지 않도록 주의해야 합니다. 이 문제는 특히 Stateful 방화벽에서 발생할 수 있습니다. 한 가지 해결 방법은 Stateless 방화벽을 구성하는 것입니다. 방화벽에 대한 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "방화벽 시스템" 또는 타사 방화벽 설명서를 참조하십시오.

▼ 멀티홈 호스트를 만드는 방법

1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2 Oracle Solaris 설치의 일부로 구성되지 않은 각 추가 네트워크 인터페이스를 구성합니다.

Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

3 패킷 전달을 사용으로 설정할 경우 이 서비스를 사용 안함으로 설정합니다.

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY      PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding    rw   on           --           off          on,off

ipadm set-prop -p forwarding=off ipv4
```

4 **(옵션) 멀티홈 호스트에 대한 동적 경로 지정을 켭니다.**

다음 명령 구문 중 하나를 사용합니다.

- **# routeadm -e ipv4-routing -u**
- **# svcadm enable route:default**

  in.routed 데몬과 연관된 SMF FMRI는 svc:/network/routing/route입니다.

예 3–3 멀티홈 호스트 구성

다음 예에서는 그림 3–1에 표시된 멀티홈 호스트를 구성하는 방법을 보여 줍니다. 이 예에서는 시스템의 호스트 이름이 hostc입니다. 이 호스트에는 두 개의 인터페이스가 있으며 모두 192.168.5.0 네트워크에 연결됩니다.

시작하려면 시스템 인터페이스의 상태를 표시합니다.

```
# dladm show-link
LINK     CLASS     MTU     STATE   BRIDGE   OVER
net0     phys      1500    up      --       --
net1     phys      1500    up      --       --

# ipadm show-addr
ADDROBJ        TYPE     STATE        ADDR
lo0/v4         static   ok           127.0.0.1/8
net0/v4        static   ok           192.168.5.82/24
```

dladm show-link 명령이 hostc에 두 개의 데이터 링크가 있는 것으로 보고합니다. 하지만 net0만 IP 주소로 구성되었습니다. hostc를 멀티홈 호스트로 구성하려면 동일한 192.168.5.0 네트워크의 IP 주소로 net1을 구성합니다. net1의 기본적인 물리적 NIC가 네트워크에 물리적으로 연결되었는지 확인합니다.

```
# ipadm create-ip net1
# ipadm create-addr static -a 192.168.5.85/24 net1
# ipadm show-addr
ADDROBJ        TYPE     STATE        ADDR
lo0/v4         static   ok           127.0.0.1/8
net0/v4        static   ok           192.168.5.82/24
net1/v4        static   ok           192.168.5.85/24
```

그런 다음 net1 인터페이스를 /etc/hosts 데이터베이스에 추가합니다.

```
# vi /etc/inet/hosts
127.0.0.1           localhost
192.168.5.82        hostc    #primary network interface for host3
192.168.5.85        hostc-2  #second interface
```

다음으로 이 서비스가 hostc에서 실행 중인 경우 패킷 전달을 끕니다.

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on        --           off          on,off

# ipadm set-prop -p forwarding=off ipv4

# routeadm
              Configuration   Current              Current
                    Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   enabled              enabled
               IPv6 routing   disabled             disabled

           Routing services   "route:default ripng:default"
```

`routeadm` 명령이 `in.routed` 데몬을 통한 동적 경로 지정이 현재 사용으로 설정된 것으로 보고합니다.

단일 인터페이스 시스템에 대한 경로 지정 구성

정적 또는 동적 경로 지정으로 단일 인터페이스 시스템을 구성할 수 있습니다. 정적 경로 지정을 사용하는 경우 호스트는 경로 지정 정보에 기본 라우터의 서비스를 사용해야 합니다. 다음 절차에서는 두 경로 지정 유형을 사용으로 설정하는 지침도 제공합니다.

▼ 단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법

다음 절차에 따라 멀티홈 호스트에서 정적 경로 지정을 구성할 수도 있습니다.

1. **관리자로 전환합니다.**

   자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2. **시스템이 속한 네트워크에 대한 IP 주소로 시스템의 IP 인터페이스를 구성합니다.**

   지침은 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

3. **텍스트 편집기에서 시스템에 사용될 라우터의 IP 주소를 추가하여 `/etc/defaultrouter` 파일을 만들거나 수정합니다.**

4. **로컬 `/etc/inet/hosts` 파일에서 기본 라우터에 대한 항목을 추가합니다.**

5. **경로 지정이 꺼져 있는지 확인합니다.**

   ```
   # routeadm
      Configuration   Current              Current
                        Option   Configuration        System State
   ```

```
----------------------------------------------------------------
              IPv4 routing   enabled             disabled
              IPv6 routing   disabled             disabled

           Routing services   "route:default ripng:default"

# svcadm disable route:default
```

6 패킷 전달이 꺼져 있는지 확인합니다.

```
# # ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on        --           off          on,off

# ipadm set-prop -p forwarding=off ipv4
```

예 3–4 단일 인터페이스 시스템에서 정적 경로 지정 구성

다음 예에서는 Figure 3–1에 표시된 172.16.1.0 네트워크의 단일 인터페이스 시스템 그림 3–1에 대해 정적 경로 지정을 구성하는 방법을 보여 줍니다. hostb는 Router 2를 기본 라우터로 사용해야 합니다. 이 예에서는 시스템의 IP 인터페이스를 이미 구성한 것으로 간주합니다.

먼저 관리자 권한으로 hostb에 로그인합니다. 그런 다음 /etc/defaultrouter 파일이 시스템에 있는지 여부를 확인합니다.

```
# cd /etc
# ls | grep defaultrouter

# vi /etc/defaultrouter
172.16.1.10
```

IP 주소 172.16.1.10은 Router 2에 속합니다.

```
# vi /etc/inet/hosts
127.0.0.1           localhost
172.16.1.18         host2    #primary network interface for host2
172.16.1.10         router2  #default router for host2

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on          --           off          on,off

# ipadm set-prop -p forwarding=off ipv4

# routeadm
   Configuration   Current             Current
                     Option   Configuration        System State
----------------------------------------------------------------
              IPv4 routing   enabled             disabled
              IPv6 routing   disabled             disabled

           Routing services   "route:default ripng:default"
```

```
# svcadm disable route:default
```

▼ 단일 인터페이스 시스템에서 동적 경로 지정을 사용으로 설정하는 방법

경로 지정 프로토콜을 사용하는 동적 경로 지정이 시스템에서 경로 지정을 관리하는 가장 간단한 방법입니다.

1. **관리자로 전환합니다.**

   자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2. **시스템이 속한 네트워크에 대한 IP 주소로 시스템의 IP 인터페이스를 구성합니다.**

   지침은 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

3. **/etc/defaultrouter 파일에서 항목을 삭제합니다.**

   /etc/defaultrouter 파일이 비어 있으면 시스템이 동적 경로 지정을 사용합니다.

4. **패킷 전달이 사용 안함으로 설정되었는지 확인합니다.**

   ```
   # ipadm set-prop -p forwarding=off ipv4
   ```

5. **시스템에서 경로 지정 프로토콜을 사용으로 설정합니다.**

   다음 명령 중 하나를 사용합니다.

   - `# routeadm -e ipv4-routing -u`
   - `# svcadm enable route:default`

예 3–5 단일 인터페이스 시스템에서 동적 경로 지정 실행

다음 예에서는 그림 3–1에 표시된 192.168.5.0 네트워크에서 단일 인터페이스 시스템 hosta에 대해 동적 경로 지정을 구성하는 방법을 보여 줍니다. 시스템에서는 Router 1을 기본 라우터로 사용합니다. 이 예에서는 시스템의 IP 인터페이스를 이미 구성한 것으로 간주합니다.

먼저 관리자 권한으로 hosta에 로그인합니다. 그런 후 시스템에 파일이 존재하면 /etc/defaultrouter 파일을 제거합니다.

```
# cd /etc
# ls | grep defaultrouter
defaultrouter

# rm defaultrouter
```

```
# routeadm   Configuration   Current              Current
                    Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled

          Routing services   "route:default ripng:default"

# svcadm enable route:default

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on          --           off          on,off

# ipadm set-prop -p forwarding=off ipv4
```

네트워크에 서브넷 추가

서브넷을 사용하지 않는 네트워크에서 서브넷을 사용하는 네트워크로 변경 중인 경우 다음 목록의 작업을 수행합니다. 목록에서는 서브넷 스키마를 이미 준비한 것으로 간주합니다.

- 서브넷에 속한 시스템에 새 서브넷 번호의 IP 주소를 지정합니다.

  자세한 내용은 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.
- 각 시스템의 `/etc/netmasks` 파일에 올바른 IP 주소 및 넷마스크를 추가합니다.
- 각 시스템의 `/etc/inet/hosts` 파일을 호스트 이름에 해당하는 올바른 IP 주소로 개정합니다.
- 서브넷의 모든 시스템을 재부트합니다.

다음 절차는 서브넷과 밀접한 관련이 있습니다. 서브넷 없이 네트워크를 구성한 후 나중에 서브넷을 구현하는 경우 다음 절차에 따라 변경 사항을 구현합니다.

▼ IPv4 주소 및 기타 네트워크 구성 매개변수 변경 방법

이 절차에서는 이전에 설치된 시스템에서 IPv4 주소, 호스트 이름 및 기타 네트워크 매개변수를 수정하는 방법에 대해 설명합니다. 절차에 따라 서버 또는 네트워크로 연결된 독립형 시스템의 IP 주소를 수정할 수 있습니다. 네트워크 클라이언트 또는 어플라이언스에는 이 절차를 사용할 수 없습니다. 단계에서는 재부트 시 지속되는 구성을 만듭니다.

주 – 특히 기본 네트워크 인터페이스의 IPv4 주소를 변경하려는 경우 지침을 따르십시오. 시스템에 다른 인터페이스를 추가하려면 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

대부분의 경우 다음 단계에서는 기존 IPv4의 점으로 구분된 십진수 표기법을 사용하여 IPv4 주소 및 서브넷 마스크를 지정합니다. 또는 CIDR 표기법을 사용하여 이 절차의 모든 해당 파일에서 IPv4 주소를 지정할 수도 있습니다.

1 **관리자로 전환합니다.**

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2 **ipadm 명령을 사용하여 IP 주소를 수정합니다.**

ipadm 명령을 통해서는 IP 주소를 직접 수정할 수 없습니다. 먼저 수정할 IP 주소를 나타내는 주소 객체를 삭제합니다. 그런 다음 동일한 주소 객체 이름을 사용하여 새 주소를 지정합니다.

```
# ipadm delete-addr addrobj
# ipadm create-addr -a IP-address interface
```

3 **해당하는 경우 system/identity:node SMF 서비스에서 호스트 이름 항목을 수정합니다.**

```
# hostname newhostname
```

이 단계는 지속적인 변경 사항에 영향을 줍니다.

4 **서브넷 마스크가 변경된 경우 /etc/netmasks 파일에서 서브넷 항목을 수정합니다.**

5 **서브넷 주소가 변경된 경우 /etc/defaultrouter의 기본 라우터 IP 주소를 새 서브넷의 기본 라우터 IP 주소로 변경합니다.**

6 **시스템을 재부트합니다.**

```
# reboot -- -r
```

예 3–6 IP 주소 및 호스트 이름 변경

이 예에서는 호스트 이름, 기본 네트워크 인터페이스의 IP 주소 및 서브넷 마스크를 변경하는 방법을 보여 줍니다. 기본 네트워크 인터페이스 net0에 대한 IP 주소가 10.0.0.14에서 192.168.34.100으로 변경됩니다.

```
# ipadm show-addr
ADDROBJ      TYPE     STATE   ADDR
lo0/v4       static   ok      127.0.0.1/8
net0/v4      static   ok      10.0.0.14/24

# ipadm delete-addr net0/v4
# ipadm create-addr -a 192.168.34.100/24 net0
# hostname mynewhostname

# ipadm show-addr
ADDROBJ         TYPE     STATE   ADDR
lo0/v4          static   ok      127.0.0.1/8
net0/v4         static   ok      192.168.34.100/24
```

```
# hostname
mynewhostname
```

참조 인터페이스를 기본 네트워크 인터페이스와 다른 IP 주소로 변경하려면 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

전송 계층 서비스 모니터 및 수정

전송 계층 프로토콜인 TCP, SCTP 및 UDP는 표준 Oracle Solaris 패키지의 일부입니다. 일반적으로 이러한 프로토콜은 개입 없이도 제대로 실행됩니다. 하지만 사이트의 요구 사항에 따라 전송 계층 프로토콜을 통해 실행되는 서비스를 기록하거나 수정해야 할 수도 있습니다. 그런 다음 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 1 장, "서비스 관리(개요)"에 설명된 대로 SMF(서비스 관리 기능)를 사용하여 서비스에 대한 프로파일을 수정해야 합니다.

`inetd` 데몬은 시스템 부트 시 표준 인터넷 서비스를 시작합니다. 이러한 서비스에는 TCP, SCTP 또는 UDP를 전송 계층 프로토콜로 사용하는 응용 프로그램이 포함됩니다. SMF 명령을 사용하여 기존 인터넷 서비스를 수정하거나 새 서비스를 추가할 수 있습니다. `inetd`에 대한 자세한 내용은 118 페이지 "inetd Internet Services Daemon"을 참조하십시오.

전송 계층 프로토콜과 관련된 작업은 다음과 같습니다.

- 모든 수신 TCP 연결 기록
- 전송 계층 프로토콜을 통해 실행되는 서비스 추가, SCTP를 예로 사용
- 액세스 제어를 위해 TCP 래퍼 기능 구성

`inetd` 데몬에 대한 자세한 내용은 `inetd(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 모든 수신 TCP 연결의 IP 주소 기록 방법

1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2 `inetd`로 관리되는 모든 서비스에 대해 TCP 추적을 사용으로 설정합니다.

```
# inetadm -M tcp_trace=TRUE
```

▼ SCTP 프로토콜을 사용하는 서비스를 추가하는 방법

SCTP 전송 프로토콜은 TCP와 유사한 방식으로 응용 프로그램 전송 프로토콜에 서비스를 제공합니다. 하지만 SCTP는 둘 중 하나 또는 모두가 멀티홈일 수 있는 두 시스템 간의 통신을 가능하게 합니다. SCTP 연결을 **연관**이라고 합니다. 연관에서 응용 프로그램은 하나 이상의 메시지 스트림으로 전송되거나 **다중 스트림**될 데이터를 구분합니다. SCTP 연결은 IP 주소가 여러 개인 끝점으로 이동할 수 있으므로 전화 기술 응용 프로그램에서 특히 중요합니다. 사이트에서 IP 필터 또는 IPsec를 사용하는 경우 보안상 SCTP의 멀티 홈 기능을 고려해야 합니다. 이러한 고려 사항 중 몇 가지는 `sctp(7P)` 매뉴얼 페이지에서 설명됩니다.

기본적으로 SCTP는 Oracle Solaris에 포함되어 있으며 추가 구성을 필요로 하지 않습니다. 단, SCTP를 사용하도록 특정 응용 프로그램 계층 서비스를 명시적으로 구성해야 합니다. `echo` 및 `discard`가 이러한 응용 프로그램에 해당합니다. 다음 절차에서는 SCTP 일대일 스타일 소켓을 사용하는 echo 서비스를 추가하는 방법을 보여 줍니다.

주 – 다음 절차에 따라 TCP 및 UDP 전송 계층 프로토콜에 대한 서비스를 추가할 수도 있습니다.

다음 작업에서는 `inetd` 데몬으로 관리되는 SCTP `inet` 서비스를 SMF 저장소에 추가하는 방법을 보여 줍니다. 그런 다음 SMF(서비스 관리 기능) 명령을 사용하여 서비스를 추가하는 방법을 보여 줍니다.

- SMF 명령에 대한 자세한 내용은 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 "SMF 명령줄 관리 유틸리티"를 참조하십시오.
- 구문 정보는 절차에서 인용된 SMF 명령에 대한 매뉴얼 페이지를 참조하십시오.
- SMF에 대한 자세한 내용은 `smf(5)` 매뉴얼 페이지를 참조하십시오.

시작하기 전에 다음 절차를 수행하기 전에 서비스에 대한 매니페스트 파일을 만드십시오. 절차에서는 `echo` 서비스에 대한 매니페스트(`echo.sctp.xml`)를 예로 사용합니다.

1. **시스템 파일에 대한 쓰기 권한이 있는 사용자 계정으로 로컬 시스템에 로그인합니다.**

2. **`/etc/services` 파일을 편집하고 새 서비스에 대한 정의를 추가합니다.**
   서비스 정의에 대한 다음 구문을 사용합니다.
   service-name |*port/protocol* | *aliases*

3. **새 서비스를 추가합니다.**
   서비스 매니페스트가 저장된 디렉토리로 이동하여 다음을 입력합니다.

   ```
   # cd dir-name
   # svccfg import service-manifest-name
   ```

svccfg의 전체 구문은 svccfg(1M) 매뉴얼 페이지를 참조하십시오.

현재 service.dir 디렉토리에 있는 echo.sctp.xml 매니페스트를 사용하여 새 SCTP echo 서비스를 추가하려고 한다고 가정합니다. 다음을 입력합니다.

```
# cd service.dir
# svccfg import echo.sctp.xml
```

4 서비스 매니페스트가 추가되었는지 확인합니다.

```
# svcs FMRI
```

FMRI 인수로 서비스 매니페스트의 FMRI(Fault Managed Resource Identifier)를 사용합니다. 예를 들어, SCTP echo 서비스의 경우 다음 명령을 사용합니다.

```
# svcs svc:/network/echo:sctp_stream
```

출력이 다음과 유사하게 표시됩니다.

```
    STATE          STIME    FMRI
disabled       16:17:00 svc:/network/echo:sctp_stream
```

svcs 명령에 대한 자세한 내용은 svcs(1) 매뉴얼 페이지를 참조하십시오.

출력은 새 서비스 매니페스트가 현재 사용 안함으로 설정되어 있음을 나타냅니다.

5 수정해야 할지 여부를 결정할 서비스의 등록 정보를 나열합니다.

```
# inetadm -l FMRI
```

inetadm 명령에 대한 자세한 내용은 inetadm(1M) 매뉴얼 페이지를 참조하십시오.

예를 들어, SCTP echo 서비스의 경우 다음을 입력합니다.

```
# inetadm -l svc:/network/echo:sctp_stream
SCOPE    NAME=VALUE
             name="echo"
             endpoint_type="stream"
             proto="sctp"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
         .
         .
         default  tcp_trace=FALSE
           default  tcp_wrappers=FALSE
```

6 새 서비스를 사용으로 설정합니다.

```
# inetadm -e FMRI
```

7 서비스가 사용으로 설정되었는지 확인합니다.

예를 들어, 새 echo 서비스의 경우 다음을 입력합니다.

```
# inetadm | grep sctp_stream
.
.
    enabled   online         svc:/network/echo:sctp_stream
```

예 3–7 SCTP 전송 프로토콜을 사용하는 서비스 추가

다음 예에서는 사용할 명령과 echo 서비스가 SCTP 전송 계층 프로토콜을 사용하도록 하는 데 필요한 파일 항목을 보여 줍니다.

```
$ cat /etc/services
.
.
echo            7/tcp
echo            7/udp
echo            7/sctp

# cd service.dir

    # svccfg import echo.sctp.xml

# svcs network/echo*
    STATE          STIME    FMRI
    disabled       15:46:44 svc:/network/echo:dgram
    disabled       15:46:44 svc:/network/echo:stream
    disabled       16:17:00 svc:/network/echo:sctp_stream

# inetadm -l svc:/network/echo:sctp_stream
    SCOPE    NAME=VALUE
             name="echo"
             endpoint_type="stream"
             proto="sctp"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
             user="root"
    default  bind_addr=""
    default  bind_fail_max=-1
    default  bind_fail_interval=-1
    default  max_con_rate=-1
    default  max_copies=-1
    default  con_rate_offline=-1
    default  failrate_cnt=40
    default  failrate_interval=60
    default  inherit_env=TRUE
    default  tcp_trace=FALSE
    default  tcp_wrappers=FALSE

# inetadm -e svc:/network/echo:sctp_stream

# inetadm | grep echo
    disabled  disabled       svc:/network/echo:stream
```

```
disabled  disabled       svc:/network/echo:dgram
enabled   online         svc:/network/echo:sctp_stream
```

▼ TCP 래퍼를 사용하여 TCP 서비스에 대한 액세스를 제어하는 방법

tcpd 프로그램은 *TCP 래퍼*를 구현합니다. TCP 래퍼는 데몬과 수신 서비스 요청 사이에서 서비스 데몬(예: ftpd)에 대한 보안 조치를 추가합니다. 또한 연결 시도 성공 및 실패를 기록합니다. TCP 래퍼는 요청 시작 위치에 따라 연결을 허용하거나 거부하여 액세스 제어를 제공할 수도 있습니다. TCP 래퍼를 사용하여 SSH, Telnet, FTP 등의 데몬을 보호할 수 있습니다. sendmail 응용 프로그램은 **Oracle Solaris 11.1에서 sendmail 서비스 관리**의 "sendmail 버전 8.12의 TCP 래퍼에 대한 지원"에 설명된 대로 TCP 래퍼를 사용할 수 있습니다.

1. **관리자로 전환합니다.**

   자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스**의 "지정된 관리 권한을 사용하는 방법"을 참조하십시오.

2. **TCP 래퍼를 사용으로 설정합니다.**

   ```
   # inetadm -M tcp_wrappers=TRUE
   ```

3. **hosts_access(3) 매뉴얼 페이지에 설명된 대로 TCP 래퍼 액세스 제어 정책을 구성합니다.**

   이 매뉴얼 페이지는 /usr/sfw/man 디렉토리에서 확인할 수 있습니다.

◆ ◆ ◆ 4 장

4 네트워크에서 IPv6 사용

이 장에서는 네트워크에서 IPv6을 사용하기 위한 작업에 대해 설명합니다. 다음 주요 항목을 다룹니다.

IPv6 인터페이스 구성

네트워크에서 IPv6을 사용하기 위한 초기 단계로, 시스템의 IP 인터페이스에서 IPv6을 구성하십시오.

Oracle Solaris 설치 프로세스 중 하나 이상의 시스템 인터페이스에서 IPv6을 사용으로 설정할 수 있습니다. 설치 중 IPv6 지원을 사용으로 설정한 경우에는 설치가 완료되면 다음과 같은 IPv6 관련 파일 및 테이블이 생성됩니다.

- `name-service/switch` SMF 서비스는 IPv6 주소를 사용하여 조회가 가능하도록 수정되었습니다.
- IPv6 주소 선택 정책 테이블이 생성됩니다. 이 테이블은 IPv6 지원 인터페이스를 통한 전송에 사용할 IP 주소 형식의 우선 순위를 정합니다.

이 절에서는 Oracle Solaris 설치가 완료된 후 인터페이스에서 IPv6을 사용으로 설정하는 방법에 대해 설명합니다.

▼ IPv6에 대해 시스템을 구성하는 방법

IPv6 노드로 사용될 모든 시스템의 인터페이스에서 IPv6을 사용으로 설정하여 IPv6 구성 프로세스를 시작하십시오. 처음에 인터페이스는 135 페이지 "자동 구성 프로세스"에 설명된 대로 자동 구성 프로세스를 통해 IPv6 주소를 가져옵니다. 그런 다음 IPv6 네트워크의 기능을 기준으로 노드의 구성을 호스트, 서버 또는 라우터로 조정합니다.

주 – 인터페이스가 현재 IPv6 접두어를 알리는 라우터와 동일한 링크에 있는 경우, 자동 구성된 주소의 일부로 해당 사이트의 접두어를 얻습니다. 자세한 내용은 64 페이지 "IPv6 지원 라우터를 구성하는 방법"을 참조하십시오.

다음 절차는 Oracle Solaris 설치 이후에 추가된 인터페이스에 대해 IPv6을 사용으로 설정하는 방법에 대해 설명합니다.

1 **적합한 명령을 사용하여 IP 인터페이스를 구성합니다.**

Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결의 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

주 – IP 주소를 지정할 경우 올바른 옵션을 사용하여 IPv6 주소를 지정해야 합니다.

```
# ipadm create-addr -T addrconf interface
```

주소를 더 추가하려면 다음 구문을 사용합니다.

```
# ipadm create-addr -a ipv6-address interface
```

---

2 **IPv6 데몬 `in.ndpd`를 시작합니다.**

```
# /usr/lib/inet/in.ndpd
```

3 **(옵션) 정적 IPv6 기본 경로를 만듭니다.**

```
# /usr/sbin/route -p add -inet6 default ipv6-address
```

4 **(옵션) 노드의 인터페이스 변수에 대한 매개변수를 정의하는 `/etc/inet/ndpd.conf` 파일을 만듭니다.**

호스트의 인터페이스에 대해 임시 주소를 만들어야 하는 경우 66 페이지 "인터페이스에 대해 임시 주소 사용"을 참조하십시오. `/etc/inet/ndpd.conf`에 대한 자세한 내용은 `ndpd.conf`(4) 매뉴얼 페이지 및 123 페이지 "`ndpd.conf` 구성 파일"을 참조하십시오.

5 **(옵션) IPv6 구성을 포함하는 IP 인터페이스의 상태를 표시하려면 다음 명령을 입력합니다.**

```
# ipadm show-addr
```

예 4–1 설치 후 IPv6 인터페이스 사용

이 예는 net0 인터페이스에서 IPv6을 사용으로 설정하는 방법을 보여줍니다. 시작하기 전에 시스템에 구성된 모든 인터페이스의 상태를 확인하십시오.

```
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADDR
lo0/v4    static   ok      127.0.0.1/8
net0/v4   static   ok      172.16.27.74/24
```

현재 net0 인터페이스만 이 시스템에 대해 구성되어 있습니다. 다음과 같이 이 인터페이스에서 IPv6을 사용으로 설정하십시오.

```
# ipadm create-addr -T addrconf net0
# ipadm create-addr -a 2001:db8:3c4d:15:203/64 net0
# /usr/lib/inet/in.ndpd

# ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      172.16.27.74/24
net0/v6       addrconf   ok      fe80::203:baff:fe13:14e1/10
lo0/v6        static     ok      ::1/128
net0/v6a      static     ok      2001:db8:3c4d:15:203/64

# route -p add -inet6 default fe80::203:baff:fe13:14e1
```

다음 순서

- IPv6 노드를 라우터로 구성하려면 64 페이지 “IPv6 라우터 구성”으로 이동합니다.
- 노드에 대한 주소 자동 구성을 사용 안함으로 설정하려면 63 페이지 “IPv6 주소 자동 구성을 해제하는 방법”을 참조하십시오.
- 노드를 서버로 조정하려면 71 페이지 “서버에서 IPv6 지원 인터페이스 관리”의 제안 사항을 참조하십시오.

