Oracle® Solaris 관리: IP 서비스

부품 번호: E38894–03
2013년 3월

ORACLE

130701@25097

목차

머리말

Oracle Solaris 관리: IP 서비스를 시작합니다. 이 설명서는 Oracle Solaris 시스템 관리의 중요한 부분을 다루고 있는 14권으로 구성된 세트의 일부입니다. 이 설명서에서는 Oracle Solaris OS를 이미 설치했다고 가정합니다. 네트워크를 구성하거나 네트워크에 필요한 네트워킹 소프트웨어를 구성할 준비가 되어 있어야 합니다.

주 – 본 Oracle Solaris 릴리스는 프로세서 아키텍처의 SPARC 및 x86 제품군을 사용하는 시스템을 지원합니다. 지원되는 시스템은 **Oracle Solaris OS: 하드웨어 호환성 목록**을 참조하십시오. 이 설명서에서는 플랫폼 유형에 따른 구현 차이가 있는 경우 이에 대하여 설명합니다.

이 문서에서 사용되는 x86 관련 용어의 의미는 다음과 같습니다.

- x86은 64비트 및 32비트 x86 호환 제품을 아우르는 큰 제품군을 의미합니다.
- x64는 특히 64비트 x86 호환 CPU와 관련됩니다.
- "32비트 x86"은 x86 기반 시스템에 대한 특정 32비트 정보를 나타냅니다.

지원되는 시스템은 **Oracle Solaris OS: 하드웨어 호환성 목록**을 참조하십시오.

이 설명서의 대상

이 책은 네트워크에 구성된 Oracle Solaris 실행 시스템의 관리 책임자를 대상으로 작성되었습니다. 본 설명서를 사용하려면 적어도 2년의 UNIX 시스템 관리 경력이 있어야 합니다. UNIX 시스템 관리 교육 과정에 참석하는 것도 도움이 될 수 있습니다.

시스템 관리 설명서의 구성

시스템 관리 설명서에서 설명하는 항목 목록은 다음과 같습니다.

설명서 제목	내용
Oracle Solaris 관리: 기본 관리	Oracle Solaris 명령 사용, 시스템 부트 및 종료, 서버 및 클라이언트 지원, 사용자 계정 및 그룹 관리, 서비스, 시스템 정보, 시스템 리소스 및 시스템 성능 관리, 소프트웨어, 콘솔 및 터미널 관리, 시스템 및 소프트웨어 문제 해결

설명서 제목	내용
시스템 관리 설명서: 고급 관리	터미널 및 모뎀, 시스템 리소스(디스크 쿼터, 계정, 크론탭), 시스템 프로세스, Oracle Solaris 소프트웨어 문제 해결
System Administration Guide: Devices and File Systems	이동식 매체, 디스크 및 장치, 파일 시스템, 데이터 백업 및 복원
Oracle Solaris 관리: IP 서비스	TCP/IP 네트워크 관리, IPv4 및 IPv6 주소 관리, DHCP, IPsec, IKE, IP 필터, IPMP(IP Network Multipathing), IPQoS
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS 및 LDAP 명명 규칙 및 디렉토리 서비스(NIS에서 LDAP으로의 전환, NIS+에서 LDAP으로의 전환 포함)
System Administration Guide: Naming and Directory Services (NIS+)	NIS+ 이름 지정 및 디렉토리 서비스
System Administration Guide: Network Services	웹 캐시 서버, 시간 관련 서비스, 네트워크 파일 시스템(NFS 및 Autofs), 메일, SLP, PPP
시스템 관리 설명서: Oracle Solaris Containers-리소스 관리 및 Oracle Solaris 영역	리소스 관리 항목 프로젝트 및 작업, 확장 계정, 리소스 제어, FSS(Fair Share Scheduler), rcapd(Resource Capping Daemon)를 통한 물리적 메모리 제어, 리소스 풀, Solaris Zones 소프트웨어 분할 기술 및 lx 브랜드 영역을 통한 가상화
System Administration Guide: Printing	인쇄 항목 및 작업, 서비스, 도구, 프로토콜 및 기술을 사용하여 인쇄 서비스와 프린터 설정 및 관리
System Administration Guide: Security Services	감사, 장치 관리, 파일 보안, BART, Kerberos 서비스, PAM, 암호화 프레임워크, 키 관리, 권한, RBAC, SASL 및 보안 셸
Oracle Solaris ZFS 관리 설명서	ZFS 저장소 풀 및 파일 시스템 만들기/관리, 스냅샷, 복제, 백업, ACL(액세스 제어 목록)을 통한 ZFS 파일 보호, 영역이 설치된 Solaris 시스템에서 ZFS 사용, 에뮬레이트된 볼륨, 문제 해결 및 데이터 복구
Trusted Extensions 관리자 절차	Trusted Extensions 관련 시스템 관리
Oracle Solaris Trusted Extensions 구성 설명서	Solaris 10 5/08 릴리스부터 Trusted Extensions를 계획, 사용으로 설정 및 처음 구성하는 방법에 대해 설명합니다.

관련 설명서

이 설명서에서는 다음과 같은 책이 참조됩니다.

- Stevens, W. Richard. **TCP/IP Illustrated, Volume 1, The Protocols**. Addison Wesley, 1994.
- Hunt Craig.**TCP/IP Network Administration, 3rd Edition**. O'Reilly, 2002.
- Perkins, Charles E. **Mobile IP Design Principles and Practices**. Massachusetts, 1998, Addison-Wesley Publishing Company.

- Solomon, James D. **Mobile IP: The Internet Unplugged**. New Jersey, 1998, Prentice-Hall, Inc.
- Ferguson, Paul 및 Geoff Huston. **Quality of Service**. John Wiley & Sons, Inc., 1998.
- Kilkki, Kalevi. **Differentiated Services for the Internet**. Macmillan Technical Publishing, 1999.

타사 웹 사이트

이 문서에는 타사 URL이 언급되어 있으며 추가 관련 정보를 제공합니다.

Oracle Solaris의 IP 필터 기능은 오픈 소스 IP 필터 소프트웨어에서 파생되었습니다. IP 필터에 대한 라이센스 약관, 직권 및 저작권 설명을 볼 수 있는 기본 경로는 `/usr/lib/ipf/IPFILTER.LICENCE`입니다. Oracle Solaris 운영 체제가 기본 경로 이외의 다른 경로에 설치된 경우 설치된 위치의 파일에 액세스할 수 있도록 지정된 경로를 수정하십시오.

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 책에서 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.`
`AaBbCc123`	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	`machine_name%` **`su`** `Password:`
aabbcc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	`rm` *filename* 명령을 사용하여 파일을 제거합니다.

표 P–1 활자체 규약 *(계속)*

활자체	설명	예
AaBbCc123	책 제목, 장, 절	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하면 **안됩니다**. **주:** 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 UNIX 시스템 프롬프트 및 수퍼 유저 프롬프트가 나와 있습니다. 명령 예에서 셸 프롬프트는 명령을 일반 사용자가 실행해야 하는지 아니면 권한이 있는 사용자가 실행해야 하는지를 나타냅니다.

표 P–2 셸 프롬프트

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

제 1 부

시스템 관리 소개: IP 서비스

이 파트에는 TCP/IP 프로토콜 제품군 및 Oracle Solaris의 해당 구현에 대한 소개 정보가 들어 있습니다.

◆ ◆ ◆ 1 장

1 Oracle Solaris TCP/IP 프로토콜 제품군(개요)

이 장에서는 TCP/IP 네트워크 프로토콜 제품군의 Oracle Solaris 구현을 소개합니다. 정보는 기본 TCP/IP 개념에 익숙하지 않는 시스템 및 네트워크 관리자를 대상으로 작성되었습니다. 본 설명서의 나머지 부분에서는 사용자가 이러한 개념에 익숙하다고 가정합니다.

이 장은 다음 정보를 포함합니다.

이 릴리스의 새로운 기능

Solaris 10 5/08부터는 모바일 IP 기능이 제거되었습니다. 모바일 IP는 Solaris 10 OS 8/07 및 이전 릴리스에서 사용할 수 있습니다.

TCP/IP 프로토콜 제품군 소개

이 절에서는 TCP/IP에 포함된 프로토콜에 대해 자세히 소개합니다. 정보가 개념적이긴 해도 프로토콜의 이름을 알아야 합니다. 또한 각 프로토콜이 수행하는 작업도 알아야 합니다.

"TCP/IP"는 **인터넷 프로토콜 제품군**을 구성하는 네트워크 프로토콜 세트에 공통으로 사용되는 머리 글자어입니다. 대부분의 텍스트는 "인터넷"이라는 용어를 사용하여 프로토콜 제품군과 전역 WAN(Wide Area Network)을 모두 기술합니다. 본 설명서에서 "TCP/IP"는 특별히 인터넷 프로토콜 제품군을 가리키며, "인터넷"은 WAN(Wide Area Network) 및 인터넷 관리 주체를 가리킵니다.

TCP/IP 네트워크를 다른 네트워크와 상호 연결하려면 네트워크의 고유한 IP 주소를 가져와야 합니다. 본 설명서 작성 시 인터넷 서비스 제공업체(ISP)로부터 이 주소를 가져옵니다.

네트워크의 호스트를 인터넷 DNS(Domain Name System)에 참가시키려면 고유한 도메인 이름을 가져와 등록해야 합니다. InterNIC는 전세계적인 레지스트리 그룹을 통해 도메인 이름 등록을 조정합니다. DNS에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

프로토콜 계층 및 Open Systems Interconnection 모델

일련의 계층으로 구조화된 대부분의 네트워크 프로토콜 제품군을 총칭하여 **프로토콜 스택**이라고 합니다. 각 계층은 특정 용도에 사용되도록 설계되었습니다. 송신 시스템과 수신 시스템 모두에 각 계층이 있습니다. 한 시스템의 특정 계층은 다른 시스템의 **피어 프로세스**가 보내거나 받는 객체를 그대로 보내거나 받습니다. 이러한 작업은 고려 중인 계층 위 또는 아래 계층의 작업과 별도로 발생합니다. 본질적으로 시스템의 각 계층은 동일한 시스템의 다른 계층과 별도로 작동합니다. 각 계층은 다른 시스템의 동일한 계층과 병렬로 작동합니다.

OSI 참조 모델

대부분의 네트워크 프로토콜 제품군은 계층으로 구조화되어 있습니다. ISO(국제 표준화 기구)에서 구조화된 계층을 사용하는 OSI(Open Systems Interconnection) 참조 모델을 설계했습니다. OSI 모델은 네트워크 작업에 대해 7개의 계층이 있는 구조를 기술합니다. 하나 이상의 프로토콜이 각 계층과 연결됩니다. 계층은 동시 작업 네트워크 간 모든 유형의 데이터 전송에서 공통적인 데이터 전송 작업을 나타냅니다.

OSI 모델은 최상위(계층 7)에서 최하위(계층 1)까지의 프로토콜 계층을 나열합니다. 다음 표에서는 모델을 보여 줍니다.

표 1–1 Open Systems Interconnection 참조 모델

계층 번호	계층 이름	설명
7	응용 프로그램	모든 사람이 사용할 수 있는 표준 통신 서비스 및 응용 프로그램으로 구성됩니다.
6	표현	정보가 시스템에서 이해할 수 있는 형식으로 수신 시스템에 전달되도록 합니다.
5	세션	동시 작업 시스템 간의 연결 및 종료를 관리합니다.
4	전송	데이터 전송을 관리합니다. 또한 수신된 데이터가 전송된 데이터와 일치하도록 보장합니다.
3	네트워크	네트워크 간의 데이터 주소 지정 및 배달을 관리합니다.
2	데이터 링크	네트워크 매체 간 데이터 전송을 처리합니다.
1	물리	네트워크 하드웨어의 특성을 정의합니다.

OSI 모델은 특정 네트워크 프로토콜 제품군에 대해 고유하지 않은 개념적 작업을 정의합니다. 예를 들어, OSI 네트워크 프로토콜 제품군은 OSI 모델의 7개 계층을 모두 구현합니다. TCP/IP는 일부 OSI 모델 계층을 사용합니다. 또한 TCP/IP는 다른 계층을 결합합니다. SNA와 같은 다른 네트워크 프로토콜은 8번째 계층을 추가합니다.

TCP/IP 프로토콜 아키텍처 모델

OSI 모델은 프로토콜 제품군을 포함하는 이상적인 네트워크 통신을 기술합니다. TCP/IP는 직접적으로 이 모델과 일치하지 않습니다. TCP/IP는 여러 OSI 계층을 단일 계층으로 결합하거나 특정 계층을 사용하지 않습니다. 다음 표에서는 TCP/IP의 Oracle Solaris 구현 계층을 보여 줍니다. 이 표는 최상위 계층(응용 프로그램)에서 최하위 계층(물리적 네트워크)까지의 계층을 보여 줍니다.

표 1–2 TCP/IP 프로토콜 스택

OSI 참조 계층 번호	해당 OSI 계층	TCP/IP 계층	TCP/IP 프로토콜 예
5,6,7	응용 프로그램, 세션, 표현	응용 프로그램	NFS, NIS, DNS, LDAP, `telnet`, `ftp`, `rlogin`, `rsh`, `rcp`, RIP, RDISC, SNMP 등
4	전송	전송	TCP, UDP, SCTP
3	네트워크	인터넷	IPv4, IPv6, ARP, ICMP
2	데이터 링크	데이터 링크	PPP, IEEE 802.2
1	물리적	물리적 네트워크	이더넷(IEEE 802.3), 토큰 링, RS-232, FDDI 등

이 표는 TCP/IP 프로토콜 계층 및 해당 OSI 모델을 보여 줍니다. 또한 TCP/IP 프로토콜 스택의 각 레벨에서 사용할 수 있는 프로토콜의 예를 보여 줍니다. 통신 트랜잭션에 관련된 각 시스템은 프로토콜 스택의 고유한 구현을 실행합니다.

물리적 네트워크 계층

물리적 네트워크 계층은 네트워크에 사용할 하드웨어의 특성을 지정합니다. 예를 들어, 물리적 네트워크 계층은 통신 매체의 물리적 특성을 지정합니다. TCP/IP의 물리적 계층은 이더넷 네트워크 매체에 대한 사양인 IEEE 802.3 및 표준 핀 커넥터에 대한 사양인 RS-232와 같은 하드웨어 표준을 기술합니다.

데이터 링크 계층

데이터 링크 계층은 이 인스턴스 TCP/IP에서 패킷의 네트워크 프로토콜 유형을 식별합니다. 또한 데이터 링크 계층은 오류 제어 및 "프레이밍"을 제공합니다. 데이터 링크 계층 프로토콜의 예로는 이더넷 IEEE 802.2 프레이밍 및 PPP(지점 간 프로토콜) 프레이밍이 있습니다.

인터넷 계층

네트워크 계층 또는 *IP* **계층**이라고도 하는 인터넷 계층은 네트워크에 대한 패킷을 수락하고 전달합니다. 이 계층에는 강력한 IP(Internet Protocol), ARP(Address Resolution Protocol) 및 ICMP(Internet Control Message Protocol)가 포함됩니다.

IP 프로토콜

IP 프로토콜 및 연결된 경로 지정 프로토콜은 전체 TCP/IP 제품군에서 가장 중요합니다. IP는 다음 기능을 담당합니다.

- **IP 주소 지정** – IP 주소 지정 규약은 IP 프로토콜의 일부입니다. 54 페이지 "IPv4 주소 지정 체계 설계"에서는 IPv4 주소 지정을 소개하고 70 페이지 "IPv6 주소 지정 개요"에서는 IPv6 주소 지정을 소개합니다.
- **호스트 간 통신** – IP는 수신 시스템의 IP 주소를 기반으로 패킷이 가져와야 하는 경로를 결정합니다.
- **패킷 형식 지정** – IP는 패킷을 **데이터그램**이라고 하는 단위로 어셈블합니다. 데이터그램은 42 페이지 "인터넷 계층: 패킷 배달이 준비되는 위치"에 자세히 설명되어 있습니다.
- **단편화** – 네트워크 매체에서 전송 시 패킷이 너무 큰 경우 송신 시스템의 IP가 패킷을 작은 단편으로 나눕니다. 그런 다음 수신 시스템의 IP는 단편을 원래 패킷으로 재구성합니다.

Oracle Solaris는 본 설명서에서 설명하는 IPv4 및 IPv6 주소 지정 형식을 모두 지원합니다. 인터넷 프로토콜 주소 지정 시 혼동되지 않도록 다음 규약 중 하나가 사용됩니다.

- 설명에서 용어 "IP"가 사용되면 해당 설명은 IPv4와 IPv6 모두에 적용됩니다.
- 설명에서 용어 "IPv4"가 사용되면 해당 설명은 IPv4에만 적용됩니다.
- 설명에서 용어 "IPv6"이 사용되면 해당 설명은 IPv6에만 적용됩니다.

ARP 프로토콜

ARP(Address Resolution Protocol)는 개념적으로 데이터 링크와 인터넷 계층 간에 존재합니다. ARP는 이더넷 주소(48비트 길이)를 알려진 IP 주소(32비트 길이)에 매핑하여 데이터그램의 방향을 해당 수신 시스템으로 지정하는 IP 기능을 지원합니다.

ICMP 프로토콜

ICMP(Internet Control Message Protocol)는 네트워크 오류 상태를 감지하고 보고합니다. ICMP는 다음에 대해 보고합니다.

- **삭제된 패킷** – 너무 빨리 도착하여 처리될 수 없는 패킷입니다.
- **연결 오류** – 대상 시스템에 도달할 수 없습니다.
- **재지정** – 송신 시스템에서 다른 라우터를 사용하도록 재지정합니다.

8 장, "TCP/IP 네트워크 관리(작업)"에는 오류 감지를 위해 ICMP를 사용하는 Oracle Solaris 명령에 대한 자세한 내용이 포함되어 있습니다.

전송 계층

TCP/IP **전송 계층**은 데이터 수신의 긍정 응답을 교체하고 손실된 패킷을 재전송하여 순서대로 오류 없이 패킷이 도착하도록 합니다. 이 유형의 통신을 **종단간**이라고 합니다. 이 레벨에서의 전송 계층 프로토콜은 TCP(전송 제어 프로토콜), UDP(사용자 데이터그램 프로토콜) 및 SCTP(흐름제어 전송 프로토콜)입니다. TCP 및 SCTP는 안정적인 종단간 서비스를 제공합니다. UDP는 불안정한 데이터그램 서비스를 제공합니다.

TCP 프로토콜

TCP를 사용으로 설정하면 응용 프로그램이 물리적 회로로 연결된 것처럼 서로 통신할 수 있습니다. TCP는 별개의 패킷이 아닌 문자별 방식으로 전송되는 것처럼 보이는 형식으로 데이터를 전송합니다. 이 전송은 다음 요소로 구성됩니다.

- 연결을 시작하는 시작 지점
- 바이트 순서의 전체 전송
- 연결을 닫는 종료 지점

TCP는 헤더를 전송된 데이터에 연결합니다. 이 헤더에는 송신 시스템의 프로세스를 수신 시스템의 피어 프로세스에 연결하는 데 유용한 많은 매개변수가 포함되어 있습니다.

TCP는 송신 호스트와 수신 호스트 간에 종단간 연결을 설정하여 해당 대상에 패킷이 도달했는지 확인합니다. 따라서 TCP는 "안정적인 연결 지향" 프로토콜로 간주됩니다.

SCTP 프로토콜

SCTP는 TCP에서 사용 가능한 응용 프로그램에 동일한 서비스를 제공하는 안정적인 연결 지향 전송 계층 프로토콜입니다. 또한 SCTP는 주소가 둘 이상이거나 **멀티홈**인 시스템 간의 연결을 지원할 수 있습니다. 송신 시스템과 수신 시스템 간의 SCTP 연결을 **연관**이라고 합니다. 연관의 데이터는 청크로 구성됩니다. SCTP는 멀티홈을 지원하므로 전자 통신 업계에서 사용되는 특정 응용 프로그램은 TCP가 아닌 SCTP에서 실행되어야 합니다.

UDP 프로토콜

UDP는 데이터그램 배달 서비스를 제공합니다. UDP는 수신 호스트와 송신 호스트 간의 연결을 확인하지 않습니다. UDP에는 연결 설정 및 확인 프로세스가 없으므로 적은 양의 데이터를 전송하는 응용 프로그램에서는 UDP를 사용합니다.

응용 프로그램 계층

응용 프로그램 계층은 누구나 사용할 수 있는 표준 인터넷 서비스 및 네트워크 응용 프로그램을 정의합니다. 이러한 서비스는 데이터를 전송 및 수신하는 전송 계층과 함께 사용됩니다. 많은 응용 프로그램 계층 프로토콜이 있습니다.

다음 목록에서는 응용 프로그램 계층 프로토콜의 예를 보여 줍니다.

- 표준 TCP/IP 서비스(예: `ftp`, `tftp` 및 `telnet` 명령)
- UNIX "r" 명령(예: `rlogin` 및 `rsh`)
- 이름 서비스(예: NIS 및 DNS(Domain Name System))
- 디렉토리 서비스(LDAP)
- 파일 서비스(예: NFS 서비스)
- 네트워크 관리를 사용으로 설정하는 SNMP(Simple Network Management Protocol)
- RDISC(Router Discovery Server Protocol) 및 RIP(Routing Information Protocol) 경로 지정 프로토콜

표준 TCP/IP 서비스

- **FTP 및 익명 FTP** – FTP(File Transfer Protocol)는 원격 네트워크 간에 파일을 전송합니다. 프로토콜에는 `ftp` 명령 및 `in.ftpd` 데몬이 포함됩니다. FTP를 사용으로 설정하면 사용자가 로컬 호스트의 명령줄에서 원격 호스트 및 파일 전송 명령 옵션의 이름을 지정할 수 있습니다. 그러면 원격 호스트의 `in.ftpd` 데몬이 로컬 호스트의 요청을 처리합니다. `rcp`와 달리 `ftp`는 원격 컴퓨터에서 UNIX 기반 운영 체제가 실행되지 않아도 작동됩니다. 원격 시스템에 익명 FTP 허용이 구성되지 않은 경우 사용자는 원격 시스템에 로그인하여 `ftp` 연결을 설정해야 합니다.

  인터넷에 연결된 **익명** *FTP* **서버**에서 많은 양의 자료를 구할 수 있습니다. 대학 및 기타 기관들이 이러한 서버를 설정하여 공용 도메인을 통해 소프트웨어, 연구 자료 및 기타 정보를 제공합니다. 이 유형의 서버에 로그인할 때 로그인 이름 `anonymous`를 사용하며, 이를 "익명 FTP 서버"라고 합니다.

  익명 FTP 사용 및 익명 FTP 서버 설정은 본 설명서에서 다루지 않습니다. 그러나 *The Whole Internet User's Guide & Catalog*와 같은 여러 설명서에서는 익명 FTP에 대해 자세히 설명합니다. FTP 사용에 대한 내용은 **System Administration Guide: Network Services** 에 나와 있습니다. `ftp(1)` 매뉴얼 페이지에서는 명령 인터프리터를 통해 호출되는 모든 `ftp` 명령 옵션을 설명합니다. `ftpd(1M)` 매뉴얼 페이지에서는 `in.ftpd` 데몬에서 제공하는 서비스를 설명합니다.

- **Telnet** – Telnet 프로토콜을 사용으로 설정하면 터미널 및 터미널 지향 프로세스가 TCP/IP를 실행하는 네트워크에서 통신할 수 있습니다. 이 프로토콜은 로컬 시스템인 경우 `telnet` 프로그램으로 구현되며 원격 시스템인 경우 `in.telnetd` 데몬으로 구현됩니다. Telnet에서는 두 호스트가 문자별 또는 라인별 단위로 통신할 수 있도록 사용자 인터페이스를 제공합니다. Telnet에는 `telnet(1)` 매뉴얼 페이지에서 완전하게 문서화된 명령 세트가 포함됩니다.

- **TFTP** – Trivial File Transfer Protocol(`tftp`)은 `ftp`와 유사한 기능을 제공하지만 이 프로토콜은 `ftp`의 대화식 연결을 설정하지 않습니다. 따라서 사용자는 디렉토리의 컨텐츠를 나열하거나 디렉토리를 변경할 수 없습니다. 사용자는 복사할 파일의 전체 이름을 알고 있어야 합니다. `tftp(1)` 매뉴얼 페이지에서는 `tftp` 명령 세트를 설명합니다.

UNIX "r" 명령

UNIX "r" 명령을 사용으로 설정하면 사용자는 원격 호스트에서 실행하는 해당 로컬 시스템에서 명령을 실행할 수 있습니다.

이러한 명령은 다음과 같습니다.

- `rcp`
- `rlogin`
- `rsh`

이러한 명령을 사용하는 방법에 대한 지침은 `rcp(1)`, `rlogin(1)` 및 `rsh(1)` 매뉴얼 페이지에 나와 있습니다.

이름 서비스

Oracle Solaris는 다음과 같은 이름 서비스를 제공합니다.

- **DNS** – DNS(Domain Name System)는 TCP/IP 네트워크의 인터넷에서 제공하는 이름 서비스입니다. DNS는 IP 주소 서비스에 호스트 이름을 제공합니다. 또한 DNS는 메일 관리를 위한 데이터베이스 역할을 수행합니다. 이 서비스에 대한 자세한 설명은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오. `resolver(3RESOLV)` 매뉴얼 페이지를 참조하십시오.
- **`/etc` 파일** – 원본 호스트 기반 UNIX 이름 시스템은 독립형 UNIX 시스템으로 개발된 다음 네트워크 사용에 맞게 조정되었습니다. 기존의 여러 UNIX 운영 체제 및 컴퓨터는 계속 이 시스템을 사용하고 있지만 복잡한 대규모 네트워크에 적합하지 않습니다.
- **NIS** – NIS(네트워크 정보 서비스)는 DNS와 독립적으로 개발되었으며 주력하는 부분은 약간 다릅니다. DNS는 숫자 IP 주소 대신 시스템 이름을 사용하여 통신을 더 간소화하는 데 주력하고, NIS는 다양한 네트워크 정보에 대한 중앙집중 제어를 제공하여 네트워크 관리를 더 용이하게 하는 데 주력합니다. NIS는 시스템 이름 및 주소, 사용자, 네트워크 자체 및 네트워크 서비스에 대한 정보를 저장합니다. NIS 이름 공간 정보는 NIS 맵에 저장됩니다. NIS 아키텍처 및 NIS 관리에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

디렉토리 서비스

Oracle Solaris는 Sun ONE(Sun Open Net Environment) 디렉토리 서버 및 기타 LDAP 디렉토리 서버와 함께 LDAP(Lightweight Directory Access Protocol)를 지원합니다. 기능 확장 영역의 차이를 통해 이름 서비스와 디렉토리 서비스를 구분합니다. 디렉토리 서비스는 이름 지정 서비스와 동일한 기능을 제공하지만 추가 기능도 제공합니다. **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

파일 서비스

NFS 응용 프로그램 계층 프로토콜은 Oracle Solaris에 대한 파일 서비스를 제공합니다. **System Administration Guide: Network Services** 에서 NFS 서비스에 대한 자세한 내용을 찾을 수 있습니다.

네트워크 관리

SNMP(Simple Network Management Protocol)를 사용으로 설정하면 네트워크의 레이아웃 및 핵심 시스템의 상태를 볼 수 있습니다. 또한 SNMP를 사용으로 설정하면 GUI(그래픽 사용자 인터페이스)에 따라 복잡한 네트워크 통계를 구할 수 있습니다. 많은 회사에서 SNMP를 구현하는 네트워크 관리 패키지를 제공합니다.

경로 지정 프로토콜

RIP(Routing Information Protocol) 및 RDISC(Router Discovery Server Protocol)는 TCP/IP 네트워크에서 사용 가능한 두 가지 경로 지정 프로토콜입니다. Oracle Solaris에서 사용 가능한 전체 경로 지정 프로토콜 목록은 표 5–1 및 표 5–2를 참조하십시오.

TCP/IP 프로토콜에서 데이터 통신을 처리하는 방법

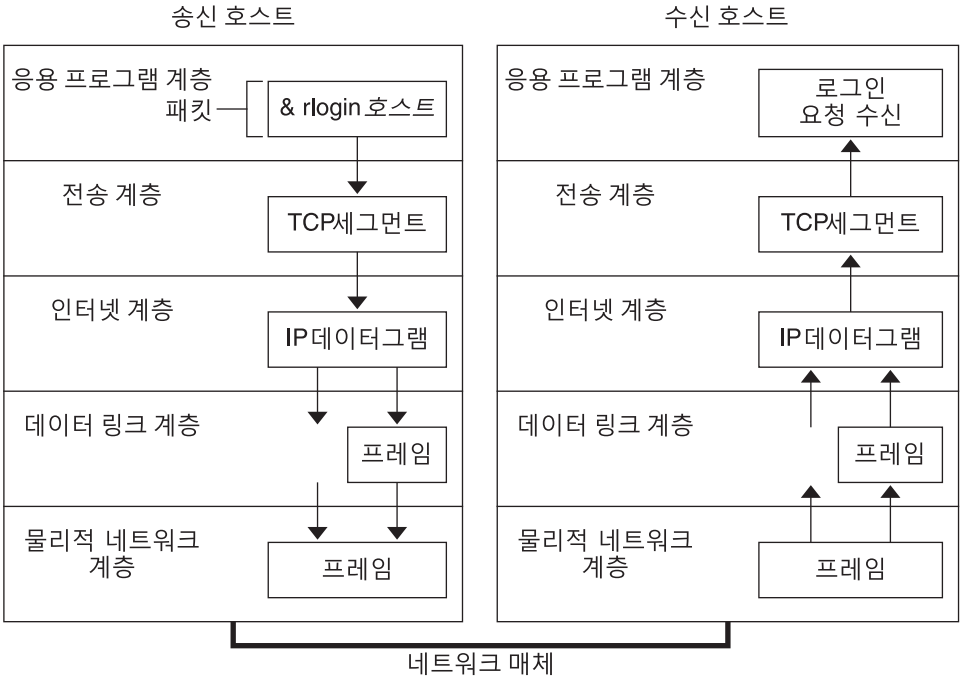
TCP/IP 응용 프로그램 계층 프로토콜을 사용하는 명령을 사용자가 실행하면 일련의 이벤트가 시작됩니다. 사용자의 명령 또는 메시지는 로컬 시스템의 TCP/IP 프로토콜 스택을 통해 전달됩니다. 그런 다음 명령 또는 메시지가 네트워크 매체를 통해 원격 시스템의 프로토콜로 전달됩니다. 송신 호스트의 각 계층에 있는 프로토콜이 원래 데이터에 정보를 추가합니다.

또한 송신 호스트의 각 계층에 있는 프로토콜은 수신 호스트에 있는 해당 피어와 상호 작용합니다. 그림 1–1에서는 이러한 상호 작용을 보여 줍니다.

데이터 캡슐화 및 TCP/IP 프로토콜 스택

패킷은 네트워크를 통해 전송되는 기본 정보 단위입니다. 기본 패킷은 송신 및 수신 시스템의 주소가 있는 헤더, 본문 또는 전송될 데이터가 있는 **페이로드**로 구성됩니다. 패킷이 TCP/IP 프로토콜 스택을 통해 이동함에 따라 각 계층의 프로토콜은 기본 헤더에서 필드를 추가하거나 제거합니다. 송신 시스템의 프로토콜이 패킷 헤더에 데이터를 추가하면 이 프로세스를 **데이터 캡슐화**라고 합니다. 또한 각 계층은 다음 그림과 같이 변경된 패킷에 대해 서로 다른 용어를 사용합니다.

그림 1–1 패킷이 TCP/IP 스택을 통해 이동하는 방법



이 절에는 패킷의 수명 주기가 요약되어 있습니다. 사용자가 명령을 실행하거나 메시지를 보낼 때 수명 주기가 시작됩니다. 수신 시스템의 해당 응용 프로그램이 패킷을 수신할 때 수명 주기가 끝납니다.

응용 프로그램 계층: 통신이 시작되는 위치

한 시스템의 사용자가 메시지를 보내거나 원격 시스템에 액세스해야 하는 명령을 실행할 때 패킷의 기록이 시작됩니다. 적합한 전송 계층 프로토콜인 TCP 또는 UDP가 패킷을 처리할 수 있도록 응용 프로그램 프로토콜이 패킷의 형식을 지정합니다.

그림 1–1과 같이 사용자가 `rlogin` 명령을 실행하여 원격 시스템에 로그인한다고 가정합니다. `rlogin` 명령은 TCP 전송 계층 프로토콜을 사용합니다. TCP는 명령의 정보를 포함하는 바이트 스트림 형식의 데이터를 수신합니다. 따라서 `rlogin`은 이 데이터를 TCP 스트림으로 보냅니다.

전송 계층: 데이터 캡슐화가 시작되는 위치

데이터가 전송 계층에 도착하면 해당 계층의 프로토콜이 데이터 캡슐화 프로세스를 시작합니다. 전송 계층은 응용 프로그램 데이터를 전송 프로토콜 데이터 단위로 캡슐화합니다.

전송 계층 프로토콜은 전송 포트 번호로 구분되는 송신 응용 프로그램과 수신 응용 프로그램 간의 가상 데이터 플로우를 만듭니다. 포트 번호는 데이터 수신 또는 송신을 위한 메모리의 전용 위치인 **포트**를 식별합니다. 또한 전송 프로토콜 계층은 안정적인 다른 서비스를 데이터 배달 순서대로 제공할 수 있습니다. 마지막 결과는 정보를 처리하는 주체가 TCP, SCTP 또는 UDP인지에 따라 달라집니다.

TCP 세그먼트화

TCP는 데이터가 수신 호스트로 성공적으로 배달되도록 보장하므로 "연결 지향" 프로토콜이라고도 합니다. 그림 1–1에서는 TCP 프로토콜이 `rlogin` 명령에서 스트림을 수신하는 방법을 보여 줍니다. 그런 다음 TCP가 응용 프로그램 계층에서 수신한 데이터를 세그먼트로 나누고 헤더를 각 세그먼트에 연결합니다.

세그먼트 헤더에는 송신 및 수신 포트, 세그먼트 순서 지정 정보 및 **체크섬**으로 알려진 데이터 필드가 포함됩니다. 두 호스트 모두에 있는 TCP 프로토콜은 체크섬 데이터를 사용하여 데이터가 오류 없이 전송되는지 여부를 확인합니다.

TCP 연결 설정

TCP는 세그먼트를 사용하여 수신 시스템이 데이터를 수신할 준비가 되었는지 여부를 확인합니다. 송신 TCP가 연결을 설정하려고 하면 TCP는 *SYN*이라고 하는 세그먼트를 수신 호스트의 TCP 프로토콜로 보냅니다. 수신 TCP는 *ACK*라고 하는 세그먼트를 반환하여 세그먼트의 성공적인 수신에 긍정 응답합니다. 송신 TCP는 다른 ACK 세그먼트를 보낸 다음 계속 데이터를 보냅니다. 이러한 제어 정보 교환을 *3*선 **핸드셰이크**라고 합니다.

UDP 패킷

UDP는 "비연결" 프로토콜입니다. TCP와 달리 UDP는 데이터가 수신 호스트에 도착했는지를 확인하지 않습니다. 대신 UDP는 응용 프로그램 계층에서 *UDP* **패킷**으로 수신한 메시지의 형식을 지정합니다. UDP는 헤더를 각 패킷에 연결합니다. 헤더에는 송신 및 수신 포트, 패킷 길이가 포함된 필드 및 체크섬이 포함됩니다.

송신 UDP 프로세스는 패킷을 수신 호스트의 해당 피어 UDP 프로세스에 보내려고 합니다. 응용 프로그램 계층은 수신 UDP 프로세스가 패킷 수신에 긍정 응답하는지 여부를 확인합니다. UDP는 수신 알림이 필요하지 않습니다. UDP는 3선 핸드셰이크를 사용하지 않습니다.

인터넷 계층: 패킷 배달이 준비되는 위치

전송 프로토콜 TCP, UDP 및 SCTP는 해당 세그먼트 및 패킷을 IP 프로토콜이 세그먼트 및 패킷을 처리하는 인터넷 계층으로 아래로 전달합니다. IP는 이러한 세그먼트 및 패킷의 형식을 *IP* **데이터그램**이라고 하는 단위로 지정하여 배달 준비합니다. 그런 다음 IP는 데이터그램이 수신 호스트에 효율적으로 배달될 수 있도록 해당 데이터그램에 대한 IP 주소를 결정합니다.

IP 데이터그램

IP는 *IP* **헤더**를 세그먼트 또는 패킷의 헤더는 물론 TCP 또는 UDP에서 추가한 정보에도 연결합니다. IP 헤더의 정보에는 송신 및 수신 호스트의 IP 주소, 데이터그램 길이 및 데이터그램 시퀀스 순서가 포함됩니다. 데이터그램이 네트워크 패킷에 대해 허용할 수 있는 바이트 크기를 초과하거나 단편화되어야 하는 경우 이 정보가 제공됩니다.

데이터 링크 계층: 프레이밍이 발생하는 위치

PPP와 같은 데이터 링크 계층 프로토콜은 IP 데이터그램의 형식을 **프레임**으로 지정합니다. 이러한 프로토콜은 세번째 헤더 및 바닥글을 연결하여 데이터그램을 "프레이밍"합니다. 프레임 헤더에는 프레임이 네트워크 매체를 통해 이동하면서 오류를 검사하는 **순환 중복 검사**(CRC) 필드가 포함됩니다. 그런 다음 데이터 링크 계층이 프레임을 물리적 계층에 전달합니다.

물리적 네트워크 계층: 프레임이 송수신되는 위치

송신 호스트의 물리적 네트워크 계층이 프레임을 수신하고 IP 주소를 네트워크 매체에 적합한 하드웨어 주소로 변환합니다. 그런 다음 물리적 네트워크 계층이 프레임을 네트워크 매체를 통해 외부로 보냅니다.

수신 호스트가 패킷을 처리하는 방법

패킷이 수신 호스트에 도착하면 패킷은 보내진 순서와 반대로 TCP/IP 프로토콜 스택을 통해 이동합니다. 그림 1–1에서는 이 경로를 보여 줍니다. 또한 수신 호스트의 각 프로토콜은 송신 호스트의 해당 피어로 패킷에 연결된 헤더 정보를 제거합니다.

프로세스는 다음과 같습니다.

1. 물리적 네트워크 계층은 패킷을 해당 프레임 형식으로 수신합니다. 물리적 네트워크 계층은 패킷의 CRC를 계산한 다음 프레임을 데이터 링크 계층으로 보냅니다.
2. 데이터 링크 계층은 프레임의 CRC가 정확하고 프레임 헤더 및 CRC를 제거하는지 확인합니다. 최종적으로 데이터 링크 프로토콜은 프레임을 인터넷 계층으로 보냅니다.
3. 인터넷 계층은 헤더의 정보를 읽어 전송을 식별합니다. 그런 다음 인터넷 계층은 패킷이 단편인지 여부를 확인합니다. 전송이 단편화된 경우 IP는 단편을 원래 데이터그램으로 재어셈블합니다. 그런 다음 IP는 IP 헤더를 제거하고 데이터그램을 전송 계층 프로토콜로 전달합니다.
4. 전송 계층(TCP, SCTP 및 UDP)은 헤더를 읽어 데이터를 수신해야 하는 응용 프로그램 계층 프로토콜을 확인합니다. 그런 다음 TCP, SCTP 또는 UDP가 관련 헤더를 제거합니다. TCP, SCTP 또는 UDP가 메시지 또는 스트림을 수신 응용 프로그램으로 보냅니다.
5. 응용 프로그램 계층이 메시지를 수신합니다. 그런 다음 응용 프로그램 계층이 송신 호스트가 요청한 작업을 수행합니다.

TCP/IP 내부 추적 지원

TCP/IP는 RST 패킷이 연결을 종료하면 TCP 통신을 로깅하여 내부 추적 지원을 제공합니다. RST 패킷이 전송되거나 수신되면 전송된 10개의 패킷에 대한 정보가 연결 정보와 함께 로깅됩니다.

TCP/IP 및 인터넷에 대한 자세한 정보 찾기

TCP/IP 및 인터넷에 대한 정보를 광범위하게 이용할 수 있습니다. 이 내용에서 다루지 않는 특정 정보가 필요한 경우 다음에 인용된 출처에서 필요한 사항을 찾을 수 있습니다.

TCP/IP에 대한 컴퓨터 설명서

현지 도서관이나 컴퓨터 서점에서 TCP/IP 및 인터넷에 대한 다양한 일반서를 구할 수 있습니다.

일반적으로 많이 보는 TCP/IP에 대한 두 설명서는 다음과 같습니다.

- Craig Hunt. **TCP/IP Network Administration** – 이 설명서에는 이기종 TCP/IP 네트워크 관리에 대한 몇 가지 이론과 매우 실용적인 정보가 포함됩니다.
- W. Richard Stevens. **TCP/IP Illustrated, Volume I** – 이 설명서는 TCP/IP 프로토콜에 대해 심층적으로 설명합니다. 이 설명서는 TCP/IP의 기술적 사전 지식이 필요한 네트워크 관리자와 네트워크 프로그래머에게 적합합니다.

TCP/IP 및 네트워킹 관련 웹 사이트

인터넷에는 TCP/IP 프로토콜 및 해당 관리만 집중적으로 다루는 수많은 웹 사이트 및 사용자 그룹이 있습니다. Oracle Corporation을 비롯한 많은 제조업체는 일반 TCP/IP 정보에 대한 웹 기반 리소스를 제공합니다. 다음은 TCP/IP 정보 및 일반 시스템 관리 정보에 유용한 웹 리소스입니다. 다음 표에서는 관련 웹 사이트와 해당 사이트에서 제공하는 네트워킹 정보에 대한 설명을 보여 줍니다.

웹 사이트	설명
The Internet Engineering Task Force (IETF) 웹 사이트 (`http://www.ietf.org/home.html`)	IETF는 인터넷 구조 및 관리를 책임지는 조직입니다. IETF 웹 사이트에는 이 조직의 다양한 작업에 대한 정보가 있습니다. 또한 이 사이트에는 IETF의 주요 게시물에 대한 링크가 있습니다.

웹 사이트	설명
Oracle Corporation's BigAdmin Portal (http://www.oracle.com/technetwork/systems/index.html)	BigAdmin에서는 Sun 컴퓨터 관리에 대한 정보를 제공합니다. 이 사이트에서는 네트워킹을 비롯하여 Oracle Solaris 관리와 관련된 FAQ, 리소스, 토론, 설명서 링크 및 기타 자료를 제공합니다.

RFC(Requests for Comment) 및 인터넷 초안

IETF(Internet Engineering Task Force) 작업 그룹은 RFC(*Requests for Comment*)로 알려진 표준 문서를 게시합니다. 개발 중인 표준은 **인터넷 초안**으로 게시됩니다. IAB(Internet Architecture Board)는 공용 도메인에 배치되기 전에 모든 RFC를 승인해야 합니다. 일반적으로 RFC 및 인터넷 초안은 개발자 및 고급 기술을 갖춘 독자를 위한 것입니다. 그러나 TCP/IP 주제를 다루는 수많은 RFC에는 시스템 관리자에게 중요한 정보가 포함됩니다. 이러한 RFC는 본 설명서의 여러 부분에서 인용되고 있습니다.

일반적으로 FYI(For Your Information) 문서는 RFC의 하위 세트로 표시됩니다. FYI에는 인터넷 표준이 아닌 정보가 포함됩니다. FYI에는 일반적인 특징의 인터넷 정보가 포함됩니다. 예를 들어, FYI 문서에는 TCP/IP 입문서 및 논문이 나열되는 참고 문헌이 포함됩니다. FYI 문서는 인터넷 관련 소프트웨어 도구에 대해 광범위한 요약을 제공합니다. 마지막으로 FYI 문서에는 인터넷 및 일반 네트워킹 용어의 용어 해설이 포함됩니다.

본 설명서 및 Oracle Solaris 시스템 관리 모음의 다른 설명서에서 관련 RFC에 대한 참조를 찾을 수 있습니다.

제 2 부

TCP/IP 관리

이 파트에서는 TCP/IP 네트워크 구성, 관리 및 문제 해결 관련 작업과 개념을 다룹니다.

◆ ◆ ◆ 2 장

2 TCP/IP 네트워크 계획(작업)

이 장에서는 체계적이고 비용 효율적인 방식으로 네트워크를 만들기 위해 해결해야 하는 문제를 설명합니다. 이러한 문제를 해결한 후에 향후 네트워크 구성 및 관리를 위한 네트워크 계획을 세울 수 있습니다.

이 장은 다음 정보를 포함합니다.

네트워크 구성 작업은 5 장, "TCP/IP 네트워크 서비스 구성 및 IPv4 주소 지정(작업)"을 참조하십시오.

네트워크 계획(작업 맵)

다음 표에서는 네트워크 구성에 대한 여러 작업을 보여 줍니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	정보
1. 하드웨어 요구 사항 및 네트워크 토폴로지를 계획합니다.	사이트에 필요한 장비 유형 및 해당 장비의 레이아웃을 결정합니다.	■ 일반적인 네트워크 토폴로지 질문은 51 페이지 “네트워크 하드웨어 결정”을 참조하십시오. ■ IPv6 토폴로지 계획은 82 페이지 “IPv6 지원을 위한 네트워크 토폴로지 준비”를 참조하십시오. ■ 특정 장비 유형에 대한 자세한 내용은 장비 제조업체 설명서를 참조하십시오.
2. 네트워크의 등록된 IP 주소를 가져옵니다.	인터넷 등을 통해 로컬 네트워크 외부로 통신할 계획인 경우 네트워크에 고유한 IP 주소가 있어야 합니다.	53 페이지 “네트워크의 IP 번호 얻기”를 참조하십시오.
3. IPv4 네트워크 접두어 또는 IPv6 사이트 접두어를 기반으로 시스템에 대한 IP 주소 지정 체계를 만듭니다.	사이트에서 주소가 배치되는 방식을 결정합니다.	51 페이지 “네트워크에 대한 IP 주소 지정 형식 결정” 또는 86 페이지 “IPv6 주소 지정 계획 준비”를 참조하십시오.
4. 네트워크에 있는 모든 시스템의 IP 주소 및 호스트 이름이 포함된 목록을 만듭니다.	이 목록을 사용하여 네트워크 데이터베이스를 작성합니다.	60 페이지 “네트워크 데이터베이스”를 참조하십시오.
5. 네트워크에서 사용할 이름 서비스를 결정합니다.	로컬 /etc 디렉토리에서 NIS, LDAP, DNS 또는 네트워크 데이터베이스를 사용할지를 결정합니다.	59 페이지 “이름 서비스 및 디렉토리 서비스 선택”을 참조하십시오.
6. 네트워크에 대해 적합한 경우 관리 세분화를 설정합니다.	사이트에서 네트워크를 관리 세분화로 구분해야 할지 여부를 결정합니다.	61 페이지 “관리 세분화”를 참조하십시오.
7. 네트워크 설계 시 라우터를 배치할 위치를 결정합니다.	라우터가 필요한 만큼 네트워크가 큰 경우 라우터를 지원하는 네트워크 토폴로지를 만듭니다.	61 페이지 “네트워크의 라우터 계획”을 참조하십시오.
8. 필요한 경우 서브넷에 대한 전략을 설계합니다.	IP 주소 공간을 관리하기 위한 서브넷을 만들거나 사용자가 사용할 수 있는 IP 주소를 추가로 만들어야 할 수도 있습니다.	IPv4 서브넷 계획은 224 페이지 “서브넷이란?”을 참조하십시오. IPv6 서브넷 계획은 87 페이지 “서브넷 번호 지정 체계 만들기”를 참조하십시오.

네트워크 하드웨어 결정

네트워크를 설계할 때 조직의 요구 사항을 가장 잘 충족하는 네트워크 유형을 결정해야 합니다. 결정해야 할 몇 가지 계획 요소에는 다음과 같은 네트워크 하드웨어가 포함됩니다.

- 네트워크 토폴로지, 레이아웃 및 네트워크 하드웨어 연결
- 네트워크에서 지원할 수 있는 호스트 시스템 수
- 네트워크에서 지원하는 호스트 유형
- 필요한 서버 유형
- 사용할 네트워크 매체의 유형(이더넷, 토큰 링, FDDI 등)
- 이 매체를 확장하거나 로컬 네트워크를 외부 네트워크에 연결할 브릿지 또는 라우터가 필요한지 여부
- 일부 시스템의 내장 인터페이스 외에 별도로 인터페이스를 구매해야 하는지 여부

이러한 요인을 기반으로 LAN(Local Area Network)의 규모를 결정할 수 있습니다.

주 – 네트워크 하드웨어를 계획하는 방법은 본 설명서에서 다루지 않습니다. 자세한 내용은 하드웨어와 함께 제공되는 설명서를 참조하십시오.

네트워크에 대한 IP 주소 지정 형식 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.

네트워크 주소 지정 체계를 계획할 때 다음 요인을 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

1990년 이후 인터넷이 전세계적으로 성장하여 사용 가능한 IP 주소가 부족해졌습니다. 이러한 상황을 개선하기 위해 IETF(Internet Engineering Task Force)가 여러 IP 주소 지정 대안을 개발했습니다.

조직의 네트워크에 둘 이상의 IP 주소가 지정되었거나 서브넷을 사용하는 경우 네트워크 IP 주소를 지정하는 조직 내 중앙 부서를 만듭니다. 이러한 부서는 지정된 네트워크 IP

주소 풀의 제어를 유지 관리하고, 필요한 네트워크, 서브넷 및 호스트 주소를 지정해야 합니다. 문제를 방지하기 위해 조직에 중복되거나 임의의 네트워크 번호가 존재하지 않도록 합니다. 오늘날 사용 중인 IP 주소의 유형은 다음과 같습니다.

IPv4 주소

이러한 32비트 주소는 TCP/IP용으로 설계된 원래 IP 주소 지정 형식입니다. 원래 IP 네트워크에는 3가지 클래스인 A, B 및 C가 있습니다. 네트워크에 지정된 **네트워크 번호**는 이 클래스 지정과 호스트를 나타내는 비트(8비트 이상)로 구성됩니다. 클래스 기반 IPv4 주소를 사용하려면 네트워크 번호에 대한 넷마스크를 구성해야 합니다. 또한 로컬 네트워크의 시스템에 사용 가능한 주소를 추가로 만들기 위해 이러한 주소는 서브넷으로 구분되기도 합니다.

오늘날 IP 주소를 *IPv4* **주소**라고 합니다. 이제 ISP에서 클래스 기반 IPv4 네트워크 번호를 얻을 수는 없지만 많은 기존 네트워크에서는 이 네트워크 번호를 계속 사용하고 있습니다. IPv4 주소 관리에 대한 자세한 내용은 55 페이지 “IPv4 주소 지정 체계 설계”를 참조하십시오.

CIDR 형식의 IPv4 주소

IETF는 IPv4 주소 부족에 대한 중/단기적인 해결책으로 CIDR(Classless Inter-Domain Routing) 주소를 개발했습니다. 또한 전역 인터넷 경로 지정 테이블의 용량 부족에 대한 해결책으로 CIDR 형식을 설계했습니다. CIDR 표기법의 IPv4 주소는 길이가 32비트이며 동일한 점으로 구분된 십진수 형식입니다. 그러나 CIDR은 가장 오른쪽 바이트 다음에 접두어 지정을 추가하여 IPv4 주소의 네트워크 부분을 정의합니다. 자세한 내용은 56 페이지 “CIDR IPv4 주소 지정 체계 설계”를 참조하십시오.

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 12 장, “DHCP 정보(개요)”를 참조하십시오.

IPv6 주소

IETF는 사용 가능한 IPv4 주소 부족에 대한 장기적 해결책으로 128비트 IPv6 주소를 배포했습니다. IPv6 주소는 IPv4에서 사용 가능한 주소 공간보다 더 큰 주소 공간을 제공합니다. Oracle Solaris는 이중 스택 TCP/IP 사용을 통해 동일한 호스트에서의 IPv4 및

IPv6 주소 지정을 지원합니다. CIDR 형식의 IPv4 주소와 같이 IPv6 주소에도 네트워크 클래스 또는 넷마스크에 대한 개념이 없습니다. CIDR처럼 IPv6 주소는 접두어를 사용하여 사이트의 네트워크를 정의하는 주소 부분을 지정합니다. IPv6에 대한 소개는 70 페이지 “IPv6 주소 지정 개요”를 참조하십시오.

개인 주소 및 설명서 접두어

IANA는 개인 네트워크에 사용하도록 IPv4 주소 블록 및 IPv6 사이트 접두어를 예약했습니다. 엔터프라이즈 네트워크 내 시스템에서는 이러한 주소를 배포할 수 있지만 개인 주소의 패킷은 인터넷을 통해 경로 지정할 수 없습니다. 개인 주소에 대한 자세한 내용은 57 페이지 “개인 IPv4 주소 사용”을 참조하십시오.

주 – 개인 IPv4 주소는 설명서용으로 예약되어 있기도 합니다. 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.

네트워크를 영구적 개인 네트워크로 계획하지 않는다면 로컬 사용자는 일반적으로 로컬 네트워크 외부로 통신할 것입니다. 따라서 네트워크를 외부로 통신할 수 있으려면 먼저 해당 조직에서 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.

주 – 현재 외부 TCP/IP 네트워크에 네트워크를 연결하지 않아도 임의로 네트워크에 IP 주소를 지정하지 마십시오. 대신 57 페이지 “개인 IPv4 주소 사용”에서 설명한 대로 개인 주소를 사용합니다.

IPv4 주소 지정 체계 설계

주 – IPv6 주소 계획 정보는 86 페이지 “IPv6 주소 지정 계획 준비”를 참조하십시오.

이 절에서는 IPv4 주소 지정 계획 설계를 지원하기 위한 IPv4 주소 지정 개요를 제공합니다. IPv6 주소에 대한 자세한 내용은 70 페이지 “IPv6 주소 지정 개요”를 참조하십시오. DHCP 주소에 대한 자세한 내용은 12 장, “DHCP 정보(개요)”를 참조하십시오.

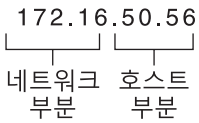
각 IPv4 기반 네트워크는 다음을 포함해야 합니다.

- ISP 또는 IR에 의해 지정되거나 IANA에서 등록한 이전 네트워크에 대해 지정된 고유한 네트워크 번호. 개인 주소를 사용할 계획인 경우 조직 내에서 고유한 네트워크 번호를 만들어야 합니다.
- 네트워크에 있는 모든 시스템의 인터페이스에 대한 고유한 IPv4 주소
- 네트워크 마스크

IPv4 주소는 58 페이지 “IP 주소를 네트워크 인터페이스에 적용하는 방법”에 설명된 대로 시스템에서 네트워크 인터페이스를 고유하게 식별하는 32비트 번호입니다. IPv4 주소는 마침표로 구분되는 4개의 8비트 필드로 구분된 십진수로 작성됩니다. 각 8비트 필드는 IPv4 주소에서 1바이트를 나타냅니다. IPv4 주소의 바이트를 나타내는 이러한 형식을 **점으로 구분된 십진수 형식**이라고도 합니다.

다음 그림에서는 IPv4 주소 `172.16.50.56`의 구성 요소 부분을 보여 줍니다.

그림 2–1 IPv4 주소 형식



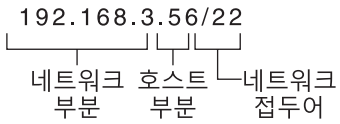
`172.16` 등록된 IPv4 네트워크 번호입니다. 클래스 기반 IPv4 표기법에서 이 번호는 IP 네트워크 클래스를 정의하기도 합니다. 이 예에서는 IANA에 의해 등록된 클래스 B입니다.

`50.56` IPv4 주소의 호스트 부분입니다. 호스트 부분은 네트워크에 있는 시스템의 인터페이스를 고유하게 식별합니다. 로컬 네트워크의 각 인터페이스에 대해 주소의 네트워크 부분은 동일하지만 호스트 부분은 서로 달라야 합니다.

클래스 기반 IPv4 네트워크를 서브넷할 계획인 경우 서브넷 마스크를 정의하거나 224 페이지 “netmasks 데이터베이스”에 설명된 대로 **넷마스크**를 정의해야 합니다.

다음 예에서는 CIDR 형식의 주소 `192.168.3.56/22`를 보여 줍니다.

그림 2–2 CIDR 형식 IPv4 주소



`192.168.3` ISP 또는 IR에서 수신된 IPv4 네트워크 번호로 구성되는 네트워크 부분입니다.

`56` 시스템의 인터페이스에 지정되는 호스트 부분입니다.

`/22` 네트워크 번호를 구성하는 주소의 비트 수를 정의하는 네트워크 접두어입니다. 또한 네트워크 접두어는 IP 주소에 대한 서브넷 마스크를 제공합니다. 네트워크 접두어는 ISP 또는 IR에 의해서도 지정됩니다.

Oracle Solaris 기반 네트워크는 표준 IPv4 주소, CIDR 형식 IPv4 주소, DHCP 주소, IPv6 주소 및 개인 IPv4 주소를 결합할 수 있습니다.

IPv4 주소 지정 체계 설계

이 절에서는 표준 IPv4 주소가 구성되는 클래스에 대해 설명합니다. 이제 IANA에서 클래스 기반 네트워크 번호를 제공하지는 않지만 많은 네트워크에서 이러한 네트워크 번호를 계속 사용하고 있습니다. 클래스 기반 네트워크 번호를 사용하여 사이트에 대한 주소 공간을 관리해야 합니다. IPv4 네트워크 클래스에 대한 전체 내용은 237 페이지 “네트워크 클래스”를 참조하십시오.

다음 표에서는 표준 IPv4 주소를 네트워크 및 호스트 주소 공간으로 구분한 것을 보여 줍니다. 각 클래스에 대해 “범위”는 네트워크 번호의 첫번째 바이트에 대한 십진수 값 범위를 지정합니다. “네트워크 주소”는 주소의 네트워크 부분에만 지정되는 IPv4 주소의 바이트 수를 나타냅니다. 각 바이트는 *xxx*로 나타냅니다. “호스트 주소”는 주소의 호스트 부분에만 지정되는 바이트 수를 나타냅니다. 예를 들어, 클래스 A 네트워크 주소에서 첫번째 바이트는 네트워크에만 지정되며 마지막 3개의 바이트는 호스트에만 지정됩니다. 클래스 C 네트워크에 대해서는 반대로 지정됩니다.

표 2–1 IPv4 클래스의 구분

클래스	바이트 범위	네트워크 번호	호스트 주소
A	0–127	*xxx*	*xxx.xxx. xxx*
B	128–191	*xxx.xxx*	*xxx.xxx*
C	192–223	*xxx.xxx. xxx*	*xxx*

IPv4 주소의 첫번째 바이트에 있는 번호는 네트워크가 클래스 A, B 또는 C인지를 정의합니다. 남은 3개 바이트의 범위는 0–255입니다. 두 번호인 0와 255는 예약됩니다. IANA에 의해 네트워크에 지정된 네트워크 클래스에 따라 각 바이트에 번호 1–254를 지정할 수 있습니다.

다음 표에서는 사용자에게 지정되는 IPv4 주소의 바이트를 보여 줍니다. 또한 이 표는 호스트에 지정할 수 있는 각 바이트 내 번호 범위를 보여 줍니다.

표 2–2 사용 가능한 IPv4 클래스 범위

네트워크 클래스	바이트 1 범위	바이트 2 범위	바이트 3 범위	바이트 4 범위
A	0–127	1–254	1–254	1–254
B	128–191	IANA에 의해 사전 지정됨	1–254	1–254
C	192–223	IANA에 의해 사전 지정됨	IANA에 의해 사전 지정됨	1–254

IPv4 서브넷 번호

호스트 수가 많은 로컬 네트워크는 서브넷으로 분리하기도 합니다. IPv4 네트워크 번호를 서브넷으로 구분하는 경우 각 서브넷에 네트워크 식별자를 지정해야 합니다. IPv4 주소의 호스트 부분에 있는 일부 비트를 네트워크 식별자로 사용하여 IPv4 주소 공간의 효율성을 극대화할 수 있습니다. 네트워크 식별자로 사용되면 주소의 지정된 부분은 서브넷 번호가 됩니다. IPv4 주소의 네트워크 및 서브넷 부분을 선택하는 비트마스크인 넷마스크를 사용하여 서브넷 번호를 만듭니다. 자세한 내용은 224 페이지 “IPv4 주소에 대한 네트워크 마스크 만들기”를 참조하십시오.

CIDR IPv4 주소 지정 체계 설계

원래 IPv4를 구성했던 네트워크 클래스는 더 이상 전역 인터넷에서 사용되지 않습니다. 오늘날 IANA는 클래스 없는 CIDR 형식의 주소를 해당 레지스트리로 전 세계에 배포합니다. 그림 2–2에 나와 있는 대로 ISP에서 얻은 모든 IPv4 주소는 CIDR 형식입니다.

CIDR 주소의 네트워크 접두어는 네트워크의 호스트에 사용할 수 있는 IPv4 주소 수를 나타냅니다. 이러한 호스트 주소는 호스트의 인터페이스에 지정됩니다. 호스트에 물리적 인터페이스가 두 개 이상 있는 경우 사용 중인 모든 물리적 인터페이스에 대해 호스트 주소를 지정해야 합니다.

또한 CIDR 주소의 네트워크 접두어는 서브넷 마스크의 길이를 정의합니다. 대부분의 Oracle Solaris 명령은 네트워크 서브넷 마스크의 CIDR 접두어 지정을 인식합니다. 그러나 Oracle Solaris 설치 프로그램 및 `/etc/netmask` **파일**을 사용하려면 점으로 구분된 십진수 표현을 사용하여 서브넷 마스크를 설정해야 합니다. 이러한 두 경우에서 다음 표에 나와 있는 대로 CIDR 네트워크 접두어의 점으로 구분된 십진수 표현을 사용합니다.

표 2–3 CIDR 접두어 및 이와 동등한 십진수

CIDR 네트워크 접두어	사용 가능한 IP 주소	동등한 점으로 구분된 십진수 서브넷
/19	8,192	255.255.224.0
/20	4,096	255.255.240.0
/21	2,048	255.255.248.0
/22	1024	255.255.252.0
/23	512	255.255.254.0
/24	256	255.255.255.0
/25	128	255.255.255.128
/26	64	255.255.255.192
/27	32	255.255.255.224

CIDR 주소에 대한 자세한 내용은 다음 자료를 참조하십시오.

- CIDR에 대한 자세한 기술 정보는 RFC 1519, Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy (`http://www.ietf.org/rfc/rfc1519.txt?number=1519`)를 참조하십시오.
- CIDR에 대한 보다 일반적인 정보는 Classless Inter-Domain Routing (CIDR) Overview (`http://www.wirelesstek.com/cidr.htm`)의 Pacific Bell Internet을 참조하십시오.
- 다른 CIDR 개요는 Wikipedia 문서인 "Classless inter-domain routing" (`http://en.wikipedia.org/wiki/Classless_inter-domain_routing`)을 참조하십시오.

개인 IPv4 주소 사용

IANA는 자체 개인 네트워크에서 사용할 회사용 IPv4 주소의 세 블록을 예약했습니다. 이러한 주소는 RFC 1918, Address Allocation for Private Internets (`http://www.ietf.org/`

rfc/rfc1918.txt?number=1918)에 정의되어 있습니다. 회사 인트라넷 내 로컬 네트워크에 있는 시스템에 대해 이러한 **개인 주소**를 사용할 수 있으며, 1918 주소라고도 합니다. 그러나 개인 주소는 인터넷에서 유효하지 않습니다. 이러한 개인 주소는 로컬 네트워크 외부로 통신해야 하는 시스템에서는 사용하지 마십시오.

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

IP 주소를 네트워크 인터페이스에 적용하는 방법

네트워크에 연결하려면 시스템에 **물리적 네트워크 인터페이스**가 한 개 이상 있어야 합니다. 각 네트워크 인터페이스에는 고유한 IP 주소가 있어야 합니다. Oracle Solaris 설치 시 설치 프로그램이 찾는 첫번째 인터페이스에 대한 IP 주소를 지정해야 합니다. 일반적으로 해당 인터페이스의 이름은 *device-name*0입니다(예: `eri0` 또는 `hme0`). 이 인터페이스는 **주 네트워크 인터페이스**로 간주됩니다.

호스트에 두번째 네트워크 인터페이스를 추가하는 경우 해당 인터페이스에는 고유한 IP 주소가 있어야 합니다. 두번째 네트워크 인터페이스를 추가하면 해당 호스트는 **멀티홈**이 됩니다. 반대로 호스트에 두번째 네트워크 인터페이스를 추가하고 IP 전달을 사용으로 설정하면 해당 호스트는 라우터가 됩니다. 자세한 내용은 112 페이지 “IPv4 라우터 구성”을 참조하십시오.

각 네트워크 인터페이스에는 장치 이름, 장치 드라이버 및 `/devices` 디렉토리의 연결된 장치 파일이 포함됩니다. 네트워크 인터페이스는 일반적으로 사용되는 두 가지 이더넷 인터페이스에 대한 장치 이름인 `eri` 또는 `smc0`과 같은 장치 이름을 사용할 수도 있습니다.

인터페이스와 관련된 정보 및 작업은 6 장, “네트워크 인터페이스 관리(작업)”를 참조하십시오.

주 – 본 설명서에서는 시스템에 이더넷 네트워크 인터페이스가 있다고 가정합니다. 다른 네트워크 매체를 사용할 계획인 경우 구성 정보에 대한 자세한 내용은 네트워크 인터페이스와 함께 제공되는 설명서를 참조하십시오.

네트워크의 이름 지정 엔티티

지정한 네트워크 IP 주소를 수신하고 IP 주소로 모든 시스템의 NIC를 구성했으면 다음 작업은 호스트에 이름을 지정하는 것입니다. 그런 다음 네트워크에서 이름 서비스를 처리하는 방법을 결정해야 합니다. 처음에 네트워크를 설정한 다음 라우터, 브리지 또는 PPP를 통해 네트워크를 확장할 때 이러한 이름을 사용합니다.

TCP/IP 프로토콜은 IP 주소를 사용하여 네트워크에서 시스템을 찾습니다. 그러나 인식할 수 있는 이름을 사용하면 시스템을 쉽게 식별할 수 있습니다. 따라서 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

많은 사이트에서는 사용자가 해당 시스템에 대한 호스트 이름을 선택할 수 있습니다. 또한 서버에는 주 네트워크 인터페이스의 IP 주소와 연결된 호스트 이름이 한 개 이상 필요합니다.

시스템 관리자는 도메인의 각 호스트 이름이 고유한지 확인해야 합니다. 즉, 네트워크에 있는 두 시스템의 이름이 모두 "fred"여서는 안 됩니다. 그러나 시스템 "fred"의 IP 주소는 여러 개가 가능합니다.

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 3가지 유형의 이름 서비스인 로컬 파일, NIS 및 DNS를 사용으로 설정할 수 있습니다. 이름 서비스는 호스트 이름, IP 주소, 이더넷 주소와 같은 네트워크의 시스템에 대한 중요한 정보를 유지 관리합니다. 또한 Oracle Solaris에서는 이름 서비스 이외에 또는 이름 서비스를 대신해서 LDAP 디렉토리 서비스를 사용할 수 있습니다. Oracle Solaris의 이름 서비스 소개는 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**의 제I부, “About Naming and Directory Services”를 참조하십시오.

네트워크 데이터베이스

운영 체제 설치 시 절차 중에 서버, 클라이언트 또는 독립형 시스템의 호스트 이름 및 IP 주소를 제공합니다. Oracle Solaris 설치 프로그램은 이 정보를 Solaris 10 11/06 및 이전 Solaris 10 릴리스의 `hosts` 및 `ipnodes` 네트워크 데이터베이스에 추가합니다. 이 데이터베이스는 네트워크의 TCP/IP 작업에 필요한 정보가 들어 있는 네트워크 데이터베이스 세트의 일부분입니다. 네트워크에 대해 선택한 이름 서비스가 이러한 데이터베이스를 읽습니다.

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. 네트워크 데이터베이스 세트에 대한 자세한 내용은 227 페이지 “네트워크 데이터베이스 및 `nsswitch.conf` 파일”을 참조하십시오.

NIS 또는 DNS를 이름 서비스로 사용

NIS 및 DNS 이름 서비스는 네트워크에 있는 여러 서버의 네트워크 데이터베이스를 유지 관리합니다. **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 에서 이러한 이름 서비스 및 데이터베이스를 구성하는 방법에 대해 설명합니다. “이름 공간” 및 “관리 도메인” 개념에 대해서도 자세히 설명합니다.

로컬 파일을 이름 서비스로 사용

NIS, LDAP 또는 DNS를 구현하지 않는 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

많은 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 없는 경우 조직은 해당 도메인 이름을 기존 최상위 레벨 도메인 중 하나의 바로 아래에 배치할 수 있습니다.

다음은 최상위 레벨 도메인의 몇 가지 예입니다.

- .com – 상업 회사(국제적 범위)
- .edu – 교육 기관(국제적 범위)
- .gov – 미국 정부 기관
- .fr – 프랑스

이름이 고유해야 한다는 규정에 따라 조직을 식별하는 이름을 선택합니다.

관리 세분화

관리 세분화에 대한 질문은 규모 및 제어의 문제를 다룹니다. 네트워크에 있는 호스트 및 서버가 많을수록 관리 작업이 복잡해집니다. 관리 세분화를 추가로 설정하여 이러한 상황을 처리할 수 있습니다. 특정 클래스의 네트워크를 추가합니다. 기존 네트워크를 서브넷으로 구분합니다.

네트워크에 대한 관리 세분화를 설정하는 것은 다음 요소에 따라 결정됩니다.

- **네트워크의 규모**

  단일 관리 세분화는 수백 개 호스트의 단일 네트워크를 처리할 수 있습니다. 이러한 모든 호스트는 동일한 물리적 위치에 있으며 동일한 관리 서비스를 필요로 합니다. 그러나 여러 관리 세분화를 설정해야 하는 경우도 있습니다. 세분화는 서브넷으로 구성된 소규모 네트워크를 사용하고 있으며 해당 네트워크가 광범위한 지리적 영역에 분산되어 있는 경우 특히 유용합니다.

- **네트워크에 있는 사용자의 요구 사항이 서로 유사한지 여부**

  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 105 페이지 “IPv4 네트워크에서의 패킷 전달 및 경로 지정”을 참조하십시오. IPv6 네트워크의 라우터 구성을 위한 자세한 내용 및 작업을 보려면 165 페이지 “IPv6 라우터 구성”을 참조하십시오.

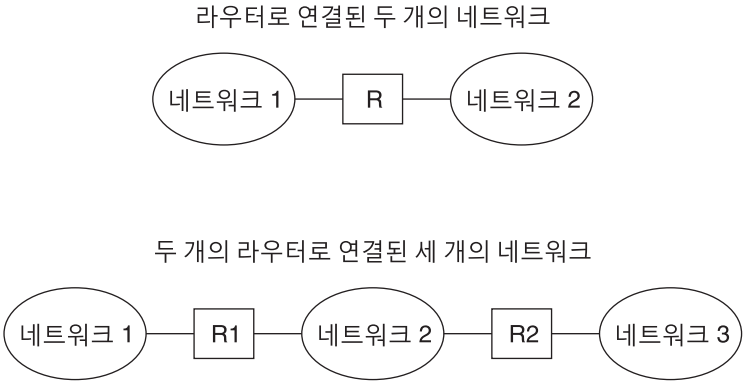
네트워크 토폴로지 개요

네트워크 토폴로지는 네트워크의 연결 방식을 기술합니다. 라우터는 네트워크를 서로 연결해주는 엔티티입니다. 라우터는 두 개 이상의 네트워크 인터페이스를 포함하고 IP 전달을 구현하는 시스템입니다. 하지만 시스템은 112 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.

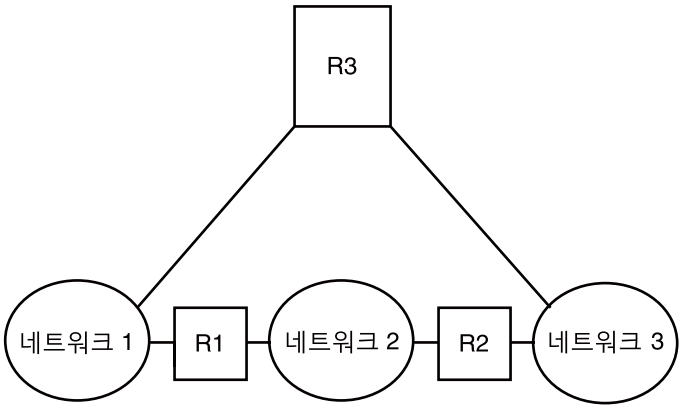
그림 2–3 기본 네트워크 토폴로지



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 그림에서는 보다 복잡한 사례를 보여 줍니다. 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 2–4 네트워크 사이에 추가 경로를 제공하는 네트워크 토폴로지



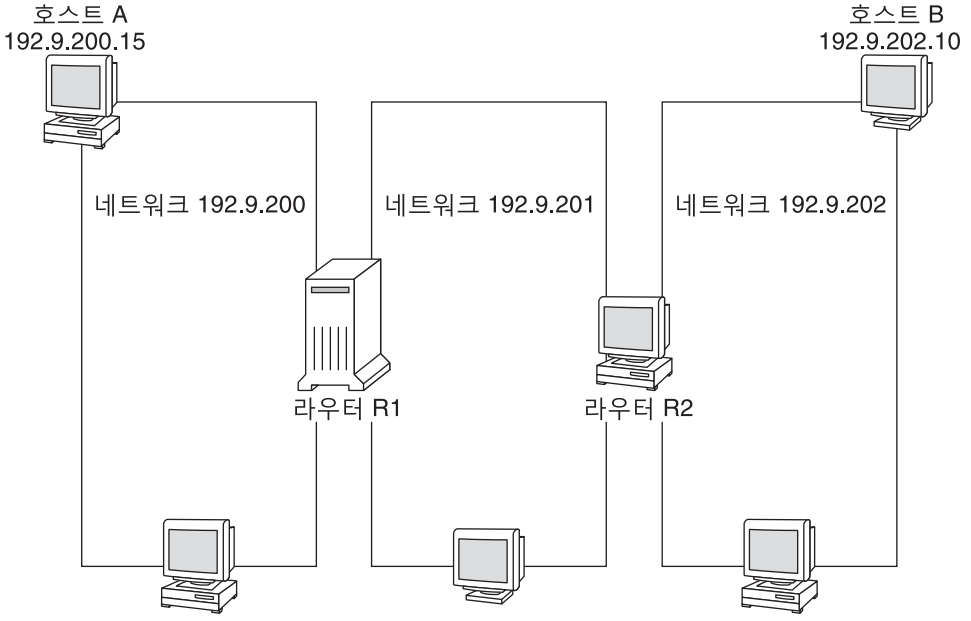
라우터가 패킷을 전송하는 방법

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 112 페이지 “IPv4 라우터 구성”을 참조하십시오.

다음 그림에서는 두 라우터로 연결된 세 네트워크의 네트워크 토폴로지를 보여줍니다.

그림 2–5 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.

◆ ◆ ◆ 3 장

3 IPv6 소개(개요)

이 장에서는 Oracle Solaris IPv6(Internet Protocol version 6) 구현에 대한 개요를 제공합니다. 이 구현에는 IPv6 주소 공간을 지원하는 연결된 데몬 및 유틸리티가 포함됩니다.

IPv6 및 IPv4 주소는 Oracle Solaris 네트워킹 환경에 함께 존재합니다. IPv6 주소로 구성된 시스템에는 해당 IPv4 주소가 포함됩니다(이러한 주소가 이미 존재하는 경우). IPv6 주소와 관련된 작업은 IPv4 작업에 영향을 주지 않으며 IPv4 작업도 IPv6 작업에 영향을 주지 않습니다.

다음 주요 내용으로 구성되어 있습니다.

IPv6에 대한 자세한 내용은 다음 장을 참조하십시오.

- IPv6 네트워크 계획 – 4 장, “IPv6 네트워크 계획(작업)”
- IPv6관련 작업 – 7 장, “IPv6 네트워크 구성(작업)” 및 8 장, “TCP/IP 네트워크 관리(작업)”.
- IPv6 세부 정보 – 11 장, “IPv6 세부 개요(참조)”

IPv6의 주요 기능

IPv6의 특징적인 기능은 IPv4에 비해 늘어난 주소 공간입니다. IPv6은 또한 이 섹션에 설명된 대로 여러 영역에서 인터넷 기능을 향상시켜 줍니다.

확장된 주소 지정

주소 지정 계층의 더 많은 레벨을 지원하도록 IP 주소 크기가 IPv4의 32비트에서 IPv6의 128비트로 늘어났습니다. 또한 IPv6은 더 많은 주소 지정 가능한 IPv6 시스템을 제공합니다. 자세한 내용은 70 페이지 "IPv6 주소 지정 개요"를 참조하십시오.

주소 자동 구성 및 Neighbor Discovery

IPv6 *ND(Neighbor Discovery)* 프로토콜은 IPv6 주소의 자동 구성을 지원합니다. **자동 구성**은 관리를 보다 쉽고 간단하게 해결할 수 있도록 IPv6 호스트가 해당 IPv6 주소를 자동으로 생성할 수 있는 기능입니다. 자세한 내용은 77 페이지 "IPv6 주소 자동 구성"을 참조하십시오.

Neighbor Discovery 프로토콜은 ARP(Address Resolution Protocol), ICMP(Internet Control Message Protocol), RDISC(라우터 검색) 및 ICMP 재지정과 같은 IPv4 프로토콜을 결합한 것입니다. IPv6 라우터는 Neighbor Discovery를 사용하여 IPv6 사이트 접두어를 알립니다. IPv6 호스트는 IPv6 라우터의 접두어를 요청하는 것을 비롯하여 여러 목적으로 Neighbor Discovery를 사용합니다. 자세한 내용은 76 페이지 "IPv6 Neighbor Discovery 프로토콜 개요"를 참조하십시오.

헤더 형식 간소화

IPv6 헤더 형식은 선택적인 특정 IPv4 헤더 필드를 삭제하거나 만듭니다. 이러한 변경 사항에 따라 늘어난 주소 공간에도 불구하고 IPv6 헤더의 대역폭 비용이 가능한 한 낮게 유지됩니다. IPv6 주소가 IPv4 주소보다 4배 이상 길지만 IPv6 헤더는 IPv4 헤더의 2배에 불과합니다.

IP 헤더 옵션에 대한 향상된 지원

보다 효율적인 전달을 위해 IP 헤더 옵션이 인코딩되는 방식이 변경되었습니다. 또한 IPv6 옵션에서 해당 길이에 대한 제한이 조금 더 완화되었습니다. 이러한 변경 사항은 이후 새 옵션을 제공할 수 있는 유연성을 향상시켜 줍니다.

IPv6 주소 지정을 위한 응용 프로그램 지원

예를 들어, 다음과 같은 많은 중요 Oracle Solaris 네트워크 서비스에서 IPv6 주소를 인식하고 지원합니다.

- 이름 서비스, 예: DNS, LDAP 및 NIS. 이러한 이름 서비스의 IPv6 지원에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.
- 인증 및 프라이버시 응용 프로그램, 예: IPsec(IP 보안 구조) 및 IKE(Internet Key Exchange). 자세한 내용은 제4부을 참조하십시오.
- IPQoS(IP 서비스 품질)로 제공되는 차등화 서비스. 자세한 내용은 제6부를 참조하십시오.
- IPMP(IP Network Multipathing)로 제공되는 페일오버 감지. 자세한 내용은 제5부를 참조하십시오.

추가 IPv6 리소스

이 단원 외에도 다음 섹션에 나열된 소스에서 IPv6에 대한 정보를 얻을 수 있습니다.

IPv6 RFC(Requests for Comment) 및 인터넷 초안

IPv6과 관련하여 여러 RFC가 제공됩니다. 다음 테이블에는 이 설명서의 작성 시점에서 중요한 IPv6 문서 및 해당 IETF(Internet Engineering Task Force) 웹 위치가 나열되어 있습니다.