▼ IPv6 주소 자동 구성을 해제하는 방법

일반적으로 호스트 및 서버의 인터페이스에 대한 IPv6 주소는 주소 자동 구성을 사용하여 생성해야 합니다. 그러나 69 페이지 “IPv6 토큰 구성”에 설명된 것과 같이, 특히 토큰을 수동으로 구성하려는 경우 주소 자동 구성을 해제할 수 있습니다.

1 노드에 대한 /etc/inet/ndpd.conf 파일을 만듭니다.

/etc/inet/ndpd.conf 파일은 특정 노드에 대한 인터페이스 변수를 정의합니다. 서버의 인터페이스에 대한 주소 자동 구성을 해제하려면 이 파일에 다음과 같은 내용이 포함되어야 합니다.

interface StatelessAddrConf false

모든 인터페이스에 대한 주소 자동 구성을 해제하려면 다음 항목을 사용합니다.

```
ifdefault StatelessAddrConf false
```

/etc/inet/ndpd.conf에 대한 자세한 내용은 ndpd.conf(4) 매뉴얼 페이지 및 123 페이지 “ndpd.conf 구성 파일”을 참조하십시오.

2 **변경 사항으로 IPv6 데몬을 업데이트합니다.**

```
# pkill -HUP in.ndpd
```

IPv6 라우터 구성

이 절에서는 IPv6 라우터 구성 작업에 대해 설명합니다. 사이트 요구 사항에 따라 선택한 작업만 수행해야 할 수 있습니다.

▼ IPv6 지원 라우터를 구성하는 방법

다음 절차에서는 시스템에 이미 IPv6이 구성되었다고 가정합니다. 절차는 61 페이지 “IPv6 인터페이스 구성”을 참조하십시오.

1 **라우터의 모든 인터페이스에서 IPv6 패킷 전달을 구성합니다.**

```
# ipadm set-prop -p forwarding=on ipv6
```

2 **경로 지정 데몬을 시작합니다.**

in.ripngd 데몬은 IPv6 경로 지정을 처리합니다. IPv6 경로 지정은 다음 방법 중 하나로 설정합니다.

- routeadm 명령을 사용합니다.

  ```
  # routeadm -e ipv6-routing -u
  ```

- 해당 SMF 명령을 사용합니다.

  ```
  # svcadm enable ripng:default
  ```

routeadm 명령에 대한 구문 정보는 routeadm(1M) 매뉴얼 페이지를 참조하십시오

3 **/etc/inet/ndpd.conf 파일을 만듭니다.**

라우터가 알릴 사이트 접두어 및 기타 구성 정보를 /etc/inet/ndpd.conf에 지정합니다. 이 파일은 IPv6 Neighbor Discovery 프로토콜을 구현하는 in.ndpd 데몬이 읽습니다.

변수 및 허용되는 값 목록은 123 페이지 “ndpd.conf 구성 파일” 및 ndpd.conf(4) 매뉴얼 페이지를 참조하십시오.

4 **/etc/inet/ndpd.conf 파일에 다음 텍스트를 입력합니다.**

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on
```

이 텍스트는 IPv6용으로 구성된 라우터의 모든 인터페이스를 통해 라우터 알림을 전송하도록 in.ndpd에 지시합니다.

5 **/etc/inet/ndpd.conf 파일에 추가 텍스트를 추가하여 라우터의 여러 인터페이스에 사이트 접두어를 구성합니다.**

이 텍스트는 다음과 같은 형식이어야 합니다.

`prefix` *global-routing-prefix*`:`*subnet ID*`/64` *interface*

다음 샘플 `/etc/inet/ndpd.conf` 파일은 `net0` 및 `net1` 인터페이스를 통해 사이트 접두어 `2001:0db8:3c4d::/48`을 알리도록 라우터를 구성합니다.

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on

if net0 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:15::0/64 net0

if net1 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:16::0/64 net1
```

6 **시스템을 재부트합니다.**

IPv6 라우터가 `ndpd.conf` 파일에 있는 사이트 접두어를 로컬 사이트에 알립니다.

예 4–2 IPv6 주소를 표시하는 `ipadm show-addr` 출력

다음 예는 64 페이지 “IPv6 라우터 구성” 절차를 완료하면 표시되는 것과 같은, `ipadm show-addr` 명령의 출력을 보여줍니다.

```
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      172.16.15.232/24
net1/v4       static     ok      172.16.16.220/24
net0/v6       addrconf   ok      fe80::203:baff:fe11:b115/10
lo0/v6        static     ok      ::1/128
net0/v6a      static     ok      2001:db8:3c4d:15:203:baff:fe11:b115/64
net1/v6       addrconf   ok      fe80::203:baff:fe11:b116/10
net1/v6a      static     ok      2001:db8:3c4d:16:203:baff:fe11:b116/64
```

이 예에서 IPv6용으로 구성된 각 인터페이스가 이제 두 개의 주소를 사용합니다. *interface*`/v6`과 같은 주소 객체 이름을 포함하는 항목에는 해당 인터페이스에 대한 링크 로컬 주소가 표시됩니다. *interface*`/v6add`와 같은 주소 객체 이름을 포함하는 항목에는 전역 IPv6 주소가 표시됩니다. 이 주소에는 인터페이스 ID 이외에도 `/etc/ndpd.conf` 파일에 구성된 사이트 접두어가 포함됩니다. `v6add` 대상은 무작위로 정의된 문자열입니다. *interface*가 IPv6 주소를 만들려는 인터페이스(예: `net0/mystring`, `net0/ipv6addr` 등)를 나타내는 경우, 다른 문자열을 정의하여 주소 객체 이름의 두번째 부분을 구성할 수 있습니다.

참조

- IPv6 네트워크 토폴로지에서 식별한 라우터에서 터널을 구성하려면 105 페이지 “`dladm` 명령을 통한 터널 구성 및 관리”를 참조하십시오.
- 네트워크에서 스위치 및 허브 구성에 대한 자세한 내용은 제조업체의 설명서를 참조하십시오.

- IPv6 호스트를 구성하려면 66 페이지 “호스트 및 서버에 대해 IPv6 인터페이스 구성 수정”을 참조하십시오.
- 서버에서 IPv6 지원을 향상시키려면 71 페이지 “서버에서 IPv6 지원 인터페이스 관리”를 참조하십시오.
- IPv6 명령, 파일 및 데몬에 대한 자세한 내용은 123 페이지 “Oracle Solaris IPv6 구현”을 참조하십시오.

호스트 및 서버에 대해 IPv6 인터페이스 구성 수정

이 단원에서는 호스트 또는 서버인 노드에서 IPv6 지원 인터페이스의 구성을 수정하는 방법에 대해 설명합니다. 대부분의 경우 IPv6 지원 인터페이스에 대해 주소 자동 구성을 사용해야 합니다. 그러나 이 단원의 작업에 설명된 것과 같이, 필요한 경우 인터페이스의 IPv6 주소를 수정할 수 있습니다.

세 가지 일반 작업을 다음 순서로 수행해야 합니다.

1. IPv6 주소 자동 구성을 해제합니다. 63 페이지 “IPv6 주소 자동 구성을 해제하는 방법”을 참조하십시오.
2. 호스트에 대해 임시 주소를 만듭니다. 67 페이지 “임시 주소를 구성하는 방법”을 참조하십시오.
3. 인터페이스 ID에 대해 IPv6 토큰을 구성합니다. 69 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”을 참조하십시오.

인터페이스에 대해 임시 주소 사용

IPv6 **임시 주소**에는 인터페이스의 MAC 주소 대신 무작위로 생성된 64비트 숫자가 인터페이스 ID로 포함됩니다. 익명으로 유지하려는 IPv6 노드의 인터페이스에 대해 임시 주소를 사용할 수 있습니다. 예를 들어 공개 웹 서버에 액세스해야 하는 호스트의 인터페이스에 대해 임시 주소를 사용할 수 있습니다. 임시 주소는 IPv6 프라이버시의 향상된 기능을 구현합니다. 이러한 향상된 기능은 “Privacy Extensions for Stateless Address Autoconfiguration in IPv6” (`http://www.ietf.org/rfc/rfc3041.txt?number=3041`)에서 제공하는 RFC 3041에 설명되어 있습니다.

필요한 경우 `/etc/inet/ndpd.conf` 파일에서 하나 이상의 인터페이스에 대해 임시 주소를 사용으로 설정할 수 있습니다. 그러나 자동 구성된 표준 IPv6 주소와 달리, 임시 주소는 64비트 서브넷 접두어와 무작위로 작성된 64비트 숫자로 구성됩니다. 이 무작위 숫자가 IPv6 주소의 인터페이스 ID 세그먼트가 됩니다. 임시 주소를 사용할 경우 링크 로컬 주소가 인터페이스 ID로 생성되지 않습니다.

임시 주소에는 기본 **선호 수명**(1일)이 지정됩니다. 임시 주소 생성을 사용으로 설정한 경우 `/etc/inet/ndpd.conf` 파일에서 다음 변수를 구성할 수도 있습니다.

valid lifetime TmpValidLifetime	호스트에서 주소가 삭제된 후 임시 주소가 존재하는 시간 범위입니다.

preferred lifetime TmpPreferredLifetime	임시 주소가 제거되기 전의 경과 시간입니다. 이 시간 범위는 유효 수명보다 짧아야 합니다.
address regeneration	선호 수명이 만료되기 이전 기간으로, 이 기간 동안 호스트에서 임시 주소를 새로 생성해야 합니다.

임시 주소의 기간은 다음과 같이 표시됩니다.

n	*n*은 초 수입니다(기본값).
n h	*n*은 시간(h) 수입니다.
n d	*n*은 일(d) 수입니다.

▼ 임시 주소를 구성하는 방법

1 필요한 경우 호스트의 인터페이스에서 IPv6을 사용으로 설정합니다.

62 페이지 "IPv6에 대해 시스템을 구성하는 방법"을 참조하십시오.

2 /etc/inet/ndpd.conf 파일을 편집하여 임시 주소 생성을 설정합니다.

- 호스트의 모든 인터페이스에서 임시 주소를 구성하려면 /etc/inet/ndpd.conf에 다음 행을 추가합니다.

  ifdefault TmpAddrsEnabled true
- 특정 인터페이스에 대해 임시 주소를 구성하려면 /etc/inet/ndpd.conf에 다음 행을 추가합니다.

  if *interface* **TmpAddrsEnabled true**

3 (옵션) 임시 주소의 유효 수명을 지정합니다.

ifdefault TmpValidLifetime *duration*

이 구문은 호스트에 있는 모든 인터페이스의 유효 수명을 지정합니다. *duration*의 값은 초, 시간 또는 일 단위여야 합니다. 기본 유효 수명은 7일입니다. TmpValidLifetime을 if *interface* 키워드와 함께 사용하여 특정 인터페이스의 임시 주소에 대한 유효 수명을 지정할 수도 있습니다.

4 (옵션) 임시 주소의 선호 수명을 지정합니다. 이 기간이 경과하면 주소가 제거됩니다.

if *interface* **TmpPreferredLifetime** *duration*

이 구문은 특정 인터페이스의 임시 주소에 대한 선호 수명을 지정합니다. 기본 선호 수명은 1일입니다. TmpPreferredLifetime을 ifdefault 키워드와 함께 사용하여 호스트의 모든 인터페이스에서 임시 주소에 대한 선호 수명을 지정할 수도 있습니다.

주 – 기본 주소 선택은 제거된 IPv6 주소에 낮은 우선 순위를 지정합니다. IPv6 임시 주소가 제거된 경우, 기본 주소 선택은 사용 가능한 주소를 패킷의 소스 주소로 선택합니다. 사용 가능한 주소는 자동으로 생성된 IPv6 주소 또는 인터페이스의 IPv4 주소일 수 있습니다. 기본 주소 선택에 대한 자세한 내용은 94 페이지 "기본 주소 선택 관리"를 참조하십시오.

5 (옵션) 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 이 시간 동안 호스트에서 임시 주소를 새로 생성해야 합니다.

```
ifdefault TmpRegenAdvance duration
```

이 구문은 호스트에 있는 모든 인터페이스의 임시 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 기본값은 5초입니다.

6 `in.ndpd` 데몬의 구성을 변경합니다.

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

7 Example 4–4에 표시된 것과 같이, 예 4–4 명령을 실행하여 임시 주소가 생성되었는지 확인합니다.

명령 출력에서 임시 주소의 CURRENT 필드에 t 플래그가 표시됩니다.

예 4–3 `/etc/inet/ndpd.conf` 파일의 임시 주소 변수

다음 예는 기본 네트워크 인터페이스에 대해 임시 주소가 사용으로 설정된 `/etc/inet/ndpd.conf` 파일의 세그먼트를 보여줍니다.

```
ifdefault TmpAddrsEnabled true

ifdefault TmpValidLifetime 14d

ifdefault TmpPreferredLifetime 7d

ifdefault TmpRegenAdvance 6s
```

예 4–4 임시 주소가 사용으로 설정된 `ipadm show-addr` 명령 출력

이 예는 임시 주소가 생성된 후 `netstat` 명령의 출력을 보여줍니다. IPv6 관련 정보만 샘플 출력에 포함되어 있습니다.

```
# ipadm show-addr -o all
ADDROBJ    TYPE     STATE CURRENT PERSISTENT ADDR
lo0/v6     static   ok    U----   ---        ::1/128
net0/v6    addrconf ok    U----   ---        fe80::a00:20ff:feb9:4c54/10
net0/v6a   static   ok    U----   ---        2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
net0/?     addrconf ok    U--t-   ---        2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

주소 객체 net0/?의 경우 t 플래그가 CURRENT 필드 아래에 설정되어 있습니다. 이 플래그는 해당 주소에 임시 인터페이스 ID가 있음을 나타냅니다.

참조
- IPv6 주소에 대한 이름 서비스 지원을 설정하려면 72 페이지 “IPv6용 이름 서비스 지원 구성”을 참조하십시오.
- 서버에 대해 IPv6 주소를 구성하려면 69 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”을 참조하십시오.
- IPv6 노드에 대한 작업을 모니터링하려면 5 장, “TCP/IP 네트워크 관리”를 참조하십시오.

IPv6 토큰 구성

IPv6 주소의 64비트 인터페이스 ID를 **토큰**이라고도 합니다. 주소 자동 구성 중 토큰은 인터페이스의 MAC 주소와 연관됩니다. 대부분의 경우 비경로 지정 노드인 IPv6 호스트와 서버는 자동 구성된 토큰을 사용해야 합니다.

그러나 시스템 유지 관리의 일부로 인터페이스가 무작위로 교체되는 서버의 경우 자동 구성된 토큰을 사용하면 문제가 발생할 수 있습니다. 인터페이스 카드가 변경되면 MAC 주소도 변경됩니다. 그 결과 정적 IP 주소에 의존하는 서버에서 문제가 발생할 수 있습니다. 네트워크 기반구조의 여러 부분(예: DNS 또는 NIS)에 서버의 인터페이스에 대한 특정 IPv6 주소가 저장되었을 수 있습니다.

주소 변경 문제를 방지하려면 IPv6 주소에서 인터페이스 ID로 사용할 토큰을 수동으로 구성하면 됩니다. 토큰을 만들려면 IPv6 주소의 인터페이스 ID 부분을 차지할 64비트 이하의 16진수를 지정하십시오. 이후 주소 자동 구성 중 Neighbor Discovery는 인터페이스의 MAC 주소를 기반으로 하는 인터페이스 ID를 만들지 않습니다. 대신 수동으로 생성된 토큰이 인터페이스 ID가 됩니다. 이 토큰은 카드가 교체된 후에도 계속 인터페이스에 지정되어 있습니다.

주 – 사용자 지정 토큰과 임시 주소의 차이점은 임시 주소는 사용자가 명시적으로 만드는 것이 아니라 무작위로 생성된다는 점입니다.

▼ 사용자 지정 IPv6 토큰을 구성하는 방법

다음 지침은 인터페이스가 자주 교체되는 서버에 특히 유용합니다. 또한 IPv6 노드에서 사용자 지정 토큰을 구성하는 경우에도 유효합니다.

1 토큰을 사용하여 구성할 인터페이스가 존재하며 인터페이스에 IPv6 주소가 구성되지 않았는지 확인합니다.

주 – 인터페이스에 구성된 IPv6 주소가 없는지 확인하십시오.

```
# ipadm show-if
IFNAME   CLASS      STATE   ACTIVE   OVER
lo0      loopback   ok       yes     ---
net0     ip         ok       yes     ---

# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
```

이 출력은 네트워크 인터페이스 net0이 구성된 IPv6 주소 없이 존재함을 보여줍니다.

2 ***xxxx*:*xxxx*:*xxxx*:*xxxx* 형식을 따르는 노드의 인터페이스에 대한 토큰으로 사용할 하나 이상의 64비트 16진수 숫자를 만듭니다.

3 **토큰을 사용하여 각 인터페이스를 구성합니다.**

각 인터페이스에 대해 다음 형식의 ipadm 명령을 사용하여 사용자 정의 인터페이스 ID(토큰)를 생성합니다.

 # ipadm create-addr -T addrconf -i *interface-ID interface*

예를 들어, 다음 명령으로 토큰을 포함하는 net0 인터페이스를 구성할 수 있습니다.

```
# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
```

주 – 토큰을 사용하여 주소 객체가 생성되면 더 이상 토큰을 수정할 수 없습니다.

4 **변경 사항으로 IPv6 데몬을 업데이트합니다.**

```
# pkill -HUP in.ndpd
```

예 4–5 IPv6 인터페이스에서 사용자 지정 토큰 구성

다음 예는 IPv6 주소 및 토큰으로 net0이 구성됨을 보여줍니다.

```
# ipadm show-if
IFNAME   CLASS      STATE    ACTIVE     OVER
lo0      loopback   ok       yes        ---
net0     ip         ok       yes        ---

# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8

# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
# pkill -HUP in.ndpd
# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v6       static     ok      ::1/128
net0/v6      addrconf   ok      fe80::1a:2b:3c:4d/10
net0/v6a     addrconf   ok      2002:a08:39f0:1:1a:2b:3c:4d/64
```

토큰이 구성되면 주소 객체 net0/v6에 링크 로컬 주소와 인터페이스 ID에 대해 구성된 1a:2b:3c:4d 주소가 생깁니다. net0/v6이 생성된 후에는 더 이상 이 인터페이스에 대해 이 토큰을 수정할 수 없습니다.

참조
- 서버의 IPv6 주소로 이름 서비스를 업데이트하려면 72 페이지 "IPv6용 이름 서비스 지원 구성"을 참조하십시오.
- 서버 성능을 모니터링하려면 5 장, "TCP/IP 네트워크 관리"를 참조하십시오.

서버에서 IPv6 지원 인터페이스 관리

서버에서 IPv6을 계획한 경우 서버 인터페이스에서 IPv6을 사용으로 설정했으므로 몇 가지 사항을 결정해야 합니다. 이러한 결정 사항은 인터페이스 IPv6 주소의 인터페이스 ID(**토큰**이라고도 함)를 구성하는 데 사용할 전략에 영향을 미칩니다.

▼ 서버 인터페이스에서 IPv6을 사용으로 설정하는 방법

이 절차는 네트워크 서버에서 IPv6을 사용으로 설정하는 일반적인 단계를 제공합니다. IPv6 구현 방식에 따라 몇 가지 단계는 다를 수 있습니다.

1. **서버의 IP 인터페이스에서 IPv6을 사용으로 설정합니다.**
   절차는 61 페이지 "IPv6 인터페이스 구성"을 참조하십시오.

2. **서버와 동일한 링크에 있는 라우터에서 IPv6 서브넷 접두어가 구성되었는지 확인합니다.**
   자세한 내용은 64 페이지 "IPv6 라우터 구성"을 참조하십시오.

3. **서버 IPv6 지원 인터페이스의 인터페이스 ID에 적합한 전략을 사용합니다.**
   기본적으로 IPv6 주소 자동 구성은 IPv6 주소의 인터페이스 ID 부분을 만들 때 인터페이스의 MAC 주소를 사용합니다. 인터페이스의 IPv6 주소가 잘 알려진 주소일 경우 한 인터페이스를 다른 인터페이스로 교체하면 문제가 발생할 수 있습니다. 새 인터페이스의 MAC 주소는 다릅니다. 주소 자동 구성 중 토큰은 새 인터페이스 ID가 생성됩니다.
   - 바꿀 계획이 없는 IPv6 지원 인터페이스의 경우에는 135 페이지 "자동 구성 프로세스"에 설명된 대로 자동 구성된 IPv6 주소를 사용합니다.
   - 로컬 네트워크 외부에 익명으로 표시되어야 하는 IPv6 지원 인터페이스의 경우, 무작위로 생성된 토큰을 인터페이스 ID로 사용합니다. 지침 및 예제는 67 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
   - 정기적으로 교체하려는 IPv6 기반 인터페이스의 경우, 인터페이스 ID에 대한 토큰을 만듭니다. 지침 및 예제는 69 페이지 "사용자 지정 IPv6 토큰을 구성하는 방법"을 참조하십시오.

IPv6용 이름 서비스 지원 구성

이 절에서는 IPv6 서비스를 지원하도록 DNS 및 NIS 이름 서비스를 구성하는 방법에 대해 설명합니다.

주 – LDAP은 IPv6 관련 구성 작업 없이 IPv6을 지원합니다.

DNS, NIS 및 LDAP 관리에 대한 자세한 내용은 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**을 참조하십시오.

▼ DNS에 IPv6 주소를 추가하는 방법

1 **IPv6 지원 노드마다 AAAA 레코드를 추가하여 해당 DNS 영역 파일을 편집합니다.**

hostname `IN AAAA` *host-address*

2 **DNS 역순 영역 파일을 편집하고 PTR 레코드를 추가합니다.**

hostaddress `IN PTR` *hostname*

DNS 관리에 대한 자세한 내용은 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**을 참조하십시오.

예 4–6 DNS 역순 영역 파일

이 예는 역순 영역 파일의 IPv6 주소를 보여줍니다.

```
$ORIGIN    ip6.int.
8.2.5.0.2.1.e.f.f.f.9.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0 \
    IN        PTR        vallejo.Eng.apex.COM.
```

▼ IPv6 이름 서비스 정보를 표시하는 방법

`nslookup` 명령을 사용하여 IPv6 이름 서비스 정보를 표시할 수 있습니다.

1 **사용자 계정으로 `nslookup` 명령을 실행합니다.**

```
% /usr/sbin/nslookup
```

기본 서버 이름과 주소가 표시되고, 이어서 `nslookup` 명령의 꺾쇠 괄호 프롬프트가 표시됩니다.

2 꺾쇠 괄호 프롬프트에 다음 명령을 입력하여 특정 호스트에 대한 정보를 확인합니다.

```
>set q=any
>hostname
```

3 AAAA 레코드만 확인하려면 다음 명령을 입력합니다.

```
>set q=AAAA
hostname
```

4 exit를 입력하여 nslookup 명령을 종료합니다.

예 4–7 nslookup 명령으로 IPv6 정보 표시

이 예는 IPv6 네트워크 환경에서 nslookup의 결과를 보여줍니다.

```
%  /usr/sbin/nslookup
Default Server:  dnsserve.local.com
Address:  10.10.50.85
> set q=AAAA
> host85
Server:  dnsserve.local.com
Address:  10.10.50.85

host85.local.com      IPv6 address = 2::9256:a00:fe12:528
> exit
```

▼ DNS IPv6 PTR 레코드가 올바르게 업데이트되었는지 확인하는 방법

이 절차에서는 nslookup 명령을 사용하여 DNS IPv6용 PTR 레코드를 표시합니다.

1 사용자 계정으로 nslookup 명령을 실행합니다.

```
% /usr/sbin/nslookup
```

기본 서버 이름과 주소가 표시되고, 이어서 nslookup 명령의 꺾쇠 괄호 프롬프트가 표시됩니다.

2 꺾쇠 괄호 프롬프트에 다음을 입력하여 PTR 레코드를 표시합니다.

```
>set q=PTR
```

3 exit를 입력하여 명령을 종료합니다.

예 4–8 nslookup 명령으로 PTR 레코드 표시

다음 예는 nslookup 명령으로 표시되는 PTR 레코드를 보여줍니다.

```
% /usr/sbin/nslookup
Default Server:  space1999.Eng.apex.COM
Address:  192.168.15.78
> set q=PTR
> 8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int

8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int name =
vallejo.ipv6.Eng.apex.COM
ip6.int nameserver = space1999.Eng.apex.COM
> exit
```

▼ NIS를 통해 IPv6 정보를 표시하는 방법

이 절차에서는 ypmatch 명령을 사용하여 NIS를 통해 IPv6 정보를 표시합니다.

- **사용자 계정으로 다음을 입력하여 NIS에 IPv6 주소를 표시합니다.**

  ```
  % ypmatch hostname hosts .byname
  ```

  지정된 *hostname*에 대한 정보가 표시됩니다.

◆ ◆ ◆ 5 장

5 TCP/IP 네트워크 관리

이 장에서는 TCP/IP 네트워크 관리 작업에 대해 설명합니다. 다음 항목을 다룹니다.

주 – 네트워크 인터페이스를 모니터하려면 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 “IP 인터페이스 및 주소 모니터링”을 참조하십시오.

이 작업은 사용자의 사이트에서 TCP/IP 네트워크 즉, IPv4 전용 또는 듀얼 스택 IPv4/IPv6이 작동 가능하다고 가정합니다. 사이트에서 IPv6을 구현하려는 경우 다음 장에서 자세한 내용을 참조하십시오.

- IPv6 구현을 계획하려면 2 장, “IPv6 주소 사용 시 고려 사항”을 참조하십시오.
- IPv6을 구성하고 듀얼 스택 네트워크 환경을 만들려면 4 장, “네트워크에서 IPv6 사용”을 참조하십시오.

주요 TCP/IP 관리 작업(작업 맵)

다음 표는 초기 구성 후 네트워크를 관리하기 위한 기타 작업(예: 네트워크 정보 표시)을 보여줍니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	정보
프로토콜별 통계 표시	특정 시스템에서 네트워크 프로토콜의 성능을 모니터링합니다.	77 페이지 "프로토콜별 통계를 표시하는 방법"
네트워크 상태 표시	소켓 및 경로 지정표 항목을 모두 표시하여 시스템을 모니터링합니다. 출력에는 IPv4에 대한 주소 그룹과 IPv6에 대한 inet6 주소 그룹이 포함됩니다.	80 페이지 "소켓 상태를 표시하는 방법"
네트워크 인터페이스의 상태 표시	네트워크 인터페이스의 성능을 모니터링합니다. 이는 전송 문제를 해결하는 데 유용합니다.	79 페이지 "네트워크 인터페이스 상태를 표시하는 방법"
패킷 전송 상태 표시	회선을 통해 전송되는 패킷의 상태를 모니터링합니다.	81 페이지 "특정 주소 유형의 패킷에 대한 전송 상태를 표시하는 방법"
IPv6 관련 명령의 화면 출력 제어	`ping`, `netstat` 및 `traceroute` 명령의 출력을 제어합니다. `inet_type`이라는 파일을 만듭니다. 이 파일에서 `DEFAULT_IP` 변수를 설정합니다.	85 페이지 "IP 관련 명령의 화면 출력을 제어하는 방법"
네트워크 트래픽 모니터링	`snoop` 명령을 사용하여 모든 IP 패킷을 표시합니다.	91 페이지 "IPv6 네트워크 트래픽을 모니터링하는 방법"
네트워크 라우터에 알려진 모든 경로 추적	`traceroute` 명령을 사용하여 모든 경로를 표시합니다.	88 페이지 "모든 경로를 추적하는 방법"

주 – 네트워크 인터페이스를 모니터하려면 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스 및 주소 모니터링"을 참조하십시오.

netstat 명령으로 네트워크 상태 모니터링

netstat 명령은 네트워크 상태 및 프로토콜 통계를 표시하는 화면을 생성합니다. TCP, SCTP 및 UDP 끝점을 표 형식으로 표시할 수 있습니다. 경로 지정표 정보 및 인터페이스 정보를 표시할 수도 있습니다.

netstat 명령은 선택한 명령줄 옵션에 따라 다양한 유형의 네트워크 데이터를 표시합니다. 이러한 표시는 시스템 관리에 가장 유용합니다. netstat의 기본 구문은 다음과 같습니다.

netstat [-m] [-n] [-s] [-i | -r] [-f *address-family*]

이 절에서는 가장 일반적으로 사용되는 netstat 명령의 옵션에 대해 설명합니다. 모든 netstat 옵션에 대한 자세한 설명은 netstat(1M) 매뉴얼 페이지를 참조하십시오.

▼ 프로토콜별 통계를 표시하는 방법

netstat -s 옵션은 UDP, TCP, SCTP, ICMP 및 IP 프로토콜에 대한 프로토콜 통계를 표시합니다.

주 – Oracle Solaris 사용자 계정을 사용하여 netstat 명령의 출력을 표시할 수 있습니다.

- **프로토콜 상태를 표시합니다.**

```
$ netstat -s
```

예 5–1 네트워크 프로토콜 통계

다음 예제는 netstat -s 명령의 출력을 보여줍니다. 출력의 일부분은 잘렸습니다. 출력은 프로토콜에 문제가 있는 영역을 나타낼 수 있습니다. 예를 들어 ICMPv4 및 ICMPv6의 통계 정보는 ICMP 프로토콜에서 오류가 발견된 위치를 나타낼 수 있습니다.

```
RAWIP
        rawipInDatagrams    =  4701     rawipInErrors       =     0
        rawipInCksumErrs    =     0     rawipOutDatagrams   =     4
        rawipOutErrors      =     0

UDP
        udpInDatagrams      = 10091     udpInErrors         =     0
        udpOutDatagrams     = 15772     udpOutErrors        =     0

TCP     tcpRtoAlgorithm     =     4     tcpRtoMin           =   400
        tcpRtoMax           = 60000     tcpMaxConn          =    -1
        .
        .
        tcpListenDrop       =     0     tcpListenDropQ0     =     0
        tcpHalfOpenDrop     =     0     tcpOutSackRetrans   =     0
```

```
IPv4    ipForwarding        =     2     ipDefaultTTL        =   255
        ipInReceives        =300182     ipInHdrErrors       =     0
        ipInAddrErrors      =     0     ipInCksumErrs       =     0
        .
        .
        ipsecInFailed       =     0     ipInIPv6            =     0
        ipOutIPv6           =     3     ipOutSwitchIPv6     =     0

IPv6    ipv6Forwarding      =     2     ipv6DefaultHopLimit =   255
        ipv6InReceives      = 13986     ipv6InHdrErrors     =     0
        ipv6InTooBigErrors  =     0     ipv6InNoRoutes      =     0
        .
        .
        rawipInOverflows    =     0     ipv6InIPv4          =     0

       ipv6OutIPv4         =     0     ipv6OutSwitchIPv4   =     0

ICMPv4  icmpInMsgs          = 43593     icmpInErrors        =     0
        icmpInCksumErrs     =     0     icmpInUnknowns      =     0
        .
        .
        icmpInOverflows     =     0

ICMPv6  icmp6InMsgs         = 13612     icmp6InErrors       =     0
        icmp6InDestUnreachs =     0     icmp6InAdminProhibs =     0
        .
        .
        icmp6OutGroupQueries=     0     icmp6OutGroupResps  =     2
        icmp6OutGroupReds   =     0

IGMP:
      12287 messages received
          0 messages received with too few bytes
          0 messages received with bad checksum
      12287 membership queries received
SCTP  sctpRtoAlgorithm     =  vanj
      sctpRtoMin           =  1000
      sctpRtoMax           = 60000
      sctpRtoInitial       =  3000
      sctpTimHearBeatProbe =     2
      sctpTimHearBeatDrop  =     0
      sctpListenDrop       =     0
      sctpInClosed         =     0
```

▼ 전송 프로토콜의 상태를 표시하는 방법

netstat 명령을 통해 전송 프로토콜의 상태를 표시할 수 있습니다. 자세한 내용은 netstat(1M) 매뉴얼 페이지를 참조하십시오.

1 시스템에서 TCP 및 SCTP 전송 프로토콜의 상태를 표시합니다.

```
$ netstat
```

2 시스템에서 특정 전송 프로토콜의 상태를 표시합니다.

```
$ netstat -P transport-protocol
```

transport-protocol 변수의 값은 tcp, sctp 또는 udp입니다.

예 5–2 TCP 및 SCTP 전송 프로토콜의 상태 표시

이 예는 기본 netstat 명령의 출력을 보여줍니다. IPv4 전용 정보가 표시됩니다.

```
$ netstat

TCP: IPv4
   Local Address      Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost-1.login      ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost-1.1014     mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT
SCTP:
Local Address    Remote Address  Swind  Send-Q  Rwind  Recv-Q StrsI/O  State
---------------- --------------  -----  ------ ------ ------  ------   -------
 *.echo            0.0.0.0            0       0 102400      0   128/1   LISTEN
 *.discard         0.0.0.0            0       0 102400      0   128/1   LISTEN
 *.9001            0.0.0.0            0       0 102400      0   128/1   LISTEN
```

예 5–3 특정 전송 프로토콜의 상태 표시

이 예는 netstat 명령의 -P 옵션을 지정한 경우에 표시되는 결과를 보여줍니다.

```
$ netstat -P tcp

TCP: IPv4
   Local Address      Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost.login        ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost.1014       mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT

TCP: IPv6
 Local Address    Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------- ---------------------- ------ ----- ------ ------------ -----
localhost.38983   localhost.32777       49152      0 49152      0 ESTABLISHED
localhost.32777   localhost.38983       49152      0 49152      0 ESTABLISHED
localhost.38986   localhost.38980       49152      0 49152      0 ESTABLISHED
```

▼ 네트워크 인터페이스 상태를 표시하는 방법

netstat 명령의 i 옵션은 로컬 시스템에 구성된 네트워크 인터페이스의 상태를 보여줍니다. 이 옵션을 사용하면 시스템이 각 네트워크에서 전송하고 수신하는 패킷 수를 확인할 수 있습니다.

- **네트워크 인터페이스의 상태를 표시합니다.**

  ```
  $ netstat -i
  ```

예 5–4 네트워크 인터페이스 상태 표시

다음 예제는 호스트 인터페이스를 통한 IPv4 및 IPv6 패킷 플로우의 상태를 보여줍니다.

예를 들어 서버에 대해 표시되는 입력 패킷 수(Ipkts)는 클라이언트를 부트하려고 할 때마다 늘어나지만, 출력 패킷 수(Opkts)는 그대로 유지됩니다. 이 출력에는 서버가 클라이언트에서 보내는 부트 요청 패킷을 파악하고 있는 것으로 표시됩니다. 그러나 서버가 이에 응답하는 방법을 알지 못합니다. 이러한 혼동은 hosts 또는 ethers 데이터베이스의 주소가 잘못되었기 때문일 수 있습니다.

그러나 시간이 경과해도 입력 패킷 수가 일정할 경우 시스템에서는 패킷을 전혀 알지 못합니다. 이 출력은 다른 유형의 오류(하드웨어 문제)를 보여줍니다.

```
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      142    0     142    0     0      0
net0  1500 host58        host58        1106302 0     52419  0     0      0

Name  Mtu  Net/Dest      Address                    Ipkts  Ierrs Opkts  Oerrs Collis
lo0   8252 localhost     localhost                   142    0     142    0     0
net0  1500 fe80::a00:20ff:feb9:4c54/10 fe80::a00:20ff:feb9:4c54 1106305 0 52422 0  0
```

▼ 소켓 상태를 표시하는 방법

netstat 명령의 -a 옵션을 사용하여 로컬 호스트에 있는 소켓의 상태를 확인할 수 있습니다.

- **소켓 및 경로 지정표 항목의 상태를 표시하려면 다음을 입력합니다.**

  사용자 계정을 사용하여 netstat의 이 옵션을 실행할 수 있습니다.

  ```
  % netstat -a
  ```

예 5–5 모든 소켓 및 경로 지정표 항목 표시

netstat -a 명령의 출력은 광범위한 통계를 표시합니다. 다음 예는 일반적인 netstat -a 출력의 일부분을 보여줍니다.

```
UDP: IPv4
   Local Address         Remote Address     State
-------------------- -------------------- -------
      *.bootpc                            Idle
host85.bootpc                             Idle
      *.*                                 Unbound
      *.*                                 Unbound
      *.sunrpc                            Idle
      *.*                                 Unbound
      *.32771                             Idle
      *.sunrpc                            Idle
      *.*                                 Unbound
      *.32775                             Idle
      *.time                              Idle
```

```
      .
      .
      *.daytime                          Idle
      *.echo                             Idle
      *.discard                          Idle

UDP: IPv6
   Local Address                   Remote Address                   State      If
-------------------------------- -------------------------------- ---------- -----
      *.*                                                         Unbound
      *.*                                                         Unbound
      *.sunrpc                                                    Idle
      *.*                                                         Unbound
      *.32771                                                     Idle
      *.32778                                                     Idle
      *.syslog                                                    Idle
      .
      .
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*               0      0 49152      0 IDLE
localhost.4999             *.*               0      0 49152      0 LISTEN
      *.sunrpc             *.*               0      0 49152      0 LISTEN
      *.*                  *.*               0      0 49152      0 IDLE
      *.sunrpc             *.*               0      0 49152      0 LISTEN
      .
      .
      *.printer            *.*               0      0 49152      0 LISTEN
      *.time               *.*               0      0 49152      0 LISTEN
      *.daytime            *.*               0      0 49152      0 LISTEN
      *.echo               *.*               0      0 49152      0 LISTEN
      *.discard            *.*               0      0 49152      0 LISTEN
      *.chargen            *.*               0      0 49152      0 LISTEN
      *.shell              *.*               0      0 49152      0 LISTEN
      *.shell              *.*               0      0 49152      0 LISTEN
      *.kshell             *.*               0      0 49152      0 LISTEN
      *.login
       .
       .
            *.*               0      0 49152      0 LISTEN
   *TCP: IPv6
 Local Address            Remote Address        Swind Send-Q Rwind Recv-Q   State If
----------------------- ----------------------- ----- ------ ----- ------    ----
   *.*                      *.*                   0      0 49152      0      IDLE
   *.sunrpc                 *.*                   0      0 49152      0      LISTEN
   *.*                      *.*                   0      0 49152      0      IDLE
   *.32774                  *.*                   0      0 49152
```

▼ 특정 주소 유형의 패킷에 대한 전송 상태를 표시하는 방법

netstat 명령의 -f 옵션을 사용하면 특정 주소 그룹의 패킷 전송과 관련된 통계를 표시할 수 있습니다.

- **IPv4 또는 IPv6 패킷 전송에 대한 통계를 표시합니다.**

  ```
  $ netstat -f inet | inet6
  ```

  IPv4 전송 정보를 표시하려면 inet을 netstat -f에 대한 인수로 입력합니다. IPv6 정보를 표시하려면 inet6을 netstat -f에 대한 인수로 사용합니다.

예 5–6 IPv4 패킷 전송 상태

다음 예는 netstat -f inet 명령의 출력을 보여줍니다.

```
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
host58.734         host19.nfsd       49640      0 49640      0 ESTABLISHED
host58.38063       host19.32782      49640      0 49640      0 CLOSE_WAIT
host58.38146       host41.43601      49640      0 49640      0 ESTABLISHED
host58.996         remote-host.login 49640      0 49206      0 ESTABLISHED
```

예 5–7 IPv6 패킷 전송 상태

다음 예는 netstat -f inet6 명령의 출력을 보여줍니다.

```
TCP: IPv6
 Local Address          Remote Address        Swind Send-Q Rwind Recv-Q   State    If
------------------- ------------------------ ----- ------ ----- ------ --------- -----
localhost.38065         localhost.32792       49152   0 49152      0    ESTABLISHED
localhost.32792         localhost.38065       49152   0 49152      0    ESTABLISHED
localhost.38089         localhost.38057       49152   0 49152      0    ESTABLISHED
```

▼ 알려진 경로의 상태를 표시하는 방법

netstat 명령의 -r 옵션은 로컬 호스트의 경로 지정표를 표시합니다. 이 표는 호스트에 알려진 모든 경로의 상태를 보여줍니다. 사용자 계정에서 netstat의 이 옵션을 실행할 수 있습니다.

- **IP 경로 지정표를 표시합니다.**

  ```
  $ netstat -r
  ```

예 5–8 netstat 명령에 의한 경로 지정표 출력

다음 예는 netstat -r 명령의 출력을 보여줍니다.

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
host15               myhost               U         1  31059  net0
10.0.0.14            myhost               U         1      0  net0
default              distantrouter        UG        1      2  net0
```

```
localhost            localhost            UH        42019361  lo0

Routing Table: IPv6
  Destination/Mask      Gateway                           Flags Ref  Use   If
--------------------- --------------------------          ----- --- ------ -----
2002:0a00:3010:2::/64 2002:0a00:3010:2:1b2b:3c4c:5e6e:abcd U     1   0      net0:1
fe80::/10             fe80::1a2b:3c4d:5e6f:12a2           U     1   23     net0
ff00::/8              fe80::1a2b:3c4d:5e6f:12a2           U     1   0      net0
default               fe80::1a2b:3c4d:5e6f:12a2           UG    1   0      net0
localhost             localhost                           UH    9   21832  lo0
```

다음 표는 netstat -r 명령의 화면 출력에 표시되는 여러 매개변수의 의미에 대해 설명합니다.

매개변수	설명
Destination Destination/Mask	경로의 대상 끝점인 호스트를 지정합니다. IPv6 경로 지정표는 6to4 터널 끝점(2002:0a00:3010:2::/64)의 접두어를 경로 대상 끝점으로 표시합니다.
Gateway	패킷 전송에 사용할 게이트웨이를 지정합니다.
Flags	경로의 현재 상태를 나타냅니다. U 플래그는 경로가 작동 중임을 나타냅니다. G 플래그는 경로가 게이트웨이임을 나타냅니다.
Use	전송된 패킷 수를 표시합니다.
Interface	전송의 소스 끝점인 로컬 호스트의 특정 인터페이스를 나타냅니다.

ping 명령으로 원격 호스트 프로빙

ping 명령으로 원격 호스트의 상태를 확인할 수 있습니다. ping을 실행하면 ICMP 프로토콜에서 지정된 호스트로 데이터그램을 전송하여 응답을 요청합니다. ICMP는 TCP/IP 네트워크에서 오류 처리를 담당하는 프로토콜입니다. ping을 사용하면 지정된 원격 호스트에 대한 IP 연결이 있는지 확인할 수 있습니다.

다음은 ping의 기본 구문입니다.

/usr/sbin/ping *host [timeout]*

이 구문에서 *host*는 원격 호스트의 이름입니다. 선택적 *timeout* 인수는 ping 명령이 계속해서 원격 호스트에 연결하려고 시도하는 시간(초)을 나타냅니다. 기본값은 20초입니다. 추가 구문 및 옵션은 ping(1M) 매뉴얼 페이지를 참조하십시오.

▼ 원격 호스트가 실행 중인지 확인하는 방법

- **ping 명령을 다음과 같은 형식으로 입력합니다.**

  $ **ping** *hostname*

hostname 호스트가 ICMP 전송을 허용하는 경우 다음 메시지가 표시됩니다.

hostname is alive

이 메시지는 *hostname*이 ICMP 요청에 응답함을 나타냅니다. 그러나 *hostname*이 작동 중지되었거나 ICMP 패킷을 수신할 수 없는 경우, ping 명령으로부터 다음과 같은 응답을 수신합니다.

no answer from *hostname*

▼ 원격 호스트가 패킷을 삭제하는 중인지 확인하는 방법

ping 명령의 -s 옵션을 사용하여 원격 호스트가 실행 중이지만 패킷이 손실되고 있는지 확인할 수 있습니다.

- **ping 명령을 다음과 같은 형식으로 입력합니다.**

  $ **ping -s** *hostname*

예 5–9 패킷 삭제를 발견하기 위한 ping 출력

ping -s *hostname* 명령은 사용자가 인터럽트 문자를 전송하거나 시간 초과가 발생할 때까지 계속해서 패킷을 지정된 호스트로 전송합니다. 다음과 같은 응답이 화면에 표시됩니다.

```
& ping -s host1.domain8
PING host1.domain8 : 56 data bytes
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=0. time=1.67 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=1. time=1.02 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=2. time=0.986 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=3. time=0.921 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=4. time=1.16 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.00 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.980 ms

^C

----host1.domain8  PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 0.921/1.11/1.67/0.26
```

패킷 손실 통계는 호스트에서 패킷이 삭제되었는지 여부를 나타냅니다. ping이 실패할 경우, ipadm 및 netstat 명령으로 보고되는 네트워크 상태를 확인하십시오. **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 "IP 인터페이스 및 주소 모니터링" 및 77 페이지 "netstat 명령으로 네트워크 상태 모니터링"을 참조하십시오.

네트워크 상태 화면 관리 및 기록

다음 작업은 잘 알려진 네트워킹 명령을 사용하여 네트워크의 상태를 확인하는 방법을 보여줍니다.

▼ IP 관련 명령의 화면 출력을 제어하는 방법

IPv4 정보만 표시하거나 IPv4와 IPv6 정보를 모두 표시하도록 netstat 명령의 출력을 제어할 수 있습니다.

1 **/etc/default/inet_type 파일을 만듭니다.**

2 **다음 항목 중에서 네트워크에 필요한 항목을 /etc/default/inet_type에 추가합니다.**

- IPv4 정보만 표시

  ```
  DEFAULT_IP=IP_VERSION4
  ```

- IPv4 및 IPv6 정보 모두 표시

  ```
  DEFAULT_IP=BOTH
  ```

  또는

  ```
  DEFAULT_IP=IP_VERSION6
  ```

  inet_type 파일에 대한 자세한 내용은 inet_type(4) 매뉴얼 페이지를 참조하십시오.

주 – netstat 명령의 -f 플래그는 inet_type 파일에 설정된 값을 대체합니다.

예 5–10 IPv4 및 IPv6 정보를 선택하도록 출력 제어

- inet_type 파일에 DEFAULT_IP=BOTH 또는 DEFAULT_IP=IP_VERSION6 변수를 지정할 경우 다음과 같이 출력되어야 합니다.

  ```
  % ipadm show-addr
  ADDROBJ       TYPE       STATE   ADDR
  lo0/v4        static     ok      127.0.0.1/8
  net0/v4       static     ok      10.46.86.54/24
  lo0/v6        static     ok      ::1/128
  net0/v6       addrconf   ok      fe80::a00:fe73:56a8/10
  net0/v6add    static     ok      2001:db8:3c4d:5:a00:fe73:56a8/64
  ```

- inet_type 파일에 DEFAULT_IP=IP_VERSION4 변수를 지정할 경우 다음과 같이 출력되어야 합니다.

  ```
  % ipadm show-addr
  ADDROBJ       TYPE       STATE   ADDR
  lo0/v4        static     ok      127.0.0.1/8
  net0/v4       static     ok      10.46.86.54/24
  ```

▼ IPv4 경로 지정 데몬의 작업을 기록하는 방법

IPv4 경로 지정 데몬인 routed의 오작동이 의심되는 경우 데몬의 작업을 추적하는 로그를 시작할 수 있습니다. routed 데몬이 시작되면 이 로그에는 모든 패킷 전송이 포함됩니다.

- **경로 지정 데몬 작업에 대한 로그 파일을 만듭니다.**

  ```
  # /usr/sbin/in.routed /var/log-file-name
  ```

주의 – 사용량이 많은 네트워크에서는 이 명령이 거의 연속적으로 출력을 생성할 수 있습니다.

예 5–11 in.routed 데몬에 대한 네트워크 로그

다음 예는 86 페이지 “IPv4 경로 지정 데몬의 작업을 기록하는 방법” 절차에서 만든 로그의 시작 부분을 보여줍니다.

```
-- 2003/11/18 16:47:00.000000 --
Tracing actions started
RCVBUF=61440
Add interface lo0  #1   127.0.0.1      -->127.0.0.1/32
   <UP|LOOPBACK|RUNNING|MULTICAST|IPv4> <PASSIVE>
Add interface net0 #2   10.10.48.112    -->10.10.48.0/25
    <UP|BROADCAST|RUNNING|MULTICAST|IPv4>
turn on RIP
Add    10.0.0.0        -->10.10.48.112      metric=0  net0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  net0  <IF|NOPROP>
```

▼ IPv6 Neighbor Discovery 데몬의 작업을 추적하는 방법

IPv6 in.ndpd 데몬의 오작동이 의심되는 경우 데몬의 작업을 추적하는 로그를 시작할 수 있습니다. 이 추적은 종료될 때까지 표준 출력에 표시됩니다. in.ndpd 데몬이 시작되면 이 추적에는 모든 패킷 전송이 포함됩니다.

1. **in.ndpd 데몬의 추적을 시작합니다.**

   ```
   # /usr/lib/inet/in.ndpd -t
   ```

2. **필요한 경우 Ctrl-C를 입력하여 추적을 종료합니다.**

예 5–12 in.ndpd 데몬 추적

다음 출력은 in.ndpd 추적의 시작 부분을 보여줍니다.

```
# /usr/lib/inet/in.ndpd -t
Nov 18 17:27:28 Sending solicitation to  ff02::2 (16 bytes) on net0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:b9:4c:54>
```

```
Nov 18 17:27:28 Received valid advert from fe80::a00:20ff:fee9:2d27 (88 bytes) on net0
Nov 18 17:27:28         Max hop limit: 0
Nov 18 17:27:28         Managed address configuration: Not set
Nov 18 17:27:28         Other configuration flag: Not set
Nov 18 17:27:28         Router lifetime: 1800
Nov 18 17:27:28         Reachable timer: 0
Nov 18 17:27:28         Reachable retrans timer: 0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:e9:2d:27>
Nov 18 17:27:28         Prefix: 2001:08db:3c4d:1::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
Nov 18 17:27:28         Prefix: 2002:0a00:3010:2::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
```

traceroute 명령으로 경로 지정 정보 표시

traceroute 명령은 원격 시스템에 대한 IP 패킷의 경로를 추적합니다. traceroute에 대한 기술적인 세부 정보는 traceroute(1M) 매뉴얼 페이지를 참조하십시오.

traceroute 명령을 사용하면 잘못된 경로 지정 구성 및 경로 지정 경로 오류를 찾을 수 있습니다. 특정 호스트에 연결할 수 없는 경우 traceroute를 사용하여 원격 호스트에 대한 패킷 경로 및 오류가 발생할 수 있는 위치를 확인할 수 있습니다.

traceroute 명령은 대상 호스트에 대한 경로를 따라 전송하는 각 게이트웨이에 대한 라운드 트립 시간도 표시합니다. 이 정보는 두 노드 간의 트래픽이 느려지는 위치를 분석하는 데 유용할 수 있습니다.

▼ 원격 호스트에 대한 경로를 찾는 방법

- **원격 시스템에 대한 경로를 찾으려면 다음을 입력합니다.**

  % **traceroute** *destination-hostname*

  사용자 계정에서 traceroute 명령을 다음 형식으로 실행할 수 있습니다.

예 5–13 traceroute 명령으로 원격 호스트에 대한 경로 표시

traceroute 명령의 다음 출력은 로컬 호스트 nearhost 에서 원격 시스템 farhost로 전송되는 패킷의 7홉 경로를 보여줍니다. 이 출력은 패킷이 각 홉을 순회하는 시간도 표시합니다.

```
istanbul% traceroute farhost.faraway.com
    traceroute to farhost.faraway.com (172.16.64.39), 30 hops max, 40 byte packets
     1  frbldg7c-86 (172.16.86.1)  1.516 ms  1.283 ms  1.362 ms
```

```
 2  bldg1a-001 (172.16.1.211)  2.277 ms  1.773 ms  2.186 ms
 3  bldg4-bldg1 (172.16.4.42)  1.978 ms  1.986 ms  13.996 ms
 4  bldg6-bldg4 (172.16.4.49)  2.655 ms  3.042 ms  2.344 ms
 5  ferbldg11a-001 (172.16.1.236)  2.636 ms  3.432 ms  3.830 ms
 6  frbldg12b-153 (172.16.153.72)  3.452 ms  3.146 ms  2.962 ms
 7  sanfrancisco (172.16.64.39)  3.430 ms  3.312 ms  3.451 ms
```

▼ 모든 경로를 추적하는 방법

이 절차는 traceroute 명령의 -a 옵션을 사용하여 모든 경로를 추적합니다.

- **로컬 시스템에서 다음 명령을 입력합니다.**

  `% traceroute -a`*host-name*

  사용자 계정에서 traceroute 명령을 다음 형식으로 실행할 수 있습니다.

예 5–14 듀얼 스택 호스트에 대한 모든 경로 추적

이 예는 듀얼 스택 호스트에 대해 가능한 모든 경로를 보여줍니다.

```
% traceroute -a v6host.remote.com
traceroute: Warning: Multiple interfaces found; using 2::56:a0:a8 @ eri0:2
traceroute to v6host (2001:db8:4a3b::102:a00:fe79:19b0),30 hops max, 60 byte packets
 1  v6-rout86 (2001:db8:4a3b:56:a00:fe1f:59a1)  35.534 ms  56.998 ms *
 2  2001:db8::255:0:c0a8:717  32.659 ms  39.444 ms *
 3  farhost.faraway.COM (2001:db8:4a3b::103:a00:fe9a:ce7b)  401.518 ms  7.143 ms *
 4  distant.remote.com (2001:db8:4a3b::100:a00:fe7c:cf35)  113.034 ms  7.949 ms *
 5  v6host (2001:db8:4a3b::102:a00:fe79:19b0)  66.111 ms *  36.965 ms

traceroute to v6host.remote.com  (192.168.10.75),30 hops max,40 byte packets
 1  v6-rout86 (172.16.86.1)  4.360 ms  3.452 ms  3.479 ms
 2  flrmpj17u.here.COM (172.16.17.131)  4.062 ms  3.848 ms  3.505 ms
 3  farhost.farway.com (10.0.0.23)  4.773 ms *  4.294 ms
 4  distant.remote.com (192.168.10.104)  5.128 ms  5.362 ms *
 5  v6host  (192.168.15.85)  7.298 ms  5.444 ms *
```

snoop 명령으로 패킷 전송 모니터링

snoop 명령을 사용하여 데이터 전송 상태를 모니터링할 수 있습니다. snoop 명령은 네트워크 패킷을 캡처한 다음 사용자가 지정한 형식으로 해당 패킷의 컨텐츠를 표시합니다. 패킷은 수신 즉시 표시하거나 파일에 저장할 수 있습니다. snoop가 중간 파일에 기록할 경우 추적 사용 조건에서 패킷 손실이 발생할 가능성이 거의 없습니다. snoop 자체는 이 파일을 해석하는 데 사용됩니다.