표 3–1 IPv6 관련 RFC 및 인터넷 초안

RFC 또는 인터넷 초안	제목	위치
RFC 2461, **Neighbor Discovery for IP Version 6 (IPv6)**	IPv6 Neighbor Discovery 프로토콜의 특징 및 기능에 대해 설명합니다.	http://www.ietf.org/rfc/rfc2461.txt$number=2461 (http://www.ietf.org/rfc/rfc2461.txt?number-2461)
RFC 3306, **Unicast-Prefix-Based IPv6 Multicast Addresses**	IPv6 멀티캐스트 주소의 형식 및 유형에 대해 설명합니다.	ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt (ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt)
RFC 3484: **Default Address Selection for Internet Protocol version 6 (IPv6)**	IPv6 기본 주소 선택에 사용되는 알고리즘에 대해 설명합니다.	http://www.ietf.org/rfc/rfc3484?number=3484 (http://www.ietf.org/rfc/rfc3484.txt?number=3484)
RFC 3513, **Internet Protocol version 6 (IPv6) Addressing Architecture**	IPv6 주소 유형에 대한 전체 세부 정보 및 여러 예제가 포함됩니다.	http://www.ietf.org/rfc/rfc3513.txt?number=3513 (http://www.ietf.org/rfc/rfc3513.txt?number=3513)

표 3–1 IPv6 관련 RFC 및 인터넷 초안 (계속)

RFC 또는 인터넷 초안	제목	위치
RFC 3587, **IPv6 Global Unicast Address Format**	IPv6 유니캐스트 주소의 표준 형식에 대해 정의합니다.	http://www.ietf.org/rfc/rfc3587.txt?number=3587 (http://www.ietf.org/rfc/rfc3587.txt?number=3587)

웹 사이트

다음 웹 사이트에서는 IPv6에 대한 유용한 정보를 제공합니다.

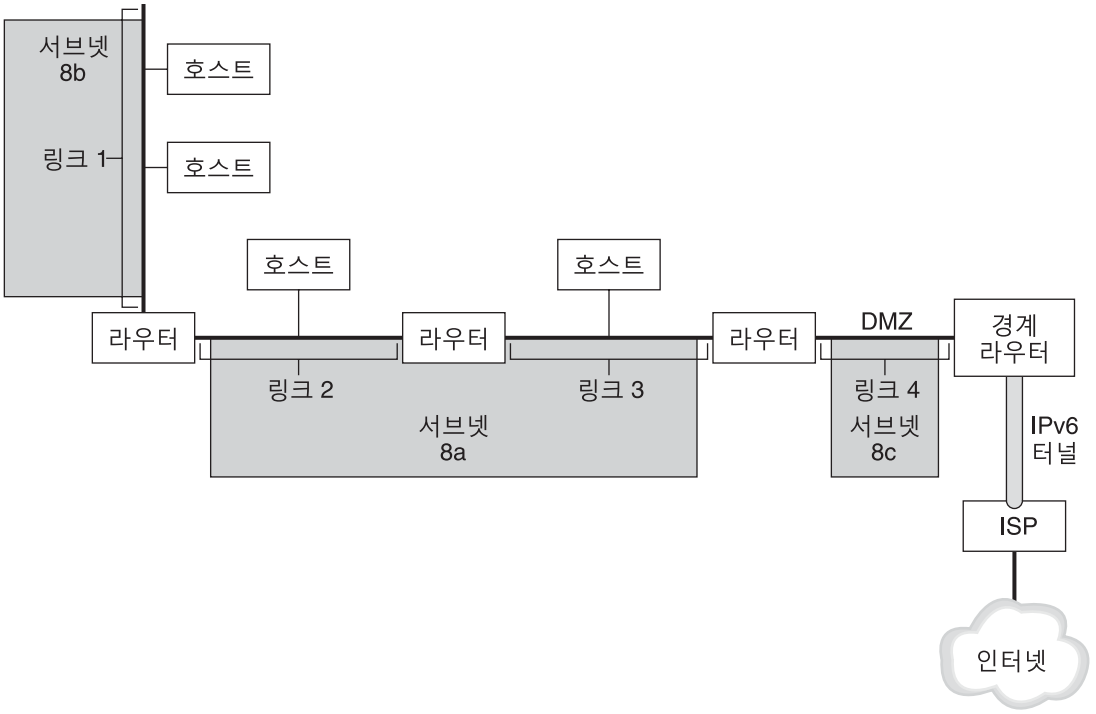
표 3–2 IPv6 관련 웹 사이트

웹 사이트	설명	위치
IPv6 포럼	이 조직의 웹 사이트에서는 IPv6 관련 프리젠테이션, 이벤트, 클래스 및 구현에 대한 전세계 링크가 제공됩니다.	http://www.ipv6forum.com
Internet Educational Task Force IPv6 작업 그룹	이 IETF 작업 그룹의 홈 페이지에서 모든 관련 IPv6 RFC 및 인터넷 초안 링크가 제공됩니다.	http://www.ietf.org/html.charters/ipv6-charter.html

IPv6 네트워크 개요

이 섹션에서는 IPv6 네트워크 토폴로지의 기본 용어에 대해 소개합니다. 다음 그림에서는 IPv6 네트워크의 기본 요소를 보여줍니다.

그림 3–1 IPv6 네트워크의 기본 구성 요소



이 그림은 IPv6 네트워크 및 ISP에 대한 연결을 보여줍니다. 내부 네트워크는 링크 1, 2, 3 및 4로 구성됩니다. 각 링크는 호스트로 채워지고 라우터로 종료됩니다. 네트워크의 DMZ인 링크 4는 경계 라우터의 한쪽 끝에서 종료됩니다. 경계 라우터는 ISP에 대해 IPv6 터널을 실행하여 네트워크에 대한 인터넷 연결을 제공합니다. 링크 2 및 3은 서브넷 8a로 관리됩니다. 서브넷 8b는 링크 1의 시스템으로만 구성됩니다. 서브넷 8c는 링크 4의 DMZ와 연속되어 있습니다.

그림 3–1에 설명된 대로 IPv6 네트워크는 기본적으로 IPv4 네트워크와 동일한 구성 요소를 포함합니다. 하지만 IPv6 용어는 IPv4 용어와 약간 다릅니다. 다음은 IPv6 컨텍스트에서 사용되는 네트워크 구성 요소의 익숙한 용어 목록입니다.

노드 IPv6 주소를 포함하는 모든 시스템 및 IPv6 지원용으로 구성된 인터페이스. 이 일반 용어는 호스트 및 라우터에 모두 적용됩니다.

IPv6 라우터 IPv6 패킷을 전달하는 노드. 라우터의 인터페이스 중 하나 이상이 IPv6 지원용으로 구성되어 있어야 합니다. IPv6 라우터는 또한 내부 네트워크를 통해 엔터프라이즈에 대한 등록된 IPv6 사이트 접두어를 알릴 수 있습니다.

IPv6 호스트 IPv6 주소를 포함하는 노드. IPv6 호스트는 IPv6 지원용으로 구성된 두 개 이상의 인터페이스를 포함할 수 있습니다. IPv4에서와 같이 IPv6 호스트는 패킷을 전달하지 않습니다.

링크 라우터의 어느 한쪽 끝에 바인딩된 단일 연속 네트워크 매체.

인접 로컬 노드와 동일한 링크에 있는 IPv6 노드.

IPv6 서브넷 IPv6 네트워크의 관리 세그먼트. IPv4에서와 같이 한 링크의 모든 노드와 직접 대응될 수 있는 IPv6 서브넷의 구성 요소. 링크의 노드는 필요에 따라 별개의 서브넷에서 관리할 수 있습니다. 또한 IPv6은 다중 링크 서브넷을 지원하며, 두 개 이상의 링크의 노드가 단일 서브넷의 구성 요소일 수 있습니다. 그림 3–1의 링크 2 및 3은 다중 링크 서브넷 8a의 구성 요소입니다.

IPv6 터널 IPv6 노드와 또 다른 IPv6 노드 끝점 사이의 가상 지점 간 경로를 제공하는 터널. IPv6은 수동으로 구성 가능한 터널 및 자동 6to4 터널을 지원합니다.

경계 라우터 네트워크 가장자리에서 로컬 네트워크 외부 끝점에 IPv6 터널의 한쪽 끝점을 제공하는 라우터. 이 라우터는 내부 네트워크에 대해 하나 이상의 IPv6 인터페이스를 포함해야 합니다. 외부 네트워크에 대해 라우터는 IPv6 인터페이스 또는 IPv4 인터페이스를 포함할 수 있습니다.

IPv6 주소 지정 개요

IPv6 주소는 노드가 아닌 인터페이스에 지정됩니다. 노드는 두 개 이상의 인터페이스를 포함할 수 있습니다. 또한 인터페이스에 두 개 이상의 IPv6 주소를 지정할 수 있습니다.

주 – IPv6 주소 형식에 대한 전체 기술 정보를 보려면 RFC 2374, IPv6 Global Unicast Address Format (`http://www.ietf.org/rfc/rfc2374.txt?number=2374`)로 이동하십시오.

IPv6은 다음 세 가지 주소 유형을 정의합니다.

유니캐스트 개별 노드의 인터페이스를 식별합니다.

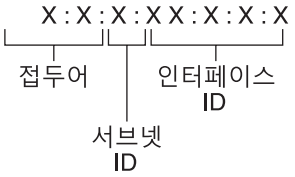
멀티캐스트 일반적으로 서로 다른 노드의 인터페이스 그룹을 식별합니다. 멀티캐스트 주소로 전송되는 패킷은 해당 **멀티캐스트 그룹**의 모든 멤버로 이동합니다.

애니캐스트 일반적으로 서로 다른 노드의 인터페이스 그룹을 식별합니다. 애니캐스트 주소로 전송되는 패킷은 물리적으로 발신자와 가장 가까운 **애니캐스트 그룹** 멤버로 이동합니다.

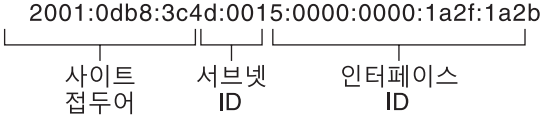
IPv6 주소의 부분

IPv6 주소는 길이가 128비트이며 8개의 16비트 필드로 구성되며, 각 필드는 콜론으로 연결되어 있습니다. IPv4 주소의 점으로 구분된 10진수 표기와 달리 각 필드는 16진수 숫자를 포함해야 합니다. 다음 그림에서 x는 16진수 숫자를 나타냅니다.

그림 3–2 기본 IPv6 주소 형식



예:



가장 왼쪽의 필드(48비트) 3개에는 **사이트 접두어**가 포함됩니다. 접두어는 일반적으로 ISP 또는 RIR(Regional Internet Registry)이 사이트에 할당하는 **공용 토폴로지**를 기술합니다.

다음 필드는 사용자(또는 다른 관리자)가 자신의 사이트에 할당하는 16비트 **서브넷** *ID*입니다. 서브넷 ID는 **사이트 토폴로지**라고도 부르는(사이트 내부의 토폴로지), **전용 토폴로지**를 기술합니다.

오른쪽 끝의 필드(64비트) 4개에는 **토큰**이라고도 부르는 **인터페이스** *ID*가 포함됩니다. 인터페이스 ID는 인터페이스의 MAC 주소로부터 자동으로 구성되거나 EUI-64 형식으로 수동으로 구성됩니다.

다시 그림 3–2의 주소를 살펴보십시오.

```
2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b
```

이 예에서는 IPv6 주소의 모든 128비트를 보여줍니다. 처음 48비트 `2001:0db8:3c4d`에는 공용 토폴로지를 나타내는 사이트 접두어가 포함됩니다. 다음 16비트 `0015`에는 사이트의 전용 토폴로지를 나타내는 서브넷 ID가 포함됩니다. 하위의 오른쪽 끝 64비트 `0000:0000:1a2f:1a2b`에는 인터페이스 ID가 포함됩니다.

IPv6 주소 축약

대부분의 IPv6 주소는 사용 가능한 128비트를 모두 차지하지 않습니다. 이에 따라 필드에 0이 채워지거나 0만 포함될 수 있습니다.

IPv6 주소 지정 아키텍처에서는 0이 연속된 16비트 필드를 나타내기 위해 두 개의 콜론(::)을 사용할 수 있습니다. 예를 들어, 인터페이스 ID의 연속된 0 필드 2개를 2개의 콜론으로 바꿔서 그림 3–2의 IPv6 주소를 축약할 수 있습니다. 그 결과는 `2001:0db8:3c4d:0015::1a2f:1a2b`와 같습니다. 다른 0 필드는 단일 0으로 표현할 수 있습니다. 또한 필드에서 앞에 오는 0을 생략할 수도 있습니다(예: `0db8`을 `db8`로 변경).

따라서 `2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b` 주소는 `2001:db8:3c4d:15::1a2f:1a2b`로 축약할 수 있습니다.

2개 콜론 표시 방법을 사용하면 IPv6 주소에서 0으로만 구성된 모든 연속된 필드를 바꿀 수 있습니다. 예를 들어, IPv6 주소 `2001:0db8:3c4d:0015:0000:d234::3eee:0000`은 `2001:db8:3c4d:15:0:d234:3eee::`로 줄일 수 있습니다.

IPv6의 접두어

IPv6 주소의 가장 왼쪽에 있는 필드는 IPv6 패킷의 경로 지정에 사용되는 접두어가 포함됩니다. IPv6 접두어의 형식은 다음과 같습니다.

prefix/length in bits

접두어 길이는 CIDR(Classless Inter-Domain Routing) 표기법으로 명시됩니다. CIDR 표기법은 접두어 길이(비트) 다음에 오는 주소 끝에 슬래시를 사용하는 것입니다. CIDR 형식의 IP 주소에 대한 자세한 내용은 56 페이지 “CIDR IPv4 주소 지정 체계 설계”를 참조하십시오.

IPv6 주소의 **사이트 접두어**는 IPv6 주소의 가장 왼쪽에 있는 비트를 최대 48비트까지 차지합니다. 예를 들어, IPv6 주소 `2001:db8:3c4d:0015:0000:0000:1a2f:1a2b/48`의 사이트 접두어는 가장 왼쪽의 48비트인 `2001:db8:3c4d`에 포함됩니다. 0이 압축된 다음 표현을 사용해서 이 접두어를 나타낼 수 있습니다.

```
2001:db8:3c4d::/48
```

주 – `2001:db8::/32` 접두어는 설명서 예에서 특별히 사용되는 특수한 IPv6 접두어입니다.

또한 라우터에 대한 네트워크의 내부 토폴로지를 정의하는 **서브넷 접두어**를 지정할 수 있습니다. IPv6 주소 예에는 다음과 같은 서브넷 접두어가 포함됩니다.

```
2001:db8:3c4d:15::/64
```

서브넷 접두어에는 항상 64비트가 포함됩니다. 이러한 비트에는 사이트 접두어의 48비트 및 서브넷 ID의 16비트가 포함됩니다.

다음 접두어는 특수 목적으로 예약되어 있습니다.

`2002::/16` 6to4 경로 지정 접두어가 뒤에 온다는 것을 나타냅니다.

`fe80::/10` 링크 로컬 주소가 뒤에 온다는 것을 나타냅니다.

`ff00::/8` 멀티캐스트 주소가 뒤에 온다는 것을 나타냅니다.

유니캐스트 주소

IPv6에는 두 가지 서로 다른 유니캐스트 주소 지정이 포함됩니다.

- 전역 유니캐스트 주소
- 링크 로컬 주소

유니캐스트 주소의 유형은 주소에서 접두어를 포함하는 가장 왼쪽(상위)의 연속된 비트로 결정됩니다.

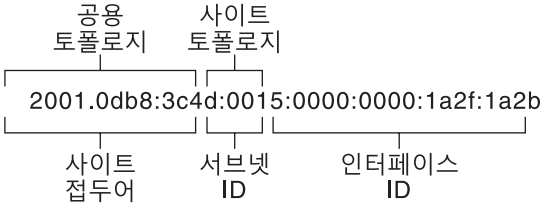
유니캐스트 주소 형식은 다음과 같은 계층으로 구성됩니다.

- 공용 토폴로지
- 사이트(전용) 토폴로지
- 인터페이스 ID

전역 유니캐스트 주소

전역 유니캐스트 주소는 인터넷에서 전역으로 고유합니다. 72 페이지 “IPv6의 접두어”에 표시된 IPv6 주소 예는 전역 유니캐스트 주소입니다. 다음 그림에서는 IPv6 주소의 각 부분과 비교하여 전역 유니캐스트 주소의 범위를 보여줍니다.

그림 3–3 전역 유니캐스트 주소의 부분



공용 토폴로지

사이트 접두어는 라우터에 대한 네트워크의 **공용 토폴로지**를 정의합니다. 회사의 사이트 접두어는 ISP 또는 RIR(Regional Internet Registry)에서 얻습니다.

사이트 토폴로지 및 IPv6 서브넷

IPv6에서 **서브넷** *ID*는 네트워크의 관리 서브넷을 정의하며 길이가 최대 16비트입니다. 서브넷 ID는 IPv6 네트워크 구성의 일부로 지정합니다. **서브넷 접두어**는 서브넷에 지정된 특정 링크를 지정하여 라우터에 대한 사이트 토폴로지를 정의합니다.

IPv6 서브넷은 개념적으로 IPv4 서브넷과 동일합니다. 일반적으로 각 서브넷은 단일 하드웨어 링크와 연결됩니다. 하지만 IPv6 서브넷 ID는 점으로 구분된 10진수 표기법이 아니라 16진수 표기법으로 표현됩니다.

인터페이스 ID

인터페이스 *ID*는 특정 노드의 인터페이스를 식별합니다. 인터페이스 ID는 서브넷 내에서 고유해야 합니다. IPv6 호스트는 Neighbor Discovery 프로토콜을 사용해서 고유 인터페이스 ID를 자동으로 생성할 수 있습니다. Neighbor Discovery는 호스트 인터페이스의 MAC 또는 EUI-64 주소를 기반으로 인터페이스 ID를 자동으로 생성합니다. 또한 IPv6 라우터 및 IPv6이 사용으로 설정된 서버에 대해 권장되는 인터페이스 ID를 수동으로 지정할 수도 있습니다. 수동 EUI-64 주소를 만드는 방법에 대한 자세한 내용은 RFC 3513 Internet Protocol Version 6 (IPv6) Addressing Architecture를 참조하십시오.

중간 단계의 전역 유니캐스트 주소

중간 단계를 위해 IPv6 프로토콜에는 IPv6 주소 내에 IPv4 주소를 포함시키는 기능이 있습니다. 이 유형의 IPv4 주소는 기존 IPv4 네트워크에 대한 IPv6 패킷의 터널링을 지원합니다. 중간 단계의 전역 유니캐스트 주소에 대한 한 가지 예는 6to4 주소입니다. 6to4 주소 지정에 대한 자세한 내용은 270 페이지 “6to4 자동 터널”을 참조하십시오.

링크 로컬 유니캐스트 주소

링크 로컬 유니캐스트 주소는 로컬 네트워크 링크에서만 사용할 수 있습니다. 링크 로컬 주소는 기업 외부에서 유효하지 않으며 인식되지도 않습니다. 다음 예에서는 링크 로컬 주소의 형식을 보여줍니다.

예 3–1 링크 로컬 유니캐스트 주소의 부분

예: fe80::123e:456d

예 3–1 링크 로컬 유니캐스트 주소의 부분 *(계속)*

링크 로컬 접두어의 형식은 다음과 같습니다.

`fe80::`*interface-ID*`/10`

다음은 링크 로컬 주소의 예입니다.

```
fe80::23a1:b152
```

`fe80` 10비트 이진 접두어 1111111010의 16진수 표현. 이 접두어는 IPv6 주소 유형을 링크 로컬로 식별합니다.

interface-ID 일반적으로 48비트 MAC 주소로부터 파생되는 인터페이스의 16진수 주소.

Oracle Solaris 설치 중 IPv6을 사용으로 설정하면 로컬 시스템에서 최하위 번호 인터페이스가 링크 로컬 주소로 구성됩니다. 로컬 링크에서 이 노드를 다른 노드와 식별하려면 각 인터페이스에 링크 로컬 주소가 한 개 이상 필요합니다. 따라서 노드의 추가 인터페이스에 대해 링크 로컬 주소를 수동으로 구성해야 합니다. 구성 후에는 노드가 자동 주소 구성 및 Neighbor Discovery를 위해 해당 링크 로컬 주소를 사용합니다.

멀티캐스트 주소

IPv6은 멀티캐스트 주소 사용을 지원합니다. 멀티캐스트 주소는 일반적으로 서로 다른 노드에 있는 인터페이스 그룹인 **멀티캐스트 그룹**을 식별합니다. 인터페이스는 여러 개의 멀티캐스트 그룹에 속할 수 있습니다. IPv6 주소의 처음 16비트가 `ff00`*n*이면, 이 주소가 멀티캐스트 주소입니다.

멀티캐스트 주소는 멀티캐스트 그룹의 멤버로 정의된 모든 인터페이스에 정보 또는 서비스를 전송하는 데 사용됩니다. 예를 들어, 멀티캐스트 주소의 한 가지 사용 방법은 로컬 링크의 모든 IPv6 노드와 통신하는 것입니다.

인터페이스의 IPv6 유니캐스트 주소를 만들 때는 커널이 해당 인터페이스를 자동으로 특정 멀티캐스트 그룹의 멤버로 지정합니다. 예를 들어, 커널은 Neighbor Discovery 프로토콜이 연결 가능성을 검색하기 위해 사용하는 Solicited Node 멀티캐스트 그룹의 멤버로 각 노드를 지정합니다. 커널은 또한 노드를 All-Nodes 또는 All Routers 멀티캐스트 그룹의 멤버로 자동으로 지정합니다.

멀티캐스트 주소에 대한 자세한 내용은 241 페이지 "IPv6 멀티캐스트 주소 세부 정보"를 참조하십시오. 기술 정보는 멀티캐스트 주소 형식을 설명하는 RFC 3306, Unicast-Prefix-based IPv6 Multicast Addresses (`ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt`)를 참조하십시오. 멀티캐스트 주소 및 그룹의 올바른 사용에 대한 자세한 내용은 RFC 3307, Allocation Guidelines for IPv6 Multicast Addresses (`ftp://ftp.rfc-editor.org/in-notes/rfc3307.txt`)를 참조하십시오.

애니캐스트 주소 및 그룹

IPv6 애니캐스트 주소는 서로 다른 IPv6 노드의 인터페이스 그룹을 식별합니다. 각 인터페이스 그룹은 **애니캐스트 그룹**이라고 합니다. 패킷이 애니캐스트 주소로 전송되면 발신자와 물리적으로 가장 가까운 애니캐스트 그룹 멤버가 패킷을 수신합니다.

주 – IPv6의 Oracle Solaris 구현에서는 애니캐스트 주소와 그룹 생성을 지원하지 않습니다. 하지만 Oracle Solaris IPv6 노드는 패킷을 애니캐스트 주소로 전송할 수 있습니다. 자세한 내용은 272 페이지 "6to4 릴레이 라우터에 대한 터널 고려 사항"을 참조하십시오.

IPv6 Neighbor Discovery 프로토콜 개요

IPv6에는 인접 노드 간 상호 작용을 처리하는 수단으로 메시징을 사용하는 Neighbor Discovery 프로토콜이 도입되었습니다. **인접 노드**는 동일 링크에 있는 IPv6 노드입니다. 예를 들어, 노드는 Neighbor Discovery 관련 메시지를 실행하여 인접한 링크 로컬 주소를 확인할 수 있습니다.

Neighbor Discovery는 IPv6 로컬 링크에서 다음과 같은 주요 작업을 제어합니다.

- **라우터 검색** – 호스트가 로컬 링크에서 라우터를 찾는 것을 도와줍니다.
- **주소 자동 구성** – 노드가 해당 인터페이스에 대해 IPv6 주소를 자동으로 구성할 수 있게 해줍니다.
- **접두어 검색** – 노드가 링크에 할당된 알려진 서브넷 접두어를 검색할 수 있게 해줍니다. 노드는 접두어를 사용해서 로컬 링크에 있는 대상과 라우터를 통해서만 연결할 수 있는 대상을 구분합니다.
- **주소 확인** – 노드가 대상의 IP 주소만 제공된 상태로 인접 노드의 링크 로컬 주소를 확인할 수 있도록 도와줍니다.
- **다음 홉 확인** – 알고리즘을 사용해서 로컬 링크 외부의 한 홉으로 패킷 수신자의 IP 주소를 확인합니다. 다음 홉은 라우터 또는 대상 노드일 수 있습니다.
- **이웃 연결 불가 감지** – 노드가 이웃에 더 이상 연결할 수 없는지 여부를 확인할 수 있도록 도와줍니다. 라우터 및 호스트 모두 주소 확인을 반복할 수 있습니다.

- **중복 주소 감지** – 노드가 사용하려는 주소가 아직 사용 중이 아닌지 확인할 수 있게 해줍니다.
- **재지정** – 특정 대상에 연결하기 위해 사용할 더 효과적인 첫번째 홉 노드를 라우터가 호스트에 알릴 수 있게 해줍니다.

Neighbor Discovery는 한 링크에서 노드 간 통신을 위해 다음과 같은 ICMP 메시지 유형을 사용합니다.

- 라우터 요청
- 라우터 알림
- 이웃 요청
- 이웃 알림
- 재지정

Neighbor Discovery 메시지 및 기타 Neighbor Discovery 프로토콜 항목에 대한 자세한 내용은 259 페이지 “IPv6 Neighbor Discovery 프로토콜”을 참조하십시오. Neighbor Discovery에 대한 기술 정보는 RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (`http://www.ietf.org/rfc/rfc2461.txt?number=2461`)을 참조하십시오.

IPv6 주소 자동 구성

IPv6의 중요 기능 중 하나는 호스트가 인터페이스를 자동 구성하는 기능입니다. Neighbor Discovery를 통해 호스트는 로컬 링크에서 IPv6 라우터를 찾고 사이트 접두어를 요청합니다. 호스트는 자동 구성 프로세스 중에 다음을 수행합니다.

- 각 인터페이스에 대한 링크 로컬 주소를 만듭니다. 이 작업에는 링크에 라우터가 필요하지 않습니다.
- 링크에서 주소의 고유성을 확인합니다. 이 작업에는 링크에 라우터가 필요하지 않습니다.
- Stateless 방식, Stateful 방식 또는 두 방식 모두를 통해 전역 주소를 가져와야 하는지 여부를 확인합니다. (이 작업에는 링크에 라우터가 필요합니다.)

Stateless 자동 구성 개요

Stateless 자동 구성을 위해서는 수동 호스트 구성이 필요하지 않고, 최소한의 라우터 구성(있는 경우)이 필요하며, 추가 서버가 필요하지 않습니다. Stateless 방식에서는 호스트가 고유한 주소를 생성할 수 있습니다. Stateless 방식에는 로컬 정보뿐만 아니라 주소를 생성하기 위해 라우터가 알리는 비로컬 정보도 사용됩니다.

인터페이스에 대한 임시 주소(자동 구성됨)를 구현할 수 있습니다. 호스트에서 하나 이상의 인터페이스에 대해 임시 주소 토큰을 사용으로 설정합니다. 그러나 자동 구성된 표준 IPv6 주소와 달리 임시 주소는 사이트 접두어와 무작위로 생성된 64비트 숫자로

구성됩니다. 이 무작위 숫자가 IPv6 주소의 인터페이스 ID 부분이 됩니다. 임시 주소를 사용할 경우 링크 로컬 주소가 인터페이스 ID로 생성되지 않습니다.

라우터는 링크에 지정된 모든 접두어를 알립니다. IPv6 호스트는 Neighbor Discovery를 사용하여 로컬 라우터에서 서브넷 접두어를 가져옵니다. 호스트는 서브넷 접두어를 인터페이스의 MAC 주소로부터 생성되는 인터페이스 ID와 조합하여 IPv6 주소를 자동으로 만듭니다. 라우터가 없으면 호스트가 로컬-링크 주소만 생성할 수 있습니다. 링크 로컬 주소는 동일 링크의 노드와 통신하는 데에만 사용할 수 있습니다.

주 – 서버의 IPv6 주소를 만들 때는 Stateless 자동 구성을 사용하지 마십시오. 호스트는 자동 구성 중 하드웨어 관련 정보를 기반으로 인터페이스 ID를 자동으로 생성합니다. 기존 인터페이스를 새 인터페이스로 스왑할 경우, 현재 인터페이스 ID가 잘못될 수 있습니다.

IPv6 터널 개요

대부분의 기업에서는 기존 IPv4 네트워크에 대한 IPv6 도입을 단계에 따라 점진적으로 수행해야 합니다. Oracle Solaris 이중 스택 네트워크 환경은 IPv4 및 IPv6 기능을 모두 지원합니다. 대부분의 네트워크에서는 IPv4 프로토콜을 사용하기 때문에 IPv6 네트워크에는 현재 경계 외부와 통신하는 방법이 필요합니다. IPv6 네트워크는 이 목적을 위해 터널을 사용합니다.

대부분의 IPv6 터널링 시나리오에서 아웃바운드 IPv6 패킷은 IPv4 패킷 내에 캡슐화됩니다. IPv6 네트워크의 경계 라우터는 다양한 IPv4 네트워크를 경유하여 대상 IPv6 네트워크의 경계 라우터에 도달하는 지점 간 터널을 설정합니다. 패킷은 터널을 통해 대상 네트워크의 경계 라우터로 전송되며, 여기서 패킷의 캡슐화가 해제됩니다. 그러면 라우터가 개별 IPv6 패킷을 대상 노드로 전달합니다.

Oracle Solaris IPv6 구현에는 다음과 같은 시나리오가 지원됩니다.

- IPv4 네트워크를 통해 두 개의 IPv6 네트워크 사이에 수동으로 구성된 터널. IPv4 네트워크는 기업 내부의 로컬 네트워크 또는 인터넷일 수 있습니다.
- 일반적으로 기업 내부에서 IPv6 네트워크를 통해 두 개의 IPv4 네트워크 사이에 수동으로 구성된 터널.
- 기업 내부의 IPv4 네트워크를 통해 또는 인터넷을 통해 두 개의 IPv6 네트워크 사이에 동적으로 구성된 자동 6to4 터널.

IPv6 터널에 대한 자세한 내용은 266 페이지 “IPv6 터널”을 참조하십시오. IPv4- IPv4 터널 및 VPN에 대한 자세한 내용은 469 페이지 “VPN(Virtual Private Networks) 및 IPsec”를 참조하십시오.

◆ ◆ ◆ 4 장 4

IPv6 네트워크 계획(작업)

새 네트워크 또는 기존 네트워크에 IPv6을 배치하려면 상당한 계획 작업이 필요합니다. 이 장에는 사이트에 IPv6을 구성하기 위해 먼저 수행해야 하는 계획 작업이 포함되어 있습니다. 기존 네트워크의 경우 IPv6 배치는 단계에 따라 점진적으로 진행해야 합니다. 이 장의 항목에서는 IPv4 전용 네트워크에서 IPv6을 단계별로 진행할 수 있도록 도와줍니다.

이 장에서는 다음 항목을 다룹니다.

IPv6 개념에 대한 소개는 3 장, "IPv6 소개(개요)"를 참조하십시오. 자세한 내용은 11 장, "IPv6 세부 개요(참조)"를 참조하십시오.

IPv6 계획(작업 맵)

다음 작업 맵의 작업을 순서에 따라 완료해서 IPv6 배치에 필요한 계획 작업을 수행하십시오.

다음 테이블에는 IPv6 네트워크 구성을 위한 서로 다른 작업이 나열되어 있습니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

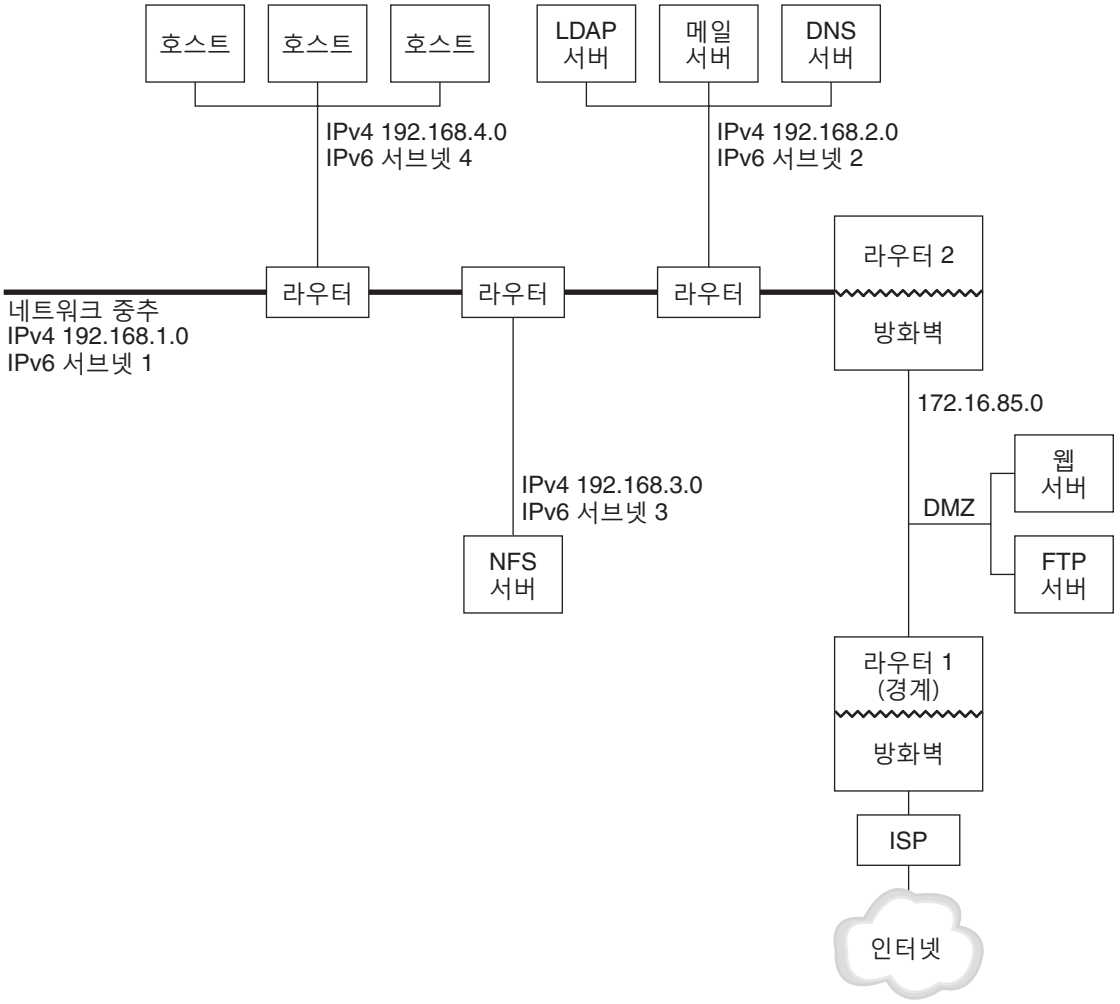
작업	설명	수행 방법
1. IPv6을 지원하도록 하드웨어를 준비합니다.	하드웨어를 IPv6으로 업그레이드할 수 있는지 확인합니다.	82 페이지 "IPv6 지원을 위한 네트워크 토폴로지 준비"

작업	설명	수행 방법
2. IPv6을 지원하는 ISP를 얻습니다.	현재 ISP에서 IPv6을 지원하는지 확인합니다. 그렇지 않으면 IPv6을 지원할 수 있는 ISP를 찾습니다. IPv6 통신용과 IPv4 통신용으로 두 개의 ISP를 사용할 수 있습니다.	
3. IPv6에서 응용 프로그램을 사용할 수 있는지 확인합니다.	IPv6 환경에서 응용 프로그램을 실행할 수 있는지 확인합니다.	84 페이지 “IPv6을 지원하도록 네트워크 서비스를 준비하는 방법”
4. 사이트 접두어를 가져옵니다.	ISP 또는 가장 가까운 RIR로부터 사이트에 대한 48비트 사이트 접두어를 얻습니다.	86 페이지 “사이트 접두어 획득”
5. 서브넷 주소 지정 계획을 만듭니다.	네트워크의 여러 노드에서 IPv6을 구성할 수 있으려면 먼저 전반적인 IPv6 네트워크 토폴로지 및 주소 지정 체계를 계획해야 합니다.	87 페이지 “서브넷 번호 지정 체계 만들기”
6. 터널 사용 계획을 설계합니다.	다른 서브넷 또는 외부 네트워크에 대한 터널을 실행할 라우터를 결정합니다.	85 페이지 “네트워크 토폴로지의 터널 계획”
7. 네트워크의 엔티티에 대한 주소 지정 계획을 만듭니다.	IPv6을 구성하기 전에 서버, 라우터 및 호스트에 대한 계획을 세워야 합니다.	87 페이지 “노드에 대한 IPv6 주소 지정 계획 만들기”
8. IPv6 보안 정책을 개발합니다.	IPv6 보안 정책을 개발하면서 IP 필터, IP 주소 아키텍처(IPsec), IKE(Internet Key Exchange) 및 기타 Oracle Solaris 보안 기능을 조사합니다.	제4부
9. (선택 사항) DMZ를 설정합니다.	보안을 위해 IPv6을 구성하기 전에 DMZ 및 해당 엔티티에 대한 주소 지정 계획이 필요합니다.	86 페이지 “IPv6 구현에 대한 보안 고려 사항”
10. 노드가 IPv6을 지원하도록 설정합니다.	모든 라우터 및 호스트에서 IPv6을 구성합니다.	165 페이지 “IPv6 라우터 구성(작업 맵)”
11. 네트워크 서비스를 설정합니다.	기존 서버가 IPv6을 지원할 수 있는지 확인합니다.	189 페이지 “주요 TCP/IP 관리 작업(작업 맵)”
12. IPv6 지원을 위해 이름 서버를 업데이트합니다.	DNS, NIS 및 LDAP 서버가 새로운 IPv6 주소로 업데이트되었는지 확인합니다.	184 페이지 “IPv6용 이름 서비스 지원 구성”

IPv6 네트워크 토폴로지 시나리오

이 장 전체의 작업에서는 일반적인 엔터프라이즈 네트워크에서 IPv6 서비스를 계획하는 방법에 대해 설명합니다. 다음 그림에서는 이 장 전체에서 참조하는 네트워크를 보여줍니다. 사용자가 제안하는 IPv6 네트워크에는 이 그림에 설명된 네트워크 링크가 일부 또는 모두 포함될 수 있습니다.

그림 4–1 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 4–1에서 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.

IPv6을 지원하도록 기존 네트워크 준비

주 – Oracle Solaris 듀얼 프로토콜 스택은 동시 IPv4 및 IPv6 작업을 지원합니다. 네트워크에 IPv6을 배치하는 중에 그리고 배치한 후에 IPv4 관련 작업을 성공적으로 수행할 수 있습니다.

IPv6에는 기존 네트워크에 대한 추가 기능이 포함됩니다. 따라서 IPv6을 처음 배치할 때는 IPv4로 작동하는 작업이 중단되지 않는지 확인해야 합니다. 이 섹션에서 다루는 주제에서는 단계별 방식으로 기존 네트워크에 IPv6을 도입하는 방법에 대해 설명합니다.

IPv6 지원을 위한 네트워크 토폴로지 준비

IPv6 배치의 첫번째 단계는 IPv6을 지원할 수 있는 네트워크의 기존 엔티티를 평가하는 것입니다. 대부분의 경우에는 IPv6을 구현할 때 네트워크 토폴로지(회선, 라우터 및 호스트)를 변경되지 않은 상태로 유지할 수 있습니다. 하지만 네트워크 인터페이스에서 IPv6 주소를 실제로 구성하기 전에 IPv6에 대한 기존 하드웨어 및 응용 프로그램을 준비해야 할 수 있습니다.

네트워크에서 IPv6으로 업그레이드할 수 있는 하드웨어를 확인합니다. 예를 들어, 하드웨어의 다음 클래스와 관련하여 IPv6이 사용 가능한지 제조업체의 설명서를 확인하십시오.

- 라우터
- 방화벽
- 서버
- 스위치

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

일부 라우터 모델은 IPv6으로 업그레이드할 수 없습니다. 자세한 정보 및 임시해결책은 215 페이지 "IPv4 라우터를 IPv6으로 업그레이드할 수 없음"을 참조하십시오.

IPv6 지원을 위한 네트워크 서비스 준비

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

- `sendmail`
- NFS
- HTTP(Apache 2.x 또는 r 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 215 페이지 "IPv6으로 서비스 업그레이드 후 발생하는 문제"를 참조하십시오.

IPv6 지원을 위한 서버 준비

서버는 IPv6 호스트로 간주되기 때문에 기본적으로 해당 IPv6 주소가 Neighbor Discovery 프로토콜에서 자동으로 구성됩니다. 하지만 많은 서버에는 유지 관리 또는 교체를 위해 전환할 수 있는 여러 NIC(네트워크 인터페이스 카드)가 포함되어 있습니다. NIC를 교체할 때는 Neighbor Discovery가 해당 NIC에 대한 새 인터페이스 ID를 자동으로 생성합니다. 이러한 동작은 특정 서버에서 허용되지 않는 동작일 수 있습니다.

따라서 서버의 각 인터페이스에 대해 IPv6 주소의 인터페이스 ID 부분은 수동으로 구성해야 할 수 있습니다. 자세한 내용은 173 페이지 "사용자 지정 IPv6 토큰을 구성하는 방법"을 참조하십시오. 나중에 기존 NIC를 교체해야 하면 이미 구성된 IPv6 주소가 교체용 NIC에 적용됩니다.

▼ 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 86 페이지 “IPv6 구현에 대한 보안 고려 사항”을 참조하십시오.
- 메일

  DNS용 MX 레코드의 경우 메일 서버의 IPv6 주소를 추가합니다.
- DNS

  DNS 관련 고려 사항은 84 페이지 “IPv6을 지원하도록 DNS를 준비하는 방법”을 참조하십시오.
- IPQoS

  IPv4에 사용된 것과 동일한 Diffserv 정책을 호스트에 대해 사용합니다. 자세한 내용은 767 페이지 “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 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 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.**

네트워크 토폴로지의 터널 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 4–1는 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 구성을 위한 절차는 176 페이지 “IPv6 지원을 위한 터널 구성 작업(작업 맵)”을 참조하십시오. 터널과 관련된 개념 정보는 266 페이지 “IPv6 터널”을 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 19 장, “IP 보안 아키텍처(개요)”를 참조하십시오.
- IKE(Internet Key Exchange) 기능을 통해 IPv6 패킷에 대한 공개 키 인증을 사용할 수 있습니다. 자세한 내용은 22 장, “Internet Key Exchange(개요)”를 참조하십시오.

IPv6 주소 지정 계획 준비

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

- 86 페이지 “사이트 접두어 획득”
- 87 페이지 “IPv6 번호 지정 체계 만들기”

사이트 접두어 획득

IPv6을 구성하기 전에 사이트 접두어를 획득해야 합니다. 사이트 접두어는 IPv6 구현에서 모든 노드에 대한 IPv6 주소를 파생시키는 데 사용됩니다. 사이트 접두어에 대한 소개는 72 페이지 “IPv6의 접두어”를 참조하십시오.

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 215 페이지 “현재 ISP가 IPv6을 지원하지 않음”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.

서브넷 번호 지정 체계 만들기

기존 IPv4 서브넷을 해당 IPv6 서브넷에 매핑하여 번호 지정 체계를 시작하십시오. 예를 들어 그림 4–1에 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

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 4–1에 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 4–1에서 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.

◆ ◆ ◆ 5 장

5 TCP/IP 네트워크 서비스 구성 및 IPv4 주소 지정(작업)

TCP/IP 네트워크 관리는 두 단계로 구성됩니다. 첫번째 단계는 하드웨어를 조립하는 것입니다. 그런 다음 TCP/IP 프로토콜을 구현하는 서비스, 데몬 및 파일을 구성합니다.

이 장에서는 IPv4 주소 지정 및 서비스를 구현하는 네트워크에서 TCP/IP를 구성하는 방법에 대해 설명합니다.

주 – 이 장에서 설명되는 대부분의 작업은 IPv4 전용 및 IPv6 사용 네트워크에 모두 적용됩니다. 이 장에서는 두 주소 지정 형식 간에 구성 작업이 다른 경우 IPv4를 구성하는 단계에 대해 설명합니다. 이 장의 작업을 확인한 다음 7 장, “IPv6 네트워크 구성(작업)”에서 해당 IPv6 작업을 상호 참조하십시오.

이 장은 다음 정보를 포함합니다.

이 장의 새로운 내용

Solaris 10 8/07에서는 다음 내용이 변경되었습니다.

- `routeadm` 명령을 사용하는 대신 SMF(서비스 관리 기능)를 통해 경로 지정을 구성 및 관리할 수 있습니다. 지침은 105 페이지 “IPv4 네트워크에서의 패킷 전달 및 경로 지정” 및 `routeadm(1M)` 매뉴얼 페이지의 절차 및 예를 참조하십시오.
- `/etc/inet/ipnodes` 파일은 더 이상 사용되지 않습니다. 개별 절차에 설명된 대로 `/etc/inet/ipnodes`는 이전 Solaris 10 릴리스에서만 사용하십시오.

IPv4 네트워크를 구성하기 전에(작업 맵)

TCP/IP를 구성하기 전에 다음 표에 나열된 작업을 수행하십시오. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
1. 네트워크 토폴로지를 설계합니다.	네트워크의 물리적 레이아웃을 확인합니다.	62 페이지 “네트워크 토폴로지 개요” 및 109 페이지 “IPv4 자율 시스템 토폴로지”
2. ISP 또는 RIR(Regional Internet Registry)에서 네트워크 번호를 얻습니다.	외부와 통신할 수 있는 사이트의 시스템에서 사용으로 설정된 등록된 네트워크 번호를 얻습니다.	55 페이지 “IPv4 주소 지정 체계 설계”.
3. 네트워크에 대한 IPv4 주소 지정 체계를 계획합니다. 해당하는 경우 서브넷 주소 지정을 포함합니다.	네트워크 번호를 기반으로 주소 지정 체계를 계획합니다.	55 페이지 “IPv4 주소 지정 체계 설계”.
4. 네트워크 토폴로지에 따라 네트워크 하드웨어를 조립합니다. 하드웨어가 제대로 작동하는지 확인합니다.	네트워크 토폴로지 설계에서 파악한 시스템, 네트워크 매체, 라우터, 스위치, 허브 및 브리지를 설정합니다.	하드웨어 매뉴얼 및 62 페이지 “네트워크 토폴로지 개요”.
5. 네트워크의 모든 시스템에 IPv4 주소 및 호스트 이름을 지정합니다.	Oracle Solaris 설치 중 또는 설치 후에 적절한 파일에 IPv4 주소를 지정합니다.	55 페이지 “IPv4 주소 지정 체계 설계” 및 101 페이지 “IPv4 주소 및 기타 네트워크 구성 매개변수 변경 방법”
6. 해당하는 경우 네트워크 인터페이스 및 라우터에 필요한 구성 소프트웨어를 실행합니다.	라우터 및 멀티홈 호스트를 구성합니다.	라우터에 대한 자세한 내용은 61 페이지 “네트워크의 라우터 계획” 및 112 페이지 “IPv4 라우터 구성”을 참조하십시오.

작업	설명	수행 방법
7. 네트워크에서 사용하는 이름 서비스 또는 디렉토리 서비스가 NIS, LDAP, DNS 또는 로컬 파일인지 확인합니다.	선택한 이름 서비스 및/또는 디렉토리 서비스를 구성합니다.	**System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** .
8. 해당하는 경우 네트워크의 도메인 이름을 선택합니다.	네트워크의 도메인 이름을 선택하고 InterNIC에 등록합니다.	**System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**

호스트 구성 모드 결정

네트워크 관리자는 호스트와 라우터에서 실행할 TCP/IP를 구성합니다(해당하는 경우). 이러한 시스템을 구성하여 로컬 시스템의 파일 또는 네트워크의 다른 시스템에 있는 파일에서 구성 정보를 얻을 수 있습니다.

다음 구성 정보가 필요합니다.

- 각 시스템의 호스트 이름
- 각 시스템의 IP 주소
- 각 시스템이 속한 도메인 이름
- 기본 라우터
- 각 시스템의 네트워크에서 사용 중인 IPv4 넷마스크

로컬 파일에서 TCP/IP 구성 정보를 얻은 시스템은 **로컬 파일 모드**로 작동합니다. 원격 네트워크 서버에서 TCP/IP 구성 정보를 얻은 시스템은 **네트워크 클라이언트 모드**로 작동합니다.

로컬 파일 모드로 실행되는 시스템

로컬 파일 모드로 실행하려면 시스템에 TCP/IP 구성 파일의 로컬 복사본이 있어야 합니다. 이러한 파일에 대해서는 217 페이지 “TCP/IP 구성 파일”에서 설명합니다. 시스템에 자체 디스크가 있어야 하지만 이는 권장 사항일 뿐 필수 사항은 아닙니다.

대부분의 서버는 로컬 파일 모드로 실행됩니다. 이 요구 사항에는 다음 서버가 포함됩니다.

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

라우터도 로컬 파일 모드로 실행됩니다.

인쇄 서버로 단독 사용되는 시스템은 로컬 파일 모드로 실행될 필요가 없습니다. 개별 호스트가 로컬 파일 모드로 실행되어야 하는지 여부는 네트워크 규모에 따라 달라집니다.

최소 규모의 네트워크를 실행 중인 경우에는 개별 호스트에서 이러한 파일을 관리하는 데 필요한 작업량이 그리 많지 않습니다. 네트워크를 수백 대의 호스트로 구성한 경우 해당 네트워크를 여러 하위 관리 도메인으로 분리하더라도 관리 작업은 어려워질 수밖에 없습니다. 따라서 대규모 네트워크의 경우에는 로컬 파일 모드를 사용하는 것이 더 비효율적입니다. 그러나 라우터와 서버는 자체 조달이 가능해야 하므로 로컬 파일 모드로 구성되어야 합니다.

네트워크 구성 서버

네트워크 구성 서버는 네트워크 클라이언트 모드로 구성된 호스트에 TCP/IP 구성 정보를 제공하는 서버입니다. 이러한 서버는 3가지 부트 프로토콜을 지원합니다.

- RARP – RARP(Reverse Address Resolution Protocol)는 이더넷 주소(48비트)를 IPv4 주소(32비트)에 매핑하는 역순 ARP입니다. 네트워크 구성 서버에서 RARP를 실행하는 경우 네트워크 클라이언트 모드로 실행 중인 호스트는 서버에서 IP 주소 및 TCP/IP 구성 파일을 얻습니다. `in.rarpd` 데몬은 RARP 서비스를 사용으로 설정합니다. 자세한 내용은 `in.rarpd(1M)` 매뉴얼 페이지를 참조하십시오.
- TFTP – TFTP(Trivial File Transfer Protocol)는 원격 시스템 간에 파일을 전송하는 응용 프로그램입니다. `in.tftpd` 데몬은 TFTP 서비스를 실행하여 네트워크 구성 서버와 해당 네트워크 클라이언트 간에 파일 전송을 사용으로 설정합니다. 자세한 내용은 `in.tftpd(1M)` 매뉴얼 페이지를 참조하십시오.
- Bootparams – Bootparams 프로토콜은 네트워크에서 부트되는 클라이언트에 필요한 부트 매개변수를 제공합니다. `rpc.bootparamd` 데몬은 이러한 서비스를 실행합니다. 자세한 내용은 `bootparamd(1M)` 매뉴얼 페이지를 참조하십시오.

네트워크 구성 서버는 NFS 파일 서버로도 작동 가능합니다.

호스트를 네트워크 클라이언트로 구성 중인 경우에는 네트워크에서 하나 이상의 시스템을 네트워크 구성 서버로 구성해야 합니다. 네트워크가 서브넷에 연결된 경우에는 네트워크 클라이언트가 있는 각 서브넷에 대해 하나 이상의 네트워크 구성 서버가 있어야 합니다.

네트워크 클라이언트 시스템

네트워크 구성 서버에서 구성 정보를 얻는 호스트는 네트워크 클라이언트 모드로 작동합니다. 네트워크 클라이언트로 구성된 시스템에는 TCP/IP 구성 파일의 로컬 복사본이 필요하지 않습니다.

네트워크 클라이언트 모드를 사용하면 대규모 네트워크의 관리 작업이 간소화됩니다. 네트워크 클라이언트 모드는 각 호스트에서 수행하는 구성 작업 수를 최소화합니다. 네트워크 클라이언트 모드를 사용하면 네트워크의 모든 시스템에 동일한 구성 표준이 적용됩니다.

모든 컴퓨터 유형에 대해 네트워크 클라이언트 모드를 구성할 수 있습니다. 예를 들어, 독립형 시스템에서 네트워크 클라이언트 모드를 구성할 수 있습니다.

혼합 구성

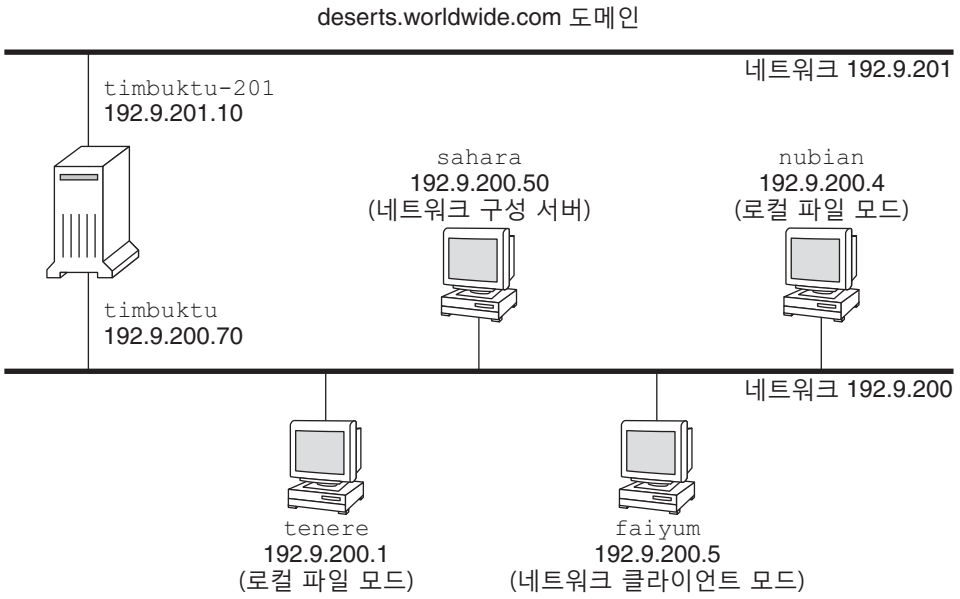
모두 로컬 파일 모드이거나 모두 네트워크 클라이언트 모드로 구성할 필요는 없습니다. 라우터와 서버는 항상 로컬 모드로 구성되어야 합니다. 호스트의 경우 로컬 파일 및 네트워크 클라이언트 모드를 혼합하여 사용할 수 있습니다.

IPv4 네트워크 토폴로지 시나리오

그림 5–1에서는 네트워크 번호가 192.9.200인 가상 네트워크의 호스트를 보여 줍니다. 네트워크에는 네트워크 구성 서버인 sahara만 있습니다. 호스트 tenere 및 nubian에는 각각 자체 디스크가 있으며 로컬 파일 모드로 실행됩니다. 호스트 faiyum에도 디스크가 있지만 이 시스템은 네트워크 클라이언트 모드로 작동합니다.

마지막으로 시스템 timbuktu는 라우터로 구성되어 있습니다. 시스템에는 두 네트워크 인터페이스가 있습니다. 첫번째 인터페이스의 이름은 timbuktu입니다. 이 인터페이스는 192.9.200 네트워크에 속합니다. 두번째 인터페이스의 이름은 timbuktu-201입니다. 이 인터페이스는 192.9.201 네트워크에 속합니다. 두 네트워크 모두 조직 도메인 deserts.worldwide.com에 속합니다. 도메인은 로컬 파일을 이름 서비스로 사용합니다.

그림 5–1 IPv4 네트워크 토폴로지 시나리오의 호스트



네트워크에 서브넷 추가(작업 맵)

서브넷을 사용하지 않는 네트워크를 서브넷을 사용하도록 변경하려면 다음 작업 맵의 작업을 수행합니다.

주 – 이 절의 정보는 IPv4 서브넷에만 적용됩니다. IPv6 서브넷 계획에 대한 자세한 내용은 82 페이지 “IPv6 지원을 위한 네트워크 토폴로지 준비” 및 87 페이지 “서브넷 번호 지정 체계 만들기”를 참조하십시오.

다음 표에서는 현재 네트워크에 서브넷을 추가하는 다양한 작업을 보여 줍니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
1. 네트워크 토폴로지에 서브넷이 필요한지 확인합니다.	서브넷의 라우터와 호스트 위치를 비롯한 새 서브넷 토폴로지를 결정합니다.	61 페이지 “네트워크의 라우터 계획”, 224 페이지 “서브넷이란?” 및 237 페이지 “네트워크 클래스”

작업	설명	수행 방법
2. 서브넷의 구성원이 될 시스템에 새 서브넷 번호와 IP 주소를 지정합니다.	Oracle Solaris 설치 중 또는 설치 후 `/etc/hostname.` *interface* 파일에서 새 서브넷 번호를 사용하는 IP 주소를 구성합니다.	51 페이지 "네트워크에 대한 IP 주소 지정 형식 결정"
3. 서브넷의 모든 잠재 시스템에서 서브넷의 네트워크 마스크를 구성합니다.	네트워크 클라이언트를 수동으로 구성하는 경우 `/etc/inet/netmasks` 파일을 수정합니다. 또는 Oracle Solaris 설치 프로그램에 넷마스크를 제공합니다.	224 페이지 "`netmasks` 데이터베이스" 및 224 페이지 "IPv4 주소에 대한 네트워크 마스크 만들기"
4. 서브넷에 있는 모든 시스템의 새 IP 주소로 네트워크 데이터베이스를 편집합니다.	모든 호스트의 `/etc/inet/hosts`(Solaris 10 11/06 및 이전 릴리스에서는 `/etc/inet/ipnodes`)를 수정하여 새 호스트 주소를 반영합니다.	219 페이지 "`hosts` 데이터베이스"
5. 모든 시스템을 재부트합니다.		

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
호스트를 로컬 파일 모드로 구성합니다.	`nodename`, `hostname`, `hosts`, `defaultdomain`, `defaultrouter` 및 `netmasks` 파일을 편집합니다.	96 페이지 "호스트를 로컬 파일 모드로 구성하는 방법"
네트워크 구성 서버를 설정합니다.	`in.tftp` 데몬을 실행하고 `hosts`, `ethers` 및 `bootparams` 파일을 편집합니다.	99 페이지 "네트워크 구성 서버 설정 방법"
호스트를 네트워크 클라이언트 모드로 구성합니다.	`hostname` 파일을 만들고 `hosts` 파일을 편집하고 `nodename` 및 `defaultdomain` 파일(있는 경우)을 삭제합니다.	100 페이지 "호스트를 네트워크 클라이언트 모드로 구성하는 방법"
네트워크 클라이언트에 대한 경로 지정 전략을 지정합니다.	호스트에서 정적 경로 지정 또는 동적 경로 지정을 사용할지 여부를 결정합니다.	123 페이지 "단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법" 및 125 페이지 "단일 인터페이스 호스트에서 동적 경로 지정을 사용으로 설정하는 방법".

작업	설명	수행 방법
기존 네트워크 구성을 수정합니다.	호스트 이름, IP 주소 및 설치 시 설정되거나 나중에 구성된 기타 매개변수를 변경합니다.	101 페이지 “IPv4 주소 및 기타 네트워크 구성 매개변수 변경 방법”

로컬 네트워크의 시스템 구성

네트워크 소프트웨어 설치는 운영 체제 소프트웨어 설치 시 수행됩니다. 이때 특정 IP 구성 매개변수는 해당 파일에 저장해야 부트 시 읽혀질 수 있습니다.

네트워크 구성 프로세스는 네트워크 구성 파일 만들기 또는 편집 작업으로 구성됩니다. 시스템 커널에서 구성 정보를 사용할 수 있는지는 조건부입니다. 즉, 가용성은 이러한 파일이 로컬에 저장되었는지(로컬 파일 모드) 또는 네트워크 구성 서버에서 얻는지(네트워크 클라이언트 모드) 여부에 따라 달라집니다.

네트워크 구성 중 제공되는 매개변수는 다음과 같습니다.

- 모든 시스템에 있는 각 네트워크 인터페이스의 IP 주소입니다.
- 네트워크에 있는 각 시스템의 호스트 이름입니다. 로컬 파일 또는 이름 서비스 데이터베이스에 호스트 이름을 입력할 수 있습니다.
- 시스템이 상주하는 NIS, LDAP 또는 DNS 도메인 이름입니다(해당하는 경우).
- 기본 라우터 주소입니다. 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. 기본 라우터에 대한 자세한 내용은 105 페이지 “IPv4 네트워크에서의 패킷 전달 및 경로 지정”을 참조하십시오. Oracle Solaris에서 지원하는 경로 지정 프로토콜 목록은 표 5–1을 참조하십시오.
- 서브넷 마스크입니다(서브넷이 있는 네트워크에서만 필요).

Oracle Solaris 설치 프로그램에서 둘 이상의 인터페이스를 시스템에서 발견한 경우 설치 중에 필요에 따라 추가 인터페이스를 구성할 수 있습니다. 자세한 내용은 **Oracle Solaris 10 1/13 설치 설명서: 기본 설치**을 참조하십시오.

이 장에서는 로컬 구성 파일 만들기 및 편집에 대한 정보를 설명합니다. 이름 서비스 데이터베이스 작업에 대한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**를 참조하십시오.

▼ 호스트를 로컬 파일 모드로 구성하는 방법

이 절차를 사용하여 로컬 파일 모드로 실행 중인 호스트에서 TCP/IP를 구성합니다.

Solaris 10 11/06 및 이후 릴리스에서 수동으로 인터페이스를 구성하는 단계는 138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”을 참조하십시오.

1 **기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **/etc 디렉토리로 변경합니다.**

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

Oracle Solaris 설치 중 시스템의 호스트 이름을 지정하면 호스트 이름이 /etc/nodename 파일에 입력됩니다. 노드 이름 항목이 시스템의 호스트 이름이 맞는지 확인합니다.

4 **시스템의 각 네트워크 인터페이스에 /etc/hostname.*interface* 파일이 있는지 확인합니다.**

/etc/hostname.*interface* 파일에 대한 파일 구문 및 기본 정보는 134 페이지 “물리적 인터페이스 관리를 위한 기본 사항”을 참조하십시오.

Oracle Solaris 설치 프로그램 설치 중 하나 이상의 인터페이스를 구성해야 합니다. 처음 구성한 인터페이스가 자동으로 **주 네트워크 인터페이스**가 됩니다. 설치 프로그램은 주 네트워크 인터페이스와 설치 시 선택적으로 구성한 다른 인터페이스에 대해 /etc/hostname.*interface* 파일을 만듭니다.

설치 중 추가 인터페이스를 구성한 경우 각 인터페이스에 해당 /etc/hostname. *interface* 파일이 있는지 확인하십시오. Oracle Solaris 설치 중 인터페이스를 반드시 둘 이상 구성할 필요는 없습니다. 그러나 나중에 인터페이스를 시스템에 추가하려면 수동으로 구성해야 합니다.

Solaris 10 11/06 및 이후 릴리스에서 수동으로 인터페이스를 구성하는 단계는 138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”을 참조하십시오.

5 **Solaris 10 11/06 및 이전 릴리스에서는 /etc/inet/ipnodes 파일의 항목이 최신 상태인지 확인합니다.**

Solaris 10 설치 프로그램은 /etc/inet/ipnodes 파일을 만듭니다. 이 파일에는 설치 중 구성된 모든 인터페이스의 노드 이름 및 IPv4 주소, IPv6 주소(해당하는 경우)가 포함되어 있습니다.

/etc/inet/ipnodes 파일에 입력할 때는 다음 형식을 사용하십시오.

IP-address node-name nicknames...

*nicknames*는 인식된 인터페이스의 추가 이름입니다.

6 **/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.

a. **/etc/inet/hosts의 기존 항목이 최신 상태인지 확인합니다.**

b. (선택 사항) 설치 후 로컬 호스트에 추가된 네트워크 인터페이스에 대한 IP 주소 및 해당 이름을 추가합니다.

c. (선택 사항) /usr 파일 시스템이 NFS 마운트된 시스템인 경우 IP 주소 또는 파일 서버의 주소를 추가합니다.

7 /etc/defaultdomain 파일에 호스트의 정규화된 도메인 이름을 입력합니다.

예를 들어, 호스트 tenere가 도메인 deserts.worldwide.com의 일부인 것으로 가정합니다. 이 경우 /etc/defaultdomain에 deserts.worldwide.com을 입력해야 합니다. 자세한 내용은 219 페이지 "/etc/defaultdomain 파일"을 참조하십시오.

8 /etc/defaultrouter 파일에 라우터의 이름을 입력합니다.

이 파일에 대한 자세한 내용은 219 페이지 "/etc/defaultrouter 파일"을 참조하십시오.

9 /etc/inet/hosts 파일에 기본 라우터 이름 및 해당 IP 주소를 입력합니다.

100 페이지 "호스트를 네트워크 클라이언트 모드로 구성하는 방법"에서 설명한 대로 추가 경로 지정 옵션을 사용할 수 있습니다. 이러한 옵션을 로컬 파일 모드 구성에 적용할 수 있습니다.

10 해당하는 경우 네트워크에 네트워크 마스크를 추가합니다.

- 호스트가 DHCP 서버에서 해당 IP 주소를 가져오는 경우 네트워크 마스크를 지정할 필요가 없습니다.
- 이 클라이언트와 동일한 네트워크에 NIS 서버를 설정한 경우 서버의 해당 데이터베이스에 netmask 정보를 추가할 수 있습니다.
- 다른 모든 조건에 대해서는 다음을 수행합니다.

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

다음 형식을 사용합니다.

```
network-number netmask
```

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 2–3에서 확인할 수 있습니다. 예를 들어, CIDR 네트워크 접두어 192.168.3.0/22를 표현하려면 다음을 사용합니다.

```
192.168.3.0 255.255.252.0
```

b. 로컬 파일이 먼저 검색되도록 /etc/nsswitch.conf에서 넷마스크의 조회 순서를 변경합니다.

```
netmasks:    files nis
```

11 **시스템을 재부트합니다.**

▼ 네트워크 구성 서버 설정 방법

설치 서버 및 부트 서버 설정에 대한 자세한 내용은 **Oracle Solaris 10 1/13 설치 설명서: 기본 설치**를 참조하십시오.

1 **기본 관리자 역할 또는 수퍼 유저로 로그인합니다**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 **잠재 네트워크 구성 서버의 루트(/) 디렉토리로 변경합니다.**

3 **/tftpboot 디렉토리를 만들어 in.tftpd 데몬을 실행합니다.**

```
# mkdir /tftpboot
```

이 명령은 시스템을 TFTP, bootparams 및 RARP 서버로 구성합니다.

4 **디렉토리에 대한 심볼릭 링크를 만듭니다.**

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

5 **/etc/inetd.conf 파일에서 tftp 라인을 사용으로 설정합니다.**

다음과 같은 항목이 있는지 확인합니다.

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

이 행은 in.tftpd가 /tftpboot에 있는 파일 이외의 다른 파일을 검색하지 않도록 합니다.

6 **hosts 데이터베이스를 편집합니다.**

네트워크에 있는 모든 클라이언트의 호스트 이름 및 IP 주소를 추가합니다.

7 **ethers 데이터베이스를 편집합니다.**

네트워크 클라이언트 모드로 실행 중인 네트워크의 모든 호스트에 대한 항목을 만듭니다.

8 **bootparams 데이터베이스를 편집합니다.**

232 페이지 "bootparams 데이터베이스"를 참조하십시오. 와일드카드 항목을 사용하거나 네트워크 클라이언트 모드로 실행되는 모든 호스트에 대한 항목을 만듭니다.

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

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

10 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 관리: 기본 관리**의 18 장, "서비스 관리(개요)"를 참조하십시오.

네트워크 클라이언트 구성

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

▼ 호스트를 네트워크 클라이언트 모드로 구성하는 방법

다음 절차를 수행하여 각 호스트가 네트워크 클라이언트 모드로 구성되도록 합니다.

1 기본 관리자 역할 또는 수퍼 유저로 로그인합니다

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 nodename 파일의 /etc 디렉토리를 검색합니다.

파일이 있으면 삭제합니다.

/etc/nodename을 삭제하면 시스템은 hostconfig 프로그램을 사용하여 네트워크 구성 서버에서 호스트 이름, 도메인 이름 및 라우터 주소를 얻습니다. 96 페이지 "로컬 네트워크의 시스템 구성"을 참조하십시오.

3 /etc/hostname.*interface* 파일이 없는 경우 이 파일을 만듭니다.

파일이 비어 있는지 확인합니다. /etc/hostname.*interface* 파일이 비어 있으면 시스템은 네트워크 구성 서버에서 IPv4 주소를 얻습니다.

4 **/etc/inet/hosts 파일에 localhost 이름과 루프백 네트워크 인터페이스의 IP 주소만 있는지 확인합니다.**

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

IPv4 루프백 인터페이스의 IP 주소는 127.0.0.1입니다.

자세한 내용은 220 페이지 "루프백 주소"를 참조하십시오. 파일에 로컬 호스트(주 네트워크 인터페이스)의 IP 주소와 호스트 이름은 없어야 합니다.

5 **/etc/defaultdomain 파일이 있는지 확인합니다.**

파일이 있으면 삭제합니다.

hostconfig 프로그램은 자동으로 도메인 이름을 설정합니다. hostconfig에서 설정한 도메인 이름을 바꾸려면 /etc/defaultdomain 파일에 대체 도메인 이름을 입력합니다.

6 **클라이언트 /etc/nsswitch.conf 파일의 검색 경로가 네트워크에 대한 이름 서비스 요구 사항을 반영하는지 확인합니다.**

▼ 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 138 페이지 "시스템 설치 후 물리적 인터페이스 구성 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. CIDR 표기법 소개는 52 페이지 "CIDR 형식의 IPv4 주소"를 참조하십시오.

1 **기본 관리자 역할 또는 수퍼 유저로 로그인합니다**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 Solaris 10 11/06 및 이전 릴리스에서는 /etc/inet/ipnodes 파일 또는 동등한 ipnodes 데이터베이스에서 IP 주소를 수정합니다.

시스템에 추가하는 각 IP 주소에 대해 다음 구문을 사용합니다.

IP-address host-name, nicknames
IP-address interface-name, nicknames

첫번째 항목에는 주 네트워크 인터페이스의 IP 주소와 시스템의 호스트 이름이 있어야 합니다. 호스트 이름의 별명을 추가할 수도 있습니다. 시스템에 다른 물리적 인터페이스를 추가하는 경우에는 /etc/inet/ipnodes에 이러한 인터페이스의 연결된 이름과 IP 주소 항목을 만듭니다.

3 시스템의 호스트 이름을 변경해야 할 경우에는 /etc/nodename 파일에서 호스트 이름 항목을 수정합니다.

4 IP 주소를 수정하고 해당하는 경우 /etc/inet/hosts 파일 또는 동등한 hosts 데이터베이스에서 호스트 이름을 수정합니다.

5 /etc/hostname.*interface* 파일에서 주 네트워크 인터페이스의 IP 주소를 수정합니다.

/etc/hostname*interface* 파일에서 다음 중 하나를 주 네트워크 인터페이스의 항목으로 사용할 수 있습니다.

- IPv4 주소(점으로 구분된 일반적인 십진수 형식)

  다음 구문을 사용하십시오.

  IPv4 address subnet mask

  넷마스크 항목은 옵션입니다. 지정하지 않으면 기본 넷마스크로 간주됩니다.

  다음은 예입니다.

  ```
  # vi hostname.eri0
  10.0.2.5 netmask 255.0.0.0
  ```

- 네트워크 구성에 적합한 경우 IPv4 주소(CIDR 표기법)

  IPv4 address/network prefix

  다음은 예입니다.

  ```
  # vi hostname.eri0
  10.0.2.5/8
  ```

  CIDR 접두어는 IPv4 주소에 해당하는 넷마스크를 지정합니다. 예를 들어, 위의 /8은 넷마스크 255.0.0.0을 나타냅니다.

- 호스트 이름

  /etc/hostname.*interface* 파일에서 시스템의 호스트 이름을 사용하려면 호스트 이름 및 연결된 IPv4 주소가 hosts 데이터베이스에도 있어야 합니다.

6 서브넷 마스크가 변경된 경우 다음 파일에서 서브넷 항목을 수정합니다.

- /etc/netmasks
- (선택 사항) /etc/hostname.*interface*

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

8 시스템을 재부트합니다.

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

예 5–1 재부트 시에도 유지될 수 있도록 IPv4 주소 및 기타 네트워크 매개변수 수정

이 예에서는 다른 서브넷으로 이동되는 시스템의 다음 네트워크 매개변수를 변경하는 방법을 보여 줍니다.

- 주 네트워크 인터페이스 eri0의 IP 주소가 10.0.0.14에서 192.168.55.14로 변경됩니다.
- 호스트 이름이 myhost에서 mynewhostname으로 변경됩니다.
- 넷마스크가 255.0.0.0에서 255.255.255.0으로 변경됩니다.
- 기본 라우터 주소가 192.168.55.200으로 변경됩니다.

시스템의 현재 상태를 확인합니다.

```
# hostname
myhost
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
```

그런 다음 해당 파일에서 시스템의 호스트 이름과 eri0의 IP 주소를 변경합니다.

```
# vi /etc/nodename
mynewhostname
```

Oracle Solaris 10 11/06 및 이전 Oracle Solaris 10 릴리스에서만 다음을 수행합니다.

```
# vi /etc/inet/ipnodes
192.168.55.14   mynewhostname      #moved system to 192.168.55 net

# vi /etc/inet/hosts
#
# Internet host table
#
127.0.0.1       localhost
192.168.55.14   mynewhostname        loghost
# vi /etc/hostname.eri0
192.168.55.14   netmask  255.255.255.0
```

마지막으로 기본 라우터의 넷마스크 및 IP 주소를 변경합니다.

```
# vi /etc/netmasks
...
192.168.55.0    255.255.255.0

# vi /etc/defaultrouter
192.168.55.200        #moved system to 192.168.55 net
#
```

이러한 사항을 변경한 후 시스템을 재부트합니다.

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

방금 설정한 구성이 재부트 후에도 유지되는지 확인합니다.

```
# hostname
mynewhostname
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.55.14 netmask ffffff00 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
```

예 5–2 현재 세션에 대해 IP 주소 및 호스트 이름 변경

이 예에서는 현재 세션에 대해서만 호스트 이름, 주 네트워크 인터페이스의 IP 주소 및 서브넷 마스크를 변경하는 방법을 보여 줍니다. 재부트할 경우 시스템은 이전 IP 주소 및 서브넷 마스크로 복원됩니다. 주 네트워크 인터페이스 `eri0`의 IP 주소가 `10.0.0.14`에서 `192.168.34.100`으로 변경됩니다.

```
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# ifconfig eri0 192.168.34.100 netmask 255.255.255.0 broadcast + up
# vi /etc/nodename
mynewhostname

# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.34.100 netmask ffffff00 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# hostname
mynewhostname
```

예 5–3 CIDR 표기법을 사용하여 현재 세션에 대해 IPv4 주소 변경

이 예에서는 CIDR 표기법을 사용하여 현재 세션에 대해서만 호스트 이름 및 IP 주소를 변경하는 방법을 보여 줍니다. 재부트할 경우 시스템은 이전 IP 주소 및 서브넷 마스크로 복원됩니다. 주 네트워크 인터페이스 `eri0`의 IP 주소가 `10.0.0.14`에서 `192.168.6.25/27`로 변경됩니다.

```
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# ifconfig eri0 192.168.6.25/27 broadcast + up
# vi /etc/nodename
mynewhostname
# ifconfig -a

lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.06.25 netmask fffffffe0 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
# hostname
mynewhostname
```

IPv4 주소에 CIDR 표기법을 사용할 때는 넷마스크를 지정할 필요가 없습니다. `ifconfig`는 네트워크 접두어 지정을 사용하여 넷마스크를 결정합니다. 예를 들어, `192.168.6.0/27` 네트워크의 경우 `ifconfig`는 넷마스크를 `ffffffe0`으로 설정합니다. 더 일반적인 `/24` 접두어 지정을 사용한 경우 넷마스크는 `ffffff00`이 됩니다. `/24` 접두어 지정을 사용하는 것은 새 IP 주소를 구성할 때 넷마스크 `255.255.255.0`을 `ifconfig`에 지정하는 것과 같습니다.

참조 주 네트워크 인터페이스 이외의 인터페이스에 대한 IP 주소를 변경하려면 **Oracle Solaris 관리: 기본 관리** 및 138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”을 참조하십시오.

IPv4 네트워크에서의 패킷 전달 및 경로 지정

이 절에서는 IPv4 네트워크에서 라우터 및 호스트의 전달과 경로 지정을 구성하는 방법을 보여주는 절차 및 예를 설명합니다.

패킷 전달은 네트워크의 시스템 간에 정보를 공유하는 기본 방법입니다. 패킷은 일반적으로 서로 다른 두 시스템에 있는 소스 인터페이스와 대상 인터페이스 간에 전달됩니다. 명령을 실행하거나 비로컬 인터페이스로 메시지를 전송할 때 시스템에서는 이러한 패킷을 로컬 네트워크로 전달합니다. 대상 IP 주소가 패킷 헤더에서 지정된 인터페이스는 로컬 네트워크에서 패킷을 검색합니다. 대상 주소가 로컬 네트워크에

있지 않은 경우 패킷이 다음 인접 네트워크 또는 **홉**으로 전달됩니다. 기본적으로 패킷 전달은 Oracle Solaris 설치 시 자동으로 구성됩니다.

경로 지정은 시스템에서 패킷이 전송될 대상을 결정하는 프로세스입니다. 시스템의 경로 지정 프로토콜은 로컬 네트워크의 다른 시스템을 "검색"합니다. 소스 시스템 및 대상 시스템이 동일한 로컬 네트워크에 있는 경우 시스템 간에 패킷이 전달되는 경로를 **직접 경로**라고 합니다. 패킷이 전달되는 대상 시스템이 소스 시스템에서 한 홉 이상 떨어져 있는 경우 소스 시스템과 대상 시스템 간의 경로를 **간접 경로**라고 합니다. 경로 지정 프로토콜은 대상 인터페이스에 대한 경로를 기억하고 시스템의 **경로 지정 테이블**에 있는 알려진 경로에 대한 데이터를 보관합니다.

라우터는 여러 물리적 인터페이스를 통해 라우터를 둘 이상의 로컬 네트워크에 연결하는 특수 구성된 시스템입니다. 따라서 경로 지정 프로토콜을 실행하는지 여부에 관계없이 라우터는 패킷을 홈 LAN 외부로 전달할 수 있습니다. 라우터에서 패킷을 전달하는 방법에 대한 자세한 내용은 61 페이지 "네트워크의 라우터 계획"을 참조하십시오.

경로 지정 프로토콜은 시스템의 경로 지정 작업을 처리하고 다른 호스트와 경로 지정 정보를 교환함으로써 원격 네트워크에 대한 알려진 경로를 유지 관리합니다. 라우터와 호스트 모두 경로 지정 프로토콜을 실행할 수 있습니다. 호스트의 경로 지정 프로토콜은 다른 라우터 및 호스트의 경로 지정 데몬과 통신합니다. 이러한 프로토콜은 호스트가 패킷 전달 대상을 결정하는 데 유용합니다. 네트워크 인터페이스가 사용으로 설정되면 시스템이 자동으로 경로 지정 데몬과 통신합니다. 이러한 데몬은 네트워크의 라우터를 모니터링하고 라우터 주소를 로컬 네트워크의 호스트로 알립니다. 일부 경로 지정 프로토콜은 경로 지정 성능을 측정하는 데 사용할 수 있는 통계를 유지 관리하기도 합니다. 패킷 전달과는 달리 Oracle Solaris 시스템에서는 경로 지정을 명시적으로 구성해야 합니다.

이 절에서는 IPv4 라우터 및 호스트에서 패킷 전달과 경로 지정을 관리하는 작업을 설명합니다. IPv6 사용 네트워크의 경로 지정에 대한 자세한 내용은 165 페이지 "IPv6 라우터 구성"을 참조하십시오.

Oracle Solaris에서 지원하는 경로 지정 프로토콜

경로 지정 프로토콜은 IGP(Interior Gateway Protocol), EGP(Exterior Gateway Protocol) 또는 두 가지의 혼합으로 분류됩니다. *IGP*는 일반적인 관리 방법으로 제어되는 네트워크에서 라우터 간에 경로 지정 정보를 교환합니다. 그림 5–3과 같이 네트워크 토폴로지에서는 라우터가 IGP를 실행하여 경로 지정 정보를 교환합니다. *EGP*를 사용으로 설정하면 로컬 인터네트워크를 외부 네트워크로 연결하는 라우터에서 외부 네트워크의 다른 라우터와 정보를 교환할 수 있습니다. 예를 들어, 회사 네트워크를 ISP와 연결하는 라우터는 EGP를 실행하여 ISP에 있는 상대 라우터와 경로 지정 정보를 교환합니다. BGP(Border Gateway Protocol)는 다른 조직과 IGP 간에 경로 지정 정보를 전달하는 데 많이 사용되는 EGP입니다.

다음 표에서는 Oracle Solaris 경로 지정 프로토콜에 대한 정보와 각 프로토콜의 관련 문서의 위치를 제공합니다.

표 5–1 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.	112 페이지 “IPv4 라우터 구성 방법”
ICMP(Internet Control Message Protocol) 라우터 검색	`in.routed`	호스트에서 네트워크의 라우터를 검색하는 데 사용됩니다.	123 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법” 및 125 페이지 “단일 인터페이스 호스트에서 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.	166 페이지 “IPv6 지원 라우터를 구성하는 방법”
ND(Neighbor Discovery) 프로토콜	`in.ndpd`	IPv6 라우터의 존재를 알리고 네트워크의 IPv6 호스트를 검색합니다.	159 페이지 “IPv6 인터페이스 구성”

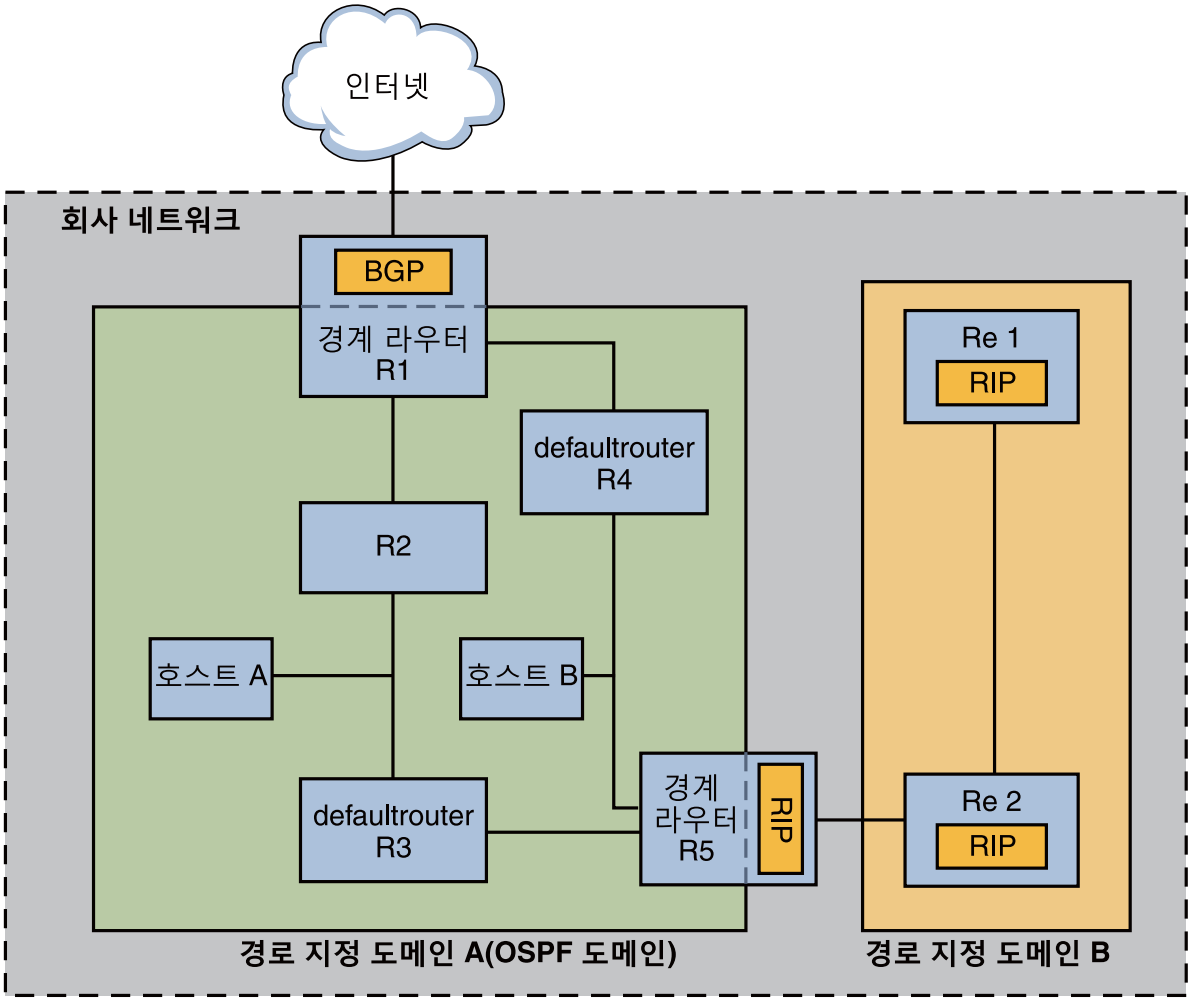
또한 Oracle Solaris에서는 오픈 소스 Quagga 경로 지정 프로토콜 제품군도 지원합니다. 이러한 프로토콜은 SFW 통합 디스크에서 사용할 수 있습니다. 단, 주 Oracle Solaris 배포에는 포함되어 있지 않습니다. 다음 표에서는 Quagga 프로토콜을 나열합니다.

표 5–2 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.

다음 그림에서는 Quagga 경로 지정 프로토콜을 사용하는 자율 시스템을 보여 줍니다.

그림 5–2 Quagga 프로토콜을 실행하는 회사 네트워크



그림에서는 두 개의 경로 지정 도메인 A 및 B로 구성된 회사 네트워크 자율 시스템을 보여 줍니다. **경로 지정 도메인**은 관리 목적 또는 단일 경로 지정 프로토콜을 사용하는 도메인이 있어서 일관된 경로 지정 정책을 적용한 인터네트워크입니다. 그림에서 두 도메인은 Quagga 프로토콜 제품군의 경로 지정 프로토콜을 실행합니다.

경로 지정 도메인 A는 단일 OSPF 도메인 ID로 관리되는 OSPF 도메인입니다. 이 도메인의 모든 시스템은 OSPF를 IGP로 실행합니다. 도메인 A에는 내부 호스트 및 라우터 외에도 두 개의 경계 라우터가 포함됩니다.

경계 라우터 R1은 회사 네트워크를 ISP에 연결하며 궁극적으로는 인터넷에 연결합니다. 회사 네트워크에서 외부와 통신할 수 있게 하기 위해 R1은 외부와 접해 있는 네트워크

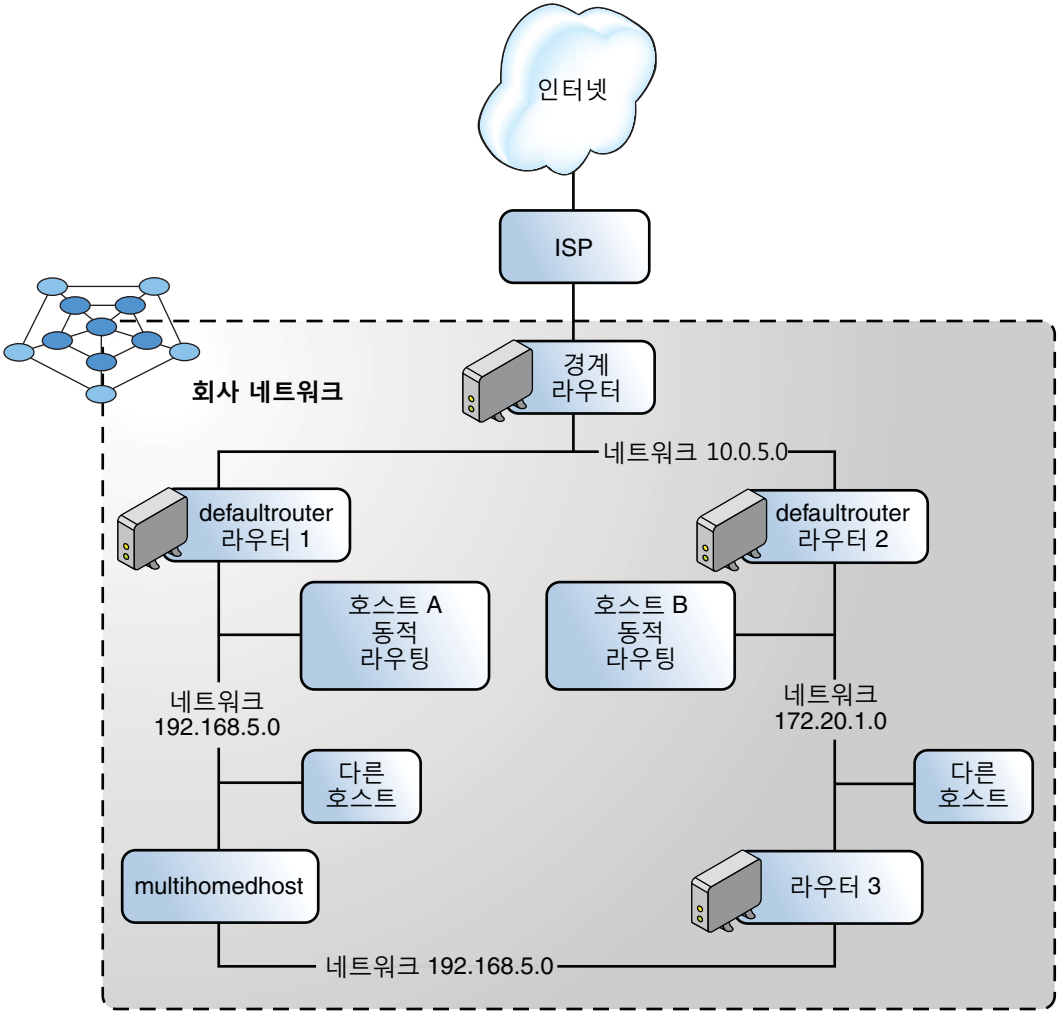
인터페이스를 통해 BGP를 실행합니다. 경계 라우터 R5는 도메인 A와 도메인 B를 연결합니다. 도메인 B의 모든 시스템은 IGP로서 RIP를 통해 관리됩니다. 따라서 경계 라우터 R5는 도메인 A 인터페이스에서는 OSPF, 도메인 B 인터페이스에서는 RIP를 실행해야 합니다.

Quagga 프로토콜에 대한 자세한 내용은 Quagga Routing Suite 웹 사이트(http://www.nongnu.org/quagga/index.html)를 참조하십시오.

IPv4 자율 시스템 토폴로지

일반적으로 라우터와 네트워크가 여러 개인 사이트에서는 네트워크 토폴로지를 단일 경로 지정 도메인 또는 **자율 시스템***(AS)*으로 관리합니다. 다음 그림에서는 소규모 AS로 간주되는 일반적인 네트워크 토폴로지를 보여 줍니다. 이 토폴로지는 이 절에서 설명하는 전체 예에서 참조됩니다.

그림 5–3 IPv4 라우터가 여러 개인 자율 시스템



그림에서는 3개의 로컬 네트워크(10.0.5.0, 172.20.1.0 및 192.168.5)로 구분된 AS를 보여 줍니다. 4개의 라우터는 패킷 전달 및 경로 지정 책임을 공유합니다. AS에는 다음 유형의 시스템이 포함됩니다.

- **경계 라우터**는 AS를 인터넷과 같은 외부 네트워크에 연결합니다. 경계 라우터는 로컬 AS에서 실행되는 IGP 외부의 네트워크와 상호 연결합니다. 경계 라우터는 BGP(Border Gateway Protocol)와 같은 EGP를 실행하여 외부 라우터(예: ISP의 라우터)와 정보를 교환합니다. 그림 5–3에서 경계 라우터의 인터페이스는 내부 네트워크 10.0.5.0, 고속 라우터 및 서비스 공급자에 연결합니다.

  경계 라우터 구성에 대한 자세한 내용은 오픈 소스 Quagga 설명서 (http://www.quagga.net/docs/docs-info.php#SEC72)에서 BGP 정보를 참조하십시오.

  BGP를 사용하여 AS를 인터넷에 연결하려는 경우 로케일에 적합한 인터넷 레지스트리에서 ASN(자율 시스템 번호)을 얻어야 합니다. ARIN(American Registry for Internet Numbers)과 같은 지역 레지스트리는 ASN을 얻는 방법에 대한 지침을 제공합니다. 예를 들어, ARIN Number Resource Policy Manual (https://www.arin.net/policy/nrpm.html#five)에는 미국 및 캐나다의 자율 시스템에 대한 ASN을 얻는 방법이 나와 있습니다. 또는 ISP에서 사용자 대신 ASN을 얻어 줄 수도 있습니다.

- **기본 라우터**는 로컬 네트워크의 모든 시스템에 대한 경로 지정 정보를 유지 관리합니다. 이러한 라우터는 일반적으로 RIP와 같은 IGP를 실행합니다. 그림 5–3에서 라우터 1의 인터페이스는 내부 네트워크 10.0.5.0과 내부 네트워크 192.168.5에 연결됩니다. 라우터 1은 192.168.5의 기본 라우터입니다. 라우터 1은 192.168.5의 모든 시스템에 대한 경로 지정 정보와 경계 라우터와 같은 다른 라우터에 대한 경로를 유지 관리합니다. 라우터 2의 인터페이스는 내부 네트워크 10.0.5.0과 내부 네트워크 172.20.1에 연결됩니다.

  기본 라우터 구성의 예는 예 5–4를 참조하십시오.

- **패킷 전달 라우터**는 패킷을 전달하지만 경로 지정 프로토콜을 실행하지 않습니다. 이 라우터 유형은 단일 네트워크에 연결된 해당 인터페이스 중 하나에서 패킷을 수신합니다. 그런 다음 이러한 패킷은 라우터의 다른 인터페이스를 통해 다른 로컬 네트워크로 전달됩니다. 그림 5–3에서 라우터 3은 네트워크 172.20.1과 192.168.5에 연결된 패킷 전달 라우터입니다.

- **멀티홈 호스트**에는 동일한 네트워크 세그먼트로 연결된 둘 이상의 인터페이스가 있습니다. 멀티홈 호스트는 패킷을 전달할 수 있으며 이는 Oracle Solaris를 실행하는 모든 시스템에서 기본값입니다. 그림 5–3에서는 두 인터페이스가 네트워크 192.168.5에 연결된 멀티홈 호스트를 보여 줍니다. 멀티홈 호스트 구성의 예는 예 5–6을 참조하십시오.

- **단일 인터페이스 호스트**는 패킷 전달뿐 아니라 중요한 구성 정보를 수신할 때도 로컬 라우터를 사용합니다. 그림 5–3에는 동적 경로 지정을 구현하는 192.168.5 네트워크의 호스트 A와 정적 라우팅을 구현하는 172.20.1 네트워크의 호스트 B가 있습니다. 동적 경로 지정을 실행하도록 호스트를 구성하려면 125 페이지 “단일 인터페이스 호스트에서 동적 경로 지정을 사용으로 설정하는 방법”을 참조하십시오. 정적 경로 지정을 실행하도록 호스트를 구성하려면 123 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법”을 참조하십시오.

IPv4 라우터 구성

이 절에서는 IPv4 라우터를 구성하는 절차 및 예를 설명합니다. IPv6 사용 라우터를 구성하려면 166 페이지 "IPv6 지원 라우터를 구성하는 방법"을 참조하십시오.

라우터는 둘 이상의 네트워크 간에 인터페이스를 제공하므로 라우터의 각 물리적 네트워크 인터페이스에 대해 고유 이름 및 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. **System Administration Guide: Network Services**의 "Planning a Dial-up PPP Link"에 설명된 대로 시스템이 PPP 링크에서 하나의 끝점으로 사용되는 경우 단일 인터페이스 시스템을 라우터로 구성할 수 있습니다.

주 – Oracle Solaris 시스템 설치 중 라우터의 모든 인터페이스를 구성할 수 있습니다. 자세한 내용은 **Oracle Solaris 10 1/13 설치 설명서: 기본 설치**를 참조하십시오.

▼ IPv4 라우터 구성 방법

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

시작하기 전에 네트워크에 라우터를 물리적으로 설치한 다음 96 페이지 "호스트를 로컬 파일 모드로 구성하는 방법"에 설명된 대로 라우터가 로컬 파일 모드로 작동하도록 구성합니다. 이 구성은 네트워크 구성 서버의 작동이 중지된 경우 라우터가 부트되도록 합니다.

1 라우터로 구성할 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 Solaris 10 1/06 릴리스부터는 `dladm show-link` 명령을 사용하여 라우터에 물리적으로 설치된 인터페이스를 확인합니다.

```
# dladm show-link
```

`dladm show-link` 명령의 다음 출력 예는 인터페이스 4개로 구성된 `qfe` NIC와 `bge` 인터페이스 2개가 시스템에서 물리적으로 사용되고 있음을 나타냅니다.

```
qfe0            type: legacy    mtu: 1500       device: qfe0
qfe1            type: legacy    mtu: 1500       device: qfe1
qfe2            type: legacy    mtu: 1500       device: qfe0
qfe3            type: legacy    mtu: 1500       device: qfe1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
```

3 **설치 중 구성되고 연결된 라우터의 인터페이스를 검토합니다.**

```
# ifconfig -a
```

`ifconfig -a` 명령의 다음 출력 예는 설치 중 구성된 인터페이스 `qfe0`을 보여 줍니다. 이 인터페이스는 `172.16.0.0` 네트워크에 속합니다. `qfe` NIC의 나머지 인터페이스인 `qfe1` - `qfe3` 및 `bge` 인터페이스는 구성되지 않았습니다.

```
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.16.26.232 netmask ffff0000 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:15
```

4 **다른 인터페이스를 구성 및 연결합니다.**

```
# ifconfig interface plumb
```

예를 들어, `qfe1`의 경우 다음을 입력합니다.

```
# ifconfig qfe1 plumb
```

주 – `ifconfig` 명령을 통해 명시적으로 구성된 인터페이스는 재부트 시 유지되지 않습니다.

5 **인터페이스에 IPv4 주소 및 넷마스크를 지정합니다.**

주의 – IPv4 라우터를 구성하여 DHCP를 통해 해당 IP 주소를 수신할 수 있지만 이 작업은 매우 숙련된 DHCP 시스템 관리자에게만 권장됩니다.

```
# ifconfig interface IPv4-address netmask netmask
```

예를 들어, IP 주소 `192.168.84.3`을 `qfe1`에 지정하려면 다음 중 하나를 수행합니다.

- 기존 IPv4 표기법을 사용하여 다음을 입력합니다.

  ```
  # ifconfig qfe1 192.168.84.3 netmask 255.255.255.0
  ```

- CIDR 표기법을 사용하여 다음을 입력합니다.

  ```
  # ifconfig qfe1 192.168.84.3/24
  ```

  접두어 `/24`는 자동으로 `255.255.255.0` 넷마스크를 `qfe1`에 지정합니다. CIDR 접두어 및 이에 해당하는 점으로 구분된 십진수 넷마스크 표현에 대한 표는 그림 2–2를 참조하십시오.

6 **(선택 사항) 재부트 시에도 인터페이스 구성이 유지되도록 하려면 각 추가 물리적 인터페이스에 대해 `/etc/hostname.`*interface* 파일을 만듭니다.**

예를 들어, `/etc/hostname.qfe1` 및 `/etc/hostname.qfe2` 파일을 만듭니다. 그런 다음 `/etc/hostname.qfe1` 파일에 호스트 이름 `timbuktu`를 입력하고 `/etc/hostname.qfe1`

파일에 호스트 이름 timbuktu-201을 입력합니다. 단일 인터페이스 구성에 대한 자세한 내용은 138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”을 참조하십시오.

이 파일을 만든 다음 구성 재부트를 수행해야 합니다.

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

7 각 인터페이스의 호스트 이름 및 IP 주소를 /etc/inet/hosts 파일에 추가합니다.

예를 들면 다음과 같습니다.

```
172.16.26.232      deadsea        #interface for network 172.16.0.0
192.168.200.20     timbuktu       #interface for network 192.168.200
192.168.201.20     timbuktu-201   #interface for network 192.168.201
192.168.200.9      gobi
192.168.200.10     mojave
192.168.200.110    saltlake
192.168.200.12     chilean
```

인터페이스 timbuktu 및 timbuktu-201은 동일한 시스템에 있습니다. timbuktu-201의 네트워크 주소는 timbuktu의 네트워크 주소와 다릅니다. 이러한 차이는 네트워크 192.168.201의 물리적 네트워크 매체가 timbuktu-201 네트워크 인터페이스에 연결되고, 네트워크 192.168.200의 매체가 timbuktu 인터페이스에 연결되기 때문입니다.

8 Solaris 10 11/06 및 Solaris 10의 이전 릴리스만 각 새 인터페이스의 IP 주소 및 호스트 이름을 /etc/inet/ipnodes 파일 또는 해당 ipnodes 데이터베이스에 추가합니다.

예를 들면 다음과 같습니다.

```
vi /etc/inet/ipnodes
172.16.26.232      deadsea        #interface for network 172.16.0.0
192.168.200.20     timbuktu       #interface for network 192.168.200
192.168.201.20     timbuktu-201   #interface for network 192.168.201
```

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

- 기존 IPv4 주소 표기법(예: 192.168.83.0)의 경우 다음과 같이 입력합니다.

  ```
  192.168.83.0    255.255.255.0
  ```

- CIDR 주소의 경우 /etc/inet/netmask 파일의 항목에 있는 점으로 구분된 십진수 버전의 접두어를 사용합니다. 네트워크 접두어 및 이에 해당하는 점으로 구분된 십진수 표현은 그림 2–2에서 확인할 수 있습니다. 예를 들어, CIDR 네트워크 접두어 192.168.3.0/22를 표현하려면 /etc/netmasks의 다음 항목을 사용합니다.

  ```
  192.168.3.0 255.255.252.0
  ```

10 라우터에서 IPv4 패킷 전달을 사용으로 설정합니다.

다음 명령 중 하나를 사용하여 패킷 전달을 사용으로 설정합니다.

- 다음과 같이 routeadm 명령을 사용합니다.

  ```
  # routeadm -e ipv4-forwarding -u
  ```

- 다음 SMF(서비스 관리 기능) 명령을 사용합니다.

```
# svcadm enable ipv4-forwarding
```

이 경우 라우터는 로컬 네트워크 외부로 패킷을 전달할 수 있습니다. 라우터는 경로 지정 테이블에 수동으로 경로를 추가할 수 있는 프로세스인 **정적 경로 지정**을 지원합니다. 이 시스템에서 정적 경로 지정을 사용한다면 라우터 구성은 완료되었습니다. 그러나 시스템 경로 지정 테이블에서 경로를 유지 관리해야 합니다. 경로 추가에 대한 자세한 내용은 118 페이지 "경로 구성" 및 route(1M) 매뉴얼 페이지를 참조하십시오.

11 (선택 사항) 경로 지정 프로토콜을 시작합니다.

경로 지정 데몬 /usr/sbin/in.routed가 자동으로 경로 지정 테이블을 업데이트합니다. 이 프로세스를 **동적 경로 지정**이라고 합니다. 다음 방법 중 하나를 사용하여 기본 IPv4 경로 지정 프로토콜을 실행합니다.

- 다음과 같이 routeadm 명령을 사용합니다.

  ```
  # routeadm -e ipv4-routing -u
  ```

- 다음 SMF 명령을 사용하여 RIP와 같은 경로 지정 프로토콜을 시작합니다.

  ```
  # svcadm enable route:default
  ```

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

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

예 5–4 네트워크에 대한 기본 라우터 구성

이 예에서는 둘 이상의 인터페이스로 구성된 시스템을 기본 라우터로 업그레이드하는 방법을 보여 줍니다. 이 작업의 목표는 그림 5–3에 나와 있는 라우터 2를 네트워크 172.20.1.0의 기본 라우터로 만드는 것입니다. 라우터 2에는 두 개의 유선 네트워크 연결(네트워크 172.20.1.0에 대한 연결과 네트워크 10.0.5.0에 대한 연결)이 포함되어 있습니다. 이 예에서는 96 페이지 "호스트를 로컬 파일 모드로 구성하는 방법"에 설명된 대로 라우터가 로컬 파일 모드로 작동하도록 구성되었다고 간주합니다.

수퍼 유저 또는 동등한 역할의 사용자로 로그인한 후 시스템 인터페이스의 상태를 확인합니다. Solaris 10 1/06부터는 다음과 같이 dladm 명령을 사용할 수 있습니다.

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1

# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.20.1.10 netmask ffff0000 broadcast 172.20.10.100
        ether 8:0:20:c1:1b:c6
```

dladm show-link의 출력은 시스템에서 3개의 연결을 사용할 수 있다는 것을 나타냅니다. ce0 인터페이스만 IP 주소로 구성되었습니다. bge0 인터페이스를 10.0.5.0 네트워크에 물리적으로 연결하여 기본 라우터 구성을 시작합니다. 그런 다음 인터페이스를 연결하여 재부트 시에도 유지되게 합니다.

```
# ifconfig bge0 plumb
# ifconfig bge0 10.0.5.10/8 up
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.20.1.10 netmask ffff0000 broadcast 172.255.255.255
        ether 8:0:20:c1:1b:c6
bge0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.5.10 netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:e5:95:c4
 # vi /etc/hostname.bge0
10.0.5.10
255.0.0.0
```

재구성 부트 명령을 사용하여 시스템을 재부트합니다.

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

새로 연결된 인터페이스 및 연결된 네트워크에 대한 정보로 다음 네트워크 데이터베이스를 구성합니다.

```
# vi /etc/inet/hosts
127.0.0.1       localhost
172.20.1.10         router2         #interface for network 172.20.1
10.0.5.10           router2-out     #interface for network 10.0.5
# vi /etc/inet/netmasks
172.20.1.0    255.255.0.0
10.0.5.0      255.0.0.0
```

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

```
# svcadm enable ipv4-forwarding
# svcadm enable route:default
```

그러면 RIP를 통한 IPv4 패킷 전달 및 동적 경로 지정이 라우터 2에서 사용으로 설정되었지만, 172.20.1.0 네트워크에 대한 기본 라우터 구성은 아직 완료되지 않은 것입니다. 다음 작업을 수행해야 합니다.

- 호스트가 새 기본 라우터에서 경로 지정 정보를 가져오도록 172.10.1.10의 각 호스트를 수정합니다. 자세한 내용은 123 페이지 “단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법”을 참조하십시오.
- 라우터 2의 경로 지정 테이블에서 경계 라우터에 대한 정적 경로 지정을 정의합니다. 자세한 내용은 117 페이지 “경로 지정 테이블 및 경로 지정 유형”을 참조하십시오.

경로 지정 테이블 및 경로 지정 유형

라우터와 호스트 모두 **경로 지정 테이블**을 유지 관리합니다. 각 시스템의 경로 지정 데몬은 모든 알려진 경로가 있는 테이블을 업데이트합니다. 시스템의 커널은 패킷을 로컬 네트워크로 전달하기 전에 경로 지정 테이블을 읽습니다. 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              172.20.1.10          UG        1    532   ce0
224.0.0.0            10.0.5.100           U         1      0   bge0
10.0.0.0             10.0.5.100           U         1      0   bge0
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 경로 지정 테이블을 유지 관리합니다. **정적 경로 지정**만 실행하는 시스템은 정보 경로 지정 및 경로 지정 테이블 업데이트에 경로 지정 프로토콜을 사용하지 않습니다. 대신 `route` 명령을 통해 시스템의 알려진 경로를 수동으로 유지 관리해야 합니다. 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.
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 5–3에 표시된 AS는 정적 경로 지정과 동적 경로 지정을 결합한 것입니다.

경로 구성

IPv4 네트워크의 동적 경로 지정을 구현하려면 `routeadm` 또는 `svcadm` 명령을 사용하여 `in.routed` 경로 지정 데몬을 시작합니다. 자세한 내용은 112 페이지 “IPv4 라우터 구성 방법”을 참조하십시오. 동적 경로 지정은 대부분의 네트워크와 자율 시스템에서 선호되는 전략입니다. 그러나 네트워크 토폴로지 또는 네트워크의 특정 시스템에 정적 경로 지정이 필요할 수 있습니다. 이런 경우에는 시스템 경로 지정 테이블을 수동으로 편집하여 게이트웨이에 대한 알려진 경로를 반영해야 합니다. 다음 절차에서는 정적 경로를 추가하는 방법을 보여 줍니다.

주 – 시스템에서는 동일한 대상에 대한 두 경로를 통해 자동으로 로드 균형 조정 또는 페일오버를 수행하지 않습니다. 이러한 기능이 필요한 경우 27 장, “IPMP 소개(개요)”에 설명된 대로 IPMP를 사용하십시오.

▼ 경로 지정 테이블에 정적 경로 지정을 추가하는 방법

1 경로 지정 테이블의 현재 상태를 봅니다.

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   ipge0
224.0.0.0            198.168.5.10          U       1  0       ipge0
default              192.168.5.10          UG      1  91908
127.0.0.1            127.0.0.1             UH      1  811302   lo0
```

2 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

3 (선택 사항) 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.

`add` 다음 경로를 추가한다는 것을 나타냅니다.

`-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 5–5 경로 지정 테이블에 정적 경로 지정 추가

다음 예에서는 시스템에 정적 경로를 추가하는 방법을 보여 줍니다. 시스템은 라우터 2이며, 그림 5–3에 나와 있는 `172.20.1.0` 네트워크의 기본 라우터입니다. 예 5–4에서 라우터 2는 동적 경로 지정을 위해 구성되었습니다. 네트워크 `172.20.1.0`의 호스트에 대해 기본 라우터로 잘 작동하게 하려면 라우터 2에 AS의 경계 라우터(`10.0.5.150`)에 대한 정적 경로가 필요합니다.

라우터 2의 경로 지정 테이블을 보려면 다음을 입력합니다.

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.20.1.10          UG        1    249 ce0
224.0.0.0            172.20.1.10          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 라우터 2가 인식하는 두 경로를 나타냅니다. 기본 경로는 라우터 2의 `172.20.1.10` 인터페이스를 게이트웨이로 사용합니다. 두번째 경로 `10.0.5.0`은 라우터 2에서 실행되는 `in.routed` 데몬을 통해 검색되었습니다. 이 경로에 대한 게이트웨이는 IP 주소가 `10.0.5.20`인 라우터 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 `10.0.5.150/24`인 경계 라우터에 대한 경로가 포함됩니다.

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.20.1.10          UG        1    249 ce0
224.0.0.0            172.20.1.10          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 둘 이상의 인터페이스로 구성된 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 제5부를 참조하십시오.

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

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 **System Administration Guide: Security Services**의 "Firewall Systems" 또는 타사 방화벽의 설명서를 참조하십시오.

▼ 멀티홈 호스트를 만드는 방법

1 **잠재 멀티홈 호스트에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 **Oracle Solaris 설치 시 구성되지 않은 각 추가 네트워크 인터페이스를 구성 및 연결합니다.**

138 페이지 "시스템 설치 후 물리적 인터페이스 구성 방법"을 참조하십시오.

3 멀티홈 호스트에서 IP 전달이 사용으로 설정되지 않았는지 확인합니다.

```
# routeadm
```

routeadm 명령을 옵션 없이 사용하면 경로 지정 데몬의 상태가 보고됩니다. routeadm 명령의 다음 출력은 IPv4 전달이 사용으로 설정되었음을 보여 줍니다.

```
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
                IPv4 routing   disabled             disabled
                IPv6 routing   disabled             disabled
              IPv4 forwarding  enabled              disabled
              IPv6 forwarding  disabled             disabled

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

4 시스템에서 패킷 전달이 사용으로 설정되어 있으면 해제합니다.

다음 명령 중 하나를 사용합니다.

- routeadm 명령의 경우 다음을 입력합니다.

  ```
  # routeadm -d ipv4-forwarding -u
  ```

- SMF를 사용하려면 다음을 입력합니다.

  ```
  # svcadm disable ipv4-forwarding
  ```

5 (선택 사항) 멀티홈 호스트에 대한 동적 경로 지정을 켭니다.

다음 명령 중 하나를 사용하여 in.routed 데몬을 사용으로 설정합니다.

- routeadm 명령의 경우 다음을 입력합니다.

  ```
  # routeadm -e ipv4-routing -u
  ```

- SMF를 사용하려면 다음을 입력합니다.

  ```
  # svcadm enable route:default
  ```

예 5–6 멀티홈 호스트 구성

다음 예에서는 그림 5–3과 같이 멀티홈 호스트를 구성하는 방법을 보여 줍니다. 이 예에서는 시스템의 호스트 이름이 hostc입니다. 이 호스트에는 두 개의 인터페이스가 있으며 모두 네트워크 192.168.5.0에 연결됩니다.

시작하려면 시스템 인터페이스의 상태를 표시합니다.

```
# dladm show-link
hme0            type: legacy    mtu: 1500       device: hme0
qfe0            type: legacy    mtu: 1500       device: qfe0
qfe1            type: legacy    mtu: 1500       device: qfe1
qfe2            type: legacy    mtu: 1500       device: qfe2
qfe3            type: legacy    mtu: 1500       device: qfe3
```

```
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 192.168.5.82 netmask ff000000 broadcast 192.255.255.255
      ether 8:0:20:c1:1b:c6
```

dladm show-link 명령은 hostc에 총 5개의 잠재 연결이 있는 두 개의 인터페이스가 있음을 보고합니다. 그러나 hme0만 연결되어 있습니다. hostc를 멀티홈 호스트로 구성하려면 qfe0 또는 qfe NIC의 다른 연결을 추가합니다. 먼저 qfe0 인터페이스를 192.168.5.0 네트워크에 물리적으로 연결합니다. 그런 다음 qfe0 인터페이스를 연결하고 이 인터페이스가 재부트 시에도 유지되게 합니다.

```
# ifconfig qf0 plumb
# ifconfig qfe0 192.168.5.85/8 up
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.5.82 netmask ff0000 broadcast 192.255.255.255
        ether 8:0:20:c1:1b:c6
qfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.5.85 netmask ff000000 broadcast 192.255.255.255
        ether 8:0:20:e1:3b:c4
 # vi /etc/hostname.qfe0
192.168.5.85
255.0.0.0
```

재구성 명령을 사용하여 시스템을 재부트합니다.

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

그런 다음 qfe0 인터페이스를 hosts 데이터베이스에 추가합니다.

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

그런 다음 패킷 전달 및 host3의 경로 지정 상태를 확인합니다.

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

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

routeadm 명령이 in.routed 데몬을 통한 동적 경로 지정 및 패킷 전달이 현재 사용으로 설정된 것으로 보고합니다. 그러나 패킷 전달은 사용 안함으로 설정해야 합니다.

```
# svcadm disable ipv4-forwarding
```

120 페이지 "멀티홈 호스트를 만드는 방법"에 설명된 대로 routeadm 명령을 사용하여 패킷 전달 기능을 해제할 수도 있습니다. 패킷 전달이 사용 안함으로 설정되면 host3이 멀티홈 호스트가 됩니다.

단일 인터페이스 시스템에 대한 경로 지정 구성

단일 인터페이스 호스트는 몇 가지 경로 지정 형태를 구현해야 합니다. 호스트가 하나 이상의 로컬 기본 라우터에서 경로를 얻는 경우에는 정적 경로 지정을 사용하도록 호스트를 구성해야 합니다. 그렇지 않은 경우에는 동적 경로 지정이 호스트에 권장됩니다. 다음 절차에서는 두 경로 지정 유형을 사용으로 설정하는 지침도 제공합니다.

▼ 단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정하는 방법

이 절차에서는 단일 인터페이스 호스트에서 정적 경로 지정을 사용으로 설정할 수 있습니다. 정적 경로 지정을 사용하는 호스트는 RIP와 같은 동적 경로 지정 프로토콜을 실행하지 않습니다. 대신 호스트는 경로 지정 정보에 기본 라우터의 서비스를 사용해야 합니다. 그림 109 페이지 "IPv4 자율 시스템 토폴로지"에서는 몇 가지 기본 라우터와 해당 클라이언트 호스트를 보여 줍니다. 특정 호스트 설치 시 기본 라우터의 이름을 지정한 경우 해당 호스트는 이미 정적 경로 지정을 사용하도록 구성되어 있습니다.

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

/etc/defaultrouter 파일에 대한 자세한 내용은 219 페이지 "/etc/defaultrouter 파일"을 참조하십시오. 정적 경로 지정 및 경로 지정 테이블에 대한 자세한 내용은 117 페이지 "경로 지정 테이블 및 경로 지정 유형"을 참조하십시오.

1. **단일 인터페이스 호스트에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2. **/etc/defaultrouter 파일이 호스트에 있는지 확인합니다.**

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

3 **텍스트 편집기를 열고 /etc/defaultrouter 파일을 만들거나 수정합니다.**

4 **기본 라우터에 대한 항목을 추가합니다.**

```
# vi  /etc/defaultrouter
router-IP
```

여기에서 *router-IP*는 사용할 호스트에 대한 기본 라우터의 IP 주소를 나타냅니다.

5 **호스트에서 경로 지정 및 패킷 전달이 실행되지 않는지 확인합니다.**

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

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

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

/etc/inet/hosts 구성에 대한 자세한 내용은 101 페이지 “IPv4 주소 및 기타 네트워크 구성 매개변수 변경 방법”을 참조하십시오.

예 5–7 단일 인터페이스 호스트의 기본 라우터 및 정적 경로 지정 구성

다음 예에서는 그림 5–3에 나와 있는 네트워크 172.20.1.0의 단일 인터페이스 호스트(hostb)에 대한 정적 경로 지정을 구성하는 방법을 보여 줍니다. hostb에서는 라우터 2를 기본 라우터로 사용해야 합니다.

먼저 수퍼 유저 또는 이와 동등한 역할로 hostb에 로그인합니다. 그런 다음 /etc/defaultrouter 파일이 호스트에 있는지 여부를 확인합니다.

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

grep에서 응답이 없으면 /etc/defaultrouter 파일을 만들어야 합니다.

```
# vi /etc/defaultrouter
172.20.1.10
```

/etc/defaultrouter 파일의 항목은 172.20.1.0 네트워크에 연결된 라우터 2 인터페이스의 IP 주소입니다. 그런 다음 현재 호스트에서 패킷 전달 또는 경로 지정이 사용으로 설정되었는지 확인합니다.

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

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

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

이 특정 호스트에 대한 패킷 전달은 사용으로 설정되어 있습니다. 이 기능을 다음과 같이 해제합니다.

```
# svcadm disable ipv4-forwarding
```

마지막으로 호스트의 /etc/inet/hosts 파일에 새 기본 라우터에 대한 항목이 있는지 확인합니다.

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

▼ 단일 인터페이스 호스트에서 동적 경로 지정을 사용으로 설정하는 방법

동적 경로 지정은 호스트의 경로 지정을 관리하기 위한 가장 쉬운 방법입니다. 동적 경로 지정을 사용하는 호스트는 IPv4의 경우 in.routed 데몬 또는 IPv6의 경우 in.ripngd 데몬에서 제공하는 경로 지정 프로토콜을 실행합니다. 다음 절차를 사용하여 단일 인터페이스 호스트에서 IPv4 동적 경로 지정을 사용으로 설정합니다. 동적 경로 지정에 대한 자세한 내용은 105 페이지 “IPv4 네트워크에서의 패킷 전달 및 경로 지정”을 참조하십시오.

1 호스트에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 /etc/defaultrouter 파일이 있는지 확인합니다.

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

3 /etc/defaultrouter가 있으면 모든 항목을 삭제합니다.

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

4 호스트에서 패킷 전달 및 경로 지정이 사용으로 설정되어 있는지 확인합니다.

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

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

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

5 패킷 전달이 사용으로 설정된 경우 해제합니다.

다음 명령 중 하나를 사용합니다.

- `routeadm` 명령의 경우 다음을 입력합니다.

  ```
  # routeadm -d ipv4-forwarding -u
  ```

- SMF를 사용하려면 다음을 입력합니다.

  ```
  # svcadm disable ipv4-forwarding
  ```

6 호스트에서 경로 지정 프로토콜을 사용하도록 설정합니다.

다음 명령 중 하나를 사용합니다.

- `routeadm` 명령의 경우 다음을 입력합니다.

  ```
  # routeadm -e ipv4-routing -u
  ```

- SMF를 사용하려면 다음을 입력합니다.

  ```
  # svcadm enable route:default
  ```

이제 IPv4 동적 경로 지정이 사용으로 설정되었습니다. 호스트의 경로 지정 테이블은 `in.routed` 데몬에 의해 동적으로 유지 관리됩니다.

예 5–8 단일 인터페이스 호스트에서 동적 경로 지정 실행

다음 예에서는 그림 5–3에 나와 있는 네트워크 `192.168.5.0`의 단일 인터페이스 호스트(`hosta`)에 대한 동적 경로 지정을 구성하는 방법을 보여 줍니다.`hosta`에서는 현재 라우터 1을 기본 라우터로 사용합니다. 그러나 이제 `hosta`는 동적 경로 지정을 실행해야 합니다.

먼저 수퍼 유저 또는 이와 동등한 역할로 `hosta`에 로그인합니다. 그런 다음 `/etc/defaultrouter` 파일이 호스트에 있는지 여부를 확인합니다.

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

`grep`에서 응답이 있으면 `hosta`에 대한 `/etc/defaultrouter` 파일이 있음을 나타냅니다.

```
# vi /etc/defaultrouter
192.168.5.10
```

파일에는 라우터 1의 IP 주소인 192.168.5.10 항목이 있습니다. 이 항목을 삭제하여 정적 경로 지정을 사용으로 설정합니다. 그런 다음 호스트에 대해 패킷 전달 및 경로 지정이 이미 사용으로 설정되어 있는지 확인해야 합니다.

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

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

hosta에 대한 경로 지정 및 패킷 전달이 모두 해제되어 있습니다. 경로 지정을 설정하여 다음과 같이 hosta에 대한 동적 경로 지정 구성을 완료합니다.

```
# svcadm enable route:default
```

전송 계층 서비스 모니터 및 수정

전송 계층 프로토콜 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 관리: 기본 관리**의 18 장, “서비스 관리(개요)”에 설명된 대로 SMF(서비스 관리 기능)를 사용하여 이러한 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. inetd에 대한 자세한 내용은 227 페이지 “inetd 인터넷 서비스 데몬”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 **로컬 시스템에서는 네트워크 관리 역할 또는 수퍼 유저로 로그인합니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **inetd로 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 멀티홈 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 discard가 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 관리: 기본 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 5–9 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 **System Administration Guide: Network Services**의 "Support for TCP Wrappers From Version 8.12 of sendmail"에 설명된 대로 TCP 래퍼도 사용할 수 있습니다.

1. **로컬 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"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 Oracle Solaris CD-ROM과 함께 패키지에 포함된 SFW CD-ROM의 `/usr/sfw/man` 디렉토리에서 확인할 수 있습니다.

◆ ◆ ◆ 6 장

6 네트워크 인터페이스 관리(작업)

이 장에서는 네트워크 인터페이스에 대한 작업 및 정보를 설명합니다.

네트워크 인터페이스 관리의 새로운 기능

이 장의 정보는 Solaris 10 1/06 릴리스부터 적용되는 인터페이스 구성에 대해 설명합니다. 새로운 Oracle Solaris 기능의 전체 목록 및 Oracle Solaris 릴리스에 대한 설명은 **Oracle Solaris 10 1/13 새로운 기능**을 참조하십시오.

Solaris 10 1/06에서는 다음과 같은 새로운 기능이 도입되었습니다.

- 인터페이스 상태를 보기 위한 새로 도입된 `dladm` 명령에 대해서는 138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”을 참조하십시오.
- 143 페이지 “VLAN(가상 LAN) 관리”에 설명된 대로 VLAN 지원이 GLDv3 인터페이스로 확장되었습니다.
- 링크 통합 지원은 149 페이지 “링크 통합 개요”에서 소개합니다.

Solaris 10 7/07에서 `/etc/inet/ipnodes`는 더 이상 사용되지 않습니다. 개별 절차에 설명된 대로 Solaris 10 이전 릴리스에서만 `/etc/inet/ipnodes`를 사용하십시오.

인터페이스 관리(작업 맵)

다음 표에서는 VLAN 및 링크 통합과 같은 특별 구성을 비롯한 네트워크 인터페이스를 구성하는 여러 작업을 보여 줍니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
시스템의 인터페이스 상태를 확인합니다.	시스템의 모든 인터페이스를 나열하고 이미 연결된 인터페이스를 확인합니다.	137 페이지 “인터페이스 상태를 가져오는 방법”
시스템 설치 후 단일 인터페이스를 추가합니다.	다른 인터페이스를 구성하여 시스템을 멀티홈 호스트 또는 라우터로 변경합니다.	138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”
SPARC: 인터페이스의 MAC 주소가 고유한지 확인합니다.	인터페이스가 시스템 MAC 주소(SPARC 전용)가 아니라 출하 시 설치된 MAC 주소로 구성되었는지 확인합니다.	141 페이지 “SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법”
VLAN(가상 LAN)을 계획합니다.	VLAN을 만들기 전에 필요한 계획 작업을 수행합니다.	146 페이지 “VLAN 구성을 계획하는 방법”
VLAN을 구성합니다.	네트워크에서 VLAN을 만들고 수정합니다.	147 페이지 “VLAN을 구성하는 방법”
통합을 계획합니다.	통합을 설계하고 통합을 구성하기 전에 필요한 계획 작업을 수행합니다.	149 페이지 “링크 통합 개요”
통합을 구성합니다.	링크 통합과 관련된 다양한 작업을 수행합니다.	153 페이지 “링크 통합을 만드는 방법”
IPMP 그룹을 계획 및 구성합니다.	IPMP 그룹의 구성원인 인터페이스에 대해 페일오버 및 페일백을 구성합니다.	671 페이지 “IPMP 그룹을 계획하는 방법” 673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”

물리적 인터페이스 관리를 위한 기본 사항

네트워크 인터페이스는 시스템과 네트워크 사이의 연결을 제공합니다. Oracle Solaris 기반 시스템은 물리적 및 논리적의 두 가지 인터페이스 유형을 포함할 수 있습니다. **물리적 인터페이스**는 소프트웨어 드라이버와 네트워크 매체를 연결하는 커넥터(예: 이더넷 케이블)로 구성됩니다. 물리적 인터페이스는 관리 또는 가용성 목적에 따라 그룹화할 수 있습니다. **논리적 인터페이스**는 일반적으로 주소를 추가하고 물리적 인터페이스에서 터널 끝점을 만들기 위해 기존 물리적 인터페이스로 구성됩니다.

주 – 논리적 네트워크 인터페이스는 사용되는 작업(예: IPv6 작업, IPMP 작업, DHCP 작업 등)에서 설명됩니다.

대부분의 컴퓨터 시스템에는 기본 시스템 보드의 제조업체에서 **내장한** 하나 이상의 물리적 인터페이스가 포함되어 있습니다. 또한 일부 시스템에는 내장 인터페이스가 두 개 이상 포함될 수 있습니다.

내장 인터페이스 외에도 별도로 구입한 인터페이스를 시스템에 추가할 수 있습니다. 별도로 구입한 인터페이스는 *NIC(***네트워크 인터페이스 카드***)*라고 부릅니다. NIC는 제조업체 지침에 따라 물리적으로 설치합니다.

주 – NIC를 **네트워크 어댑터**라고도 부릅니다.

시스템 설치 중 Oracle Solaris 설치 프로그램은 물리적으로 설치된 모든 인터페이스를 감지하고 각 인터페이스 이름을 표시합니다. 인터페이스 목록 중 하나 이상의 인터페이스를 구성해야 합니다. 설치 중 구성하는 첫번째 인터페이스는 **기본 네트워크 인터페이스**가 됩니다. 기본 네트워크 인터페이스의 IP 주소는 `/etc/nodename` 파일에 저장된 시스템의 구성된 호스트 이름과 연결됩니다. 하지만 설치 중 또는 설치 후에 추가 인터페이스를 구성할 수 있습니다.

네트워크 인터페이스 이름

각 물리적 인터페이스는 고유한 장치 이름으로 식별됩니다. 장치 이름의 구문은 다음과 같습니다.

`<driver-name><instance-number>`

Oracle Solaris 시스템의 드라이버 이름은 `ce`, `hme`, `bge`, `e1000g` 및 기타 다른 드라이버 이름을 포함할 수 있습니다. *instance-number* 변수는 시스템에 설치된 해당 드라이버 유형의 인터페이스 수에 따라 0에서 *n*까지 값을 포함할 수 있습니다.

예를 들어, 호스트 시스템과 서버 시스템 모두에서 기본 네트워크 인터페이스로 자주 사용되는 100BASE-TX 고속 이더넷 인터페이스를 가정해 보십시오. 이 인터페이스에 대해 일반적으로 사용되는 드라이버 이름은 `eri`, `qfe` 및 `hme`입니다. 기본 네트워크 인터페이스로 사용되는 고속 이더넷 인터페이스는 장치 이름이 `eri0` 또는 `qfe0`으로 지정됩니다.

`eri` 및 `hme`와 같은 NIC는 인터페이스를 하나만 포함합니다. 하지만 많은 NIC 제품들에 다중 인터페이스가 포함됩니다. 예를 들어 Quad Fast Ethernet(`qfe`) 카드는 `qfe0`부터 `qfe3`까지 4개의 인터페이스를 포함합니다.

인터페이스 연결

인터페이스를 통해 시스템 및 네트워크 사이에 트래픽을 전달할 수 있으려면 인터페이스를 먼저 **연결**해야 합니다. 연결 프로세스에는 인터페이스를 장치 이름과

연결하는 과정이 포함됩니다. 그런 후 인터페이스를 IP 프로토콜에서 사용할 수 있도록 스트림을 설정합니다. 물리적 인터페이스와 논리적 인터페이스를 모두 연결해야 합니다. 인터페이스는 부트 시퀀스의 일부로 또는 `ifconfig` 명령의 적합한 구문을 사용해서 명시적으로 연결됩니다.

설치 중 인터페이스를 구성할 때 인터페이스가 자동으로 연결됩니다. 설치 중 시스템에 추가 인터페이스를 구성하지 않으려는 경우 해당 인터페이스가 연결되지 않습니다.

Oracle Solaris 인터페이스 유형

Solaris 10 1/06 릴리스부터 Oracle Solaris에서는 다음과 같은 두 가지 유형의 인터페이스가 지원됩니다.

- **레거시 인터페이스** – 이러한 인터페이스는 DLPI 인터페이스 및 GLDv2 인터페이스입니다. 일부 레거시 인터페이스 유형은 `eri`, `qfe` 및 `ce`입니다. `dladm show-link` 명령으로 인터페이스 상태를 검사할 때는 이러한 인터페이스가 "legacy"로 보고됩니다.
- **비VLAN 인터페이스** – 이러한 인터페이스는 GLDv3 인터페이스입니다.

주 – 현재까지 GLDv3은 `bge`, `xge` 및 `e1000g` 인터페이스 유형에서 지원됩니다.

개별 네트워크 인터페이스 관리

Oracle Solaris 설치 후에는 다음 목적으로 시스템에서 인터페이스를 구성 및 관리할 수 있습니다.

- 멀티홈 호스트가 되도록 시스템을 업그레이드합니다. 자세한 내용은 120 페이지 "멀티홈 호스트 구성"을 참조하십시오.
- 호스트를 라우터로 변경합니다. 라우터 구성에 대한 자세한 내용은 112 페이지 "IPv4 라우터 구성"을 참조하십시오.
- VLAN의 일부로 인터페이스를 구성합니다. 자세한 내용은 143 페이지 "VLAN(가상 LAN) 관리"를 참조하십시오.
- 집계의 멤버로 인터페이스를 구성합니다. 자세한 내용은 149 페이지 "링크 통합 개요"를 참조하십시오.
- IPMP 그룹에 인터페이스를 추가합니다. IPMP 그룹 구성에 대한 자세한 내용은 671 페이지 "고가용성을 위해 IPMP 그룹 사용"을 참조하십시오.

이 섹션에는 Solaris 10 1/06 릴리스부터 시작하여 개별 네트워크 인터페이스 구성에 대한 정보가 포함됩니다. 다음 그룹화 중 하나로 인터페이스를 구성하는 방법에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- VLAN 인터페이스 구성에 대한 자세한 내용은 143 페이지 "VLAN(가상 LAN) 관리"를 참조하십시오.

- 통합 인터페이스 구성에 대한 자세한 내용은 149 페이지 “링크 통합 개요”를 참조하십시오.
- IPMP 그룹의 멤버로 인터페이스를 구성하는 방법에 대한 자세한 내용은 671 페이지 “고가용성을 위해 IPMP 그룹 사용”을 참조하십시오.

▼ 인터페이스 상태를 가져오는 방법

Solaris 10 1/06부터 이 절차에서는 시스템에서 현재 사용할 수 있는 인터페이스 및 해당 상태를 확인하는 방법에 대해 설명합니다. 이 절차에서는 또한 현재 연결된 인터페이스를 보여줍니다. 이전 Solaris 10 3/05를 사용 중인 경우에는 191 페이지 “특정 인터페이스에 대한 정보를 얻는 방법”을 참조하십시오.

1 **인터페이스를 구성할 시스템에서 기본 관리자 역할을 맡거나 수퍼 유저로 전환합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **시스템에 현재 설치되어 있는 인터페이스를 확인합니다.**

```
# dladm show-link
```

이 단계에서는 dladm(1M) 매뉴얼 페이지에서 자세히 설명하는 dladm 명령을 사용합니다. 이 명령은 현재 구성된 인터페이스에 관계없이 검색된 모든 인터페이스 드라이버를 보고합니다.

3 **시스템에서 현재 연결된 인터페이스를 확인합니다.**

```
# ifconfig -a
```

ifconfig 명령에는 인터페이스 연결을 포함하여 많은 추가 기능이 있습니다. 자세한 내용은 ifconfig(1M) 매뉴얼 페이지를 참조하십시오.

예 6–1 dladm 명령을 사용하여 인터페이스 상태 가져오기

다음 예에서는 dladm 명령의 상태 표시를 보여줍니다.

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
```

dladm show-link 출력에는 로컬 호스트에 사용할 수 있는 4개의 인터페이스 드라이버가 나타납니다. ce 및 bge 인터페이스를 모두 VLAN에 대해 구성할 수 있습니다. 하지만 non-VLAN 유형의 GLDV3 인터페이스만 링크 집계에 사용할 수 있습니다.

다음 예에서는 ifconfig - a 명령의 상태 표시를 보여줍니다.

```
# ifconfig -a
 lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>mtu 1500 index 3
        inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
       ether 0:3:ba:7:84:5e
bge0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4>mtu 1500 index 2
        inet 10.8.57.39 netmask ffffff00 broadcast 10.8.57.255
       ether 0:3:ba:29:fc:cc
```

`ifconfig -a` 명령의 출력에는 `ce0` 및 `bge0`의 두 가지 인터페이스의 통계만 표시됩니다. 이 출력은 `ce0` 및 `bge0`만 연결되었고 네트워크 트래픽에 사용할 준비가 되었음을 보여줍니다. 이러한 인터페이스는 VLAN에서 사용할 수 있습니다. `bge0`이 연결되었기 때문에 더 이상 이 인터페이스를 집계에 사용할 수 없습니다.

▼ 시스템 설치 후 물리적 인터페이스 구성 방법

시작하기 전에

- 추가 인터페이스에 대해 사용하려는 IPv4 주소를 확인합니다.
- 구성할 물리적 인터페이스가 시스템에 물리적으로 설치되었는지 확인합니다. 별도로 구입한 NIC 하드웨어 설치에 대한 자세한 내용은 해당 NIC와 함께 제공된 제조업체 지침을 참조하십시오.
- 인터페이스를 바로 설치했으면 다음 단계를 진행하기 전에 재구성 부트를 수행하십시오.

1 인터페이스를 구성할 시스템에서 기본 관리자 역할을 맡거나 수퍼 유저로 전환합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 시스템에 현재 설치되어 있는 인터페이스를 확인합니다.

```
# dladm show-link
```

3 각 인터페이스를 구성하고 연결합니다.

```
# ifconfig interface plumb up
```

예를 들어, `qfe0`의 경우 다음을 입력합니다.

```
# ifconfig qfe0 plumb up
```

주 – `ifconfig` 명령을 사용해서 명시적으로 구성된 인터페이스는 재부트 시 보존되지 않습니다.

---

4 인터페이스에 IPv4 주소 및 넷마스크를 지정합니다.

```
# ifconfig interface IPv4-address netmask+netmask
```

예를 들어, qfe0의 경우 다음을 입력합니다.

```
# ifconfig
qfe0 192.168.84.3 netmask + 255.255.255.0
```

주 – 기존 IPv4 표기법 또는 CIDR 표기법으로 IPv4 주소를 지정할 수 있습니다.

5 새로 구성된 인터페이스가 연결되었고 구성되었는지 또는 "UP" 상태인지 확인합니다.

```
# ifconfig
-a
```

표시된 각 인터페이스의 상태 표시줄을 확인합니다. 출력의 상태 라인에 UP 플래그가 포함되었는지 확인합니다. 예를 들면 다음과 같습니다.

```
qfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 2
```

6 (선택 사항) 재부트 시에도 인터페이스 구성이 지속되도록 하려면 다음 단계를 수행합니다.

a. 구성할 각 인터페이스에 대해 /etc/hostname.*interface* 파일을 만듭니다.

예를 들어, qfe0 인터페이스를 추가하려면 다음 파일을 만들어야 합니다.

```
# vi /etc/hostname.qfe0
```

주 – 동일한 인터페이스에 대한 대체 호스트 이름 파일을 만드는 경우, 대체 파일도 이름 지정 형식 hostname.[0–9]* (예: hostname.qfe0.a123)를 따라야 합니다. hostname.qfe0.bak 또는 hostname.qfe0.old 등의 이름은 잘못된 이름으로 시스템 부트 중 스크립트에서 무시됩니다.

또한 제공된 인터페이스의 해당 호스트 이름은 한 개뿐이어야 합니다. 인터페이스에 대한 대체 호스트 이름 파일을 유효한 파일 이름(예: /etc/hostname.qfe 및 /etc/hostname.qfe.a123)으로 만들 경우 부트 스크립트에서 두 호스트 이름 파일의 내용을 참조하여 구성을 시도하므로 오류가 발생합니다. 이러한 오류를 방지하려면 제공된 구성에 사용하지 않을 호스트 이름 파일에는 잘못된 파일 이름을 제공하십시오.

b. /etc/hostname.*interface* 파일을 편집합니다.

최소한 인터페이스의 IPv4 주소를 파일에 추가합니다. 기존 IPv4 표기법 또는 CIDR 표기법을 사용하여 인터페이스의 IP 주소를 지정할 수 있습니다. 또한 넷마스크 및 기타 구성 정보를 파일에 추가할 수도 있습니다.

주 – 인터페이스에 IPv6 주소를 추가하려면 169 페이지 “호스트 및 서버에 대해 IPv6 인터페이스 구성 수정”을 참조하십시오.

c. Solaris 10 11/06 및 이전 Solaris 10 릴리스의 경우 새 인터페이스에 대한 항목을 `/etc/inet/ipnodes` 파일에 추가합니다.

d. 새 인터페이스의 항목을 `/etc/inet/hosts` 파일에 추가합니다.

e. 재구성 부트 수행.

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

f. `/etc/hostname.`*interface* 파일에 만든 인터페이스가 구성되었는지 확인합니다.

```
# ifconfig -a
```

예를 보려면 예 6–2를 참조하십시오.

예 6–2 영구적인 인터페이스 구성 추가

이 예에서는 `qfe0` 및 `qfe1` 인터페이스를 호스트에 구성하는 방법을 보여줍니다. 이러한 인터페이스는 재부트 시에도 지속됩니다.

```
# dladm show-link
eri0    type: legacy    mtu: 1500       device: eri0
qfe0    type: legacy    mtu: 1500       device: qfe0
qfe1    type: legacy    mtu: 1500       device: qfe1
qfe2    type: legacy    mtu: 1500       device: qfe2
qfe3    type: legacy    mtu: 1500       device: qfe3
bge0    type: non-vlan  mtu: 1500       device: bge0
# vi /etc/hostname.qfe0
192.168.84.3 netmask 255.255.255.0
# vi /etc/hostname.qfe1
192.168.84.72 netmask 255.255.255.0
# vi /etc/inet/hosts
# Internet host table
#
127.0.0.1       localhost
10.0.0.14       myhost
192.168.84.3       interface-2
192.168.84.72       interface-3
For Solaris 10 11/06 and earlier releases:# vi /etc/inet/ipnodes
10.0.0.14 myhost
192.168.84.3       interface-2
192.168.84.72       interface-3
```

이제 시스템을 재부트합니다.

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

시스템 부트 후에는 인터페이스 구성을 확인합니다.

```
ifconfig -a
# ifconfig -a lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu
8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.14netmask ff000000 broadcast 10.255.255.255
        ether 8:0:20:c1:8b:c3
qfe0:flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
      inet 192.168.84.3 netmask ffffff00 broadcast 192.255.255.255
      ether 8:0:20:c8:f4:1d
qfe1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>mtu 1500 index 4
        inet 192.168.84.72 netmask ffffff00 broadcast 10.255.255.255
       ether 8:0:20:c8:f4:1e
```

참조

- IPv6 주소를 인터페이스에 구성하려면 160 페이지 “현재 세션에 대해 IPv6 인터페이스를 사용으로 설정하는 방법”을 참조하십시오.
- IPMP(IP Network Multipathing)를 사용해서 인터페이스에 대한 페일오버 감지 및 페일백을 설정하려면 28 장, “IPMP 관리(작업)”를 참조하십시오.

▼ 물리적 인터페이스를 제거하는 방법

1. **인터페이스를 제거할 시스템에서 기본 관리자 역할을 맡거나 수퍼 유저로 전환합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2. **물리적 인터페이스를 제거합니다.**

   ```
   # ifconfig interface down unplumb
   ```

   예를 들어, `qfe1` 인터페이스를 제거하려면 다음을 입력합니다.

   ```
   # ifconfig qfe1 down unplumb
   ```

▼ SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법

MAC 주소를 구성하려면 이 절차를 사용하십시오.

일부 응용 프로그램의 경우 호스트의 모든 인터페이스에서 고유한 MAC 주소를 사용해야 합니다. 그러나 모든 SPARC 기반 시스템은 시스템 전체에 적용되는 MAC 주소를 사용하며, 기본적으로 이 주소가 모든 인터페이스에서 사용됩니다. 다음은 SPARC 시스템의 인터페이스에 대해 출하 시 설치된 MAC 주소를 구성할 수 있는 두 가지 경우입니다.

- 링크 집계의 경우, 인터페이스의 출하 시 설정된 MAC 주소를 집계 구성에 사용해야 합니다.

- IPMP 그룹의 경우 그룹에 있는 각 인터페이스가 고유한 MAC 주소를 사용해야 합니다. 이러한 인터페이스는 출하 시 설치된 MAC 주소를 사용해야 합니다.

EEPROM 매개변수 `local-mac-address?`는 SPARC 시스템의 모든 인터페이스에서 시스템 전체에 적용되는 MAC 주소를 사용하는지 아니면 고유한 MAC 주소를 사용하는지 여부를 지정합니다. 다음 절차는 `eeprom` 명령을 사용하여 `local-mac-address?`의 현재 값을 확인하고 필요한 경우 변경하는 방법을 보여줍니다.

1 인터페이스를 구성할 시스템에서 기본 관리자 역할을 맡거나 수퍼 유저로 전환합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 시스템의 모든 인터페이스가 현재 시스템 차원의 MAC 주소를 사용하는지 확인합니다.

```
# eeprom local-mac-address?
local-mac-address?=false
```

이 예에서 `eeprom` 명령에 대한 응답 `local-mac-address?=false`는 모든 인터페이스가 시스템 차원의 MAC 주소를 사용함을 나타냅니다. 인터페이스가 IPMP 그룹의 구성원이 되려면 먼저 `local-mac-address?=false` 값을 `local-mac-address?=true`로 변경해야 합니다. 집계에 대해서도 `local-mac-address?=false`를 `local-mac-address?=true`로 변경해야 합니다.

3 필요한 경우 `local-mac-address?` 값을 다음과 같이 변경합니다.

```
# eeprom local-mac-address?=true
```

시스템을 재부트하면 출하 시 설치된 MAC 주소를 사용하는 인터페이스가 이제 시스템 전체에 적용되는 MAC 주소가 아닌 이러한 출하 시 설정을 사용합니다. 출하 시 설정된 MAC 주소를 사용하지 않는 인터페이스는 계속 시스템 전체에 적용되는 MAC 주소를 사용합니다.

4 시스템에 있는 모든 인터페이스의 MAC 주소를 확인합니다.

여러 인터페이스가 동일한 MAC 주소를 가진 경우를 찾습니다. 이 예에서는 모든 인터페이스에서 시스템 전체에 적용되는 MAC 주소인 `8:0:20:0:0:1`을 사용합니다.

```
ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
      inet 127.0.0.1 netmask ff000000
hme0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 10.0.0.112 netmask ffffff80 broadcast 10.0.0.127
      ether 8:0:20:0:0:1
ce0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 10.0.0.114 netmask ffffff80 broadcast 10.0.0.127
      ether 8:0:20:0:0:1
ce1: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 10.0.0.118 netmask ffffff80 broadcast 10.0.0.127
      ether 8:0:20:0:0:1
```

주 – 둘 이상의 네트워크 인터페이스가 여전히 동일한 MAC 주소를 사용하는 경우 다음 단계로 진행합니다. 그렇지 않은 경우 마지막 단계로 이동합니다.

5 **필요한 경우 모든 인터페이스가 고유한 MAC 주소를 갖도록 나머지 인터페이스를 수동으로 구성합니다.**

`/etc/hostname.`*interface* 파일에서 특정 인터페이스에 대해 고유한 MAC 주소를 지정합니다.

이전 단계의 예에서는 로컬에서 관리되는 MAC 주소를 사용하여 `ce0` 및 `ce1`을 구성해야 합니다. 예를 들어, 로컬로 관리되는 MAC 주소 `06:05:04:03:02`로 `ce1`을 재구성하려면 `/etc/hostname.ce1`에 다음 라인을 추가해야 합니다.

```
ether 06:05:04:03:02
```

주 – 수동으로 구성한 MAC 주소가 네트워크에서 다른 MAC 주소와 충돌되지 않도록 하려면 IEEE 802.3 표준에 정의된 대로 항상 **로컬로 관리되는** MAC 주소를 구성해야 합니다.

또한 `ifconfig ether` 명령을 사용하여 현재 세션에 대한 인터페이스의 MAC 주소를 구성할 수 있습니다. 하지만 `ifconfig`로 변경한 사항은 재부트 시에 보존되지 않습니다. 자세한 내용은 `ifconfig(1M)` 매뉴얼 페이지를 참조하십시오.

6 **시스템을 재부트합니다.**

VLAN(가상 LAN) 관리

*VLAN(***가상** *LAN)*은 TCP/IP 프로토콜 스택의 데이터 링크 계층에서 LAN(Local Area Network)의 하위 분할입니다. 스위치 기술을 사용하는 LAN에 대해 VLAN을 만들 수 있습니다. VLAN에 사용자 그룹을 지정하면 전체 로컬 네트워크에 대한 네트워크 관리와 보안을 향상시킬 수 있습니다. 동일한 시스템의 인터페이스를 서로 다른 VLAN에 지정할 수도 있습니다.

다음을 수행해야 하는 경우 로컬 네트워크를 VLAN으로 분할해 보십시오.

- 작업 그룹의 논리적 분할을 만듭니다.

  예를 들어, 건물 한 층의 모든 호스트가 스위치 기반 로컬 네트워크 한 개에 연결되어 있다고 가정합니다. 한 층의 각 작업 그룹에 대해 별도의 VLAN을 만들 수 있습니다.
- 작업 그룹에 대해 서로 다른 보안 정책을 적용합니다.

  예를 들어, 재무 부서와 정보 기술 부서의 보안 요구는 완전히 다릅니다. 두 부서의 시스템이 동일한 로컬 네트워크를 공유하는 경우 각 부서에 대해 별도의 VLAN을 만들 수 있습니다. 그런 다음 VLAN별로 적절한 보안 정책을 적용할 수 있습니다.
- 작업 그룹을 관리 가능한 브로드캐스트 도메인으로 분할합니다.

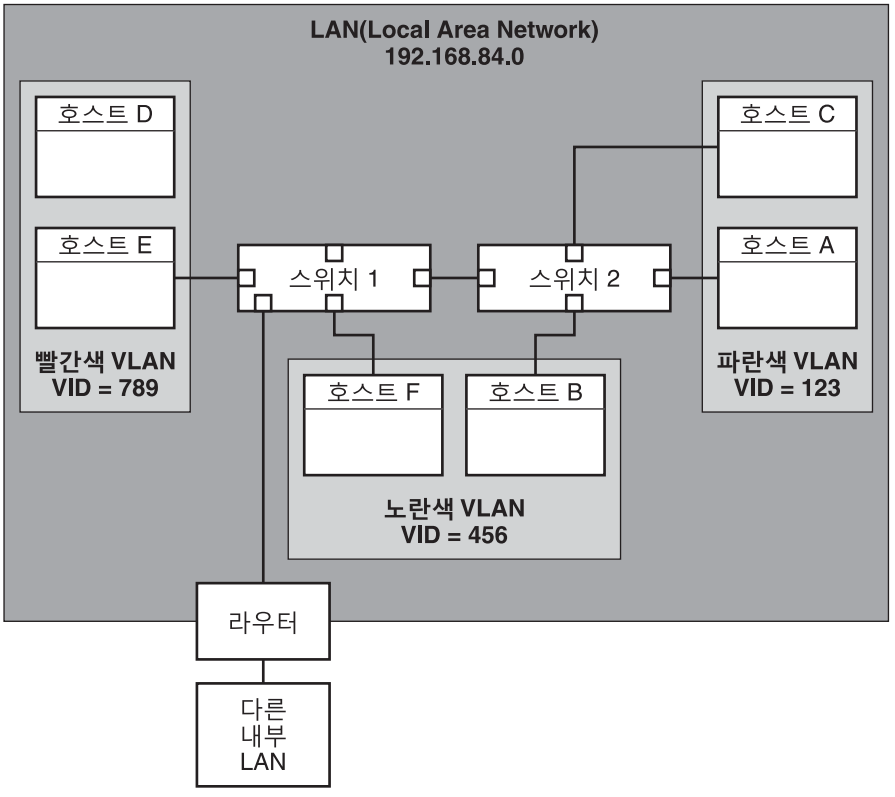
VLAN을 사용하면 브로드캐스트 도메인의 크기가 감소하며 네트워크 효율성이 향상됩니다.

VLAN 토폴로지 개요

스위치 LAN 기술을 사용하면 로컬 네트워크의 시스템을 VLAN으로 구성할 수 있습니다. 로컬 네트워크를 VLAN으로 분할하려면 먼저 VLAN 기술을 지원하는 스위치를 구해야 합니다. VLAN 토폴로지 설계에 따라 스위치의 모든 포트가 단일 VLAN이나 여러 VLAN을 서비스하도록 구성할 수 있습니다. 각 스위치 제조업체에 따라 스위치 포트 구성 절차가 달라집니다.

다음 그림에서는 서브넷 주소가 `192.168.84.0`인 LAN을 보여줍니다. 이 LAN은 빨간색, 노란색 및 파란색인 세 개의 VLAN으로 분할됩니다.

그림 6–1 VLAN 세 개가 있는 LAN(Local Area Network)



LAN 192.168.84.0의 연결은 스위치 1과 2에 의해 처리됩니다. 빨간색 VLAN에는 Accounting 작업 그룹의 시스템이 있습니다. Human Resources 작업 그룹의 시스템은 노란색 VLAN에 있습니다. Information Technologies 작업 그룹의 시스템은 파란색 VLAN에 지정됩니다.

VLAN 태그 및 물리적 연결 지점

LAN(Local Area Network)의 각 VLAN은 VLAN 태그 또는 *VLAN ID(VID)*로 식별됩니다. VID는 VLAN 구성 중에 지정됩니다. VID는 각 VLAN에 대해 1-4094 사이의 고유 ID를 제공하는 12비트 식별자입니다. 그림 6–1에서 빨간색 VLAN의 VID는 789, 노란색 VLAN의 VID는 456이며 파란색 VLAN의 VID는 123입니다.

VLAN 지원 스위치를 구성하는 경우 VID를 각 포트에 지정해야 합니다. 포트의 VID는 다음 그림과 같이 포트에 연결되는 인터페이스에 지정된 VID와 동일해야 합니다.

그림 6–2 VLAN을 사용하는 네트워크의 스위치 구성

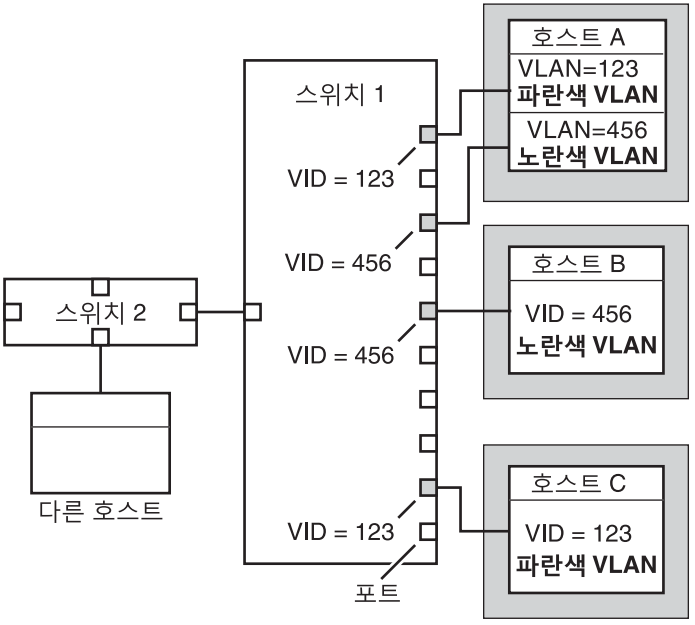


그림 6–2에서는 여러 VLAN에 연결된 여러 호스트를 보여 줍니다. 두 호스트는 같은 VLAN에 속합니다. 이 그림에서 세 호스트의 기본 네트워크 인터페이스는 스위치 1에 연결됩니다. 호스트 A는 파란색 VLAN의 구성원입니다. 따라서 호스트 A의 인터페이스는 VID 123으로 구성됩니다. 이 인터페이스는 VID 123으로 구성된 스위치 1의 포트 1에 연결됩니다. 호스트 B는 VID가 456인 노란색 VLAN의 구성원입니다.

호스트 B의 인터페이스는 VID 456으로 구성된 스위치 1의 포트 5에 연결됩니다. 마지막으로 호스트 C의 인터페이스는 스위치 1의 포트 9에 연결됩니다. 파란색 VLAN은 VID 123으로 구성됩니다.

또한 이 그림에서는 단일 호스트가 여러 VLAN에 속할 수도 있음을 보여 줍니다. 예를 들어 호스트 A에는 호스트 인터페이스를 통해 2개의 VLAN이 구성되어 있습니다. 두번째 VLAN은 VID 456으로 구성되어 있으며 마찬가지로 VID 456으로 구성된 포트 3에 연결되어 있습니다. 따라서 호스트 A는 파란색 VLAN 및 노란색 VLAN의 구성원입니다.

VLAN 구성 중 VLAN의 **물리적 연결 지점**(*PPA*)을 지정해야 합니다. 이 수식을 사용하여 PPA 값을 얻습니다.

driver-name `+ VID * 1000 +` *device-instance*

device-instance 번호는 1000보다 작아야 합니다.

예를 들어, VLAN 456의 일부로 구성하려면 `ce1` 인터페이스에 대해 다음 PPA를 만들어야 합니다.

```
ce + 456 * 1000 + 1= ce456001
```

네트워크의 VLAN 계획

다음 절차를 사용하여 네트워크의 VLAN을 계획할 수 있습니다.

▼ VLAN 구성을 계획하는 방법

1. **로컬 네트워크 토폴로지를 검사하고 VLAN으로 하위 분할하기에 적합한 위치를 확인합니다.**

   이러한 토폴로지의 기본 예는 그림 6–1을 참조하십시오.

2. **VID에 대한 번호 지정 체계를 만들고 각 VLAN에 VID를 지정합니다.**

   주 – VLAN 번호 지정 체계가 네트워크에 이미 있을 수도 있습니다. 이 경우 기존 VLAN 번호 지정 체계에 VID를 만들어야 합니다.

   ---

3. **각 시스템에서 특정 VLAN의 멤버가 될 인터페이스를 결정합니다.**

   a. **시스템에 구성된 인터페이스를 확인합니다.**

      ```
      # dladm show-link
      ```

   b. **시스템의 각 데이터 링크와 연결할 VID를 식별합니다.**

   c. **VLAN으로 구성하려면 각 인터페이스에 대해 PPA를 만듭니다.**

시스템의 모든 인터페이스를 같은 VLAN에서 구성할 필요는 없습니다.

4 **네트워크 스위치에 대한 인터페이스 연결을 확인합니다.**
각 인터페이스의 VID와 각 인터페이스가 연결된 스위치 포트를 확인합니다.

5 **연결된 인터페이스와 동일한 VID로 스위치의 각 포트를 구성합니다.**
구성 지침은 스위치 제조업체의 설명서를 참조하십시오.

VLAN 구성

Oracle Solaris는 이제 다음 인터페이스 유형에서 VLAN을 지원합니다.

- ce
- bge
- xge
- e1000g

레거시 인터페이스 유형 중 ce 인터페이스만 VLAN의 구성원이 될 수 있습니다. 여러 유형의 인터페이스를 같은 VLAN에서 구성할 수 있습니다.

주 – 여러 VLAN을 하나의 IPMP 그룹으로 구성할 수 있습니다. IPMP 그룹에 대한 자세한 내용은 660 페이지 “IPMP 인터페이스 구성”을 참조하십시오.

▼ VLAN을 구성하는 방법

1 **기본 관리자 역할 또는 수퍼 유저로 로그인합니다**
기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **시스템에서 사용 중인 인터페이스 유형을 확인합니다.**

```
# dladm show-link
```

출력에는 사용할 수 있는 인터페이스 유형이 표시됩니다.

```
ce0             type: legacy    mtu: 1500       device: ce0
 ce1             type: legacy    mtu: 1500       device: ce1
 bge0            type: non-vlan  mtu: 1500       device: bge0
 bge1            type: non-vlan  mtu: 1500       device: bge1
 bge2            type: non-vlan  mtu: 1500       device: bge2
```

3 **인터페이스를 VLAN의 일부로 구성합니다.**

```
# ifconfig interface-PPA plumb IP-address up
```

예를 들어, 다음 명령을 사용하여 새 IP 주소 10.0.0.2인 인터페이스 ce1을 VID 123의 VLAN으로 구성합니다.

```
# ifconfig ce123001 plumb 10.0.0.2
up
```

주 – 다른 인터페이스와 마찬가지로 IPv4 및 IPv6 주소를 VLAN에 지정할 수 있습니다.