Promiscuous 모드에서 기본 인터페이스에 대한 패킷을 캡처하려면 사용자가 네트워크 관리 역할을 사용하거나 수퍼유저여야 합니다. 요약 양식에서 snoop는 최고 레벨 프로토콜에 해당하는 데이터만 표시합니다. 예를 들어 NFS 패킷은 NFS 정보만 표시합니다. 기본 RPC, UDP, IP 및 이더넷 프레임 정보는 표시되지 않지만, 상세 정보 표시 옵션을 선택하면 표시될 수 있습니다.

snoop 명령을 자주 그리고 일관되게 사용하면 정상적인 시스템 동작에 익숙해질 수 있습니다. 패킷 분석에 대한 지원 정보는 최근 백서 및 RFC에서 특정 영역(예: NFS 또는 NIS)의 전문가 권장 사항을 참조하십시오. snoop 및 옵션 사용에 대한 자세한 내용은 snoop(1M) 매뉴얼 페이지를 참조하십시오.

▼ 모든 인터페이스의 패킷을 확인하는 방법

1 **시스템에 연결된 인터페이스에 대한 정보를 출력합니다.**

```
# ipadm show-if
```

snoop 명령은 일반적으로 첫번째 비루프백 장치(보통 기본 네트워크 인터페이스)를 사용합니다.

2 **예 5–15에 표시된 것과 같이, snoop를 인수 없이 입력하여 패킷 캡처를 시작합니다.**

3 **Ctrl-C를 사용하여 프로세스를 정지합니다.**

예 5–15 snoop 명령의 출력

기본 snoop 명령은 듀얼 스택 호스트에 대해 다음과 비슷한 출력을 반환합니다.

```
% snoop
Using device /dev/net (promiscuous mode)
router5.local.com -> router5.local.com ARP R 10.0.0.13, router5.local.com is
    0:10:7b:31:37:80
router5.local.com -> BROADCAST     TFTP Read "network-confg" (octet)
myhost -> DNSserver.local.com      DNS C 192.168.10.10.in-addr.arpa. Internet PTR ?
DNSserver.local.com  myhost        DNS R 192.168.10.10.in-addr.arpa. Internet PTR
   niserve2.
.
.
.
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (5 destinations)
```

이 출력에 캡처된 패킷은 주소 분석용 NIS 및 DNS 서버에 대한 조치를 비롯하여 원격 로그인 섹션을 보여줍니다. 로컬 라우터에서 보내는 정기 ARP 패킷 및 in.ripngd에 대한 IPv6 링크 로컬 주소 알림도 포함됩니다.

▼ snoop 출력을 파일로 캡처하는 방법

1 **snoop 세션을 파일로 캡처합니다.**

```
# snoop -o filename
```

예를 들면 다음과 같습니다.

```
# snoop -o /tmp/cap
Using device /dev/eri (promiscuous mode)
30 snoop: 30 packets captured
```

이 예에서는 패킷 30이 /tmp/cap 파일에 캡처되었습니다. 이 파일은 디스크 공간이 충분한 모든 디렉토리에 있을 수 있습니다. 캡처된 패킷 수는 명령줄에 표시되는데, Ctrl-C를 누르면 언제든지 중단할 수 있습니다.

snoop는 호스트 시스템에 많은 네트워킹 로드를 만드는데, 이로 인해 결과가 왜곡될 수 있습니다. 실제 결과를 표시하려면 세번째 시스템에서 snoop를 실행하십시오.

2 snoop 출력 캡처 파일을 검사합니다.

```
# snoop -i filename
```

예 5–16 snoop 출력 캡처 파일의 내용

다음 출력은 snoop -i 명령의 출력과 같은 다양한 캡처를 보여줍니다.

```
# snoop -i /tmp/cap
1   0.00000 fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375
    ICMPv6 Neighbor advertisement
...
10  0.91493    10.0.0.40 -> (broadcast)  ARP C Who is 10.0.0.40, 10.0.0.40 ?
34  0.43690 nearserver.here.com  -> 224.0.1.1  IP  D=224.0.1.1 S=10.0.0.40 LEN=28,
      ID=47453, TO =0x0, TTL=1
35  0.00034  10.0.0.40 -> 224.0.1.1    IP  D=224.0.1.1 S=10.0.0.40 LEN=28, ID=57376,
     TOS=0x0, TTL=47
```

▼ IPv4 서버와 클라이언트 간 패킷을 확인하는 방법

1 클라이언트 또는 서버에 연결된 허브와 떨어져 snoop 시스템을 설정합니다.

세번째 시스템(snoop 시스템)은 방해하는 모든 트래픽을 확인하므로 snoop 추적은 회선에서 실제로 발생한 사항을 반영합니다.

2 옵션과 함께 snoop를 입력한 다음 출력을 파일에 저장합니다.

3 출력 내용을 검사하고 해석합니다.

snoop 캡처 파일에 대한 자세한 내용은 RFC 1761, Snoop Version 2 Packet Capture File Format (http://www.ietf.org/rfc/rfc1761.txt?number=1761)을 참조하십시오.

▼ IPv6 네트워크 트래픽을 모니터링하는 방법

snoop 명령으로 IPv6 패킷만 표시할 수 있습니다.

- **IPv6 패킷을 캡처합니다.**

  ```
  # snoop ip6
  ```

  snoop 명령에 대한 자세한 내용은 snoop(1M) 매뉴얼 페이지를 참조하십시오

예 5–17 IPv6 네트워크 트래픽만 표시

다음 예는 노드에서 snoop ip6 명령을 실행할 경우 표시되는 출력과 같은 일반 출력을 보여줍니다.

```
# snoop ip6
fe80::a00:20ff:fecd:4374 -> ff02::1:ffe9:2d27 ICMPv6 Neighbor solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
      solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
      solicitation
fe80::a00:20ff:febb:e09 -> ff02::9      RIPng R (11 destinations)
fe80::a00:20ff:fee9:2d27 -> ff02::1:ffcd:4375 ICMPv6 Neighbor solicitation
```

IP 계층 장치를 사용하여 패킷 모니터링

IP 계층 장치는 IP 관찰을 향상하기 위해 Oracle Solaris에서 도입되었습니다. 이 장치는 시스템의 네트워크 인터페이스와 연관된 주소를 사용하는 모든 패킷에 액세스할 수 있습니다. 이 주소에는 비루프백 인터페이스 또는 논리적 인터페이스에서 호스트된 주소 및 로컬 주소가 포함됩니다. IPv4 주소와 IPv6 주소 둘 다의 트래픽을 관찰할 수 있습니다. 따라서 시스템을 대상으로 하는 모든 트래픽을 모니터링할 수 있습니다. 트래픽은 루프백 IP 트래픽, 원격 시스템에서 보내는 패킷, 시스템에서 전송 중인 패킷 또는 전송된 모든 트래픽일 수 있습니다.

IP 계층 장치를 사용하면 전역 영역 관리자가 영역 간 트래픽과 영역 내 트래픽을 모니터링할 수 있습니다. 비전역 영역의 관리자도 해당 영역에서 전송하고 수신한 트래픽을 관찰할 수 있습니다.

IP 계층에서 트래픽을 모니터링하기 위해 새 옵션인 -I가 snoop 명령에 추가되었습니다. 이 옵션은 명령이 기본 링크 계층 장치 대신 새 IP 계층 장치를 사용하여 트래픽 데이터를 표시하도록 지정합니다.

▼ IP 계층에서 패킷을 확인하는 방법

1. **필요한 경우 시스템에 연결된 인터페이스에 대한 정보를 출력합니다.**

   ```
   # ipadm show-if
   ```

2 **특정 인터페이스에서 IP 트래픽을 캡처합니다.**

```
# snoop -I interface [-V | -v]
```

패킷 확인 예

모든 예는 다음과 같은 시스템 구성을 기반으로 합니다.

```
# ipadm show-addr
ADDROBJ       TYPE        STATE   ADDR
lo0/v4        static      ok      127.0.0.1/8
net0/v4       static      ok      192.68.25.5/24
lo0/?         static      ok      127.0.0.1/8
net0/?        static      ok      172.0.0.3/24
net0/?        static      ok      172.0.0.1/24
lo0/?         static      ok      127.0.0.1/8
```

sandbox 및 toybox라는 두 영역이 다음 IP 주소를 사용한다고 가정합니다.

- sandbox – 172.0.0.3
- toybox – 172.0.0.1

시스템의 서로 다른 인터페이스에서 snoop -I 명령을 실행할 수 있습니다. 표시되는 패킷 정보는 사용자가 전역 영역 관리자인지 아니면 비전역 영역의 관리자인지 여부에 따라 달라집니다.

예 5–18 루프백 인터페이스의 트래픽

```
# snoop -I lo0
Using device ipnet/lo0 (promiscuous mode)
   localhost -> localhost    ICMP Echo request (ID: 5550 Sequence number: 0)
   localhost -> localhost    ICMP Echo reply (ID: 5550 Sequence number: 0)
```

상세 정보 출력을 생성하려면 -v 옵션을 사용하십시오.

```
# snoop -v -I lo0
Using device ipnet/lo0 (promiscuous mode)
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 1 arrived at 10:40:33.68506
IPNET:  Packet size = 108 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 4
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 0
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
...
```

IP 계층에서는 패킷 관찰이 지원되므로 새 IPNET 헤더가 관찰 중인 패킷의 앞에 표시됩니다. 소스 및 대상 ID가 모두 표시됩니다. ID '0'은 트래픽이 전역 영역에서 생성됨을 나타냅니다.

예 5–19 로컬 영역에 있는 net0 장치의 패킷 플로우

```
# snoop -I net0
Using device ipnet/net0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=62117 Syn Seq=195630514 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=62117 S=22 Syn Ack=195630515 Seq=195794440 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=62117 Ack=195794441 Seq=195630515 Len=0 Win=49152
sandbox -> toybox TCP D=62117 S=22 Push Ack=195630515 Seq=195794441 Len=20 Win=491
```

이 출력은 시스템 내의 서로 다른 영역에서 발생하는 트래픽을 보여줍니다. 로컬에서 다른 영역으로 전달되는 패킷을 비롯하여 net0 IP 주소와 연관된 모든 패킷을 확인할 수 있습니다. 상세 정보 출력을 생성하면 패킷 플로우와 관련된 영역을 확인할 수 있습니다.

```
# snoop -I net0 -v port 22
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 5 arrived at 15:16:50.85262
IPNET:  Packet size = 64 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 0
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 1
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:         .... ..0. = not ECN capable transport
IP:         .... ...0 = no ECN congestion experienced
IP:   Total length = 40 bytes
IP:   Identification = 22629
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 0000
IP:   Source address = 172.0.0.1, 172.0.0.1
IP:   Destination address = 172.0.0.3, 172.0.0.3
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 46919
TCP:  Destination port = 22
TCP:  Sequence number = 3295338550
```

예 5–19 로컬 영역에 있는 net0 장치의 패킷 플로우 *(계속)*

```
TCP:  Acknowledgement number = 3295417957
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        0... .... = No ECN congestion window reduced
TCP:        .0.. .... = No ECN echo
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP         .... 0... = No push
TCP         .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 49152
TCP:  Checksum = 0x0014
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
```

IPNET 헤더는 패킷이 전역 영역(ID 0)에서 Sandbox(ID 1)로 제공됨을 나타냅니다.

예 5–20 영역 식별을 통한 트래픽 관찰

```
# snoop -I hme0 sandboxsnoop -I net0 sandbox
Using device ipnet/hme0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=61658 Syn Seq=374055417 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=61658 S=22 Syn Ack=374055418 Seq=374124525 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=61658 Ack=374124526 Seq=374055418 Len=0 Win=49152
#
```

영역을 식별하여 패킷을 관찰하는 기능은 영역이 여러 개 있는 시스템에 유용합니다. 현재는 영역 ID로만 영역을 식별할 수 있습니다. 영역 이름과 함께 snoop를 사용하는 것은 지원되지 않습니다.

기본 주소 선택 관리

Oracle Solaris에서는 한 인터페이스에서 여러 개의 IP 주소를 사용할 수 있습니다. 예를 들어 네트워크 다중 경로(IPMP)와 같은 기술이 여러 네트워크 인터페이스 카드(NIC)를 사용하여 동일한 IP 링크 계층에 연결할 수 있도록 해줍니다. 이러한 링크는 여러 개의 IP 주소를 사용할 수 있습니다. 또한 IPv6 지원 시스템의 인터페이스에는 적어도 하나의 인터페이스에 대해 링크 로컬 IPv6 주소 하나, IPv6 경로 지정 주소 하나 이상 및 IPv4 주소 하나가 포함됩니다.

시스템에서 트랜잭션이 시작되면 응용 프로그램은 `getaddrinfo` 소켓을 호출합니다. `getaddrinfo`는 대상 시스템에서 사용 중인 가능한 주소를 검색합니다. 그러면 커널에서 이 목록의 우선 순위를 정해 패킷에 사용할 최적의 대상을 찾습니다. 이 프로세스를 **대상 주소 순서 지정**이라고 합니다. 패킷에 대한 최적의 대상 주소가 제공된 경우 Oracle Solaris 커널에서 소스 주소에 적합한 형식을 선택합니다. 이 프로세스를 **주소 선택**이라고 합니다. 대상 주소 순서 지정에 대한 자세한 내용은 `getaddrinfo(3SOCKET)` 매뉴얼 페이지를 참조하십시오.

IPv4 전용 및 듀얼 스택 IPv4/IPv6 시스템 모두 기본 주소 선택을 수행해야 합니다. 대부분의 경우에는 기본 주소 선택 방식을 변경할 필요가 없습니다. 그러나 IPMP를 지원하거나 6to4 주소 형식을 선호하는 경우 주소 형식의 우선 순위를 변경해야 할 수 있습니다.

▼ IPv6 주소 선택 정책 테이블을 관리하는 방법

다음 절차는 주소 선택 정책 테이블을 수정하는 방법에 대해 설명합니다. IPv6 기본 주소 선택에 대한 개념 정보는 ipaddrsel 명령을 참조하십시오.

주의 – 다음 작업에 표시된 이유가 아니면 IPv6 주소 선택 정책 테이블을 변경하지 마십시오. 정책 테이블이 잘못 구성된 경우 네트워크 문제가 발생할 수 있습니다. 다음 절차에서 수행된 것과 같이, 정책 테이블의 백업 복사본을 반드시 저장하십시오.

1 현재 IPv6 주소 선택 정책 테이블을 검토합니다.

```
# ipaddrsel
# Prefix                  Precedence Label
::1/128                           50 Loopback
::/0                              40 Default
2002::/16                         30 6to4
::/96                             20 IPv4_Compatible
::ffff:0.0.0.0/96                 10 IPv4
```

2 기본 주소 정책 테이블의 백업 복사본을 만듭니다.

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

3 텍스트 편집기를 사용하여 `/etc/inet/ipaddrsel.conf`에 사용자 정의 내용을 추가합니다.

`/etc/inet/ipaddrsel`의 항목에 다음 구문을 사용합니다.

prefix/prefix-length precedence label [# comment]

다음은 정책 테이블에 대해 수행할 수 있는 몇 가지 일반적인 수정 사항입니다.

- 6to4 주소에 가장 높은 우선 순위를 제공합니다.

  ```
  2002::/16                       50 6to4
  ::1/128                         45 Loopback
  ```

  이제 6to4 주소에 가장 높은 우선 순위인 50이 지정됩니다. 루프백의 경우 우선 순위가 이제 50에서 45로 변경됩니다. 기타 주소 지정 형식은 그대로 유지합니다.

- 특정 대상 주소와의 통신에 사용할 특정 소스 주소를 지정합니다.

  ```
  ::1/128                         50 Loopback
  2001:1111:1111::1/128           40 ClientNet
  2001:2222:2222::/48             40 ClientNet
  ::/0                            40 Default
  ```

이 특정 항목은 물리적 인터페이스가 한 개뿐인 호스트에 유용합니다.
2001:1111:1111::1/128은 2001:2222:2222::/48 네트워크 내에서 대상에 대해 바운드되는 모든 패킷에 대한 소스 주소로 선호됩니다. 우선 순위 40은 소스 주소 2001:1111:1111::1/128에 대한 우선 순위로, 해당 인터페이스에 대해 구성된 다른 주소 형식보다 높습니다.

- IPv6 주소보다 IPv4 주소를 선호합니다.

```
::ffff:0.0.0.0/96                60 IPv4
::1/128                          50 Loopback
.
.
```

IPv4 형식 ::ffff:0.0.0.0/96의 우선 순위가 10(기본값)에서 60(테이블의 가장 높은 우선 순위)으로 변경되었습니다.

4 수정된 정책 테이블을 커널로 로드합니다.

```
ipaddrsel -f /etc/inet/ipaddrsel.conf
```

5 수정된 정책 테이블에 문제가 있는 경우 기본 IPv6 주소 선택 정책 테이블을 복원합니다.

```
# ipaddrsel -d
```

▼ 현재 세션에 대해서만 IPv6 주소 선택 정책 테이블을 수정하는 방법

/etc/inet/ipaddrsel.conf, 파일을 편집하면 수정 사항이 재부트 후에도 지속됩니다. 수정된 정책 테이블이 현재 세션에서만 사용되도록 하려면 다음 절차를 수행하십시오.

1 /etc/inet/ipaddrsel의 내용을 *filename*으로 복사합니다. 여기서 *filename*은 사용자가 선택한 파일의 이름을 나타냅니다.

```
# cp /etc/inet/ipaddrsel filename
```

2 *filename*의 정책 테이블을 원하는 지정 사항으로 편집합니다.

3 수정된 정책 테이블을 커널로 로드합니다.

```
# ipaddrsel -f filename
```

시스템을 재부트할 때까지 커널에서 새 정책 테이블을 사용합니다.

◆ ◆ ◆ 6 장

6 IP 터널 구성

이 장에서는 IP 터널에 대한 설명 및 Oracle Solaris에서 터널을 구성 및 유지 관리하는 절차에 대해 다룹니다.

IP 터널 개요

IP 터널은 중간 네트워크에서 도메인의 프로토콜을 지원하지 않을 경우 도메인 간에 데이터 패킷을 전송할 수 있도록 해줍니다. 예를 들면 IPv6 프로토콜이 도입됨으로써 IPv6 네트워크에는 대부분의 네트워크가 IPv4 프로토콜을 사용하는 환경의 경계를 벗어나 통신할 수 있는 방법이 필요합니다. 통신은 터널을 통해 가능해집니다. IP 터널은 IP를 사용하여 연결 가능한 두 노드 간에 가상 링크를 제공합니다. 따라서 이 링크를 사용하면 IPv4 네트워크를 통해 IPv6 패킷을 전송할 수 있으므로 두 IPv6 사이트 간 IPv6 통신이 가능해집니다.

Oracle Solaris 11에서 IP 터널 관리

이 Oracle Solaris 릴리스에서는 네트워크 데이터 링크 관리를 위한 새 모델과 일치하도록 터널 관리가 수정되었습니다. 이제 터널은 새 `dladm` 하위 명령을 사용하여 생성되고 구성됩니다. 또한 터널은 새 관리 모델의 다른 데이터 링크 기능을 사용할 수도 있습니다. 예를 들어 관리상 선택한 이름이 지원되므로 터널에 의미 있는 이름을 지정할 수 있습니다. `dladm` 하위 명령에 대한 자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

터널 유형

터널링은 다른 패킷 내에서 IP 패킷을 캡슐화하는 것입니다. 캡슐화는 패킷의 프로토콜을 지원하지 않는 중간 네트워크를 통해 패킷이 대상에 도달할 수 있도록 해줍니다.

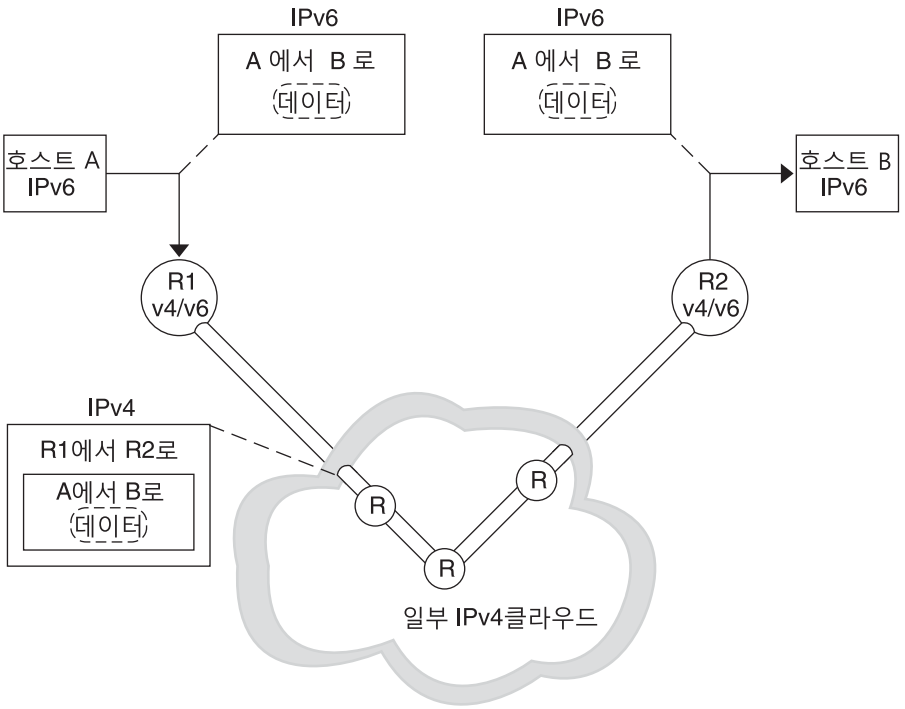
터널은 패킷 캡슐화의 유형에 따라 달라집니다. Oracle Solaris에서 지원되는 터널의 유형은 다음과 같습니다.

- *IPv4* **터널** – IPv4 또는 IPv6 패킷은 IPv4 헤더에 캡슐화되고 미리 구성된 유니캐스트 IPv4 대상으로 전송됩니다. 터널을 경유하는 패킷을 보다 명확하게 하기 위해 IPv4 터널을 *IPv4 over IPv4* **터널** 또는 *IPv6 over IPv4* **터널**이라고도 합니다.
- *IPv6* **터널** – IPv4 또는 IPv6 패킷은 IPv6 헤더에 캡슐화되고 미리 구성된 유니캐스트 IPv6 대상에 전송됩니다. 터널을 경유하는 패킷을 보다 명확하게 하기 위해 IPv6 터널을 *IPv4 over IPv6* **터널** 또는 *IPv6 over IPv6* **터널**이라고도 합니다.
- *6to4* **터널** – IPv6 패킷은 IPv4 헤더에 캡슐화되고 패킷당 기준에 따라 자동으로 결정된 IPv4 대상에 전송됩니다. 이때 6to4 프로토콜에 정의된 알고리즘을 기준으로 결정됩니다.

결합된 IPv6 및 IPv4 네트워크 환경에서의 터널

IPv6 도메인이 있는 대부분의 사이트에서는 IPv4 네트워크를 순회하여 다른 IPv6 도메인과 통신하는데, 이는 IPv6 전용 네트워크보다 IPv4 네트워크에서 더 일반적입니다. 다음 그림은 IPv4 라우터를 경유하는 두 IPv6 호스트 간의 터널링 방식을 보여줍니다. IPv4 라우터는 그림에서 "R"로 표시되어 있습니다.

그림 6–1 IPv6 터널링 방식



이 그림에서 터널은 두 개의 라우터로 구성되는데, 이 라우터는 IPv4 네트워크를 경유하여 두 라우터 간에 가상 지점 간 링크를 갖도록 구성되어 있습니다.

IPv6 패킷은 IPv4 패킷 내에서 캡슐화됩니다. IPv6 네트워크의 경계 라우터는 다양한 IPv4 네트워크를 경유하여 대상 IPv6 네트워크의 경계 라우터에 도달하는 지점 간 터널을 설정합니다. 패킷은 터널을 경유하여 대상 경계 라우터로 전송되며, 여기서 패킷이 캡슐화 해제됩니다. 그러면 라우터가 개별 IPv6 패킷을 대상 노드로 전달합니다.

6to4 터널

Oracle Solaris는 주소 지정을 IPv4에서 IPv6으로 전환하는 데 선호하는 중간 방식으로 6to4 터널을 제공합니다. 6to4 터널은 분리된 IPv6 사이트가 IPv6을 지원하지 않는 IPv4 네트워크를 경유하여 자동 터널을 넘어 통신할 수 있도록 해줍니다. 6to4 터널을 사용하려면 IPv6 네트워크의 경계 라우터를 6to4 자동 터널의 한 끝점으로 구성해야 합니다. 그러면 6to4 라우터가 다른 6to4 사이트 또는 필요한 경우 원시 IPv6, 비6to4 사이트에 대한 터널에 참여할 수 있습니다.

이 절에서는 다음 6to4 항목에 대한 참조 자료를 제공합니다.

- 6to4 터널 토폴로지
- 6to4 터널을 경유하는 패킷에 대한 설명
- 6to4 라우터와 6to4 릴레이 라우터 간 터널의 토폴로지
- 6to4 릴레이 라우터 지원을 구성하기 전에 고려할 사항

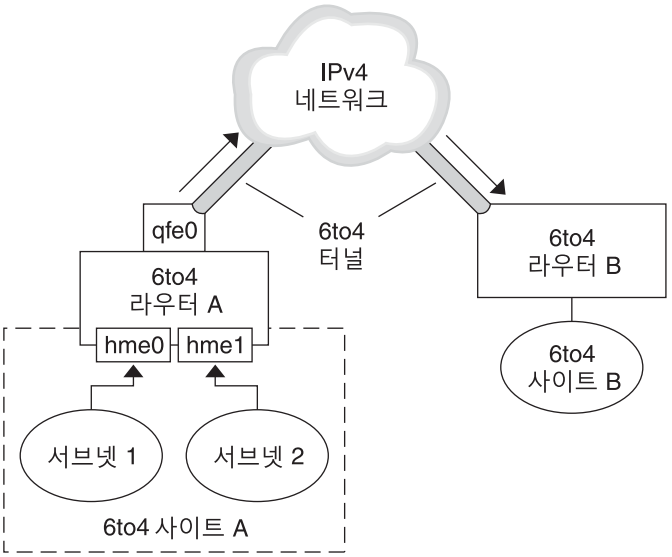
다음 표는 유용한 정보를 추가로 얻기 위해 6to4 터널 및 리소스를 구성하는 추가 작업에 대해 설명합니다.

작업 또는 세부 정보	정보
6to4 터널 구성 작업	110 페이지 “6to4 터널을 구성하는 방법”
6to4 관련 RFC	RFC 3056, "Connection of IPv6 Domains via IPv4 Clouds" (`http://www.ietf.org/rfc/rfc3056.txt`)
6to4 릴레이 라우터에 대한 터널을 지원하는 `6to4relay` 명령에 대한 세부 정보	`6to4relay`(1M)
6to4 보안 문제	Security Considerations for 6to4 (`http://www.ietf.org/rfc/rfc3964.txt`)

6to4 터널 토폴로지

6to4 터널은 모든 위치에서 모든 6to4 사이트에 대한 IPv6 연결을 제공합니다. 마찬가지로, 터널이 릴레이 라우터로 전달되도록 구성된 경우 터널은 원시 IPv6 인터넷을 비롯한 모든 IPv6 사이트에 대한 링크 역할도 합니다. 다음 그림은 6to4 터널이 6to4 사이트 간에 이러한 연결을 제공하는 방식을 보여줍니다.

그림 6–2 6to4 사이트 간 터널



이 그림은 분리된 두 6to4 네트워크인 사이트 A 및 사이트 B를 보여줍니다. 각 사이트는 IPv4 네트워크에 대한 외부 연결을 포함하는 라우터를 구성했습니다. IPv4 네트워크를 경유하는 6to4 터널은 6to4 사이트를 연결합니다.

IPv6 사이트가 6to4 사이트가 되려면 먼저 6to4 지원을 위해 적어도 하나의 라우터 인터페이스를 구성해야 합니다. 이 인터페이스는 IPv4 네트워크에 대한 외부 연결을 제공해야 합니다. `qfe0`에 구성한 주소는 전역적으로 고유해야 합니다. 이 그림에서 라우터 A의 인터페이스인 `qfe0`은 사이트 A를 IPv4 네트워크에 연결해 줍니다. `qfe0`을 6to4 의사 인터페이스로 구성하기 전에 이미 `qfe0` 인터페이스가 IPv4 주소를 사용하도록 구성되어 있어야 합니다.

그림에서 6to4 사이트 A는 두 개의 서브넷으로 구성되며, 두 서브넷은 라우터 A의 `hme0` 및 `hme1` 인터페이스에 연결됩니다. 사이트 A의 서브넷에 있는 모든 IPv6 호스트는 라우터 A로부터 알림을 수신하면 6to4 파생 주소를 사용하도록 재구성됩니다.

사이트 B는 또 다른 분리된 6to4 사이트입니다. 사이트 A에서 보내는 트래픽을 올바르게 수신하려면 사이트 B의 경계 라우터가 6to4를 지원하도록 구성되어야 합니다. 그렇지 않으면 라우터가 사이트 A로부터 수신하는 패킷이 인식되지 않고 삭제됩니다.

6to4 터널을 경유하는 패킷 플로우

이 절에서는 6to4 사이트의 호스트에서 원격 6to4 사이트의 호스트로의 패킷 플로우에 대해 설명합니다. 이 시나리오는 그림 6–2에 표시된 토폴로지를 사용합니다. 또한 이 시나리오는 6to4 라우터와 6to4 호스트가 이미 구성되어 있다고 가정합니다.

1. 6to4 사이트 A의 서브넷 1에 있는 호스트가 6to4 사이트 B에 있는 호스트를 대상으로 지정하는 전송을 보냅니다. 각 패킷 헤더에는 6to4 파생 소스 주소와 6to4 파생 대상 주소가 있습니다.
2. 사이트 A의 라우터가 IPv4 헤더 내에서 각 6to4 패킷을 캡슐화합니다. 이 프로세스에서 라우터는 캡슐화 헤더의 IPv4 대상 주소를 사이트 B의 라우터 주소로 설정합니다. 터널 인터페이스를 경유하는 각 IPv6 패킷의 IPv6 대상 주소에는 IPv4 대상 주소도 포함되어 있습니다. 따라서 라우터가 캡슐화 헤더에 설정된 IPv4 대상 주소를 확인할 수 있습니다. 그런 다음 라우터는 표준 IPv4 경로 지정 프로시저를 사용하여 IPv4 네트워크를 통해 패킷을 전달합니다.
3. 패킷이 거쳐 가는 IPv4 라우터는 전달 시 패킷의 IPv4 대상 주소를 사용합니다. 이 주소는 라우터 B에 있는 인터페이스의 전역적으로 고유한 IPv4 주소이며, 6to4 의사 인터페이스로도 사용됩니다.
4. 사이트 A의 패킷이 라우터 B에 도달하여 IPv4 헤더에서 IPv6 패킷이 캡슐화 해제됩니다.
5. 그런 다음 라우터 B가 IPv6 패킷의 대상 주소를 사용하여 패킷을 사이트 B의 수신자 호스트로 전달합니다.

6to4 릴레이 라우터에 대한 터널 고려 사항

6to4 릴레이 라우터는 원시 IPv6, 비6to4 네트워크와 통신해야 하는 6to4 라우터에서 터널 끝점으로 사용됩니다. 릴레이 라우터는 기본적으로 6to4 사이트와 원시 IPv6 사이트를 연결해 줍니다. 이 솔루션은 안전하지 않으므로 기본적으로 Oracle Solaris에서는 6to4 릴레이 라우터 지원이 사용으로 설정되어 있지 않습니다. 그러나 사이트에 이러한 터널이 필요할 경우 `6to4relay` 명령을 사용하여 다음과 같은 터널링 시나리오를 사용으로 설정할 수 있습니다.

그림 6–3 6to4 사이트와 6to4 릴레이 라우터 간 터널

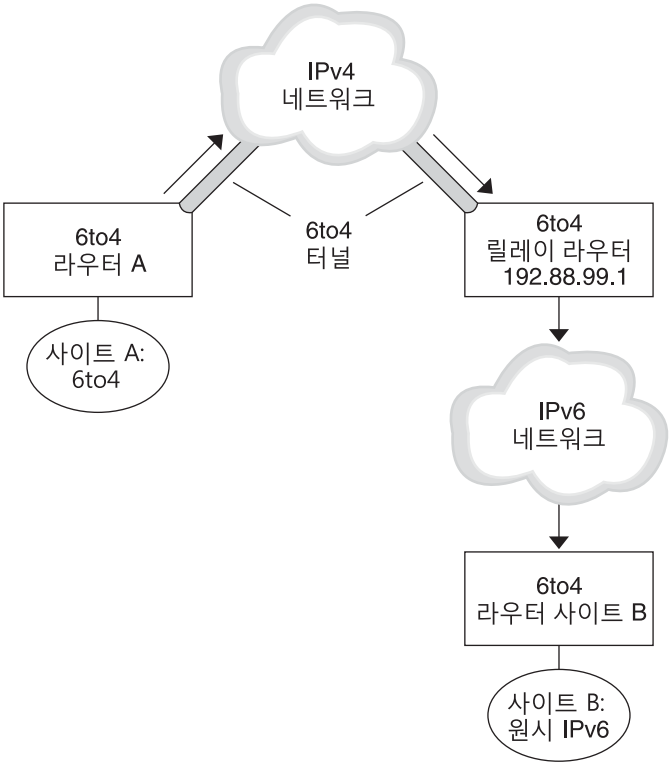


그림 6–3에서 6to4 사이트 A는 원시 IPv6 사이트 B에 있는 노드와 통신해야 합니다. 이 그림은 사이트 A에서 IPv4 네트워크를 경유하여 6to4 터널에 도달하는 트래픽 경로를 보여줍니다. 터널의 끝점은 6to4 라우터 A와 6to4 릴레이 라우터입니다. 6to4 릴레이 라우터를 넘어가면 IPv6 사이트 B가 연결되어 있는 IPv6 네트워크입니다.

6to4 사이트와 원시 IPv6 사이트 간 패킷 플로우

이 절에서는 6to4 사이트에서 원시 IPv6 사이트로의 패킷 플로우에 대해 설명합니다. 이 시나리오는 그림 6–3에 표시된 토폴로지를 사용합니다.

1. 6to4 사이트 A에 있는 호스트가 원시 IPv6 사이트 B에 있는 호스트를 대상으로 지정하는 전송을 보냅니다. 각 패킷 헤더에는 6to4 파생 주소가 소스 주소로 포함되어 있습니다. 대상 주소는 표준 IPv6 주소입니다.
2. 사이트 A의 6to4 라우터가 IPv4 헤더 내에서 각 패킷을 캡슐화합니다. 이 헤더에는 6to4 릴레이 라우터의 IPv4 주소가 대상으로 포함되어 있습니다. 6to4 라우터는 표준 IPv4 경로 지정 프로시저를 사용하여 IPv4 네트워크를 통해 패킷을 전달합니다. 패킷이 거쳐 가는 IPv4 라우터는 패킷을 6to4 릴레이 라우터로 전달합니다.

3. 사이트 A와 물리적으로 가장 가까운 애니캐스트 6to4 릴레이 라우터가 192.88.99.1 애니캐스트 그룹에 전송되는 패킷을 검색합니다.

주 – 6to4 릴레이 라우터 애니캐스트 그룹의 일부인 6to4 릴레이 라우터의 IP 주소는 192.88.99.1입니다. 이 애니캐스트 주소는 6to4 릴레이 라우터의 기본 주소입니다. 특정 6to4 릴레이 라우터를 사용해야 하는 경우 기본 주소를 대체하고 해당 라우터의 IPv4 주소를 지정할 수 있습니다.

---

4. 릴레이 라우터가 6to4 패킷에서 IPv4 헤더를 캡슐화 해제하여 원시 IPv6 대상 주소를 표시합니다.
5. 이제 패킷 라우터가 IPv6 전용 패킷을 IPv6 네트워크로 전송합니다. 이 네트워크에서 패킷이 사이트 B의 라우터에 의해 검색됩니다. 라우터가 패킷을 대상 IPv6 노드로 전달합니다.

터널 배치

IP 터널을 제대로 배치하려면 두 가지 기본 작업을 수행해야 합니다. 먼저 터널 링크를 만드십시오. 그런 다음 터널을 경유하는 IP 인터페이스를 구성하십시오. 이 절에서는 터널 및 해당 IP 인터페이스를 만들기 위한 요구 사항에 대해 설명합니다.

터널 만들기 요구 사항

터널을 성공적으로 만들려면 다음 요구 사항을 알고 있어야 합니다.

- 리터럴 IP 주소 대신 호스트 이름을 사용할 경우 해당 이름이 터널 유형과 호환되는 유효한 IP 주소로 분석되어야 합니다.
- 만드는 IPv4 또는 IPv6 터널이 구성된 다른 터널과 동일한 터널 소스 주소 및 터널 대상 주소를 공유해서는 안됩니다.
- 만드는 IPv4 또는 IPv6 터널이 기존 6to4 터널과 동일한 터널 소스 주소를 공유해서는 안됩니다.
- 6to4 터널을 만드는 경우, 터널이 구성된 다른 터널과 동일한 터널 소스 주소를 공유해서는 안됩니다.

네트워크에서 터널 설정에 대한 자세한 내용은 30 페이지 “네트워크에서 터널 사용 계획”을 참조하십시오.

터널 및 IP 인터페이스 요구 사항

각 터널 유형에는 터널을 경유하도록 구성한 IP 인터페이스에 대한 특정 IP 주소 요구 사항이 있습니다. 요구 사항은 다음 표에 요약되어 있습니다.

표 6–1 터널 및 IP 인터페이스 요구 사항

터널 유형	터널을 통해 허용되는 IP 인터페이스	IP 인터페이스 요구 사항
IPv4 터널	IPv4 인터페이스	로컬 및 원격 주소를 수동으로 지정해야 합니다.
	IPv6 인터페이스	`ipadm create-addr -T addrconf` 명령을 실행하면 로컬 및 원격 링크 로컬 주소가 자동으로 설정됩니다. 자세한 내용은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.
IPv6 터널	IPv4 인터페이스	로컬 및 원격 주소를 수동으로 지정해야 합니다.
	IPv6 인터페이스	`ipadm create-addr -T addrconf` 명령을 실행하면 로컬 및 원격 링크 로컬 주소가 자동으로 설정됩니다. 자세한 내용은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.
6to4 터널	IPv6 인터페이스만	`ipadm create-ip` 명령을 실행하면 기본 IPv6 주소가 자동으로 설정됩니다. 자세한 내용은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

6to4 터널의 기본 IPv6 인터페이스 주소는 `ipadm` 명령으로 다른 IPv6 주소를 지정하여 대체할 수 있습니다.

마찬가지로, IPv4 또는 IPv6 터널을 경유하는 IPv6 인터페이스에 대해 자동으로 설정되는 링크 로컬 주소를 대체하려면 터널의 호스트 파일에 다른 소스 및 대상 주소를 지정하면 됩니다.

dladm 명령을 통한 터널 구성 및 관리

이 절에서는 `dladm` 명령을 사용하여 터널을 구성하는 절차에 대해 설명합니다.

dladm 하위 명령

이 Oracle Solaris 릴리스부터 터널 관리가 IP 인터페이스 구성과 분리되었습니다. IP 터널의 데이터 링크 측면은 `dladm` 명령으로 관리됩니다. 또한 IP 터널 인터페이스를 비롯한 IP 인터페이스 구성은 `ipadm` 명령으로 수행됩니다.

IP 터널을 구성하는 데 사용되는 `dladm`의 하위 명령은 다음과 같습니다.

- create-iptun
- modify-iptun
- show-iptun
- delete-iptun
- set-linkprop

dladm 명령에 대한 자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

주 – IP 터널 관리는 IPsec 구성과 밀접한 관련이 있습니다. 예를 들어 IPsec VPN(Virtual Private Network)은 IP 터널링의 주요 용도 중 하나입니다. Oracle Solaris의 보안에 대한 자세한 내용은 **Oracle Solaris 11.1의 네트워크 보안**의 6 장, "IP 보안 아키텍처(개요)"를 참조하십시오. IPsec을 구성하려면 **Oracle Solaris 11.1의 네트워크 보안**의 7 장, "IPsec 구성(작업)"을 참조하십시오.

터널 구성(작업 맵)

작업	설명	수행 방법
IP 터널 만들기	네트워크를 통한 통신에 사용할 터널을 구성합니다.	106 페이지 "IP 터널을 만들고 구성하는 방법"
터널 구성 수정	터널의 소스 또는 대상 주소 등 터널의 원래 매개변수를 변경합니다.	113 페이지 "IP 터널 구성을 수정하는 방법"
터널 구성 표시	특정 터널 또는 시스템의 모든 IP 터널에 대한 구성 정보를 표시합니다.	115 페이지 "IP 터널 구성을 표시하는 방법"
터널 삭제	터널 구성을 삭제합니다.	116 페이지 "IP 터널을 삭제하는 방법"

▼ IP 터널을 만들고 구성하는 방법

1 터널을 만듭니다.

```
# dladm create-iptun [-t] -T type -a [local|remote]=addr,... tunnel-link
```

이 명령에 사용할 수 있는 옵션 및 인수는 다음과 같습니다.

`-t` 임시 터널을 만듭니다. 기본적으로 이 명령은 영구 터널을 만듭니다.

주 – 터널을 경유하는 영구 IP 인터페이스를 구성하려는 경우 `-t` 옵션을 사용하지 말고 영구 터널을 만들어야 합니다.

`-T` *type*

만들려는 터널의 유형을 지정합니다. 이 인수는 모든 터널 유형을 만드는 데 필수입니다.

`-a [local|remote]=`*address,...*

로컬 주소 및 원격 터널 주소에 해당하는 리터럴 IP 주소 또는 호스트 이름을 지정합니다. 주소는 유효해야 하며 이미 시스템에 생성되어 있어야 합니다. 터널의 유형에 따라 주소를 한 개만 지정하거나, 로컬 및 원격 주소를 모두 지정합니다. 로컬 및 원격 주소를 모두 지정하는 경우 주소를 쉼표로 구분해야 합니다.

- `IPv4` 터널이 작동하려면 로컬 및 원격 IPv4 주소가 필요합니다.
- `IPv6` 터널이 작동하려면 로컬 및 원격 IPv6 주소가 필요합니다.
- `6to4` 터널이 작동하려면 로컬 IPv4 주소가 필요합니다.

주 – 영구 IP 터널 데이터 링크 구성에 호스트 이름을 주소로 사용하는 경우 호스트 이름은 구성 저장소에 저장됩니다. 이후에 시스템을 부트할 때 이름이 터널을 만든 당시에 사용했던 IP 주소와 다른 IP 주소로 분석되는 경우 터널이 새 구성을 사용하게 됩니다.

tunnel-link

IP 터널 링크를 지정합니다. 네트워크 링크 관리에서 의미 있는 이름이 지원되는 경우, 터널 이름이 더 이상 만들려는 터널의 유형으로 제한되지 않습니다. 대신 관리상 선택한 이름을 터널에 지정할 수 있습니다. 터널 이름은 문자열과 PPA(Physical Point of Attachment) 번호로 구성됩니다(예: *mytunnel0*). 의미 있는 이름의 지정과 관련된 규칙을 보려면 **Oracle Solaris 11 네트워킹 소개**의 "유효한 링크 이름 규칙"을 참조하십시오.

터널 링크를 지정하지 않으면 다음과 같은 이름 지정 규칙에 따라 자동으로 이름이 제공됩니다.

- IPv4 터널: `ip.tun#`
- IPv6 터널: `ip6.tun#`

- 6to4 터널: `ip.6to4tun#`

#은 만드는 터널 유형에 사용 가능한 가장 낮은 PPA 번호입니다.

2 (옵션) 홉 한계 또는 캡슐화 한계에 대한 값을 설정합니다.

```
# dladm set-linkprop -p [hoplimit=value] [encaplimit=value] tunnel-link
```

hoplimit — IPv6 경유 터널링에 대한 터널 인터페이스의 홉 한계를 지정합니다. *hoplimit*는 IPv4 경유 터널의 IPv4 TTL(time to live) 필드에 해당합니다.

encaplimit — 패킷에 허용되는 중첩 터널링의 레벨 수를 지정합니다. 이 옵션은 IPv6 터널에만 적용됩니다.

패킷에 허용되는 중첩 터널링의 레벨 수를 지정합니다. 이 옵션은 IPv6 터널에만 적용됩니다.

주 – `hoplimit` 및 `encaplimit`에 대해 설정한 값은 허용되는 범위 내에 있어야 합니다. `hoplimit` 및 `encaplimit`는 터널 링크 등록 정보입니다. 따라서 이러한 등록 정보는 다른 링크 등록 정보의 경우와 동일한 `dladm` 하위 명령으로 관리됩니다. 해당 하위 명령은 `dladm set-linkprop`, `dladm reset-linkprop` 및 `dladm show-linkprop`입니다. 링크 관리를 위해 `dladm` 명령과 함께 사용되는 여러 하위 명령은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

---

3 터널을 경유하는 IP 인터페이스를 만듭니다.

```
# ipadm create-ip tunnel-interface
```

여기서 *tunnel-interface*는 터널 링크와 동일한 이름을 사용합니다.

4 터널 인터페이스에 로컬 및 원격 IP 주소를 지정합니다.

```
# ipadm create-addr [-t] -a local=address,remote=address interface
```

`-t` — 터널을 경유하는 영구 IP 구성이 아닌 임시 IP 구성을 나타냅니다. 이 옵션을 사용하지 않으면 IP 인터페이스 구성은 영구 구성이 됩니다.

`-a local=`*address*`,remote=`*address* — 터널 인터페이스의 IP 주소를 지정합니다. 소스 및 대상 IP 주소가 모두 필수이며, `local` 및 `remote`로 표시됩니다. 로컬 및 원격 주소는 IPv4 또는 IPv6 주소일 수 있습니다.

interface — 터널 인터페이스를 지정합니다.

`ipadm` 명령에 대한 자세한 내용 및 터널 인터페이스를 포함하여 IP 인터페이스를 구성하기 위한 서로 다른 옵션들에 대한 자세한 내용은 `ipadm(1M)` 매뉴얼 페이지 및 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**을 참조하십시오.

5 **터널 구성 정보를 /etc/hosts 파일에 추가합니다.**

6 **(옵션) 터널 IP 인터페이스 구성의 상태를 확인합니다.**

```
# ipadm show-addr interface
```

예 6–1 IPv4 터널을 경유하는 IPv6 인터페이스 만들기

이 예제에서는 지속적인 IPv6 over IPv4 터널을 만드는 방법을 보여줍니다.

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 private0
# dladm set-linkprop -p hoplimit=200 private0
# ipadm create-ip private0
# ipadm create-addr -T addrconf private0
# ipadm show-addr private/
ADDROBJ       TYPE     STATE   ADDR
private0/v6   static   ok      fe80::a08:392e/10 --> fe80::8191:9a56
```

대체 주소를 추가하려면 동일한 구문을 사용합니다. 예를 들어 다음과 같이 전역 주소를 추가할 수 있습니다.

```
# ipadm create-addr -a local=2001:db8:4728::1, \
remote=2001:db8:4728::2 private0
# ipadm show-addr private0/
ADDROBJ          TYPE       STATE   ADDR
private0/v6      addrconf   ok      fe80::a08:392e/10 --> fe80::8191:9a56
private0/v6a     static     ok      2001:db8:4728::1 --> 2001:db8:4728::2
```

IPv6 주소의 2001:db8 접두어는 설명서 예제에 특별히 사용되는 특수 IPv6 접두어입니다.

예 6–2 IPv4 터널을 경유하는 IPv4 인터페이스 만들기

이 예제에서는 지속적인 IPv4 over IPv4 터널을 만드는 방법을 보여줍니다.

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 vpn0
# ipadm create-ip vpn0
# ipadm create-addr -a local=10.0.0.1,remote=10.0.0.2 vpn0
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADDR
lo0/v4    static   ok      127.0.0.1
vpn0/v4   static   ok      10.0.0.1-->10.0.0.2
```

이 터널을 경유하는 패킷에 보안 연결을 제공하도록 IPsec 정책을 추가로 구성할 수 있습니다. IPsec 구성에 대한 자세한 내용은 **Oracle Solaris 11.1의 네트워크 보안**의 7 장, “IPsec 구성(작업)”을 참조하십시오.

예 6–3 IPv6 터널을 경유하는 IPv6 인터페이스 만들기

이 예제에서는 지속적인 IPv6 over IPv6 터널을 만드는 방법을 보여줍니다.

```
# dladm create-iptun -T ipv6 -a local=2001:db8:feed::1234,remote=2001:db8:beef::4321 \
tun0
# ipadm create-ip tun0
# ipadm create-addr -T addrconf tun0
# ipadm show-addr
ADDROBJ   TYPE       STATE   ADDR
lo0/v6    static     ok      ::1/128
tun0/v6   addrconf   ok      2001:db8:feed::1234 --> 2001:db8:beef::4321
```

전역 주소 또는 대체 로컬 및 원격 주소 등의 주소를 추가하려면 다음과 같이 ipadm 명령을 사용하십시오.

```
# ipadm create-addr \
-a local=2001:db8::4728:56bc,remote=2001:db8::1428:57ab tun0
# ipadm show-addr tun0
ADDROBJ    TYPE      STATE ADDR
tun0/v6    addrconf  ok    2001:db8:feed::1234 --> 2001:db8:beef::4321
tun0/v6a   static    ok    2001:db8::4728:56bc --> 2001:db8::1428:57ab
```

▼ 6to4 터널을 구성하는 방법

6to4 터널에서 6to4 라우터는 네트워크의 6to4 사이트에 있는 노드에 대한 IPv6 라우터로 사용되어야 합니다. 따라서 6to4 라우터를 구성할 때 물리적 인터페이스에서 해당 라우터가 IPv6 라우터로도 구성되어야 합니다. IPv6 경로 지정에 대한 자세한 내용은 140 페이지 “IPv6 경로 지정”을 참조하십시오.

1 6to4 터널을 만듭니다.

```
# dladm create-iptun -T 6to4 -a local=address  tunnel-link
```

이 명령에 사용할 수 있는 옵션 및 인수는 다음과 같습니다.

-a local=*address* 터널 로컬 주소를 지정합니다. 이 주소가 시스템에 이미 존재해야 유효한 주소입니다.

tunnel-link IP 터널 링크를 지정합니다. 네트워크 링크 관리에서 의미 있는 이름이 지원되는 경우, 터널 이름이 더 이상 만들려는 터널의 유형으로 제한되지 않습니다. 대신 관리상 선택한 이름을 터널에 지정할 수 있습니다. 터널 이름은 문자열과 PPA 번호로 구성됩니다(예: *mytunnel0*). 의미 있는 이름의 지정과 관련된 규칙을 보려면 **Oracle Solaris 11 네트워킹 소개**의 “유효한 링크 이름 규칙”을 참조하십시오.

2 터널 IP 인터페이스를 만듭니다.

```
# ipadm create-ip tunnel-interface
```

여기서 *tunnel-interface*는 터널 링크와 동일한 이름을 사용합니다.

3 **(옵션) 터널용 대체 IPv6 주소를 추가합니다.**

4 **다음 두 행을 추가하여 6to4 경로 지정을 알리도록 /etc/inet/ndpd.conf 파일을 편집합니다.**

```
if subnet-interface AdvSendAdvertisements 1
IPv6-address subnet-interface
```

첫번째 행은 알림을 수신하는 서브넷을 지정합니다. *subnet-interface*는 서브넷이 연결되어 있는 링크를 나타냅니다. 두번째 행의 IPv6 주소에는 6to4 터널의 IPv6 주소에 사용되는 6to4 접두어 2000이 지정됩니다.

ndpd.conf 파일에 대한 자세한 내용은 ndpd.conf(4) 매뉴얼 페이지를 참조하십시오.

5 **IPv6 전달을 사용으로 설정합니다.**

```
# ipadm set-prop -p forwarding=on ipv6
```

6 **라우터를 재부트합니다.**

또는 /etc/inet/in.ndpd 데몬에 대해 sighup을 실행하여 라우터 알림 전송을 시작할 수 있습니다. 6to4 접두어를 수신하기 위해 각 서브넷의 IPv6 노드가 이제 새 6to4 파생 주소로 자동 구성됩니다.

7 **6to4 사이트에서 사용되는 이름 서비스에 노드의 새 6to4 파생 주소를 추가합니다.**

지침은 72 페이지 “IPv6용 이름 서비스 지원 구성”을 참조하십시오.