4 **(선택 사항) 재부트 시에도 VLAN 설정이 유지되도록 하려면 각 인터페이스에 대해 VLAN의 일부로 구성된 `hostname.`*interface-PPA* 파일을 만듭니다.**

```
# cat hostname.interface-PPA
IPv4-address
```

5 **스위치에서 VLAN 태그 처리 및 VLAN 포트를 설정하여 시스템에서 설정한 VLAN과 부합되도록 합니다.**

예 6–3 VLAN 구성

이 예에서는 장치 bge1 및 bge2를 VID 123인 VLAN으로 구성하는 방법을 보여 줍니다.

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
# ifconfig bge123001 plumb 10.0.0.1 up
# ifconfig bge123002 plumb 10.0.0.2 up
# cat hostname.bge123001   10.0.0.1
# cat hostname.bge123002   10.0.0.2
# ifconfig -a
 lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
         inet 127.0.0.1 netmask ff000000
 bge123001: flags=201000803<UP,BROADCAST,MULTICAST,IPv4,CoS> mtu 1500 index 2
         inet 10.0.0.1 netmask ff000000 broadcast 10.255.255.255
         ether 0:3:ba:7:84:5e
bge123002:flags=201000803 <UP,BROADCAST,MULTICAST,IPv4,CoS> mtu 1500 index 3
         inet 10.0.0.2 netmask ff000000 broadcast 10.255.255.255
         ether 0:3:ba:7:84:5e
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4>mtu 1500 index 4
         inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
         ether 0:3:ba:7:84:5e
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
bge123001       type: vlan 123  mtu: 1500       device: bge1
bge123002       type: vlan 123  mtu: 1500       device: bge2
```

링크 통합 개요

주 – 원래 Solaris 10 릴리스 및 이전 버전의 Solaris OS에서는 링크 통합을 지원하지 않습니다. 이러한 이전 Solaris 릴리스의 링크 통합을 만들려면 **Sun Trunking 1.3 Installation and Users Guide**에서 설명한 대로 Sun Trunking을 사용합니다.

Oracle Solaris는 네트워크 인터페이스를 링크 통합으로 구성하는 기능을 지원합니다. **링크 통합**은 단일 논리 장치로 구성된 시스템의 여러 인터페이스로 구성됩니다. **트렁킹**이라고도 하는 링크 통합은 IEEE 802.3ad Link Aggregation Standard (`http://www.ieee802.org/3/index.html`)에서 정의됩니다.

IEEE 802.3ad Link Aggregation Standard는 여러 개의 전이중 이더넷 링크 기능을 단일 논리 링크로 결합하는 방법을 제공합니다. 이 링크 통합 그룹은 실제로 단일 링크인 것처럼 처리됩니다.

다음은 링크 통합의 기능입니다.

- **대역폭 증가** - 여러 링크의 기능이 하나의 논리 링크로 결합됩니다.
- **자동 페일오버/페일백** - 실패한 링크의 트래픽이 통합에서 작동하는 링크로 페일오버됩니다.
- **로드 균형 조정** - 인바운드 및 아웃바운드 트래픽이 소스 및 대상 MAC 또는 IP 주소와 같이 사용자가 선택한 로드 균형 조정 정책에 따라 분산됩니다.
- **중복 지원** - 병렬 통합으로 두 시스템을 구성할 수 있습니다.
- **관리 향상** - 모든 인터페이스가 단일 장치로 관리됩니다.
- **네트워크 주소 풀의 드레인 감소** - 전체 통합에 IP 주소 한 개를 지정할 수 있습니다.

링크 통합 기본 사항

기본 링크 통합 토폴로지에는 물리적 인터페이스 세트가 포함된 단일 통합이 사용됩니다. 다음과 같은 경우 기본 링크 통합을 사용할 수 있습니다.

- 많은 트래픽을 분산하여 응용 프로그램을 실행하는 시스템의 경우, 해당 응용 프로그램의 트래픽에 통합을 전용으로 사용할 수 있습니다.
- IP 주소 공간이 제한적임에도 불구하고 많은 대역폭이 필요한 사이트의 경우 대량의 인터페이스 통합에 대해 IP 주소가 하나만 필요합니다.
- 내부 인터페이스의 존재를 숨겨야 하는 사이트의 경우, 통합의 IP 주소가 외부 응용 프로그램으로부터 해당 인터페이스를 숨깁니다.

그림 6–3에서는 인기 웹 사이트를 호스트하는 서버에 대한 통합을 보여 줍니다. 이 사이트에는 인터넷 고객과 사이트 데이터베이스 서버 간의 질의 트래픽을 위해 더 많은 대역폭이 필요합니다. 보안상, 서버의 개별 인터페이스 존재를 외부 응용

프로그램으로부터 숨겨야 합니다. 솔루션은 IP 주소가 192.168.50.32인 aggr1 통합입니다. 이 통합은 bge0에서 bge2까지의 인터페이스 세 개로 구성됩니다. 이러한 인터페이스는 고객 질의에 대한 응답으로 트래픽을 보내는 데만 사용됩니다. 모든 인터페이스에서 보낸 패킷 트래픽의 송신 주소는 aggr1의 IP 주소인 192.168.50.32입니다.

그림 6–3 기본 링크 통합 토폴로지

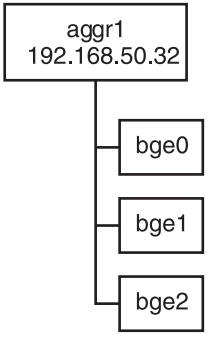
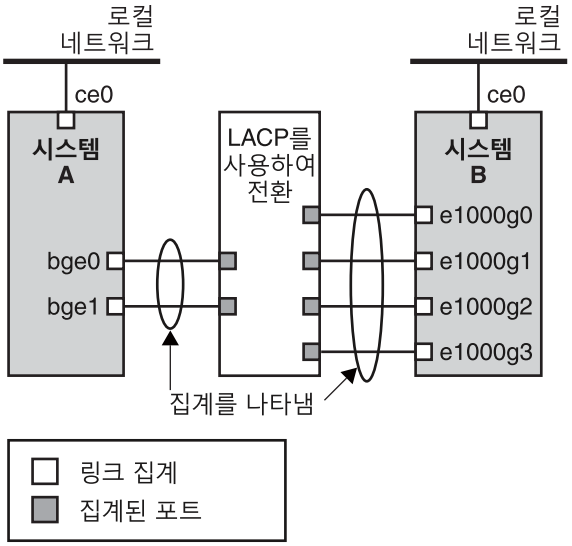


그림 6–4에서는 두 시스템이 포함된 로컬 네트워크를 보여주며, 각 시스템에 통합이 구성되어 있습니다. 두 시스템은 스위치로 연결되어 있습니다. 스위치를 통해 통합을 실행해야 하는 경우 해당 스위치가 통합 기술을 지원해야 합니다. 이 구성 유형은 특히 고가용성과 중복 시스템에 유용합니다.

이 그림에서 시스템 A에는 bge0과 bge1의 두 인터페이스로 구성된 통합이 있습니다. 이러한 인터페이스는 통합된 포트를 통해 스위치에 연결됩니다. 시스템 B에는 e1000g0에서 e1000g3까지 인터페이스 4개로 구성된 통합이 있습니다. 이러한 인터페이스도 스위치의 통합된 포트에 연결됩니다.

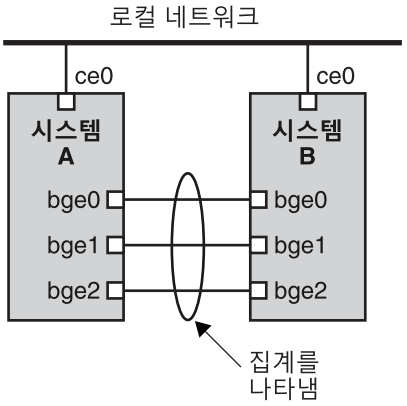
그림 6–4 스위치를 사용한 링크 통합 토폴로지



인접(Back-to-Back) 링크 통합

인접(Back-to-Back) 링크 통합 토폴로지에서는 다음 그림과 같이 케이블을 통해 서로 직접 연결된 별도의 두 시스템이 사용됩니다. 시스템은 병렬 통합을 실행합니다.

그림 6–5 기본 인접(Back-to-Back) 통합 토폴로지



이 그림에서 시스템 A의 `bge0` 장치는 시스템 B의 `bge0`에 직접 연결되어 있고 나머지 장치도 같은 방식으로 연결되어 있습니다. 이 경우 시스템 A와 B가 중복 및 고가용성과

두 시스템 간의 고속 통신을 지원할 수 있습니다. 각 시스템에는 로컬 네트워크 내의 트래픽 플로우에 대해 ce0 인터페이스도 구성되어 있습니다.

인접(Back-to-Back) 링크 통합의 가장 일반적인 응용 프로그램은 미러링된 데이터베이스 서버입니다. 두 서버를 함께 업데이트해야 하므로 상당한 대역폭, 고속 트래픽 플로우 및 안정성이 필요합니다. 인접(Back-to-Back) 링크 통합의 가장 일반적인 사용은 데이터 센터에서 이루어집니다.

정책 및 로드 균형 조정

링크 통합을 사용하려는 경우 송신 트래픽에 대한 정책 정의를 고려해 보십시오. 이 정책은 사용 가능한 통합 링크에 패킷을 배포하여 로드 균형 조정을 설정하는 방법을 지정할 수 있습니다. 통합 정책에 가능한 계층 지정자 및 해당 중요성은 다음과 같습니다.

- **L2** - 각 패킷의 MAC(L2) 헤더를 해싱하여 송신 링크를 결정합니다.
- **L3** - 각 패킷의 IP(L3) 헤더를 해싱하여 송신 링크를 결정합니다.
- **L4** - 각 패킷의 TCP, UDP 또는 기타 ULP(L4) 헤더를 해싱하여 송신 링크를 결정합니다.

이러한 정책의 모든 조합도 유효합니다. 기본 정책은 L4입니다. 자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

통합 모드 및 스위치

통합 토폴로지에 스위치를 통한 연결이 사용되는 경우 스위치가 *LACP(Link Aggregation Control Protocol)*를 지원하는지 여부를 확인해야 합니다. 스위치가 LACP를 지원하는 경우 스위치와 통합에 대해 LACP를 구성해야 합니다. 하지만 LACP를 작동하려는 다음 **모드** 중 하나를 정의할 수 있습니다.

- **Off 모드** - 통합의 기본 모드입니다. *LACPDU*라고 하는 LACP 패킷이 생성되지 않습니다.
- **Active 모드** - 시스템에서 정기적인 간격으로 LACPDU를 생성하며, 이 간격을 사용자가 지정할 수 있습니다.
- **Passive 모드** - 시스템이 스위치로부터 LACPDU를 받는 경우에만 LACPDU를 생성합니다. 통합과 스위치가 모두 Passive 모드로 구성되어 있는 경우 LACPDU를 교환할 수 없습니다.

구문 정보는 dladm(1M) 매뉴얼 페이지 및 스위치 제조업체의 설명서를 참조하십시오.

링크 통합의 요구 사항

링크 통합 구성은 다음 요구 사항에 따라 제한됩니다.

- `dladm` 명령을 사용하여 통합을 구성해야 합니다.
- 연결된 인터페이스는 통합의 구성원이 될 수 없습니다.
- 인터페이스는 GLDv3 유형(`xge`, `e1000g` 및 `bge`)이어야 합니다.
- 통합의 모든 인터페이스가 동일한 속도 및 전이중 모드로 실행되어야 합니다.
- EEPROM 매개변수 `local-mac-address?`에서 MAC 주소의 값을 "true"로 설정해야 합니다. 지침은 141 페이지 "SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법"을 참조하십시오.

▼ 링크 통합을 만드는 방법

시작하기 전에

주 – 링크 통합은 동일한 속도로 작동하는 전이중 P2P 연결에서만 작동합니다. 통합의 인터페이스가 이 요구 사항을 준수하는지 확인하십시오.

통합 토폴로지에 스위치를 사용하는 경우 스위치에서 다음을 수행했는지 확인합니다.

- 통합으로 사용할 포트를 구성했습니다.
- 스위치가 LACP를 지원하는 경우 Active 모드나 Passive 모드로 LACP를 구성했습니다.

1 기본 관리자 역할 또는 수퍼 유저로 로그인합니다

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 시스템에 현재 설치되어 있는 인터페이스를 확인합니다.

```
# dladm show-link
```

3 연결된 인터페이스를 확인합니다.

```
# ifconfig -a
```

4 통합을 만듭니다.

```
# dladm create-aggr -d interface -d interface [...]key
```

interface 통합의 일부가 될 인터페이스의 장치 이름을 나타냅니다.

key 통합을 식별하는 번호입니다. 가장 낮은 키 번호는 1입니다. 0은 키로 허용되지 않습니다.

예를 들면 다음과 같습니다.

```
# dladm create-aggr -d bge0 -d bge1 1
```

5 새로 만든 통합을 구성 및 연결합니다.

```
# ifconfig aggrkey plumb IP-address up
```

예를 들면 다음과 같습니다.

```
# ifconfig aggr1  plumb 192.168.84.14 up
```

6 방금 만든 통합의 상태를 확인합니다.

```
# dladm show-aggr
```

다음 출력이 제공됩니다.

```
key: 1 (0x0001) policy: L4      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby
```

출력에는 key 1, policy L4가 만들어진 통합이 표시됩니다.

7 (선택 사항) 재부트 시에도 링크 통합의 IP 구성이 유지되도록 만듭니다.

a. IPv4 주소의 링크 통합의 경우 /etc/hostname.aggr*key* 파일을 만듭니다. IPv6 기반의 링크 통합의 경우 /etc/hostname6.aggr*key* 파일을 만듭니다.

b. 링크 통합의 IPv4 또는 IPv6 주소를 파일에 입력합니다.

예를 들어, 이 절차에서 만든 통합에 대해 다음 파일을 만듭니다.

```
# vi /etc/hostname.aggr1
192.168.84.14
```

c. 재구성 부트 수행.

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

d. /etc/hostname.aggr*key* 파일에 입력한 링크 통합 구성이 구성되었는지 확인합니다.

```
# ifconfig -a
.
.
aggr1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.84.14 netmask ff000000 broadcast 192.255.255.
```

예 6–4 링크 통합 만들기

이 예에서는 bge0 및 bge1 두 장치의 링크 통합을 만드는 명령과 그 결과 출력을 보여 줍니다.

```
# dladm show-link
ce0             type: legacy    mtu: 1500       device: ce0
ce1             type: legacy    mtu: 1500       device: ce1
bge0            type: non-vlan  mtu: 1500       device: bge0
bge1            type: non-vlan  mtu: 1500       device: bge1
bge2            type: non-vlan  mtu: 1500       device: bge2
# ifconfig -a
lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
        ether 0:3:ba:7:84:5e
# dladm create-aggr -d bge0 -d bge1 1
# ifconfig aggr1 plumb 192.168.84.14 up
# dladm show-aggr
key: 1 (0x0001) policy: L4      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby

# ifconfig -a
lo0: flags=2001000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
ce0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.84.253 netmask ffffff00 broadcast 192.168.84.255
        ether 0:3:ba:7:84:5e
aggr1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.84.14 netmask ff000000 broadcast 192.255.255.255
        ether 0:3:ba:7:84:5e
```

통합에 사용되는 두 인터페이스는 이전에 `ifconfig`로 연결되지 않았습니다.

▼ 통합을 수정하는 방법

이 절차에서는 통합 정의를 다음과 같이 변경하는 방법을 보여줍니다.

- 통합 정책 수정
- 통합 모드 변경

1 기본 관리자 역할 또는 수퍼 유저로 로그인합니다

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 통합을 수정하여 정책을 변경합니다.

 # dladm modify-aggr -P*policy key*

policy 152 페이지 “정책 및 로드 균형 조정”에 설명된 대로 L2, L3 및 L4 정책 중 하나 이상을 나타냅니다.

key 통합을 식별하는 번호입니다. 가장 낮은 키 번호는 1입니다. 0은 키로 허용되지 않습니다.

3 통합의 장치가 연결된 스위치에서 LACP가 실행 중인 경우에는 LACP를 지원하도록 통합을 수정합니다.

스위치가 passive 모드로 LACP를 실행하는 경우 통합에 대해 active 모드를 구성해야 합니다.

```
# dladm modify-aggr -l LACP mode -t timer-value key
```

`-l` *LACP mode* 통합을 실행할 LACP 모드를 나타냅니다. 값은 `active`, `passive` 및 `off`입니다.

`-t` *timer-value* LACP 타이머 값(`short` 또는 `long`)을 나타냅니다.

key 통합을 식별하는 번호입니다. 가장 낮은 키 번호는 1입니다. 0은 키로 허용되지 않습니다.

예 6–5 링크 통합 수정

이 예에서는 통합 `aggr1`의 정책을 L2로 수정한 다음 활성 LACP 모드를 설정하는 방법을 보여 줍니다.

```
# dladm modify-aggr -P L2 1
# dladm modify-aggr -l active -t short 1
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby
```

▼ 통합에서 인터페이스를 제거하는 방법

1 기본 관리자 역할 또는 수퍼 유저로 로그인합니다

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 통합에서 인터페이스를 제거합니다.

```
# dladm remove-aggr -d interface
```

예 6–6 통합에서 인터페이스 제거

이 예에서는 통합 `aggr1`에서 인터페이스를 제거하는 방법을 보여 줍니다.

```
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby
# dladm remove-aggr -d bge1 1
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
```

▼ 통합을 삭제하는 방법

1 기본 관리자 역할 또는 수퍼 유저로 로그인합니다

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 통합을 삭제합니다.

```
# dladm delete-aggr key
```

key 통합을 식별하는 번호입니다. 가장 낮은 키 번호는 1입니다. 0은 키로 허용되지 않습니다.

예 6–7 통합을 삭제하는 방법

이 예에서는 통합 aggr1을 제거하는 방법을 보여 줍니다.

```
# dladm show-aggr
key: 1 (0x0001) policy: L2      address: 0:3:ba:7:84:5e (auto)
     device   address           speed     duplex  link    state
# dladm delete-aggr -d 1
```

▼ 링크 통합에 VLAN을 구성하는 방법

인터페이스에 VLAN을 구성하는 것과 동일한 방식으로 링크 통합에 VLAN을 만들 수도 있습니다. VLAN에 대한 설명은 143 페이지 “VLAN(가상 LAN) 관리”를 참조하십시오. 이 절에서는 VLAN과 링크 통합의 구성을 결합합니다.

시작하기 전에 링크 통합을 만듭니다. 통합에 대해 VLAN을 만들 때 필요한 집계의 key 값을 기억해 두십시오. 링크 통합을 만들려면 153 페이지 “링크 통합을 만드는 방법”을 참조하십시오.

1. **링크 통합이 이전에 만들어져 있으면 해당 통합의 키를 가져옵니다.**

   ```
   # dladm show-aggr
   ```

2. **링크 통합에 대해 VLAN을 만듭니다.**

   ```
   # ifconfig aggrVIDkey plumb
   ```

   구문 설명

   VID VLAN의 ID입니다.

   key VLAN이 만들어진 링크 통합의 키입니다. 키는 3자릿수여야 합니다. 예를 들어, 통합의 키가 1이면 VLAN의 이름에 포함할 키 번호는 001이어야 합니다.

3. **단계 2를 반복하여 통합에 대해 다른 VLAN을 만듭니다.**

4. **유효한 IP 주소로 VLAN을 구성합니다.**

5. **지속 VLAN 구성을 만들려면 IP 주소 정보를 해당 /etc/hostname.*VLAN* 구성 파일에 추가합니다.**

예 6–8 링크 통합에 여러 VLAN 구성

이 예에서는 링크 통합에 VLAN 두 개가 구성됩니다. dladm show-aggr 명령의 출력을 통해 링크 통합의 키가 1임을 알 수 있습니다. VLAN에는 각각 VID 193과 194가 지정됩니다.

```
# dladm show-aggr
key: 1 (0x0001) policy: L4      address: 0:3:ba:7:84:5e (auto)
device   address           speed         duplex  link    state
bge0     0:3:ba:7:b5:a7    1000  Mbps    full    up      attached
bge1     0:3:ba:8:22:3b    0     Mbps    unknown down    standby

# ifconfig aggr193001 plumb
# ifconfig aggr193001 192.168.10.0/24 up

# ifconfig aggr194001 plumb
# ifconfig aggr194001 192.168.20.0/24 up

# vi /etc/hostname.aggr193001
192.168.10.0/24

# vi /etc/hostname.aggr194001
192.168.20.0/24
```

◆ ◆ ◆ 7 장

7 IPv6 네트워크 구성(작업)

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

IPv6에 대한 다양한 유형의 정보는 다음 리소스를 참조하십시오.

- IPv6 개념에 대한 개요: 3 장, “IPv6 소개(개요)”
- IPv6 계획 작업: 4 장, “IPv6 네트워크 계획(작업)”
- IP 터널 사용을 위한 준비: 85 페이지 “네트워크 토폴로지의 터널 계획”
- 참조 정보: Chapter 11, IPv6 세부 개요(참조)

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.

- IPv6에 대해 사용으로 설정된 각 인터페이스에는 이제 연관된 `/etc/hostname6.`*interface* 파일이 포함됩니다(예: `hostname6.dmfe0`).
- Solaris 10 11/06 및 이전 릴리스의 경우 `/etc/inet/ipnodes` 파일이 만들어졌습니다. 설치 후 이 파일에는 일반적으로 IPv6 및 IPv4 루프백 주소만 포함됩니다.

- `/etc/nsswitch.conf` 파일은 IPv6 주소를 사용하여 조회가 가능하도록 수정되었습니다.
- `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.

인터페이스에서 IPv6을 사용으로 설정(작업 맵)

다음 테이블에는 IPv6 인터페이스 구성을 위한 서로 다른 작업이 나열되어 있습니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
Oracle Solaris(으)로 이미 설치된 시스템의 인터페이스에서 IPv6을 사용으로 설정합니다.	Oracle Solaris가 설치된 후 인터페이스에서 IPv6을 사용으로 설정하려면 이 작업을 사용합니다.	160 페이지 "현재 세션에 대해 IPv6 인터페이스를 사용으로 설정하는 방법"
재부트 시 IPv6이 사용으로 설정된 인터페이스가 지속되도록 만듭니다.	인터페이스의 IPv6 주소를 영구적으로 만들려면 이 작업을 사용합니다.	162 페이지 "영구적인 IPv6 인터페이스를 사용으로 설정하는 방법"
IPv6 주소 자동 구성을 해제합니다.	IPv6 주소의 인터페이스 ID 부분을 수동으로 구성해야 하는 경우 이 작업을 사용합니다.	164 페이지 "IPv6 주소 자동 구성을 해제하는 방법"

▼ 현재 세션에 대해 IPv6 인터페이스를 사용으로 설정하는 방법

IPv6 노드로 사용될 모든 시스템의 인터페이스에서 IPv6을 사용으로 설정하여 IPv6 구성 프로세스를 시작하십시오. 처음에 인터페이스는 77 페이지 "IPv6 주소 자동 구성"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 166 페이지 "IPv6 지원 라우터를 구성하는 방법"을 참조하십시오.

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

시작하기 전에 하드웨어 및 소프트웨어 업그레이드, 주소 지정 계획 준비 등 IPv6 네트워크에 대한 계획 작업을 완료합니다. 자세한 내용은 79 페이지 "IPv6 계획(작업 맵)"을 참조하십시오.

1. **예상 IPv6 노드에 기본 관리자 또는 수퍼 유저로 로그인합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2. **인터페이스에서 IPv6을 사용으로 설정합니다.**

   ```
   # ifconfig interface inet6 plumb up
   ```

3. **IPv6 데몬 `in.ndpd`를 시작합니다.**

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

주 – `ifconfig -a6` 명령을 사용해서 노드의 IPv6이 사용으로 설정된 인터페이스 상태를 표시할 수 있습니다.

예 7–1 설치 후 IPv6 인터페이스 사용

이 예는 `gfe0` 인터페이스에서 IPv6을 사용으로 설정하는 방법을 보여줍니다. 시작하기 전에 시스템에 구성된 모든 인터페이스의 상태를 확인하십시오.

```
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1000863 <UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500
           index 2
        inet 172.16.27.74 netmask ffffff00 broadcast 172.16.27.255
        ether 0:3:ba:13:14:e1
```

현재 `gfe0` 인터페이스만 이 시스템에 대해 구성되어 있습니다. 다음과 같이 이 인터페이스에서 IPv6을 사용으로 설정하십시오.

```
# ifconfig qfe0 inet6 plumb up
# /usr/lib/inet/in.ndpd
# ifconfig -a6
```

```
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 0:3:ba:13:14:e1
        inet6 fe80::203:baff:fe13:14e1/10
```

이 예에서는 `qfe0`에서 IPv6이 사용으로 설정되기 전과 후의 시스템 인터페이스 상태를 보여줍니다. `ifconfig`의 `-a6` 옵션은 `qfe0` 및 루프백 인터페이스에 대한 IPv6 정보만 보여줍니다. 출력은 링크 로컬 주소만 `qfe0`, `fe80::203:baff:fe13:14e1/10`에 대해 구성되었음을 나타냅니다. 이 주소는 노드의 로컬 링크에서 어떠한 라우터도 아직 사이트 접두어를 알리지 않음을 나타냅니다.

IPv6이 사용으로 설정된 후에는 `ifconfig -a` 명령을 사용하여 시스템의 모든 인터페이스에 대해 IPv4 및 IPv6 주소를 모두 표시할 수 있습니다.

다음 순서

- IPv6 노드를 라우터로 구성하려면 165 페이지 "IPv6 라우터 구성"으로 이동합니다.
- 재부트 시에 IPv6 인터페이스 구성을 유지 관리하려면 162 페이지 "영구적인 IPv6 인터페이스를 사용으로 설정하는 방법"을 참조하십시오.
- 노드에 대한 주소 자동 구성을 사용 안함으로 설정하려면 164 페이지 "IPv6 주소 자동 구성을 해제하는 방법"을 참조하십시오.
- 노드를 서버로 조정하려면 175 페이지 "서버에서 IPv6 지원 인터페이스 관리"의 제안 사항을 참조하십시오.

▼ 영구적인 IPv6 인터페이스를 사용으로 설정하는 방법

이 절차에서는 이후 재부트 시에도 영구적으로 보존되는 자동 구성된 IPv6 주소로 IPv6 인터페이스를 사용으로 설정하는 방법에 대해 설명합니다.

주 – 인터페이스가 현재 IPv6 접두어를 알리는 라우터와 동일한 링크에 있는 경우, 자동 구성된 주소의 일부로 해당 사이트의 접두어를 얻습니다. 자세한 내용은 166 페이지 "IPv6 지원 라우터를 구성하는 방법"을 참조하십시오.

---

1 IPv6 노드에 기본 관리자 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 설치 후 추가된 인터페이스에 대한 IPv6 주소를 만듭니다.

a. 구성 파일을 만듭니다.

```
# touch /etc/hostname6.interface
```

b. 주소를 구성 파일에 추가합니다.

ipv6-address up
...

3 정적 IPv6 기본 경로를 만듭니다.

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

4 (선택 사항) 노드의 인터페이스 변수에 대한 매개변수를 정의하는 /etc/inet/ndpd.conf 파일을 만듭니다.

호스트의 인터페이스에 대해 임시 주소를 만들어야 하는 경우 169 페이지 "인터페이스에 대해 임시 주소 사용"을 참조하십시오. /etc/inet/ndpd.conf에 대한 자세한 내용은 ndpd.conf(4) 매뉴얼 페이지 및 245 페이지 "ndpd.conf 구성 파일"을 참조하십시오.

5 노드를 재부트합니다.

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

재부트 프로세스가 라우터 검색 패킷을 전송합니다. 라우터가 사이트 접두어로 응답하면 노드가 전역 IPv6 주소를 포함하는 해당 /etc/hostname6.*interface* 파일로 인터페이스를 구성할 수 있습니다. 그렇지 않으면 IPv6이 사용으로 설정된 인터페이스가 링크 로컬 주소만 사용하여 구성됩니다. 재부트하면 in.ndpd 및 기타 네트워크 데몬도 IPv6 모드로 다시 시작됩니다.

예 7–2 재부트 시 IPv6 인터페이스를 영구적으로 만들기

이 예에서는 재부트 시에 qfe0 인터페이스에 대한 IPv6 구성을 영구적으로 만드는 방법을 보여줍니다. 이 예에서는 로컬 링크의 라우터가 사이트 접두어 및 서브넷 ID 2001:db8:3c4d:15/64를 알립니다.

첫번째, 시스템의 인터페이스 상태를 확인합니다.

```
# ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1000863 <UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4> mtu 1500
           index 2
          inet 172.16.27.74 netmask ffffff00 broadcast 172.16.27.255
          ether 0:3:ba:13:14:e1

# touch /etc/hostname6.qfe0
# vi /etc/hostname6.qfe0
inet6 fe80::203:baff:fe13:1431/10 up
addif 2001:db8:3c4d:15:203:baff:fe13:14e1/64 up

# route -p add -inet6 default fe80::203:baff:fe13:1431
# reboot -- -r
```

구성한 IPv6 주소가 여전히 qfe0 인터페이스에 적용되어 있는지 확인합니다.

```
# ifconfig -a6
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
       ether 0:3:ba:13:14:e1
       inet6 fe80::203:baff:fe13:14e1/10
 qfe0:1: flags=2180841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500
          index 2
        inet6 2001:db8:3c4d:15:203:baff:fe13:14e1/64
```

`ifconfig -a6`의 출력은 `qfe0`에 대한 두 항목을 보여줍니다. 표준 `qfe0` 항목에는 MAC 주소 및 링크 로컬 주소가 포함됩니다. 보조 항목인 `qfe0:1`은 `qfe0` 인터페이스에 추가 IPv6 주소에 대해 의사 인터페이스가 만들어졌음을 나타냅니다. 새로운 전역 IPv6 주소 `2001:db8:3c4d:15:203:baff:fe13:14e1/64`에는 로컬 라우터가 알린 사이트 접두어와 서브넷 ID가 포함됩니다.

다음 순서

- 새 IPv6 노드를 라우터로 구성하려면 165 페이지 "IPv6 라우터 구성"으로 이동합니다.
- 노드에 대한 주소 자동 구성을 사용 안함으로 설정하려면 164 페이지 "IPv6 주소 자동 구성을 해제하는 방법"을 참조하십시오.
- 새 노드를 서버로 조정하려면 175 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 172 페이지 "IPv6 토큰 구성"에 설명된 것과 같이, 특히 토큰을 수동으로 구성하려는 경우 주소 자동 구성을 해제할 수 있습니다.

1 IPv6 노드에 기본 관리자 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 노드에 대한 `/etc/inet/ndpd.conf` 파일을 만듭니다.

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

if-variable-name `StatelessAddrConf false`

`/etc/inet/ndpd.conf`에 대한 자세한 내용은 `ndpd.conf(4)` 매뉴얼 페이지 및 245 페이지 "ndpd.conf 구성 파일"을 참조하십시오.

3 변경 사항으로 IPv6 데몬을 업데이트합니다.

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

IPv6 라우터 구성

네트워크에서 IPv6을 구성하는 첫번째 단계는 라우터에서 IPv6을 구성하는 것입니다. 라우터 구성에는 이 섹션에서 설명되는 여러 개의 고유 작업들이 포함됩니다. 시스템 요구 사항에 따라 이러한 작업 중 일부 또는 모두를 수행해야 할 수 있습니다.

IPv6 라우터 구성(작업 맵)

IPv6 네트워크를 구성하려면 다음 테이블에서 표시된 순서에 따라 다음 작업들을 수행합니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
1. IPv6 구성을 시작하기 전에 필수 필요 조건을 완료했는지 확인합니다.	IPv6이 사용으로 설정된 라우터를 구성하려면 먼저 IPv6이 사용으로 설정된 인터페이스에서 계획 작업 및 Oracle Solaris 설치를 완료해야 합니다.	4 장, “IPv6 네트워크 계획(작업)” 및 159 페이지 “IPv6 인터페이스 구성”.
2. 라우터를 구성합니다.	네트워크의 사이트 접두어를 정의합니다.	166 페이지 “IPv6 지원 라우터를 구성하는 방법”
3. 라우터에서 터널 인터페이스를 구성합니다.	라우터에서 수동 터널 또는 6to4 터널 인터페이스를 설정합니다. 로컬 IPv6 네트워크가 다른 격리된 IPv6 네트워크와 통신하려면 터널이 필요합니다.	■ 179 페이지 “6to4 터널을 구성하는 방법” ■ 177 페이지 “IPv6 Over IPv4 터널을 수동으로 구성하는 방법” ■ 178 페이지 “IPv6 Over IPv6 터널을 수동으로 구성하는 방법” ■ 178 페이지 “IPv4 Over IPv6 터널을 구성하는 방법”
4. 네트워크에서 스위치를 구성합니다.	네트워크 구성에 스위치가 포함된 경우 구성 프로세스의 현재 시점에서 IPv6용으로 구성합니다.	스위치 제조업체 설명서를 참조하십시오.
5. 네트워크에서 허브를 구성합니다.	네트워크 구성에 허브가 포함된 경우 구성 프로세스의 현재 시점에서 IPv6용으로 구성합니다.	허브 제조업체 설명서를 참조하십시오.
6. IPv6에 대한 네트워크 이름 서비스를 구성합니다.	라우터가 IPv6용으로 구성된 후 IPv6 주소를 인식할 수 있도록 기본 이름 서비스(DNS, NIS 또는 LDAP)를 구성합니다.	184 페이지 “DNS에 IPv6 주소를 추가하는 방법”

작업	설명	수행 방법
7. (선택 사항) 호스트 및 서버에서 IPv6이 사용으로 설정된 인터페이스에 대한 주소를 수정합니다.	IPv6 라우터 구성 후에는 IPv6이 사용으로 설정된 호스트 및 서버에서 추가 항목을 수정합니다.	169 페이지 "호스트 및 서버에 대해 IPv6 인터페이스 구성 수정"
IPv6을 지원하도록 응용 프로그램을 구성합니다.	다른 응용 프로그램에는 IPv6 지원을 위해 다른 작업이 필요할 수 있습니다.	응용 프로그램 설명서를 참조하십시오.

▼ IPv6 지원 라우터를 구성하는 방법

이 절차에서는 Oracle Solaris 설치 중에 라우터의 모든 인터페이스가 IPv6용으로 구성되었다고 가정합니다.

1 IPv6 라우터가 될 시스템에서 기본 관리자 역할을 맡거나 수퍼 유저로 전환합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 라우터에서 설치 중에 IPv6용으로 구성된 인터페이스를 검토합니다.

```
# ifconfig -a
```

출력에서 IPv6용으로 구성하려는 인터페이스가 이제 링크 로컬 데이터베이스로 연결되었는지 확인합니다. `ifconfig -a`의 다음 샘플 명령 출력은 라우터의 인터페이스에 대해 구성된 IPv4 및 IPv6 주소를 보여줍니다.

```
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.16.26.232 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:15
dmfe1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4 mtu 1500 index 3
        inet 172.16.26.220 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:16
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
dmfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 0:3:ba:11:b1:15
        inet6 fe80::203:baff:fe11:b115/10
dmfe1: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 3
        ether 0:3:ba:11:b1:16
        inet6 fe80::203:baff:fe11:b116/10
```

출력에서는 또한 기본 네트워크 인터페이스 `dmfe0` 및 추가 인터페이스 `dmfe1`이 설치 중에 IPv6 링크 로컬 주소 `fe80::203:baff:fe11:b115/10` 및 `fe80::203:baff:fe11:b116/10`으로 구성되었음을 보여줍니다.

3 라우터의 모든 인터페이스에서 IPv6 패킷 전송을 구성합니다.

Solaris 10 11/03 및 이전 릴리스의 경우 다음 명령을 사용합니다.

```
# routeadm -e ipv6-forwarding -u
```

패킷 전달을 사용으로 설정하려면 다음 중 하나를 사용합니다.

- 다음과 같이 `routeadm` 명령을 사용합니다.

  ```
  # routeadm -e ipv6-forwarding -u
  ```

- 다음과 같이 다음 SMF(서비스 관리 기능) 명령을 사용합니다.

  ```
  # svcadm enable ipv6-forwarding
  ```

4 경로 지정 데몬을 시작합니다.

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

Solaris 10 11/06 및 이전 릴리스의 경우 다음 명령을 입력하여 `in.ripngd`를 시작합니다.

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

- `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.

5 `/etc/inet/ndpd.conf` 파일을 만듭니다.

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

변수 및 허용되는 값 목록은 245 페이지 “ndpd.conf 구성 파일” 및 `ndpd.conf(4)` 매뉴얼 페이지를 참조하십시오.

6 `/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.ndpd`에 지시합니다.

7 `/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 dmfe0 및 dmfe1 인터페이스를 통해 사이트 접두어 2001:0db8:3c4d::/48을 알리도록 라우터를 구성합니다.

```
ifdefault AdvSendAdvertisements true
prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on

if dmfe0 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:15::0/64 dmfe0

if dmfe1 AdvSendAdvertisements 1
prefix 2001:0db8:3c4d:16::0/64 dmfe1
```

8 **시스템을 재부트합니다.**

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

예 7–3 IPv6 인터페이스를 표시하는 ifconfig 출력

다음 예는 165 페이지 "IPv6 라우터 구성" 절차를 완료하면 표시되는 것과 같은, ifconfig -a 명령의 출력을 보여줍니다.

```
lo0: flags=1000849 <UP LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.16.15.232 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:15
dmfe1: flags=1000843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4 mtu 1500 index 3
        inet 172.16.16.220 netmask ffffff00 broadcast 172.16.26.255
        ether 0:3:ba:11:b1:16
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
dmfe0: flags=2100841 <UP,RUNNING,MULTICAST,ROUTER,IPv6> mtu 1500 index 2
        ether 0:3:ba:11:b1:15
        inet6 fe80::203:baff:fe11:b115/10
dmfe0:1: flags=2180841 <UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6> mtu 1500
          index 2
        inet6 2001:db8:3c4d:15:203:baff:fe11:b115/64
dmfe1: flags=2100841 <UP,RUNNING,MULTICAST,ROUTER,IPv6> mtu 1500 index 3
        ether 0:3:ba:11:b1:16
        inet6 fe80::203:baff:fe11:b116/10
dmfe1:1: flags=2180841 <UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6> mtu 1500
           index 3
        inet6 2001:db8:3c4d:16:203:baff:fe11:b116/64
```

이 예에서 IPv6용으로 구성된 각 인터페이스가 이제 두 개의 주소를 사용합니다. 인터페이스 이름을 포함하는 항목(예: dmfe0)에는 해당 인터페이스에 대한 링크 로컬 주소가 표시됩니다. *interface:n* 형식의 항목(예: dmfe0:1)은 전역 IPv6 주소를 보여줍니다. 이 주소에는 인터페이스 ID 이외에도 /etc/ndpd.conf 파일에 구성된 사이트 접두어가 포함됩니다.

참조 ■ IPv6 네트워크 토폴로지에서 식별한 라우터의 터널을 구성하려면 176 페이지 "IPv6 지원을 위한 터널 구성"을 참조하십시오.

- 네트워크에서 스위치 및 허브 구성에 대한 자세한 내용은 제조업체의 설명서를 참조하십시오.
- IPv6 호스트를 구성하려면 169 페이지 “호스트 및 서버에 대해 IPv6 인터페이스 구성 수정”을 참조하십시오.
- 서버에서 IPv6 지원을 향상시키려면 175 페이지 “서버에서 IPv6 지원 인터페이스 관리”를 참조하십시오.
- IPv6 명령, 파일 및 데몬에 대한 자세한 내용은 245 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 77 페이지 “Stateless 자동 구성 개요”에 설명된 대로 IPv6이 사용으로 설정된 인터페이스에 대해 주소 자동 구성을 사용해야 합니다. 그러나 이 단원의 작업에 설명된 것과 같이, 필요한 경우 인터페이스의 IPv6 주소를 수정할 수 있습니다.

IPv6 인터페이스 구성 수정(작업 맵)

다음 테이블에는 기존 IPv6 네트워크를 수정하기 위한 서로 다른 작업이 나열되어 있습니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
IPv6 주소 자동 구성을 해제합니다.	IPv6 주소의 인터페이스 ID 부분을 수동으로 구성해야 하는 경우 이 작업을 사용합니다.	164 페이지 “IPv6 주소 자동 구성을 해제하는 방법”
호스트에 대해 임시 주소를 만듭니다.	주소의 하위 64비트로 사용되는 임의로 만들어진 임시 주소를 구성하여 호스트의 인터페이스 ID를 숨깁니다.	170 페이지 “임시 주소를 구성하는 방법”
시스템의 인터페이스 ID에 대한 토큰을 구성합니다.	IPv6 주소에서 인터페이스 ID로 사용할 64비트 토큰을 만듭니다.	173 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”

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 IPv6 호스트에 기본 관리자 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 필요한 경우 호스트의 인터페이스에서 IPv6을 사용으로 설정합니다.

160 페이지 "현재 세션에 대해 IPv6 인터페이스를 사용으로 설정하는 방법"을 참조하십시오.

3 /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**

4 (선택 사항) 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.

5 (선택 사항) 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 210 페이지 “기본 주소 선택 관리”를 참조하십시오.

---

6 (선택 사항) 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.

7 in.ndpd 데몬의 구성을 변경합니다.

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

8 예 7–5에 표시된 것처럼 ifconfig -a6 명령을 실행하여 임시 주소가 만들어졌는지 확인합니다.

ifconfig의 출력에는 인터페이스 정의와 동일한 라인에 TEMPORARY라는 단어가 포함되어야 합니다.

예 7–4 /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 7–5 임시 주소가 사용으로 설정된 ifconfig -a6 명령 출력

이 예는 임시 주소가 생성된 후 `ifconfig` 명령의 출력을 보여줍니다.

```
# ifconfig -a6
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
     inet6 ::1/128
hme0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
     ether 8:0:20:b9:4c:54
     inet6 fe80::a00:20ff:feb9:4c54/10
hme0:1: flags=2080841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
     inet6 2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
hme0:2: flags=802080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6,TEMPORARY> mtu 1500 index 2
      inet6 2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

`hme0:2` 인터페이스 다음의 라인에는 `TEMPORARY` 단어가 포함되어 있습니다. 이 대상은 `2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64`에 임시 인터페이스 ID가 포함되었음을 나타냅니다.

참조

- IPv6 주소에 대한 이름 서비스 지원을 설정하려면 184 페이지 “IPv6용 이름 서비스 지원 구성”을 참조하십시오.
- 서버에 대해 IPv6 주소를 구성하려면 173 페이지 “사용자 지정 IPv6 토큰을 구성하는 방법”을 참조하십시오.
- IPv6 노드에 대한 작업을 모니터링하려면 8 장, “TCP/IP 네트워크 관리(작업)”를 참조하십시오.

IPv6 토큰 구성

IPv6 주소의 64비트 인터페이스 ID를 **토큰**이라고 합니다. 70 페이지 “IPv6 주소 지정 개요”를 참조하십시오. 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 토큰으로 구성하려는 인터페이스가 연결되었는지 확인합니다.

해당 IPv6 주소에 대해 토큰을 구성하려면 먼저 인터페이스를 연결해야 합니다.

```
# ifconfig a6
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 0:3:ba:13:14:e1
        inet6 fe80::203:baff:fe13:14e1/10
```

이 출력에서는 네트워크 인터페이스 `qfe0`이 연결되었고 링크 로컬 주소 `fe80::203:baff:fe13:14e1/10`이 포함되었음을 보여줍니다. 이 주소는 설치 중에 자동으로 구성되었습니다.

2 노드 인터페이스에 대한 토큰으로 사용할 64비트 16진수를 하나 이상 만듭니다. 토큰 예는 74 페이지 "링크 로컬 유니캐스트 주소"를 참조하십시오.

3 토큰을 사용하여 각 인터페이스를 구성합니다.

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

`ifconfig` *interface* `inet6 token` *address/64*

예를 들어 다음 명령으로 토큰을 포함하는 `qfe0` 인터페이스를 구성할 수 있습니다.

```
# ifconfig qfe0 inet6 token ::1a:2b:3c:4d/64
```

사용자가 지정한 토큰을 포함할 모든 인터페이스에 대해 이 단계를 반복합니다.

4 (선택 사항) 재부트 시 새로운 IPv6 주소가 지속되도록 만듭니다.

a. 토큰으로 구성한 각 인터페이스에 대해 `/etc/hostname6.`*interface* 파일을 편집하거나 만듭니다.

b. 각 `/etc/hostname.6` *interface* 파일의 하단에 다음 텍스트를 추가합니다.

`token ::`*`token-name`*`/64`

예를 들어, `/etc/hostname6.`*`interface`* 파일의 하단에 다음 텍스트를 추가할 수 있습니다.

```
token ::1a:2b:3c:4d/64
```

시스템이 재부트된 다음 `/etc/hostname6.` *interface* 파일에서 구성한 토큰이 인터페이스의 IPv6 주소에 적용됩니다. 이 IPv6 주소는 이후 재부트 시에도 지속됩니다.

5 변경 사항으로 IPv6 데몬을 업데이트합니다.

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

예 7–6 IPv6 인터페이스에서 사용자 지정 토큰 구성

다음 예에서 `bge0:1` 인터페이스에는 자동 구성된 IPv6 주소가 포함됩니다. 서브넷 접두어 `2001:db8:3c4d:152:/64`는 노드의 로컬 링크에 있는 라우터로 알려집니다. 인터페이스 ID `2c0:9fff:fe56:8255`는 `bge0:1`의 MAC 주소로부터 생성됩니다.

```
# ifconfig -a6
lo0: flags=2002000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
bge0: flags=2100801 <UP,MULTICAST,IPv6> mtu 1500 index 5
        inet6 fe80::2c0:9fff:fe56:8255/10
        ether 0:c0:9f:56:82:55
bge0:1: flags=2180801 <UP, MULTICAST,ADDRCONF,IPv6>mtu 1500 index 5
        inet6 2001:db8:3c4d:152:c0:9fff:fe56:8255/64
# ifconfig bge0 inet6 token ::1a:2b:3c:4d/64
# vi /etc/hostname6.bge0
token ::1a:2b:3c:4d/64
# pkill -HUP in.ndpd
# ifconfig -a6
lo0: flags=2002000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
bge0: flags=2100801 <UP,MULTICAST,IPv6> mtu 1500 index 5
        inet6 fe80::2c0:9fff:fe56:8255/10
        ether 0:c0:9f:56:82:55
bge0:1: flags=2180801 <UP, MULTICAST,ADDRCONF,IPv6>mtu 1500 index 5
        inet6 2001:db8:3c4d:152:1a:2b:3c:4d/64
```

토큰이 구성되면 `bge0:1`의 두번째 상태 라인에 있는 전역 주소에 이제 해당 인터페이스 ID에 대해 구성된 `1a:2b:3c:4d`가 포함됩니다.

참조
- 서버의 IPv6 주소로 이름 서비스를 업데이트하려면 184 페이지 “IPv6용 이름 서비스 지원 구성”을 참조하십시오.
- 서버 성능을 모니터링하려면 8 장, “TCP/IP 네트워크 관리(작업)”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

시작하기 전에 다음 절차는 다음 조건을 가정합니다.

- Oracle Solaris가 이미 서버에 설치되어 있습니다.
- Oracle Solaris 설치 중 또는 설치 후에 159 페이지 "IPv6 인터페이스 구성"의 절차에 따라 서버 인터페이스에서 IPv6을 사용으로 설정했습니다.

적용 가능한 경우 IPv6을 지원하도록 응용 프로그램 소프트웨어를 업그레이드합니다. IPv4 프로토콜 스택에서 실행되는 여러 응용 프로그램은 IPv6에서도 성공적으로 실행됩니다. 자세한 내용은 84 페이지 "IPv6을 지원하도록 네트워크 서비스를 준비하는 방법"을 참조하십시오.

1 서버에서 기본 관리자 역할을 맡거나 수퍼 유저로 전환합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 서버와 동일한 링크에 있는 라우터에서 IPv6 서브넷 접두어가 구성되었는지 확인합니다.

자세한 내용은 165 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.

- 교체하지 않으려는 IPv6이 사용으로 설정된 인터페이스의 경우에는 77 페이지 "IPv6 주소 자동 구성"에서 소개하는 자동 구성된 IPv6 주소를 사용합니다.
- 로컬 네트워크 외부에 익명으로 표시되어야 하는 IPv6 지원 인터페이스의 경우, 무작위로 생성된 토큰을 인터페이스 ID로 사용합니다. 지침 및 예제는 170 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
- 정기적으로 교체하려는 IPv6 기반 인터페이스의 경우, 인터페이스 ID에 대한 토큰을 만듭니다. 지침 및 예제는 173 페이지 "사용자 지정 IPv6 토큰을 구성하는 방법"을 참조하십시오.

IPv6 지원을 위한 터널 구성 작업(작업 맵)

다음 테이블에는 서로 다른 유형의 IPv6 터널을 구성하기 위한 서로 다른 작업이 나열됩니다. 이 표에는 수행할 각 작업에 대한 설명과 작업을 수행할 특정 단계가 자세히 설명된 현재 설명서의 절을 제공합니다.

작업	설명	수행 방법
IPv6 over IPv4 터널을 수동으로 구성합니다.	IPv4 네트워크에서 IPv6 터널을 수동으로 만드는, 대부분 IPv4로 구성된 대규모 엔터프라이즈 네트워크 내에서 원격 IPv6 네트워크에 연결하기 위한 솔루션입니다.	177 페이지 "IPv6 Over IPv4 터널을 수동으로 구성하는 방법"
IPv6 over IPv6 터널을 수동으로 구성합니다.	IPv6 네트워크에서 IPv6 터널을 수동으로 구성합니다. 일반적으로 대규모 엔터프라이즈 네트워크에서 사용됩니다.	178 페이지 "IPv6 Over IPv6 터널을 수동으로 구성하는 방법"
IPv4 over IPv6 터널을 수동으로 구성합니다.	IPv6 네트워크에서 IPv4 터널을 수동으로 구성합니다. IPv4 및 IPv6 네트워크를 모두 포함하는 대규모 네트워크에 유용합니다.	178 페이지 "IPv4 Over IPv6 터널을 구성하는 방법"
IPv6 over IPv4 터널을 자동으로 구성합니다(6to4 터널).	인터넷을 통해 외부 IPv6 사이트에 연결하기 위한 솔루션인 자동 6to4 터널을 만듭니다.	179 페이지 "6to4 터널을 구성하는 방법"
6to4 라우터와 6to4 릴레이 라우터 간 터널을 구성합니다.	`6to4relay` 명령을 실행하여 6to4 릴레이 라우터에 대해 터널을 사용으로 설정합니다.	182 페이지 "6to4 릴레이 라우터에 대한 6to4 터널을 구성하는 방법"

IPv6 지원을 위한 터널 구성

IPv6 네트워크는 대규모 IPv4 환경 내에서 격리된 엔티티인 경우가 많습니다. IPv6 네트워크의 노드는 엔터프라이즈 내에서 또는 원격으로 격리된 IPv6 네트워크의 노드와 통신해야 할 수 있습니다. 일반적으로 IPv6 호스트는 터널 끝점으로도 작동할 수 있지만 IPv6 라우터 간에 터널을 구성할 수 있습니다. 터널 계획 정보는 85 페이지 "네트워크 토폴로지의 터널 계획"을 참조하십시오.

IPv6 네트워크에 대해 구성된 터널을 자동 또는 수동으로 설정할 수 있습니다. Oracle Solaris IPv6 구현에는 다음과 같은 유형의 터널 캡슐화가 지원됩니다.

- IPv6 over IPv4 터널
- IPv6 over IPv6 터널

- IPv4 over IPv6 터널
- 6to4 터널

터널에 대한 개념 설명은 266 페이지 “IPv6 터널”을 참조하십시오.

▼ IPv6 Over IPv4 터널을 수동으로 구성하는 방법

이 절차에서는 IPv4 네트워크를 통해 IPv6 노드에서 원격 IPv6 노드로 터널을 설정하는 방법에 대해 설명합니다.

1 **로컬 터널 끝점에 기본 관리자 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **`/etc/hostname6.ip.tun`*n* 파일을 만듭니다.**

여기서 *n*은 첫번째 터널을 나타내는 0부터 시작하는 터널 번호를 나타냅니다. 그런 후 다음 하위 단계를 수행하여 항목을 추가합니다.

a. **터널 소스 주소 및 터널 대상 주소를 추가합니다.**

`tsrc` *IPv4-source-address* `tdst` *IPv4-destination-address* `up`

b. **(선택 사항) 소스 IPv6 주소 및 대상 IPv6 주소에 대한 논리적 인터페이스를 추가합니다.**

`addif` *IPv6-source-address* *IPv6-destination-address*

이 인터페이스의 주소를 자동으로 구성하려면 이 하위 단계를 생략합니다. 터널의 링크 로컬 주소는 구성할 필요가 없습니다.

3 **시스템을 재부트합니다.**

4 **터널의 반대쪽 끝점에서 이 작업을 반복합니다.**

예 7–7 수동 IPv6 Over IPv4 터널에 대한 `/etc/hostname6.ip.tun` 파일의 항목

이 샘플 `/etc/hostname6.ip.tun` 파일은 전역 소스 주소 및 전역 대상 주소가 수동으로 구성된 터널을 보여줍니다.

```
tsrc 192.168.8.20 tdst 192.168.7.19 up
addif 2001:db8:3c4d:8::fe12:528 2001:db8:3c4d:7:a00:20ff:fe12:1234 up
```

▼ IPv6 Over IPv6 터널을 수동으로 구성하는 방법

이 절차에서는 IPv6 네트워크를 통해 IPv6 노드에서 원격 IPv6 노드로 터널을 설정하는 방법에 대해 설명합니다.

1 로컬 터널 끝점에 기본 관리자 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 `/etc/hostname6.ip6.tun` *n* 파일을 만듭니다.

*n*에 대해 0, 1, 2 등의 값을 사용합니다. 그런 후 다음 하위 단계를 수행하여 항목을 추가합니다.

a. 터널 소스 주소 및 터널 대상 주소를 추가합니다.

`tsrc` *IPv6-source-address* `tdst` *IPv6-destination-address*
IPv6-packet-source-address *IPv6-packet-destination-address* `up`

b. (선택 사항) 소스 IPv6 주소 및 대상 IPv6 주소에 대한 논리적 인터페이스를 추가합니다.

`addif` *IPv6-source-address* *IPv6-destination-address* `up`

이 인터페이스의 주소를 자동으로 구성하려면 이 단계를 생략합니다. 터널의 링크 로컬 주소는 구성할 필요가 없습니다.

3 시스템을 재부트합니다.

4 터널의 반대쪽 끝점에서 이 절차를 반복합니다.

예 7–8 IPv6 Over IPv6 터널에 대한 `/etc/hostname6.ip6.tun` 파일의 항목

이 예에서는 IPv6 over IPv6 터널에 대한 항목을 보여줍니다.

```
tsrc 2001:db8:3c4d:22:20ff:0:fe72:668c tdst 2001:db8:3c4d:103:a00:20ff:fe9b:a1c3
fe80::4 fe80::61 up
```

▼ IPv4 Over IPv6 터널을 구성하는 방법

이 절차에서는 IPv6 네트워크를 통해 두 개의 IPv4 호스트 사이에 터널을 구성하는 방법에 대해 설명합니다. 회사 네트워크가 이기종으로 구성되었고 IPv4 서브넷과 구분되는 IPv6 서브넷을 사용하는 경우 이 절차를 따르십시오.

1. **로컬 IPv4 터널 끝점에 기본 관리자 또는 수퍼 유저로 로그인합니다.**
   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2. **`/etc/hostname.ip6.tun`*n* 파일을 만듭니다.**
   *n*에 대해 0, 1, 2 등의 값을 사용합니다. 그런 후 다음 단계를 수행하여 항목을 추가합니다.

   a. **터널 소스 주소 및 터널 대상 주소를 추가합니다.**
      `tsrc` *IPv6-source-address* `tdst` *IPv6-destination-address*

   b. **(선택 사항) 소스 IPv6 주소 및 대상 IPv6 주소에 대한 논리적 인터페이스를 추가합니다.**
      `addif` *IPv6-source-address* *IPv6-destination-address* `up`

3. **로컬 호스트를 재부트합니다.**

4. **터널의 반대쪽 끝점에서 이 절차를 반복합니다.**

예 7–9 IPv4 Over IPv6 터널에 대한 `/etc/hostname6.ip6.tun`의 항목

이 예에서는 IPv4 over IPv6 터널에 대한 항목을 보여줍니다.

```
tsrc 2001:db8:3c4d:114:a00:20ff:fe72:668c tdst 2001:db8:3c4d:103:a00:20ff:fe9b:a1c3
10.0.0.4 10.0.0.61 up
```

▼ 6to4 터널을 구성하는 방법

IPv6 네트워크가 원격 IPv6 네트워크와 통신해야 하는 경우 자동 6to4 터널을 사용하는 것이 좋습니다. 6to4 터널 구성 프로세스에는 경계 라우터를 *6to4* 라우터로 구성하는 과정이 포함됩니다. 6to4 라우터는 네트워크와 원격 IPv6 네트워크의 끝점 라우터 사이에 있는 6to4 터널의 끝점으로 작동합니다.

시작하기 전에 IPv6 네트워크에서 6to4 경로 지정을 구성하기 전에 다음 작업이 수행되어 있어야 합니다.

- 169 페이지 "호스트 및 서버에 대해 IPv6 인터페이스 구성 수정"에 설명된 대로 해당 6to4 사이트의 모든 적합한 노드에서 IPv6 구성해야 합니다.
- 6to4 라우터로 지정할 IPv4 네트워크에 연결된 라우터를 하나 이상 선택해야 합니다.
- IPv4 네트워크에서 해당 6to4 라우터의 인터페이스에 대해 전역으로 고유한 IPv4 주소를 구성해야 합니다. IPv4 주소는 정적이어야 합니다.

주 – 12 장, “DHCP 정보(개요)”에 설명된 대로 동적으로 할당된 IPv4 주소는 사용하지 마십시오. 동적으로 할당된 전역 주소는 시간이 지난 후 변경되어 IPv6 주소 지정 계획에 부정적인 영향을 줄 수 있습니다.

1 해당 6to4 라우터에 기본 관리자 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 `/etc/hostname6.ip.6to4tun0` 파일을 만들어서 라우터에 6to4 의사 인터페이스를 구성합니다.

- 서브넷 ID=0 및 호스트 ID=1의 권장 방식을 사용할 경우 `/etc/hostname6.ip.6to4tun0`에 대해 짧은 형식을 사용합니다.

  `tsrc` *IPv4-address* `up`

- 서브넷 ID 및 호스트 ID에 대해 다른 방식을 사용하려면 `/etc/hostname6.ip.6to4tun0`에 대해 긴 형식을 사용합니다.

  `tsrc` *IPv4-address* `2002:`*IPv4-address*`:`*subnet-ID*`:`*interface-ID*`:/64 up`

`/etc/hostname6.ip.6to4tun0`에 필요한 매개변수는 다음과 같습니다.

`tsrc`	이 인터페이스가 터널 소스로 사용됨을 나타냅니다.
IPv4-address	점으로 구분된 10진수 형식으로 물리적 인터페이스에서 구성된 IPv4 주소를 6to4 의사 인터페이스가 되도록 지정합니다.

남은 매개변수는 선택 사항입니다. 하지만 선택적 매개변수를 하나 지정할 경우에는 모든 선택적 매개변수를 지정해야 합니다.

`2002`	6to4 접두어를 지정합니다.
IPv4-address	의사 인터페이스의 IPv4 주소를 16진수 표기법으로 지정합니다.
subnet-ID	0이 아닌 서브넷 ID를 16진수 표기법으로 지정합니다.
interface-ID	1이 아닌 인터페이스 ID를 지정합니다.
`/64`	6to4 접두어 길이가 64비트 길이임을 나타냅니다.
`up`	6to4 인터페이스를 "up"으로 구성합니다.

주 – 네트워크에서 두 개의 IPv6 터널은 동일한 소스 주소 및 동일한 대상 주소를 포함할 수 없습니다. 따라서 패킷이 삭제됩니다. 이 유형의 이벤트는 6to4 라우터가 `atun` 명령을 통해 터널링도 수행할 경우에 발생할 수 있습니다. `atun`에 대한 자세한 내용은 tun(7M) 매뉴얼 페이지를 참조하십시오.

3 **(선택 사항) 라우터에서 추가 6to4 의사 인터페이스를 만듭니다.**

해당하는 각 6to4 의사 인터페이스는 이미 구성된 전역으로 고유한 IPv4 주소를 포함해야 합니다.

4 **6to4 라우터를 재부트합니다.**

5 **인터페이스의 상태를 확인합니다.**

```
# ifconfig ip.6to4tun0 inet6
```

인터페이스가 올바르게 구성되었으면 다음과 비슷한 출력이 표시됩니다.

```
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6> mtu 1480 index 11
        inet tunnel src 111.222.33.44
        tunnel hop limit 60
        inet6 2002:6fde:212c:10:/64
```

6 **6to4 경로 지정을 알리도록 `/etc/inet/ndpd.conf` 파일을 편집합니다.**

자세한 내용은 `ndpd.conf(4)` 매뉴얼 페이지를 참조하십시오.

a. **첫번째 라인에서 알림을 수신할 서브넷을 지정합니다.**

다음 형식으로 `if` 항목을 만듭니다.

`if` *subnet-interface* `AdvSendAdvertisements 1`

예를 들어, 6to4 경로 지정을 `hme0` 인터페이스에 연결된 서브넷에 알리려면 *subnet-interface*를 `hme0`으로 바꿉니다.

```
if hme0 AdvSendAdvertisements 1
```

b. **6to4 접두어를 알림의 두번째 라인으로 추가합니다.**

다음 형식으로 `prefix` 항목을 만듭니다.

`prefix 2002:`*IPv4-address*`:`*subnet-ID*`::/64` *subnet-interface*

7 **라우터를 재부트합니다.**

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

8 **6to4 사이트에서 사용되는 이름 서비스에 노드의 새 6to4 파생 주소를 추가합니다.**

지침은 184 페이지 “IPv6용 이름 서비스 지원 구성”을 참조하십시오.

예 7–10 6to4 라우터 구성(짧은 형식)

다음은 `/etc/hostname6.ip.6to4tun0`의 짧은 형식 예입니다.

```
# cat /etc/hostname6.ip.6to4tun0
tsrc 111.222.33.44 up
```

예 7–11 6to4 라우터 구성(긴 형식)

다음은 /etc/hostname6.ip.6to4tun0의 긴 형식 예입니다.

```
# cat /etc/hostname6.ip.6to4tun0
tsrc 111.222.33.44 2002:6fde:212c:20:1/64 up
```

예 7–12 6to4 의사 인터페이스를 표시하는 ifconfig 출력

다음 샘플에서는 6to4 의사 인터페이스에 대한 ifconfig 명령의 출력을 보여줍니다.

```
# ifconfig ip.6to4tun0 inet6
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6> mtu 1480 index 11
        inet tunnel src 192.168.87.188
        tunnel hop limit 60
        inet6 2002:c0a8:57bc::1/64
```

예 7–13 /etc/inet/ndpd.conf의 6to4 알림

다음 샘플 /etc/inet/ndpd.conf 파일은 두 서브넷에서 6to4 경로 지정을 알립니다.

```
if qfe0 AdvSendAdvertisements 1
prefix  2002:c0a8:57bc:10::/64 qfe0

if qfe1 AdvSendAdvertisements 1
prefix  2002:c0a8:57bc:2::/64 qfe1
```

자세한 정보 6to4 사이트에서 다중 라우터 구성

다중 라우터 사이트의 경우 6to4 라우터 이외의 라우터는 6to4를 지원하기 위한 추가 구성이 필요할 수 있습니다. 사이트에서 RIP가 사용되는 경우에는 각 비6to4 라우터에서 6to4 라우터에 대한 정적 경로를 구성해야 합니다. 시판되는 경로 지정 프로토콜을 사용할 경우에는 6to4 라우터에 대한 정적 경로를 만들 필요가 없습니다.

▼ 6to4 릴레이 라우터에 대한 6to4 터널을 구성하는 방법

주의 – 주요 보안 문제로 인해 6to4 릴레이 라우터 지원은 기본적으로 Oracle Solaris에서 사용 안함으로 설정되어 있습니다. 6to4 릴레이 라우터로 터널링 시 발생하는 보안 문제를 참조하십시오.

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

- 179 페이지 "6to4 터널을 구성하는 방법"에 설명된 대로 사이트에서 6to4 라우터 구성
- 6to4 릴레이 라우터에 대한 터널링과 관련된 보안 문제 검토

1 6to4 라우터에 기본 관리자 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 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 `192.88.99.1`을 사용합니다. 사용자의 사이트와 물리적으로 가장 가까운 애니캐스트 릴레이 라우터가 6to4 터널의 끝점이 됩니다. 이 릴레이 라우터는 6to4 사이트와 고유 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.

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

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

4 (선택 사항) 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. **(선택 사항) 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 7–14 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
```

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 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**를 참조하십시오.

▼ DNS에 IPv6 주소를 추가하는 방법

1. **기본 또는 보조 DNS 서버에 기본 관리자 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

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

hostname IN AAAA *host-address*

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

hostaddress `IN PTR` *hostname*

DNS 관리에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**를 참조하십시오.

예 7–15 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.
```

NIS에 IPv6 주소 추가

Solaris 10 11/06 및 이전 릴리스에서는 NIS에 대해 `ipnodes.byname` 및 `ipnodes.byaddr`의 두 맵이 추가되었습니다. 이러한 맵에는 IPv4 및 IPv6 호스트 이름 및 주소 연결이 모두 포함되었습니다. IPv6을 인식하는 도구에는 `ipnodes` NIS 맵이 사용되었습니다. `hosts.byname` 및 `hosts.byaddr` 맵에는 IPv4 호스트 이름 및 주소 연결만 포함되었습니다. 이러한 맵은 기존 응용 프로그램을 지원할 수 있도록 변경되지 않았습니다. `ipnodes` 맵 관리는 `hosts.byname` 맵 `hosts.byaddr` 맵의 관리와 비슷합니다. Solaris 10 11/06의 경우 IPv4 주소를 포함하는 *hosts* 맵을 업데이트할 때 *ipnode* 맵도 동일한 정보로 업데이트됩니다.

주 – Oracle Solaris 10의 이후 릴리스에서는 `ipnodes` 맵이 사용되지 않습니다. `ipnodes` 맵의 IPv6 기능은 이제 `hosts` 맵에서 유지 관리됩니다.

NIS 맵 관리를 위한 지침은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**의 5 장, “Setting Up and Configuring NIS Service”을 참조하십시오.

▼ 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 7–16 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 7–17 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 ipnodes.***byname*

  지정된 *hostname*에 대한 정보가 표시됩니다.

주 – Solaris 10 11/06 이후의 Oracle Solaris 릴리스에는 더 이상 ipnodes 맵이 포함되지 않습니다. ipnodes의 IPv6 기능은 이제 hosts 맵에서 유지 관리됩니다.

예 7–18 ypmatch 명령으로 출력된 IPv6 주소

Solaris 10 11/06 및 이전 릴리스의 경우 다음 샘플에서 ipnodes.byname 데이터베이스에 대한 ypmatch 작업의 결과를 보여줍니다.

```
% ypmatch farhost hosts ipnodes.byname
2001:0db8:3c4d:15:a00:20ff:fe12:5286       farhost
```

▼ 이름 서비스와 독립적인 IPv6 정보를 표시하는 방법

이 절차는 Solaris 10 11/06 및 이전 릴리스에서 대해서만 사용할 수 있습니다. 이후 릴리스의 경우 hosts 데이터베이스에서 동일한 작업을 수행할 수 있습니다.

- **사용자 계정으로 다음 명령을 입력합니다.**

  % **getent ipnodes** *hostname*

  지정된 *host-name*에 대한 정보가 표시됩니다.

예 7–19 ipnodes 데이터베이스의 IPv6 정보 표시

다음 샘플에서는 getent 명령의 출력을 보여줍니다.

```
% getent ipnodes vallejo

2001:0db8:8512:2:56:a00:fe87:9aba    myhost myhost
fe80::56:a00:fe87:9aba     myhost myhost
```

◆ ◆ ◆ 8 장

8 TCP/IP 네트워크 관리(작업)

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

주 – 네트워크 인터페이스를 모니터하려면 190 페이지 "`ifconfig` 명령으로 인터페이스 구성 모니터링"를 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 4 장, "IPv6 네트워크 계획(작업)"을 참조하십시오.
- IPv6을 구성하고 듀얼 스택 네트워크 환경을 만들려면 7 장, "IPv6 네트워크 구성(작업)"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
인터페이스에 대한 구성 정보를 표시합니다.	시스템에서 각 인터페이스의 현재 구성을 확인합니다.	191 페이지 "특정 인터페이스에 대한 정보를 얻는 방법"
인터페이스 주소 지정을 표시합니다.	로컬 시스템의 모든 인터페이스에 대한 주소 지정을 확인합니다.	192 페이지 "인터페이스 주소 지정을 표시하는 방법"
프로토콜별 통계를 표시합니다.	특정 시스템에서 네트워크 프로토콜의 성능을 모니터링합니다.	194 페이지 "프로토콜별 통계를 표시하는 방법"
네트워크 상태를 표시합니다.	소켓 및 경로 지정 테이블 항목을 모두 표시하여 시스템을 모니터링합니다. 출력에는 IPv4에 대한 주소 그룹과 IPv6에 대한 inet6 주소 그룹이 포함됩니다.	198 페이지 "소켓 상태를 표시하는 방법"
네트워크 인터페이스의 상태를 표시합니다.	네트워크 인터페이스의 성능을 모니터링합니다. 이는 전송 문제를 해결하는 데 유용합니다.	197 페이지 "네트워크 인터페이스 상태를 표시하는 방법"
패킷 전송 상태를 표시합니다.	회선을 통해 전송되는 패킷의 상태를 모니터링합니다.	199 페이지 "특정 주소 유형의 패킷에 대한 전송 상태를 표시하는 방법"
IPv6 관련 명령의 화면 출력을 제어합니다.	`ping`, `netstat`, `ifconfig` 및 `traceroute` 명령의 출력을 제어합니다. `inet_type`이라는 파일을 만들고, 이 파일에서 `DEFAULT_IP` 변수를 설정합니다.	202 페이지 "IP 관련 명령의 화면 출력을 제어하는 방법"
네트워크 트래픽을 모니터링합니다.	`snoop` 명령을 사용하여 모든 IP 패킷을 표시합니다.	209 페이지 "IPv6 네트워크 트래픽을 모니터링하는 방법"
네트워크 라우터에 알려진 모든 경로를 추적합니다.	`traceroute` 명령을 사용하여 모든 경로를 표시합니다.	206 페이지 "모든 경로를 추적하는 방법"

ifconfig 명령으로 인터페이스 구성 모니터링

`ifconfig` 명령을 사용하여 인터페이스에 수동으로 IP 주소를 지정하고 인터페이스 매개변수를 수동으로 구성할 수 있습니다. 또한 Oracle Solaris 시작 스크립트는 `ifconfig`를 실행하여 6to4 터널 끝점과 같은 의사 인터페이스를 구성합니다.

본 설명서에는 다양한 `ifconfig` 명령의 여러 옵션을 사용하는 많은 작업이 포함되어 있습니다. 이 명령의 옵션과 변수 등에 대한 자세한 설명은 `ifconfig(1M)` 매뉴얼 페이지를 참조하십시오. `ifconfig`의 기본 구문은 다음과 같습니다.

`ifconfig` *interface [protocol-family]*

▼ 특정 인터페이스에 대한 정보를 얻는 방법

ifconfig 명령을 사용하여 특정 시스템의 인터페이스에 대한 기본 정보를 확인합니다. 예를 들어, 간단한 ifconfig 질의의 경우 다음을 확인할 수 있습니다.

- 시스템에 있는 모든 인터페이스의 장치 이름
- 인터페이스에 지정된 모든 IPv4 및 모든 IPv6 주소(해당하는 경우)
- 이러한 인터페이스가 현재 구성되어 있는지 여부

다음 절차에서는 ifconfig 명령을 사용하여 시스템 인터페이스에 대한 기본 구성 정보를 얻는 방법을 보여 줍니다.

1 로컬 호스트에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 특정 인터페이스에 대한 정보를 얻습니다.

```
# ifconfig interface
```

ifconfig 명령의 출력 형식은 다음과 같습니다.

- 상태 라인

  ifconfig 명령 출력의 첫번째 라인에는 인터페이스 이름과 현재 인터페이스와 연결된 상태 플래그가 포함됩니다. 또한 상태 라인에는 특정 인터페이스용으로 구성된 MTU(최대 전송 단위)와 색인 번호가 포함됩니다. 상태 라인을 사용하여 인터페이스의 현재 상태를 확인할 수 있습니다.

- IP 주소 정보 라인

  ifconfig 출력의 두번째 라인에는 인터페이스용으로 구성된 IPv4 주소 또는 IPv6 주소가 포함됩니다. IPv4 주소의 경우 구성된 넷마스크 및 브로드캐스트 주소도 표시됩니다.

- MAC 주소 라인

  수퍼 유저 또는 비슷한 역할로 ifconfig 명령을 실행하면 ifconfig 출력에 세번째 라인이 포함됩니다. IPv4 주소의 경우 세번째 라인에는 인터페이스에 지정된 MAC 주소(이더넷 계층 주소)가 표시됩니다. IPv6 주소의 경우 출력의 세번째 라인에는 MAC 주소를 통해 IPv6 in.ndpd 데몬에서 생성되는 링크 로컬 주소가 표시됩니다.

예 8–1 ifconfig 명령의 기본 인터페이스 정보

다음 예에서는 ifconfig 명령을 사용하여 특정 호스트의 eri 인터페이스에 대한 정보를 얻는 방법을 보여 줍니다.

```
# ifconfig eri
eri0: flags=863<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 1
```

```
inet 10.0.0.112 netmask ffffff80 broadcast 10.8.48.127
ether 8:0:20:b9:4c:54
```

다음 표에서는 ifconfig 질의의 변수 정보, 화면에 변수를 표시하는 방법 및 제공되는 정보의 종류에 대한 설명을 보여 줍니다. 위 출력 결과가 예로 사용됩니다.

변수	화면 출력	설명
인터페이스 이름	eri0	ifconfig 명령에서 상태가 요청된 인터페이스의 장치 이름을 나타냅니다.
인터페이스 상태	flags=863<UP	현재 인터페이스와 연결된 플래그를 포함하여 인터페이스의 상태를 표시합니다. 인터페이스가 현재 초기화되었는지(UP) 또는 초기화되지 않았는지(DOWN) 여부를 확인할 수 있습니다.
브로드캐스트 상태	BROADCAST	인터페이스에서 IPv4 브로드캐스트를 지원한다는 것을 나타냅니다.
전송 상태	RUNNING	시스템에서 인터페이스를 통해 패킷을 전송한다는 것을 나타냅니다.
멀티캐스트 상태	MULTICAST, IPv4	인터페이스에서 멀티캐스트 전송을 지원한다는 것을 표시합니다. 예제 인터페이스에서는 IPv4 멀티캐스트 전송을 지원합니다.
최대 전송 단위	mtu 1500	이 인터페이스의 최대 전송 크기가 1500옥테트임을 표시합니다.
IP 주소	inet 10.0.0.112	인터페이스에 지정된 IPv4 또는 IPv6 주소를 표시합니다. 예제 인터페이스 eri0의 IPv4 주소는 10.0.0.112입니다.
넷마스크	netmask ffffff80	특정 인터페이스의 IPv4 넷마스크를 표시합니다. IPv6 주소는 넷마스크를 사용하지 않습니다.
MAC 주소	ether 8:0:20:b9:4c:54	인터페이스의 이더넷 계층 주소를 표시합니다.

▼ 인터페이스 주소 지정을 표시하는 방법

라우터 및 멀티홈 호스트에는 둘 이상의 인터페이스가 있으며, 각 인터페이스에 둘 이상의 IP 주소가 지정되는 경우도 있습니다. ifconfig 명령을 사용하여 시스템의 인터페이스에 지정되는 주소를 모두 표시할 수 있습니다. 또한 ifconfig 명령을 사용하여 IPv4 또는 IPv6 주소 지정만 표시할 수도 있습니다. 인터페이스의 MAC 주소를 추가적으로 표시하려면 먼저 수퍼 유저 또는 적절한 역할로 로그인해야 합니다.

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

1 로컬 시스템에서는 네트워크 관리 역할 또는 수퍼 유저로 로그인합니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 모든 인터페이스에 대한 정보를 얻습니다.

`ifconfig -a` 명령의 변형을 사용하여 다음을 수행할 수 있습니다.

- 시스템에 있는 모든 인터페이스의 모든 주소를 봅니다.

  ```
  # ifconfig -a
  ```

- 시스템의 인터페이스에 지정된 모든 IPv4 주소를 봅니다.

  ```
  # ifconfig -a4
  ```

- 로컬 시스템에서 IPv6이 사용으로 설정된 경우 시스템의 인터페이스에 지정된 모든 IPv6 주소를 표시합니다.

  ```
  ifconfig -a6
  ```

예 8–2 모든 인터페이스에 대한 주소 지정 정보 표시

이 예에서는 주 네트워크 인터페이스인 `qfe0`만 있는 호스트의 항목을 표시합니다. 그러나 `ifconfig` 출력에는 현재 `qfe0`에 3가지 형태의 주소인 loopback(`lo0`), IPv4(`inet`) 및 IPv6(`inet6`)이 지정된 것으로 표시됩니다. IPv6 출력 섹션의 인터페이스 `qfe0` 라인에는 링크 로컬 IPv6 주소가 표시됩니다. `qfe0`의 두번째 주소는 `qfe0:1` 라인에 표시됩니다.

```
% ifconfig -a
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.112 netmask ffffff80 broadcast 10.0.0.127
        ether 8:0:20:b9:4c:54
lo0: flags=2000849 <UP,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 8:0:20:b9:4c:54
        inet6 fe80::a00:20ff:feb9:4c54/10
qfe0:1: flags=2080841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
        inet6 2001:db8:3c4d:48:a00:20ff:feb9:4c54/64
```

예 8–3 모든 IPv4 인터페이스에 대한 주소 지정 정보 표시

이 예에서는 멀티홈 호스트에 대해 구성된 IPv4 주소를 표시합니다. 이 형태의 `ifconfig` 명령은 수퍼 유저로 로그인하지 않아도 실행할 수 있습니다.

```
% ifconfig -a4
lo0: flags=1000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
qfe0: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.112 netmask ffffff80 broadcast 10.0.0.127
        ether 8:0:20:b9:4c:54
qfe1: flags=1004843 <UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 10.0.0.118 netmask ffffff80 broadcast 10.0.0.127
        ether 8:0:20:6f:5e:17
```

예 8–4 모든 IPv6 인터페이스에 대한 주소 지정 정보 표시

이 예에서는 특정 호스트에 대해 구성된 IPv6 주소만 표시합니다. 이 형태의 `ifconfig` 명령은 수퍼 유저로 로그인하지 않아도 실행할 수 있습니다.

```
% ifconfig -a6
lo0: flags=2000849 <UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 <UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 8:0:20:b9:4c:54
        inet6 fe80::a00:20ff:feb9:4c54/10
qfe0:1: flags=2080841 <UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 index 2
        inet6 2001:db8:3c4d:48:a00:20ff:feb9:4c54/64
```

이 `ifconfig`의 출력은 호스트의 단일 인터페이스에 지정된 다음 3가지 유형의 IPv6 주소 형태를 보여 줍니다.

`lo0`
: IPv6 루프백 주소입니다.

`inet6 fe80::a00:20ff:feb9:4c54/10`
: 주 네트워크 인터페이스에 지정된 링크 로컬 주소입니다.

`inet6 2001:db8:3c4d:48:a00:20ff:feb9:4c54/64`
: 서브넷 접두어가 포함된 IPv6 주소입니다. 출력에서 `ADDRCONF`는 이 주소가 호스트에 의해 자동 구성되었음을 나타냅니다.

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. `netstat`의 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 8–5 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 8–6 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 8–7 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 8–8 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, ipnodes 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
hme0  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
hme0  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 지정 테이블 항목의 상태를 표시하려면 다음을 입력합니다.**

  사용자 계정을 사용하여 netstat의 이 옵션을 실행할 수 있습니다.

  ```
  % netstat -a
  ```

예 8–9 모든 소켓 및 경로 지정 테이블 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 inet6를 netstat -f에 대한 인수로 사용합니다.

예 8–10 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 8–11 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 지정 테이블을 표시합니다. 이 표는 호스트에 알려진 모든 경로의 상태를 보여줍니다. 사용자 계정에서 netstat의 이 옵션을 실행할 수 있습니다.

- **IP 경로 지정 테이블을 표시합니다.**

  ```
  $ netstat -r
  ```

예 8–12 netstat 명령에 의한 경로 지정 테이블 출력

다음 예는 netstat -r 명령의 출력을 보여줍니다.

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
host15               myhost               U         1  31059  hme0
10.0.0.14            myhost               U         1      0  hme0
default              distantrouter        UG        1      2  hme0
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 hme0:1
fe80::/10                fe80::1a2b:3c4d:5e6f:12a2    U       1     23 hme0
ff00::/8                 fe80::1a2b:3c4d:5e6f:12a2    U       1      0 hme0
default                  fe80::1a2b:3c4d:5e6f:12a2    UG      1      0 hme0
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 지정 테이블은 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.

매개변수	설명
Use	전송된 패킷 수를 표시합니다.
Interface	전송의 소스 끝점인 로컬 호스트의 특정 인터페이스를 나타냅니다.

ping 명령으로 원격 호스트 확인

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 8–13 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, ifconfig 및 netstat 명령으로 보고되는 네트워크 상태를 확인하십시오. 190 페이지 “ifconfig 명령으로 인터페이스 구성 모니터링” 및 194 페이지 “netstat 명령으로 네트워크 상태 모니터링”를 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 IPv4 및 IPv6 정보를 모두 표시하도록 netstat 및 ifconfig 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 – `ifconfig` 명령의 `-4` 및 `-6` 플래그는 `inet_type` 파일에 설정된 값을 대체합니다. `netstat` 명령의 `-f` 플래그는 또한 `inet_type` 파일에 설정된 값을 대체합니다.

예 8–14 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.

```
% ifconfig -a
lo0: flags=1000849 mtu 8232 index 1
        inet 10.10.0.1 netmask ff000000
qfe0: flags=1000843 mtu 1500 index 2
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.86.255
        ether 8:0:20:56:a8
lo0: flags=2000849 mtu 8252 index 1
        inet6 ::1/128
qfe0: flags=2000841 mtu 1500 index 2
        ether 8:0:20:56:a8
        inet6 fe80::a00:fe73:56a8/10
qfe0:1: flags=2080841 mtu 1500 index 2
        inet6 2001:db8:3c4d:5:a00:fe73:56a8/64
```

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

```
% ifconfig -a
lo0: flags=849 mtu 8232
        inet 10.10.0.1 netmask ff000000
qfe0: flags=843 mtu 1500
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.86.255
        ether 8:0:20:56:a8
```

▼ IPv4 경로 지정 데몬의 작업을 기록하는 방법

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

1 로컬 호스트에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 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 8–15 `in.routed` 데몬에 대한 네트워크 로그

다음 예는 203 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 hme0 #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  hme0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  hme0  <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. **로컬 IPv6 노드에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2. **`in.ndpd` 데몬의 추적을 시작합니다.**

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

3. **필요한 경우 Ctrl-C를 입력하여 추적을 종료합니다.**

예 8–16 `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 hme0
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 hme0
```

```
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 8–17 traceroute 명령으로 원격 호스트에 대한 경로 표시

traceroute 명령의 다음 출력은 로컬 호스트 nearhost에서 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 8–18 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 수퍼 유저여야 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 **로컬 호스트에서 네트워크 관리 역할을 맡거나 수퍼 유저로 전환합니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **시스템에 연결된 인터페이스에 대한 정보를 출력합니다.**

```
# ifconfig -a
```

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

3 **예 8–19에 표시된 것과 같이, snoop를 인수 없이 입력하여 패킷 캡처를 시작합니다.**

4 **Ctrl-C를 사용하여 프로세스를 정지합니다.**

예 8–19 snoop 명령의 출력

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

```
% snoop
Using device /dev/hme (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)
farhost.remote.com -> myhost       RLOGIN C port=993
myhost ->   nisserve2          NIS C MATCH 10.0.0.64 in ipnodes.byaddr
nisserve2 ->    myhost             NIS R MATCH No such key
blue-112 -> slave-253-2        NIS C MATCH 10.0.0.112 in ipnodes.byaddr
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 로컬 호스트에서 네트워크 관리 역할을 맡거나 수퍼 유저로 전환합니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 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.

3 snoop 출력 캡처 파일을 검사합니다.

```
# snoop -i filename
```

예 8–20 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 **snoop 시스템에서 네트워크 관리 역할을 맡거나 수퍼 유저로 전환합니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

3 **옵션과 함께 snoop를 입력한 다음 출력을 파일에 저장합니다.**

4 **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.

1 **로컬 노드에서 네트워크 관리 역할을 맡거나 수퍼 유저로 전환합니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **IPv6 패킷을 캡처합니다.**

```
# snoop ip6
```

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

예 8–21 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
```

기본 주소 선택 관리

Oracle Solaris에서는 한 인터페이스에서 여러 개의 IP 주소를 사용할 수 있습니다. 예를 들어 IPMP(IP Network Multipathing)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 250 페이지 “ipaddrsel 명령”을 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 기본 관리자 역할 또는 수퍼 유저로 로그인합니다

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 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
```

3 기본 주소 정책 테이블의 백업 복사본을 만듭니다.

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

4 텍스트 편집기를 사용하여 /etc/inet/ipaddrsel.conf에 사용자 정의 내용을 추가합니다.

/etc/inet/ipaddrsel의 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.

5 수정된 정책 테이블을 커널로 로드합니다.

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

6 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. **기본 관리자 역할 또는 수퍼 유저로 로그인합니다**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2. **`/etc/inet/ipaddrsel`의 내용을 *filename*으로 복사합니다. 여기서 *filename*은 사용자가 선택한 파일의 이름을 나타냅니다.**

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

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

4. **수정된 정책 테이블을 커널로 로드합니다.**

   ```
   # ipaddrsel -f filename
   ```

   시스템을 재부트할 때까지 커널에서 새 정책 테이블을 사용합니다.

◆ ◆ ◆ 9 장

9 네트워크 문제 해결(작업)

이 장에서는 네트워크에서 발생할 수 있는 일반적인 문제에 대한 해결 방법에 대해 설명합니다. 다음 항목을 다룹니다.

네트워크 문제 해결의 새로운 내용

Solaris 10 7/07에서 `/etc/inet/ipnodes` 파일은 더 이상 사용되지 않습니다. 개별 절차에 설명된 대로 `/etc/inet/ipnodes`는 이전 Solaris 10 릴리스에서만 사용하십시오.

일반 네트워크 문제 해결 팁

네트워크 문제를 나타내는 첫번째 신호 중 하나는 하나 이상의 호스트에서 통신이 끊기는 것입니다. 호스트가 처음에 네트워크에 추가될 때부터 호스트가 응답이 없다면 구성 파일 중 하나에 문제가 있는 것일 수 있습니다. 잘못된 네트워크 인터페이스 카드도 문제일 수 있습니다. 한 호스트에서 갑자기 문제가 발생한다면 네트워크 인터페이스가 원인일 수 있습니다. 네트워크의 호스트가 서로 통신할 수는 있지만 다른 네트워크와 통신할 수 없는 경우 라우터 문제일 수 있습니다. 또는 다른 네트워크에 문제가 있는 것일 수 있습니다.

`ifconfig` 명령을 사용하면 네트워크 인터페이스에 대한 정보를 얻을 수 있습니다. `netstat` 명령을 사용하면 경로 지정 테이블 및 프로토콜 통계를 표시할 수 있습니다. 타사 네트워크 진단 프로그램은 여러 가지 문제 해결 도구를 제공합니다. 자세한 내용은 타사 설명서를 참조하십시오.

네트워크 성능을 저하시키는 문제의 원인은 명확하지 않습니다. 예를 들어 `ping`과 같은 도구를 사용하여 호스트에 의한 패킷 손실과 같은 문제를 수량화할 수 있습니다.

기본 진단 검사 실행

네트워크 문제가 발생하면 일련의 소프트웨어 검사를 실행하여 기본적인 소프트웨어 관련 문제를 진단하고 수정할 수 있습니다.

▼ 기본 네트워크 소프트웨어 검사를 수행하는 방법

1. **로컬 시스템에서는 네트워크 관리 역할 또는 수퍼 유저로 로그인합니다.**

   역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2. **netstat 명령을 사용하여 네트워크 정보를 표시합니다.**

   netstat 명령에 대한 구문 및 정보는 194 페이지 "netstat 명령으로 네트워크 상태 모니터링" 및 netstat(1M) 매뉴얼 페이지를 참조하십시오.

3. **IPv6을 사용 중인 경우 hosts 데이터베이스(및 Solaris 10 11/06 및 이전 릴리스의 경우 ipnodes 데이터베이스)를 검사하여 항목이 올바르고 최신 상태인지 확인합니다.**

   /etc/inet/hosts 데이터베이스에 대한 자세한 내용은 219 페이지 "hosts 데이터베이스" 및 hosts(4) 매뉴얼 페이지를 참조하십시오. /etc/inet/ipnodes 데이터베이스에 대한 자세한 내용은 223 페이지 "ipnodes 데이터베이스" 및 ipnodes(4) 매뉴얼 페이지를 참조하십시오.

4. **RARP(Reverse Address Resolution Protocol)를 실행 중인 경우 ethers 데이터베이스에서 이더넷 주소를 검사하여 항목이 올바르며 최신 상태인지 확인합니다.**

5. **telnet 명령을 사용하여 로컬 호스트에 연결을 시도합니다.**

   telnet에 대한 구문 및 정보는 telnet(1) 매뉴얼 페이지를 참조하십시오.

6. **네트워크 데몬 inetd가 실행 중인지 확인합니다.**

   ```
   # ps -ef | grep inetd
   ```

   다음 출력은 inetd 데몬이 실행 중인지 확인합니다.

   ```
   root 57 1 0 Apr 04 ? 3:19 /usr/sbin/inetd -s
   ```

7. **네트워크에서 IPv6이 사용 가능한 경우 IPv6 데몬 in.ndpd가 실행 중인지 확인합니다.**

   ```
   # ps -ef | grep in.ndpd
   ```

   다음 출력은 in.ndpd 데몬이 실행 중인지 확인합니다.

   ```
   root 123  1 0  Oct 27 ?  0:03 /usr/lib/inet/in.ndpd
   ```

IPv6 배치 시 발생하는 일반적인 문제

이 절에서는 사이트에서 IPv6을 계획하고 배치할 때 발생할 수 있는 문제에 대해 설명합니다. 실제 계획 작업은 4 장, "IPv6 네트워크 계획(작업)"을 참조하십시오.

IPv4 라우터를 IPv6으로 업그레이드할 수 없음

기존 장비를 업그레이드할 수 없는 경우 IPv6 지원 장비를 구매해야 할 수 있습니다. IPv6 지원을 위해 수행해야 하는 장비별 절차는 제조업체의 설명서를 확인하십시오.

특정 IPv4 라우터는 IPv6 지원을 위해 업그레이드할 수 없습니다. 이 상황이 사용자의 토폴로지에 해당하는 경우 물리적으로 IPv6 라우터를 IPv4 라우터 옆에 연결하십시오. 그런 다음 IPv4 라우터를 통해 IPv6 라우터에서 터널링할 수 있습니다. 터널 구성을 위한 작업은 176 페이지 "IPv6 지원을 위한 터널 구성 작업(작업 맵)"을 참조하십시오.

IPv6으로 서비스 업그레이드 후 발생하는 문제

IPv6을 지원하도록 서비스를 준비할 때 다음과 같은 상황이 발생할 수 있습니다.

- 특정 응용 프로그램의 경우 IPv6으로 이식한 후에도 기본적으로 IPv6 지원이 설정되지 않습니다. IPv6을 설정하도록 이 응용 프로그램을 구성해야 할 수 있습니다.
- 서버에서 여러 서비스를 실행하고 이 중 일부는 IPv4 전용 서비스, 일부는 IPv4 및 IPv6 서비스인 경우 문제가 발생할 수 있습니다. 일부 클라이언트에서 두 유형의 서비스를 사용해야 하는데 이로 인해 서버 측에서 혼동을 일으킬 수 있습니다.

현재 ISP가 IPv6을 지원하지 않음

IPv6을 배치하려는데 현재 ISP가 IPv6 주소 지정을 제공하지 않을 경우 ISP를 변경하지 않는 다음 대안을 고려하십시오.

- 사이트에서 IPv6 통신용 다른 회선을 제공하는 ISP를 이용합니다. 이 솔루션은 비용이 많이 듭니다.
- **가상** *ISP*를 사용합니다. 가상 ISP는 사이트에 링크가 아닌 IPv6 연결을 제공합니다. 대신 사용자의 사이트에서 IPv4 ISP를 경유하여 가상 ISP에 연결되는 터널을 만듭니다.
- 사용자의 ISP를 경유하여 다른 IPv6 사이트에 연결되는 6to4 터널을 사용합니다. 주소의 경우, 6to4 라우터의 등록된 IPv4 주소를 IPv6 주소의 공용 토폴로지 부분으로 사용합니다.

6to4 릴레이 라우터로 터널링 시 발생하는 보안 문제

원래 6to4 라우터와 6to4 릴레이 라우터 간 터널은 비보안 상태입니다. 따라서 터널에는 다음과 같은 보안 문제가 내재되어 있습니다.

- 6to4 릴레이 라우터는 패킷 캡슐화 및 캡슐화 해제를 수행하지만 패킷 내에 포함된 데이터는 검사하지 않습니다.
- 6to4 릴레이 라우터에서 주로 발생하는 문제는 주소 스푸핑입니다. 수신 트래픽의 경우 6to4 라우터가 릴레이 라우터의 IPv4 주소를 소스의 IPv6 주소에 대응시킬 수 없습니다. 따라서 IPv6 호스트의 주소가 쉽게 스푸핑될 수 있습니다. 6to4 릴레이 라우터의 주소도 스푸핑될 수 있습니다.
- 기본적으로 6to4 라우터와 6to4 릴레이 라우터 간에는 신뢰 방식이 존재하지 않습니다. 따라서 6to4 라우터는 6to4 릴레이 라우터를 신뢰할 수 있는지 또는 적합한 6to4 릴레이 라우터인지 식별할 수 없습니다. 6to4 사이트와 IPv6 대상 간에는 신뢰 관계가 존재해야 합니다. 그렇지 않으면 두 사이트가 공격 받을 가능성이 있습니다.

6to4 릴레이 라우터에 내재되어 있는 이러한 문제 및 기타 문제는 **Security Considerations for 6to4**의 Internet Draft에 설명되어 있습니다. 일반적으로 다음과 같은 경우에만 6to4 릴레이 라우터에 대한 지원을 사용으로 설정해야 합니다.

- 6to4 사이트가 신뢰할 수 있는 개인 IPv6 네트워크와 통신하려는 경우. 예를 들어 격리된 6to4 사이트와 고유 IPv6 사이트로 구성된 캠퍼스 네트워크에서 6to4 릴레이 라우터 지원을 사용으로 설정할 수 있습니다.
- 6to4 사이트가 특정 고유 IPv6 호스트와 통신할 수 밖에 없는 비즈니스 이유가 있는 경우
- **Security Considerations for 6to4**, Internet Draft에서 권장하는 검사 및 신뢰 모델을 구현한 경우

◆ ◆ ◆ 10장

10 TCP/IP 및 IPv4에 대한 자세한 정보(참조)

이 장에서는 파일 항목의 유형, 용도 및 형식을 포함하여 네트워크 구성 파일에 대한 TCP/IP 네트워크 참조 정보를 제공합니다. 기존 네트워크 데이터베이스에 대해서도 자세히 설명합니다. 또한 정의된 네트워크 분류 및 서브넷 번호를 기반으로 IPv4 주소의 구조가 파생되는 방법에 대해서도 설명합니다.

이 장은 다음 정보를 포함합니다.

TCP/IP 및 IPv4의 새로운 기능에 대한 자세한 정보

Solaris 10 7/07에서는 /etc/inet/ipnodes 파일은 더 이상 사용되지 않습니다. 개별 절차에 설명된 대로 /etc/inet/ipnodes는 이전 Solaris 10 릴리스에서만 사용하십시오.

TCP/IP 구성 파일

네트워크의 각 시스템은 다음 TCP/IP 구성 파일 및 네트워크 데이터베이스에서 해당 TCP/IP 구성 정보를 가져옵니다.

- /etc/hostname.*interface* 파일
- /etc/nodename 파일
- /etc/defaultdomain 파일
- /etc/defaultrouter 파일(선택 사항)
- hosts 데이터베이스
- Solaris 10 11/06 및 이전 릴리스의 ipnodes 데이터베이스
- netmasks 데이터베이스(선택 사항)

Oracle Solaris 설치 프로그램은 이러한 파일을 설치 프로세스 중에 만듭니다. 또한 이 절에 설명된 대로 수동으로 파일을 편집할 수 있습니다. hosts 및 netmasks 데이터베이스는 Oracle Solaris 네트워크의 사용 가능한 이름 서비스에서 읽은 네트워크 데이터베이스 중 두 데이터베이스입니다. 227 페이지 "네트워크 데이터베이스 및 nsswitch.conf 파일"에서는 네트워크 데이터베이스의 개념을 자세히 설명합니다. Solaris 10 11/06 및 이전 릴리스의 ipnodes 파일에 대한 자세한 내용은 223 페이지 "ipnodes 데이터베이스"를 참조하십시오.

/etc/hostname.*interface* 파일

이 파일은 로컬 호스트의 물리적 네트워크 인터페이스를 정의합니다. 로컬 시스템에 /etc/hostname.*interface* 파일이 하나 이상 있어야 합니다. Oracle Solaris 설치 프로그램은 설치 프로세스 중에 찾은 첫번째 인터페이스에 대한 /etc/hostname.*interface* 파일을 만듭니다. 이 인터페이스는 대개 가장 낮은 장치 번호(예: eri0)를 사용하며 이를 **주 네트워크 인터페이스**라고 합니다. 설치 프로그램에서 추가 인터페이스를 찾으면 설치 프로세스 중에 해당 인터페이스를 선택적으로 구성할 수 있습니다.

주 – 동일한 인터페이스에 대한 대체 호스트 이름 파일을 만드는 경우, 대체 파일도 이름 지정 형식 hostname.[0–9]* (예: hostname.qfe0.a123)를 따라야 합니다. hostname.qfe0.bak 또는 hostname.qfe0.old 등의 이름은 잘못된 이름으로 시스템 부트 중 스크립트에서 무시됩니다.

또한 제공된 인터페이스의 해당 호스트 이름은 한 개뿐이어야 합니다. 인터페이스에 대한 대체 호스트 이름 파일을 유효한 파일 이름(예: /etc/hostname.qfe 및 /etc/hostname.qfe.a123)으로 만들 경우 부트 스크립트에서 두 호스트 이름 파일의 내용을 참조하여 구성을 시도하므로 오류가 발생합니다. 이러한 오류를 방지하려면 제공된 구성에 사용하지 않을 호스트 이름 파일에는 잘못된 파일 이름을 제공하십시오.

설치 후 시스템에 새 네트워크 인터페이스를 추가하려면 138 페이지 "시스템 설치 후 물리적 인터페이스 구성 방법"에서 설명하는 대로 해당 인터페이스에 대한 /etc/hostname.*interface* 파일을 만들어야 합니다. 또한 Oracle Solaris 소프트웨어에서 새 네트워크 인터페이스를 인식하고 사용하려면 인터페이스의 장치 드라이버를 해당 디렉토리로 로드해야 합니다. 적합한 *interface* 이름 및 장치 드라이버 지침은 새 네트워크 인터페이스와 함께 제공되는 설명서를 참조하십시오.

기본 /etc/hostname.*interface* 파일에는 네트워크 인터페이스와 연결된 호스트 이름 또는 IPv4 주소만 항목으로 포함됩니다. IPv4 주소는 점으로 구분된 일반적인 십진수 형식 또는 CIDR 표기법으로 표현될 수 있습니다. 호스트 이름을 /etc/hostname.*interface* 파일의 항목으로 사용하면 해당 호스트 이름은 /etc/inet/hosts 파일에도 존재해야 합니다.

예를 들어, smc0이 tenere 시스템의 주 네트워크 인터페이스라고 가정합니다. /etc/hostname.smc0 파일은 점으로 구분된 십진수 표기법 또는 CIDR 표기법의 IPv4 주소 또는 호스트 이름 tenere를 해당 항목으로 포함할 수 있습니다.

주 – IPv6은 네트워크 인터페이스를 정의하는 데 `/etc/hostname6.`*interface* 파일을 사용합니다. 자세한 내용은 249 페이지 “IPv6 인터페이스 구성 파일”을 참조하십시오.

/etc/nodename 파일

이 파일은 로컬 시스템의 호스트 이름만 항목으로 포함해야 합니다. 예를 들어, 시스템 `timbuktu`의 파일 `/etc/nodename`에는 항목 `timbuktu`가 포함됩니다.

/etc/defaultdomain 파일

이 파일은 로컬 호스트의 네트워크가 속한 관리 도메인의 정규화된 도메인 이름만 항목으로 포함해야 합니다. 이 이름을 Oracle Solaris 설치 프로그램에 제공하거나 나중에 파일을 편집할 수 있습니다. 네트워크 도메인에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

/etc/defaultrouter 파일

이 파일은 네트워크에 직접 연결되는 각 라우터에 대한 항목을 포함할 수 있습니다. 항목은 네트워크 간에 라우터로 사용되는 네트워크 인터페이스의 이름이어야 합니다. `/etc/defaultrouter` 파일이 있으면 시스템이 정적 경로 지정을 지원하도록 구성되었음을 나타냅니다.

hosts 데이터베이스

`hosts` 데이터베이스는 네트워크에 있는 시스템의 IPv4 주소 및 호스트 이름을 포함합니다. NIS 또는 DNS 이름 서비스 또는 LDAP 디렉토리 서비스를 사용하면 `hosts` 데이터베이스는 호스트 정보에 지정된 데이터베이스에서 유지 관리됩니다. 예를 들어, NIS를 실행하는 네트워크에서 `hosts` 데이터베이스는 `hostsbyname` 파일에서 유지 관리됩니다.

로컬 파일을 이름 서비스로 사용하면 `hosts` 데이터베이스는 `/etc/inet/hosts` 파일에서 유지 관리됩니다. 이 파일에는 주 네트워크 인터페이스의 호스트 이름과 IPv4 주소, 시스템에 연결된 다른 네트워크 인터페이스 및 시스템에서 확인해야 하는 다른 네트워크 주소가 포함됩니다.

주 – BSD 기반 운영 체제와의 호환성을 위해 /etc/hosts 파일은 /etc/inet/hosts에 대한 심볼릭 링크입니다.

/etc/inet/hosts 파일 형식

/etc/inet/hosts 파일은 다음의 기본 구문을 사용합니다. 복잡한 구문 정보는 hosts(4) 매뉴얼 페이지를 참조하십시오.

IPv4-address hostname [nicknames] [#comment]

IPv4-address 로컬 호스트가 인식해야 하는 각 인터페이스에 대한 IPv4 주소를 포함합니다.

hostname 설정 시 시스템에 지정된 호스트 이름과 로컬 호스트가 인식해야 하는 추가 네트워크 인터페이스에 지정된 호스트 이름을 포함합니다.

[nickname] 호스트의 별명을 포함하는 선택적 필드입니다.

[#comment] 설명을 포함하는 선택적 필드입니다.

초기 /etc/inet/hosts 파일

시스템에서 Oracle Solaris 설치 프로그램을 실행하면 프로그램이 초기 /etc/inet/hosts 파일을 구성합니다. 이 파일에는 로컬 호스트에 필요한 최소 항목이 포함됩니다. 항목에는 루프백 주소, 호스트 IPv4 주소 및 호스트 이름이 포함됩니다.

예를 들어, Oracle Solaris 설치 프로그램은 그림 5–1과 같이 시스템 tenere에 대한 다음 /etc/inet/hosts 파일을 만들 수 있습니다.

예 10–1 시스템 tenere에 대한 /etc/inet/hosts 파일

```
127.0.0.1     localhost         loghost    #loopback address
192.168.200.3   tenere                        #host name
```

루프백 주소

예 10–1에서는 IPv4 주소 127.0.0.1이 **루프백 주소**입니다. 루프백 주소는 프로세스 간 통신을 허용하기 위해 로컬 시스템에서 사용되는 예약된 네트워크 인터페이스입니다. 이 주소를 사용으로 설정하면 호스트가 자신에게 패킷을 보낼 수 있습니다. 190 페이지 “ifconfig 명령으로 인터페이스 구성 모니터링”에 설명된 대로 ifconfig 명령은 구성 및 테스트에 루프백 주소를 사용합니다. TCP/IP 네트워크의 모든 시스템은 로컬 호스트의 IPv4 루프백에 대해 IP 주소 127.0.0.1을 사용해야 합니다.

호스트 이름

IPv4 주소 192.168.200.1 및 이름 tenere는 로컬 시스템의 주소 및 호스트 이름입니다. 이러한 주소 및 이름은 시스템의 주 네트워크 인터페이스에 지정됩니다.

다중 네트워크 인터페이스

일부 시스템은 라우터이거나 멀티홈 호스트이므로 두 개 이상의 네트워크 인터페이스가 포함됩니다. 시스템에 연결된 각 네트워크 인터페이스에는 자체 IP 주소 및 연결된 이름이 필요합니다. 설치 시 주 네트워크 인터페이스를 구성해야 합니다. 설치 시 특정 시스템에 다중 인터페이스가 있는 경우 Oracle Solaris 설치 프로그램에서는 이러한 추가 인터페이스에 대한 프롬프트를 표시합니다. 선택적으로 하나 이상의 추가 인터페이스를 이때 구성하거나 나중에 수동으로 구성할 수 있습니다.

Oracle Solaris 설치 이후 시스템의 `/etc/inet/hosts` 파일에 인터페이스 정보를 추가하여 라우터 또는 멀티홈 호스트에 대한 추가 인터페이스를 구성할 수 있습니다. 라우터 및 멀티홈 호스트 구성에 대한 자세한 내용은 112 페이지 “IPv4 라우터 구성” 및 120 페이지 “멀티홈 호스트 구성”을 참조하십시오.

예 10–2에서는 그림 5–1에 나와 있는 시스템 `timbuktu`에 대한 `/etc/inet/hosts` 파일을 보여 줍니다.

예 10–2 시스템 timbuktu에 대한 `/etc/inet/hosts` 파일

```
127.0.0.1        localhost     loghost
192.168.200.70   timbuktu      #This is the local host name
192.168.201.10   timbuktu-201  #Interface to network 192.9.201
```

이러한 두 인터페이스를 사용하여 `timbuktu`는 네트워크 `192.168.200` 및 `192.168.201`을 라우터로 연결합니다.

hosts 데이터베이스에 대한 이름 서비스의 영향

NIS 및 DNS 이름 서비스와 LDAP 디렉토리 서비스는 하나 이상의 서버에서 호스트 이름 및 주소를 유지 관리합니다. 이러한 서버는 서버 네트워크의 모든 호스트 및 라우터(해당하는 경우)에 대한 정보가 포함된 `hosts` 데이터베이스를 유지 관리합니다. 이러한 서비스에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

로컬 파일에서 이름 서비스를 제공하는 경우

로컬 파일을 이름 서비스로 사용하는 네트워크에서 로컬 파일 모드로 실행하는 시스템은 네트워크에 있는 다른 시스템의 IPv4 주소 및 호스트 이름에 대해 개별 `/etc/inet/hosts` 파일을 참조합니다. 따라서 이러한 시스템의 `/etc/inet/hosts` 파일은 다음 항목을 포함해야 합니다.

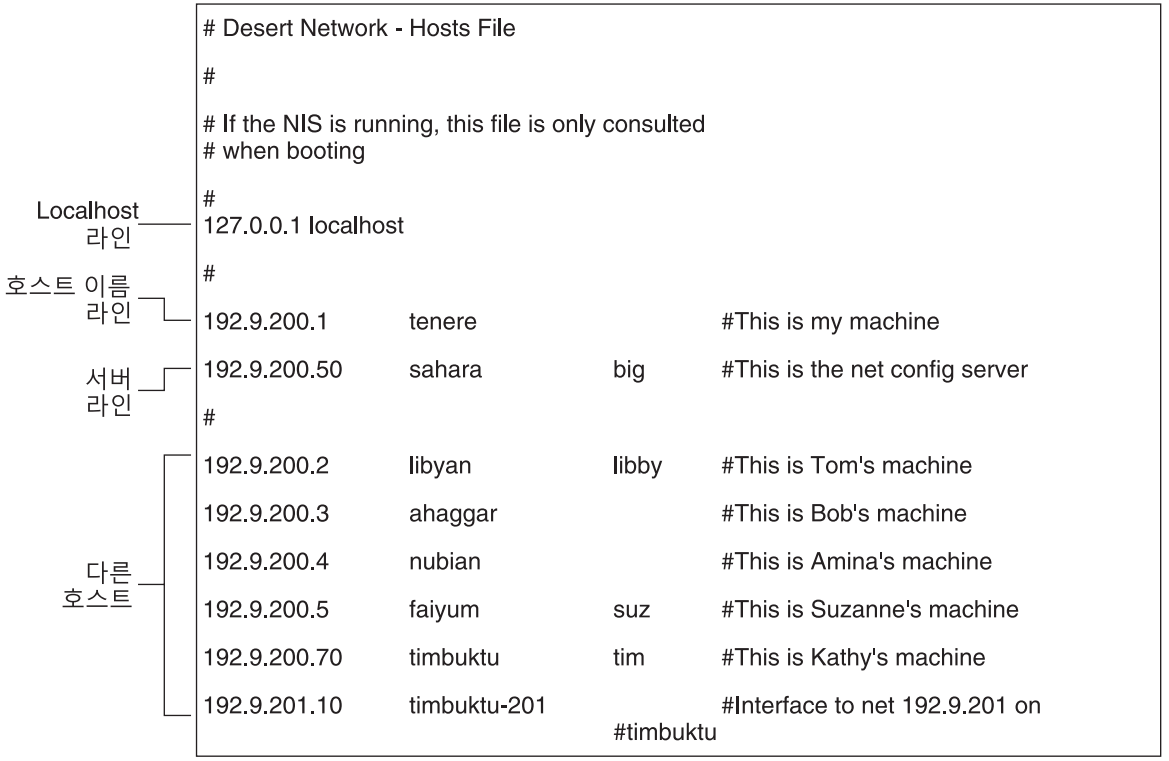
- 루프백 주소
- 로컬 시스템의 IPv4 주소 및 호스트 이름(주 네트워크 인터페이스)
- 이 시스템에 연결된 추가 네트워크 인터페이스의 IPv4 주소 및 호스트 이름(해당하는 경우)

- 로컬 네트워크에 있는 모든 호스트의 IPv4 주소 및 호스트 이름
- 이 시스템에서 알고 있어야 하는 라우터의 IPv4 주소 및 호스트 이름(해당하는 경우)
- 시스템이 해당 호스트 이름으로 참조해야 하는 시스템의 IPv4 주소

그림 10–1에서는 시스템 `tenere`에 대한 `/etc/inet/hosts` 파일을 보여 줍니다. 이 시스템은 로컬 파일 모드로 실행합니다. 파일에 `192.9.200` 네트워크의 모든 시스템에 대한 IPv4 주소 및 호스트 이름이 포함되어 있는지 확인합니다. 파일에는 IPv4 주소 및 인터페이스 이름 `timbuktu-201`도 포함됩니다. 이 인터페이스는 `192.9.200` 네트워크를 `192.9.201` 네트워크에 연결합니다.

네트워크 클라이언트로 구성된 시스템은 로컬 `/etc/inet/hosts` 파일을 해당 루프백 주소 및 IPv4 주소로 사용합니다.

그림 10–1 로컬 파일 모드로 실행 중인 시스템에 대한 `/etc/inet/hosts` 파일



ipnodes 데이터베이스

주 – Solaris 10 11/06 이후 릴리스에서는 더 이상 ipnodes 데이터베이스가 포함되지 않습니다. 해당 후속 릴리스에서는 ipnodes의 IPv6 기능이 hosts 데이터베이스로 마이그레이션됩니다.

/etc/inet/ipnodes 파일은 IPv4 주소와 IPv6 주소를 모두 저장합니다. 또한 IPv4 주소를 점으로 구분된 일반적인 십진수 또는 CIDR 표기법으로 저장할 수 있습니다. 이 파일은 호스트 이름을 해당 IPv4 및 IPv6 주소와 연결하는 로컬 데이터베이스로 사용됩니다. 호스트 이름 및 해당 주소를 /etc/inet/ipnodes와 같은 정적 파일에 저장하지 마십시오. 그러나 테스트를 위해 IPv6 주소를 IPv4 주소가 /etc/inet/hosts에 저장되는 방식과 동일하게 파일에 저장하십시오. ipnodes 파일은 hosts 파일과 동일한 형식 규칙을 사용합니다. /etc/inet/hosts에 대한 자세한 내용은 219 페이지 “hosts 데이터베이스”를 참조하십시오. ipnodes 파일에 대한 자세한 내용은 ipnodes(4) 매뉴얼 페이지를 참조하십시오.

IPv6 사용 응용 프로그램은 /etc/inet/ipnodes 데이터베이스를 사용합니다. IPv4 주소만 포함하는 기존 /etc/hosts 데이터베이스는 기존 응용 프로그램을 지원하기 위해 그대로 남아 있습니다. ipnodes 데이터베이스가 없으면 IPv6 사용 응용 프로그램은 기존 hosts 데이터베이스를 사용합니다.

주 – 주소를 추가하려면 IPv4 주소를 hosts 및 ipnodes 파일에 모두 추가해야 합니다. IPv6 주소는 ipnodes 파일에만 추가합니다.

예 10–3 /etc/inet/ipnodes 파일

이 예와 같이 호스트 이름 주소를 호스트 이름으로 그룹화해야 합니다.

```
#
# Internet IPv6 host table
# with both IPv4 and IPv6 addresses
#
::1     localhost
2001:db8:3b4c:114:a00:20ff:fe78:f37c   farsite.com farsite farsite-v6
fe80::a00:20ff:fe78:f37c     farsite-11.com farsitell
192.168.85.87                    farsite.com farsite farsite-v4
2001:db8:86c0:32:a00:20ff:fe87:9aba   nearsite.com nearsite nearsite-v6
fe80::a00:20ff:fe87:9aba     nearsite-11.com nearsitell
10.0.0.177                   nearsite.com nearsite nearsite-v4 loghost
```

netmasks 데이터베이스

네트워크에서 서브넷을 설정한 경우 netmasks 데이터베이스를 네트워크 구성의 일부분으로**만** 편집해야 합니다. netmasks 데이터베이스는 네트워크 및 연결된 서브넷 마스크의 목록으로 구성됩니다.

주 – 서브넷을 만들 때 각각의 새 네트워크는 별도의 물리적 네트워크여야 합니다. 서브넷을 단일 물리적 네트워크에 적용할 수 없습니다.

서브넷이란?

서브넷은 대규모 인터네트워크에서 제한된 32비트 IPv4 주소 지정 공간을 최대화하고 경로 지정 테이블의 크기를 줄이는 방식입니다. 서브넷은 주소 클래스를 사용하여 호스트 주소 공간의 일부분을 네트워크 주소에 할당하는 방식을 제공하므로 사용자가 더 많은 네트워크를 가질 수 있습니다. 새 네트워크 주소에 할당되는 호스트 주소 공간의 일부분을 **서브넷 번호**라고 합니다.

서브넷은 IPv4 주소 공간을 보다 효율적으로 사용할 수 있게 만들 뿐만 아니라 여러 관리 이점도 제공합니다. 네트워크 수가 늘어남에 따라 경로 지정이 매우 복잡해질 수 있습니다. 예를 들어, 소규모 조직이 각 로컬 네트워크에 클래스 C 번호를 지정할 수 있습니다. 이 조직의 규모가 커짐에 따라 서로 다른 여러 네트워크 번호를 관리하기가 복잡해질 수 있습니다. 더 나은 방법으로는 조직의 각 주요 부서에 클래스 B 네트워크 번호를 조금씩 할당하는 것입니다. 예를 들어, 엔지니어링 부서와 운영 부서에 각각 하나의 클래스 B 네트워크를 할당할 수 있습니다. 그런 다음 서브넷에 의해 얻은 추가 네트워크 번호를 사용하여 각 클래스 B 네트워크를 추가 네트워크로 구분할 수 있습니다. 또한 이러한 구분을 통해 라우터 간에 통신해야 하는 경로 지정 정보의 양을 줄일 수 있습니다.

IPv4 주소에 대한 네트워크 마스크 만들기

서브넷 프로세스의 일부분으로 네트워크 전역 **넷마스크**를 선택해야 합니다. 넷마스크는 서브넷 번호를 나타내는 호스트 주소 공간의 비트 및 비트 수와 호스트 번호를 나타내는 비트 및 비트 수를 결정합니다. 전체 IPv4 주소는 32비트로 구성된다는 점을 유념하십시오. 주소 클래스에 따라 호스트 주소 공간을 나타내는 데 최대 24비트 및 최소 8비트를 사용할 수 있습니다. 넷마스크는 netmasks 데이터베이스에 지정됩니다.

서브넷을 사용할 경우 TCP/IP를 구성하기 전에 넷마스크를 결정해야 합니다. 네트워크 구성의 일부분으로 운영 체제를 설치할 경우 Oracle Solaris 설치 프로그램이 네트워크에 대한 넷마스크를 요청합니다.

54 페이지 “IPv4 주소 지정 체계 설계”에 설명된 대로 32비트 IP 주소는 네트워크 부분과 호스트 부분으로 구성됩니다. 32비트는 4비트로 나뉩니다. 네트워크 클래스에 따라 각 바이트는 네트워크 번호 또는 호스트 번호에 지정됩니다.

예를 들어, 클래스 B IPv4 주소에서 왼쪽의 2바이트는 네트워크 번호에 지정되고 오른쪽의 2바이트는 호스트 번호에 지정됩니다. 클래스 B IPv4 주소 `172.16.10`에서 오른쪽의 2바이트를 호스트에 지정할 수 있습니다.

서브넷을 구현하려는 경우 호스트 번호에 지정되는 바이트의 일부 비트를 사용하여 서브넷 주소에 적용해야 합니다. 예를 들어, 16비트 호스트 주소 공간은 65,534개의 호스트에 대한 주소 지정을 제공합니다. 세번째 바이트를 서브넷 주소에 적용하고 네번째 바이트를 호스트 주소에 적용하는 경우 최대 254개의 네트워크에 대한 주소를 지정할 수 있습니다. 각 네트워크에는 최대 254개의 호스트가 가능합니다.

서브넷 주소 및 호스트 주소에 적용된 호스트 주소 바이트의 비트는 **서브넷 마스크**에 의해 결정됩니다. 서브넷 마스크는 서브넷 주소로 사용할 바이트에서 비트를 선택하는 데 사용됩니다. 넷마스크 비트는 인접해야 하지만 바이트 경계에 맞출 필요는 없습니다.

비트 논리적 AND 연산자를 사용하여 IPv4 주소에 넷마스크를 적용할 수 있습니다. 이 작업은 주소의 네트워크 번호 및 서브넷 번호 위치를 선택합니다.

넷마스크는 이진 표현으로 설명할 수 있습니다. 이진에서 십진으로 변환하는 데 계산기를 사용할 수 있습니다. 다음 예에서는 넷마스크의 십진 형식과 이진 형식을 모두 보여 줍니다.

넷마스크 `255.255.255.0`이 IPv4 주소 `172.16.41.101`에 적용되면 결과는 IPv4 주소 `172.16.41.0`입니다.

`172.16.41.101 & 255.255.255.0 = 172.16.41.0`

이진 형식으로 연산하면 다음과 같습니다.

10000001.10010000.00101001.01100101(IPv4 주소)

AND 연산

11111111.11111111.11111111.00000000(넷마스크)

이제 시스템이 네트워크 번호 `172.16` 대신 네트워크 번호 `172.16.41`을 찾습니다. 네트워크의 번호가 `172.16.41`인 경우 해당 번호는 시스템에서 점검하고 찾는 번호입니다. IPv4 주소 공간의 세번째 바이트에 최대 254개의 값을 지정할 수 있으므로 서브넷을 통해 이전에는 한 네트워크에만 사용 가능했던 공간을 이제는 네트워크 254개에 주소 공간을 만들 수 있습니다.

두 개의 추가 네트워크에 주소 공간만 제공하고 있는 경우 다음 서브넷 마스크를 사용할 수 있습니다.

`255.255.192.0`

이 넷마스크는 다음 결과를 제공합니다.

11111111.11111111.1100000.00000000

이 결과에서는 호스트 주소에 14비트를 계속 사용할 수 있습니다. 모든 0과 1이 예약되므로 호스트 번호에 대해 2비트 이상을 예약해야 합니다.

/etc/inet/netmasks 파일

네트워크에서 NIS 또는 LDAP을 실행하는 경우 이러한 이름 서비스에 대한 서버는 netmasks 데이터베이스를 유지 관리합니다. 로컬 파일을 이름 서비스로 사용하는 네트워크의 경우 이 정보는 /etc/inet/netmasks 파일에서 유지 관리됩니다.

주 – BSD 기반 운영 체제와의 호환성을 위해 /etc/netmasks 파일은 /etc/inet/netmasks에 대한 심볼릭 링크입니다.

다음 예에서는 클래스 B 네트워크에 대한 /etc/inet/netmasks 파일을 보여 줍니다.

예 10–4 클래스 B 네트워크에 대한 /etc/inet/netmasks 파일

```
# The netmasks file associates Internet Protocol (IPv4) address
# masks with IPv4 network numbers.
#
#     network-number    netmask
#
# Both the network-number and the netmasks are specified in
# “decimal dot” notation, e.g:
#
#        128.32.0.0   255.255.255.0
192.168.0.0  255.255.255.0
```

/etc/netmasks 파일이 존재하지 않는 경우 텍스트 편집기를 사용하여 해당 파일을 만듭니다. 다음 구문을 사용하십시오.

network-number netmask-number

자세한 내용은 netmasks(4) 매뉴얼 페이지를 참조하십시오.

넷마스크 번호를 만들 때 ISP 또는 인터넷 레지스트리(서브넷 번호가 아님) 및 넷마스크 번호로 지정된 네트워크 번호를 /etc/inet/netmasks에 입력합니다. 각 서브넷 마스크는 별도의 라인에 있어야 합니다.

예를 들면 다음과 같습니다.

```
128.78.0.0         255.255.248.0
```

또한 네트워크 번호에 대한 심볼릭 이름을 /etc/inet/hosts 파일에 입력할 수 있습니다. 그런 다음 네트워크 번호 대신 이러한 네트워크 이름을 명령에 대한 매개변수로 사용할 수 있습니다.

inetd 인터넷 서비스 데몬

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 sctp로 지정되어야 합니다.

- SMF 명령 사용에 대한 지침은 **Oracle Solaris 관리: 기본 관리**의 "SMF 명령줄 관리 유틸리티"를 참조하십시오.
- SMF 명령을 사용하여 SCTP를 통해 실행되는 서비스를 추가하는 작업은 128 페이지 "SCTP 프로토콜을 사용하는 서비스를 추가하는 방법"을 참조하십시오.
- IPv4 요청과 IPv6 요청을 모두 처리하는 서비스 추가에 대한 자세한 내용은 227 페이지 "inetd 인터넷 서비스 데몬"을 참조하십시오.

네트워크 데이터베이스 및 nsswitch.conf 파일

네트워크 데이터베이스는 네트워크를 구성하는 데 필요한 정보를 제공하는 파일입니다. 네트워크 데이터베이스는 다음과 같습니다.

- hosts
- netmasks
- ethers 데이터베이스
- bootparams
- protocols
- services
- networks

네트워크가 서브넷된 경우 구성 프로세스의 일부분으로 hosts 데이터베이스 및 netmasks 데이터베이스를 편집합니다. 시스템을 네트워크 클라이언트로 구성하는 데 두

네트워크 데이터베이스인 bootparams 및 ethers가 사용됩니다. 남은 데이터베이스는 운영 체제에서 사용되며 편집이 필요할 수 있습니다.

nsswitch.conf 파일은 네트워크 데이터베이스가 아니지만 이 파일을 관련 네트워크 데이터베이스와 함께 구성해야 합니다. nsswitch.conf는 로컬 파일, NIS, DNS 또는 LDAP과 같이 특정 시스템에 사용할 이름 서비스를 지정합니다.

네트워크 데이터베이스에 대한 이름 서비스의 영향

네트워크 데이터베이스의 형식은 네트워크에 대해 선택한 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.

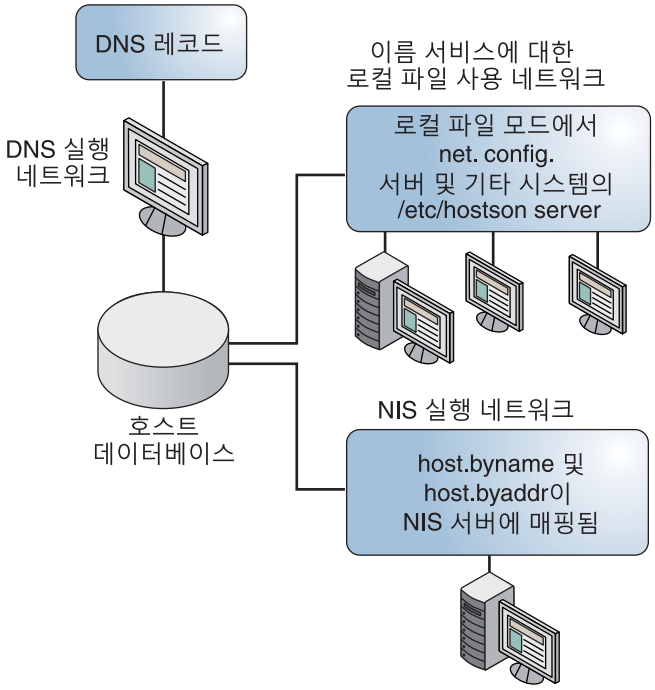
네트워크 데이터베이스는 다음과 같이 사용됩니다.

- 로컬 파일을 이름 서비스로 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.

다음 그림에서는 이러한 이름 서비스에서 사용하는 hosts 데이터베이스의 형식을 보여 줍니다.

그림 10–2 이름 서비스에서 사용하는 hosts 데이터베이스의 형식



다음 표에서는 네트워크 데이터베이스 및 해당 로컬 파일과 NIS 맵을 보여 줍니다.

주 – ipnodes 데이터베이스는 Solaris 10 11/06 이후의 Oracle Solaris 릴리스에서 제거되었습니다.

표 10–1 네트워크 데이터베이스 및 해당 이름 서비스 파일

네트워크 데이터베이스	로컬 파일	NIS 맵
hosts	/etc/inet/hosts	hosts.byaddr hosts.byname
netmasks	/etc/inet/netmasks	netmasks.byaddr
ethers	/etc/ethers	ethers.byname ethers.byaddr
bootparams	/etc/bootparams	bootparams
protocols	/etc/inet/protocols	protocols.byname protocols.bynumber
services	/etc/inet/services	services.byname
networks	/etc/inet/networks	networks.byaddr networks.byname

본 설명서에서는 로컬 파일을 이름 서비스로 사용하는 네트워크에서 볼 수 있는 네트워크 데이터베이스를 설명합니다.

- hosts 데이터베이스에 대한 정보는 219 페이지 “hosts 데이터베이스”에 나와 있습니다.
- netmasks 데이터베이스에 대한 정보는 224 페이지 “netmasks 데이터베이스”에 나와 있습니다.
- Solaris 10 11/06 및 이전 릴리스에서 ipnodes 데이터베이스에 대한 정보는 223 페이지 “ipnodes 데이터베이스”에 나와 있습니다.

NIS, DNS 및 LDAP에서 네트워크 데이터베이스 연결에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

nsswitch.conf 파일

/etc/nsswitch.conf 파일은 네트워크 데이터베이스의 검색 순서를 정의합니다. Oracle Solaris 설치 프로그램은 설치 프로세스 중에 지정하는 이름 서비스를 기반으로 로컬 시스템에 대한 기본 /etc/nsswitch.conf 파일을 만듭니다. 로컬 파일을 이름 서비스로 지정하고 “없음” 옵션을 선택한 경우 결과 nsswitch.conf 파일은 다음 예와 유사합니다.

예 10–5 파일을 이름 서비스로 사용하는 네트워크에 대한 nsswitch.conf

```
# /etc/nsswitch.files:
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file contains "switch.so" as a
# nametoaddr library for "inet" transports.

passwd:          files
group:           files
hosts:           files
networks:        files
protocols:       files
rpc:             files
ethers:          files
netmasks:        files
bootparams:      files
publickey:       files
# At present there isn't a 'files' backend for netgroup; the
# system will figure it out pretty quickly,
# and won't use netgroups at all.
netgroup:        files
automount:       files
aliases:         files
services:        files
sendmailvars:    files
```

해당 파일은 nsswitch.conf(4) 매뉴얼 페이지에 자세히 설명되어 있습니다. 기본 구문은 다음과 같습니다.

database name-service-to-search

database 필드에는 운영 체제에서 검색되는 여러 유형의 데이터베이스 중 하나를 나열할 수 있습니다. 예를 들어, 이 필드에는 passwd 또는 aliases나 네트워크 데이터베이스와 같이 사용자에게 영향을 주는 데이터베이스를 지정할 수 있습니다. 매개변수 *name-service-to-search*에는 네트워크 데이터베이스의 값 files, nis 또는 nis+를 사용할 수 있습니다. hosts 데이터베이스는 dns를 검색할 이름 서비스로 사용할 수 있습니다. 또한 nis+ 및 files와 같이 둘 이상의 이름 서비스를 나열할 수 있습니다.

예 10–5에 표시된 검색 옵션은 files뿐입니다. 따라서 로컬 시스템은 해당 /etc 및 /etc/inet 디렉토리에 있는 파일에서 네트워크 데이터베이스 정보, 보안 및 자동 마운트 정보를 가져옵니다.

nsswitch.conf 변경

/etc 디렉토리에는 Oracle Solaris 설치 프로그램에서 만든 nsswitch.conf 파일이 포함됩니다. 또한 이 디렉토리에는 다음 이름 서비스에 대한 템플리트 파일이 포함됩니다.

- nsswitch.files
- nsswitch.nis

한 이름 서비스에서 다른 이름 서비스로 변경하려는 경우 해당 템플리트를 nsswitch.conf에 복사할 수 있습니다. 또한 선택적으로 nsswitch.conf 파일을 편집하고 기본 이름 서비스를 변경하여 개별 데이터베이스를 검색할 수 있습니다.

예를 들어, NIS를 실행하는 네트워크에서 네트워크 클라이언트에 대한 nsswitch.conf 파일을 변경해야 할 수 있습니다. bootparams 및 ethers 데이터베이스에 대한 검색 경로는 files를 첫번째 옵션으로 나열한 다음 nis를 나열해야 합니다. 다음 예에서는 올바른 검색 경로를 보여 줍니다.

예 10–6 NIS를 실행하는 네트워크의 클라이언트에 대한 nsswitch.conf

```
# /etc/nsswitch.conf:#
.
.
passwd:         files nis
group:          files nis

# consult /etc "files" only if nis is down.
hosts:          nis     [NOTFOUND=return] files
networks:       nis     [NOTFOUND=return] files
protocols:      nis     [NOTFOUND=return] files
rpc:            nis     [NOTFOUND=return] files
ethers:         files   [NOTFOUND=return] nis
netmasks:       nis     [NOTFOUND=return] files
bootparams:     files   [NOTFOUND=return] nis
```

예 10–6 NIS를 실행하는 네트워크의 클라이언트에 대한 nsswitch.conf *(계속)*

```
publickey:      nis
netgroup:       nis

automount:      files nis
aliases:        files nis

# for efficient getservbyname() avoid nis
services:       files nis
sendmailvars:   files
```

이름 서비스 스위치에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 를 참조하십시오.

bootparams 데이터베이스

bootparams 데이터베이스는 네트워크 클라이언트 모드로 부트되도록 구성된 시스템에서 사용되는 정보를 포함합니다. 네트워크에 네트워크 클라이언트가 있는 경우 이 데이터베이스를 편집해야 합니다. 절차는 100 페이지 “네트워크 클라이언트 구성”을 참조하십시오. 데이터베이스는 /etc/bootparams 파일에 입력된 정보를 사용하여 작성됩니다.

bootparams(4) 매뉴얼 페이지에는 이 데이터베이스에 대한 전체 구문이 포함되어 있습니다. 기본 구문은 다음과 같습니다.

system-name file-key-server-name:pathname

각 네트워크 클라이언트 시스템의 경우 항목에는 클라이언트 이름, 키 목록, 서버 이름 및 경로 이름과 같은 정보가 포함될 수 있습니다. 각 입력 항목의 첫번째 항목은 클라이언트 시스템의 이름입니다. 첫번째 항목을 제외한 모든 항목은 옵션입니다. 예는 다음과 같습니다.

예 10–7 bootparams 데이터베이스

```
myclient   root=myserver : /nfsroot/myclient  \
swap=myserver : /nfsswap//myclient \
dump=myserver : /nfsdump/myclient
```

이 예에서 dump= 용어는 클라이언트 호스트가 덤프 파일을 찾지 않도록 지정합니다.

bootparams에 대한 와일드카드 항목

대부분의 경우 클라이언트를 지원하도록 bootparams 데이터베이스를 편집할 때 와일드카드 항목을 사용합니다. 이 항목은 다음과 같습니다.

```
*  root=server:/path dump=:
```

별표(*) 와일드카드는 이 항목이 bootparams 데이터베이스 내에서 특별하게 이름이 지정되지 않은 모든 클라이언트에 적용됨을 나타냅니다.

ethers 데이터베이스

ethers 데이터베이스는 /etc/ethers 파일에 입력된 정보를 사용하여 작성됩니다. 이 데이터베이스는 호스트 이름을 해당 MAC(*Media Access Control*) 주소에 연결합니다. RARP 데몬을 실행하고 있는 경우에만 ethers 데이터베이스를 만들어야 합니다. 즉, 네트워크 클라이언트를 구성하고 있는 경우 이 데이터베이스를 만들어야 합니다.

RARP는 이 파일을 사용하여 MAC 주소를 IP 주소에 매핑합니다. RARP 데몬 in.rarpd를 실행하고 있는 경우 ethers 파일을 설정하고 데몬을 실행 중인 모든 호스트에서 이 파일을 유지 관리하여 네트워크에 변경 내용을 적용해야 합니다.

ethers(4) 매뉴얼 페이지에는 이 데이터베이스에 대한 전체 구문이 포함되어 있습니다. 기본 구문은 다음과 같습니다.

MAC-address hostname #comment

MAC-address	호스트의 MAC 주소
hostname	호스트의 공식 이름
#comment	파일의 항목에 첨부할 설명

장비 제조업체가 MAC 주소를 제공합니다. 시스템 부트 프로세스 중 시스템에서 MAC 주소를 표시하지 않는 경우 자세한 내용은 하드웨어 설명서를 참조하십시오.

ethers 데이터베이스에 항목을 추가할 때 호스트 이름이 hosts의 기본 이름에 해당되는지, 그리고 Solaris 10 11/06 및 이전 릴리스의 경우 별칭이 아닌 ipnodes 데이터베이스에 해당되는지 확인합니다. 예를 들면 다음과 같습니다.

예 10–8 ethers 데이터베이스의 항목

```
8:0:20:1:40:16  fayoum
8:0:20:1:40:15  nubian
8:0:20:1:40:7   sahara    # This is a comment
8:0:20:1:40:14  tenere
```

기타 네트워크 데이터베이스

나머지 네트워크 데이터베이스는 편집이 필요한 경우가 거의 없습니다.

networks 데이터베이스

networks 데이터베이스는 일부 응용 프로그램이 번호가 아닌 이름을 사용 및 표시할 수 있도록 네트워크 이름을 네트워크 번호와 연결합니다. networks 데이터베이스는 /etc/inet/networks 파일의 정보를 기반으로 합니다. 이 파일에는 네트워크가 라우터를 통해 연결되는 모든 네트워크의 이름이 포함됩니다.

Oracle Solaris 설치 프로그램은 초기 networks 데이터베이스를 구성합니다. 그러나 기존 네트워크 토폴로지에 새 네트워크를 추가하는 경우 이 데이터베이스를 업데이트해야 합니다.

networks(4) 매뉴얼 페이지에는 /etc/inet/networks에 대한 전체 구문이 포함되어 있습니다. 기본 형식은 다음과 같습니다.

network-name network-number nickname(s) #comment

network-name	네트워크의 공식 이름
network-number	ISP 또는 인터넷 레지스트리가 지정하는 번호
nickname	알려진 네트워크의 다른 이름
#comment	파일의 항목에 첨부할 설명

networks 파일을 유지 관리해야 합니다. netstat 프로그램은 이 데이터베이스의 정보를 사용하여 상태 표를 생성합니다.

샘플 /etc/networks 파일은 다음과 같습니다.

예 10–9 /etc/networks 파일

```
#ident    "@(#)networks    1.4    92/07/14 SMI"    /* SVr4.0 1.1    */
#
# The networks file associates Internet Protocol (IP) network
# numbers with network names. The format of this file is:
#
#     network-name              network-number              nicnames . . .

# The loopback network is used only for intra-machine communication
loopback              127

#
# Internet networks
#
arpanet     10       arpa  # Historical
#
# local networks

eng   192.168.9 #engineering
acc   192.168.5 #accounting
prog  192.168.2 #programming
```

protocols 데이터베이스

protocols 데이터베이스는 시스템에 설치된 TCP/IP 프로토콜 및 해당 프로토콜 번호를 나열합니다. Oracle Solaris 설치 프로그램은 자동으로 데이터베이스를 만듭니다. 이 파일은 관리가 거의 필요하지 않습니다.

protocols(4) 매뉴얼 페이지에서는 이 데이터베이스의 구문을 설명합니다. /etc/inet/protocols 파일의 예는 다음과 같습니다.

예 10–10 /etc/inet/protocols 파일

```
#
# Internet (IP) protocols
#
ip    0   IP    # internet protocol, pseudo protocol number
icmp  1   ICMP  # internet control message protocol
tcp   6   TCP   # transmission control protocol
udp  17   UDP   # user datagram protocol
```

services 데이터베이스

services 데이터베이스는 TCP/UDP 서비스의 이름 및 잘 알려진 포트 번호를 나열합니다. 이 데이터베이스는 네트워크 서비스를 호출하는 프로그램에서 사용됩니다. Oracle Solaris 설치 프로그램은 자동으로 services 데이터베이스를 만듭니다. 일반적으로 이 데이터베이스는 관리가 필요하지 않습니다.

services(4) 매뉴얼 페이지에는 전체 구문 정보가 포함되어 있습니다. 다음은 일반적인 /etc/inet/services 파일의 일부를 발췌한 것입니다.

예 10–11 /etc/inet/services 파일

```
#
# Network services
#
echo      7/udp
echo      7/tcp
echo      7/sctp6
discard   9/udp     sink null
discard   11/tcp
daytime   13/udp
daytime   13/tcp
netstat   15/tcp
ftp-data  20/tcp
ftp       21/tcp
telnet    23/tcp
time      37/tcp    timeserver
time      37/udp    timeserver
name      42/udp    nameserver
whois     43/tcp    nickname
```

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 5–1 및 표 5–2를 참조하십시오.

RIP(Routing Information Protocol)

RIP는 시스템이 부트될 때 자동으로 시작되는 경로 지정 데몬인 `in.routed`에 의해 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.

네트워크 클래스

주 – 이전의 많은 네트워크가 아직도 클래스 기반이기는 하지만 더 이상 IANA에서 클래스 기반 네트워크 번호를 사용할 수 없습니다.

이 절에서는 IPv4 네트워크 클래스에 대한 자세한 정보를 제공합니다. 각 클래스는 주소의 네트워크 부분에 비트를 더 추가하거나 더 줄이면서 32비트 IPv4 주소 공간을 다양하게 사용합니다. 이러한 클래스를 클래스 A, 클래스 B 및 클래스 C라고 합니다.

클래스 A 네트워크 번호

클래스 A 네트워크 번호는 IPv4 주소의 첫번째 8비트를 해당 "네트워크 부분"으로 사용합니다. 남은 24비트는 다음 그림과 같이 IPv4 주소의 호스트 부분을 포함합니다.

그림 10–3 클래스 A 주소의 바이트 지정

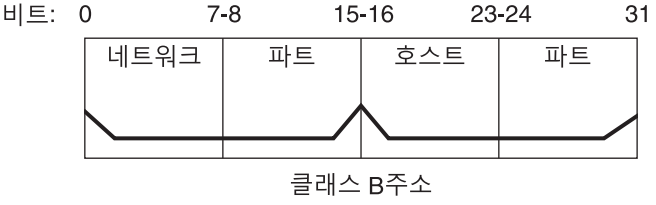
클래스 A주소

클래스 A 네트워크 번호의 첫번째 바이트에 지정된 값 범위는 0–127입니다. IPv4 주소가 `75.4.10.4`라고 가정합니다. 첫번째 바이트의 값 75는 호스트가 클래스 A 네트워크에 있음을 나타냅니다. 남은 바이트 `4.10.4`는 호스트 주소를 설정합니다. 클래스 A 번호의 첫번째 바이트만 IANA에서 등록합니다. 남은 3바이트 사용은 네트워크 번호의 소유자가 결정합니다. 127개의 클래스 A 네트워크만 존재합니다. 이러한 번호 중 하나는 각각 최대 16,777,214개의 호스트를 수용할 수 있습니다.

클래스 B 네트워크 번호

클래스 B 네트워크 번호는 네트워크 번호에 대해 16비트를 사용하고 호스트 번호에 대해 16비트를 사용합니다. 클래스 B 네트워크 번호의 첫번째 바이트 범위는 128–191입니다. 번호 `172.16.50.56`에서 첫번째 2바이트인 `172.16`은 IANA에 등록되어 있으며 네트워크 주소를 구성합니다. 마지막 2바이트인 `50.56`은 호스트 주소를 포함하며 네트워크 번호의 소유자에 의해 지정됩니다. 다음 그림에서는 클래스 B 주소를 그래픽으로 보여 줍니다.

그림 10–4 클래스 B 주소의 바이트 지정



클래스 B는 대개 네트워크에 호스트가 여러 개 있는 조직에 지정됩니다.

클래스 C 네트워크 번호

클래스 C 네트워크 번호는 네트워크 번호에 대해 24비트를 사용하고 호스트 번호에 대해 8비트를 사용합니다. 클래스 C 네트워크 번호는 호스트 수(최대 값: 254)가 적은 네트워크에 대해 적합합니다. 클래스 C 네트워크 번호는 IPv4 주소의 첫번째 3바이트를 사용합니다. 네번째 바이트만 네트워크 소유자에 의해 지정됩니다. 다음 그림에서는 클래스 C 주소의 바이트를 그래픽으로 보여 줍니다.

그림 10–5 클래스 C 주소의 바이트 지정

클래스 C 네트워크 번호의 첫번째 바이트 범위는 192–223입니다. 두번째 바이트와 세번째 바이트의 범위는 각각 1– 255입니다. 일반적인 클래스 C 주소는 `192.168.2.5`입니다. 첫번째 3바이트인 `192.168.2`는 네트워크 번호를 형성합니다. 이 예의 마지막 바이트인 5는 호스트 번호입니다.

◆ ◆ ◆ 11 장

11 IPv6 세부 개요(참조)

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

IPv6 개요는 3 장, “IPv6 소개(개요)”를 참조하십시오. IPv6 지원 네트워크 구성에 대한 작업은 7 장, “IPv6 네트워크 구성(작업)”을 참조하십시오.

IPv6 세부 개요의 새로운 내용

Solaris 10 7/07에서는 `/etc/inet/ipnodes` 파일은 더 이상 사용되지 않습니다. 개별 절차에 설명된 대로 `/etc/inet/ipnodes`는 이전 Solaris 10 릴리스에서만 사용하십시오.

IPv6 주소 지정 형식 고급 정보

3 장, “IPv6 소개(개요)”에서는 가장 일반적인 IPv6 주소 지정 형식인 유니캐스트 사이트 주소 및 링크 로컬 주소에 대해 설명합니다. 이 섹션에는 3 장, “IPv6 소개(개요)”에서 자세히 다루지 않는 주소 지정 형식에 대한 고급 설명이 포함되어 있습니다.

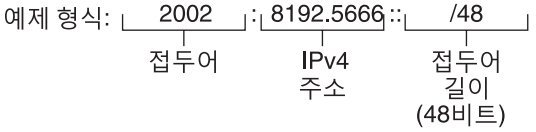
6to4 파생 주소

라우터 또는 호스트 끝점에서 6to4 터널을 구성할 계획이면 끝점 시스템의 `/etc/inet/ndpd.conf` 파일에서 6to4 사이트 접두어를 알려야 합니다. 6to4 터널 구성에 대한 소개 및 작업 정보는 179 페이지 “6to4 터널을 구성하는 방법”을 참조하십시오.

다음 그림에서는 6to4 사이트 접두어의 각 부분을 보여줍니다.

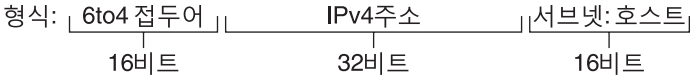
그림 11–1 6to4 사이트 접두어의 각 부분

예제 6to4 주소: 2002:8192:5666::/48



다음 그림은 `ndpd.conf` 파일에 포함될 수 있는 6to4 사이트의 서브넷 접두어의 각 부분을 보여줍니다.

그림 11–2 6to4 서브넷 접두어의 각 부분



예제 6to4 주소: 2002:8192.5666:1: :/64

이 테이블에서는 6to4 서브넷 접두어의 각 부분, 해당 길이 및 정의에 대해 설명합니다.

부분	길이	정의
접두어	16비트	6to4 접두어 레이블 2002(0x2002)

부분	길이	정의
IPv4 주소	32비트	6to4 인터페이스에 이미 구성된 고유한 IPv4 주소. 알림을 위해서는 IPv4의 점으로 구분된 10진수 표시가 아니라 IPv4 주소의 16진수 표시를 지정합니다.
서브넷 ID	16비트	6to4 사이트에서 링크에 대해 고유한 값이어야 하는 서브넷 ID

호스트의 6to4 파생 주소 지정

IPv6 호스트가 라우터 알림을 사용해서 6to4 파생 접두어를 수신하면 호스트가 인터페이스에서 6to4 파생 주소를 자동으로 다시 구성합니다. 주소 형식은 다음과 같습니다.

prefix:IPv4-address:subnet-ID:interface-ID/64

6to4 인터페이스를 포함하는 호스트에서 `ifconfig -a` 명령의 출력은 다음과 같습니다.

```
qfe1:3: flags=2180841<UP,RUNNING,MULTICAST,ADDRCONF,ROUTER,IPv6>
 mtu 1500 index 7
        inet6 2002:8192:56bb:9258:a00:20ff:fea9:4521/64
```

이 출력에서 6to4 파생 주소는 `inet6` 다음에 표시됩니다.

이 테이블에서는 6to4 파생 주소의 각 부분, 해당 길이 및 각 부분이 제공하는 정보에 대해 설명합니다.

주소 부분	길이	정의
prefix	16비트	`2002` - 6to4 접두어
IPv4-address	32비트	`8192:56bb` - 6to4 라우터에 구성된 6to4 의사 인터페이스에 대한 16진수로 표시된 IPv4 주소
subnet-ID	16비트	`9258` - 이 호스트가 멤버인 서브넷의 주소
interface-ID	64비트	`a00:20ff:fea9:4521` - 6to4에 대해 구성된 호스트 인터페이스의 인터페이스 ID

IPv6 멀티캐스트 주소 세부 정보

IPv6 멀티캐스트 주소는 **멀티캐스트 그룹**이라는 정의된 인터페이스 그룹에 동일 정보 또는 서비스를 배포하기 위한 방법을 제공합니다. 일반적으로 멀티캐스트 그룹의 인터페이스는 서로 다른 노드에 있습니다. 인터페이스는 여러 개의 멀티캐스트 그룹에 속할 수 있습니다. 멀티캐스트 주소로 전송되는 패킷은 멀티캐스트 그룹의 모든 멤버로 이동합니다. 예를 들어, 멀티캐스트 주소는 IPv4 브로드캐스트 주소 기능과 비슷한 정보 브로드캐스트를 위해 사용할 수 있습니다.

다음 표에서는 멀티캐스트 주소의 형식을 보여줍니다.

표 11–1 IPv6 멀티캐스트 주소 형식

8비트	4비트	4비트	8비트	8비트	64비트	32비트
11111111	*FLGS*	SCOP	*Reserved*	*Plen*	*Network prefix*	*Group ID*

다음은 각 필드의 컨텐츠에 대한 요약 설명입니다.

- 11111111 – 주소를 멀티캐스트 주소로 식별합니다.
- *FLGS* – 0,0,P,T의 4개 플래그로 구성된 세트입니다. 처음 두 개의 플래그는 0이어야 합니다. P 필드는 다음 값 중 하나를 포함합니다.
  - 0 = 네트워크 접두어를 기반으로 지정되지 않은 멀티캐스트 주소
  - 1 = 네트워크 접두어를 기반으로 지정된 멀티캐스트 주소

  P가 1로 설정되었으면 T도 1이어야 합니다.
- *Reserved* - 예약된 0 값
- *Plen* - 사이트 접두어를 기반으로 지정된 멀티캐스트 주소에 대해 사이트 접두어에서 서브넷을 식별하는 비트 수
- *Group ID* - 영구 또는 동적인 멀티캐스트 그룹의 식별자

멀티캐스트 형식에 대한 전체 세부 정보는 RFC 3306, "Unicast-Prefix-based IPv6 Multicast Addresses (`ftp://ftp.rfc-editor.org/in-notes/rfc3306.txt`)를 참조하십시오.

일부 IPv6 멀티캐스트 주소는 IANA(Internet Assigned Numbers Authority)에서 영구적으로 지정됩니다. 이에 대한 일부 예로는 모든 IPv6 호스트 및 IPv6 라우터에 필요한 All Nodes 멀티캐스트 주소 및 All Routers 멀티캐스트 주소를 들 수 있습니다. IPv6 멀티캐스트 주소를 동적으로 할당할 수도 있습니다. 멀티캐스트 주소 및 그룹의 올바른 사용에 대한 자세한 내용은 RFC 3307, "Allocation Guidelines for IPv6 Multicast Addresses"를 참조하십시오.

IPv6 패킷 헤더 형식

IPv6 프로토콜은 기본 IPv6 헤더 및 IPv6 확장 헤더를 포함하는 헤더 세트를 정의합니다. 다음 그림에서는 IPv6 헤더에 표시되는 필드 및 필드가 나타나는 순서를 보여줍니다.

그림 11–3 IPv6 기본 헤더 형식

다음 목록에서는 각 헤더 필드의 기능에 대해 설명합니다.

- **버전** – Internet Protocol의 4비트 버전 번호 = 6.
- **트래픽 클래스** – 8비트 트래픽 클래스 필드.
- **플로우 레이블** – 20비트 필드.
- **페이로드 길이** – IPv6 헤더 다음에 오는 나머지 패킷인 16비트 unsigned integer(옥테트).
- **다음 헤더** – 8비트 선택기. IPv6 헤더 바로 다음에 오는 헤더 유형을 식별합니다. IPv4 프로토콜 필드와 동일한 값을 사용합니다.
- **홉 제한** – 8비트 unsigned integer. 패킷 다음에 오는 각 노드마다 1씩 감소됩니다. 홉 제한이 0으로 줄어들면 패킷이 삭제됩니다.
- **소스 주소** – 128비트. 패킷 초기 발신자의 주소입니다.
- **대상 주소** – 128비트. 의도된 패킷 수신자의 주소입니다. 선택적인 경로 지정 헤더가 있는 경우 의도된 수신자는 반드시 실제 수신자일 필요가 없습니다.

IPv6 확장 헤더

IPv6 옵션은 패킷의 IPv6 헤더 및 전송 계층 헤더 사이에 있는 별도의 확장 헤더에 배치됩니다. 대부분의 IPv6 확장 헤더는 패킷이 해당 최종 대상에 도착할 때까지 패킷의 전달 경로를 따라 라우터에서 검사 또는 처리되지 않습니다. 이 기능은 옵션을 포함하는 패킷에 대한 라우터 성능을 크게 향상시켜 줍니다. IPv4에서 옵션이 있으면 라우터가 모든 옵션을 검사해야 합니다.

IPv4 옵션과 달리 IPv6 확장 헤더는 임의 길이일 수 있습니다. 또한 패킷이 포함하는 옵션 수도 40바이트로 제한되지 않습니다. 이 기능은 IPv6 옵션이 처리되는 방식 외에도 IPv4에 사용할 수 없는 기능에 대해 IPv6 옵션을 사용할 수 있도록 허용합니다.

이후 옵션 헤더를 처리할 때의 성능 및 이후 전송 프로토콜을 향상시키려면 IPv6 옵션 길이가 항상 8옥테트의 배수여야 합니다. 길이가 8옥테트의 배수이면 이후 헤더가 정렬된 상태로 유지됩니다.

현재 정의된 IPv6 확장 헤더는 다음과 같습니다.

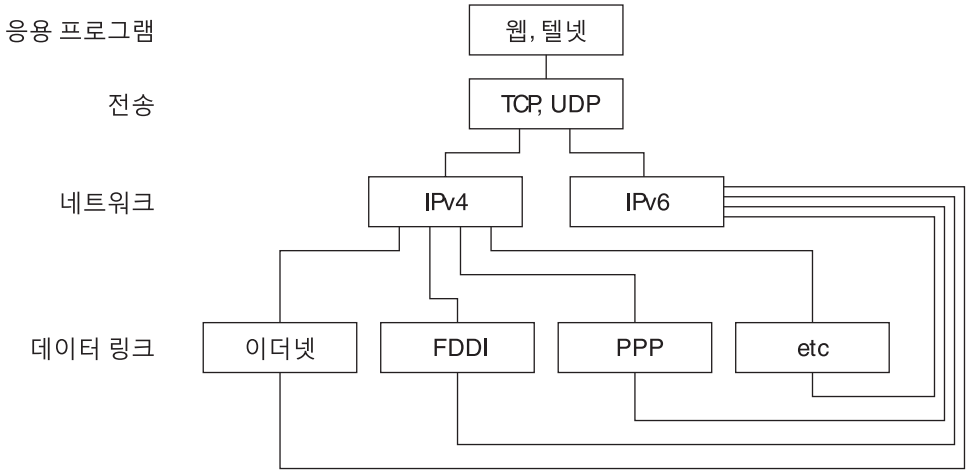
- **경로 지정** – 확장된 경로 지정(예: IPv4 느슨한 소스 경로)
- **단편화** – 단편화 및 재어셈블
- **인증** – 무결성 및 인증, 보안
- **보안 페이로드 캡슐화** – 기밀성
- **홉 단위 옵션** – 홉 단위 처리가 필요한 특수 옵션
- **대상 옵션** – 대상 노드에서 검사할 선택적인 정보

이중 스택 프로토콜

이중 스택이라는 용어는 일반적으로 응용 프로그램에서 네트워크 계층까지 프로토콜 스택에 있는 모든 레벨의 완전한 중복성을 의미합니다. 완전한 중복성의 한 가지 예로 OSI와 TCP/IP 프로토콜을 모두 실행하는 시스템을 들 수 있습니다.

Oracle Solaris는 **이중 스택**입니다. 즉, Oracle Solaris는 IPv4와 IPv6 프로토콜을 모두 구현합니다. 운영 체제를 설치할 때는 IP 계층에서 IPv6 프로토콜을 사용으로 설정할지 또는 기본 IPv4 프로토콜만 사용할지를 선택할 수 있습니다. 나머지 TCP/IP 스택은 동일합니다. 따라서 동일한 전송 프로토콜인 TCP, UDP 및 SCTP를 IPv4 및 IPv6 모두에서 실행할 수 있습니다. 또한 동일한 응용 프로그램을 IPv4 및 IPv6 모두에서 실행할 수 있습니다. 그림 11–4에서는 인터넷 프로토콜 제품군의 여러 계층을 통해 IPv4 및 IPv6 프로토콜이 이중 스택으로 작동하는 방법을 보여줍니다.

그림 11–4 이중 스택 프로토콜 아키텍처



이중 스택 시나리오에서 호스트 및 라우터 모두의 하위 세트는 IPv4뿐만 아니라 IPv6을 지원하도록 업그레이드됩니다. 이중 스택 접근 방법은 업그레이드된 노드가 항상 IPv4를 사용해서 IPv4 전용 노드와 상호 작용할 수 있도록 보장합니다.

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.

- 245 페이지 “ndpd.conf 구성 파일”
- 249 페이지 “IPv6 인터페이스 구성 파일”
- 250 페이지 “/etc/inet/ipaddrsel.conf 구성 파일”

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 11–2 /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 11–3 /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 11–3 /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 166 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 173 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 170 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 170 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
TmpRegenAdvance	False	임시 주소가 제거되기 전에 제공되는 선행 시간을 지정합니다. 자세한 내용은 170 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.
TmpValidLifetime	False	임시 주소의 유효 수명을 설정합니다. 자세한 내용은 170 페이지 "임시 주소를 구성하는 방법"을 참조하십시오.

다음 표는 IPv6 접두어 구성에 사용되는 변수를 보여줍니다.

표 11–4 /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	설정되지 않음	접두어의 선호 만료 날짜를 지정합니다.
AdvPreferredLifetime	604800초	Prefix Information(접두어 정보) 옵션의 선호 수명에 지정할 값을 지정합니다.
AdvValidExpiration	설정되지 않음	접두어의 유효 만료 날짜를 지정합니다.
AdvValidLifetime	2592000초	구성할 접두어의 유효 수명을 지정합니다.

예 11–1 /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 11–1 /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
```

IPv6 인터페이스 구성 파일

IPv6은 시작 시에 `/etc/hostname6.`*interface* 파일을 사용하여 IPv6 논리적 인터페이스를 자동으로 정의합니다. Oracle Solaris 설치 중 IPv6 Enabled(IPv6 사용) 옵션을 선택하면 `/etc/hostname.` *interface* 파일 외에도 설치 프로그램이 기본 네트워크 인터페이스에 대해 `/etc/hostname6.`*interface* 파일을 만듭니다.

설치 중에 물리적 인터페이스가 두 개 이상 감지되면 이러한 인터페이스를 구성할지 여부를 묻는 프롬프트가 표시됩니다. 설치 프로그램은 사용자가 표시하는 각 추가 인터페이스에 대해 IPv4 물리적 인터페이스 구성 파일 및 IPv6 논리적 인터페이스 구성 파일을 만듭니다.

IPv4 인터페이스에서와 같이 Oracle Solaris 설치 후 IPv6 인터페이스를 수동으로 구성할 수도 있습니다. 새 인터페이스에 대해 `/etc/hostname6.` *interface* 파일을 만듭니다. 인터페이스를 수동으로 구성하는 방법에 대한 자세한 내용은 6 장, “네트워크 인터페이스 관리(작업)”를 참조하십시오.

네트워크 인터페이스 구성 파일 이름의 구문은 다음과 같습니다.

`hostname.`*interface*
`hostname6.`*interface*

interface 변수의 구문은 다음과 같습니다.

dev`[.`*module*`[.`*module* `...]]`*PPA*

dev 네트워크 인터페이스 장치를 나타냅니다. 장치는 물리적 네트워크 인터페이스(예: `eri` 또는 `qfe`) 또는 논리적 인터페이스(예: 터널)일 수 있습니다. 자세한 내용은 249 페이지 “IPv6 인터페이스 구성 파일”을 참조하십시오.