예 6–4 6to4 터널 만들기

이 예에서 서브넷 인터페이스는 bge0이며, /etc/inet/ndpd.conf가 적합한 단계에서 이 인터페이스를 참조하게 됩니다.

이 예제에서는 6to4 터널을 만드는 방법을 보여줍니다. IPv6 인터페이스만 6to4 터널을 경유하도록 구성할 수 있습니다.

```
# dladm create-iptun -T 6to4 -a local=192.168.35.10 tun0
# ipadm create-ip tun0
# ipadm show-addr
ADDROBJ       TYPE     STATE   ADDR
lo0/v4        static   ok      127.0.0.1/8
net0/v4       static   ok      192.168.35.10/24
lo0/v6        static   ok      ::1/128
tun0/_a       static   ok      2002:c0a8:57bc::1/64

# ipadm create-addr -a 2002:c0a8:230a::2/16 tun0
# ipadm create-addr -a 2002:c0a8:230a::3/16 tun0
# ipadm show-addr tun0
ADDROBJ       TYPE     STATE   ADDR
lo0/v4        static   ok      127.0.0.1/8
net0/v4       static   ok      192.168.35.10/24
lo0/v6        static   ok      ::1/128
```

```
tun0/_a         static   ok      2002:c0a8:57bc::1/64
tun0/v6         static   ok      2002:c0a8:230a::2/16
tun0/v6a        static   ok      2002:c0a8:230a::3/16

# vi /etc/inet/ndpd.conf
if bge0 AdvSendAdvertisements 1
2002:c0a8:57bc::1/64 bge0

# ipadm set-prop -p forwarding=on ipv6
```

6to4 터널에 대한 IPv6 주소 접두어는 2002입니다.

▼ 6to4 릴레이 라우터에 대한 6to4 터널을 구성하는 방법

주의 – 주요 보안 문제로 인해 6to4 릴레이 라우터 지원은 기본적으로 Oracle Solaris에서 사용 안함으로 설정되어 있습니다. **Troubleshooting Network Issues**의 "Security Issues When Tunneling to a 6to4 Relay Router"을 참조하십시오.

시작하기 전에 6to4 릴레이 라우터에 대한 터널을 사용으로 설정하기 전에 다음 작업을 수행해야 합니다.

- 사이트에서 6to4 라우터 구성(106 페이지 "IP 터널을 만들고 구성하는 방법"에 설명됨)
- 6to4 릴레이 라우터에 대한 터널링과 관련된 보안 문제 검토

1 다음 형식 중 하나를 사용하여 6to4 릴레이 라우터에 대한 터널을 사용으로 설정합니다.

- 애니캐스트 6to4 릴레이 라우터에 대한 터널을 사용으로 설정합니다.

  ```
  # /usr/sbin/6to4relay -e
  ```

  -e 옵션은 6to4 라우터와 애니캐스트 6to4 릴레이 라우터 간에 터널을 설정합니다. 애니캐스트 6to4 릴레이 라우터는 잘 알려진 IPv4 주소 192.88.99.1을 사용합니다. 사용자의 사이트와 물리적으로 가장 가까운 애니캐스트 릴레이 라우터가 6to4 터널의 끝점이 됩니다. 이 릴레이 라우터는 6to4 사이트와 원시 IPv6 사이트 간 패킷 전달을 처리합니다.

  애니캐스트 6to4 릴레이 라우터에 대한 자세한 내용은 RFC 3068, "An Anycast Prefix for 6to4 Relay Routers" (ftp://ftp.rfc-editor.org/in-notes/rfc3068.txt)를 참조하십시오.

- 특정 6to4 릴레이 라우터에 대한 터널을 사용으로 설정합니다.

  ```
  # /usr/sbin/6to4relay -e -a relay-router-address
  ```

-a 옵션은 특정 라우터 주소가 뒤에 이어짐을 나타냅니다. *relay-router-address*는 터널을 사용으로 설정할 특정 6to4 릴레이 라우터의 IPv4 주소로 바꿉니다.

6to4 릴레이 라우터에 대한 터널은 6to4 터널 의사 인터페이스를 제거할 때까지 활성 상태로 유지됩니다.

2 터널이 더 이상 필요하지 않을 경우 6to4 릴레이 라우터에 대한 터널을 삭제합니다.

```
# /usr/sbin/6to4relay -d
```

3 (옵션) 6to4 릴레이 라우터에 대한 터널이 재부트 후에도 보존되도록 합니다.

6to4 라우터가 재부트될 때마다 사이트에서 6to4 릴레이 라우터에 대한 터널을 원래 상태로 복원해야 하는 이유가 있을 수 있습니다. 이 시나리오를 지원하려면 다음을 수행해야 합니다.

a. /etc/default/inetinit 파일을 편집합니다.

파일의 맨 끝 행을 수정해야 합니다.

b. ACCEPT6TO4RELAY=NO 행의 "NO" 값을 "YES"로 변경합니다.

c. (옵션) 재부트 후에도 보존되는 특정 6to4 릴레이 라우터에 대한 터널을 만듭니다.

RELAY6TO4ADDR 매개변수에 대해 192.88.99.1 주소를 사용하려는 6to4 릴레이 라우터의 IPv4 주소로 변경합니다.

예 6–5 6to4 릴레이 라우터 지원에 대한 상태 정보 가져오기

/usr/bin/6to4relay 명령을 사용하여 6to4 릴레이 라우터에 대한 지원을 사용으로 설정할지 여부를 확인할 수 있습니다. 다음 예는 6to4 릴레이 라우터에 대한 지원이 사용 안함으로 설정된 경우(Oracle Solaris의 기본값)의 출력을 보여줍니다.

```
# /usr/sbin/6to4relay
6to4relay: 6to4 Relay Router communication support is disabled.
```

6to4 릴레이 라우터에 대한 지원이 사용으로 설정되면 다음과 같은 출력이 표시됩니다.

```
# /usr/sbin/6to4relay
6to4relay: 6to4 Relay Router communication support is enabled.
IPv4 remote address of Relay Router=192.88.99.1
```

▼ IP 터널 구성을 수정하는 방법

- **터널 구성을 변경합니다.**

```
# dladm modify-iptun -a [local|remote]=addr,... tunnel-link
```

기존 터널의 유형은 수정할 수 없습니다. 따라서 -T *type* 옵션은 이 명령에 사용할 수 없습니다. 수정 가능한 터널 매개변수는 다음과 같습니다.

-a [local|remote]=*address,...* 로컬 주소 및 원격 터널 주소에 해당하는 리터럴 IP 주소 또는 호스트 이름을 지정합니다. 터널의 유형에 따라 주소를 한 개만 지정하거나, 로컬 및 원격 주소를 모두 지정합니다. 로컬 및 원격 주소를 모두 지정하는 경우 주소를 쉼표로 구분해야 합니다.

- IPv4 터널이 작동하려면 로컬 및 원격 IPv4 주소가 필요합니다.
- IPv6 터널이 작동하려면 로컬 및 원격 IPv6 주소가 필요합니다.
- 6to4 터널이 작동하려면 로컬 IPv4 주소가 필요합니다.

영구 IP 터널 데이터 링크 구성에 호스트 이름을 주소로 사용하는 경우 호스트 이름은 구성 저장소에 저장됩니다. 이후에 시스템을 부트할 때 이름이 터널을 만든 당시에 사용했던 IP 주소와 다른 IP 주소로 분석되는 경우 터널이 새 구성을 사용하게 됩니다.

터널의 로컬 및 원격 주소를 변경하는 경우 해당 주소가 수정하려는 터널의 유형과 일치하는지 확인합니다.

주 – 터널 링크의 이름을 변경하려면 modify-iptun 하위 명령을 사용하지 마십시오. 대신 dladm rename-link를 사용하십시오.

```
# dladm rename-link old-tunnel-link new-tunnel-link
```

마찬가지로, hoplimit 또는 encaplimit와 같은 터널 등록 정보를 변경하려면 modify-iptun 명령을 사용하지 마십시오. 대신 dladm set-linkprop 명령을 사용하여 해당 등록 정보의 값을 설정하십시오.

예 6–6 터널의 주소 및 등록 정보 수정

이 예는 두 개의 절차로 구성됩니다. 먼저 IPv4 터널 vpn0의 로컬 및 원격 주소가 일시적으로 변경됩니다. 나중에 시스템을 재부트하면 터널이 원래 주소를 사용하도록 복원됩니다. 두번째 절차는 vpn0의 hoplimit를 60으로 변경합니다.

```
# dladm modify-iptun -t -a local=10.8.48.149,remote=192.1.2.3 vpn0

# dladm set-linkprop -p hoplimit=60 vpn0
```

▼ IP 터널 구성을 표시하는 방법

- **IP 터널 구성을 표시합니다.**

  # dladm show-iptun [-p] -o *fields* **[***tunnel-link***]**

  이 명령과 함께 사용할 수 있는 옵션은 다음과 같습니다.

  `-p` 시스템에서 분석 가능한 형식으로 정보를 표시합니다. 이 인수는 선택적입니다.

  `-o` *fields* 특정 터널 정보를 표시하는 선택한 필드를 표시합니다.

  tunnel-link 표시할 구성 정보를 포함하는 터널을 지정합니다. 이 인수는 선택적입니다. 터널 이름을 생략하면 시스템에 있는 모든 터널에 대한 정보가 표시됩니다.

예 6–7 모든 터널에 대한 정보 표시

이 예에서는 한 개의 터널만 시스템에 존재합니다.

```
# dladm show-iptun
LINK     TYPE     FLAGS     LOCAL          REMOTE
tun0     6to4     --        192.168.35.10  --
vpn0     ipv4     --        10.8.48.149    192.1.2.3
```

예 6–8 시스템에서 분석 가능한 형식으로 선택한 필드 표시

이 예에서는 터널 정보를 포함하는 특정 필드만 표시됩니다.

```
# dladm show-iptun -p -o link,type,local
tun0:6to4:192.168.35.10
vpn0:ipv4:10.8.48.149
```

▼ IP 터널 등록 정보를 표시하는 방법

- **터널 링크의 등록 정보를 표시합니다.**

  # dladm show-linkprop [-c] [-o *fields***] [***tunnel-link***]**

  이 명령과 함께 사용할 수 있는 옵션은 다음과 같습니다.

  `-c` 시스템에서 분석 가능한 형식으로 정보를 표시합니다. 이 인수는 선택적입니다.

  `-o` *fields* 링크 등록 정보에 대한 특정 정보를 제공하는 선택한 필드를 표시합니다.

  tunnel-link 표시할 등록 정보에 대한 정보를 포함하는 터널을 지정합니다. 이 인수는 선택적입니다. 터널 이름을 생략하면 시스템에 있는 모든 터널에 대한

정보가 표시됩니다.

예 6–9 터널 등록 정보 표시

이 예는 터널의 링크 등록 정보를 모두 표시하는 방법을 보여줍니다.

```
# dladm show-linkprop tun0
LINK     PROPERTY     PERM     VALUE     DEFAULT     POSSIBLE
tun0     autopush     --       --        --          --
tun0     zone         rw       --        --          --
tun0     state        r-       up        up          up,down
tun0     mtu          r-       65515     --          576-65495
tun0     maxbw        rw       --        --          --
tun0     cpus         rw       --        --          --
tun0     priority     rw       high      high        low,medium,high
tun0     hoplimit     rw       64        64          1-255
```

▼ IP 터널을 삭제하는 방법

1 인터페이스의 유형에 따라 적절한 구문을 사용하여 터널을 경유하도록 구성된 IP 인터페이스를 제거합니다.

```
# ipadm delete-ip tunnel-link
```

주 – 터널을 성공적으로 삭제하기 위해서는 터널에 설정된 기존 IP 인터페이스가 없어야 합니다.

---

2 IP 터널을 삭제합니다.

```
# dladm delete-iptun tunnel-link
```

이 명령의 유일한 옵션은 터널을 일시적으로 삭제하는 `-t`입니다. 시스템을 재부트하면 터널이 복원됩니다.

예 6–10 IPv6 인터페이스로 구성된 IPv6 터널 삭제

이 예에서는 영구 터널이 영구적으로 삭제됩니다.

```
# ipadm delete-ip ip6.tun0
# dladm delete-iptun ip6.tun0
```

◆ ◆ ◆ 7 장

7 IPv4 참조

이 장에서는 파일 항목의 유형, 용도 및 형식을 포함하여 네트워크 구성 파일에 대한 TCP/IP 네트워크 참조 정보를 제공합니다.

이 장은 다음 정보를 포함합니다.

TCP/IP 구성 파일

네트워크에서 구성 정보는 네트워크의 작동 방식을 규제하는 여러 파일과 데이터베이스에 저장됩니다. 이 절에서는 이러한 파일에 대한 간략한 설명을 제공합니다. 네트워크에 대한 변경 사항을 구현할 때 일부 파일은 업데이트 및 유지 관리가 필요합니다. 거의 또는 전혀 관리가 필요하지 않은 파일도 있습니다.

`/etc/defaultrouter` 이 파일에는 네트워크에 직접 연결된 라우터의 IP 인터페이스 이름이 포함됩니다. 시스템에서 이 파일은 선택 사항입니다. 파일이 존재할 경우 시스템은 정적 경로 지정을 지원하도록 구성됩니다.

`/etc/inet/hosts` 이 파일에는 네트워크의 IPv4 주소와 이 주소를 구성하는 해당 인터페이스 이름이 포함됩니다. NIS 또는 DNS 이름 서비스나 LDAP 디렉토리 서비스를 사용하는 경우 호스트 정보는 서버에 존재하는 다른 데이터베이스(예: `hosts.byname`)에 저장됩니다. 자세한 내용은 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**을 참조하십시오.

`/etc/inet/netmasks` 이 파일에는 네트워크 번호(예: `192.168.0.0`) 및 해당 네트워크 번호의 네트마스크 정보(예: `255.255.255.0`)가 포함됩니다. NIS 또는 LDAP를 사용하는 네트워크에서 이 정보는 서버의

	네트마스크 데이터베이스에 저장됩니다. 자세한 내용은 netmasks(4) 매뉴얼 페이지를 참조하십시오.
/etc/bootparams	이 파일에는 네트워크 클라이언트 모드로 부트하도록 구성된 시스템에 대한 부트 프로세스를 결정하는 매개변수가 포함됩니다. 자세한 내용은 38 페이지 "시스템 구성 모드 설정"을 참조하십시오. 이 파일은 로컬 파일 모드를 사용하지 않는 경우 이름 서비스에서 사용하는 bootparams 데이터베이스를 만들기 위한 기준입니다. 파일의 내용 및 형식에 대한 자세한 내용은 bootparams(4) 매뉴얼 페이지를 참조하십시오.
/etc/ethers	이 파일은 호스트 이름과 해당 MAC 주소를 연결합니다. 이 파일은 시스템이 네트워크 클라이언트로 구성된 네트워크에서 사용할 ethers 데이터베이스를 만들기 위한 기준입니다. 자세한 내용은 ethers(4) 매뉴얼 페이지를 참조하십시오.
/etc/inet/networks	이 파일은 네트워크 이름과 네트워크 번호를 연관시켜 놓았습니다. 주석으로 데이터베이스의 각 항목에 대한 부연 설명을 추가할 수도 있습니다. 이 파일이 있기 때문에 응용 프로그램에서 네트워크 번호 대신 네트워크 이름을 사용하고 표시할 수 있습니다. 예를 들어, netstat 프로그램은 이 데이터베이스의 정보를 사용하여 상태 테이블을 생성합니다. 라우터를 통해 로컬 네트워크에 연결하는 모든 부속 네트워크는 이 파일에 포함되어야 합니다. 자세한 내용은 networks(4) 매뉴얼 페이지를 참조하십시오.
/etc/inet/protocols	이 파일은 시스템에 설치된 TCP/IP 프로토콜 및 해당 프로토콜 번호를 나열합니다. 이 파일은 관리가 거의 필요하지 않습니다. 자세한 내용은 protocols(4) 매뉴얼 페이지를 참조하십시오.
/etc/inet/services	이 파일은 TCP와 UDP 서비스의 이름 및 잘 알려진 해당 포트 번호를 나열합니다. 이 파일은 네트워크 서비스를 호출하는 프로그램에서 사용됩니다. 일반적으로 이 파일은 관리가 필요하지 않습니다. 자세한 내용은 services(4) 매뉴얼 페이지를 참조하십시오.

inetd Internet Services Daemon

inetd 데몬은 시스템이 부트할 때 인터넷 표준 서비스를 시작하고 시스템이 실행 중일 때 서비스를 다시 시작할 수 있습니다. inetd 데몬에서 시작되는 표준 인터넷 서비스를 수정하거나 서비스를 추가하려면 SMF(서비스 관리 기능)를 사용합니다.

inetd에서 시작되는 서비스를 관리하려면 다음 SMF 명령을 사용합니다.

`svcadm` 서비스에 대한 관리 작업(사용으로 설정, 사용 안함으로 설정 또는 다시 시작 등)에 사용됩니다. 자세한 내용은 `svcadm(1M)` 매뉴얼 페이지를 참조하십시오.

`svcs` 서비스 상태 쿼리에 사용됩니다. 자세한 내용은 `svcs(1)` 매뉴얼 페이지를 참조하십시오.

`inetadm` 서비스의 등록 정보 표시 및 수정에 사용됩니다. 자세한 내용은 `inetadm(1M)` 매뉴얼 페이지를 참조하십시오.

특정 서비스에 대한 `inetadm` 프로파일의 `proto` 필드는 서비스가 실행되는 전송 계층 프로토콜을 나타냅니다. 서비스가 IPv4 전용인 경우 `proto` 필드가 `tcp`, `udp` 또는 `sctp`로 지정되어야 합니다.

- SMF 명령 사용 지침은 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 "SMF 명령줄 관리 유틸리티"를 참조하십시오.
- SMF 명령을 사용하여 SCTP를 통해 실행되는 서비스를 추가하는 작업은 57 페이지 "SCTP 프로토콜을 사용하는 서비스를 추가하는 방법"을 참조하십시오.
- IPv4 요청과 IPv6 요청을 모두 처리하는 서비스 추가에 대한 자세한 내용은 118 페이지 "inetd Internet Services Daemon"을 참조하십시오.

name-service/switch SMF 서비스

`name-service/switch` SMF 서비스는 구성 정보에 대한 네트워크 데이터베이스의 검색 순서를 정의합니다. 이전에 구성 파일에 저장되었던 네트워크 구성 정보 중 일부(예: 기본 도메인)는 이 SMF 서비스의 등록 정보가 되도록 변환되었습니다. 이 SMF 서비스의 등록 정보는 시스템에서 이름 서비스의 구현을 결정합니다. 등록 정보는 다음과 같습니다.

```
% svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring     solaris.smf.value.name-service.switch
config/default              astring     files
config/password             astring     "files nis"
config/group                astring     "files nis"
config/host                 astring     "files dns nis"
config/network              astring     "nis [NOTFOUND=return] files"
config/protocol             astring     "nis [NOTFOUND=return] files"
config/rpc                  astring     "nis [NOTFOUND=return] files"
config/ether                astring     "nis [NOTFOUND=return] files"
config/netmask              astring     "files nis"
config/bootparam            astring     "nis [NOTFOUND=return] files"
config/publickey            astring     "nis [NOTFOUND=return] files"
config/netgroup             astring     nis
config/automount            astring     "files nis"
config/alias                astring     "files nis"
config/service              astring     "files nis"
config/printer              astring     "user nis"
config/auth_attr            astring     "files nis"
```

```
config/prof_attr            astring             "files nis"
config/project              astring             "files nis"
```

각 등록 정보에 대해 설정된 값은 네트워크 사용자에게 영향을 주는 정보(예: 암호, 별칭 또는 네트워크 마스크)를 검색하는 이름 서비스를 결정합니다. 예를 들어, 자동 마운트 및 암호 등록 정보는 `files` 및 `nis`로 설정됩니다. 따라서 자동 마운트 정보 및 암호 정보는 파일과 NIS 서비스에서 가져옵니다.

한 이름 서비스에서 다른 이름 서비스로 변경하려는 경우 선택한 이름 서비스를 사용으로 설정하도록 `name-service/switch` SMF 서비스의 해당 등록 정보를 설정해야 합니다.

예를 들어, 네트워크에서 LDAP 이름 지정 서비스를 사용하려는 경우를 가정해 보겠습니다. SMF 서비스의 다음 등록 정보를 구성해야 합니다.

- `config/default`가 파일 및 LDAP를 사용하도록 설정되어야 합니다.
- `config/host`가 파일 및 DNS를 사용하도록 설정되어야 합니다.
- `config/netgroup`이 LDAP를 사용하도록 설정되어야 합니다.
- `config/printer`가 사용자, 파일 및 LDAP를 사용하도록 설정되어야 합니다.

그러므로 이러한 등록 정보를 올바르게 설정하려면 다음 명령을 입력해야 합니다.

```
# svccfg -s name-service/switch setprop config/default = astring: '"files ldap"'
# svccfg -s name-service/switch setprop config/host = astring: '"files dns"'
# svccfg -s name-service/switch setprop config/netgroup = astring: '"ldap"'
# svccfg -s name-service/switch setprop config/printer = astring: '"user files ldap"'
# svccfg -s name-service/switch:default refresh
```

이름 서비스 스위치에 대한 자세한 내용은 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**을 참조하십시오.

네트워크 데이터베이스에 대한 이름 서비스의 영향

네트워크 데이터베이스의 형식은 해당 네트워크에 대해 선택하는 이름 서비스의 유형에 따라 달라집니다. 예를 들어, `hosts` 데이터베이스에는 적어도 로컬 시스템의 호스트 이름과 IPv4 주소 및 로컬 시스템에 직접 연결된 네트워크 인터페이스가 포함됩니다. 하지만 `hosts` 데이터베이스에는 네트워크의 서비스 이름 유형에 따라 다른 IPv4 주소와 호스트 이름이 포함될 수 있습니다.

네트워크 데이터베이스는 다음과 같이 사용됩니다.

- 이름 서비스에 대해 로컬 파일을 사용하는 네트워크는 `/etc/inet` 및 `/etc` 디렉토리의 파일에 의존합니다.
- NIS는 NIS 맵이라는 데이터베이스를 사용합니다.
- DNS는 호스트 정보가 있는 레코드를 사용합니다.

주 – DNS 부트 및 데이터 파일은 네트워크 데이터베이스에 직접 연결되지 않습니다.

NIS, DNS 및 LDAP에서 네트워크 데이터베이스 연결에 대한 자세한 내용은 **Oracle Solaris 11.1에서 이름 지정 및 디렉토리 서비스 작업**을 참조하십시오.

Oracle Solaris의 경로 지정 프로토콜

이 절에서는 Oracle Solaris에서 지원되는 두 가지 경로 지정 프로토콜인 RIP(Routing Information Protocol) 및 RDISC(ICMP Router Discovery)에 대해 설명합니다. RIP 및 RDISC는 모두 표준 TCP/IP 프로토콜입니다. Oracle Solaris에서 사용 가능한 전체 경로 지정 프로토콜 목록은 표 7–1 및 표 7–2를 참조하십시오.

RIP(Routing Information Protocol)

RIP은 시스템이 부트할 때 자동으로 시작되는 경로 지정 데몬인 `in.routed`로 구현됩니다. 라우터에서 `s` 옵션을 지정하여 실행하면 `in.routed`는 커널 경로 지정 테이블을 모든 접근 가능한 네트워크에 대한 경로로 채우고 모든 네트워크 인터페이스를 통해 "접근 가능성"을 알립니다.

호스트에서 `q` 옵션을 지정하여 실행하면 `in.routed`는 경로 지정 정보를 추출하지만 접근 가능성을 알리지는 않습니다. 호스트에서 경로 지정 정보는 두 가지 방법으로 추출할 수 있습니다.

- `S` 플래그(대문자 "S": "공간 절약 모드")를 지정하지 **않습니다**. `in.routed`는 라우터에서 만드는 것과 동일하게 전체 경로 지정 테이블을 만듭니다.
- `S` 플래그를 지정합니다. `in.routed`는 각 사용 가능한 라우터에 대해 단일 기본 경로가 포함된 최소 커널 테이블을 만듭니다.

RDISC(ICMP Router Discovery) 프로토콜

호스트는 RDISC를 사용하여 라우터에서 경로 지정 정보를 가져옵니다. 따라서 호스트에서 RDISC를 실행하는 경우 라우터 정보를 교환하려면 라우터도 다른 프로토콜(예: RIP)을 실행해야 합니다.

RDISC는 라우터와 호스트에서 모두 실행되어야 하는 `in.routed`로 구현됩니다. 호스트에서 `in.routed`는 RDISC를 사용하여 RDISC를 통해 자신을 알리는 라우터에서 기본 경로를 찾습니다. 라우터에서 `in.routed`는 RDISC를 사용하여 직접 연결된 네트워크의 호스트에 기본 경로를 알립니다. `in.routed(1M)` 매뉴얼 페이지 및 `gateways(4)` 매뉴얼 페이지를 참조하십시오.

Oracle Solaris의 경로 지정 프로토콜 표

다음 표는 Oracle Solaris에서 지원되는 모든 경로 지정 프로토콜을 나열합니다.

표 7–1 Oracle Solaris 경로 지정 프로토콜

프로토콜	연결된 데몬	설명	수행 방법
RIP(Routing Information Protocol)	`in.routed`	IPv4 패킷을 경로 지정하고 경로 지정 테이블을 유지 관리하는 IGP입니다.	43 페이지 “IPv4 라우터 구성 방법”
ICMP(Internet Control Message Protocol) 라우터 검색	`in.routed`	호스트에서 네트워크의 라우터를 검색하는 데 사용됩니다.	51 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법” 및 53 페이지 “단일 인터페이스 시스템에서 동적 경로 지정을 사용으로 설정하는 방법”
RIPng(Routing Information Protocol, next generation) 프로토콜	`in.ripngd`	IPv6 패킷을 경로 지정하고 경로 지정 테이블을 유지 관리하는 IGP입니다.	64 페이지 “IPv6 지원 라우터를 구성하는 방법”
ND(Neighbor Discovery) 프로토콜	`in.ndpd`	IPv6 라우터의 존재를 알리고 네트워크의 IPv6 호스트를 검색합니다.	61 페이지 “IPv6 인터페이스 구성”

다음 표는 Oracle Solaris에서도 지원되는 오픈 소스 Quagga 경로 지정 프로토콜 제품군을 나열합니다.

표 7–2 오픈 소스 Quagga 프로토콜

프로토콜	데몬	설명
RIP 프로토콜	`ripd`	IPv4 패킷을 경로 지정하고 주변에 경로 지정 테이블을 알리는 IPv4 거리 벡터링 IGP입니다.
RIPng	`ripngd`	IPv6 거리 벡터링 IGP입니다. IPv6 패킷을 경로 지정하고 경로 지정 테이블을 유지 관리합니다.
OSPF(Open Shortest Path First) 프로토콜	`ospfd`	패킷 경로 지정 및 고가용성 네트워킹을 위한 IPv4 링크 상태 IGP입니다.
BGP(Border Gateway Protocol)	`bgpd`	관리 도메인 간에 경로 지정을 위한 IPv4 및 IPv6 EGP입니다.