Module 장치가 연결될 때 장치에 푸시할 하나 이상의 `STREAMS` 모듈을 나열합니다.

PPA 물리적 연결 지점을 나타냅니다.

[.[.]] 구문도 허용됩니다.

예 11–2 IPv6 인터페이스 구성 파일

다음은 유효한 IPv6 구성 파일 이름의 예입니다.

```
hostname6.qfe0
hostname.ip.tun0
hostname.ip6.tun0
hostname6.ip6to4tun0
hostname6.ip.tun0
hostname6.ip6.tun0
```

/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 11–5에 표시된 내용이 포함됩니다.

/etc/inet/ipaddrsel.conf의 내용은 편집할 수 있습니다. 그러나 대부분의 경우 이 파일을 수정하지 않는 것이 좋습니다. 수정이 필요할 경우 210 페이지 "IPv6 주소 선택 정책 테이블을 관리하는 방법" 절차를 참조하십시오. ippaddrsel.conf에 대한 자세한 내용은 251 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 11–5 IPv6 주소 선택 정책 테이블

접두어	우선 순위	정의
::1/128	50	루프백
::/0	40	기본값

표 11–5 IPv6 주소 선택 정책 테이블 *(계속)*

접두어	우선순위	정의
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 ifconfig를 사용하여 이 주소를 선호 주소로 플래그 지정할 수 있습니다.
- 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 고유 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 고유 IPv6 사이트로 전송합니다. 6to4 지원 사이트가 고유 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 272 페이지 “6to4 릴레이 라우터에 대한 터널 고려 사항”을 참조하십시오. 6to4 릴레이 라우터 지원을 사용하려는 경우 179 페이지 “6to4 터널을 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 11–3 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 11–4 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 11–5 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.

예 11–5 6to4 릴레이 라우터를 지정하여 상태 표시 *(계속)*

`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 지원에 대한 `ifconfig` 명령 확장

`ifconfig` 명령은 IPv6 인터페이스 및 연결할 터널링 모듈을 사용으로 설정합니다. `ifconfig`는 ioctls의 확장 세트를 사용해서 IPv4 및 IPv6 네트워크 인터페이스를 모두 구성합니다. 다음은 IPv6 작업을 지원하는 `ifconfig` 옵션에 대한 설명입니다. `ifconfig`와 관련된 IPv4 및 IPv6 작업 모두의 범위는 190 페이지 “ifconfig 명령으로 인터페이스 구성 모니터링”를 참조하십시오.

`index`	인터페이스 인덱스를 설정합니다.
`tsrc/tdst`	터널 소스 또는 대상을 설정합니다.
`addif`	사용 가능한 다음 논리적 인터페이스를 만듭니다.
`removeif`	특정 IP 주소의 논리적 인터페이스를 삭제합니다.
`destination`	인터페이스에 대한 지점 간 대상 주소를 설정합니다.
`set`	인터페이스에 대한 주소 또는 넷마스크를 설정하거나 둘 다 설정합니다.
`subnet`	인터페이스의 서브넷 주소를 설정합니다.
`xmit/-xmit`	인터페이스에서 패킷 전송을 사용 또는 사용 안함으로 설정합니다.

7 장, “IPv6 네트워크 구성(작업)”에서는 IPv6 구성 절차를 제공합니다.

예 11–6 `ifconfig` 명령의 `-addif` 옵션을 사용하여 논리적 IPv6 인터페이스 추가

`ifconfig` 명령의 다음 형식에서는 `hme0:3` 논리적 인터페이스를 만듭니다.

```
# ifconfig hme0 inet6 addif up
Created new logical interface hme0:3
```

이 형식의 `ifconfig`는 새로운 인터페이스 만들기를 확인합니다.

```
# ifconfig hme0:3 inet6
hme0:3: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        inet6  inet6 fe80::203:baff:fe11:b321/10
```

예 11–7 `ifconfig` 명령의 `-removeif` 옵션을 사용하여 논리적 IPv6 인터페이스 제거

`ifconfig` 명령의 다음 형식에서는 `hme0:3` 논리적 인터페이스를 제거합니다.

```
# ifconfig hme0:3 inet6 down
```

예 11–7 ifconfig 명령의 -removeif 옵션을 사용하여 논리적 IPv6 인터페이스 제거 *(계속)*

```
# ifconfig hme0 inet6 removeif 1234::5678
```

예 11–8 ifconfig를 사용하여 IPv6 터널 소스 구성

```
# ifconfig ip.tun0 inet6 plumb index 13
```

물리적 인터페이스 이름과 연결할 터널을 엽니다.

```
# ifconfig ip.tun0 inet6
ip.tun0: flags=2200850<POINTOPOINT,RUNNING,MULTICAST,NONUD,
#IPv6> mtu 1480 index 13
        inet tunnel src 0.0.0.0
        inet6 fe80::/10 --> ::
```

TCP/IP가 터널 장치를 사용하고 장치 상태를 보고하기 위해 필요한 스트림을 구성합니다.

```
# ifconfig ip.tun0 inet6 tsrc 120.46.86.158 tdst 120.46.86.122
```

터널의 소스 및 대상 주소를 구성합니다.

```
# ifconfig ip.tun0 inet6
ip.tun0: flags=2200850<POINTOPOINT,RUNNING,MULTICAST,NONUD,
IPv6> mtu 1480 index 13
        inet tunnel src 120.46.86.158  tunnel dst 120.46.86.122
        inet6 fe80::8192:569e/10 --> fe80::8192:567a
```

구성 후 장치의 새 상태를 보고합니다.

예 11–9 ifconfig를 통해 6to4 터널 구성(긴 형식)

이 6to4 의사 인터페이스 구성 예에서는 서브넷 ID 1을 사용하고 호스트 ID를 16진수 형식으로 지정합니다.

```
# ifconfig ip.6to4tun0 inet6 plumb
# ifconfig ip.6to4tun0 inet tsrc 129.146.86.187 \
2002:8192:56bb:1::8192:56bb/64 up

# ifconfig ip.6to4tun0 inet6
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6>mtu 1480 index 11
        inet tunnel src 129.146.86.187
        tunnel hop limit 60
        inet6 2002:8192:56bb:1::8192:56bb/64
```

예 11–10 ifconfig를 통해 6to4 터널 구성(짧은 형식)

이 예에서는 6to4 터널 구성을 위한 짧은 형식을 보여줍니다.

```
# ifconfig ip.6to4tun0 inet6 plumb
# ifconfig ip.6to4tun0 inet tsrc 129.146.86.187 up
```

예 11–10 ifconfig를 통해 6to4 터널 구성(짧은 형식) (계속)

```
# ifconfig ip.6to4tun0 inet6
ip.6to4tun0: flags=2200041<UP,RUNNING,NONUD,IPv6>mtu 1480 index 11
        inet tunnel src 129.146.86.187
        tunnel hop limit 60
        inet6 2002:8192:56bb::1/64
```

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_IP를 영구적으로 설정하면 netstat가 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 198 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 209 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 201 페이지 “ping 명령으로 원격 호스트 확인”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 205 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.

-d 디버깅을 설정합니다.

-D 특정 이벤트에 대한 디버깅을 설정합니다.

-f 기본 /etc/inet/ndpd.conf 파일 대신 구성 데이터를 읽도록 파일을 지정합니다.

-I 각 인터페이스에 대한 관련 정보를 출력합니다.

-n 라우터 알림을 루프백하지 않습니다.

-r 수신된 패킷을 무시합니다.

-v 다양한 유형의 진단 메시지를 보고하도록 상세 정보 표시 모드를 지정합니다.

-t 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 11–2은 이 파일에 나타날 수 있는 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 대체 포트 번호를 지정합니다.

-q — 경로 지정 정보를 표시하지 않습니다.

-s — 데몬이 라우터로 사용되지 않는 경우에도 경로 지정 정보를 표시합니다.

-P — Poison Reverse를 사용하지 못하도록 합니다.

-S — in.ripngd가 라우터로 사용되지 않는 경우 데몬은 각 라우터의 기본 경로만 통과합니다.

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 관리: 기본 관리**의 "SMF 명령줄 관리 유틸리티"를 참조하십시오.
- SMF를 사용하여 SCTP를 통해 실행되는 IPv4 서비스 매니페스트를 구성하는 예제 작업은 128 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. 주요 Neighbor Discovery 기능에 대한 개요는 76 페이지 “IPv6 Neighbor Discovery 프로토콜 개요”를 참조하십시오.

이 절에서는 Neighbor Discovery 프로토콜의 다음 기능에 대해 설명합니다.

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

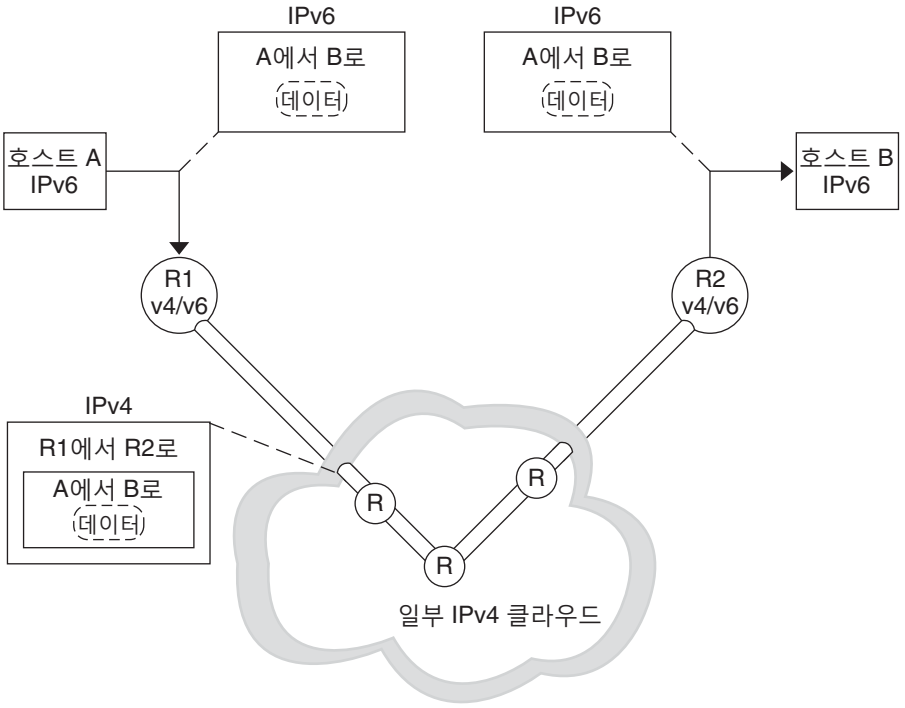
라우터 알림 메시지에는 호스트가 송신 패킷에 사용해야 하는 인터넷 매개변수(예: 홉 한계)가 포함되어 있습니다. 선택적으로 링크 매개변수(예: 링크 MTU)도 포함될 수 있습니다. 이 기능으로 중요한 매개변수를 중앙에서 관리할 수 있습니다. 매개변수는 라우터에 대해 설정될 수 있으며 연결된 모든 호스트에 자동으로 전파됩니다.

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.

IPv6 터널

이중 스택, IPv4/IPv6 사이트에서 종속성을 최소화하기 위해서는 두 IPv6 노드 사이의 경로에 있는 모든 라우터가 IPv6을 지원할 필요가 없습니다. 이러한 네트워크 구성을 지원하는 방식을 **터널링**이라고 부릅니다. 기본적으로 IPv6 패킷은 IPv4 패킷 내에 배치된 후 IPv4 라우터를 통해 경로 지정됩니다. 다음 그림은 IPv4 라우터를 경유하는 터널링 방식을 보여줍니다. IPv4 라우터는 그림에서 "R"로 표시되어 있습니다.

그림 11–5 IPv6 터널링 방식



Oracle Solaris IPv6 구현에는 두 가지 유형의 터널링 방식이 포함됩니다.

- 그림 11–5에서와 같이 두 라우터 사이에 구성된 터널
- 끝점 호스트에서 종료되는 자동 터널

구성된 터널은 MBONE의 IPv4 멀티캐스트 백본과 같이 현재 다른 목적으로 인터넷에서 사용됩니다. 운영적으로 터널은 두 개의 라우터로 구성되는데, 이 라우터는 IPv4 네트워크를 경유하여 두 라우터 간에 가상 지점 간 링크를 갖도록 구성되어 있습니다. 이러한 터널 종류는 예측 가능한 미래를 위해 인터넷의 일부 부분에 사용될 가능성이 높습니다.

자동 터널에는 IPv4와 호환되는 주소가 필요합니다. 자동 터널은 IPv6 라우터를 사용할 수 없을 때 IPv6 노드에 연결하는 데 사용할 수 있습니다. 이러한 터널은 자동 터널링 네트워크 인터페이스를 구성하여 이중 스택 호스트 또는 이중 스택 라우터에서 시작될 수 있습니다. 터널은 항상 이중 스택 호스트에서 종료됩니다. 이러한 터널은 IPv4 호환 가능 대상 주소에서 주소를 추출하여 터널의 끝점인 대상 IPv4 주소를 동적으로 확인합니다.

구성된 터널

터널링 인터페이스의 형식은 다음과 같습니다.

`ip.tun` *ppa*

*ppa*는 물리적 연결 지점입니다.

시스템 시작 시 터널링 모듈(tun)은 가상 인터페이스를 만들기 위해 IP 상위 `ifconfig` 명령으로 푸시됩니다. 푸시 작업은 적합한 `hostname6.*` 파일을 만들어서 수행됩니다.

예를 들어, IPv4 네트워크를 통해 IPv6 패킷을 캡슐화하기 위한 터널(IPv6 over IPv4)을 만들려면 다음 파일 이름을 만듭니다.

`/etc/hostname6.ip.tun0`

이 파일의 컨텐츠는 인터페이스가 연결된 후 `ifconfig`로 전달됩니다. 이 컨텐츠는 지점 간 터널을 구성하는 데 필요한 매개변수가 됩니다.

예 11–11 IPv6 Over IPv4 터널을 위한 `hostname6.ip.tun0` 파일

다음은 `hostname6.ip.tun0` 파일의 항목 예입니다.

```
tsrc 10.10.10.23 tdst 172.16.7.19 up
addif 2001:db8:3b4c:1:5678:5678::2 up
```

이 예에서 IPv4 소스 및 대상 주소는 IPv6 링크 로컬 주소를 자동 구성하기 위한 토큰으로 사용됩니다. 이러한 주소는 `ip.tun0` 인터페이스에 대한 소스 및 대상입니다. 두 개의 인터페이스가 구성됩니다. `ip.tun0` 인터페이스가 구성됩니다. 논리적 인터페이스인 `ip.tun0:1`도 구성됩니다. 논리적 인터페이스에는 `addif` 명령으로 지정된 소스 및 대상 IPv6 주소가 포함됩니다.

이러한 구성 파일의 컨텐츠는 시스템이 다중 사용자 모드로 시작될 때 변경 없이 `ifconfig`에 전달됩니다. 예 11–11의 항목은 다음에 해당합니다.

```
# ifconfig ip.tun0 inet6 plumb
# ifconfig ip.tun0 inet6 tsrc 10.0.0.23 tdst 172.16.7.19 up
# ifconfig ip.tun0 inet6 addif 2001:db8:3b4c:1:5678:5678::2 up
```

다음은 이 터널에 대한 `ifconfig -a`의 출력을 보여줍니다.

```
ip.tun0: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,
  NONUD,IPv6> mtu 1480 index 6
        inet tunnel src 10.0.0.23  tunnel dst 172.16.7.19
        inet6 fe80::c0a8:6417/10 --> fe80::c0a8:713
ip.tun0:1: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
  index 5
        inet6 2001:db8:3b4c:1:5678:5678::2
```

다음 구문을 사용해서 구성 파일에 라인을 추가해서 보다 논리적인 인터페이스를 구성할 수 있습니다.

```
addif IPv6-source IPv6-destination up
```

주 – 터널 끝이 터널을 통해 하나 이상의 접두어를 알리는 IPv6 라우터인 경우 터널 구성 파일에 `addif` 명령이 필요하지 않습니다. 다른 모든 주소는 자동 구성되므로 `tsrc` 및 `tdst`만 필요할 수 있습니다.

일부 경우에는 특정 터널에 대해 특정 소스 및 대상 링크 로컬 주소를 수동으로 구성해야 합니다. 이러한 링크 로컬 주소를 포함하도록 구성 파일의 첫번째 라인을 변경하십시오. 다음 라인은 예입니다.

```
tsrc 10.0.0.23 tdst 172.16.7.19 fe80::1/10 fe80::2 up
```

소스 링크 로컬 주소의 접두어 길이는 10입니다. 이 예에서 `ip.tun0` 인터페이스는 다음과 비슷합니다.

```
ip.tun0: flags=2200850<UP,POINTOPOINT,RUNNING,MULTICAST,NONUD,IPv6> mtu 1480
index 6
        inet tunnel src 10.0.0.23  tunnel dst 172.16.7.19
        inet6 fe80::1/10 --> fe80::2
```

IPv6 네트워크를 통해 IPv6 패킷을 캡슐화하기 위한 터널(IPv6 over IPv6)을 만들려면 다음 파일 이름을 만듭니다.

```
/etc/hostname6.ip6.tun0
```

예 11–12 IPv6 over IPv6 터널을 위한 `hostname6.ip6.tun0` 파일

다음은 IPv6 네트워크를 통한 IPv6 캡슐화를 위한 `hostname6.ip6.tun0` 파일의 항목 예입니다.

```
tsrc 2001:db8:3b4c:114:a00:20ff:fe72:668c
        tdst 2001:db8:15fa:25:a00:20ff:fe9b:a1c3
fe80::4 fe80::61 up
```

IPv6 네트워크를 통해 IPv4 패킷을 캡슐화하기 위한 터널(IPv4 over IPv6)을 만들려면 다음 파일 이름을 만듭니다.

```
/etc/hostname.ip6.tun0
```

예 11–13 IPv4 Over IPv6 터널을 위한 `hostname.ip6.tun0` 파일

다음은 IPv6 네트워크를 통한 IPv4 캡슐화를 위한 `hostname.ip6.tun0` 파일의 항목 예입니다.

예 11–13 IPv4 Over IPv6 터널을 위한 hostname.ip6.tun0 파일 *(계속)*

```
tsrc 2001:db8:3b4c:114:a00:20ff:fe72:668c
        tdst 2001:db8:15fa:25:a00:20ff:fe9b:a1c3
10.0.0.4 10.0.0.61 up
```

IPv4 네트워크를 통해 IPv4 패킷을 캡슐화하기 위한 터널(IPv4 over IPv4)을 만들려면 다음 파일 이름을 만듭니다.

```
/etc/hostname.ip.tun0
```

예 11–14 IPv4 Over IPv4 터널을 위한 hostname.ip.tun0

다음은 IPv4 네트워크를 통한 IPv4 캡슐화를 위한 hostname.ip.tun0 파일의 항목 예입니다.

```
tsrc 172.16.86.158 tdst 192.168.86.122
10.0.0.4 10.0.0.61 up
```

tun에 대한 자세한 내용은 tun(7M) 매뉴얼 페이지를 참조하십시오. IPv6으로 전환하는 중 터널링 개념에 대한 일반적인 설명을 보려면 78 페이지 “IPv6 터널 개요”를 참조하십시오. 터널 구성을 위한 절차에 대한 설명은 176 페이지 “IPv6 지원을 위한 터널 구성 작업(작업 맵)”을 참조하십시오.

6to4 자동 터널

Oracle Solaris에서는 주소 지정을 IPv4에서 IPv6으로 전환하는 데 선호하는 중간 방식으로 6to4 터널을 제공합니다. 6to4 터널은 격리된 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 고유 IPv6, 비6to4 사이트에 대한 터널에 참여할 수 있습니다.

이 절에서는 다음 6to4 항목에 대한 참조 자료를 제공합니다.

- 6to4 터널 토폴로지
- 6to4 주소 지정, 알림 형식 포함
- 6to4 터널을 경유하는 패킷에 대한 설명
- 6to4 라우터와 6to4 릴레이 라우터 간 터널의 토폴로지
- 6to4 릴레이 라우터 지원을 구성하기 전에 고려할 사항

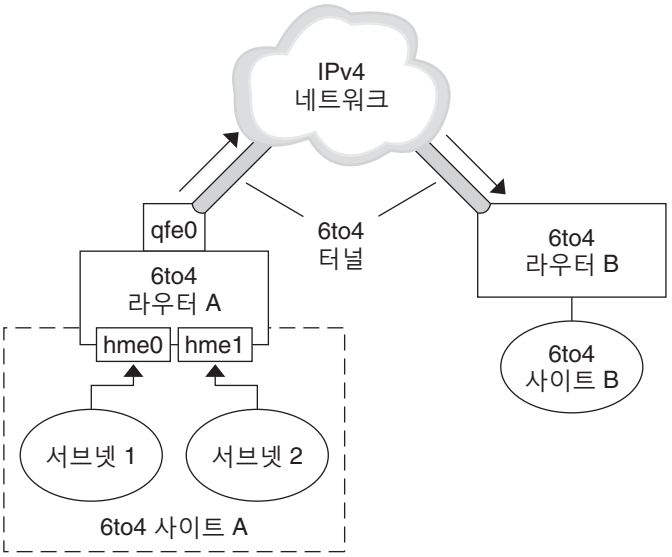
다음 표는 유용한 정보를 추가로 얻기 위해 6to4 터널 및 리소스를 구성하는 추가 작업에 대해 설명합니다.

작업 또는 세부 정보	정보
6to4 터널 구성 작업	179 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 고유 IPv6 인터넷을 비롯한 모든 IPv6 사이트에 대한 링크 역할도 합니다. 다음 그림은 6to4 터널이 6to4 사이트 간에 이러한 연결을 제공하는 방식을 보여줍니다.

그림 11–6 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 11–6에 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 고유 IPv6, 비6to4 네트워크와 통신해야 하는 6to4 라우터에서 터널 끝점으로 사용됩니다. 릴레이 라우터는 기본적으로 6to4 사이트와 고유 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 11–7 6to4 사이트와 6to4 릴레이 라우터 간 터널

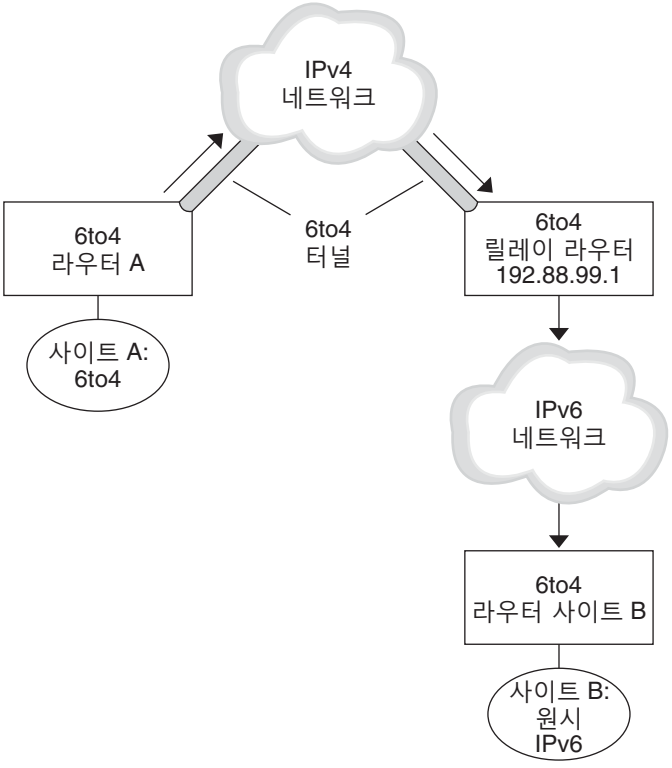


그림 11–7에서 6to4 사이트 A는 고유 IPv6 사이트 B에 있는 노드와 통신해야 합니다. 이 그림은 사이트 A에서 IPv4 네트워크를 경유하여 6to4 터널에 도달하는 트래픽 경로를 보여줍니다. 터널의 끝점은 6to4 라우터 A와 6to4 릴레이 라우터입니다. 6to4 릴레이 라우터를 넘어가면 IPv6 사이트 B가 연결되어 있는 IPv6 네트워크입니다.

6to4 사이트와 고유 IPv6 사이트 간 패킷 플로우

이 절에서는 6to4 사이트에서 고유 IPv6 사이트로의 패킷 플로우에 대해 설명합니다. 이 시나리오는 그림 11–7에 표시된 토폴로지를 사용합니다.

1. 6to4 사이트 A에 있는 호스트가 고유 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 `192.88.99.1`입니다. 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 고유 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.

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.

`nsswitch.conf` 파일의 변경 사항

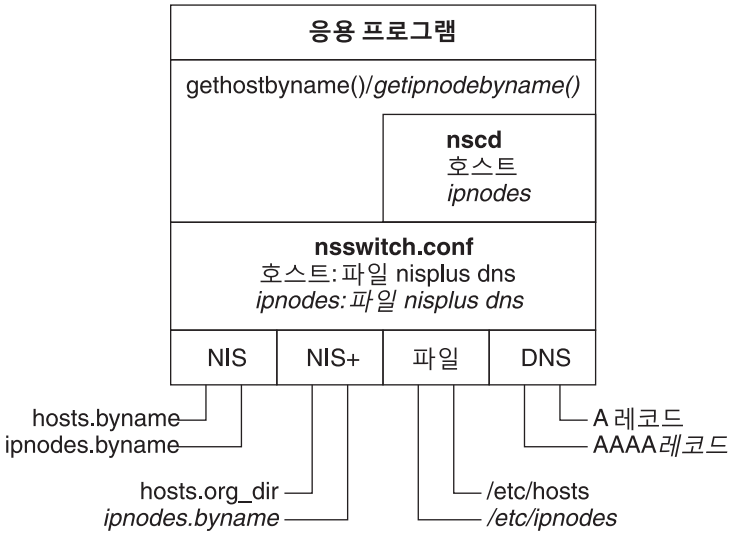
Solaris 10 11/06 및 이전 릴리스에서는 `/etc/inet/ipnodes`를 통해 IPv6 주소를 조회하는 기능 외에도 IPv6 지원이 NIS, LDAP 및 DNS 이름 서비스에 추가되었습니다. 따라서 `nsswitch.conf` 파일이 IPv6 조회를 지원하도록 수정되었습니다.

```
hosts:  files dns nisplus [NOTFOUND=return]
ipnodes: files dns nisplus [NOTFOUND=return]
```

주 – 다중 이름 서비스에서 `ipnodes`를 검색하기 위해 `/etc/nsswitch.conf` 파일을 변경하기 전에 이러한 `ipnodes` 데이터베이스에 IPv4 및 IPv6 주소를 채웁니다. 그렇지 않으면 불필요한 지연으로 인해 가능한 부트 시 지연을 포함한 호스트 주소 확인이 발생할 수 있습니다.

다음 다이어그램에서는 `gethostbyname` 및 `getipnodebyname` 명령을 사용하는 응용 프로그램에 대해 `nsswitch.conf` 파일과 새로운 이름 서비스 데이터베이스 사이의 새로운 관계를 보여줍니다. 기울임꼴로 표시된 항목이 새 항목입니다. `gethostbyname` 명령은 `/etc/inet/hosts`에 저장된 IPv4 주소만 검사합니다. Solaris 10 11/06 및 이전 릴리스에서 `getipnodebyname` 명령은 `nsswitch.conf` 파일의 `ipnodes` 항목에 지정된 데이터베이스를 조회합니다. 조회가 실패하면 명령이 `nsswitch.conf` 파일의 `hosts` 항목에 지정된 데이터베이스를 검사합니다.

그림 11–8 `nsswitch.conf` 및 이름 서비스 사이의 관계



이름 서비스에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**를 참조하십시오.

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.

제 3 부

DHCP

이 부분은 DHCP(Dynamic Host Configuration Protocol)에 대한 개념적 정보를 포함하고 DHCP 서비스를 계획, 구성, 관리하고 문제를 해결하기 위한 작업을 설명합니다.

◆ ◆ ◆ 12장

12 DHCP 정보(개요)

이 장에서는 DHCP(Dynamic Host Configuration Protocol)를 소개하고 프로토콜의 근간을 이루는 개념을 설명합니다. 또한 네트워크에서 DHCP 사용 시의 이점을 설명합니다.

이 장은 다음 정보를 포함합니다.

DHCP 프로토콜 정보

DHCP 프로토콜을 사용하여 TCP/IP 네트워크의 호스트 시스템을 부트할 때 네트워크에 대해 자동으로 구성할 수 있습니다. DHCP는 클라이언트-서버 방식을 사용합니다. 서버는 클라이언트에 대한 구성 정보를 저장 및 관리하고, 클라이언트 요청 시 해당 정보를 제공합니다. 이 정보에는 클라이언트의 IP 주소와 클라이언트에 사용 가능한 네트워크 서비스 정보가 포함됩니다.

DHCP는 이전 프로토콜인 BOOTP(TCP/IP 네트워크를 통해 부트하도록 설계)에서 발전한 것입니다. 클라이언트와 서버 간의 메시지에 대해 DHCP는 BOOTP와 동일한 형식을 사용합니다. 그러나 BOOTP 메시지와 달리, DHCP 메시지는 클라이언트에 대한 네트워크 구성 데이터를 포함할 수 있습니다.

DHCP의 주요 장점은 임대를 통해 IP 주소 지정을 관리할 수 있다는 것입니다. **임대**를 사용하면 IP 주소가 사용 중이 아닐 때 재생 이용할 수 있습니다. 재생 이용된 IP 주소는 다른 클라이언트에 재지정할 수 있습니다. DHCP를 사용하는 사이트는 모든 클라이언트에 영구 IP 주소를 지정했을 때 필요한 것보다 작은 IP 주소 풀을 사용할 수 있습니다.

DHCP 사용 시의 이점

DHCP는 시간이 오래 걸리는 TCP/IP 네트워크 설정 작업이나 일상적인 네트워크 관리 작업을 줄일 수 있습니다. Oracle Solaris 구현에서 DHCP는 IPv4에만 작동합니다.

DHCP는 다음과 같은 이점을 제공합니다.

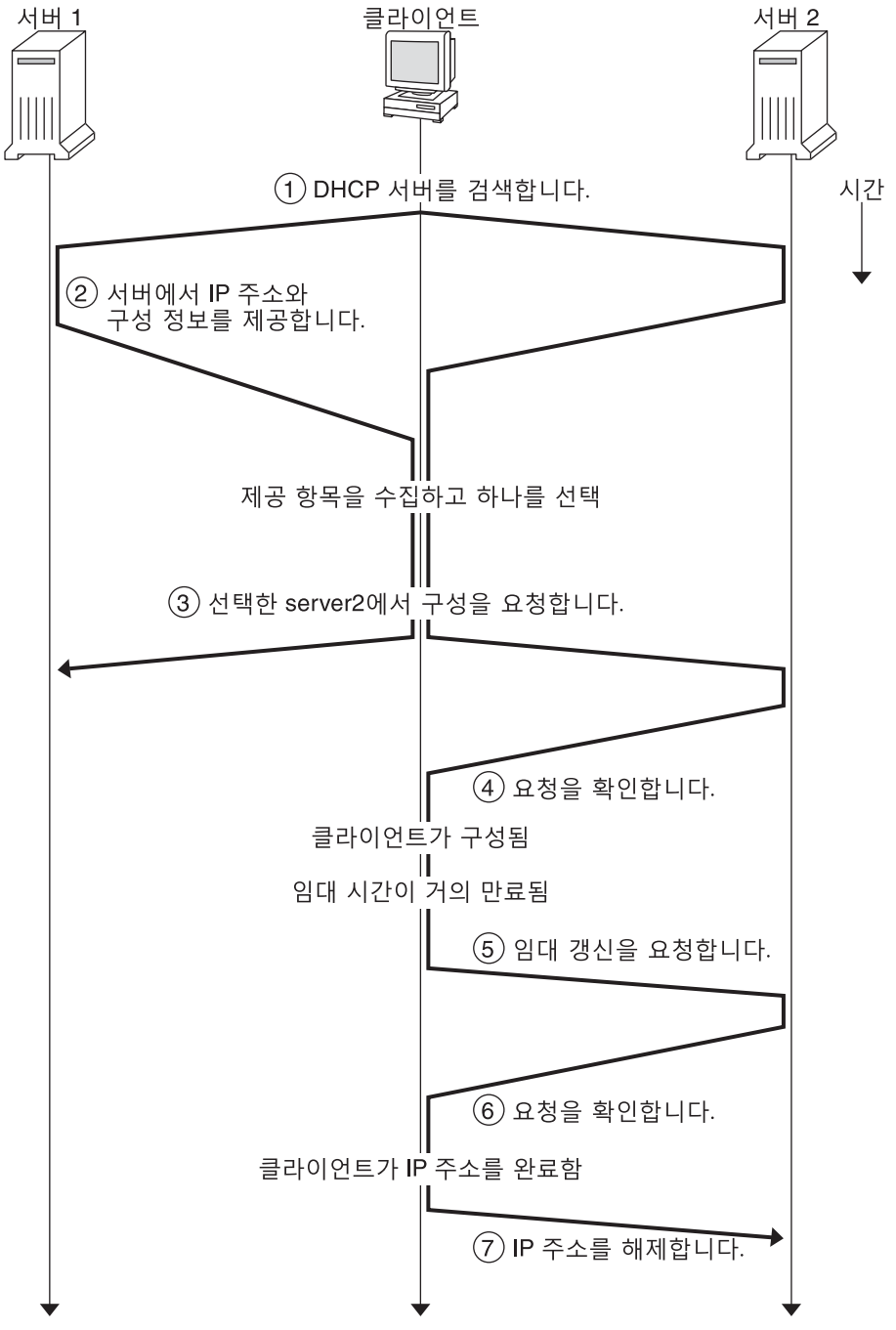
- **IP 주소 관리** – DHCP의 주요 이점은 간편한 IP 주소 관리입니다. DHCP가 없는 네트워크에서는 IP 주소를 수동으로 지정해야 합니다. 매우 신중하게 각 클라이언트에 고유한 IP 주소를 지정하고 각 클라이언트를 개별적으로 구성해야 합니다. 클라이언트가 다른 네트워크로 이동하면 해당 클라이언트를 수동으로 수정해야 합니다. DHCP가 사용으로 설정된 경우 관리자 개입 없이 DHCP 서버가 IP 주소를 관리하고 지정합니다. DHCP 서버로부터 새 네트워크에 적절한 새 클라이언트 정보를 얻으므로 수동 재구성 없이 다른 서브넷으로 클라이언트를 이동할 수 있습니다.
- **중앙화된 네트워크 클라이언트 구성** – 특정 클라이언트 또는 특정 클라이언트 유형에 대해 맞춤식 구성을 만들 수 있습니다. 구성 정보는 DHCP 데이터 저장소의 한 곳에 저장됩니다. 구성을 변경하기 위해 클라이언트에 로그인할 필요가 없습니다. 간단히 데이터 저장소의 정보를 변경하면 여러 클라이언트를 변경할 수 있습니다.
- **BOOTP 클라이언트 지원** – BOOTP 서버와 DHCP 서버는 모두 클라이언트에서 브로드캐스트를 수신하고 응답합니다. DHCP 서버는 DHCP 클라이언트는 물론 BOOTP 클라이언트의 요청에 응답할 수 있습니다. BOOTP 클라이언트는 IP 주소 및 서버에서 부트하는 데 필요한 정보를 수신합니다.
- **로컬 클라이언트 및 원격 클라이언트 지원** – BOOTP는 한 네트워크에서 다른 네트워크로 메시지 중계를 제공합니다. DHCP는 여러 가지 방법으로 BOOTP 중계 기능을 활용합니다. 대부분의 네트워크 라우터는 BOOTP 중계 에이전트로 작동하여 클라이언트 네트워크에 없는 서버로 BOOTP 요청을 전달하도록 구성할 수 있습니다. DHCP 요청은 BOOTP 요청과 구별하기 어렵기 때문에 DHCP 요청을 동일한 방법으로 라우터에 중계할 수 있습니다. 또한 BOOTP 중계를 지원하는 라우터를 사용할 수 없는 경우 DHCP 서버가 BOOTP 중계 에이전트로 작동하도록 구성할 수 있습니다.
- **네트워크 부트** – 클라이언트는 RARP(Reverse Address Resolution Protocol) 및 `bootparams` 파일을 사용하는 대신, DHCP를 사용하여 네트워크의 서버에서 부트하는 데 필요한 정보를 얻을 수 있습니다. DHCP 서버는 IP 주소, 부트 서버, 네트워크 구성 정보 등 클라이언트가 작동하는 데 필요한 모든 정보를 제공할 수 있습니다. DHCP 요청을 서브넷에서 중계할 수 있으므로 DHCP 네트워크 부트를 사용할 때 네트워크에서 훨씬 적은 부트 서버를 배치할 수 있습니다. RARP로 부트하려면 각 서브넷에 부트 서버가 필요합니다.
- **대형 네트워크 지원** – 수백만 개의 DHCP 클라이언트가 포함된 네트워크에서 DHCP를 사용할 수 있습니다. DHCP 서버는 멀티스레딩을 사용하여 많은 클라이언트 요청을 동시에 처리합니다. 또한 대량의 데이터 처리를 위해 최적화된 데이터 저장소를 지원합니다. 데이터 저장소 액세스는 별도의 프로세싱 모듈로 처리됩니다. 이 데이터 저장소 접근법을 통해 필요한 데이터베이스에 대한 지원을 추가할 수 있습니다.

DHCP의 작동 방식

먼저 DHCP 서버를 설치하고 구성해야 합니다. 구성 중 클라이언트가 네트워크에서 작동하는 데 필요한 네트워크 정보를 지정합니다. 이 정보가 갖춰진 후에 클라이언트가 네트워크 정보를 요청 및 수신할 수 있습니다.

다음 다이어그램에 DHCP 서비스의 이벤트 순서가 표시됩니다. 원 안의 숫자는 다이어그램에 이어진 설명에서 번호 매기기 항목에 해당합니다.

그림 12–1 DHCP 서비스의 이벤트 순서



앞의 다이어그램은 다음 단계를 보여줍니다.

1. 클라이언트가 로컬 서브넷의 제한된 브로드캐스트 주소(255.255.255.255)로 *Discover* **메시지**를 브로드캐스트하여 DHCP 서버를 검색합니다. 라우터가 존재하고 BOOTP 중계 에이전트로 작동하도록 구성된 경우 여러 서브넷의 다른 DHCP 서버로 요청이 전달됩니다. 클라이언트의 **브로드캐스트**에는 Oracle Solaris의 DHCP 구현에서 클라이언트의 MAC(Media Access Control) 주소로부터 파생된 고유한 ID가 포함됩니다. 이더넷 네트워크에서 MAC 주소는 이더넷 주소와 동일합니다.

   Discover 메시지를 받은 DHCP 서버는 다음 정보를 확인하여 클라이언트의 네트워크를 결정할 수 있습니다.

   - 어떤 네트워크 인터페이스에서 요청이 들어왔습니까? 서버는 클라이언트가 인터페이스로 연결된 네트워크에 있는지, 또는 클라이언트가 해당 네트워크에 연결된 BOOTP 중계 에이전트를 사용 중인지 확인합니다.
   - 요청에 BOOTP 중계 에이전트의 IP 주소가 들어 있습니까? 요청이 중계 에이전트를 통해 전달된 경우 요청 헤더에 중계 에이전트의 주소가 삽입됩니다. 서버가 **중계 에이전트 주소**를 감지한 경우 중계 에이전트가 클라이언트의 네트워크에 연결되어야 하므로 주소의 네트워크 부분이 클라이언트의 네트워크 주소를 나타냅니다.
   - 클라이언트의 네트워크가 서브넷으로 나뉩니까? 서버가 `netmasks` 테이블을 참조하여 중계 에이전트의 주소 또는 요청을 받은 네트워크 인터페이스의 주소가 가리키는 네트워크에서 사용된 서브넷 마스크를 찾습니다. 일단 서버가 사용된 서브넷 마스크를 알고 나면 네트워크 주소의 어떤 부분이 호스트 부분인지 결정하고, 클라이언트에 적절한 IP 주소를 선택할 수 있습니다. `netmasks`에 대한 자세한 내용은 `netmasks(4)` 매뉴얼 페이지를 참조하십시오.

2. DHCP 서버가 클라이언트의 네트워크를 결정한 후에 적절한 IP 주소를 선택하고 주소가 아직 사용 중이 아닌지 확인합니다. 그런 다음 DHCP 서버가 *Offer* **메시지**를 브로드캐스트하여 클라이언트에 응답합니다. Offer 메시지에는 선택된 IP 주소와 클라이언트에 구성할 수 있는 서비스 정보가 포함됩니다. 각 서버는 클라이언트가 IP 주소의 사용 여부를 결정할 때까지 제공된 IP 주소를 임시로 예약합니다.

3. 클라이언트가 제공된 서비스 개수와 유형을 기반으로 최상의 제안을 선택합니다. 클라이언트가 최상의 제안을 제출한 서버의 IP 주소를 가리키는 요청을 브로드캐스트합니다. 브로드캐스트는 모든 응답 DHCP 서버가 클라이언트가 서버를 선택했음을 알고 있다고 보장합니다. 선택되지 않은 서버는 제공받은 IP 주소의 예약을 취소할 수 있습니다.

4. 선택된 서버가 클라이언트에 대한 IP 주소를 할당하고 DHCP 데이터 저장소에 정보를 저장합니다. 또한 클라이언트에 확인 메시지(ACK)를 보냅니다. **확인 메시지**는 클라이언트에 대한 네트워크 구성 매개변수를 포함합니다. 클라이언트가 `ping` 유틸리티를 사용하여 다른 시스템에서 IP 주소를 사용 중이 아닌지 테스트합니다. 그런 다음 클라이언트가 부트를 계속하여 네트워크에 참여합니다.

5. 클라이언트가 임대 시간을 모니터합니다. 정해진 기간이 경과된 경우 클라이언트가 선택한 서버에 임대 시간을 늘리라는 새 메시지를 보냅니다.

6. 요청을 받은 DHCP 서버는 관리자가 설정한 로컬 임대 정책을 고수하는 경우 임대 시간을 연장합니다. 서버가 20초 안에 응답하지 않으면 클라이언트가 요청을 브로드캐스트하여 다른 DHCP 서버 중 하나가 임대를 연장할 수 있도록 합니다.
7. 클라이언트에 더 이상 IP 주소가 필요하지 않으면 IP 주소가 해제되었음을 서버에 알립니다. 이 통지는 정상적인 종료 중에 발생할 수 있으며 수동으로 실행할 수도 있습니다.

DHCP 서버

DHCP 서버는 호스트 시스템에서 Oracle Solaris의 데몬으로 실행됩니다. 이 서버에는 다음과 같은 두 가지 기본 기능이 있습니다.

- **IP 주소 관리** – DHCP 서버는 일정 범위의 IP 주소를 제어하고 이러한 주소를 영구적으로 또는 지정된 기간 동안 클라이언트에 할당합니다. 서버는 임대 방식을 사용하여 클라이언트가 영구적이 아닌 주소를 사용할 수 있는 기간을 결정합니다. 주소가 더 이상 사용되지 않으면 풀로 반환되어 재지정될 수 있습니다. 서버는 한 주소를 여러 클라이언트가 사용하지 않도록 하기 위해 해당 DHCP 네트워크 테이블에 클라이언트와 IP 주소의 바인딩에 대한 정보를 유지 관리합니다.
- **클라이언트에 대한 네트워크 구성 제공** – 서버는 IP 주소를 지정하고 호스트 이름, 브로드캐스트 주소, 네트워크 서브넷 마스크, 기본 게이트웨이, 이름 서비스를 비롯하여 네트워크 구성에 대한 기타 다양한 정보를 제공합니다. 네트워크 구성 정보는 서버의 `dhcptab` 데이터베이스에서 가져옵니다.

또한 DHCP 서버를 다음과 같은 추가 기능을 수행하도록 구성할 수 있습니다.

- **BOOTP 클라이언트 요청에 응답** – 서버는 BOOTP 서버를 검색하는 BOOTP 클라이언트의 브로드캐스트를 수신 대기하고 BOOTP 클라이언트에 IP 주소 및 부트 매개변수를 제공합니다. 정보는 관리자가 정적으로 구성했어야 합니다. DHCP 서버는 BOOTP 서버와 DHCP 서버 기능을 동시에 수행할 수 있습니다.
- **요청 중계** – 서버는 BOOTP 및 DHCP 요청을 다른 서브넷의 해당 서버에 중계합니다. 서버는 BOOTP 중계 에이전트로 구성된 경우 DHCP 또는 BOOTP 서비스를 제공할 수 없습니다.
- **DHCP 클라이언트에 대한 네트워크 부트 지원 제공** – 서버는 DHCP 클라이언트에 네트워크를 통해 부트하는 데 필요한 정보(IP 주소, 부트 매개변수 및 네트워크 구성 정보)를 제공할 수 있습니다. 서버는 DHCP 클라이언트가 WAN(Wide Area Network)을 통해 부트 및 설치하는 데 필요한 정보도 제공합니다.
- **호스트 이름을 제공하는 클라이언트에 대해 DNS 테이블 업데이트** – DHCP 서비스 요청에 `Hostname` 옵션 및 값을 제공하는 클라이언트에 대해 서버는 클라이언트를 대신하여 DNS 업데이트를 시도할 수 있습니다.

DHCP 서버 관리

DHCP 관리자 또는 287 페이지 “DHCP 명령줄 유틸리티”에 설명된 명령줄 유틸리티를 사용하여 수퍼 유저로 DHCP 서버를 시작, 중지 및 구성할 수 있습니다. 일반적으로 DHCP 서버는 시스템이 부트할 때 자동으로 시작되고 시스템이 종료될 때 중지되도록 구성됩니다. 일반적인 상황에서는 서버를 수동으로 시작하거나 중지할 필요가 없습니다.

DHCP 데이터 저장소

DHCP 서버가 사용하는 모든 데이터는 데이터 저장소에 유지 관리됩니다. 데이터 저장소는 일반 텍스트 파일, NIS+ 테이블 또는 이진 형식 파일로 구성될 수 있습니다. DHCP 서비스를 구성하는 동안 사용할 데이터 저장소 유형을 선택합니다. 298 페이지 “DHCP 데이터 저장소 선택” 절에서는 데이터 저장소 유형 간 차이점에 대해 설명합니다. DHCP 관리자 또는 `dhcpconfig` 명령을 사용하여 한 형식에서 다른 형식으로 데이터 저장소를 변환할 수 있습니다.

한 DHCP 서버의 데이터 저장소에서 다른 서버의 데이터 저장소로 데이터를 이동할 수도 있습니다. 서버가 서로 다른 데이터 저장소 형식을 사용하는 경우에도 데이터 저장소에서 작동하는 내보내기 및 가져오기 유틸리티를 사용할 수 있습니다. DHCP 관리자 또는 `dhcpconfig` 명령을 사용하여 데이터 저장소의 전체 내용을 내보내거나 가져올 수도 있고 일부 데이터만 내보내거나 가져올 수도 있습니다.

주 – 자체 코드 모듈을 개발하여 DHCP(서버 및 관리 도구)와 데이터베이스 간 인터페이스를 제공한다면 DHCP 데이터 저장소에 어떤 데이터베이스 또는 파일 형식도 사용할 수 있습니다. 자세한 내용은 **Solaris DHCP Service Developer’s Guide** 를 참조하십시오.

DHCP 데이터 저장소 내부에는 두 가지 유형의 테이블이 있습니다. DHCP 관리자 또는 명령줄 유틸리티를 사용하여 이러한 테이블의 내용을 보고 관리할 수 있습니다. 데이터 테이블은 다음과 같습니다.

- `dhcptab` **테이블** – 클라이언트에 전달할 수 있는 구성 정보 테이블입니다.
- **DHCP 네트워크 테이블** – 테이블 이름에 지정된 네트워크에 있는 DHCP 및 BOOTP 클라이언트에 대한 정보가 포함된 테이블입니다. 예를 들어, `192.168.32.0` 네트워크는 이름에 `192_168_32_0`이 포함된 테이블을 가질 수 있습니다.

dhcptab 테이블

`dhcptab` 테이블은 클라이언트가 DHCP 서버에서 얻을 수 있는 모든 정보를 포함합니다. DHCP 서버는 시작될 때마다 `dhcptab` 테이블을 검색합니다. `dhcptab` 테이블의 파일 이름은 사용된 데이터 저장소에 따라 다릅니다. 예를 들어, NIS+ 데이터 저장소 `SUNWnisplus`에 의해 만들어진 `dhcptab` 테이블은 `SUNWnisplus1_dhcptab`입니다.

DHCP 프로토콜은 클라이언트에 전달될 수 있는 여러 표준 정보 항목을 정의합니다. 이러한 항목은 매개변수, 기호 또는 옵션으로 불립니다. 옵션은 DHCP 프로토콜에 숫자 코드 및 텍스트 레이블로 정의되지만, 값은 포함되지 않습니다. 다음 표는 일반적으로 사용되는 표준 옵션 몇 가지를 보여줍니다.

표 12–1 DHCP 표준 옵션 예

코드	레이블	설명
1	Subnet	서브넷 마스크 IP 주소
3	Router	라우터의 IP 주소
6	DNSserv	DNS 서버의 IP 주소
12	Hostname	클라이언트 호스트 이름의 텍스트 문자열
15	DNSdmain	DNS 도메인 이름

일부 옵션은 서버 구성 중 정보를 제공하면 자동으로 값이 지정됩니다. 나중에 다른 옵션에 명시적으로 값을 지정할 수도 있습니다. 옵션 및 해당 값은 구성 정보를 제공하기 위해 클라이언트에 전달됩니다. 예를 들어, 옵션/값 쌍, `DNSdmain=Georgia.Peach.COM`은 클라이언트의 DNS 도메인 이름을 `Georgia.Peach.COM`으로 설정합니다.

옵션은 클라이언트에 더 쉽게 정보를 전달할 수 있도록 **매크로**라 불리는 컨테이너에 다른 옵션과 함께 그룹화될 수 있습니다. 일부 매크로는 서버 구성 중 자동으로 만들어지며 구성 중 값이 지정되는 옵션을 포함합니다. 매크로는 다른 매크로를 포함할 수도 있습니다.

`dhcptab` 테이블의 형식은 `dhcptab(4)` 매뉴얼 페이지에 설명되어 있습니다. DHCP 관리자에서 Options(옵션) 및 Macros(매크로) 탭에 표시되는 모든 정보는 `dhcptab` 테이블에서 가져옵니다. 옵션에 대한 자세한 내용은 290 페이지 “DHCP 옵션 정보”를 참조하십시오. 매크로에 대한 자세한 내용은 290 페이지 “DHCP 매크로 정보”를 참조하십시오.

`dhcptab` 테이블을 수동으로 편집해서는 안됩니다. `dhtadm` 명령 또는 DHCP 관리자를 사용하여 옵션 및 매크로를 만들거나 삭제 또는 수정해야 합니다.

DHCP 네트워크 테이블

DHCP 네트워크 테이블은 클라이언트 식별자를 IP 주소 및 각 주소와 연관된 구성 매개변수에 매핑합니다. 네트워크 테이블의 형식은 `dhcp_network(4)` 매뉴얼 페이지에 설명되어 있습니다. DHCP 관리자에서 Addresses(주소) 탭에 표시되는 모든 정보는 네트워크 테이블에서 가져옵니다.

DHCP 관리자

DHCP 관리자는 DHCP 서비스와 연관된 모든 관리 작업을 수행하는 데 사용할 수 있는 GUI(그래픽 사용자 인터페이스) 도구입니다. DHCP 관리자를 사용하여 서버 및 서버가 사용하는 데이터를 관리할 수 있습니다. 수퍼 유저로 DHCP 관리자를 실행해야 합니다.

다음과 같은 방식으로 서버에서 DHCP 관리자를 사용할 수 있습니다.

- DHCP 서버 구성 및 구성 해제
- DHCP 서버 시작, 중지 및 다시 시작
- DHCP 서비스 사용/사용 안함으로 설정
- DHCP 서버 설정 사용자 정의

DHCP 관리자에서는 다음과 같은 방법으로 IP 주소, 네트워크 구성 매크로 및 네트워크 구성 옵션을 관리할 수 있습니다.

- DHCP 관리를 받도록 네트워크 추가 및 삭제
- DHCP 관리를 받는 IP 주소 보기, 추가, 수정, 삭제 및 해제
- 네트워크 구성 매크로 보기, 추가, 수정 및 삭제
- 비표준 네트워크 구성 옵션 보기, 추가, 수정 및 삭제

DHCP 관리자에서는 다음과 같은 방법으로 DHCP 데이터 저장소를 관리할 수 있습니다.

- 새 데이터 저장소 형식으로 데이터를 변환합니다.
- 첫번째 서버에서 내보내고 두번째 서버에서 가져오는 방식으로 DHCP 데이터를 한 DHCP 서버에서 다른 DHCP 서버로 이동할 수 있습니다.

DHCP 관리자에는 이 도구를 사용하여 수행할 수 있는 절차에 대한 자세한 온라인 도움말이 포함되어 있습니다. 자세한 내용은 318 페이지 “DHCP 관리자 정보”를 참조하십시오.

DHCP 명령줄 유틸리티

모든 DHCP 관리 기능은 명령줄 유틸리티를 사용하여 수행할 수 있습니다. 수퍼 유저 또는 DHCP 관리 프로파일에 지정된 사용자로 로그인하면 유틸리티를 실행할 수 있습니다. 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

다음 표는 유틸리티를 나열하고 각 유틸리티의 용도에 대해 설명합니다.

표 12–2 DHCP 명령줄 유틸리티

명령	설명 및 용도	매뉴얼 페이지 링크
`in.dhcpd`	DHCP 서비스 데몬입니다. 명령줄 인수를 사용하여 몇 가지 런타임 옵션을 설정할 수 있습니다.	`in.dhcpd(1M)`

표 12–2 DHCP 명령줄 유틸리티 (계속)

명령	설명 및 용도	매뉴얼 페이지 링크
dhcpconfig	DHCP 서버를 구성하고 구성을 해제하는 데 사용됩니다. 이 유틸리티를 통해 DHCP 관리자의 여러 기능을 명령줄에서 수행할 수 있습니다. 이 유틸리티는 기본적으로 일부 구성 기능을 자동화하려는 사이트의 스크립트에서 사용하기 위한 것입니다. dhcpconfig는 서버 시스템의 네트워크 토폴로지 파일에서 정보를 수집하여 초기 구성을 위한 유용한 정보를 만듭니다.	dhcpconfig(1M)
dhtadm	DHCP 클라이언트에 대한 구성 옵션 및 매크로를 추가, 삭제 및 수정하는 데 사용됩니다. 이 유틸리티를 사용하면 dhcptab 테이블을 간접적으로 편집할 수 있으므로 dhcptab 테이블의 올바른 형식이 보장됩니다. dhcptab 테이블을 직접 편집하지 않아야 합니다.	dhtadm(1M)
pntadm	DHCP 네트워크 테이블을 관리하는 데 사용됩니다. 이 유틸리티를 사용하여 다음 작업을 수행할 수 있습니다. ■ DHCP 관리를 받도록 IP 주소 및 네트워크 추가 및 제거 ■ 지정된 IP 주소에 대해 네트워크 구성 수정 ■ DHCP 관리를 받는 IP 주소 및 네트워크에 대한 정보 표시	pntadm(1M)

DHCP 명령에 대한 역할 기반 액세스 제어

dhcpconfig, dhtadm 및 pntadm 명령에 대한 보안은 RBAC(역할 기반 액세스 제어) 설정에 의해 결정됩니다. 기본적으로 수퍼 유저만 명령을 실행할 수 있습니다. 다른 사용자 이름으로 명령을 사용하려면 321 페이지 “DHCP 명령에 사용자 액세스 설정”에 설명된 대로 DHCP 관리 프로파일에 사용자 이름을 지정해야 합니다.

DHCP 서버 구성

DHCP 서버를 실행할 시스템에서 DHCP 관리자를 처음 실행할 때 DHCP 서버를 구성합니다.

DHCP 관리자 서버 구성 대화 상자에서는 하나의 네트워크에서 DHCP 서버를 사용으로 설정하고 실행하는 데 필요한 필수 정보를 입력하라는 메시지를 표시합니다. 일부 기본값은 기존 시스템 파일에서 가져옵니다. 네트워크에 대해 시스템을 구성하지 않은 경우에는 기본값이 없습니다. DHCP 관리자는 다음 정보를 묻는 메시지를 표시합니다.

- 서버 역할(DHCP 서버 또는 BOOTP 중계 에이전트)
- 데이터 저장소 유형(파일, 이진 파일, NIS+ 또는 사이트 특정 유형)
- 선택한 데이터 저장소 유형에 대한 데이터 저장소 구성 매개변수
- 호스트 레코드 업데이트에 사용할 이름 서비스(있는 경우)(/etc/hosts, NIS+ 또는 DNS)

- 임대 기간 및 클라이언트가 임대를 갱신할 수 있는지 여부
- DNS 서버의 DNS 도메인 이름 및 IP 주소
- DHCP 서비스에 대해 구성하려는 첫번째 네트워크의 네트워크 주소 및 서브넷 마스크
- 네트워크 유형(LAN(Local Area Network) 또는 지점 간 네트워크)
- 라우터 검색 또는 특정 라우터의 IP 주소
- NIS 서버의 NIS 도메인 이름 및 IP 주소
- NIS+ 서버의 NIS+ 도메인 이름 및 IP 주소

`dhcpconfig` 명령을 사용하여 DHCP 서버를 구성할 수도 있습니다. 이 유틸리티는 기존 시스템 파일에서 자동으로 정보를 수집하여 유용한 초기 구성을 제공합니다. 따라서 `dhcpconfig`를 실행하기 전에 해당 파일이 올바른지 확인해야 합니다. `dhcpconfig`가 정보를 가져오기 위해 사용하는 파일에 대한 자세한 내용은 `dhcpconfig(1M)` 매뉴얼 페이지를 참조하십시오.

IP 주소 할당

DHCP 서버는 다음과 같은 유형의 IP 주소 할당을 지원합니다.

- **수동 할당** – 서버는 특정 DHCP 클라이언트에 대해 사용자가 선택한 특정 IP 주소를 제공합니다. 이 주소를 재생 이용하거나 다른 클라이언트에 지정할 수 없습니다.
- **자동 또는 영구 할당** – 서버가 만료 시간이 없는 IP 주소를 제공하므로 사용자가 지정을 변경하거나 클라이언트가 주소를 해제할 때까지 IP 주소가 영구적으로 클라이언트에 연결됩니다.
- **동적 할당** – 서버가 특정 기간에 대한 임대가 포함된 IP 주소를 요청 클라이언트에 제공합니다. 임대가 만료되면 서버는 주소를 회수하여 다른 클라이언트에 지정할 수 있습니다. 기간은 서버에 구성된 임대 시간에 의해 결정됩니다.

네트워크 구성 정보

DHCP 클라이언트에 제공할 정보를 결정합니다. DHCP 서버를 구성할 때 네트워크에 대한 필수 정보를 제공합니다. 나중에 클라이언트에 제공할 정보를 추가할 수 있습니다.

DHCP 서버는 네트워크 구성 정보를 옵션/값 쌍 및 매크로의 형태로 `dhcptab` 테이블에 저장합니다. 옵션은 클라이언트에 제공할 네트워크 데이터에 대한 키워드입니다. 값은 옵션에 지정되어 DHCP 메시지를 통해 클라이언트에 전달됩니다. 예를 들어, NIS 서버 주소는 `NISservs`라는 옵션을 통해 전달됩니다. `NISservs` 옵션은 DHCP 서버가 지정하는 IP 주소 목록과 동일한 값을 가집니다. 매크로는 클라이언트에 제공할 여러 옵션을 편리하게 그룹화하는 방법을 제공합니다. DHCP 관리자를 사용하여 옵션을 그룹화하는 매크로를 만들고 옵션에 값을 지정할 수 있습니다. 명령줄 도구를 선호하는 경우 DHCP 구성 테이블 관리 유틸리티인 `dhtadm`을 사용하여 옵션 및 매크로 작업을 수행할 수 있습니다.

DHCP 옵션 정보

DHCP에서 **옵션**은 클라이언트에 전달될 네트워크 정보 조각입니다. DHCP 관련 문맥에서는 옵션을 **기호** 또는 **태그**라고도 합니다. 옵션은 숫자 코드와 텍스트 레이블로 정의됩니다. 옵션은 DHCP 서비스에서 사용될 때 값을 받습니다.

DHCP 프로토콜은 일반적으로 지정되는 네트워크 데이터에 대한 다수의 표준 옵션을 정의합니다. 여기에는 `Subnet`, `Router`, `Broadcst`, `NIS+dom`, `Hostname` 및 `LeaseTim` 등이 해당됩니다. 표준 옵션 전체 목록은 `dhcp_inittab(4)` 매뉴얼 페이지를 참조하십시오. 표준 옵션 키워드는 어떤 방법으로도 수정할 수 없습니다. 하지만 옵션을 매크로에 포함할 때 사용자 네트워크 관련 옵션에 값을 지정할 수 있습니다.

표준 옵션으로 표현되지 않는 데이터에 대해 새 옵션을 만들 수 있습니다. 새로 만드는 옵션은 다음 세 가지 범주 중 하나로 분류되어야 합니다.

- **Extended(확장)** – 표준 DHCP 옵션이 되었지만 아직 DHCP 서버 구현에 포함되지 않은 옵션용입니다. 사용하려는 표준 옵션을 알고 있지만 DHCP 서버를 업그레이드하고 싶지 않은 경우 확장 옵션을 사용할 수 있습니다.
- **Site(사이트)** – 사이트에 고유한 옵션용입니다. 이러한 옵션을 만듭니다.
- **Vendor(공급업체)** – 하드웨어, 공급업체 플랫폼과 같이 특정 클래스의 클라이언트에만 적용되는 옵션용입니다. DHCP 구현에는 Oracle Solaris 클라이언트에 대한 여러 공급업체 옵션이 포함되어 있습니다. 예를 들어, `SrootIP4` 옵션은 네트워크에서 부트하는 클라이언트가 해당 루트(/) 파일 시스템에 사용해야 하는 서버의 IP 주소를 지정하는 데 사용됩니다.

DHCP 옵션을 만들고, 수정 및 삭제하는 절차는 15 장, “DHCP 관리(작업)”를 참조하십시오.

DHCP 매크로 정보

DHCP 서비스에서 **매크로**는 네트워크 구성 옵션 및 여기에 지정된 값의 모음입니다. 매크로는 특정 클라이언트 또는 클라이언트 유형에 전달할 옵션을 그룹화하기 위해 만들어집니다. 예를 들어, 특정 서브넷의 모든 클라이언트를 위한 매크로는 서브넷 마스크, 라우터 IP 주소, 브로드캐스트 주소, NIS+ 도메인 및 임대 시간에 대한 옵션/값 쌍을 포함할 수 있습니다.

DHCP 서버의 매크로 처리

DHCP 서버는 매크로를 처리할 때 매크로에 정의된 네트워크 옵션 및 값을 클라이언트에 전송되는 DHCP 메시지에 포함합니다. 서버는 특정 유형의 클라이언트에 대해 일부 매크로를 자동으로 처리합니다.

서버가 매크로를 자동으로 처리하려면 매크로 이름이 다음 표의 범주 중 하나를 따라야 합니다.

표 12–3 자동 처리를 위한 DHCP 매크로 범주

매크로 범주	설명
클라이언트 클래스	매크로 이름이 클라이언트 컴퓨터 유형, 운영 체제 또는 이 둘 모두로 표시되는 클라이언트의 클래스와 일치합니다. 예를 들어, 서버에 `SUNW.Sun-Blade-100`이라는 매크로가 있으면 하드웨어 구현이 `SUNW,Sun-Blade-100`인 모든 클라이언트는 자동으로 `SUNW.Sun-Blade-100` 매크로의 값을 받습니다.
네트워크 주소	매크로 이름이 DHCP 관리 네트워크 IP 주소와 일치합니다. 예를 들어, 서버에 `10.53.224.0`이라는 이름의 매크로가 있는 경우 `10.53.224.0` 네트워크에 연결된 모든 클라이언트는 자동으로 `10.53.224.0` 매크로의 값을 받습니다.
클라이언트 ID	매크로 이름이 클라이언트에 대한 고유한 식별자와 일치합니다. 이 식별자는 대개 이더넷 또는 MAC 주소에서 파생됩니다. 예를 들어, 서버에 `08002011DF32`라는 이름의 매크로가 있으면 클라이언트 ID가 `08002011DF32`(이더넷 주소 `8:0:20:11:DF:32`에서 파생됨)인 클라이언트는 자동으로 `08002011DF32`라는 이름의 매크로에 있는 값을 받습니다.

표 12–3에 나열된 범주 중 하나를 사용하지 않는 이름을 가진 매크로는 다음 중 하나가 성립되는 경우에만 처리될 수 있습니다.

- 매크로가 IP 주소에 매핑됩니다.
- 매크로가 자동으로 처리되는 다른 매크로에 포함되어 있습니다.
- 매크로가 IP 주소에 매핑되는 다른 매크로에 포함되어 있습니다.

주 – 서버를 구성할 때 서버 이름과 일치하는 이름의 매크로가 기본적으로 만들어집니다. 이 서버 매크로는 자동 처리되는 이름 유형 중 하나로 이름이 지정되지 않았기 때문에 모든 클라이언트에 대해 자동으로 처리되지 **않습니다**. 나중에 서버에서 IP 주소를 만드는 경우 이 IP 주소는 기본적으로 서버 매크로를 사용하도록 매핑됩니다.

매크로 처리 순서

DHCP 클라이언트가 DHCP 서비스를 요청하면 DHCP 서버는 클라이언트와 일치하는 매크로를 결정합니다. 서버는 매크로 범주를 통해 처리 순서를 결정하여 매크로를 처리합니다. 가장 일반적인 범주가 제일 먼저 처리되고 가장 구체적인 범주가 마지막으로 처리됩니다. 매크로는 다음 순서로 처리됩니다.

1. 클라이언트 클래스 매크로 - 가장 일반적인 범주
2. 네트워크 주소 매크로 - 클라이언트 클래스보다 구체적임
3. IP 주소에 매핑된 매크로 - 네트워크 주소보다 구체적임
4. 클라이언트 ID 매크로 - 가장 구체적인 범주. 한 클라이언트에만 적용됨

다른 매크로에 포함된 매크로는 포함하는 매크로의 일부로 처리됩니다.

한 옵션이 여러 매크로에 포함되어 있는 경우 마지막으로 처리되는 가장 구체적인 범주의 매크로에 있는 옵션 값이 사용됩니다. 예를 들어, 네트워크 주소 매크로에 값이

24시간인 임대 시간 옵션이 포함되어 있고 클라이언트 ID 매크로에 값이 8시간인 임대 시간 옵션이 포함되어 있는 경우 클라이언트는 8시간의 임대 시간을 받습니다.

DHCP 매크로의 크기 제한

매크로의 모든 옵션에 지정되는 값의 합계는 옵션 코드와 길이 정보를 포함하여 255바이트를 초과하지 않아야 합니다. 이 제한은 DHCP 프로토콜에 의해 강제됩니다.

이 제한의 영향을 받을 가능성이 높은 매크로는 Oracle Solaris 설치 서버의 파일 경로를 전달하는 데 사용되는 매크로입니다. 일반적으로 필요한 공급업체에 대한 최소한의 정보를 전달해야 합니다. 경로 이름이 필요한 옵션에서는 짧은 경로 이름을 사용해야 합니다. 긴 경로에 대한 심볼릭 링크를 만드는 경우에는 더 짧은 링크 이름을 전달할 수 있습니다.

DHCP 클라이언트

"클라이언트"라는 용어는 때때로 네트워크에서 클라이언트 역할을 수행하는 물리적 시스템을 지칭합니다. 그러나 이 문서에 설명된 DHCP 클라이언트는 소프트웨어 엔티티입니다. DHCP 클라이언트는 시스템의 Oracle Solaris에서 실행되는 데몬(`dhcpagent`)으로, DHCP 서버에서 네트워크 구성을 수신하도록 구성됩니다. 다른 공급업체의 DHCP 클라이언트도 DHCP 서버의 서비스를 사용할 수 있습니다. 하지만 이 문서에서는 DHCP 클라이언트에 대해서만 설명합니다.

DHCP 클라이언트에 대한 자세한 내용은 16 장, "DHCP 클라이언트 구성 및 관리"를 참조하십시오.

◆ ◆ ◆ 13 장

13 DHCP 서비스 계획(작업)

지금 만들고 있는 네트워크에서 또는 기존에 있는 네트워크에서 DHCP 서비스를 사용할 수 있습니다. 네트워크를 설정하는 경우 DHCP 서비스를 설정하기 전에 2 장, “TCP/IP 네트워크 계획(작업)”을 참조하십시오. 네트워크가 이미 존재하는 경우 이 장에서 작업을 계속하십시오.

이 장에서는 네트워크에 DHCP 서비스를 설정하기 전에 해야 할 일을 설명합니다. DHCP 서비스를 설정하기 위해 명령줄 유틸리티 `dhcpconfig`를 사용할 수도 있지만, 이 정보는 DHCP 관리자와 함께 사용하도록 제공되었습니다.

이 장은 다음 정보를 포함합니다.

DHCP 서비스용 네트워크 준비(작업 맵)

DHCP를 사용하도록 네트워크를 설정하기 전에, 하나 이상의 서버 구성을 위한 결정 사항에 도움이 되는 정보를 수집해야 합니다. 다음 표의 작업 맵을 사용하여 DHCP용 네트워크 준비를 위한 작업을 확인합니다. 이 표는 작업, 각 작업이 완수하는 내용, 그리고 개별 작업의 수행 단계를 기술하는 절을 설명합니다.

작업	설명	수행 방법
네트워크 토폴로지를 매핑합니다.	네트워크에 사용 가능한 서비스를 결정하고 찾습니다.	294 페이지 “네트워크 토폴로지 매핑”

작업	설명	수행 방법
필요한 DHCP 서버 수를 결정합니다.	필요한 DHCP 서버 수를 결정하기 위한 기초로, 예상된 DHCP 클라이언트 수를 사용합니다.	295 페이지 “DHCP 서버 수 결정”
시스템 파일 및 `netmasks` 테이블을 업데이트합니다.	네트워크 토폴로지를 정확히 반영합니다.	296 페이지 “시스템 파일 및 넷마스크 테이블 업데이트”

네트워크 토폴로지 매핑

아직 수행하지 않은 경우 네트워크의 물리적 구조를 매핑해야 합니다. 라우터 및 클라이언트의 위치, 그리고 네트워크 서비스를 제공하는 서버의 위치를 나타냅니다. 이 네트워크 토폴로지 맵을 바탕으로 DHCP 서비스에 사용할 서버를 결정할 수 있습니다. 또한 DHCP 서버가 클라이언트에 제공하는 구성 정보를 결정하는 데 도움을 줄 수 있습니다.

네트워크 계획에 대한 자세한 내용은 2 장, “TCP/IP 네트워크 계획(작업)”을 참조하십시오.

DHCP 구성 프로세스가 서버의 시스템 및 네트워크 파일에서 네트워크 정보를 수집할 수 있습니다. 296 페이지 “시스템 파일 및 넷마스크 테이블 업데이트”에서 이러한 파일을 설명합니다. 그러나 다른 서비스 정보를 클라이언트에 제공하고 싶은 경우 서버의 매크로에 정보를 입력해야 합니다. 네트워크 토폴로지를 조사하면서 클라이언트가 인지할 서버의 IP 주소를 기록합니다. 예를 들어, 다음 서버가 네트워크에 서비스를 제공할 수 있습니다. DHCP 구성은 이러한 서버를 검색하지 않습니다.

- 시간 서버
- 로그 서버
- 인쇄 서버
- 설치 서버
- 부트 서버
- 웹 프록시 서버
- 스왑 서버
- X Window 글꼴 서버
- TFTP(Trivial File Transfer Protocol) 서버

회피할 네트워크 토폴로지

일부 IP 네트워크 환경에서 여러 LAN(Local Area Network)이 동일한 네트워크 하드웨어 매체를 공유합니다. 네트워크가 여러 네트워크 하드웨어 인터페이스나 여러 논리적 인터페이스를 사용할 수 있습니다. DHCP는 이러한 종류의 공유 매체 네트워크에서 잘 작동하지 않습니다. 여러 LAN이 동일한 물리적 네트워크에서 실행되는 경우 DHCP 클라이언트의 요청이 모든 네트워크 하드웨어 인터페이스에 도착합니다. 그 결과 클라이언트가 모든 IP 네트워크에 동시에 연결된 것처럼 보입니다.

DHCP가 클라이언트에 적절한 IP 주소를 지정하려면 클라이언트의 네트워크 주소를 결정할 수 있어야 합니다. 여러 개의 네트워크가 하드웨어 매체에 존재하는 경우 서버가 클라이언트의 네트워크를 결정할 수 없습니다. 서버가 네트워크 번호를 모르면 IP 주소를 지정할 수 없습니다.

단 하나의 네트워크에만 DHCP를 사용할 수 있습니다. 한 네트워크가 사용자의 DHCP 요구에 맞지 않으면 네트워크를 재구성해야 합니다. 다음 제안을 고려해야 합니다.

- 서브넷에 가변 길이 서브넷 마스크(VLSM)를 사용하여 IP 주소 공간을 효율적으로 활용합니다. 동일한 물리적 네트워크에서 여러 네트워크를 실행할 필요는 없습니다. 가변 길이 서브넷 구현에 대한 내용은 `netmasks(4)` 매뉴얼 페이지를 참조하십시오. CIDR(Classless Inter-Domain Routing) 및 VLSM에 대한 자세한 내용은 `http://www.ietf.org/rfc/rfc1519.txt`를 참조하십시오.
- 장치가 서로 다른 물리적 LAN에 지정되도록 스위치에 포트를 구성합니다. 이 기법은 DHCP에 필요한 매핑을 하나의 LAN 대 하나의 IP 네트워크로 유지합니다. 포트 구성에 대한 내용은 스위치의 설명서를 참조하십시오.

DHCP 서버 수 결정

선택한 데이터 저장소 옵션은 DHCP 클라이언트 지원을 위해 보유해야 할 서버 수에 직접적인 영향을 미칩니다. 다음 표는 각 데이터 저장소에 대해 하나의 DHCP 서버가 지원할 수 있는 DHCP 및 BOOTP 클라이언트의 최대 개수를 보여줍니다.

표 13–1 하나의 DHCP 서버에서 지원되는 최대 클라이언트 수 예상치

데이터 저장소 유형	지원되는 최대 클라이언트 수
텍스트 파일	10,000
NIS+	40,000
이진 파일	100,000

이 최대 개수는 일반적인 지침으로 절대적인 수치가 아닙니다. DHCP 서버의 클라이언트 용량은 서버가 처리할 초당 트랜잭션 수에 크게 좌우됩니다. 임대 시간 및 사용 패턴은 트랜잭션 비율에 큰 영향을 미칩니다. 예를 들어, 임대 시간이 12시간으로 설정되고 사용자가 야간에 시스템을 끈다고 가정해 보십시오. 많은 사용자가 아침에 동시에 시스템을 켜면 많은 클라이언트가 동시에 임대를 요청하므로 서버가 트랜잭션 피크를 처리해야 합니다. DHCP 서버는 이러한 환경에서 보다 적은 클라이언트를 지원할 수 있습니다. DHCP 서버는 임대 시간이 긴 환경에서 또는 케이블 모뎀과 같이 끊임없이 연결된 장치로 구성된 환경에서 보다 많은 클라이언트를 지원할 수 있습니다.

298 페이지 “DHCP 데이터 저장소 선택” 절에서 데이터 저장소의 유형을 비교합니다.

시스템 파일 및 넷마스크 테이블 업데이트

DHCP 구성 중 DHCP 도구는 서버 구성에 사용할 수 있는 정보를 찾기 위해 서버에서 다양한 시스템 파일을 스캔합니다.

서버 구성을 위해 DHCP 관리자 또는 `dhcpconfig`를 실행하기 전에 시스템 파일의 정보가 최신인지 확인해야 합니다. 서버를 구성한 후에 오류를 발견하면 DHCP 관리자 또는 `dhtadm`을 사용하여 서버에서 매크로를 수정하십시오.

다음 표는 DHCP 서버 구성 중 수집된 정보 및 해당 정보의 소스를 나열합니다. DHCP를 서버에 구성하기 전에 이 정보가 서버에 올바르게 설정되어 있는지 확인하십시오. 서버를 구성한 후에 시스템 파일을 변경할 경우 이러한 변경 사항이 반영되도록 서비스를 재구성해야 합니다.

표 13–2 DHCP 구성에 사용된 정보

정보	소스	설명
표준 시간대	시스템 날짜, 시간대 설정	날짜 및 시간대는 Oracle Solaris 설치 중 초기에 설정됩니다. `date` 명령을 사용하여 날짜를 변경할 수 있습니다. `TZ` 환경 변수가 설정되도록 `svc:/system/environment:init` SMF 서비스의 `timezone/localtime` 등록 정보를 변경하여 시간대를 변경할 수 있습니다. 자세한 내용은 `TIMEZONE(4)` 매뉴얼 페이지를 참조하십시오.
DNS 매개변수	`/etc/resolv.conf`	DHCP 서버는 `/etc/resolv.conf` 파일을 사용하여 DNS 도메인 이름 및 DNS 서버 주소와 같은 DNS 매개변수를 얻습니다. `resolv.conf`에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)** 또는 `resolv.conf(4)` 매뉴얼 페이지를 참조하십시오.
NIS 또는 NIS+ 매개변수	시스템 도메인 이름, `nsswitch.conf`, NIS 또는 NIS+	DHCP 서버는 `domainname` 명령을 사용하여 서버 시스템의 도메인 이름을 얻습니다. `nsswitch.conf` 파일은 도메인 기반 정보를 찾을 위치를 서버에 알려줍니다. 서버 시스템이 NIS 또는 NIS+ 클라이언트인 경우 DHCP 서버는 NIS 또는 NIS+ 서버 IP 주소를 가져올 질의를 수행합니다. 자세한 내용은 `nsswitch.conf(4)` 매뉴얼 페이지를 참조하십시오.

표 13–2 DHCP 구성에 사용된 정보 *(계속)*

정보	소스	설명
기본 라우터	시스템 경로 지정 테이블, 사용자 프롬프트	DHCP 서버는 네트워크 경로 지정 테이블을 검색하여 로컬 네트워크에 연결된 클라이언트의 기본 라우터를 찾습니다. 동일한 네트워크에 없는 클라이언트의 경우 DHCP 서버가 정보를 묻는 프롬프트를 표시해야 합니다.
서브넷 마스크	네트워크 인터페이스, `netmasks` 테이블	DHCP 서버는 고유의 네트워크 인터페이스를 찾아서 로컬 클라이언트에 대한 넷마스크 및 브로드캐스트 주소를 결정합니다. 중계 에이전트에 의해 요청이 전달된 경우 서버가 중계 에이전트 네트워크의 `netmasks` 테이블에서 서브넷 마스크를 얻습니다.
브로드캐스트 주소	네트워크 인터페이스, `netmasks` 테이블	로컬 네트워크의 경우 DHCP 서버가 네트워크 인터페이스를 질의하여 브로드캐스트 주소를 얻습니다. 원격 네트워크의 경우 서버가 BOOTP 중계 에이전트의 IP 주소와 원격 네트워크의 넷마스크를 사용하여 네트워크의 브로드캐스트 주소를 계산합니다.

DHCP 서버 구성을 위한 결정 사항(작업 맵)

이 절에서는 네트워크에 첫번째 DHCP 서버를 구성하기 전에 결정할 사항을 설명합니다. 다음 표는 DHCP를 사용하도록 네트워크를 구성하는 데 필요한 결정 사항을 안내하고, 각 작업의 수행 단계를 설명하는 절로 링크합니다.

작업	설명	수행 방법
DHCP용 서버를 선택합니다.	서버가 DHCP 서비스 실행을 위한 시스템 요구 사항을 충족하는지 확인합니다.	298 페이지 “DHCP 서비스를 실행할 호스트 선택”
데이터 저장소를 선택합니다.	데이터 저장소 유형을 비교하여 사이트에 가장 적합한 데이터 저장소를 결정합니다.	298 페이지 “DHCP 데이터 저장소 선택”
임대 정책을 설정합니다.	사이트에 적절한 임대 정책을 결정하기 위해 IP 주소 임대를 알아봅니다.	299 페이지 “임대 정책 설정”

작업	설명	수행 방법
라우터 주소 또는 라우터 검색을 선택합니다.	DHCP 클라이언트가 라우터 검색 또는 특정 라우터를 사용할지 여부를 결정합니다.	300 페이지 “DHCP 클라이언트에 대한 라우터 결정”

DHCP 서비스를 실행할 호스트 선택

네트워크 토폴로지를 염두에 두고, 다음 시스템 요구 사항을 사용하여 DHCP 서버를 설정할 호스트를 선택할 수 있습니다.

다음 요구 사항을 호스트가 충족해야 합니다.

- 호스트에서 Solaris 2.6 릴리스 이상을 실행해야 합니다. 많은 수의 클라이언트를 지원해야 하는 경우 Solaris 8 7/01 릴리스 이상 버전을 설치해야 합니다.
- 호스트가 네트워크에서 직접 또는 BOOTP 중계 에이전트를 통해, DHCP를 사용하려는 클라이언트가 속한 모든 네트워크에 액세스할 수 있어야 합니다.
- 호스트가 경로 지정을 사용하도록 구성되어야 합니다.
- 호스트가 네트워크 토폴로지를 반영하는 올바로 구성된 `netmasks` 테이블을 가지고 있어야 합니다.

DHCP 데이터 저장소 선택

DHCP 데이터를 텍스트 파일, 이진 파일 또는 NIS+ 디렉토리 서비스에 저장하도록 선택할 수 있습니다. 다음 표는 각 유형의 데이터 저장소에 대한 특징을 요약하고, 각 데이터 저장소 유형이 사용될 환경을 나타냅니다.

표 13–3 DHCP 데이터 저장소 비교

데이터 저장소 유형	성능	관리	공유	환경
이진 파일	높은 성능, 높은 용량	유지 관리 비용이 낮고 필요한 데이터베이스 서버가 없습니다. DHCP 관리자 또는 `dhtadm` 및 `pntadm`으로 내용을 확인해야 합니다. 정기적인 파일 백업이 제안됩니다.	데이터 저장소를 DHCP 서버 간에 공유할 수 없습니다.	수많은 네트워크가 있고 네트워크당 수천 개의 클라이언트가 포함된 중대형 환경에 적합합니다. 중소형 ISP에 유용합니다.

표 13–3 DHCP 데이터 저장소 비교 *(계속)*

데이터 저장소 유형	성능	관리	공유	환경
NIS+	중간 성능 및 용량, NIS+ 서비스의 성능 및 용량에 의존	DHCP 서버 시스템은 NIS+ 클라이언트로 구성되어야 합니다. NIS+ 서비스 유지 관리가 필요합니다. DHCP 관리자 또는 `dhtadm` 및 `pntadm`으로 내용을 확인해야 합니다. `nisbackup`을 사용하여 정기적으로 백업을 하는 것이 좋습니다.	DHCP 데이터는 NIS+에 분산되고 여러 서버가 동일한 컨테이너에 액세스할 수 있습니다.	네트워크당 클라이언트가 5000개 이하인 중소형 환경에 적합합니다.
텍스트 파일	중간 성능, 낮은 용량	유지 관리 비용이 낮고 필요한 데이터베이스 서버가 없습니다. DHCP 관리자, `dhtadm` 또는 `pntadm` 없이 ASCII 형식을 읽을 수 있습니다. 정기적인 파일 백업이 제안됩니다.	NFS 마운트 지점을 통해 내보낸 하나의 파일 시스템에 DHCP 데이터가 저장된 경우 데이터 저장소를 DHCP 서버 간에 공유할 수 있습니다.	클라이언트가 10,000개 미만이고 네트워크당 수백에서 천 개 정도의 클라이언트가 포함된 소형 환경에 적합합니다.

전통적인 NIS는 빠른 증분 업데이트를 지원하지 않으므로 데이터 저장소 옵션으로 제공되지 않습니다. 네트워크가 NIS를 사용하는 경우 데이터 저장소에 텍스트 파일 또는 이진 파일을 사용해야 합니다.

임대 정책 설정

임대는 DHCP 서버에서 DHCP 클라이언트가 특정 IP 사용하도록 허용한 시간을 가리킵니다. 초기 서버 구성 중 사이트 차원의 임대 정책을 지정해야 합니다. **임대 정책**은 임대 시간을 나타내고 클라이언트가 임대를 갱신할 수 있는지 여부를 지정합니다. 서버는 사용자가 제공한 정보를 사용하여 구성 중 만든 기본 매크로에 옵션 값을 설정합니다. 구성 매크로에 옵션을 설정하여 특정 클라이언트 또는 클라이언트 유형에 대해 서로 다른 임대 정책을 설정할 수 있습니다.

임대 시간은 임대가 유효한 기간을 시, 일, 주 단위로 지정합니다. 클라이언트가 IP 주소에 지정되거나 IP 주소에 대한 임대를 재협상할 때 임대 만료 날짜 및 시간이 계산됩니다. 클라이언트가 DHCP 확인 시 임대 시간의 기간이 시간 기록에 추가됩니다. 예를 들어, DHCP 확인 시간 기록이 September 16, 2005 9:15 A.M.이고 임대 시간이 24시간이라고 가정해 보십시오. 이 예에서 임대 만료 시간은 September 17, 2005 9:15 A.M.입니다. 임대 만료 시간은 클라이언트의 DHCP 네트워크 레코드에 저장되고, DHCP 관리자 또는 `pntadm` 유틸리티로 볼 수 있습니다.

임대 시간 값은 만료된 주소가 신속히 재생 이용되도록 비교적 작게 설정해야 합니다. 또한 임대 시간 값은 DHCP 서비스 장애를 이겨낼 만큼 충분히 커야 합니다. DHCP 서비스가 실행되는 시스템을 수리하는 동안 클라이언트가 작동할 수 있어야 합니다. 일반적인 지침은 예상된 시스템 중단 시간의 2배로 시간을 지정하는 것입니다. 예를 들어, 결함 부품을 입수하고 교체하는 데 4시간이 필요하고 시스템을 재부트하는 경우 임대 시간을 8시간으로 지정하십시오.

임대 협상 옵션은 임대가 만료되기 전에 클라이언트가 서버와 임대를 재협상할 수 있는지 여부를 결정합니다. 임대 협상이 허용된 경우 클라이언트가 남은 임대 시간을 추적합니다. 임대 시간의 절반이 경과된 경우 클라이언트가 DHCP 서버에 해당 임대를 원래 임대 시간으로 연장하도록 요청합니다. IP 주소보다 시스템 수가 많은 환경에서는 임대 협상을 사용 안함으로 설정해야 합니다. 그러면 IP 주소 사용에 시간 제한이 적용됩니다. IP 주소가 충분한 경우 임대가 만료될 때 클라이언트가 네트워크 인터페이스를 강제로 끌어내리는 것을 막으려면 임대 협상을 사용으로 설정해야 합니다. 클라이언트가 새 임대를 얻도록 하면 NFS 및 telnet 세션과 같은 TCP 연결이 중단될 수 있습니다. 서버 구성 중 모든 클라이언트에 대한 임대 협상을 사용으로 설정할 수 있습니다. 구성 매크로에서 `LeaseNeg` 옵션을 사용하여 특정 클라이언트 또는 특정 클라이언트 유형에 대해 임대 협상을 사용으로 설정할 수 있습니다.

주 – 네트워크에 서비스를 제공하는 시스템은 IP 주소를 보존해야 합니다. 이러한 시스템은 단기 임대가 되면 안됩니다. 이러한 시스템에 영구 임대된 IP 주소가 아닌 예약된 수동 IP 주소를 지정하면 DHCP를 사용할 수 있습니다. 그러면 시스템의 IP 주소가 더 이상 사용되지 않을 때를 감지할 수 있습니다.

DHCP 클라이언트에 대한 라우터 결정

호스트 시스템은 로컬 네트워크를 벗어난 네트워크 통신에 라우터를 사용합니다. 호스트는 이러한 라우터의 IP 주소를 알아야 합니다.

DHCP 서버를 구성할 때 두 가지 방법 중 하나로 DHCP 클라이언트에 라우터 주소를 제공해야 합니다. 한 가지 방법은 라우터에 특정 IP 주소를 제공하는 것입니다. 그러나 선호되는 방법은 클라이언트가 라우터 검색 프로토콜을 사용하여 라우터를 찾도록 지정하는 것입니다.

네트워크의 클라이언트가 라우터 검색을 수행할 수 있는 경우 라우터가 하나만 있더라도 라우터 검색 프로토콜을 사용해야 합니다. 라우터 검색을 통해 클라이언트는 네트워크의 라우터 변경 사항에 쉽게 적응할 수 있습니다. 예를 들어, 라우터 실패 후 새 주소를 가진 라우터로 교체된다고 가정해 보십시오. 새 라우터 주소를 가져오기 위해 새 네트워크 구성을 얻을 필요 없이 자동으로 클라이언트가 새 주소를 검색할 수 있습니다.

IP 주소 관리를 위한 결정 사항(작업 맵)

DHCP 서비스 설정의 일부로, 서버가 관리할 IP 주소의 여러 측면을 결정합니다. 네트워크에 여러 개의 DHCP 서버가 필요한 경우 일부 IP 주소의 책임을 각 서버에 지정할 수 있습니다. 주소에 대한 책임을 분담하는 방법을 결정해야 합니다. 다음 표는 DHCP를 네트워크에 사용할 때 IP 주소를 관리하기 위한 작업을 설명하는 작업 맵입니다. 또한 각 작업의 수행 방법을 기술하는 적절한 절에 대한 링크를 포함합니다.

작업	설명	정보
서버가 관리할 주소를 지정합니다.	DHCP 서버가 관리할 주소 개수와 이러한 주소의 내용을 결정합니다.	301 페이지 “IP 주소의 개수 및 범위”
서버가 클라이언트의 호스트 이름을 자동으로 생성하는지 결정합니다.	호스트 이름의 생성 여부를 결정할 수 있도록 클라이언트 호스트 이름의 생성 방법을 알아봅니다.	301 페이지 “클라이언트 호스트 이름 생성”
클라이언트에 지정할 구성 매크로를 결정합니다.	클라이언트에 적절한 매크로를 선택할 수 있도록 클라이언트 구성 매크로를 알아봅니다.	302 페이지 “기본 클라이언트 구성 매크로”
사용할 임대 유형을 결정합니다.	DHCP 클라이언트에 가장 적합한 유형이 무엇인지 결정하기 위해 임대 유형을 알아봅니다.	303 페이지 “동적 및 영구 임대 유형”

IP 주소의 개수 및 범위

초기 서버 구성 중 DHCP 관리자는 총 주소 수와 블록의 첫번째 주소를 지정하여 DHCP 관리하에 IP 주소를 한 블록 또는 범위로 추가할 수 있습니다. DHCP 관리자는 이 정보에서 인접한 주소 목록을 추가합니다. 인접하지 않는 주소 블록이 여러 개 있는 경우 초기 구성 후에 DHCP 관리자의 주소 마법사를 다시 실행하여 나머지를 추가할 수 있습니다.

IP 주소를 구성하기 전에, 추가할 주소의 초기 블록에 포함된 주소 개수와 범위의 첫번째 주소의 IP 주소를 알아봅니다.

클라이언트 호스트 이름 생성

DHCP의 동적 특성이란, IP 주소가 사용 중인 시스템의 호스트 이름과 영구적으로 연관되지 않음을 의미합니다. DHCP 관리 도구에서 이 옵션을 선택하면 각 IP 주소와 연관될 클라이언트 이름을 생성할 수 있습니다. 클라이언트 이름은 접두어(또는 루트 이름)에 대시와 서버 지정 번호를 더해서 구성합니다. 예를 들어, 루트 이름이 `charlie`인 경우 클라이언트 이름은 `charlie-1`, `charlie-2`, `charlie-3` 등입니다.

기본적으로 생성된 클라이언트 이름은 관리 DHCP 서버의 이름으로 시작합니다. 이 전략은 여러 개의 DHCP 서버가 있는 환경에 유용합니다. DHCP 네트워크 테이블에서 주어진 DHCP 서버가 관리하는 클라이언트를 재빨리 확인할 수 있기 때문입니다. 그러나 루트 이름을 다른 선택한 이름으로 변경할 수 있습니다.

IP 주소를 구성하기 전에 DHCP 관리 도구에서 클라이언트 이름을 생성할지 결정하고, 만일 그렇다면 클라이언트 이름에 사용할 루트 이름도 결정합니다.

DHCP 구성 중 호스트 이름을 등록하도록 지정할 경우 생성된 클라이언트 이름은 `/etc/inet/hosts`, DNS 또는 NIS+의 IP 주소에 매핑할 수 있습니다. 자세한 내용은 334 페이지 “클라이언트 호스트 이름 등록”을 참조하십시오.

기본 클라이언트 구성 매크로

DHCP에서 **매크로**는 네트워크 구성 옵션과 지정된 값의 모음입니다. DHCP 서버는 매크로를 사용하여 DHCP 클라이언트에 보낼 네트워크 구성 정보를 결정할 수 있습니다.

DHCP 서버를 구성할 때 관리 도구가 시스템 파일에서, 그리고 지정한 프롬프트나 명령줄 옵션을 통해 직접 정보를 수집합니다. 이 정보를 사용하여 관리 도구는 다음 매크로를 만듭니다.

- **네트워크 주소 매크로** – 네트워크 주소 매크로는 클라이언트 네트워크의 IP 주소와 일치하도록 명명됩니다. 예를 들어, 네트워크가 `192.68.0.0`인 경우 네트워크 주소 매크로도 `192.68.0.0`으로 명명됩니다. 매크로에는 네트워크에 속한 클라이언트에 필요한 서브넷 마스크, 네트워크 브로드캐스트 주소, 기본 라우터나 라우터 검색 토큰 그리고 서버가 NIS/NIS+를 사용하는 경우 NIS/NIS+ 도메인 및 서버와 같은 정보가 포함됩니다. 기타 네트워크에 적용 가능한 옵션이 포함될 수 있습니다. 네트워크 주소 매크로는 291 페이지 “매크로 처리 순서”에 설명된 대로, 네트워크에 있는 모든 클라이언트에 대해 자동으로 처리됩니다.
- **로케일 매크로** – 로케일 매크로는 `Locale`로 명명됩니다. 매크로에는 시간대를 지정하는 UTC(협정 세계시)와의 오프셋(초)이 포함됩니다. 로케일 매크로는 자동으로 처리되지 않지만 서버 매크로에 포함됩니다.
- **서버 매크로** – 서버 매크로는 서버의 호스트 이름과 일치하도록 명명됩니다. 예를 들어, 서버 이름이 `pineola`인 경우 매크로 이름도 `pineola`로 명명됩니다. 서버 매크로에는 임대 정책, 시간 서버, DNS 도메인, DNS 서버 및 기타 구성 프로그램이 시스템 파일에서 얻을 수 있는 정보가 포함됩니다. 서버 매크로에 로케일 매크로가 포함되므로 DHCP 서버는 로케일 매크로를 서버 매크로의 일부로 처리합니다.

  첫번째 네트워크에 대한 IP 주소를 구성할 때 구성 중인 주소를 사용하는 모든 DHCP 클라이언트에 사용할 클라이언트 구성 매크로를 선택해야 합니다. 선택한 매크로는 IP 주소에 매핑됩니다. 서버 매크로에는 이 서버를 사용하는 모든 클라이언트에 필요한 정보가 포함되므로 기본적으로 서버 매크로가 선택됩니다.

클라이언트는 네트워크 주소 매크로에 포함된 옵션을 IP 주소에 매핑된 매크로의 옵션보다 먼저 수신합니다. 이러한 프로세싱 순서로 인해 네트워크 주소 매크로의

옵션과 충돌하는 경우 서버 매크로의 옵션이 우선권을 갖습니다. 매크로 처리 순서에 대한 자세한 내용은 291 페이지 “매크로 처리 순서”를 참조하십시오.

동적 및 영구 임대 유형

임대 유형에 따라 구성 중인 IP 주소에 임대 정책을 적용할지 여부가 결정됩니다. 초기 서버 구성 중 DHCP 관리자는 추가할 주소에 대해 동적 또는 영구 임대를 선택할 수 있습니다. `dhcpconfig` 명령으로 DHCP 서버를 구성하는 경우 동적 임대가 사용됩니다.

IP 주소가 **동적 임대**인 경우 DHCP 서버가 주소를 관리할 수 있습니다. DHCP 서버는 IP 주소를 클라이언트에 할당하고, 임대 시간을 연장하고, 주소가 더 이상 사용되지 않을 때를 감지하고, 주소를 재생 이용할 수 있습니다. IP 주소가 **영구 임대**인 경우 DHCP 서버가 주소를 할당만 할 수 있습니다. 그러면 명시적으로 주소를 해제할 때까지 클라이언트가 주소를 소유합니다. 주소가 해제되면 서버가 다른 클라이언트에 주소를 지정할 수 있습니다. 주소가 영구 임대 유형으로 구성된 동안은 임대 정책을 따르지 않습니다.

IP 주소 범위를 구성할 때 선택한 임대 유형이 범위의 모든 주소에 적용됩니다. DHCP의 효율성을 높이려면 대부분의 주소에 동적 임대를 사용해야 합니다. 나중에 개별 주소를 수정하여 필요한 경우 영구 주소로 만들 수 있습니다. 그러나 총 영구 임대 수는 최소한으로 유지해야 합니다.

예약된 IP 주소 및 임대 유형

IP 주소를 특정 클라이언트에 수동으로 지정하여 예약할 수 있습니다. 예약된 주소는 영구 임대 또는 동적 임대와 연관될 수 있습니다. 예약된 주소가 영구 임대에 지정된 경우 다음 문장이 참입니다.

- 주소에 바인드된 클라이언트에만 주소를 할당할 수 있습니다.
- DHCP 서버는 다른 클라이언트에 주소를 할당할 수 없습니다.
- 주소는 DHCP 서버에 의해 재생 이용될 수 없습니다.

예약된 주소가 동적 임대에 지정된 경우 주소에 바인드된 클라이언트에만 주소를 할당할 수 있습니다. 그러나 클라이언트가 임대 시간을 추적하고 마치 주소가 예약되지 않은 것처럼 임대 연장을 협상할 수 있습니다. 이 전략을 통해 네트워크 테이블을 참조하여 클라이언트가 언제 주소를 사용 중인지 추적할 수 있습니다.

초기 구성 중 모든 IP 주소에 대해 예약된 주소를 만들 수 없습니다. 예약된 주소는 개별 주소에 아주 조금만 사용해야 합니다.

다중 DHCP 서버 계획

IP 주소를 관리할 여러 개의 DHCP 서버를 구성하려면 다음 지침을 고려하십시오.

- 각 서버가 주소 범위를 담당하고 책임이 겹치지 않도록 IP 주소 풀을 분배합니다.
- 가능한 경우 NIS+를 데이터 저장소로 선택하십시오. 그렇지 않으면 텍스트 파일을 선택하고 데이터 저장소의 절대 경로로 공유 디렉토리를 지정합니다. 이진 파일 데이터 저장소는 공유할 수 없습니다.
- 파일 소유권이 정확히 할당되고 서버 기반 매크로를 자동으로 만들 수 있도록 각 서버를 별도로 구성합니다.
- 서버가 최신 정보를 사용하도록 `dhcptab` 테이블에서 옵션 및 매크로를 지정된 간격으로 스캔하도록 서버를 설정합니다. 335 페이지 "DHCP 서버에 대한 성능 옵션 사용자 정의"에 설명된 대로 DHCP 관리자를 사용하여 `dhcptab`의 자동 읽기 일정을 잡을 수 있습니다.
- 서버 간에 서로를 지원하도록 모든 클라이언트가 모든 DHCP 서버에 액세스할 수 있도록 합니다. 유효한 IP 주소 임대를 가진 클라이언트가 클라이언트의 주소를 소유하는 서버에 연결할 수 없을 때 구성을 확인하거나 임대를 연장하려고 시도할 수 있습니다. 20초 동안 기본 서버에 연결을 시도한 후에 다른 서버가 클라이언트에 응답할 수 있습니다. 클라이언트가 특정 IP 주소를 요청했는데 주소를 소유하는 서버를 사용할 수 없을 때 다른 서버 중 하나가 요청을 처리합니다. 이 경우 클라이언트가 요청된 주소를 수신하지 않습니다. 클라이언트는 응답 DHCP 서버가 소유한 IP 주소를 수신합니다.

원격 네트워크의 DHCP 구성 계획

초기 DHCP 구성 후에 DHCP 관리하에 원격 네트워크에 IP 주소를 배치할 수 있습니다. 그러나 시스템 파일이 서버의 로컬에 없으므로 DHCP 관리자와 `dhcpconfig`가 기본값을 제공하기 위해 정보를 조회할 수 없습니다. 따라서 사용자가 정보를 제공해야 합니다. 원격 네트워크를 구성하기 전에 다음 정보를 알아야 합니다.

- 원격 네트워크의 IP 주소.
- 원격 네트워크의 서브넷 마스크. 이 정보는 이름 서비스의 `netmasks` 테이블에서 얻을 수 있습니다. 네트워크가 로컬 파일을 사용하는 경우 네트워크의 시스템에서 `/etc/netmasks`를 찾습니다. 네트워크가 NIS+를 사용하는 경우 `niscat netmasks.org_dir` 명령을 사용합니다. 네트워크가 NIS를 사용하는 경우 `ypcat -k netmasks.byaddr` 명령을 사용합니다. `netmasks` 테이블에 모든 관리할 서브넷에 대한 모든 토폴로지 정보가 들어 있는지 확인합니다.
- 네트워크 유형. 클라이언트는 LAN(Local Area Network) 연결 또는 PPP(Point-to-Point Protocol)를 통해 네트워크에 연결합니다.
- 경로 지정 정보. 클라이언트가 라우터 검색을 사용할 수 있습니까? 아니면, 사용할 라우터의 IP 주소를 결정해야 합니다.
- 적용 가능한 경우, NIS 도메인 및 NIS 서버.

- 적용 가능한 경우, NIS+ 도메인 및 NIS+ 서버.

DHCP 네트워크 추가 절차는 340 페이지 “DHCP 네트워크 추가”를 참조하십시오.

DHCP를 구성할 도구 선택

정보를 수집하고 DHCP 서비스를 계획했으면 DHCP 서버를 구성할 준비가 된 것입니다. DHCP 관리자 또는 명령줄 유틸리티 `dhcpconfig`를 사용하여 서버를 구성할 수 있습니다. DHCP 관리자에서 옵션을 선택하고 데이터를 지정하면 DHCP 서버에서 사용되는 `dhcptab` 및 네트워크 테이블을 만들 수 있습니다. `dhcpconfig` 유틸리티는 명령줄 옵션을 사용하여 데이터를 지정할 수 있습니다.

DHCP 관리자 기능

DHCP 관리자는 Java™ 기술 기반의 GUI 도구로, DHCP 구성 마법사를 제공합니다. DHCP 서버로 구성되지 않은 시스템에서 처음 DHCP 관리자를 실행하면 구성 마법사가 자동으로 시작됩니다. DHCP 구성 마법사는 데이터 저장소 형식, 임대 정책, DNS/NIS/NIS+ 서버 및 도메인, 라우터 주소 등 서버 구성에 필요한 필수적인 정보를 묻는 일련의 대화 상자를 제공합니다. 일부 정보는 마법사가 시스템 파일에서 가져오므로 사용자는 정보가 정확한지만 확인하면 됩니다. 필요한 경우 정보를 수정하십시오.

대화 상자를 진행하고 정보를 승인하면 DHCP 서버 데몬이 서버 시스템에서 시작됩니다. 그런 다음 네트워크의 IP 주소를 구성하는 주소 추가 마법사를 시작할 것인지 묻습니다. 서버의 네트워크만 초기에 DHCP용으로 구성되고 다른 서버 옵션은 기본값이 제공됩니다. 초기 구성이 완료된 후에 DHCP 관리자를 다시 실행하여 네트워크를 추가하고 다른 서버 옵션을 수정할 수 있습니다.

DHCP 구성 마법사에 대한 자세한 내용은 307 페이지 “DHCP 관리자를 사용하여 DHCP 서버 구성 및 구성 해제”를 참조하십시오. DHCP 관리자에 대한 자세한 내용은 318 페이지 “DHCP 관리자 정보”를 참조하십시오.

dhcpconfig 기능

`dhcpconfig` 유틸리티는 DHCP 서버를 구성 및 구성 해제하고 새 데이터 저장소로 변환하고 다른 DHCP 서버에서 데이터를 가져오거나 내보낼 수 있는 옵션을 지원합니다. `dhcpconfig` 유틸리티를 사용하여 DHCP 서버를 구성하는 경우 296 페이지 “시스템 파일 및 넷마스크 테이블 업데이트”에 설명된 대로 시스템 파일에서 정보를 얻습니다. DHCP 관리자에서 하듯이 시스템 파일에서 얻은 정보를 보고 확인할 수는 없습니다. 따라서 `dhcpconfig`를 실행하기 전에 시스템 파일을 업데이트하는 것이 중요합니다. 또한 명령줄 옵션을 사용하면 `dhcpconfig`가 기본적으로 시스템 파일에서 얻은 값을 대체할 수 있습니다. `dhcpconfig` 명령은 스크립트에 사용할 수 있습니다. 자세한 내용은 `dhcpconfig(1M)` 매뉴얼 페이지를 참조하십시오.

DHCP 관리자와 dhcpconfig 비교

다음 표는 두 서버 구성 도구 간의 차이점을 요약한 것입니다.

표 13–4 DHCP 관리자와 dhcpconfig 명령 비교

기능	DHCP 관리자	dhcpconfig에 옵션 포함
시스템에서 수집되는 네트워크 정보	시스템 파일에서 수집된 정보를 보고, 필요한 경우 변경할 수 있습니다.	명령줄 옵션으로 네트워크 정보를 지정할 수 있습니다.
구성 속도	중요치 않은 서버 옵션의 프롬프트를 생략하고, 대신 기본값을 사용하여 구성 프로세스 속도를 높입니다. 초기 구성 후에 중요치 않은 옵션을 변경할 수 있습니다.	가장 빠른 구성 프로세스이지만, 많은 옵션에 값을 지정해야 할 수 있습니다.

14 장, “DHCP 서비스 구성(작업)”에서 DHCP 관리자 또는 `dhcpconfig` 유틸리티를 사용하여 서버를 구성하기 위한 절차를 설명합니다.

◆ ◆ ◆ 14장

14 DHCP 서비스 구성(작업)

네트워크에 DHCP 서비스를 구성할 때 첫번째 DHCP 서버를 구성하고 시작합니다. 다른 DHCP 서버는 나중에 추가할 수 있으며, 데이터 저장소가 공유 데이터를 지원하는 경우 공유 위치에서 동일한 데이터에 액세스할 수 있습니다. 이 장에서는 DHCP 서버를 구성하고 DHCP 관리하에 네트워크 및 연관된 IP 주소를 배치하기 위한 작업을 설명합니다. 또한 DHCP 서버 구성을 해제하는 방법을 설명합니다.

각 작업에는 DHCP 관리자에서 작업을 수행하는 절차와 동일 작업을 `dhcpconfig` 유틸리티로 실행하는 절차가 포함됩니다. 이 장은 다음 정보를 포함합니다.

- 307 페이지 “DHCP 관리자를 사용하여 DHCP 서버 구성 및 구성 해제”
- 314 페이지 “`dhcpconfig` 명령을 사용하여 DHCP 서버 구성 및 구성 해제”

DHCP 서비스를 구성하는 데 문제가 있으면 17 장, “DHCP 문제 해결(참조)”을 참조하십시오.

DHCP 서비스를 구성한 후에 DHCP 서비스 관리에 대한 내용은 15 장, “DHCP 관리(작업)”를 참조하십시오.

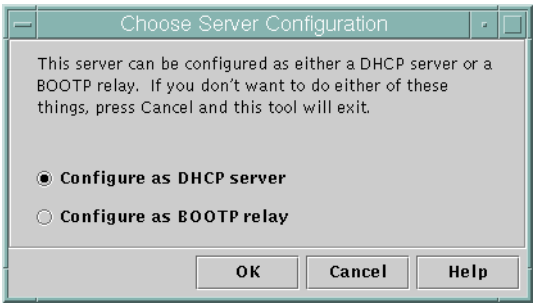
DHCP 관리자를 사용하여 DHCP 서버 구성 및 구성 해제

이 절에는 DHCP 관리자로 DHCP 서버를 구성 및 구성 해제하는 절차가 포함됩니다. DHCP 관리자를 사용하려면 CDE 또는 GNOME과 같은 X Window 시스템을 실행 중이어야 합니다.

DHCP 관리자는 `/usr/sadm/admin/bin/dhcpmgr` 명령과 함께 수퍼 유저로 실행할 수 있습니다. 유틸리티에 대한 일반 정보는 318 페이지 “DHCP 관리자 정보”를 참조하십시오. DHCP 관리자 실행에 대한 자세한 내용은 324 페이지 “DHCP 서비스를 시작 및 중지하는 방법(DHCP 관리자)”을 참조하십시오.

DHCP용으로 구성되지 않은 서버에서 DHCP 관리자를 실행할 때 다음 화면이 표시됩니다. DHCP 서버 또는 BOOTP 중계 에이전트를 구성할지 여부를 지정할 수 있습니다.

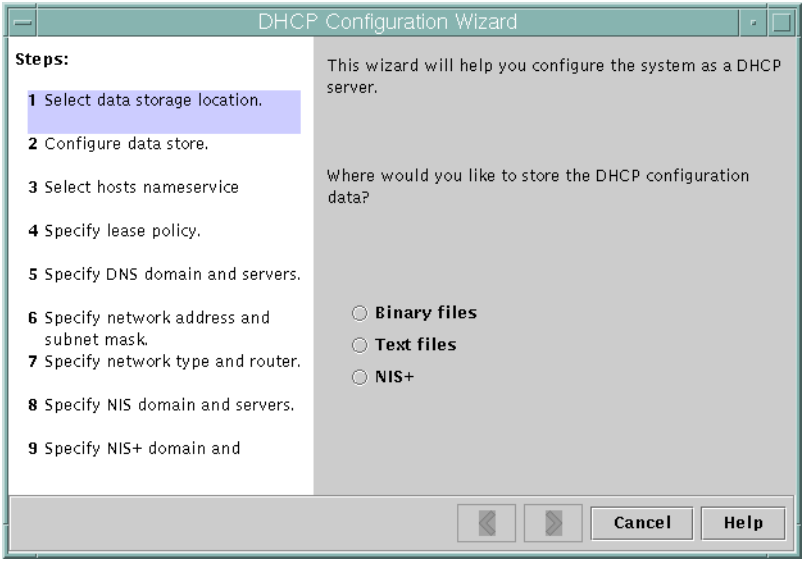
그림 14–1 DHCP 관리자의 Choose Server Configuration(서버 구성 선택) 대화 상자



DHCP 서버 구성

DHCP 서버를 구성할 때 DHCP 관리자가 DHCP 구성 마법사를 시작하고 서버를 구성하는 데 필요한 정보를 묻습니다. 다음 그림은 마법사의 초기 화면을 보여줍니다.

그림 14–2 DHCP 구성 마법사의 초기 화면



마법사 프롬프트에 대답을 마치면 DHCP 관리자가 다음 표에 나열된 항목을 만듭니다.

표 14–1 DHCP 서버 구성 중 생성된 항목

항목	설명	내용
서비스 구성 파일 /etc/inet/dhcpsvc.conf	서버 구성 옵션에 대한 키워드 및 값을 기록합니다.	데이터 저장소 유형 및 위치, 그리고 시스템을 부트할 때 DHCP 데몬을 시작하기 위해 in.dhcpd와 함께 사용되는 옵션. 이 파일을 수동으로 편집하지 마십시오. DHCP 구성 정보를 수정하려면 dhcpmgr 또는 dhcpconfig를 사용해야 합니다.
dhcptab 테이블	아직 존재하지 않는 경우 DHCP 관리자가 dhcptab 테이블을 만듭니다.	매크로 및 옵션과 함께 지정된 값.
Locale로 명명된 로케일 매크로(선택 사항)	UTC(협정 세계시)와 로컬 시간대의 오프셋(초)을 포함합니다.	UTCoffst 옵션과 함께 지정된 시간(초).
서버의 노드 이름과 일치하도록 명명된 서버 매크로	DHCP 서버를 구성한 관리자의 입력에 의해 값이 결정되는 옵션을 포함합니다. 서버가 소유한 주소를 사용하는 모든 클라이언트에 옵션이 적용됩니다.	Locale 매크로와 다음 옵션: ■ Timeserv - 서버의 기본 IP 주소를 가리키도록 설정됩니다. ■ LeaseTim - 임대할 시간(초)으로 설정됩니다. ■ LeaseNeg - 협상 가능한 임대를 선택한 경우 사용됩니다. ■ DNSdmain 및 DNSserv - DNS가 구성된 경우 사용됩니다. ■ Hostname - 값이 지정되면 **안됩니다**. 이 옵션이 존재하면 이름 서비스에서 호스트 이름을 얻어야 합니다.
클라이언트 네트워크의 네트워크 주소와 이름이 같은 네트워크 주소 매크로	DHCP 서버를 구성한 관리자의 입력에 의해 값이 결정되는 옵션을 포함합니다. 매크로 이름으로 지정된 네트워크에 상주하는 모든 클라이언트에 옵션이 적용됩니다.	다음 옵션: ■ Subnet - 로컬 서브넷에 대한 서브넷 마스크로 설정됩니다. ■ Router - 라우터의 IP 주소로 설정됩니다. RDiscvyF - 클라이언트가 라우터 검색을 사용하도록 합니다. ■ Broadcst - 브로드캐스트 IP 주소로 설정됩니다. 이 옵션은 지점간 네트워크가 아닌 경우에만 존재합니다. ■ MTU - 최대 전송 단위입니다. ■ NISdmain 및 NISservs - NIS가 구성된 경우 사용됩니다. ■ NIS+dom 및 NIS+serv - NIS+가 구성된 경우

표 14–1 DHCP 서버 구성 중 생성된 항목 *(계속)*

항목	설명	내용
네트워크 테이블	네트워크에 대한 IP 주소를 만들 때까지 빈 테이블이 생성됩니다.	IP 주소를 추가할 때까지 내용 없음.

▼ DHCP 서버를 구성하는 방법(DHCP 관리자)

시작하기 전에 DHCP 서버를 구성하기 전에 13 장, "DHCP 서비스 계획(작업)"을 읽었는지 확인하십시오. 특히, 다음 작업을 수행하려면 297 페이지 "DHCP 서버 구성을 위한 결정 사항(작업 맵)"의 지침을 사용해야 합니다.

- DHCP 서버로 사용할 시스템을 선택합니다.
- 데이터 저장소, 임대 정책 및 라우터 정보를 결정합니다.

1. **서버 시스템에 수퍼 유저로 로그인합니다.**

2. **DHCP 관리자를 시작합니다.**

   ```
   #/usr/sadm/admin/bin/dhcpmgr &
   ```

3. **Configure as DHCP Server(DHCP 서버로 구성) 옵션을 선택합니다.**

   DHCP Configuration Wizard(DHCP 구성 마법사)가 시작되고 서버 구성을 안내합니다.

4. **계획 단계에서 결정한 사항을 기반으로 옵션을 선택하고 요청된 정보를 입력합니다.**

   어려움이 있으면, 마법사 창에서 Help(도움말)를 눌러 웹 브라우저를 열고 DHCP Configuration Wizard(DHCP 구성 마법사)의 도움말을 표시합니다.

5. **요청된 정보 지정을 마쳤을 때 Finish(완료)를 눌러 서버 구성을 완료합니다.**

6. **Start Address Wizard(주소 마법사를 시작하시겠습니까) 프롬프트에서 Yes(예)를 눌러 서버에 대한 IP 주소를 구성합니다.**

   Add Addresses to Network(네트워크에 주소 추가) 마법사에서 DHCP 통제하에 놓을 주소를 지정할 수 있습니다.

7. **계획 단계에서 결정한 사항을 기반으로 프롬프트에 대답합니다.**

   자세한 내용은 301 페이지 "IP 주소 관리를 위한 결정 사항(작업 맵)"을 참조하십시오. 어려움이 있으면, 마법사 창에서 Help(도움말)를 눌러 웹 브라우저를 열고 Add Addresses to Network(네트워크에 주소 추가) 마법사의 도움말을 표시합니다.

8. **선택 사항을 검토하고, Finish(완료)를 눌러 네트워크 테이블에 IP 주소를 추가합니다.**

   지정된 범위의 각 주소에 대한 레코드로 네트워크 테이블이 업데이트됩니다.

참조 340 페이지 "DHCP 네트워크 추가"에 설명된 대로, 네트워크 마법사를 사용하여 DHCP 서버에 네트워크를 더 추가할 수 있습니다.

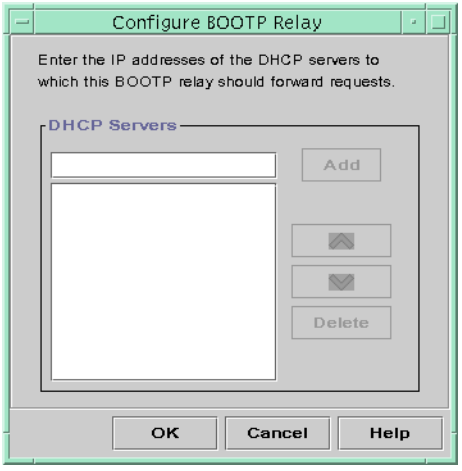
BOOTP 중계 에이전트 구성

BOOTP 중계 에이전트를 구성할 때 DHCP 관리자가 다음 조치를 취합니다.

- 요청을 중계할 하나 이상의 DHCP 서버에 대한 IP 주소를 묻습니다.
- BOOTP 중계 서비스에 필요한 설정을 저장합니다.

다음 그림은 BOOTP 중계 에이전트를 구성하도록 선택할 때 표시되는 화면을 보여줍니다.

그림 14–3 DHCP 관리자의 Configure BOOTP Relay(BOOTP 중계 구성) 대화 상자



▼ BOOTP 중계 에이전트를 구성하는 방법(DHCP 관리자)

시작하기 전에 BOOTP 중계 에이전트를 구성하기 전에 13 장, "DHCP 서비스 계획(작업)"을 읽었는지 확인하십시오. 특히, 사용할 시스템을 선택하려면 298 페이지 "DHCP 서비스를 실행할 호스트 선택"을 참조해야 합니다.

1. **서버 시스템에 수퍼 유저로 로그인합니다.**
2. **DHCP 관리자를 시작합니다.**

```
#/usr/sadm/admin/bin/dhcpmgr &
```

시스템이 DHCP 서버나 BOOTP 중계 에이전트로 구성되지 않은 경우 DHCP 구성 마법사가 시작됩니다. 시스템이 이미 DHCP 서버로 구성된 경우 먼저 서버 구성을 해제해야 합니다. 312 페이지 "DHCP 서버 및 BOOTP 중계 에이전트 구성 해제"을 참조하십시오.

3 **Configure as BOOTP Relay(BOOTP 중계로 구성)를 선택합니다.**

Configure BOOTP Relay(BOOTP 중계 구성) 대화 상자가 열립니다.

4 **하나 이상의 DHCP 서버에 대한 IP 주소 또는 호스트 이름을 입력하고 Add(추가)를 누릅니다.**

지정된 DHCP 서버가 이 BOOTP 중계 에이전트에서 수신된 BOOTP 또는 DHCP 요청을 처리하도록 구성되어야 합니다.

5 **OK(확인)를 눌러 대화 상자를 종료합니다.**

DHCP 관리자는 응용 프로그램을 종료하려면 File(파일) 메뉴만 제공하고 서버를 관리하려면 Service(서비스) 메뉴만 제공합니다. 사용 안함으로 설정된 메뉴 옵션은 DHCP 서버에만 유용합니다.

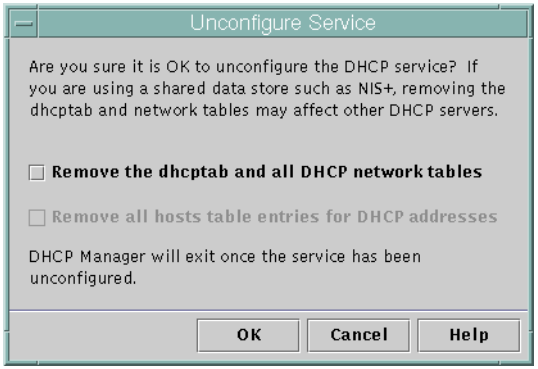
DHCP 서버 및 BOOTP 중계 에이전트 구성 해제

DHCP 서버나 BOOTP 중계 에이전트 구성을 해제할 때 DHCP 관리자가 다음 조치를 취합니다.

- DHCP 데몬(`in.dhpcd`) 프로세스를 중지합니다.
- 데몬 시작 및 데이터 저장소 위치에 대한 정보를 기록하는 `/etc/inet/dhcpsvc.conf` 파일을 제거합니다.

다음 그림은 DHCP 서버 구성을 해제하도록 선택할 때 표시되는 화면을 보여줍니다.

그림 14–4 DHCP 관리자의 Unconfigure Service(서비스 구성 해제) 대화 상자



구성 해제된 서버의 DHCP 데이터

DHCP 서버 구성을 해제할 때 `dhcptab` 테이블 및 DHCP 네트워크 테이블로 무엇을 할지 결정해야 합니다. 데이터를 서버 간에 공유하는 경우 `dhcptab` 및 DHCP 네트워크 테이블을 제거하면 안됩니다. 테이블을 제거하면 DHCP를 네트워크에서 사용할 수 없게 됩니다. 데이터는 NIS+ 또는 내보낸 로컬 파일 시스템을 통해 공유될 수 있습니다. `/etc/inet/dhcpsvc.conf` 파일은 사용된 데이터 저장소와 해당 위치를 기록합니다.

DHCP 서버 구성을 해제하되, 데이터를 제거하는 옵션을 선택하지 않으면 데이터를 그대로 유지할 수 있습니다. 서버 구성을 해제하고 데이터를 그대로 유지하는 경우 DHCP 서버를 사용 안함으로 설정하는 것입니다.

다른 DHCP 서버가 IP 주소의 소유권을 얻도록 하려면 DHCP 데이터를 다른 DHCP 서버로 이동해야 합니다. 현재 서버 구성을 해제하기 전에 데이터를 이동해야 합니다. 자세한 내용은 390 페이지 “DHCP 서버 간 구성 데이터 이동(작업 맵)”을 참조하십시오.

확실히 데이터를 제거하고 싶으면 `dhcptab` 및 네트워크 테이블을 제거하는 옵션을 선택할 수 있습니다. DHCP 주소에 대한 클라이언트 이름을 생성했으면 이러한 항목도 호스트 테이블에서 제거하도록 선택할 수 있습니다. 클라이언트 이름 항목을 DNS, `/etc/inet/hosts` 또는 NIS+에서 제거할 수 있습니다.

BOOTP 중계 에이전트 구성을 해제하기 전에, DHCP 서버로 요청을 전달하기 위해 이 에이전트에 의존하는 클라이언트가 없는지 확인합니다.

▼ DHCP 서버 또는 BOOTP 중계 에이전트 구성을 해제하는 방법(DHCP 관리자)

1 **수퍼 유저로 로그인합니다.**

2 **DHCP 관리자를 시작합니다.**

```
#/usr/sadm/admin/bin/dhcpmgr &
```

3 **Service(서비스) 메뉴에서 Unconfigure(구성 해제)를 선택합니다.**

Unconfigure Service(서비스 구성 해제) 대화 상자가 표시됩니다. 서버가 BOOTP 중계 에이전트인 경우 중계 에이전트 구성을 해제할 것인지 확인하는 대화 상자가 표시됩니다. 서버가 DHCP 서버인 경우 DHCP 데이터로 무엇을 할지 결정하고 대화 상자에서 항목을 선택해야 합니다. 그림 14–4를 참조하십시오.

4 **(옵션) 데이터를 제거하는 옵션을 선택합니다.**

서버가 NIS+를 통해, 또는 NFS를 통해 공유되는 파일을 통해 공유 데이터를 사용하는 경우 데이터를 제거하는 어떤 옵션도 선택하지 마십시오. 서버가 공유 데이터를 사용하지 않는 경우 데이터를 제거하는 옵션을 하나 또는 양쪽 다 선택하십시오.

데이터 제거에 대한 자세한 내용은 313 페이지 “구성 해제된 서버의 DHCP 데이터”를 참조하십시오.

5 **OK(확인)를 눌러 서버 구성을 해제합니다.**

Unconfigure Service(서비스 구성 해제) 대화 상자 및 DHCP 관리자가 닫힙니다.

dhcpconfig 명령을 사용하여 DHCP 서버 구성 및 구성 해제

이 절에는 dhcpconfig를 명령줄 옵션과 함께 사용하여 DHCP 서버 또는 BOOTP 중계 에이전트를 구성 및 구성 해제하는 절차가 포함됩니다.

▼ DHCP 서버를 구성하는 방법(dhcpconfig -D)

시작하기 전에 DHCP 서버를 구성하기 전에 13 장, “DHCP 서비스 계획(작업)”을 읽었는지 확인하십시오. 특히, 다음 작업을 수행하려면 297 페이지 “DHCP 서버 구성을 위한 결정 사항(작업 맵)”의 지침을 사용해야 합니다.

- DHCP 서버로 사용할 시스템을 선택합니다.
- 데이터 저장소, 임대 정책 및 라우터 정보를 결정합니다.

1 **DHCP 서버를 구성하려는 시스템에 로그인합니다.**

2 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

3 **다음 형식의 명령을 입력하여 DHCP 서버를 구성합니다.**

#/usr/sbin/dhcpconfig -D -r *datastore* **-p** *location*

*datastore*는 SUNWfiles, SUNWbinfiles 또는 SUNWnisplus 중 하나입니다.

*location*은 DHCP 데이터를 저장할 데이터 저장소 종속 위치입니다. SUNWfiles 및 SUNWbinfiles에 대해 위치는 절대 경로 이름이어야 합니다. SUNWnisplus의 경우 위치는 전체 NIS+ 디렉토리여야 합니다.

예를 들어, 다음과 비슷한 명령을 입력할 수 있습니다.

```
dhcpconfig -D -r SUNWbinfiles -p /var/dhcp
```

dhcpconfig 유틸리티는 호스트의 시스템 파일과 네트워크 파일을 사용하여 DHCP 서버 구성에 사용되는 값을 결정합니다. 기본값을 대체할 수 있는 dhcpconfig 명령의 추가 옵션에 대한 내용은 dhcpconfig(1M) 매뉴얼 페이지를 참조하십시오.

4 **하나 이상의 네트워크를 DHCP 서비스에 추가합니다.**

네트워크 추가 절차는 342 페이지 “DHCP 네트워크를 추가하는 방법(dhcpconfig)”을 참조하십시오.

▼ BOOTP 중계 에이전트를 구성하는 방법(dhcpconfig -R)

시작하기 전에 298 페이지 “DHCP 서비스를 실행할 호스트 선택”에 나열된 요구 사항을 사용하여 BOOTP 중계 에이전트로 사용할 시스템을 선택합니다.

1 **BOOTP 중계 에이전트로 구성하려는 서버에 로그인합니다.**

2 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

3 **다음 형식의 명령을 입력하여 BOOTP 중계 에이전트를 구성합니다.**

```
# /usr/sbin/dhcpconfig -R server-addresses
```

요청을 전달할 DHCP 서버의 IP 주소를 하나 이상 지정합니다. 하나 이상의 주소를 지정하는 경우 콤마로 주소를 구분하십시오.

예를 들어, 다음과 비슷한 명령을 입력할 수 있습니다.

```
/usr/sbin/dhcpconfig -R 192.168.1.18,192.168.42.132
```

▼ DHCP 서버 또는 BOOTP 중계 에이전트 구성을 해제하는 방법(dhcpconfig -U)

1 **구성을 해제할 DHCP 서버 또는 BOOTP 중계 에이전트 시스템에 로그인합니다.**

2 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

3 **DHCP 서버 또는 BOOTP 중계 에이전트 구성을 해제합니다.**

```
# /usr/sbin/dhcpconfig -U
```

서버가 공유 데이터를 사용하지 않는 경우 `-x` 옵션을 사용하여 `dhcptab` 및 네트워크 테이블을 제거할 수 있습니다. 서버가 공유 데이터를 사용하는 경우 `-x` 옵션을 사용하지 마십시오. `-h` 옵션을 사용하여 호스트 테이블에서 호스트 이름을 제거할 수 있습니다. `dhcpconfig` 옵션에 대한 자세한 내용은 `dhcpconfig(1M)` 매뉴얼 페이지를 참조하십시오.

데이터 제거에 대한 자세한 내용은 313 페이지 “구성 해제된 서버의 DHCP 데이터”를 참조하십시오.

◆ ◆ ◆ 15 장

15 DHCP 관리(작업)

이 장에서는 DHCP 서비스를 관리할 때 유용한 작업에 대해 설명합니다. 이 장에는 서버, BOOTP 중계 에이전트 및 클라이언트에 대한 작업이 포함됩니다. 각 작업에는 DHCP 관리자에서 작업을 수행하는 데 도움이 되는 절차와 동일 작업을 DHCP 명령줄 유틸리티로 실행하는 절차가 포함됩니다. DHCP 명령줄 유틸리티는 매뉴얼 페이지에 더 자세하게 설명되어 있습니다.

이 장을 시작하기 전에 먼저 DHCP 서비스 및 네트워크의 초기 구성을 완료해야 합니다. DHCP 구성은 14 장, “DHCP 서비스 구성(작업)”을 참조하십시오.

이 장은 다음 정보를 포함합니다.

DHCP 관리자 정보

DHCP 관리자는 DHCP 서비스에서 관리 작업을 수행하는 데 사용할 수 있는 GUI(그래픽 사용자 인터페이스) 도구입니다.

DHCP 관리자 창

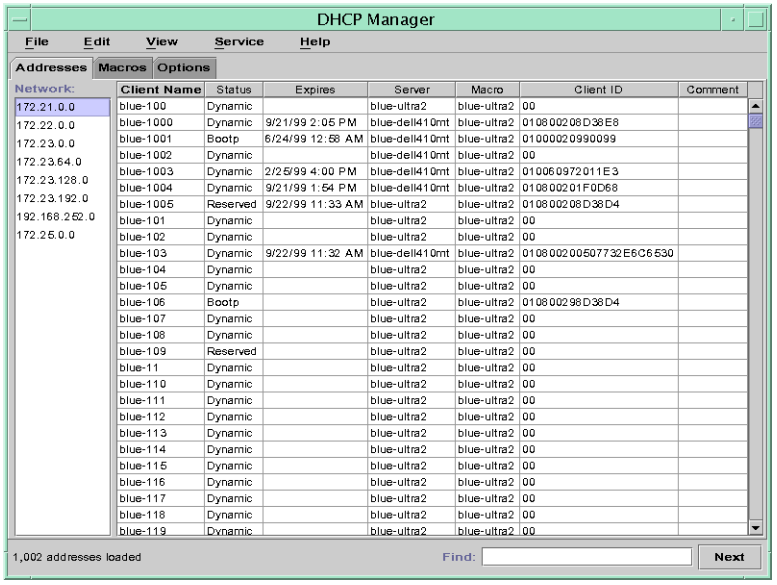
DHCP 관리자 창의 모양은 DHCP 관리자가 실행되는 시스템에 DHCP 서버가 구성된 방식에 따라 다릅니다.

DHCP 관리자는 시스템이 DHCP 서버로 구성된 경우 탭 기반 창을 사용합니다. 작업할 정보 유형에 대한 탭을 선택합니다. DHCP 관리자에는 다음과 같은 탭이 있습니다.

- **Addresses(주소)** 탭 – DHCP 관리 대상인 모든 네트워크 및 IP 주소가 나열됩니다. Addresses(주소) 탭에서 네트워크 및 IP 주소에 대한 작업을 수행할 수 있습니다. 항목을 개별적으로 또는 블록으로 추가하거나 삭제할 수 있습니다. 개별적으로 네트워크 또는 IP 주소의 등록 정보를 수정할 수도 있고, 주소 블록에 대해 동시에 동일하게 등록 정보를 수정할 수도 있습니다. DHCP 관리자를 시작하면 Addresses(주소) 탭이 먼저 열립니다.
- **Macros(매크로)** 탭 – DHCP 구성 테이블(`dhcptab`)에 있는 사용 가능한 모든 매크로와 해당 매크로 내에 포함된 옵션이 나열됩니다. Macros(매크로) 탭에서 매크로를 만들거나 삭제할 수 있습니다. 옵션을 추가하고 옵션에 값을 제공하여 매크로를 수정할 수도 있습니다.
- **Options(옵션)** 탭 – 이 DHCP 서버에 대해 정의된 모든 옵션이 나열됩니다. 이 탭에 나열되는 옵션은 DHCP 프로토콜에 정의된 표준 옵션이 아닙니다. 표준 옵션을 확장한 옵션이 나열되며, 이러한 옵션은 Extended(확장), Vendor(공급업체) 또는 Site(사이트) 클래스를 가지고 있습니다. 표준 옵션은 어떤 방법으로도 변경할 수 없으므로 여기에 나열되지 않습니다.

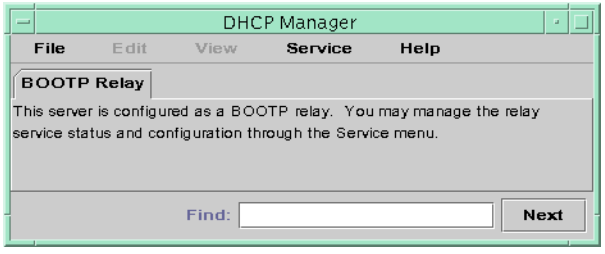
다음 그림은 DHCP 서버에서 DHCP 관리자를 시작하는 경우 DHCP 관리자 창이 어떻게 표시되는지 보여줍니다.

그림 15–1 DHCP 서버 시스템의 DHCP 관리자



서버가 BOOTP 중계 에이전트로 구성된 경우 DHCP 관리자 창에는 이러한 탭이 표시되지 않습니다. BOOTP 중계 에이전트에는 이러한 정보가 필요하지 않습니다. BOOTP 중계 에이전트의 등록 정보를 수정하고 DHCP 관리자로 DHCP 데몬을 중지하거나 시작하는 것만 가능합니다. 다음 그림은 BOOTP 중계 에이전트로 구성된 시스템에서 DHCP 관리자가 어떻게 표시되는지 보여줍니다.

그림 15–2 BOOTP 중계 에이전트의 DHCP 관리자



DHCP 관리자 메뉴

DHCP 관리자 메뉴에는 다음과 같은 항목이 있습니다.

- **File(파일)** – DHCP 관리자를 종료합니다.
- **Edit(편집)** – 네트워크, 주소, 매크로 및 옵션에 대한 관리 작업을 수행합니다.

- **View(보기)** – 현재 선택된 탭의 모양을 변경합니다.
- **Service(서비스)** – DHCP 데몬 및 데이터 저장소를 관리합니다.
- **Help(도움말)** – 웹 브라우저를 열고 DHCP 관리자에 대한 도움말을 표시합니다.

DHCP 관리자가 BOOTP 중계 에이전트에서 실행되는 경우 Edit(편집) 및 View(보기) 메뉴는 사용으로 설정되지 않습니다.

모든 DHCP 관리 작업은 Edit(편집) 및 Service(서비스) 메뉴를 통해 수행됩니다.

Edit(편집) 메뉴의 명령을 사용하여 선택한 탭의 항목을 만들고, 삭제 및 수정할 수 있습니다. 항목에는 네트워크, 주소, 매크로 및 옵션이 포함될 수 있습니다. Addresses(주소) 탭이 선택되면 Edit(편집) 메뉴에 마법사도 표시됩니다. 마법사는 네트워크 및 복수 IP 주소를 만드는 과정을 안내하는 일련의 대화 상자입니다.

Service(서비스) 메뉴에는 DHCP 데몬을 관리할 수 있는 명령이 표시됩니다. Service(서비스) 메뉴에서는 다음 작업을 수행할 수 있습니다.

- DHCP 데몬을 시작 및 중지합니다.
- DHCP 데몬을 사용/사용 안함으로 설정합니다.
- 서버 구성을 수정합니다.
- 서버 구성을 해제합니다.
- 데이터 저장소를 변환합니다.
- 서버에서 데이터를 내보내고 가져옵니다.

DHCP 관리자 시작 및 중지

DHCP 서버 시스템에서 수퍼 유저로 DHCP 관리자를 실행해야 합니다. DHCP 관리자를 원격으로 실행해야 하는 경우 X Window 원격 표시 기능을 사용하여 사용자 시스템으로 표시를 전송할 수 있습니다.

▼ DHCP 관리자를 시작 및 중지하는 방법

1. **DHCP 서버 시스템에 수퍼 유저로 로그인합니다.**
2. **(옵션) 원격으로 DHCP 서버 시스템에 로그인한 경우 다음과 같은 방법으로 로컬 시스템에 DHCP 관리자를 표시합니다.**

   a. **로컬 시스템에서 다음을 입력합니다.**

      # xhost +*server-name*

   b. **원격 DHCP 서버 시스템에서 다음을 입력합니다.**

      # DISPLAY=*local-hostname***;export DISPLAY**

3 **DHCP 관리자를 시작합니다.**

```
# /usr/sadm/admin/bin/dhcpmgr &
```

DHCP 관리자 창이 열립니다. 서버가 DHCP 서버로 구성된 경우 창에 Addresses(주소) 탭이 표시됩니다. 서버가 BOOTP 중계 에이전트로 구성된 경우 창에 탭이 표시되지 않습니다.

4 **DHCP 관리자를 중지하려면 File(파일) 메뉴에서 Exit(종료)를 선택합니다.**

DHCP 관리자 창이 닫힙니다.

DHCP 명령에 사용자 액세스 설정

기본적으로 `root` 사용자만 `dhcpconfig`, `dhtadm` 및 `pntadm` 명령을 실행할 수 있습니다. `root`가 아닌 사용자가 명령을 사용하려면 이러한 명령에 대해 RBAC(역할 기반 액세스 제어)를 설정할 수 있습니다.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

`rbac(5)`, `exec_attr(4)`, `user_attr(4)` 등의 매뉴얼 페이지도 유용합니다.

다음 절차는 사용자가 DHCP 명령을 실행할 수 있도록 DHCP Management 프로파일을 지정하는 방법을 설명합니다.

▼ DHCP 명령에 사용자 액세스를 부여하는 방법

1 **DHCP 서버 시스템에 수퍼 유저로 로그인합니다.**

2 **`/etc/user_attr` 파일에 사용자나 역할을 추가합니다.**

`/etc/user_attr` 파일을 편집하여 다음 형태의 항목을 추가합니다. DHCP 서비스를 관리할 사용자나 역할마다 하나씩 항목을 추가합니다.

username`::::type=normal;profiles=DHCP Management`

예를 들어, 사용자 `ram`에 대해 다음 항목을 추가합니다.

```
ram::::type=normal;profiles=DHCP Management
```

DHCP 서버 작업

▼ ISC DHCP 서버를 구성하는 방법

이러한 단계를 사용하여 초기에 ISC DHCP 서버를 구성할 수 있습니다.

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **DHCP 구성 파일을 편집합니다.**

`/etc/dhcp/dhcpd4.conf` 또는 `/etc/dhcp/dhcpd6.conf` 파일을 만듭니다. 자세한 내용은 `dhcpd.conf`(5) 매뉴얼 페이지를 참조하십시오.

3 **필요한 서비스를 사용으로 설정합니다.**

```
# svcadm enable service
```

*service*는 다음 값 중 하나일 수 있습니다.

`svc:/network/dhcp/server:ipv4`	IPv4 클라이언트에서 DHCP 및 BOOTP 요청을 제공합니다.
`svc:/network/dhcp/server:ipv6`	IPv6 클라이언트에서 DHCP 및 BOOTP 요청을 제공합니다.
`svc:/network/dhcp/relay:ipv4`	IPv4 클라이언트에서 DHCP 서버의 네트워크로 DHCP 및 BOOTP 요청을 중계합니다.
`svc:/network/dhcp/relay:ipv6`	IPv6 클라이언트에서 DHCP 서버의 네트워크로 DHCP 및 BOOTP 요청을 중계합니다.

▼ DHCP 서비스의 구성을 수정하는 방법

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **DHCP 구성 파일을 편집합니다.**

`/etc/dhcp/dhcpd4.conf` 또는 `/etc/dhcp/dhcpd6.conf` 파일을 편집합니다. 자세한 내용은 `dhcpd.conf`(5) 매뉴얼 페이지를 참조하십시오.

3 **SMF 데이터를 새로 고칩니다.**

```
# svcadm refresh service
```

DHCP 서비스 시작 및 중지

이 절에서는 DHCP 관리자 및 `dhcpconfig` 명령을 사용하여 DHCP 서비스를 시작하고 중지하는 방법을 설명합니다. DHCP 서비스는 SMF(서비스 관리 기능) 명령으로도 시작하고 중지할 수 있습니다. DHCP 서비스에서 SMF 명령을 사용하는 방법은 325 페이지 "DHCP 서비스 및 서비스 관리 기능"을 참조하십시오.

DHCP 서비스를 시작 및 중지하는 작업에는 DHCP 데몬의 작업에 영향을 미칠 수 있는 여러 레벨의 작업이 포함됩니다. 원하는 결과를 얻을 수 있는 올바른 절차를 선택하려면 각 작업의 의미를 파악하고 있어야 합니다. 작업 조건은 다음과 같습니다.

- **start, stop 및 restart 명령**은 현재 세션에 대해서만 데몬에 영향을 줍니다. 예를 들어, DHCP 서비스를 중지하면 데몬이 종료되지만 시스템을 재부트하면 다시 시작됩니다. 서비스를 중지할 때 DHCP 데이터 테이블은 영향을 받지 않습니다. DHCP 관리자 또는 SMF 명령을 사용하여 DHCP 서비스를 사용으로 설정하거나 사용 안함으로 설정하지 않고 일시적으로 시작하거나 중지할 수 있습니다.
- **enable 및 disable 명령**은 현재 및 향후 세션에 대해 데몬에 영향을 미칩니다. DHCP 서비스를 사용 안함으로 설정하면 현재 실행 중인 데몬이 종료되고 서버를 재부트해도 시작되지 않습니다. 시스템 부트 시 자동으로 시작되도록 DHCP 데몬을 설정해야 합니다. DHCP 데이터 테이블은 영향을 받지 않습니다. DHCP 관리자, `dhcpconfig` 명령 또는 SMF 명령을 사용하여 DHCP 서비스를 사용으로 설정하거나 사용 안함으로 설정할 수 있습니다.
- **unconfigure 명령**을 사용하면 데몬을 종료하고, 시스템 재부트 시 데몬이 시작되지 않도록 하고, DHCP 데이터 테이블을 제거할 수 있습니다. DHCP 관리자 또는 `dhcpconfig` 명령을 사용하여 DHCP 서비스의 구성을 해제할 수 있습니다. 구성 해제는 14 장, "DHCP 서비스 구성(작업)"에 설명되어 있습니다.

주 – 서버에 네트워크 인터페이스가 여러 개 있지만 일부 네트워크에서는 DHCP 서비스를 제공하지 않으려는 경우 338 페이지 "DHCP 모니터링을 위한 네트워크 인터페이스 지정"을 참조하십시오.

다음 절차를 통해 DHCP 서비스를 시작, 중지하거나 사용/사용 안함으로 설정할 수 있습니다.

▼ DHCP 서비스를 시작 및 중지하는 방법(DHCP 관리자)

1. **DHCP 서버 시스템에 수퍼 유저로 로그인합니다.**
2. **DHCP 관리자를 시작합니다.**

   ```
   # /usr/sadm/admin/bin/dhcpmgr &
   ```

3. **다음 중 하나를 선택합니다.**
   - **Service(서비스) 메뉴에서 Start(시작)를 선택하여 DHCP 서비스를 시작합니다.**
   - **Service(서비스) 메뉴에서 Stop(중지)을 선택하여 DHCP 서비스를 중지합니다.**
     DHCP 데몬은 다시 시작되거나 시스템이 재부트될 때까지 중지됩니다.
   - **Service(서비스) 메뉴에서 Restart(다시 시작)를 선택하여 DHCP 서비스를 중지하고 즉시 다시 시작합니다.**

▼ DHCP 서비스를 사용/사용 안함으로 설정하는 방법(DHCP 관리자)

- **DHCP 관리자에서 다음 중 하나를 선택합니다.**
  - **Service(서비스) 메뉴에서 Enable(사용)을 선택하여 시스템이 부트할 때 DHCP 데몬이 자동으로 시작하도록 구성합니다.**
    DHCP 서비스는 사용으로 설정된 즉시 시작됩니다.
  - **Service(서비스) 메뉴에서 Disable(사용 안함)을 선택하여 시스템이 부트할 때 DHCP 데몬이 자동으로 시작하지 않도록 설정합니다.**
    DHCP 서비스는 사용 안함으로 설정된 즉시 중지됩니다.

▼ DHCP 서비스를 사용/사용 안함으로 설정하는 방법(dhcpconfig -S)

1. **DHCP 서버 시스템에 로그인합니다.**
2. **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**
   DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

3 **다음 중 한 가지를 선택합니다.**

- **DHCP 서비스를 사용으로 설정하려면 다음 명령을 입력합니다.**

  ```
  # /usr/sbin/dhcpconfig -S -e
  ```

- **DHCP 서비스를 사용 안함으로 설정하려면 다음 명령을 입력합니다.**

  ```
  # /usr/sbin/dhcpconfig -S -d
  ```

DHCP 서비스 및 서비스 관리 기능

SMF(서비스 관리 기능)는 **Oracle Solaris 관리: 기본 관리**의 18 장, "서비스 관리(개요)"에 설명되어 있습니다. SMF `svcadm` 명령을 사용하여 DHCP 서버를 사용으로 설정하고 시작하거나, 사용 안함으로 설정하고 중지할 수 있습니다. 하지만 SMF 명령을 사용하여 DHCP 도구에서 설정할 수 있는 DHCP 서비스 옵션을 수정할 수는 없습니다. 특히 `/etc/dhcp/dhcpsvc.conf` 파일에 저장된 서비스 옵션은 SMF 도구를 사용하여 설정할 수 없습니다.

다음 표에서는 DHCP 명령을 해당 SMF 명령에 매핑합니다.

표 15–1 DHCP 서버 작업에 대한 SMF 명령

작업	DHCP 명령	SMF 명령
DHCP 서비스를 사용으로 설정합니다.	`dhcpconfig -S -e`	`svcadm enable svc:/network/dhcp-server`
DHCP 서비스를 사용 안함으로 설정합니다.	`dhcpconfig -S -d`	`svcadm disable svc:/network/dhcp-server`
현재 세션에 대해서만 DHCP 서비스를 시작합니다.	없음	`svcadm enable -t svc:/network/dhcp-server`
현재 세션에 대해 DHCP 서비스를 중지합니다.	없음	`svcadm disable -t svc:/network/dhcp-server`
DHCP 서비스를 다시 시작합니다.	`dhcpconfig -S -r`	`svcadm restart svc:/network/dhcp-server`

DHCP 서비스 옵션 수정(작업 맵)

DHCP 서비스의 일부 추가 기능에 대한 값을 변경할 수 있습니다. DHCP 관리자의 초기 구성 중에는 이러한 기능이 제공되지 않았을 수 있습니다. 서비스 옵션을 변경하려면 DHCP 관리자의 Modify Service Options(서비스 옵션 수정) 대화 상자를 사용합니다. `dhcpconfig` 명령으로 옵션을 지정할 수도 있습니다.

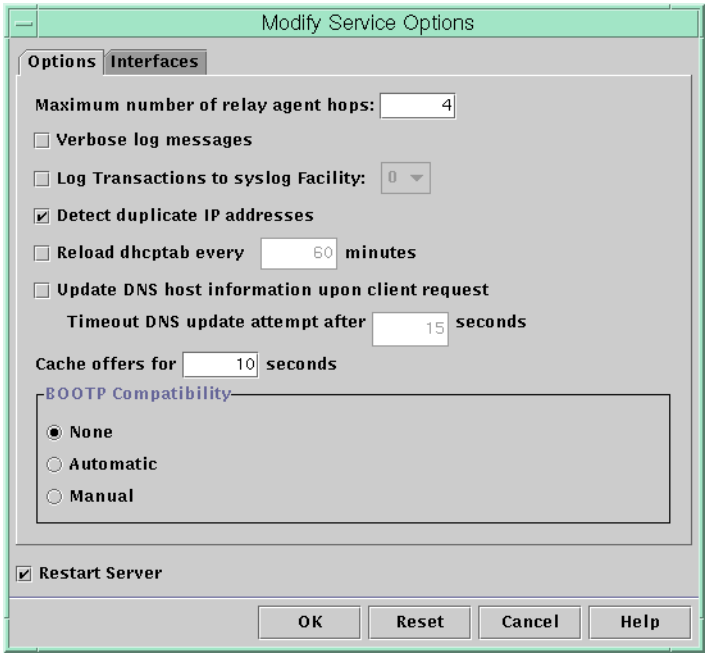
다음 표는 DHCP 서비스 옵션 수정 작업을 설명하는 맵입니다. 이 표에는 각 작업을 수행하는 절차에 대한 링크가 포함되어 있습니다.

작업	설명	수행 방법
로깅 옵션을 변경합니다.	로깅을 사용 또는 사용 안함으로 설정하고, DHCP 트랜잭션을 기록하는 데 사용할 `syslog` 기능을 선택합니다.	329 페이지 “상세 정보 DHCP 로그 메시지를 생성하는 방법(DHCP 관리자)” 329 페이지 “상세 정보 DHCP 로그 메시지를 생성하는 방법(명령줄)” 330 페이지 “DHCP 트랜잭션 로깅을 사용/사용 안함으로 설정하는 방법(DHCP 관리자)” 331 페이지 “DHCP 트랜잭션 로깅을 사용/사용 안함으로 설정하는 방법(명령줄)” 331 페이지 “별도의 `syslog` 파일에 DHCP 트랜잭션을 기록하는 방법”
DNS 업데이트 옵션을 변경합니다.	호스트 이름을 제공하는 클라이언트에 대해 DNS 항목을 동적으로 추가하는 서버 기능을 사용 또는 사용 안함으로 설정합니다. 서버가 DNS 업데이트 시도에 사용하는 최대 시간을 결정합니다.	333 페이지 “DHCP 클라이언트에 대한 동적 DNS 업데이트를 사용으로 설정하는 방법”
중복 IP 주소 감지를 사용 또는 사용 안함으로 설정합니다.	클라이언트에 IP 주소를 제공하기 전에 해당 주소가 이미 사용 중인지 파악하는 DHCP 서버의 기능을 사용 또는 사용 안함으로 설정합니다.	336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(DHCP 관리자)” 336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(명령줄)”
DHCP 서버의 구성 정보 읽기 옵션을 변경합니다.	지정된 간격에 `dhcptab`을 자동으로 읽는 기능을 사용 또는 사용 안함으로 설정하거나 읽기 간격을 변경합니다.	336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(DHCP 관리자)” 336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(명령줄)”

작업	설명	수행 방법
중계 에이전트 홉 수를 변경합니다.	DHCP 데몬에 의해 삭제되기 전에 요청이 이동할 수 있는 네트워크 수를 늘리거나 줄입니다.	336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(DHCP 관리자)” 336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(명령줄)”
IP 주소 제공이 캐시되는 기간을 변경합니다.	DHCP 서비스가 제공된 IP 주소를 새 클라이언트에 제공하기 전에 예약해 두는 시간(초 단위)을 늘리거나 줄입니다.	336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(DHCP 관리자)” 336 페이지 “DHCP 성능 옵션을 사용자 정의하는 방법(명령줄)”

다음 그림은 DHCP 관리자의 Modify Service Options(서비스 옵션 수정) 대화 상자를 보여줍니다.

그림 15–3 DHCP 관리자의 Modify Service Options(서비스 옵션 수정) 대화 상자



DHCP 로깅 옵션 변경

DHCP 서비스는 DHCP 서비스 메시지 및 DHCP 트랜잭션을 `syslog`에 기록할 수 있습니다. `syslog`에 대한 자세한 내용은 `syslogd(1M)` 및 `syslog.conf(4)` 매뉴얼 페이지를 참조하십시오.

syslog에 기록되는 DHCP 서비스 메시지에는 다음이 포함됩니다.

- DHCP 서비스가 클라이언트 또는 사용자의 요청을 수행할 수 없는 조건을 알리는 오류 메시지
- 비정상적이지만 DHCP 서비스가 요청을 수행할 수는 있는 조건을 알리는 경고 및 알림

DHCP 데몬의 상세 정보 표시 옵션을 사용하여 보고되는 정보의 양을 늘릴 수 있습니다. 상세 정보 메시지 출력은 DHCP 문제를 해결하는 데 도움이 될 수 있습니다. 329 페이지 "상세 정보 DHCP 로그 메시지를 생성하는 방법(DHCP 관리자)"을 참조하십시오.

또 다른 유용한 문제 해결 방법은 트랜잭션 로깅입니다. 트랜잭션은 DHCP 서버 또는 BOOTP 중계와 클라이언트 간에 교환되는 모든 내용에 대한 정보를 제공합니다. DHCP 트랜잭션에는 다음과 같은 메시지 유형이 포함됩니다.

- ASSIGN – IP 주소 지정
- ACK – 서버는 클라이언트가 제공된 IP 주소를 수락함을 확인하고 구성 매개변수를 전송함
- EXTEND – 임대 연장
- RELEASE – IP 주소 해제
- DECLINE – 클라이언트가 주소 지정을 거부함
- INFORM – 클라이언트가 네트워크 구성 매개변수를 요청하지만 IP 주소는 요청하지 않음
- NAK – 서버가 이전에 사용된 IP 주소를 사용하겠다는 클라이언트의 요청을 확인하지 않음
- ICMP_ECHO – 서버가 잠재적인 IP 주소가 다른 호스트에서 이미 사용 중임을 감지함

BOOTP 중계 트랜잭션에는 다음과 같은 메시지 유형이 포함됩니다.

- RELAY-CLNT – 메시지가 DHCP 클라이언트에서 DHCP 서버로 중계되고 있음
- RELAY–SRVR – 메시지가 DHCP 서버에서 DHCP 클라이언트로 중계되고 있음

DHCP 트랜잭션 로깅은 기본적으로 사용 안함으로 설정됩니다. 사용으로 설정된 경우 DHCP 트랜잭션 로깅은 기본적으로 syslog의 `local0` 기능을 사용합니다. DHCP 트랜잭션 메시지는 syslog 심각도 레벨 **알림**으로 생성됩니다. 이 심각도 레벨이 지정되면 DHCP 트랜잭션은 다른 시스템 알림이 기록되는 파일에 기록됩니다. 하지만 `local` 기능이 사용되므로 DHCP 트랜잭션 메시지를 다른 알림과 별도로 기록할 수 있습니다. 트랜잭션 메시지를 별도로 기록하려면 `syslog.conf` 파일을 편집하여 별도 로그 파일을 지정해야 합니다. `syslog.conf` 파일에 대한 자세한 내용은 `syslog.conf`(4) 매뉴얼 페이지를 참조하십시오.

330 페이지 "DHCP 트랜잭션 로깅을 사용/사용 안함으로 설정하는 방법(DHCP 관리자)"에 설명된 대로 트랜잭션 로깅을 사용 또는 사용 안함으로 설정하고, 다른 syslog 기능(`local0` ~ `local7`)을 지정할 수 있습니다. 서버 시스템의 `syslog.conf`

파일에서 DHCP 트랜잭션 메시지를 별도의 파일에 저장하도록 syslogd에 지시할 수도 있습니다. 자세한 내용은 331 페이지 “별도의 syslog 파일에 DHCP 트랜잭션을 기록하는 방법”을 참조하십시오.

▼ 상세 정보 DHCP 로그 메시지를 생성하는 방법(DHCP 관리자)

1. **DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**

   DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

   Modify Service Options(서비스 옵션 수정) 대화 상자가 열리고 Options(옵션) 탭이 표시됩니다. 그림 15–3을 참조하십시오.

2. **Verbose Log Messages(상세 정보 로그 메시지)를 선택합니다.**

3. **Restart Server(서버 다시 시작)를 선택합니다.**

   Restart Server(서버 다시 시작) 옵션은 대화 상자의 아래쪽에 있습니다.

4. **OK(확인)를 누릅니다.**

   이 옵션을 재설정하지 않는 한 데몬은 이 세션 및 각 차후 세션에서 상세 정보 표시 모드로 실행됩니다. 상세 정보 표시 모드를 사용하면 메시지 표시에 걸리는 시간 때문에 데몬의 효율성이 떨어질 수 있습니다.

▼ 상세 정보 DHCP 로그 메시지를 생성하는 방법(명령줄)

1. **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

   DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

   역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2. **상세 정보 표시 모드를 설정하려면 다음 명령을 입력합니다.**

   ```
   # /usr/sbin/dhcpconfig -P VERBOSE=true
   ```

   상세 정보 표시 모드를 해제하지 않는 한 다음 번 DHCP 서버가 시작되면 서버는 상세 정보 표시 모드로 실행됩니다.

상세 정보 표시 모드를 해제하려면 다음 명령을 입력합니다.

```
# /usr/sbin/dhcpconfig -P VERBOSE=
```

이 명령은 VERBOSE 키워드에 값을 설정하지 않으므로 이 키워드가 서버의 구성 파일에서 제거됩니다.

상세 정보 표시 모드를 사용하면 메시지 표시에 걸리는 시간 때문에 데몬의 효율성이 떨어질 수 있습니다.

▼ DHCP 트랜잭션 로깅을 사용/사용 안함으로 설정하는 방법(DHCP 관리자)

이 절차에서는 모든 후속 DHCP 서버 세션에 대해 트랜잭션 로깅을 사용 또는 사용 안함으로 설정합니다.

1. **DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**

   DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.

2. **Log Transactions to Syslog Facility(Syslog 기능에 대한 로그 트랜잭션)를 선택합니다.**

   트랜잭션 로깅을 사용 안함으로 설정하려면 이 옵션의 선택을 취소합니다.

3. **(옵션) DHCP 트랜잭션을 기록하는 데 사용할 로컬 기능을 0에서 7 사이로 선택합니다.**

   기본적으로 DHCP 트랜잭션은 시스템 알림이 기록되는 위치에 기록됩니다. 이 위치는 `syslogd` 구성 방법에 따라 다릅니다. 다른 시스템 알림과 별도의 파일에 DHCP 트랜잭션을 기록하려면 331 페이지 "별도의 `syslog` 파일에 DHCP 트랜잭션을 기록하는 방법"을 참조하십시오.

   트랜잭션 로깅이 사용으로 설정되면 메시지 파일의 크기가 빠르게 증가할 수 있습니다.

4. **Restart Server(서버 다시 시작)를 선택합니다.**

5. **OK(확인)를 누릅니다.**

   로깅을 사용 안함으로 설정하지 않는 한 데몬은 이 세션 및 각 차후 세션에 대해 선택한 `syslog` 기능에 트랜잭션을 기록합니다.

▼ DHCP 트랜잭션 로깅을 사용/사용 안함으로 설정하는 방법(명령줄)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 단계 중 하나를 선택합니다.**

- **DHCP 트랜잭션 로깅을 사용으로 설정하려면 다음 명령을 입력합니다.**

  ```
  # /usr/sbin/dhcpconfig -P LOGGING_FACILITY=syslog-local-facility
  ```

  *syslog-local-facility*는 0에서 7 사이의 숫자입니다. 이 옵션을 생략하는 경우 0이 사용됩니다.