◆ ◆ ◆ 8 장

8 IPv6 참조

이 장에서는 Oracle Solaris IPv6 구현에 대한 다음 참조 정보에 대해 설명합니다.

IPv6 지원 네트워크 구성에 대한 작업은 4 장, “네트워크에서 IPv6 사용”을 참조하십시오. IP 터널에 대한 모든 정보는 6 장, “IP 터널 구성”을 참조하십시오.

Oracle Solaris IPv6 구현

이 절에서는 Oracle Solaris에서 IPv6을 사용하는 파일, 명령 및 데몬에 대해 설명합니다.

IPv6 구성 파일

이 절에서는 IPv6 구현에 포함된 구성 파일에 대해 설명합니다.

ndpd.conf 구성 파일

/etc/inet/ndpd.conf 파일은 in.ndpd Neighbor Discovery 데몬에서 사용하는 옵션을 구성하는 데 사용됩니다. 라우터의 경우 주로 ndpd.conf를 사용하여 링크에 알릴 사이트 접두어를 구성하십시오. 호스트의 경우 ndpd.conf를 사용하여 주소 자동 구성을 해제하거나 임시 주소를 구성하십시오.

다음 표는 ndpd.conf 파일에 사용되는 키워드를 보여줍니다.

표 8–1 /etc/inet/ndpd.conf 키워드

변수	설명
ifdefault	모든 인터페이스에 대한 라우터 동작을 지정합니다. 라우터 매개변수 및 해당 값을 설정하려면 다음 구문을 사용하십시오. ifdefault [*variable-value*]
prefixdefault	접두어 알림에 대한 기본 동작을 지정합니다. 라우터 매개변수 및 해당 값을 설정하려면 다음 구문을 사용하십시오. prefixdefault [*variable-value*]
if	인터페이스별 매개변수를 설정합니다. 다음 구문을 사용하십시오. if *interface* [*variable-value*]
prefix	인터페이스별 접두어 정보를 알립니다. 다음 구문을 사용하십시오. prefix *prefix/length interface* [*variable-value*]

ndpd.conf 파일에서 이 표의 키워드를 라우터 구성 변수 세트와 함께 사용하십시오. 이러한 변수는 RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (http://www.ietf.org/rfc/rfc2461.txt?number=2461)에 자세히 정의되어 있습니다.

다음 표는 인터페이스 구성에 사용되는 변수를 간략한 설명과 함께 보여줍니다.

표 8–2 /etc/inet/ndpd.conf 인터페이스 구성 변수

변수	기본값	정의
AdvRetransTimer	0	라우터에서 보내는 알림 메시지의 Retrans Timer(재전송 타이머) 필드 값을 지정합니다.
AdvCurHopLimit	인터넷의 현재 반경	라우터에서 보내는 알림 메시지의 현재 홉 한계로 지정할 값을 지정합니다.
AdvDefaultLifetime	3 + MaxRtrAdvInterval	라우터 알림의 기본 수명을 지정합니다.
AdvLinkMTU	0	라우터에서 전송할 MTU(최대 전송 단위) 값을 지정합니다. 0은 라우터에 MTU 옵션이 지정되지 않았음을 나타냅니다.
AdvManaged Flag	False	라우터 알림의 Manage Address Configuration(주소 구성 관리) 플래그에 지정할 값을 나타냅니다.
AdvOtherConfigFlag	False	라우터 알림의 Other Stateful Configuration(기타 Stateful 구성) 플래그에 지정할 값을 나타냅니다.
AdvReachableTime	0	라우터에서 보내는 알림 메시지의 Reachable Time(연결 가능 시간) 필드 값을 지정합니다.

표 8–2 /etc/inet/ndpd.conf 인터페이스 구성 변수 (계속)

변수	기본값	정의
AdvSendAdvertisements	False	노드가 알림을 전송하고 라우터 요청에 응답할지 여부를 나타냅니다. 라우터 알림 기능을 설정하려면 ndpd.conf 파일에서 이 변수를 명시적으로 "TRUE"로 설정해야 합니다. 자세한 내용은 64 페이지 "IPv6 지원 라우터를 구성하는 방법"을 참조하십시오.
DupAddrDetect Transmits	1	로컬 노드 주소의 중복 주소 감지 중 Neighbor Discovery 프로토콜이 보내야 하는 연속 이웃 요청 메시지 수를 정의합니다.
MaxRtrAdvInterval	600초	요청되지 않은 멀티캐스트 알림을 보내는 최대 간격을 지정합니다.
MinRtrAdvInterval	200초	요청되지 않은 멀티캐스트 알림을 보내는 최소 간격을 지정합니다.
StatelessAddrConf	True	Stateless 주소 자동 구성을 통해 노드에서 IPv6 주소가 구성되는지 여부를 제어합니다. ndpd.conf에서 False가 선언된 경우 주소를 수동으로 구성해야 합니다. 자세한 내용은 69 페이지 "사용자 지정 IPv6 토큰을 구성하는 방법"을 참조하십시오.
TmpAddrsEnabled	False	모든 인터페이스에 대해 또는 노드의 특정 인터페이스에 대해 임시 주소를 생성할지 여부를 나타냅니다. 자세한 내용은 67 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
TmpMaxDesyncFactor	600초	in.ndpd가 시작되면 선호 수명 변수 TmpPreferredLifetime에서 차감될 임의 값을 지정합니다. TmpMaxDesyncFactor 변수의 목적은 네트워크에 있는 모든 시스템이 임시 주소를 동시에 재생성하지 않도록 하는 것입니다. TmpMaxDesyncFactor를 사용하여 해당 임의 값에 대한 상한을 변경할 수 있습니다.
TmpPreferredLifetime	False	임시 주소의 선호 수명을 설정합니다. 자세한 내용은 67 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
TmpRegenAdvance	False	임시 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 자세한 내용은 67 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
TmpValidLifetime	False	임시 주소의 유효 수명을 설정합니다. 자세한 내용은 67 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.

다음 표는 IPv6 접두어 구성에 사용되는 변수를 보여줍니다.

표 8–3 /etc/inet/ndpd.conf 접두어 구성 변수

변수	기본값	정의
AdvAutonomousFlag	True	Prefix Information(접두어 정보) 옵션의 Autonomous Flag(자동 플래그) 필드에 지정할 값을 지정합니다.
AdvOnLinkFlag	True	Prefix Information(접두어 정보) 옵션의 온-링크 플래그("L-bit")에 지정할 값을 지정합니다.
AdvPreferredExpiration	Not set	접두어의 선호 만료 날짜를 지정합니다.
AdvPreferredLifetime	604800초	Prefix Information(접두어 정보) 옵션의 선호 수명에 지정할 값을 지정합니다.
AdvValidExpiration	Not set	접두어의 유효 만료 날짜를 지정합니다.
AdvValidLifetime	2592000초	구성할 접두어의 유효 수명을 지정합니다.

예 8–1 /etc/inet/ndpd.conf 파일

다음 예는 ndpd.conf 파일에서 키워드 및 구성 변수가 사용되는 방식을 보여줍니다. 변수를 활성화하려면 주석(#)을 제거하십시오.

```
# ifdefault       [variable-value ]*
# prefixdefault [variable-value ]*
# if ifname     [variable-value ]*
# prefix prefix/length  ifname
#
#  Per interface configuration variables
#
#DupAddrDetectTransmits
#AdvSendAdvertisements
#MaxRtrAdvInterval
#MinRtrAdvInterval
#AdvManagedFlag
#AdvOtherConfigFlag
#AdvLinkMTU
#AdvReachableTime
#AdvRetransTimer
#AdvCurHopLimit
#AdvDefaultLifetime
#
# Per Prefix:  AdvPrefixList configuration variables
#
#
#AdvValidLifetime
#AdvOnLinkFlag
#AdvPreferredLifetime
#AdvAutonomousFlag
#AdvValidExpiration
#AdvPreferredExpiration

ifdefault AdvReachableTime 30000 AdvRetransTimer 2000
prefixdefault AdvValidLifetime 240m AdvPreferredLifetime 120m
```

예 8–1 /etc/inet/ndpd.conf 파일 (계속)

```
if qe0 AdvSendAdvertisements 1
prefix 2:0:0:56::/64 qe0
prefix fec0:0:0:56::/64 qe0

if qe1 AdvSendAdvertisements 1
prefix 2:0:0:55::/64 qe1
prefix fec0:0:0:56::/64 qe1

if hme1 AdvSendAdvertisements 1
prefix  2002:8192:56bb:1::/64 qfe0

if hme1 AdvSendAdvertisements 1
prefix  2002:8192:56bb:2::/64 hme1
```

/etc/inet/ipaddrsel.conf 구성 파일

/etc/inet/ipaddrsel.conf 파일에는 IPv6 기본 주소 선택 정책 테이블이 포함되어 있습니다. IPv6이 사용 가능한 상태로 Oracle Solaris를 설치하면 이 파일에는 표 8–4에 표시된 내용이 포함됩니다.

/etc/inet/ipaddrsel.conf의 내용은 편집할 수 있습니다. 그러나 대부분의 경우 이 파일을 수정하지 않는 것이 좋습니다. 수정이 필요할 경우 95 페이지 “IPv6 주소 선택 정책 테이블을 관리하는 방법” 절차를 참조하십시오. ippaddrsel.conf에 대한 자세한 내용은 128 페이지 “IPv6 주소 선택 정책 테이블을 수정하는 이유” 및 ipaddrsel.conf(4) 매뉴얼 페이지를 참조하십시오.

IPv6 관련 명령

이 절에서는 Oracle Solaris IPv6 구현으로 추가된 명령에 대해 설명합니다. 또한 IPv6을 지원하도록 기존 명령을 수정하는 방법에 대해서도 설명합니다.

ipaddrsel 명령

ipaddrsel 명령을 사용하여 IPv6 기본 주소 선택 정책 테이블을 수정할 수 있습니다.

Oracle Solaris 커널은 IPv6 기본 주소 선택 정책 테이블을 사용하여 IPv6 패킷 헤더에 대한 대상 주소 순서 지정 및 소스 주소 선택을 수행합니다. /etc/inet/ipaddrsel.conf 파일에는 정책 테이블이 포함되어 있습니다.

다음 표는 정책 테이블의 기본 주소 형식 및 우선 순위를 보여줍니다. IPv6 주소 선택에 대한 기술적인 세부 정보는 inet6(7P) 매뉴얼 페이지를 참조하십시오.

표 8–4 IPv6 주소 선택 정책 테이블

접두어	우선 순위	정의
::1/128	50	루프백
::/0	40	기본값
2002::/16	30	6to4
::/96	20	IPv4 호환 가능
::ffff:0:0/96	10	IPv4

이 표에서 IPv6 접두어(::1/128 및 ::/0)가 6to4 주소(2002::/16) 및 IPv4 주소(::/96 및 ::ffff:0:0/96)보다 우선적으로 사용됩니다. 따라서 기본적으로 커널은 다른 IPv6 대상으로 이동하는 패킷에 대해 전역 IPv6 주소의 인터페이스를 선택합니다. IPv4 주소의 인터페이스는 특히 IPv6 대상으로 이동하는 패킷에 대해 낮은 우선 순위를 갖습니다. 선택한 IPv6 소스 주소가 제공될 경우, 커널에서는 대상 주소에 대해 IPv6 형식도 사용됩니다.

IPv6 주소 선택 정책 테이블을 수정하는 이유

대부분의 경우에는 IPv6 기본 주소 선택 정책 테이블을 변경할 필요가 없습니다. 정책 테이블을 관리해야 하는 경우 ipaddrsel 명령을 사용하십시오.

다음과 같은 경우에 정책 테이블을 수정할 수 있습니다.

- 시스템 인터페이스가 6to4 터널에 사용되는 경우, 6to4 주소에 더 높은 우선 순위를 제공할 수 있습니다.
- 특정 소스 주소를 특정 대상 주소와의 통신에만 사용하려는 경우, 이 주소를 정책 테이블에 추가하면 됩니다. 그런 다음 ipadm을 사용하여 이 주소를 선호 주소로 플래그 지정할 수 있습니다. ipadm 명령에 대한 자세한 내용은 ipadm(1M) 매뉴얼 페이지를 참조하십시오
- IPv4 주소가 IPv6 주소보다 우선적으로 사용되게 하려는 경우, ::ffff:0:0/96의 우선 순위를 더 높은 숫자로 변경할 수 있습니다.
- 제거된 주소에 더 높은 우선 순위를 지정해야 하는 경우, 제거된 주소를 정책 테이블에 추가하면 됩니다. 예를 들어 사이트 로컬 주소는 이제 IPv6에서 제거되었습니다. 이러한 주소의 앞에는 fec0::/10이 붙습니다. 사이트 로컬 주소에 더 높은 우선 순위를 제공하도록 정책 테이블을 변경할 수 있습니다.

ipaddrsel 명령에 대한 자세한 내용은 ipaddrsel(1M) 매뉴얼 페이지를 참조하십시오.

6to4relay 명령

6to4 **터널링**을 사용하면 분리된 6to4 사이트 간에 통신할 수 있습니다. 그러나 원시 비6to4 IPv6 사이트를 포함하는 패킷을 전송하려면 6to4 라우터가 6to4 릴레이 라우터를

사용하여 터널을 설정해야 합니다. 그러면 *6to4* **릴레이 라우터**가 6to4 패킷을 IPv6 네트워크 및 원시 IPv6 사이트로 전송합니다. 6to4 지원 사이트가 원시 IPv6 사이트와 데이터를 교환해야 하는 경우 `6to4relay` 명령을 사용하여 해당 터널을 사용으로 설정하십시오.

릴레이 라우터 사용은 보안되지 않으므로 Oracle Solaris에서는 기본적으로 릴레이 라우터가 사용 안함으로 설정되어 있습니다. 이 시나리오를 배치하기 전에 6to4 릴레이 라우터에 대한 터널 생성과 관련된 문제를 주의 깊게 고려하십시오. 6to4 릴레이 라우터에 대한 자세한 내용은 102 페이지 “6to4 릴레이 라우터에 대한 터널 고려 사항”을 참조하십시오. 6to4 릴레이 라우터 지원을 사용하려는 경우 106 페이지 “IP 터널을 만들고 구성하는 방법”에서 관련 절차를 참조하십시오.

6to4relay 구문

`6to4relay` 명령의 구문은 다음과 같습니다.

```
6to4relay -e [-a IPv4-address] -d -h
```

`-e` 6to4 라우터와 애니캐스트 6to4 릴레이 라우터 간 터널에 대한 지원을 사용으로 설정합니다. 그러면 터널 끝점 주소가 `192.88.99.1`(6to4 릴레이 라우터의 애니캐스트 그룹에 대한 기본 주소)로 설정됩니다.

`-a` *IPv4-address* 지정된 *IPv4-address*를 사용하여 6to4 라우터와 6to4 릴레이 라우터 간 터널에 대한 지원을 사용으로 설정합니다.

`-d` 6to4 릴레이 라우터에 대한 터널링 지원을 사용 안함으로 설정합니다. 이는 Oracle Solaris의 기본값입니다.

`-h` `6to4relay`에 대한 도움말을 표시합니다.

자세한 내용은 `6to4relay`(1M) 매뉴얼 페이지를 참조하십시오.

예 8–2 6to4y 릴레이 라우터 지원의 기본 상태 표시

인수가 없는 `6to4relay` 명령은 6to4 릴레이 라우터 지원의 현재 상태를 표시합니다. 이 예는 IPv6의 Oracle Solaris 구현에 대한 기본값을 보여줍니다.

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is disabled
```

예 8–3 6to4 릴레이 라우터 지원을 사용으로 설정하여 상태 표시

릴레이 라우터 지원이 사용으로 설정된 경우, `6to4relay`는 다음과 같은 출력을 표시합니다.

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is enabled
IPv4 destination address of Relay Router=192.88.99.1
```

예 8–4 6to4 릴레이 라우터를 지정하여 상태 표시

`6to4relay` 명령에 `-a` 옵션과 IPv4 주소를 지정한 경우, `-a`와 함께 제공한 IPv4 주소가 `192.88.99.1` 대신 표시됩니다.

`6to4relay`는 `-d`, `-e` 및 `-a` *IPv4 address* 옵션이 성공적으로 실행되면 이를 보고하지 않습니다. 그러나 이러한 옵션을 실행할 때 생성될 수 있는 오류 메시지는 `6to4relay`가 표시합니다.

IPv6 지원을 위한 netstat 명령 수정 사항

`netstat` 명령이 IPv4 및 IPv6 네트워크 상태를 모두 표시합니다. `/etc/default/inet_type` 파일에서 `DEFAULT_IP` 값을 설정하거나 `-f` 명령줄 옵션을 사용하여 표시할 프로토콜 정보를 선택할 수 있습니다. `DEFAULT_IP`를 영구적으로 설정하면 `netstat`가 IPv4 정보만 표시합니다. `-f` 옵션을 사용하여 이 설정을 대체할 수 있습니다. `inet_type` 파일에 대한 자세한 내용은 `inet_type(4)` 매뉴얼 페이지를 참조하십시오.

`netstat` 명령의 `-p` 옵션은 net-to-media 테이블(IPv4의 경우 ARP 테이블이고, IPv6의 경우 이웃 캐시임)을 표시합니다. 자세한 내용은 `netstat(1M)` 매뉴얼 페이지를 참조하십시오. 이 명령의 사용 절차에 대한 설명은 80 페이지 “소켓 상태를 표시하는 방법”을 참조하십시오.

IPv6 지원을 위한 snoop 명령 수정 사항

`snoop` 명령이 IPv4 및 IPv6 패킷을 모두 캡처할 수 있습니다. 이 명령은 IPv6 헤더, IPv6 확장 헤더, ICMPv6 헤더 및 Neighbor Discovery 프로토콜 데이터를 표시할 수 있습니다. 기본적으로 `snoop` 명령은 IPv4 및 IPv6 패킷을 모두 표시합니다. `ip` 또는 `ip6` 프로토콜 키워드를 지정하면 `snoop` 명령은 IPv4 또는 IPv6 패킷만 표시합니다. IPv6 필터 옵션을 사용하여 IPv6 패킷만 표시하도록 모든 패킷(IPv4 및 IPv6)을 필터링할 수 있습니다. 자세한 내용은 `snoop(1M)` 매뉴얼 페이지를 참조하십시오. `snoop` 명령 사용 절차는 91 페이지 “IPv6 네트워크 트래픽을 모니터링하는 방법”을 참조하십시오.

IPv6 지원을 위한 route 명령 수정 사항

`route` 명령이 IPv4 및 IPv6 경로 모두에서 작동합니다. 이때 기본값은 IPv4 경로입니다. 명령줄에서 `route` 명령 바로 뒤에 `-inet6` 옵션을 사용하면 작업이 IPv6 경로에서 수행됩니다. 자세한 내용은 `route(1M)` 매뉴얼 페이지를 참조하십시오.

IPv6 지원을 위한 ping 명령 수정 사항

`ping` 명령이 IPv4 및 IPv6 프로토콜을 모두 사용하여 대상 호스트를 프로빙합니다. 이름 서버가 지정된 대상 호스트에 대해 반환하는 주소에 따라 프로토콜 선택이 달라집니다. 기본적으로 이름 서버가 대상 호스트에 대해 IPv6 주소를 반환하는 경우 `ping` 명령은

IPv6 프로토콜을 사용합니다. 이름 서버가 IPv4 주소만 반환하는 경우 ping 명령은 IPv4 프로토콜을 사용합니다. -A 명령줄 옵션을 사용하여 사용할 프로토콜을 지정하면 이 작업이 대체됩니다.

자세한 내용은 ping(1M) 매뉴얼 페이지를 참조하십시오. ping 사용 절차는 83 페이지 “ping 명령으로 원격 호스트 프로빙”을 참조하십시오.

IPv6 지원을 위한 traceroute 명령 수정 사항

traceroute 명령을 사용하여 특정 호스트에 대해 IPv4 및 IPv6 경로를 추적할 수 있습니다. 프로토콜 관점에서 traceroute는 ping과 동일한 알고리즘을 사용합니다. 이 선택을 대체하려면 -A 명령줄 옵션을 사용하십시오. -a 명령줄 옵션을 사용하면 멀티홈 호스트의 각 주소에 대해 개별 경로를 추적할 수 있습니다.

자세한 내용은 traceroute(1M) 매뉴얼 페이지를 참조하십시오. traceroute 사용 절차는 87 페이지 “traceroute 명령으로 경로 지정 정보 표시”를 참조하십시오.

IPv6 관련 데몬

이 절에서는 IPv6 관련 데몬에 대해 설명합니다.

in.ndpd 데몬(Neighbor Discovery용)

in.ndpd 데몬은 IPv6 Neighbor Discovery 프로토콜 및 라우터 검색을 구현합니다. 이 데몬은 IPv6에 대한 주소 자동 구성도 구현합니다. 다음은 in.ndpd의 지원되는 옵션을 보여줍니다.

-a	stateless 및 stateful 주소 자동 구성을 해제합니다.
-d	디버깅을 설정합니다.
-f *config-file*	기본 /etc/inet/ndpd.conf 파일 대신 구성을 읽을 파일을 지정합니다.
-t	모든 송신 및 수신 패킷의 패킷 추적을 설정합니다.

in.ndpd 데몬은 /etc/inet/ndpd.conf 구성 파일에 설정된 매개변수와 /var/inet/ndpd_state. *interface* 시작 파일의 매개변수로 제어됩니다.

/etc/inet/ndpd.conf 파일이 있으면 이 파일이 구문 분석되어 노드를 라우터로 구성하는 데 사용됩니다. 표 8–1은 이 파일에 나타날 수 있는 유효 키워드를 보여줍니다. 호스트가 부트되는 즉시 라우터가 사용 가능하지 않을 수 있습니다. 라우터에 의해 알려진 패킷은 삭제될 수 있습니다. 또한 호스트에 연결되지 않을 수도 있습니다.

/var/inet/ndpd_state.*interface* 파일은 상태 파일입니다. 이 파일은 각 노드에서 정기적으로 업데이트됩니다. 노드가 실패하여 다시 시작되었을 때 라우터가 없는 경우 노드가 인터페이스를 구성할 수 있습니다. 이 파일에는 파일이 마지막으로 업데이트된

당시의 인터페이스 주소 및 파일 유효 기간이 포함되어 있습니다. 또한 이전 라우터 알림에서 "학습한" 기타 매개변수도 포함되어 있습니다.

주 – 상태 파일의 내용은 변경할 필요가 없습니다. `in.ndpd` 데몬이 자동으로 상태 파일을 유지 관리합니다.

구성 변수 및 허용되는 값 목록은 `in.ndpd(1M)` 매뉴얼 페이지 및 `ndpd.conf(4)` 매뉴얼 페이지를 참조하십시오.

in.ripngd 데몬(IPv6 경로 지정용)

`in.ripngd` 데몬은 IPv6 라우터에 대한 차세대 경로 지정 정보 프로토콜(RIPng)을 구현합니다. RIPng는 IPv6에 해당하는 RIP입니다. `routeadm` 명령으로 IPv6 라우터를 구성하고 IPv6 경로 지정을 설정하면 `in.ripngd` 데몬이 라우터에서 RIPng를 구현합니다.

다음은 RIPng의 지원되는 옵션을 보여줍니다.

`-p` *n* — *n*은 RIPng 패킷을 전송 또는 수신하는 데 사용되는 UDP 포트 번호를 지정합니다.

`-P` — Poison Reverse를 사용하지 못하도록 합니다.

`-q` — 경로 지정 정보를 표시하지 않습니다.

`-s` — 데몬이 라우터로 사용되지 않는 경우에도 경로 지정 정보를 표시합니다.

`-t` — 모든 전송 및 수신된 패킷을 표준 출력으로 출력합니다.

`-v` — 시간 기록을 포함하여 경로 지정 테이블에 대한 모든 변경 사항을 표준 출력으로 출력합니다.

inetd 데몬 및 IPv6 서비스

IPv6 지원 서버 응용 프로그램은 IPv4 요청과 IPv6 요청을 모두 처리하거나, IPv6 요청만 처리할 수 있습니다. 서버는 항상 IPv6 소켓을 통해 요청을 처리합니다. 또한 해당 클라이언트가 사용하는 것과 동일한 프로토콜을 사용합니다.

IPv6용 서비스를 추가하거나 수정하려면 SMF(서비스 관리 기능)에서 제공하는 명령을 사용하십시오.

- SMF 명령에 대한 자세한 내용은 **Oracle Solaris 11.1에서 서비스 및 결함 관리**의 "SMF 명령줄 관리 유틸리티"를 참조하십시오.
- SMF를 사용하여 SCTP를 통해 실행되는 IPv4 서비스 매니페스트를 구성하는 예제 작업은 57 페이지 "SCTP 프로토콜을 사용하는 서비스를 추가하는 방법"을 참조하십시오.

IPv6 서비스를 구성하려면 해당 서비스에 대한 inetadm 프로파일의 proto 필드 값에 적합한 값이 나열되어야 합니다.

- IPv4 및 IPv6 요청을 모두 처리하는 서비스의 경우 tcp6, udp6 또는 sctp를 선택합니다. tcp6, udp6 또는 sctp6의 값이 proto일 경우 inetd는 서버에 IPv6 소켓을 전달합니다. IPv4 클라이언트에 요청이 있는 경우 서버에는 IPv4 매핑 주소가 포함됩니다.
- IPv6 요청만 처리하는 서비스의 경우 tcp6only 또는 udp6only를 선택합니다. proto에 대해 이러한 값 중 하나를 사용할 경우, inetd는 서버에 IPv6 소켓을 전달합니다.

Oracle Solaris 명령을 다른 구현으로 바꿀 경우 해당 서비스 구현이 IPv6을 지원하는지 확인해야 합니다. 구현이 IPv6을 지원하지 않는 경우 proto 값을 tcp, udp 또는 sctp로 지정해야 합니다.

다음은 IPv4 및 IPv6을 둘 다 지원하고 SCTP를 통해 실행되는 echo 서비스 매니페스트에 대해 inetadm이 실행되도록 하는 프로파일입니다.

```
# inetadm -l svc:/network/echo:sctp_stream
    SCOPE    NAME=VALUE      name="echo"
             endpoint_type="stream"
             proto="sctp6"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
             user="root"
    default  bind_addr=""
    default  bind_fail_max=-1
    default  bind_fail_interval=-1
    default  max_con_rate=-1
    default  max_copies=-1
    default  con_rate_offline=-1
    default  failrate_cnt=40
    default  failrate_interval=60
    default  inherit_env=TRUE
    default  tcp_trace=FALSE
    default  tcp_wrappers=FALSE
```

proto 필드의 값을 변경하려면 다음 구문을 사용하십시오.

```
# inetadm -m FMRI proto="transport-protocols"
```

Oracle Solaris 소프트웨어에 제공되는 모든 서버에는 proto를 tcp6, udp6 또는 sctp6로 지정하는 프로파일 항목이 하나만 있으면 됩니다. 그러나 원격 셸 서버(shell) 및 원격 실행 서버(exec)는 이제 단일 서비스 인스턴스로 구성됩니다. 이 경우 proto 값에는 tcp 및 tcp6only 값이 포함됩니다. 예를 들어 shell의 proto 값을 설정하려면 다음 명령을 실행하십시오.

```
# inetadm -m network/shell:default proto="tcp,tcp6only"
```

소켓을 사용하는 IPv6 지원 서버를 작성하는 방법에 대한 자세한 내용은 **Programming Interfaces Guide** 의 IPv6 extensions to the Socket API를 참조하십시오.

IPv6용 서비스 구성 시 고려 사항

IPv6용 서비스를 추가하거나 수정할 경우 다음 사항에 유의하십시오.

- `proto` 값을 `tcp6`, `sctp6` 또는 `udp6`로 지정해야 IPv4 및 IPv6 연결이 모두 가능합니다. `proto` 값을 `tcp`, `sctp` 또는 `udp`로 지정한 경우 서비스는 IPv4만 사용합니다.
- `inetd`에 대해 일대다 스타일 SCTP 소켓을 사용하는 서비스 인스턴스를 추가할 수는 있지만, 이는 권장되지 않습니다. 일대다 스타일 SCTP 소켓에서는 `inetd`가 작동하지 않습니다.
- `wait-status` 또는 `exec` 등록 정보가 다르기 때문에 서비스에 두 개의 항목이 필요할 경우, 원래 서비스에서 두 개의 인스턴스/서비스를 만들어야 합니다.

IPv6 Neighbor Discovery 프로토콜

IPv6은 RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (`http://www.ietf.org/rfc/rfc2461.txt?number=2461`)에 설명된 Neighbor Discovery 프로토콜을 사용합니다.

이 절에서는 Neighbor Discovery 프로토콜의 다음 기능에 대해 설명합니다.

- 134 페이지 “Neighbor Discovery에서 제공하는 ICMP 메시지”
- 135 페이지 “자동 구성 프로세스”
- 137 페이지 “이웃 요청 및 연결 불가”
- 137 페이지 “중복 주소 감지 알고리즘”
- 138 페이지 “ARP 및 관련 IPv4 프로토콜과 Neighbor Discovery 비교”

Neighbor Discovery에서 제공하는 ICMP 메시지

Neighbor Discovery에서는 5개의 새 ICMP(Internet Control Message Protocol) 메시지를 정의합니다. 이 메시지의 목적은 다음과 같습니다.

- **라우터 요청** – 인터페이스가 사용으로 설정되면 호스트에서 라우터 요청 메시지를 보낼 수 있습니다. 유도는 라우터 알림을 다음 예정 시간에 생성하는 대신 즉시 생성하도록 라우터에 요청합니다.
- **라우터 알림** – 라우터는 자신의 존재, 다양한 링크 매개변수 및 다양한 인터넷 매개변수를 알립니다. 라우터 알림은 정기적으로 수행되거나 라우터 요청 메시지에 대한 응답으로 수행됩니다. 라우터 알림에는 온-링크 결정 또는 주소 구성에 사용되는 접두어, 제안되는 홉 한계 값 등이 포함됩니다.
- **이웃 요청** – 노드가 이웃의 링크 계층 주소를 확인하기 위해 이웃 요청 메시지를 전송합니다. 이웃 요청 메시지는 캐시된 링크 계층 주소를 통해 여전히 이웃에 연결할 수 있는지 확인할 목적으로도 전송됩니다. 이웃 요청은 중복 주소 감지에도 사용됩니다.

- **이웃 알림** – 노드가 이웃 요청 메시지에 대한 응답으로 이웃 알림 메시지를 전송합니다. 또한 링크 계층 주소 변경을 알리기 위해 요청되지 않은 이웃 알림도 전송합니다.
- **재지정** – 라우터는 재지정 메시지를 사용하여 보다 나은 대상의 첫번째 홉을 호스트에 알리거나 대상이 동일한 링크에 있음을 알립니다.

자동 구성 프로세스

이 절에서는 자동 구성 중 인터페이스에서 수행하는 일반적인 단계에 대해 간략히 설명합니다. 자동 구성은 멀티캐스트 가능 링크에서만 수행됩니다.

1. 예를 들어 멀티캐스트 가능 인터페이스는 노드의 시스템 시작 중에 사용으로 설정됩니다.
2. 노드는 인터페이스에 대한 링크 로컬 주소를 생성하여 자동 구성 프로세스를 시작합니다.

   링크 로컬 주소는 인터페이스의 MAC(매체 액세스 제어) 주소에서 생성됩니다.
3. 노드가 임시 링크 로컬 주소를 대상으로 포함하는 이웃 요청 메시지를 전송합니다.

   이 메시지의 목적은 예상 주소를 링크의 다른 노드에서 아직 사용하고 있지 않음을 확인하는 것입니다. 확인 후 링크 로컬 주소를 인터페이스에 지정할 수 있습니다.

   a. 다른 노드에서 이미 제안된 주소를 사용하고 있는 경우 주소가 이미 사용 중임을 나타내는 이웃 알림을 노드에서 반환합니다.

   b. 다른 노드에서도 동일한 주소를 사용하려고 하는 경우 해당 노드에서도 대상에 대한 이웃 요청을 전송합니다.

      이웃 요청 전송/재전송 횟수 및 연속 요청 간격은 링크별로 다릅니다. 필요한 경우 이러한 매개변수를 설정할 수 있습니다.
4. 노드에서 예상 링크 로컬 주소가 고유하지 않다고 판단될 경우 자동 구성이 중지됩니다. 이 경우 인터페이스의 링크 로컬 주소를 수동으로 구성해야 합니다.

   간단하게 복구하려면 기본 식별자를 대체하는 대체 인터페이스 ID를 제공하면 됩니다. 그러면 고유한 새 인터페이스 ID를 사용하여 자동 구성 방식이 다시 시작될 수 있습니다.
5. 노드에서 예상 링크 로컬 주소가 고유하다고 판단될 경우 노드가 주소를 인터페이스에 지정합니다.

   이 경우 노드가 이웃 노드와 IP 레벨로 연결됩니다. 나머지 자동 구성 단계는 호스트에 의해서만 수행됩니다.

라우터 알림 획득

자동 구성의 다음 단계는 라우터 알림을 확보하거나 라우터가 없음을 확인하는 것입니다. 라우터가 있을 경우 호스트에서 수행해야 하는 자동 구성의 유형을 지정하는 라우터 알림이 전송됩니다.

라우터는 라우터 알림을 정기적으로 전송합니다. 그러나 연속 알림 간격은 일반적으로 자동 구성을 수행하는 호스트의 대기 시간보다 깁니다. 알림을 신속하게 확보하기 위해 호스트는 하나 이상의 라우터 요청을 모든 라우터 멀티캐스트 그룹에 전송합니다.

접두어 구성 변수

라우터 알림에는 또한 Stateless 주소 자동 구성이 접두어를 생성하는 데 사용되는 정보를 포함하는 접두어 변수도 있습니다. 라우터 알림의 Stateless Address Autoconfiguration(Stateless 주소 자동 구성) 필드는 개별적으로 처리됩니다. 접두어 정보를 포함하는 한 옵션 필드 즉, Address Autoconfiguration(주소 자동 구성) 플래그는 옵션이 Stateless 자동 구성에도 적용되는지 여부를 나타냅니다. 이 옵션 필드가 적용되는 경우 추가 옵션 필드에 서브넷 접두어가 수명 값과 함께 포함됩니다. 이 값은 접두어로부터 생성된 주소가 선호 및 유효 주소로 유지되는 시간을 나타냅니다.

라우터에서는 라우터 알림을 정기적으로 생성하기 때문에 호스트는 계속 새로운 알림을 수신합니다. IPv6 지원 호스트는 각 알림에 포함된 정보를 처리합니다. 그런 다음 정보를 추가합니다. 호스트는 또한 이전 알림에서 수신된 정보를 새로 고칩니다.

주소 고유성

보안을 위해 모든 주소는 인터페이스에 지정되기 전에 고유한지 테스트해야 합니다. Stateless 자동 구성을 통해 생성되는 주소마다 상황이 다릅니다. 주소의 고유성은 주로 인터페이스 ID에서 구성되는 주소 부분에 의해 결정됩니다. 따라서 노드에서 이미 링크 로컬 주소의 고유성이 확인된 경우 추가 주소를 개별적으로 테스트할 필요가 없습니다. 주소는 동일한 인터페이스 ID에서 생성되어야 합니다. 반대로, 수동으로 확보된 모든 주소는 개별적으로 고유한지 테스트해야 합니다. 어떤 사이트의 시스템 관리자는 중복 주소 감지를 수행할 때 발생하는 오버헤드가 이점을 능가한다고 생각합니다. 이 사이트의 경우 인터페이스별 구성 플래그를 설정하여 중복 주소 감지 사용을 사용 안함으로 설정할 수 있습니다.

호스트가 라우터 알림을 기다리는 동안 링크 로컬 주소를 생성하고 고유성을 확인하면 자동 구성 프로세스를 신속하게 수행할 수 있습니다. 라우터는 라우터 요청에 대한 응답을 몇 초 동안 지연시킬 수 있습니다. 따라서 두 단계를 연속해서 수행할 경우 자동 구성을 완료하는 데 필요한 총 시간이 상당히 길어질 수 있습니다.

이웃 요청 및 연결 불가

Neighbor Discovery는 **이웃 요청** 메시지를 사용하여 둘 이상의 노드에 동일한 유니캐스트 주소가 지정되었는지 확인합니다. **이웃 연결 불가 감지**는 이웃 오류 또는 이웃에 대한 정방향 경로 오류를 찾아냅니다. 이 감지의 경우 이웃으로 전송된 패킷이 실제로 해당 이웃에 도달했다는 긍정적인 확인이 필요합니다. 이웃 연결 불가 감지는 또한 노드의 IP 계층에서 패킷이 올바르게 처리되고 있는지도 확인합니다.

이웃 연결 불가 감지는 상위 계층 프로토콜 및 이웃 요청 메시지라는 두 소스에서 보내는 확인을 사용합니다. 가능한 경우 상위 계층 프로토콜은 연결이 **진행 중**이라는 긍정적인 확인을 제공합니다. 예를 들어 새 TCP 긍정 응답이 수신될 경우 이전에 전송된 데이터가 올바르게 전달되었음이 확인됩니다.

노드가 상위 계층 프로토콜로부터 긍정적인 확인을 받지 못할 경우 유니캐스트 이웃 요청 메시지를 전송합니다. 이 메시지는 다음 홉에서 연결 가능성을 확인해 주는 이웃 알림을 요청합니다. 불필요한 네트워크 트래픽을 줄이려면 노드가 활발하게 패킷을 전송하는 이웃에게만 프로브 메시지를 전송해야 합니다.

중복 주소 감지 알고리즘

구성된 모든 주소가 특정 링크에서 고유한지 확인하기 위해 노드는 주소에 대해 **중복 주소 감지** 알고리즘을 실행합니다. 주소를 인터페이스에 지정하기 전에 노드에서 이 알고리즘을 실행해야 합니다. 중복 주소 감지 알고리즘은 모든 주소에 대해 수행됩니다.

이 절에 설명된 자동 구성 프로세스는 라우터가 아닌 호스트에만 적용됩니다. 호스트 자동 구성에는 라우터가 알리는 정보가 사용되므로 라우터를 다른 방식으로 구성해야 합니다. 그러나 라우터는 이 장에 설명된 방식을 사용하여 링크 로컬 주소를 생성합니다. 또한 라우터는 주소를 인터페이스에 지정하기 전에 모든 주소에 대한 중복 주소 감지 알고리즘을 성공적으로 전달합니다.

프록시 알림

대상 주소 대신 패킷을 수락하는 라우터는 비대체 이웃 알림을 발행할 수 있습니다. 라우터는 이웃 요청에 응답할 수 없는 대상 주소에 대한 패킷을 수락할 수 있습니다. 현재는 프록시 사용이 지정되어 있지 않습니다. 그러나 프록시 알림을 사용하면 오프 링크가 이동된 모바일 노드와 같은 경우를 잠재적으로 처리할 수 있습니다. 프록시 사용은 이 프로토콜을 구현하는 노드를 처리하는 일반적인 방식은 아닙니다.

인바운드 로드 균형 조정

복제된 인터페이스를 포함하는 노드의 경우 동일한 링크의 여러 네트워크 인터페이스에서 패킷 수신 로드에 대한 균형을 조정해야 합니다. 이러한 노드에서는 여러 개의 링크 로컬 주소가 동일한 인터페이스에 지정되어 있습니다. 예를 들어 한 개의 네트워크 드라이버가 여러 네트워크 인터페이스를 링크 로컬 주소가 여러 개인 하나의 논리적 인터페이스로 표시할 수 있습니다.

로드 균형 조정은 라우터가 소스 링크 로컬 주소를 라우터 알림 패킷에서 생략하는 방식으로 처리됩니다. 따라서 이웃은 이웃 요청 메시지를 사용하여 라우터의 링크 로컬 주소를 알아내야 합니다. 그러면 요청을 발행한 주체에 따라 달라지는 링크 로컬 주소가 반환된 이웃 알림 메시지에 포함될 수 있습니다.

링크 로컬 주소 변경

링크 로컬 주소가 변경되었음을 알고 있는 노드는 요청되지 않은 멀티캐스트 이웃 알림 패킷을 전송할 수 있습니다. 이 노드의 경우 멀티캐스트 패킷을 모든 노드에 전송하여 잘못된 캐시된 링크 로컬 주소를 업데이트할 수 있습니다. 요청되지 않은 알림은 성능 향상을 위한 목적으로만 전송됩니다. 이웃 연결 불가 감지 알고리즘은 지연이 다소 길어지더라도 모든 노드가 새로운 주소를 안정적으로 검색할 수 있도록 해줍니다.

ARP 및 관련 IPv4 프로토콜과 Neighbor Discovery 비교

IPv6 Neighbor Discovery 프로토콜의 기능은 IPv4 프로토콜의 ARP(Address Resolution Protocol), ICMP(Internet Control Message Protocol) 라우터 검색 및 ICMP 재지정을 결합한 것입니다. IPv4에는 이웃 연결 불가 감지에 대해 일반적으로 합의된 프로토콜이나 방식이 없습니다. 그러나 호스트 요구 사항에 사용 불능 게이트웨이 감지에 대한 알고리즘이 지정되어 있습니다. 사용 불능 게이트웨이 감지는 이웃 연결 불가 감지를 통해 해결되는 문제의 일부입니다.

다음은 Neighbor Discovery 프로토콜을 관련 IPv4 프로토콜 세트와 비교한 목록입니다.

- 라우터 검색은 기본 IPv6 프로토콜 세트의 일부입니다. IPv6 호스트의 경우 라우터를 찾기 위해 경로 지정 프로토콜을 `snoop`할 필요가 없습니다. IPv4의 경우 라우터를 찾기 위해 ARP, ICMP 라우터 검색 및 ICMP 재지정을 사용합니다.
- IPv6 라우터 알림은 링크 로컬 주소를 전달합니다. 라우터의 링크 로컬 주소를 분석하기 위해 추가 패킷 교환이 필요하지 않습니다.
- 라우터 알림은 링크에 대한 사이트 접두어를 전달합니다. IPv4의 경우와 마찬가지로, 넷마스크를 구성하기 위해 별도의 방식이 필요하지 않습니다.

- 라우터 알림을 통해 주소 자동 구성이 가능해집니다. IPv4에서는 자동 구성이 구현되지 않았습니다.
- Neighbor Discovery를 사용하면 IPv6 라우터가 링크에 사용할 호스트의 MTU를 알릴 수 있습니다. 따라서 잘 알려진 MTU가 없는 링크에 대해 동일한 MTU 값이 모든 노드에서 사용됩니다. 동일한 네트워크에 있는 IPv4 호스트는 다른 MTU를 사용할 수 있습니다.
- IPv4 브로드캐스트 주소와 달리, IPv6 주소 결정 멀티캐스트는 40억(2^32)개 이상의 멀티캐스트 주소에 분산되어 있으므로 대상이 아닌 노드에서 주소 결정 관련 인터럽트가 상당히 줄어듭니다. 또한 비IPv6 시스템의 경우 전혀 인터럽트가 발생하지 않습니다.
- IPv6 재지정에는 첫번째 새 홉의 링크 로컬 주소가 포함되어 있습니다. 재지정 수신 시 별도의 주소 결정이 필요하지 않습니다.
- 여러 사이트 접두어가 동일한 IPv6 네트워크와 연관될 수 있습니다. 기본적으로 호스트는 라우터 알림을 통해 모든 로컬 사이트 접두어를 알게 됩니다. 그러나 라우터 알림에서 일부 또는 전체 접두어를 생략하도록 라우터를 구성할 수 있습니다. 이 경우 호스트는 대상이 원격 네트워크에 있다고 가정합니다. 따라서 호스트는 트래픽을 라우터로 전송합니다. 그러면 라우터가 재지정을 적절하게 발행할 수 있습니다.
- IPv4와 달리, IPv6 재지정 메시지의 수신자는 새로운 다음 홉이 로컬 네트워크에 있다고 가정합니다. IPv4에서는 네트워크 마스크에 따라 로컬 네트워크에 있지 않은 다음 홉을 지정하는 재지정 메시지가 호스트에서 무시됩니다. IPv6 재지정 방식은 IPv4의 XRedirect 기능과 비슷합니다. 재지정 방식은 비브로드캐스트 및 공유 매체 링크에 유용합니다. 이러한 네트워크에서 노드는 로컬 링크 대상에 대한 모든 접두어를 검사하면 안됩니다.
- IPv6 이웃 연결 불가 감지는 라우터에서 오류가 발생할 경우 패킷 전달을 향상해 줍니다. 이 기능은 부분적으로 오류가 발생한 링크나 분할된 링크를 통한 패킷 전달을 향상해 줍니다. 또한 링크 로컬 주소가 변경된 노드를 통한 패킷 전달도 향상해 줍니다. 예를 들어 모바일 노드는 사용되지 않는 ARP 캐시 덕분에 연결을 유지한 상태로 로컬 네트워크에서 이동할 수 있습니다. IPv4에는 이웃 연결 불가 감지에 해당하는 방식이 없습니다.
- ARP와 달리, Neighbor Discovery는 이웃 연결 불가 감지를 통해 반 링크 오류를 감지합니다. Neighbor Discovery는 양방향 연결이 없을 경우 트래픽이 이웃에게 전송되지 못하도록 합니다.
- IPv6 호스트는 라우터를 고유하게 식별하는 링크 로컬 주소를 사용하여 라우터 연관을 유지할 수 있습니다. 라우터를 식별하는 기능은 라우터 알림 및 재지정 메시지에 필요합니다. 사이트에 새 전역 접두어가 사용될 경우 호스트에서 라우터 연관이 유지되어야 합니다. IPv4에는 라우터를 식별하는 해당 방식이 없습니다.
- 수신 시 Neighbor Discovery 메시지의 홉 한계는 255이기 때문에 프로토콜은 오프 링크 노드에서 발생하는 스푸핑 공격의 영향을 받지 않습니다. 반대로, IPv4 오프 링크 노드의 경우 ICMP 재지정 메시지를 전송할 수 있습니다. IPv4 오프 링크 노드의 경우 또한 라우터 알림 메시지도 전송할 수 있습니다.

- ICMP 계층에 주소 결정을 배치하면 Neighbor Discovery는 ARP보다 더 매체 독립적이 됩니다. 따라서 표준 IP 인증 및 보안 방식을 사용할 수 있습니다.

IPv6 경로 지정

ICIDR(Classless Inter-Domain Routing)에 의거하여 Pv6 경로 지정은 IPv4 경로 지정과 거의 동일합니다. 주소가 32비트 IPv4 주소 대신 128비트 IPv6 주소라는 점만 다릅니다. 매우 간단한 확장을 통해 IPv4의 모든 경로 지정 알고리즘(예: OSPF, RIP, IDRP, IS-IS)을 IPv6의 경로를 지정하는 데 사용할 수 있습니다.

IPv6에는 또한 강력한 새로운 경로 지정 기능을 지원하는 단순 경로 지정 확장도 포함되어 있습니다. 새로운 경로 지정 기능은 다음과 같습니다.

- 정책, 성능 및 비용 등을 기준으로 하는 공급자 선택
- 호스트 이동성, 현재 위치로 경로 지정
- 자동 주소 재지정, 새 주소로 경로 지정

새로운 경로 지정 기능은 IPv6 경로 지정 옵션을 사용하는 IPv6 주소의 순서를 만들어 이용할 수 있습니다. IPv6 소스는 경로 지정 옵션을 사용하여 패킷 대상으로 이동하는 중에 방문할 하나 이상의 중간 노드 또는 토폴로지 그룹을 나열할 수 있습니다. 이 기능은 IPv4의 느슨한 소스 및 레코드 경로 옵션과 매우 비슷합니다.

주소 순서를 일반 기능으로 만들려면 대부분의 경우 IPv6 호스트에서 호스트가 수신하는 패킷의 경로를 역순으로 설정해야 합니다. IPv6 인증 헤더를 사용하여 패킷이 성공적으로 인증되어야 합니다. 패킷에 주소 순서가 포함되어 있어야 패킷이 원래 전송자에게 반환됩니다. 이 기술은 IPv6 호스트 구현에서 소스 경로의 처리 및 전환이 강제로 지원되도록 합니다. 소스 경로의 처리 및 전환은 공급자가 새로운 IPv6 기능(예: 공급자 선택 및 확장 주소)을 구현하는 호스트와 작업할 수 있도록 하는 데 중요합니다.

라우터 알림

멀티캐스트 가능 링크 및 지점 간 링크에서 각 라우터는 라우터의 사용 가능함을 알리는 라우터 알림 패킷을 정기적으로 멀티캐스트 그룹에 전송합니다. 호스트는 모든 라우터로부터 라우터 알림을 수신하여 기본 라우터 목록을 작성합니다. 라우터는 몇 초 내에 호스트가 라우터의 존재를 알 수 있도록 자주 라우터 알림을 생성합니다. 그러나 라우터는 알림 부재를 통해 라우터 오류를 감지할 만큼 자주 알림을 전송하지 않습니다. 이웃 연결 불가를 확인하는 별도의 감지 알고리즘을 통해 오류를 감지할 수 있습니다.

라우터 알림 접두어

라우터 알림에는 호스트가 라우터와 동일한 링크(온 링크)에 있는지 확인하는 데 사용되는 서브넷 접두어 목록이 포함되어 있습니다. 접두어 목록은 자동 주소 구성에도

사용됩니다. 접두어와 연관된 플래그는 특정 접두어의 의도된 사용을 지정합니다. 호스트는 알림의 온 링크 접두어를 사용하여 패킷 대상이 온 링크인 시점 또는 라우터 외부에 있는 시점을 확인하는 데 사용되는 목록을 작성하고 유지 관리합니다. 대상이 알림의 온 링크 접두어에 의해 처리되지 않더라도 대상은 온 링크 상태일 수 있습니다. 이 경우 라우터가 재지정을 전송할 수 있습니다. 재지정은 대상이 이웃임을 발신자에게 알립니다.

라우터는 라우터 알림 및 접두어별 플래그를 사용하여 Stateless 주소 자동 구성을 수행하는 방법을 호스트에 알릴 수 있습니다.

라우터 알림 메시지

라우터 알림 메시지에는 호스트가 송신 패킷에 사용해야 하는 인터넷 매개변수(예: 홉 한계)가 포함되어 있습니다. 선택적으로 링크 매개변수(예: 링크 MTU)도 포함될 수 있습니다. 이 기능으로 중요한 매개변수를 중앙에서 관리할 수 있습니다. 매개변수는 라우터에 대해 설정될 수 있으며 연결된 모든 호스트에 자동으로 전파됩니다.

노드는 대상 노드에 해당 링크 계층 주소를 반환하도록 요청하는 이웃 요청을 멀티캐스트 그룹에 전송하는 방식으로 주소 결정을 수행합니다. 멀티캐스트 이웃 요청 메시지는 대상 주소의 요청된 노드 멀티캐스트 주소로 전송됩니다. 대상은 유니캐스트 이웃 알림 메시지에 링크 계층 주소를 반환합니다. 패킷의 단일 요청-응답 쌍만으로 개시자와 대상이 서로의 링크 계층 주소를 결정할 수 있습니다. 이웃 요청에는 개시자의 링크 계층 주소가 포함되어 있습니다.

Oracle Solaris 이름 서비스에 대한 IPv6 확장

이 절에서는 IPv6 구현으로 도입된 이름 지정 변경 사항에 대해 설명합니다. IPv6 주소는 Oracle Solaris 이름 지정 서비스, NIS, LDAP, DNS 및 파일에 저장할 수 있습니다. IPv6 RPC 전송을 통해 NIS를 사용하여 원하는 NIS 데이터를 검색할 수도 있습니다.

IPv6에 대한 DNS 확장

IPv6 관련 리소스 레코드인 AAAA 리소스 레코드는 RFC 1886 **IP 버전 6 지원을 위한 DNS 확장**에 지정되었습니다. 이 AAAA 레코드는 호스트 이름을 128비트 IPv6 주소에 매핑합니다. PTR 레코드는 여전히 IPv6에서 IP 주소를 호스트 이름에 매핑하는 데 사용됩니다. 128비트 주소의 32 x 4 비트 니블은 IPv6 주소에 대해 역순 처리됩니다. 각 니블은 해당 16진 ASCII 값으로 변환됩니다. 그런 다음 `ip6.int`가 추가됩니다.

이름 서비스 명령에 대한 변경 사항

IPv6을 지원하기 위해 기존 이름 서비스 명령을 사용하여 IPv6 주소를 조회할 수 있습니다. 예를 들어 `ypmatch` 명령은 새 NIS 맵에서 작동합니다. `nslookup` 명령은 DNS에서 새 AAAA 레코드를 조회할 수 있습니다.

NFS 및 RPC IPv6 지원

NFS 소프트웨어 및 원격 프로시저 호출(RPC) 소프트웨어는 일관된 방식으로 IPv6을 지원합니다. NFS 서비스와 관련된 기존 명령은 변경되지 않았습니다. 대부분의 RPC 응용 프로그램도 별다른 변경 없이 IPv6에서 실행될 수 있습니다. 전송 정보를 포함하는 일부 고급 RPC 응용 프로그램의 경우 업데이트가 필요할 수 있습니다.

IPv6 Over ATM 지원

Oracle Solaris는 IPv6 over ATM, 영구 가상 회선(PVC) 및 정적 전환 가상 회선(SVC)을 지원합니다.

색인

M

N

P

Q

R

S

T

U

V

게

경