  기본적으로 DHCP 트랜잭션은 시스템 알림이 기록되는 위치에 기록됩니다. 이 위치는 `syslogd` 구성 방법에 따라 다릅니다. 다른 시스템 알림과 별도의 파일에 DHCP 트랜잭션을 기록하려면 331 페이지 “별도의 syslog 파일에 DHCP 트랜잭션을 기록하는 방법”을 참조하십시오.

  트랜잭션 로깅이 사용으로 설정되면 메시지 파일의 크기가 빠르게 증가할 수 있습니다.

- **DHCP 트랜잭션 로깅을 사용 안함으로 설정하려면 다음 명령을 입력합니다.**

  ```
  # /usr/sbin/dhcpconfig -P LOGGING_FACILITY=
  ```

  매개변수에 값을 제공하지 않아야 합니다.

▼ 별도의 syslog 파일에 DHCP 트랜잭션을 기록하는 방법

1 **DHCP 서버 시스템에서 수퍼 유저 또는 동등한 역할의 사용자로 로그인합니다.**

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

DHCP 관리 프로파일에 지정된 역할로는 이 작업을 수행하지 못할 수 있습니다. 역할에 `syslog` 파일을 편집할 수 있는 권한이 있어야 합니다.

2 **서버 시스템의 `/etc/syslog.conf` 파일을 편집하여 다음 형식의 행을 추가합니다.**

`local`*n*`.notice` *path-to-logfile*

*n*은 트랜잭션 로깅에 대해 지정한 `syslog` 기능 번호이고 *path-to-logfile*은 트랜잭션 기록에 사용할 파일의 전체 경로입니다.

예를 들어 다음 행을 추가할 수 있습니다.

```
local0.notice /var/log/dhcpsrvc
```

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

DHCP 서버에 의한 동적 DNS 업데이트를 사용으로 설정

DNS는 인터넷에 대한 이름-주소 및 주소-이름 서비스를 제공합니다. DNS 매핑이 지정되면 호스트 이름 또는 IP 주소를 통해 시스템에 도달할 수 있습니다. 해당 도메인 외부에서도 시스템에 도달할 수 있습니다.

DHCP 서비스는 두 가지 방법으로 DNS를 사용할 수 있습니다.

- DHCP 서버는 서버가 클라이언트에 지정하는 IP 주소에 매핑된 호스트 이름을 조회할 수 있습니다. 그런 다음 서버는 클라이언트의 기타 구성 정보와 함께 클라이언트의 호스트 이름을 반환합니다.
- DHCP 서버가 DNS를 업데이트하도록 구성된 경우 DHCP 서버는 클라이언트를 대신해서 DNS 매핑을 수행하려고 시도할 수 있습니다. 클라이언트는 DHCP 서비스를 요청할 때 자체 호스트 이름을 제공할 수 있습니다. DNS를 업데이트하도록 구성된 경우 DHCP 서버는 클라이언트가 제시한 호스트 이름으로 DNS를 업데이트하려고 시도합니다. DNS 업데이트가 성공하면 DHCP 서버는 요청된 호스트 이름을 클라이언트에 반환합니다. DNS 업데이트가 성공하지 못하면 DHCP 서버는 다른 호스트 이름을 클라이언트에 반환합니다.

자체 호스트 이름을 제공하는 DHCP 클라이언트에 대해 DNS 서비스를 업데이트하도록 DHCP 서비스를 설정할 수 있습니다. DNS 업데이트 기능이 작동하려면 DNS 서버, DHCP 서버 및 DHCP 클라이언트가 올바르게 설정되어야 합니다. 또한 요청된 호스트 이름을 도메인의 다른 시스템에서 사용하고 있지 않아야 합니다.

DHCP 서버의 DNS 업데이트 기능은 다음 조건이 성립하는 경우에만 작동합니다.

- DNS 서버가 RFC 2136을 지원합니다.
- DHCP 서버 시스템이나 DNS 서버 시스템에서 DNS 소프트웨어가 BIND v8.2.2, 패치 레벨 5 이상을 기반으로 합니다.
- DNS 서버가 DHCP 서버의 동적 DNS 업데이트를 허용하도록 구성되어 있습니다.
- DHCP 서버가 동적 DNS 업데이트를 수행하도록 구성되어 있습니다.
- DHCP 서버에서 DHCP 클라이언트의 네트워크에 대해 DNS 지원이 구성되어 있습니다.

- DHCP 요청 메시지에 요청된 호스트 이름을 제공하도록 DHCP 클라이언트가 구성되어 있습니다.
- 요청된 호스트 이름이 DHCP 소유 주소에 해당합니다. 또한 이 호스트 이름은 해당하는 주소를 가질 수 없습니다.

▼ DHCP 클라이언트에 대한 동적 DNS 업데이트를 사용으로 설정하는 방법

주 – 동적 DNS 업데이트는 **보안 위험**이 될 수 있습니다.

기본적으로 Oracle Solaris DNS 데몬(`in.named`)은 동적 업데이트를 허용하지 않습니다. 동적 DNS 업데이트에 대한 권한은 DNS 서버 시스템의 `named.conf` 구성 파일에서 부여됩니다. 다른 보안은 제공되지 않습니다. 사용자에게 이 기능이 주는 편리함과 동적 DNS 업데이트를 사용할 때 발생하는 보안 위험을 신중하게 비교해야 합니다.

---

1 DNS 서버에서 수퍼 유저로 `/etc/named.conf` 파일을 편집합니다.

2 `named.conf` 파일에서 해당 도메인에 대한 `zone`섹션을 찾습니다.

3 DHCP 서버의 IP 주소를 `allow-update` 키워드에 추가합니다.

`allow-update` 키워드가 존재하지 않으면 삽입합니다.

예를 들어 DHCP 서버가 주소 `10.0.0.1`과 `10.0.0.2`에 있는 경우 `dhcp.domain.com` 영역에 대해 `named.conf` 파일을 다음과 같이 수정해야 합니다.

```
zone "dhcp.domain.com" in {
             type master;
             file "db.dhcp";
             allow-update { 10.0.0.1; 10.0.0.2; };
};

zone "10.IN-ADDR.ARPA" in {
             type master;
             file "db.10";
             allow-update { 10.0.0.1; 10.0.0.2; };
};
```

DHCP 서버가 DNS 서버에서 A 및 PTR 레코드를 모두 업데이트하도록 허용하려면 두 영역 모두에 대해 `allow-update`를 사용으로 설정해야 합니다.

4 DHCP 서버에서 DHCP 관리자를 시작합니다.

```
# /usr/sadm/admin/bin/dhcpmgr &
```

자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

5 **Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**
Modify Service Options(서비스 옵션 수정) 대화 상자가 열립니다.

6 **Update DNS Host Information Upon Client Request(클라이언트 요청 시 DNS 호스트 정보 업데이트)를 선택합니다.**

7 **시간 초과가 발생하기 전에 DNS 서버의 응답을 기다리는 시간(초 단위)을 지정한 후 OK(확인)를 누릅니다.**
대개 기본값인 15초면 적당합니다. 시간 초과 문제가 있는 경우 나중에 값을 늘릴 수 있습니다.

8 **Macros(매크로) 탭을 눌러 올바른 DNS 도메인이 지정되었는지 확인합니다.**
`DNSdmain` 옵션은 올바른 도메인 이름을 포함하여 동적 DNS 업데이트 지원을 예상하는 모든 클라이언트에 전달되어야 합니다. 기본적으로 `DNSdmain`은 각 IP 주소에 바인드되는 구성 매크로로 사용되는 서버 매크로에 지정됩니다.

9 **DHCP 서비스를 요청할 때 해당 호스트 이름을 지정하도록 DHCP 클라이언트를 설정합니다.**
DHCP 클라이언트를 사용하는 경우 412 페이지 "DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법"을 참조하십시오. 클라이언트가 DHCP 클라이언트가 아닌 경우 호스트 이름을 지정하는 방법을 보려면 해당 클라이언트의 설명서를 참조하십시오.

클라이언트 호스트 이름 등록

DHCP 서버가 DHCP 서비스에 배치된 IP 주소에 대해 호스트 이름을 생성하도록 설정하면 DHCP 서버는 해당 호스트 이름을 NIS+, `/etc/inet/hosts` 또는 DNS 이름 서비스에 등록할 수 있습니다. NIS는 프로그램에서 NIS 맵을 업데이트 및 전파하도록 허용하는 프로토콜을 제공하지 않기 때문에 NIS에서는 호스트 이름을 등록할 수 없습니다.

주 – DNS 서버와 DHCP 서버가 같은 시스템에서 실행되고 있는 경우에만 DHCP 서버는 생성된 호스트 이름으로 DNS를 업데이트할 수 있습니다.

DHCP 클라이언트가 호스트 이름을 제공하고 DNS 서버가 DHCP 서버의 동적 업데이트를 허용하도록 구성되는 경우 DHCP 서버는 클라이언트를 대신하여 DNS를 업데이트할 수 있습니다. 동적 업데이트는 DNS와 DHCP 서버가 서로 다른 시스템에서 실행되는 경우에도 수행될 수 있습니다. 이 기능을 사용으로 설정하는 방법은 332 페이지 "DHCP 서버에 의한 동적 DNS 업데이트를 사용으로 설정"을 참조하십시오.

다음 표는 다양한 이름 서비스의 DHCP 클라이언트 시스템에 대한 클라이언트 호스트 이름 등록을 요약하여 보여줍니다.

표 15–2 이름 서비스의 클라이언트 호스트 이름 등록

	호스트 이름을 등록하는 주체	
이름 서비스	**DHCP 생성 호스트 이름**	**DHCP 클라이언트 제공 호스트 이름**
NIS	NIS 관리자	NIS 관리자
NIS+	DHCP 도구	DHCP 도구
`/etc/hosts`	DHCP 도구	DHCP 도구
DNS	DNS 서버가 DHCP 서버와 같은 시스템에서 실행되는 경우 DHCP 도구	동적 DNS 업데이트를 수행하도록 구성된 경우 DHCP 서버
	DNS 서버가 다른 시스템에서 실행되는 경우 DNS 관리자	DHCP 서버가 동적 DNS 업데이트를 수행하도록 구성되지 않은 경우 DNS 관리자

412 페이지 “DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법”에 설명된 대로 DHCP 클라이언트는 적절히 구성된 경우 DHCP 요청에서 특정 호스트 이름을 요청할 수 있습니다. 다른 DHCP 클라이언트의 경우 이 기능이 지원되는지 확인하려면 공급업체 설명서를 참조하십시오.

DHCP 서버에 대한 성능 옵션 사용자 정의

DHCP 서버의 성능에 영향을 미치는 옵션을 변경할 수 있습니다. 이러한 옵션은 다음 표에 설명되어 있습니다.

표 15–3 DHCP 서버 성능에 영향을 미치는 옵션

서버 옵션	**설명**	**키워드**
최대 BOOTP 중계 에이전트 홉 수	요청이 지정된 수보다 많은 BOOTP 중계 에이전트를 거쳐 이동한 경우 이 요청은 삭제됩니다. 최대 중계 에이전트 홉 수의 기본값은 4입니다. 대부분의 네트워크에서는 이 숫자로 충분합니다. DHCP 요청이 DHCP 서버에 도달하기 전에 여러 개의 BOOTP 중계 에이전트를 거치는 경우 네트워크에 4개보다 많은 홉이 필요할 수 있습니다.	`RELAY_HOPS=`*integer*
중복 주소를 감지합니다.	기본적으로 서버는 IP 주소를 클라이언트에 제공하기 전에 해당 주소를 핑합니다. 핑에 대한 응답이 없으면 주소가 이미 사용 중이 아님이 확인됩니다. 서버가 제공하는 데 걸리는 시간을 줄이기 위해 이 기능을 사용 안함으로 설정할 수 있습니다. 하지만 이 기능을 사용 안함으로 설정하면 중복 IP 주소를 사용하게 될 위험이 있습니다.	`ICMP_VERIFY=TRUE/FALSE`

표 15–3 DHCP 서버 성능에 영향을 미치는 옵션 *(계속)*

서버 옵션	설명	키워드
지정된 간격으로 dhcptab을 자동으로 다시 로드합니다.	서버가 지정한 간격(분)으로 dhcptab을 자동으로 읽도록 설정할 수 있습니다. 네트워크 구성 정보가 자주 변경되지 않고 DHCP 서버를 여러 대 가지고 있지 않은 경우에는 dhcptab을 자동으로 다시 로드할 필요가 없습니다. 또한 DHCP 관리자에서는 데이터를 변경한 후 서버가 dhcptab을 다시 로드하는 옵션을 선택할 수 있습니다.	RESCAN_INTERVAL=*min*
지정된 간격 동안 IP 주소 제공을 캐시합니다.	서버가 클라이언트에 IP 주소를 제공한 후 해당 제공은 캐시됩니다. 제공이 캐시되는 동안 서버는 주소를 다시 제공하지 않습니다. 제공이 캐시되는 시간(초 단위)을 변경할 수 있습니다. 기본값은 10초입니다. 느린 네트워크에서는 제공 시간을 늘려야 할 수 있습니다.	OFFER_CACHE_TIMEOUT=*sec*

다음 절차에서는 이러한 옵션을 변경하는 방법을 설명합니다.

▼ DHCP 성능 옵션을 사용자 정의하는 방법(DHCP 관리자)

1 **DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **원하는 옵션을 변경합니다.**

옵션에 대한 자세한 내용은 표 15–3을 참조하십시오.

3 **Restart Server(서버 다시 시작)를 선택합니다.**

4 **OK(확인)를 누릅니다.**

▼ DHCP 성능 옵션을 사용자 정의하는 방법(명령줄)

이 절차로 옵션을 변경하는 경우 변경된 옵션을 사용하려면 DHCP 서버를 다시 시작해야 합니다.

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 하나 이상의 옵션 수정:

```
# /usr/sbin/dhcpconfig -P keyword=value,keyword=value...
```

*keyword=value*는 다음 키워드 중 하나일 수 있습니다.

`RELAY_HOPS=`*integer*	데몬이 DHCP 또는 BOOTP 데이터그램을 삭제하기 전에 발생할 수 있는 최대 중계 에이전트 홉 수를 지정합니다.
`ICMP_VERIFY=TRUE/FALSE`	자동 중복 IP 주소 감지를 사용 또는 사용 안함으로 설정합니다. 이 키워드를 `FALSE`로 설정하지 않는 것이 좋습니다.
`RESCAN_INTERVAL=`*minutes*	DHCP 서버가 `dhcptab` 정보를 자동으로 다시 읽도록 예약하는 데 사용할 간격을 분 단위로 지정합니다.
`OFFER_CACHE_TIMEOUT=`*seconds*	DHCP 서버가 DHCP 클라이언트 검색까지 확장된 제공을 캐시해야 하는 시간(초)을 지정합니다. 기본 설정은 10초입니다.

예 15–1 DHCP 성능 옵션 설정

다음은 이 모든 명령 옵션을 지정하는 방법의 예입니다.

```
# dhcpconfig -P RELAY_HOPS=2,ICMP_VERIFY=TRUE,\
RESCAN_INTERVAL=30,OFFER_CACHE_TIMEOUT=20
```

DHCP 네트워크 추가, 수정 및 제거(작업 맵)

DHCP 서버를 구성하는 경우 DHCP 서비스를 사용하기 위해 적어도 하나의 네트워크를 구성해야 합니다. 언제든지 네트워크를 추가할 수 있습니다.

다음 표는 초기 구성 후 DHCP 네트워크 작업 시 수행할 수 있는 추가 작업을 설명하는 맵입니다. 작업 맵에는 작업을 수행하는 절차에 대한 링크가 포함되어 있습니다.

작업	설명	수행 방법
서버 네트워크 인터페이스에서 DHCP 서비스를 사용 또는 사용 안함으로 설정합니다.	기본 동작은 DHCP 요청에 대해 모든 네트워크 인터페이스를 모니터하는 것입니다. 일부 인터페이스에서는 DHCP 요청을 수락하지 않도록 하려면 모니터되는 인터페이스 목록에서 인터페이스를 제거합니다.	339 페이지 “DHCP 모니터링에 대해 네트워크 인터페이스를 지정하는 방법(DHCP 관리자)”
DHCP 서비스에 새 네트워크를 추가합니다.	네트워크의 IP 주소를 관리하기 위해 네트워크를 DHCP 관리 아래에 둡니다.	341 페이지 “DHCP 네트워크를 추가하는 방법(DHCP 관리자)” 342 페이지 “DHCP 네트워크를 추가하는 방법(dhcpconfig)”
DHCP 관리 네트워크의 매개변수를 변경합니다.	특정 네트워크의 클라이언트에 전달되는 정보를 수정합니다.	343 페이지 “DHCP 네트워크 구성을 수정하는 방법(DHCP 관리자)” 344 페이지 “DHCP 네트워크 구성을 수정하는 방법(dhtadm)”
DHCP 서비스에서 네트워크를 삭제합니다.	네트워크의 IP 주소가 더 이상 DHCP에 의해 관리되지 않도록 네트워크를 제거합니다.	346 페이지 “DHCP 네트워크를 제거하는 방법(DHCP 관리자)” 347 페이지 “DHCP 네트워크를 제거하는 방법(pntadm)”

DHCP 모니터링을 위한 네트워크 인터페이스 지정

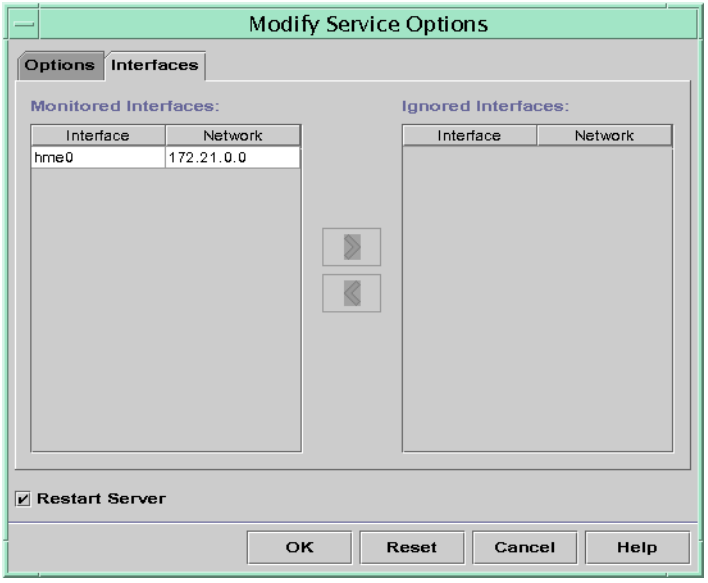
기본적으로 dhcpconfig 및 DHCP 관리자의 구성 마법사는 모두 DHCP 서버가 모든 서버 시스템의 네트워크 인터페이스를 모니터하도록 구성합니다. 서버 시스템에 새 네트워크 인터페이스를 추가하는 경우 DHCP 서버는 시스템이 부트되면 새 인터페이스를 자동으로 모니터합니다. 그러면 이 네트워크 인터페이스를 통해 모니터할 네트워크를 추가할 수 있습니다.

모니터할 네트워크 인터페이스 및 무시할 인터페이스도 지정할 수 있습니다. 네트워크에서 DHCP 서비스를 제공하지 않는 경우 해당 인터페이스를 무시할 수 있습니다.

모든 인터페이스를 무시하도록 지정한 다음 새 인터페이스를 설치하면 DHCP 서버는 새 인터페이스를 무시합니다. 새 인터페이스를 서버의 모니터되는 인터페이스 목록에 추가해야 합니다. DHCP 관리자 또는 dhcpconfig 유틸리티를 사용하여 인터페이스를 지정할 수 있습니다.

이 절에는 DHCP가 모니터하거나 무시할 네트워크 인터페이스를 지정하는 절차가 포함되어 있습니다. DHCP 관리자 절차에서는 DHCP 관리자의 Modify Service Options(서비스 옵션 수정) 대화 상자에 있는 Interfaces(인터페이스) 탭을 사용합니다. 다음 그림을 참조하십시오.

그림 15–4 DHCP 관리자의 Modify Service Options(서비스 옵션 수정) 대화 상자에 있는 Interfaces(인터페이스) 탭



▼ DHCP 모니터링에 대해 네트워크 인터페이스를 지정하는 방법(DHCP 관리자)

1 **DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**

Modify Service Options(서비스 옵션 수정) 대화 상자가 표시됩니다.

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **Interfaces(인터페이스) 탭을 선택합니다.**

3 **해당 네트워크 인터페이스를 선택합니다.**

4 **화살표 버튼을 눌러 인터페이스를 해당 목록으로 이동합니다.**

예를 들어 인터페이스를 무시하려면 Monitored Interfaces(모니터된 인터페이스) 목록에서 인터페이스를 선택한 다음 오른쪽 화살표 버튼을 누릅니다. 그러면 해당 인터페이스가 Ignored Interfaces(무시된 인터페이스) 목록에 표시됩니다.

5 **Restart Server(서버 다시 시작)를 선택하고 OK(확인)를 누릅니다.**

변경된 사항은 재부트를 해도 유지됩니다.

▼ DHCP 모니터링에 대해 네트워크 인터페이스를 지정하는 방법(dhcpconfig)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **DHCP 서버 시스템에서 다음 명령을 입력합니다.**

```
# /usr/sbin/dhcpconfig -P INTERFACES=int,int,...
```

*int, int,...*는 모니터할 인터페이스 목록입니다. 인터페이스 이름은 쉼표로 구분해야 합니다.

예를 들어 다음 명령을 사용하여 `ge0` 및 `ge1`만 모니터할 수 있습니다.

```
#/usr/sbin/dhcpconfig -P INTERFACES=ge0,ge1
```

무시할 인터페이스는 `dhcpconfig` 명령줄에서 생략해야 합니다.

이 명령으로 변경한 사항은 재부트해도 유지됩니다.

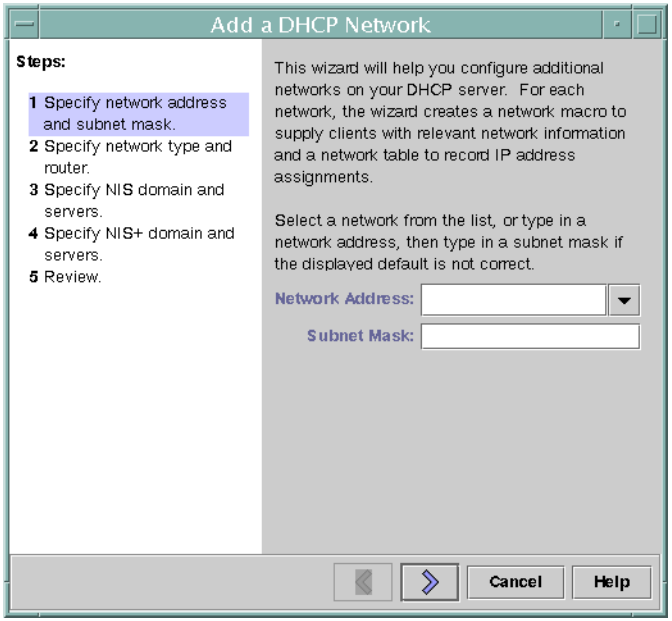
DHCP 네트워크 추가

DHCP 관리자를 사용하여 서버를 구성하는 경우 첫번째 네트워크도 동시에 구성됩니다. 첫번째 네트워크는 대개 서버 시스템 기본 인터페이스의 로컬 네트워크입니다. 추가 네트워크를 구성하려면 DHCP 관리자의 DHCP 네트워크 마법사를 사용하십시오.

`dhcpconfig -D` 명령을 사용하여 서버를 구성하는 경우 DHCP 서비스를 사용할 모든 네트워크를 개별적으로 구성해야 합니다. 자세한 내용은 342 페이지 "DHCP 네트워크를 추가하는 방법(dhcpconfig)"을 참조하십시오.

다음 그림은 DHCP 관리자의 DHCP 네트워크 마법사의 초기 대화 상자를 보여줍니다.

그림 15–5 DHCP 관리자의 네트워크 마법사



새 네트워크를 구성할 때 DHCP 관리자는 다음 구성 요소를 만듭니다.

- 데이터 저장소의 네트워크 테이블. 새 네트워크는 DHCP 관리자의 Addresses(주소) 탭에 있는 네트워크 목록에 표시됩니다.
- 이 네트워크에 있는 클라이언트에 필요한 정보를 포함하는 네트워크 매크로. 네트워크 매크로의 이름은 네트워크의 IP 주소와 일치합니다. 네트워크 매크로는 데이터 저장소의 `dhcptab` 테이블에 추가됩니다.

▼ DHCP 네트워크를 추가하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Addresses(주소) 탭을 누릅니다.**

DHCP 서비스에 대해 이미 구성된 모든 네트워크가 나열됩니다.

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **Edit(편집) 메뉴에서 Network Wizard(네트워크 마법사)를 선택합니다.**

3 **옵션을 선택하거나 요청된 정보를 입력합니다. 계획 단계에서 결정한 사항을 사용하여 지정할 정보를 결정합니다.**

계획은 304 페이지 “원격 네트워크의 DHCP 구성 계획”에 설명되어 있습니다.

마법사 사용에 어려움이 있다면 마법사 창에서 Help(도움말)를 누릅니다. 웹 브라우저에 DHCP 네트워크 마법사에 대한 도움말이 표시됩니다.

4 **요청된 정보 지정을 마치면 Finish(마침)를 눌러 네트워크 구성을 완료합니다.**

네트워크 마법사는 빈 네트워크 테이블을 만드는데, 이 테이블은 창의 왼쪽에 표시됩니다.

네트워크 마법사는 네트워크의 IP 주소와 이름이 일치하는 네트워크 매크로도 만듭니다.

5 **(옵션) 매크로의 내용을 보려면 Macros(매크로) 탭을 선택하고 보려는 네트워크 매크로를 선택합니다.**

마법사에서 제공한 정보가 네트워크 매크로의 옵션 값으로 삽입되었는지 확인할 수 있습니다.

참조 네트워크의 IP 주소를 DHCP로 관리하기 전에 먼저 네트워크 주소를 추가해야 합니다. 자세한 내용은 354 페이지 "DHCP 서비스에 IP 주소 추가"를 참조하십시오.

네트워크 테이블을 비워두어도 DHCP 서버는 클라이언트에게 구성 정보를 제공할 수 있습니다. 자세한 내용은 387 페이지 "정보만 수신하도록 DHCP 클라이언트 설정(작업 맵)"을 참조하십시오.

▼ DHCP 네트워크를 추가하는 방법(dhcpconfig)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **DHCP 서버 시스템에서 다음 명령을 입력합니다.**

```
# /usr/sbin/dhcpconfig -N network-address
```

*network-address*는 DHCP 서비스에 추가할 네트워크의 IP 주소입니다. `-N` 옵션과 함께 사용할 수 있는 하위 옵션은 `dhcpconfig`(1M) 매뉴얼 페이지를 참조하십시오.

하위 옵션을 사용하지 않는 경우 `dhcpconfig`는 네트워크 파일을 사용하여 네트워크에 대한 정보를 가져옵니다.

참조 네트워크의 IP 주소를 DHCP로 관리하기 전에 먼저 네트워크 주소를 추가해야 합니다. 자세한 내용은 354 페이지 "DHCP 서비스에 IP 주소 추가"를 참조하십시오.

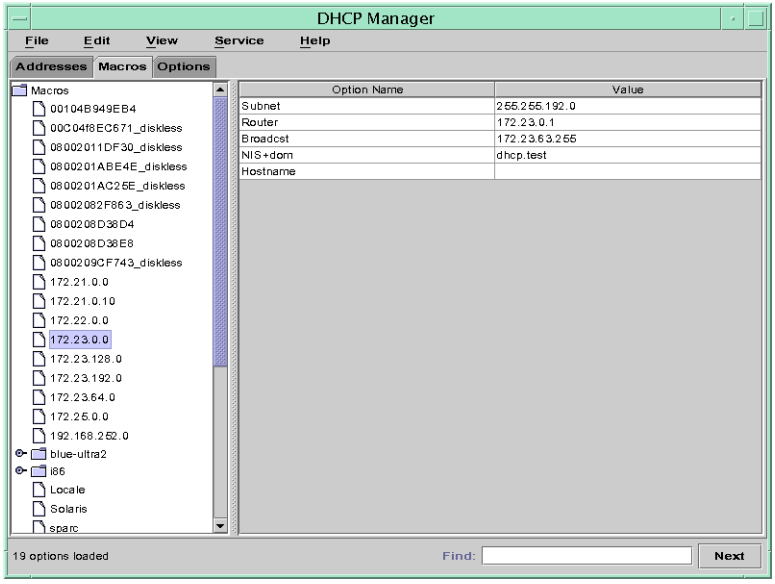
네트워크 테이블을 비워두어도 DHCP 서버는 클라이언트에게 구성 정보를 제공할 수 있습니다. 자세한 내용은 387 페이지 “정보만 수신하도록 DHCP 클라이언트 설정(작업 맵)”을 참조하십시오.

DHCP 네트워크 구성 수정

DHCP 서비스에 네트워크를 추가한 후 원래 제공한 구성 정보를 수정할 수 있습니다. 구성 정보는 네트워크에서 클라이언트에 정보를 전달하는 데 사용되는 네트워크 매크로에 저장됩니다. 네트워크 구성을 변경하려면 네트워크 매크로를 수정해야 합니다.

다음 그림은 DHCP 관리자의 Macros(매크로) 탭을 보여줍니다.

그림 15–6 DHCP 관리자의 Macros(매크로) 탭



▼ DHCP 네트워크 구성을 수정하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

이 DHCP 서버에 대해 정의된 모든 매크로가 왼쪽 창에 나열됩니다.

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **변경할 네트워크 구성과 이름이 일치하는 네트워크 매크로를 선택합니다.**

네트워크 매크로 이름은 네트워크 IP 주소입니다.

3 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**

Macro Properties(매크로 등록 정보) 대화 상자에는 매크로에 포함된 옵션 테이블이 표시됩니다.

4 **수정할 옵션을 선택합니다.**

옵션 이름 및 해당 값은 대화 상자 위쪽의 텍스트 필드에 표시됩니다.

5 **(옵션) 옵션 이름을 수정하거나 Select(선택) 버튼을 선택하여 옵션 이름 목록을 표시합니다.**

Select Option(옵션 선택) 대화 상자에는 모든 DHCP 표준 옵션 목록과 각 옵션에 대한 간단한 설명이 표시됩니다.

6 **(옵션) Select Option(옵션 선택) 대화 상자에서 옵션 이름을 선택하고 OK(확인)를 누릅니다.**

Option Name(옵션 이름) 필드에 새 옵션 이름이 표시됩니다.

7 **새 옵션 값을 입력하고 Modify(수정)를 누릅니다.**

8 **(옵션) 대화 상자에서 Select(선택)를 선택하여 네트워크 매크로에 옵션을 추가할 수도 있습니다.**

매크로 수정에 대한 일반적인 정보는 368 페이지 “DHCP 매크로 수정”을 참조하십시오.

9 **Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택하고 OK(확인)를 누릅니다.**

이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

▼ DHCP 네트워크 구성을 수정하는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **네트워크의 모든 클라이언트에 대한 정보가 포함된 매크로를 파악합니다.**

네트워크 매크로 이름은 네트워크 IP 주소와 일치합니다.

이 정보가 포함된 매크로를 모르는 경우 `dhtadm -P` 명령을 통해 `dhcptab` 테이블을 표시하여 모든 매크로를 나열할 수 있습니다.

3 **옵션 값을 변경하려면 다음 형식으로 명령을 입력합니다.**

```
# dhtadm -M -m macro-name -e 'symbol=value' -g
```

`dhtadm` 명령줄 옵션에 대한 자세한 내용은 `dhtadm(1M)` 매뉴얼 페이지를 참조하십시오.

예 15–2 dhtadm 명령을 사용하여 DHCP 매크로 수정

예를 들어 `10.25.62.0` 매크로의 임대 시간을 57600초로 변경하고 NIS 도메인을 `sem.example.com`으로 변경하려면 다음 명령을 입력합니다.

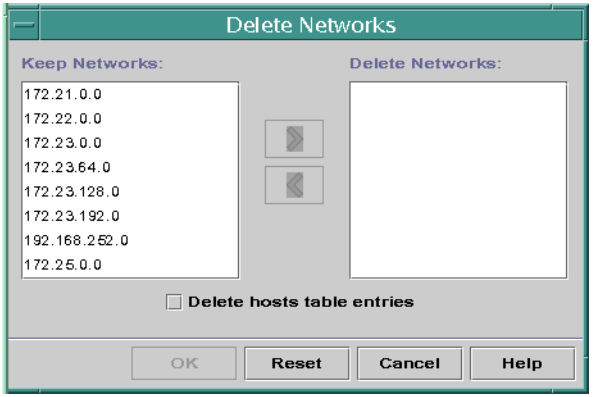
```
# dhtadm -M -m 10.25.62.0 -e 'LeaseTim=57600' -g

# dhtadm -M -m 10.25.62.0 -e 'NISdmain=sem.example.com' -g
```

`-g` 옵션을 지정하면 DHCP 데몬이 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

DHCP 네트워크 제거

DHCP 관리자를 사용하면 여러 개의 네트워크를 한 번에 제거할 수 있습니다. 해당 네트워크에서 DHCP 관리 IP 주소와 연관된 호스트 테이블 항목을 자동으로 제거할 수도 있습니다. 다음 그림은 DHCP 관리자의 Delete Networks(네트워크 삭제) 대화 상자를 보여줍니다.

그림 15–7 DHCP 관리자의 Delete Networks(네트워크 삭제) 대화 상자



pntadm 명령에서는 네트워크를 삭제하기 전에 해당 네트워크에서 각 IP 주소 항목을 삭제해야 합니다. 한 번에 하나의 네트워크만 삭제할 수 있습니다.

▼ DHCP 네트워크를 제거하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **Edit(편집) 메뉴에서 Delete Networks(네트워크 삭제)를 선택합니다.**

Delete Networks(네트워크 삭제) 대화 상자가 열립니다.

3 **Keep Networks(네트워크 유지) 목록에서 삭제할 네트워크를 선택합니다.**

Ctrl 키를 누른 상태로 마우스를 눌러 네트워크를 여러 개 선택할 수 있습니다. Shift 키를 누른 상태로 마우스를 눌러 네트워크 범위를 선택할 수 있습니다.

4 **오른쪽 화살표 버튼을 눌러 선택한 네트워크를 Delete Networks(네트워크 삭제) 목록으로 이동합니다.**

5 **이 네트워크의 DHCP 주소에 대한 호스트 테이블 항목을 제거하려면 Delete Host Table Entries(호스트 테이블 항목 삭제)를 선택합니다.**

호스트 테이블 항목을 삭제해도 이 주소에 대한 DNS 서버의 호스트 등록은 삭제되지 않습니다. 항목은 로컬 이름 서비스에서만 삭제됩니다.

6 **OK(확인)를 누릅니다.**

▼ DHCP 네트워크를 제거하는 방법(pntadm)

이 절차에서는 네트워크를 제거하기 전에 DHCP 네트워크 테이블에서 네트워크의 IP 주소를 삭제합니다. 주소는 호스트 이름이 `hosts` 파일 또는 데이터베이스에서 제거되도록 보장하기 위해 삭제됩니다.

1. **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

   DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

   역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2. **이름 서비스에서 IP 주소 및 해당 호스트 이름을 제거하려면 다음 형식으로 명령을 입력합니다.**

   # pntadm -D -y *IP-address*

   예를 들어 IP 주소 `10.25.52.1`을 제거하려면 다음 명령을 입력합니다.

   # pntadm -D -y 10.25.52.1

   `-y` 옵션은 호스트 이름을 삭제하도록 지정합니다.

3. **네트워크의 각 주소에 대해 `pntadm -D -y` 명령을 반복합니다.**

   대량으로 주소를 삭제하는 경우 `pntadm` 명령을 실행하는 스크립트를 만들 수 있습니다.

4. **모든 주소가 삭제되면 다음 명령을 입력하여 DHCP 서비스에서 네트워크를 삭제합니다.**

   # pntadm -R *network-IP-address*

   예를 들어 `10.25.52.0` 네트워크를 제거하려면 다음 명령을 입력합니다.

   # pntadm -R 10.25.52.0

   `pntadm` 유틸리티 사용에 대한 자세한 내용은 `pntadm`(1M) 매뉴얼 페이지를 참조하십시오.

DHCP 서비스로 BOOTP 클라이언트 지원(작업 맵)

DHCP 서버에서 BOOTP 클라이언트를 지원하려면 BOOTP와 호환되도록 DHCP 서버를 설정해야 합니다. DHCP를 사용할 수 있는 BOOTP 클라이언트를 지정하려면 DHCP 서버의 네트워크 테이블에 BOOTP 클라이언트를 등록합니다. 또는 BOOTP 클라이언트에 자동 할당되도록 다수의 IP 주소를 예약할 수 있습니다.

주 – 주소에 영구 임대를 명시적으로 지정하는지 여부와 관계없이 BOOTP 주소는 영구적으로 지정됩니다.

다음 표에서는 BOOTP 클라이언트를 지원하기 위해 수행해야 할 수 있는 작업에 대해 설명합니다. 작업 맵에는 작업을 수행하는 데 사용되는 절차에 대한 링크가 포함되어 있습니다.

작업	설명	수행 방법
자동 BOOTP 지원을 설정합니다.	DHCP 관리 네트워크 또는 DHCP 관리 네트워크에 중계 에이전트로 연결된 네트워크에서 모든 BOOTP 클라이언트에 IP 주소를 제공합니다. BOOTP 클라이언트가 배타적으로 사용하는 주소 풀을 예약해야 합니다. 서버가 많은 BOOTP 클라이언트를 지원해야 하는 경우 이 옵션이 더 유용할 수 있습니다.	348 페이지 "모든 BOOTP 클라이언트에 대한 지원을 설정하는 방법(DHCP 관리자)"
수동 BOOTP 지원을 설정합니다.	DHCP 서비스에 수동으로 등록된 BOOTP 클라이언트에 대해서만 IP 주소를 제공합니다. 이 옵션을 사용하려면 클라이언트의 ID를 BOOTP 클라이언트용으로 표시된 특정 IP 주소에 바인드해야 합니다. 이 옵션은 BOOTP 클라이언트 수가 적거나 DHCP 서버를 사용할 수 있는 BOOTP 클라이언트를 제한하고 싶을 때 유용합니다.	349 페이지 "등록된 BOOTP 클라이언트에 대한 지원을 설정하는 방법(DHCP 관리자)"

▼ 모든 BOOTP 클라이언트에 대한 지원을 설정하는 방법(DHCP 관리자)

1. **DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**

   Modify Service Options(서비스 옵션 수정) 대화 상자가 열립니다.

   DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.

2. **대화 상자의 BOOTP Compatibility(BOOTP 호환성) 섹션에서 Automatic(자동)을 선택합니다.**

3. **Restart Server(서버 다시 시작)를 선택하고 OK(확인)를 누릅니다.**

4 **Addresses(주소) 탭을 선택합니다.**

5 **BOOTP 클라이언트용으로 예약할 주소를 선택합니다.**
첫번째 주소를 누르고 Shift 키를 누른 상태로 마지막 주소를 눌러 주소 범위를 선택합니다. Ctrl 키를 누른 상태로 각 주소를 눌러 연속되지 않은 여러 개의 주소를 선택합니다.

6 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**
Modify Multiple Addresses(복수 주소 수정) 대화 상자가 열립니다.

7 **BOOTP 섹션에서 Assign All Addresses Only to BOOTP Clients(BOOTP 클라이언트에만 모든 주소 할당)를 선택합니다.**
다른 모든 옵션은 Keep Current Settings(현재 설정 유지)로 설정되어야 합니다.

8 **OK(확인)를 누릅니다.**
이제 모든 BOOTP 클라이언트가 이 DHCP 서버에서 주소를 얻을 수 있습니다.

▼ 등록된 BOOTP 클라이언트에 대한 지원을 설정하는 방법(DHCP 관리자)

1 **DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.**
Modify Service Options(서비스 옵션 수정) 대화 상자가 열립니다.
DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **대화 상자의 BOOTP Compatibility(BOOTP 호환성) 섹션에서 Manual(수동)을 선택합니다.**

3 **Restart Server(서버 다시 시작)를 선택하고 OK(확인)를 누릅니다.**

4 **Addresses(주소) 탭을 선택합니다.**

5 **특정 BOOTP 클라이언트에 지정할 주소를 선택합니다.**

6 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**
Address Properties(주소 등록 정보) 대화 상자가 열립니다.

7 **Address Properties(주소 등록 정보) 대화 상자에서 Lease(임대) 탭을 선택합니다.**

8 **Client ID(클라이언트 ID) 필드에 클라이언트의 식별자를 입력합니다.**

이더넷 네트워크의 BOOTP Oracle Solaris 클라이언트의 경우 클라이언트 ID는 클라이언트의 16진수 이더넷 주소에서 파생된 문자열입니다. 클라이언트 ID에는 이더넷의 ARP(Address Resolution Protocol) 유형(01)을 나타내는 접두어가 포함됩니다. 예를 들어 이더넷 주소가 `8:0:20:94:12:1e`인 BOOTP 클라이언트는 클라이언트 ID `0108002094121E`를 사용합니다.

참고 – Oracle Solaris 클라이언트 시스템의 수퍼 유저로 다음 명령을 입력하여 인터페이스의 이더넷 주소를 얻습니다.

```
# ifconfig -a
```

---

9 **Reserved(예약)를 선택하여 이 클라이언트에 대해 IP 주소를 예약합니다.**

10 **Assign Only to BOOTP Clients(BOOTP 클라이언트에만 할당)를 선택하고 OK(확인)를 누릅니다.**

Addresses(주소) 탭에서 BOOTP는 Status(상태) 필드에 표시되고 지정한 클라이언트 ID는 Client ID(클라이언트 ID) 필드에 나열됩니다.

DHCP 서비스에서 IP 주소 작업(작업 맵)

DHCP 관리자 또는 `pntadm` 명령을 사용하여 IP 주소를 추가하고, 주소 등록 정보를 수정하고, DHCP 서비스에서 주소를 제거할 수 있습니다. IP 주소를 사용하기 전에 표 15–4를 참조하여 IP 주소 등록 정보에 대한 내용을 파악해야 합니다. 이 표는 DHCP 관리자 및 `pntadm` 사용자에 필요한 정보를 제공합니다.

주 – 표 15–4에는 IP 주소를 추가 및 수정하는 동안 `pntadm`을 사용하여 IP 주소 등록 정보를 지정하는 예가 들어 있습니다. `pntadm`에 대한 자세한 내용은 `pntadm(1M)` 매뉴얼 페이지를 참조하십시오.

다음 작업 맵은 IP 주소를 추가, 수정 또는 제거하기 위해 수행해야 하는 작업을 나열합니다. 작업 맵에는 작업을 수행하는 데 사용되는 절차에 대한 링크도 포함되어 있습니다.

작업	설명	수행 방법
DHCP 서비스에 단일 또는 복수 IP 주소를 추가합니다.	DHCP 관리자를 사용하여 DHCP 서비스에 의해 이미 관리되는 네트워크에서 IP 주소를 추가합니다.	356 페이지 “단일 IP 주소를 추가하는 방법(DHCP 관리자)” 356 페이지 “기존 IP 주소를 복제하는 방법(DHCP 관리자)” 357 페이지 “복수 IP 주소를 추가하는 방법(DHCP 관리자)” 357 페이지 “IP 주소를 추가하는 방법(pntadm)”
IP 주소의 등록 정보를 변경합니다.	표 15–4에 설명된 IP 주소 등록 정보를 변경합니다.	359 페이지 “IP 주소 등록 정보를 수정하는 방법(DHCP 관리자)” 360 페이지 “IP 주소 등록 정보를 수정하는 방법(pntadm)”
DHCP 서비스에서 IP 주소를 제거합니다.	DHCP에서 지정된 IP 주소를 사용하지 못하도록 방지합니다.	361 페이지 “IP 주소를 사용할 수 없는 주소로 표시하는 방법(DHCP 관리자)” 361 페이지 “IP 주소를 사용할 수 없는 주소로 표시하는 방법(pntadm)” 362 페이지 “DHCP 서비스에서 IP 주소를 삭제하는 방법(DHCP 관리자)” 363 페이지 “DHCP 서비스에서 IP 주소를 삭제하는 방법(pntadm)”
DHCP 클라이언트에 일관된 IP 주소를 지정합니다.	클라이언트가 해당 구성을 요청할 때마다 동일한 IP 주소를 수신하도록 클라이언트를 설정합니다.	364 페이지 “DHCP 클라이언트에 일관성 있는 IP 주소를 지정하는 방법(DHCP 관리자)” 365 페이지 “DHCP 클라이언트에 일관성 있는 IP 주소를 지정하는 방법(pntadm)”

다음 표는 IP 주소의 등록 정보를 나열하고 설명합니다.

표 15–4 IP 주소 등록 정보

등록 정보	설명	pntadm 명령에서 지정하는 방법
네트워크 주소	작업 중인 IP 주소를 포함하는 네트워크 주소입니다. 네트워크 주소는 DHCP 관리자의 Addresses(주소) 탭에 있는 네트워크 목록에 표시됩니다.	네트워크 주소는 IP 주소를 만들고, 수정 또는 삭제하는 데 사용되는 pntadm 명령줄의 마지막 인수여야 합니다. 예를 들어 10.21.0.0 네트워크에 IP 주소를 추가하려면 다음을 입력합니다. **pntadm -A** *ip-address options* **10.21.0.0**

표 15–4 IP 주소 등록 정보 *(계속)*

등록 정보	설명	pntadm 명령에서 지정하는 방법
IP 주소	만들거나, 수정 또는 삭제하려는 주소입니다. IP 주소는 DHCP 관리자의 Addresses(주소) 탭의 첫번째 열에 표시됩니다.	IP 주소는 `pntadm` 명령에 `-A`, `-M` 및 `-D` 옵션도 제공해야 합니다. 예를 들어 IP 주소 `10.21.5.12`를 수정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12` *options* `10.21.0.0`**
클라이언트 이름	호스트 테이블의 IP 주소에 매핑되는 호스트 이름입니다. 이 이름은 주소가 만들어질 때 DHCP 관리자에 의해 자동으로 생성될 수 있습니다. 단일 주소를 만드는 경우 이름을 제공할 수 있습니다.	`-h` 옵션으로 클라이언트 이름을 지정합니다. 예를 들어 `10.21.5.12`에 대해 클라이언트 이름 `carrot12`를 지정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -h carrot12 10.21.0.0`**
서버가 소유	IP 주소를 관리하고 IP 주소 할당에 대한 DHCP 클라이언트의 요청에 응답하는 DHCP 서버입니다.	`-s` 옵션으로 소유하는 서버 이름을 지정합니다. 예를 들어, `blue2` 서버가 `10.21.5.12`를 소유하도록 지정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -s blue2 10.21.0.0`**
구성 매크로	DHCP 서버가 `dhcptab` 테이블에서 네트워크 구성 옵션을 가져오는 데 사용하는 매크로입니다. 서버를 구성할 때와 네트워크를 추가할 때 몇 개의 매크로가 자동으로 만들어집니다. 매크로에 대한 자세한 내용은 290 페이지 “DHCP 매크로 정보”를 참조하십시오. 주소가 만들어질 때 서버 매크로도 만들어집니다. 서버 매크로는 각 주소에 대한 구성 매크로로 지정됩니다.	`-m` 옵션으로 매크로 이름을 지정합니다. 예를 들어, 서버 매크로 `blue2`를 `10.21.5.12`에 지정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -m blue2 10.21.0.0`**
클라이언트 ID	DHCP 서비스 내에서 고유한 텍스트 문자열입니다. 클라이언트 ID가 00으로 표시되는 경우 해당 주소는 클라이언트에 할당되지 않습니다. IP 주소의 등록 정보를 수정할 때 클라이언트 ID를 지정하면 이 주소는 해당 클라이언트에 배타적으로 바인드됩니다. 클라이언트 ID는 DHCP 클라이언트의 판매자가 결정합니다. 클라이언트가 DHCP 클라이언트가 아닌 경우 자세한 내용은 해당 클라이언트 설명서를 참조하십시오.	`-i` 옵션으로 클라이언트 ID를 지정합니다. 예를 들어, 클라이언트 ID `08002094121E`를 주소 `10.21.5.12`에 지정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -i 0108002094121E 10.21.0.0`**

표 15–4 IP 주소 등록 정보 *(계속)*

등록 정보	설명	pntadm 명령에서 지정하는 방법
	DHCP 클라이언트의 경우 클라이언트 ID는 클라이언트의 16진수 하드웨어 주소에서 파생됩니다. 클라이언트 ID에는 네트워크 유형의 ARP 코드(예: 이더넷의 경우 01)를 나타내는 접두어가 포함됩니다. ARP 코드는 IANA(Internet Assigned Numbers Authority)에 의해 Assigned Numbers 표준의 ARP Parameters 절(`http://www.iana.com/numbers.html`)에 지정됩니다. 예를 들어, 16진수 이더넷 주소 `8:0:20:94:12:1e`를 가진 Oracle Solaris 클라이언트는 클라이언트 ID `0108002094121E`를 사용합니다. 클라이언트가 현재 주소를 사용 중인 경우 클라이언트 ID는 DHCP 관리자 및 `pntadm`에 나열됩니다. **팁:** Oracle Solaris 클라이언트 시스템의 수퍼 유저로서 `ifconfig -a` 명령을 입력하여 인터페이스의 이더넷 주소를 가져올 수 있습니다.	
예약	주소가 클라이언트 ID로 지시되는 클라이언트에 배타적으로 예약되고 DHCP 서버는 이 주소를 재생 이용할 수 없다고 지정하는 설정입니다. 이 옵션을 선택하는 경우 주소를 클라이언트에 수동으로 지정합니다.	`-f` 옵션을 사용하여 주소를 예약하는지 아니면 수동으로 지정하는지 지정합니다. 예를 들어, IP 주소 `10.21.5.12`를 클라이언트에 대해 예약하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -f MANUAL 10.21.0.0`**
임대 유형 또는 정책	DHCP가 클라이언트의 IP 주소 사용을 관리하는 방법을 결정하는 설정입니다. 임대는 동적이거나 영구적입니다. 전체 설명은 303 페이지 “동적 및 영구 임대 유형”을 참조하십시오.	`-f` 옵션으로 주소가 영구적으로 지정되도록 지정합니다. 주소는 기본적으로 동적으로 임대됩니다. 예를 들어, IP 주소 `10.21.5.12`가 영구 임대를 가지도록 지정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -f PERMANENT 10.21.0.0`**
임대 만료 날짜	임대가 만료되는 날짜로, 동적 임대가 지정된 경우에만 적용됩니다. 날짜는 *mm/dd/yyyy* 형식으로 지정됩니다.	`-e` 옵션으로 임대 만료 날짜를 지정합니다. 예를 들어, 2006년 1월 1일을 만료 날짜로 지정하려면 다음을 입력합니다. **`pntadm -M 10.21.5.12 -e 01/01/2006 10.21.0.0`**

표 15–4 IP 주소 등록 정보 (계속)

등록 정보	설명	pntadm 명령에서 지정하는 방법
BOOTP 설정	주소를 BOOTP 클라이언트용으로 예약된 주소로 표시하는 설정입니다. BOOTP 클라이언트 지원에 대한 자세한 내용은 347 페이지 “DHCP 서비스로 BOOTP 클라이언트 지원(작업 맵)”을 참조하십시오.	-f 옵션을 사용하여 BOOTP 클라이언트용으로 주소를 예약합니다. 예를 들어, IP 주소 10.21.5.12를 BOOTP 클라이언트용으로 예약하려면 다음을 입력합니다. **pntadm -M 10.21.5.12 -f BOOTP 10.21.0.0**
사용할 수 없는 설정	주소를 클라이언트에 지정할 수 없도록 표시하는 설정입니다.	-f 옵션을 사용하여 주소를 사용할 수 없는 주소로 표시합니다. 예를 들어, IP 주소 10.21.5.12를 사용할 수 없는 주소로 표시하려면 다음을 입력합니다. **pntadm -M 10.21.5.12 -f UNUSABLE 10.21.0.0**

DHCP 서비스에 IP 주소 추가

IP 주소를 추가하기 전에 해당 주소를 소유하는 네트워크를 DHCP 서비스에 추가해야 합니다. 네트워크를 추가하는 방법은 340 페이지 “DHCP 네트워크 추가”를 참조하십시오.

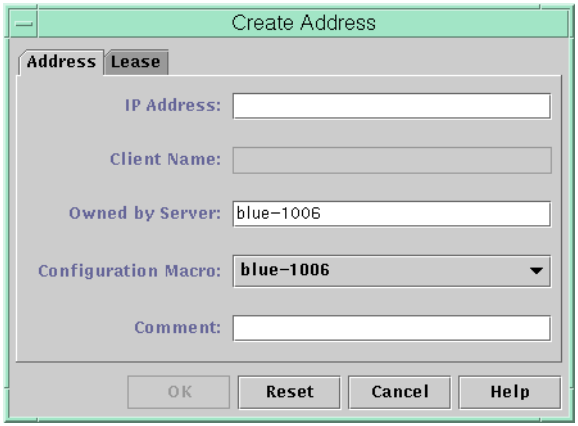
DHCP 관리자 또는 pntadm 명령으로 주소를 추가할 수 있습니다.

이미 DHCP 서비스에 의해 관리되는 네트워크에서는 DHCP 관리자를 사용하여 다음과 같은 몇 가지 방식으로 주소를 추가할 수 있습니다.

- **단일 IP 주소 추가** – 하나의 새 IP 주소를 DHCP 관리를 받도록 지정합니다.
- **기존 IP 주소 복제** – DHCP에 의해 관리되는 기존 IP 주소의 등록 정보를 복사하여 새 IP 주소 및 클라이언트 이름을 제공합니다.
- **복수 IP 주소 범위 추가** – 주소 마법사를 사용하여 일련의 IP 주소를 DHCP 관리를 받도록 지정합니다.

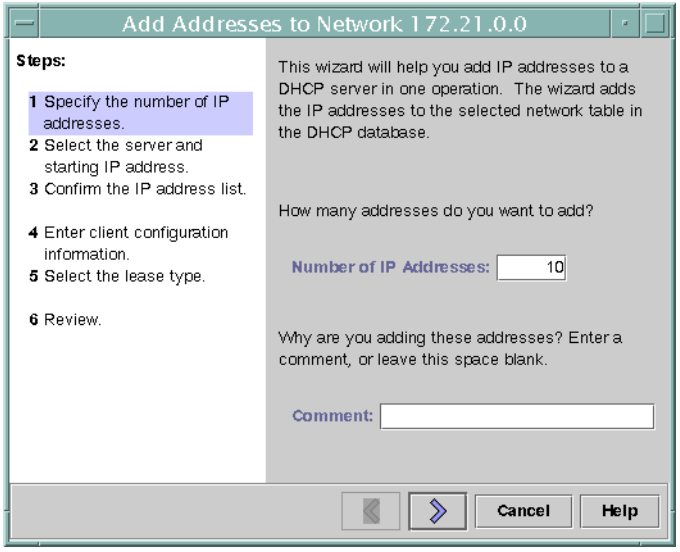
다음 그림은 Create Address(주소 만들기) 대화 상자를 보여줍니다. 텍스트 필드에 기존 주소 값이 표시되는 것을 제외하고는 Duplicate Address(주소 복제) 대화 상자는 Create Address(주소 만들기) 대화 상자와 동일합니다.

그림 15–8 DHCP 관리자의 Create Address(주소 만들기) 대화 상자



다음 그림은 IP 주소 범위를 추가하는 데 사용되는 Add Addresses to Network(네트워크에 주소 추가) 마법사의 첫번째 대화 상자를 보여줍니다.

그림 15–9 DHCP 관리자의 Add Addresses to Network(네트워크에 주소 추가) 마법사



▼ 단일 IP 주소를 추가하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**
DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.

2 **새 IP 주소를 추가할 네트워크를 선택합니다.**

3 **Edit(편집) 메뉴에서 Create(만들기)를 선택합니다.**
Create Address(주소 만들기) 대화 상자가 열립니다.

4 **Address(주소) 및 Lease(임대) 탭에서 주소 설정에 대한 값을 선택하거나 입력합니다.**
Help(도움말) 버튼을 선택하여 대화 상자에 대한 도움말을 표시하는 웹 브라우저를 엽니다. 설정에 대한 자세한 내용은 표 15–4를 참조하십시오.

5 **OK(확인)를 누릅니다.**

▼ 기존 IP 주소를 복제하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**
DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.

2 **새 IP 주소가 있는 네트워크를 선택합니다.**

3 **복제할 등록 정보가 있는 주소를 선택합니다.**

4 **Edit(편집) 메뉴에서 Duplicate(복제)를 선택합니다.**

5 **IP Address(IP 주소) 필드에 새 IP 주소를 지정합니다.**

6 **(옵션) 주소에 대한 새 클라이언트 이름을 지정합니다.**
복제 중인 주소가 사용하는 것과 동일한 이름은 사용할 수 없습니다.

7 **(옵션) 필요한 경우 기타 옵션 값을 수정합니다.**
대부분의 다른 옵션 값은 동일하게 유지되어야 합니다.

8 **OK(확인)를 누릅니다.**

▼ 복수 IP 주소를 추가하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **새 IP 주소를 추가할 네트워크를 선택합니다.**

3 **Edit(편집) 메뉴에서 Address Wizard(주소 마법사)를 선택합니다.**

Add Addresses to Network(네트워크에 주소 추가) 대화 상자에는 IP 주소 등록 정보 값을 제공하라는 메시지가 표시됩니다. 등록 정보에 대한 자세한 내용은 표 15–4를 참조하거나 대화 상자에서 Help(도움말) 버튼을 선택하십시오. 더 자세한 내용은 301 페이지 “IP 주소 관리를 위한 결정 사항(작업 맵)”을 참조하십시오.

4 **각 화면을 마치면 오른쪽 화살표 버튼을 누르고 마지막 화면에서는 Finish(마침)를 누릅니다.**

Addresses(주소) 탭이 새 주소로 업데이트됩니다.

▼ IP 주소를 추가하는 방법(pntadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 IP 주소를 추가합니다.**

```
# pntadm -A ip-address options network-address
```

`pntadm -A`에서 사용할 수 있는 옵션 목록은 `pntadm`(1M) 매뉴얼 페이지를 참조하십시오. 또한 표 15–4는 옵션을 지정하는 `pntadm` 명령의 몇 가지 예를 보여줍니다.

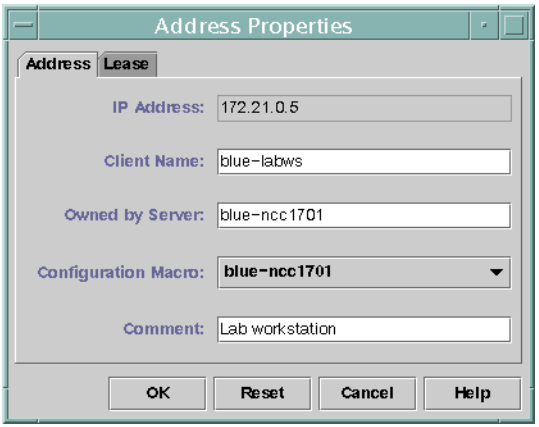
주 – `pntadm`으로 주소를 여러 개 추가하는 스크립트를 작성할 수 있습니다. 이러한 예를 보려면 예 18–1을 참조하십시오.

DHCP 서비스에서 IP 주소 수정

DHCP 관리자 또는 pntadm -M 명령을 사용하여 표 15–4에 설명된 모든 주소 등록 정보를 수정할 수 있습니다. pntadm -M에 대한 자세한 내용은 pntadm(1M) 매뉴얼 페이지를 참조하십시오.

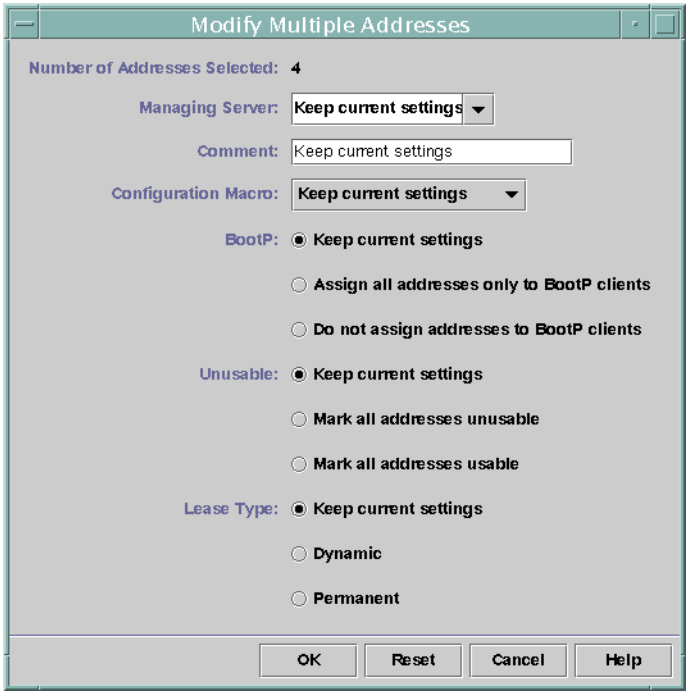
다음 그림은 IP 주소 등록 정보를 수정하는 데 사용하는 Address Properties(주소 등록 정보) 대화 상자를 보여줍니다.

그림 15–10 DHCP 관리자의 Address Properties(주소 등록 정보) 대화 상자



다음 그림은 복수 IP 주소를 수정하는 데 사용하는 Modify Multiple Addresses(복수 주소 수정) 대화 상자를 보여줍니다.

그림 15–11 DHCP 관리자의 Modify Multiple Addresses(복수 주소 수정) 대화 상자



▼ IP 주소 등록 정보를 수정하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **IP 주소의 네트워크를 선택합니다.**

3 **수정할 IP 주소를 하나 이상 선택합니다.**

주소를 여러 개 수정하려면 Ctrl 키를 누른 상태로 마우스를 눌러 주소를 여러 개 선택합니다. Shift 키를 누른 상태로 마우스를 눌러 주소 블록을 선택할 수도 있습니다.

4 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**

Address Properties(주소 등록 정보) 대화 상자 또는 Modify Multiple Address(복수 주소 수정) 대화 상자가 열립니다.

5 **해당하는 등록 정보를 변경합니다.**

등록 정보에 대한 자세한 내용은 Help(도움말) 버튼을 누르거나 표 15–4를 참조하십시오.

6 **OK(확인)를 누릅니다.**

▼ IP 주소 등록 정보를 수정하는 방법(pntadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 IP 주소 등록 정보를 수정합니다.**

```
# pntadm -M ip-address options network-address
```

`pntadm` 명령에서 사용할 수 있는 다양한 옵션은 `pntadm`(1M) 매뉴얼 페이지를 참조하십시오.

표 15–4는 옵션을 지정하는 `pntadm` 명령의 몇 가지 예를 보여줍니다.

DHCP 서비스에서 IP 주소 제거

DHCP 서비스가 특정 IP 주소 또는 주소 그룹을 관리하지 못하도록 해야 하는 경우가 있습니다. DHCP에서 주소를 제거하기 위해 사용하는 방법은 변경 사항이 일시적인지 영구적인지에 따라 다릅니다.

- 주소를 일시적으로 사용할 수 없도록 하려면 360 페이지 “IP 주소를 DHCP 서비스에서 사용할 수 없는 주소로 표시”에 설명된 대로 Address Properties(주소 등록 정보) 대화 상자에서 주소를 사용할 수 없는 주소로 표시합니다.
- DHCP 클라이언트가 주소를 영구적으로 사용하지 못하게 하려면 362 페이지 “DHCP 서비스에서 IP 주소 삭제”에 설명된 대로 DHCP 네트워크 테이블에서 주소를 삭제합니다.

IP 주소를 DHCP 서비스에서 사용할 수 없는 주소로 표시

`-f UNUSABLE` 옵션과 함께 `pntadm -M` 명령을 사용하여 주소를 사용할 수 없는 주소로 표시할 수 있습니다.

개별 주소를 표시하려면 그림 15–10에 설명된 대로 DHCP 관리자에서 Address Properties(주소 등록 정보) 대화 상자를 사용합니다. 그림 15–11에 설명된 대로 Modify Multiple Addresses(복수 주소 수정) 대화 상자를 사용하여 다음과 같은 절차로 복수 주소를 표시할 수 있습니다.

▼ IP 주소를 사용할 수 없는 주소로 표시하는 방법(DHCP 관리자)

1. **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**
   DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.
2. **IP 주소의 네트워크를 선택합니다.**
3. **사용할 수 없는 주소로 표시할 IP 주소를 하나 이상 선택합니다.**
   여러 개의 주소를 사용할 수 없는 주소로 표시하려는 경우 Ctrl 키를 누른 상태로 마우스를 눌러 여러 개의 주소를 선택합니다. Shift 키를 누른 상태로 마우스를 눌러 주소 블록을 선택할 수도 있습니다.
4. **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**
   Address Properties(주소 등록 정보) 대화 상자 또는 Modify Multiple Address(복수 주소 수정) 대화 상자가 열립니다.
5. **주소를 하나만 수정하려면 Lease(임대) 탭을 선택합니다.**
6. **Address is Unusable(주소를 사용할 수 없음)을 선택합니다.**
   여러 주소를 편집하려면 Mark All Addresses Unusable(모든 주소를 사용할 수 없는 주소로 표시)을 선택합니다.
7. **OK(확인)를 누릅니다.**

▼ IP 주소를 사용할 수 없는 주소로 표시하는 방법(pntadm)

1. **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**
   DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

   역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.
2. **다음 형식으로 명령을 입력하여 IP 주소를 사용할 수 없는 주소로 표시합니다.**

   ```
   # pntadm -M ip-address -f UNUSABLE network-address
   ```

   예를 들어, `10.64.3.3`을 사용할 수 없는 주소로 표시하려면 다음을 입력합니다.

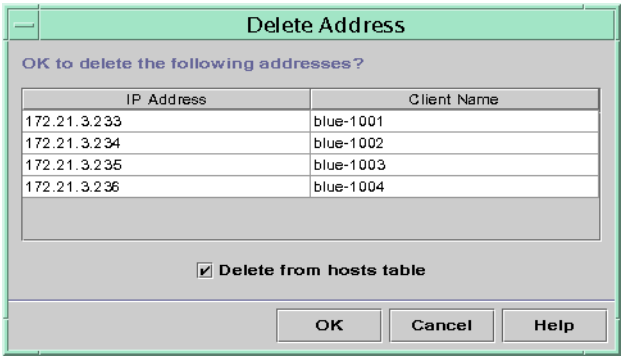
```
pntadm -M 10.64.3.3 -f UNUSABLE 10.64.3.0
```

DHCP 서비스에서 IP 주소 삭제

DHCP에서 IP 주소를 관리하지 않게 하려면 해당 주소를 DHCP 네트워크 테이블에서 삭제해야 합니다. `pntadm -D` 명령 또는 DHCP 관리자의 Delete Address(주소 삭제) 대화 상자를 사용할 수 있습니다.

다음 그림은 Delete Address(주소 삭제) 대화 상자를 보여줍니다.

그림 15–12 DHCP 관리자의 Delete Address(주소 삭제) 대화 상자



▼ DHCP 서비스에서 IP 주소를 삭제하는 방법(DHCP 관리자)

1. **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**

   DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2. **IP 주소의 네트워크를 선택합니다.**

3. **삭제할 IP 주소를 하나 이상 선택합니다.**

   주소를 여러 개 삭제하려면 Ctrl 키를 누른 상태로 마우스를 눌러 주소를 여러 개 선택합니다. Shift 키를 누른 상태로 마우스를 눌러 주소 블록을 선택할 수도 있습니다.

4. **Edit(편집) 메뉴에서 Delete(삭제)를 선택합니다.**

   Delete Address(주소 삭제) 대화 상자에는 선택한 주소가 표시되므로 삭제할지 여부를 확인할 수 있습니다.

5 **호스트 테이블에서 호스트 이름을 삭제하려면 Delete From Hosts Table(호스트 테이블에서 삭제)을 선택합니다.**

호스트 이름이 DHCP 관리자에 의해 생성된 경우 호스트 테이블에서 해당 이름을 삭제해야 할 수 있습니다.

6 **OK(확인)를 누릅니다.**

▼ DHCP 서비스에서 IP 주소를 삭제하는 방법(pntadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **다음 형식의 명령을 입력하여 IP 주소를 삭제합니다.**

```
# pntadm -D ip-address options network-address
```

-y 옵션을 포함하는 경우 호스트 이름이 해당 호스트 이름을 유지 관리하는 이름 서비스에서 삭제됩니다.

예를 들어, `10.64.3.0` 네트워크에서 `10.64.3.3` 주소를 삭제하고 해당 호스트 이름을 삭제하려면 다음을 입력합니다.

```
pntadm -D 10.64.3.3 -y 10.64.3.0
```

예약된 IP 주소를 DHCP 클라이언트에 지정

DHCP 서비스는 이전에 DHCP를 통해 주소를 획득한 클라이언트에 동일한 IP 주소를 제공하려고 시도합니다. 하지만 주소가 이미 다른 클라이언트에 재지정되었을 수도 있습니다.

네트워크에 중요한 라우터, NIS 또는 NIS+ 서버, DNS 서버 및 기타 호스트는 DHCP 클라이언트가 아니어야 합니다. 네트워크에 서비스를 제공하는 호스트는 네트워크에 의존하여 자체 IP 주소를 획득하지 않아야 합니다. 또한 인쇄 서버 또는 파일 서버와 같은 클라이언트는 일관성 있는 IP 주소를 가져야 합니다. 이러한 클라이언트는 자신의 네트워크 구성을 수신할 수 있으며 DHCP 서버에서 일관성 있는 IP 주소를 지정받을 수도 있습니다.

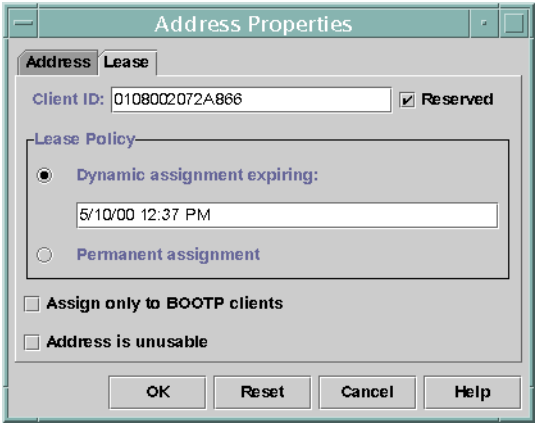
클라이언트가 해당 구성을 요청할 때마다 클라이언트에 동일한 IP 주소를 제공하도록 DHCP 서버를 설정할 수 있습니다. 클라이언트 ID를 클라이언트가 사용할 IP 주소에 수동으로 지정하여 해당 주소를 클라이언트용으로 예약합니다. 예약된 주소가 동적

임대 또는 영구 임대 중 하나를 사용하도록 설정할 수 있습니다. 클라이언트의 주소가 동적 임대를 사용하는 경우 주소 사용을 쉽게 추적할 수 있습니다. 디스크가 없는 클라이언트는 동적 임대를 통해 예약된 주소를 사용해야 하는 클라이언트의 예입니다. 클라이언트의 주소가 영구 임대를 사용하는 경우 주소 사용을 추적할 수 없습니다. 클라이언트가 영구 임대를 획득하면 다시는 서버에 연결하지 않습니다. 이 클라이언트는 IP 주소를 해제하고 DHCP 임대 협상을 다시 시작해야만 업데이트된 구성 정보를 얻을 수 있습니다.

`pntadm -M` 명령 또는 DHCP 관리자의 Address Properties(주소 등록 정보) 대화 상자를 사용하여 임대 등록 정보를 설정할 수 있습니다.

다음 그림은 임대를 수정하는 데 사용되는 Address Properties(주소 등록 정보) 대화 상자의 Lease(임대) 탭을 보여줍니다.

그림 15–13 DHCP 관리자의 Address Properties(주소 등록 정보) Lease(임대) 탭



▼ DHCP 클라이언트에 일관성 있는 IP 주소를 지정하는 방법(DHCP 관리자)

1. **DHCP 관리자에서 Addresses(주소) 탭을 선택합니다.**
   DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.
2. **해당 네트워크를 선택합니다.**
3. **클라이언트가 사용할 IP 주소를 두 번 누릅니다.**
   Address Properties(주소 등록 정보) 창이 열립니다.

4 **Lease(임대) 탭을 선택합니다.**

5 **Client ID(클라이언트 ID) 필드에 클라이언트 ID를 입력합니다.**

클라이언트 ID는 클라이언트의 하드웨어 주소에서 파생됩니다. 자세한 내용은 표 15–4의 클라이언트 ID 항목을 참조하십시오.

6 **서버에서 IP 주소를 재생 사용하지 못하게 하려면 Reserved(예약) 옵션을 선택합니다.**

7 **창의 Lease Policy(임대 정책) 영역에서 Dynamic(동적) 또는 Permanent(영구) 지정을 선택합니다.**

클라이언트가 임대 갱신을 위해 협상하게 하려면 Dynamic(동적)을 선택합니다. 이렇게 하면 주소 사용 시 추적할 수 있습니다. Reserved(예약)를 선택했으므로 동적 임대가 지정되더라도 주소를 재생 사용할 수 없습니다. 이 임대에 대해서는 만료 날짜를 지정할 필요가 없습니다. DHCP 서버는 임대 시간을 사용하여 만료 날짜를 계산합니다.

Permanent(영구)를 선택하면 트랜잭션 로깅을 사용하지 않는 한 IP 주소 사용을 추적할 수 없습니다.

8 **OK(확인)를 누릅니다.**

▼ DHCP 클라이언트에 일관성 있는 IP 주소를 지정하는 방법(pntadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 임대 플래그를 설정합니다.**

```
# pntadm -M ip-address -i client-id -f MANUAL+BOOTP network-address
```

예를 들어, MAC 주소가 `08:00:20:94:12:1E`인 DHCP 클라이언트가 항상 IP 주소 `10.21.5.12`를 받게 하려면 다음을 입력합니다.

```
pntadm -M 10.21.5.12 -i 0108002094121E -f MANUAL+BOOTP 10.21.0.0
```

참고 – 클라이언트 식별자를 결정하는 방법은 표 15–4의 클라이언트 ID 항목을 참조하십시오.

DHCP 매크로 작업(작업 맵)

DHCP **매크로**는 DHCP 옵션의 컨테이너입니다. DHCP 서비스는 매크로를 사용하여 클라이언트에 전달할 옵션을 수집합니다. 서버를 구성하면 DHCP 관리자 및 `dhcpconfig` 유틸리티에서 다수의 매크로가 자동으로 만들어집니다. 매크로에 대한 백그라운드 정보는 290 페이지 "DHCP 매크로 정보"를 참조하십시오. 기본적으로 만들어지는 매크로에 대한 정보는 14 장, "DHCP 서비스 구성(작업)"을 참조하십시오.

네트워크에 변경 사항이 발생하면 클라이언트에 전달되는 구성 정보를 변경해야 할 수 있습니다. 구성 정보를 변경하려면 DHCP 매크로를 사용해야 합니다. DHCP 매크로를 보고, 만들고, 수정, 복제 및 삭제할 수 있습니다.

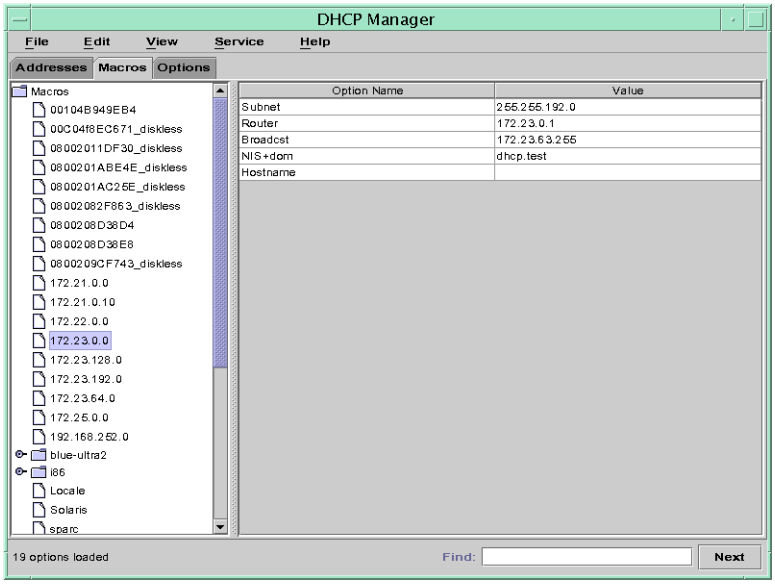
매크로에 대한 작업을 수행하려면 `dhcp_inittab(4)` 매뉴얼 페이지에 설명된 DHCP 표준 옵션에 대해 알고 있어야 합니다.

다음 작업 맵은 DHCP 매크로를 보고, 만들고, 수정하고, 삭제하는 데 도움이 되는 작업을 나열합니다. 맵에는 각 작업을 수행하는 방법을 자세히 보여주는 절에 대한 링크가 포함되어 있습니다.

작업	설명	수행 방법
DHCP 매크로를 봅니다.	DHCP 서버에 정의된 모든 매크로 목록을 표시합니다.	367 페이지 "DHCP 서버에 정의된 매크로를 보는 방법(DHCP 관리자)" 368 페이지 "DHCP 서버에 정의된 매크로를 보는 방법(`dhtadm`)"
DHCP 매크로를 만듭니다.	DHCP 클라이언트를 지원하는 새 매크로를 만듭니다.	373 페이지 "DHCP 매크로를 만드는 방법(DHCP 관리자)" 374 페이지 "DHCP 매크로를 만드는 방법(`dhtadm`)"
DHCP 클라이언트로 매크로에 전달되는 값을 수정합니다.	기존 옵션을 수정하거나, 매크로에 옵션을 추가하거나, 매크로에서 옵션을 제거하여 매크로를 변경합니다.	369 페이지 "DHCP 매크로에서 옵션 값을 변경하는 방법(DHCP 관리자)" 370 페이지 "DHCP 매크로에서 옵션 값을 변경하는 방법(`dhtadm`)" 370 페이지 "DHCP 매크로에 옵션을 추가하는 방법(DHCP 관리자)" 371 페이지 "DHCP 매크로에 옵션을 추가하는 방법(`dhtadm`)" 371 페이지 "DHCP 매크로에서 옵션을 삭제하는 방법(DHCP 관리자)" 372 페이지 "DHCP 매크로에서 옵션을 삭제하는 방법(`dhtadm`)"
DHCP 매크로를 삭제합니다.	더 이상 사용되지 않는 DHCP 매크로를 제거합니다.	375 페이지 "DHCP 매크로를 삭제하는 방법(DHCP 관리자)" 375 페이지 "DHCP 매크로를 삭제하는 방법(`dhtadm`)"

다음 그림은 DHCP 관리자 창의 Macros(매크로) 탭을 보여줍니다.

그림 15–14 DHCP 관리자의 Macros(매크로) 탭



▼ DHCP 서버에 정의된 매크로를 보는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

창 왼쪽의 Macros(매크로) 영역에는 DHCP 서버에 정의된 모든 매크로가 사전순으로 표시됩니다. 앞에 폴더 아이콘이 붙어있는 매크로는 다른 매크로에 대한 참조를 포함하는 반면 앞에 문서 아이콘이 붙어있는 매크로는 다른 매크로를 참조하지 않습니다.

2 **매크로 폴더를 열려면 폴더 아이콘 왼쪽의 핸들 아이콘을 누릅니다.**

선택한 매크로에 포함된 매크로가 나열됩니다.

3 **매크로 내용을 보려면 매크로 이름을 누릅니다.**

옵션 및 옵션에 지정된 값이 표시됩니다.

▼ DHCP 서버에 정의된 매크로를 보는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 명령을 입력하여 매크로를 표시합니다.**

```
# dhtadm -P
```

이 명령은 DHCP 서버에 정의된 모든 매크로와 기호를 비롯하여 `dhcptab` 테이블의 내용을 지정된 형식으로 표준 출력에 표시합니다.

DHCP 매크로 수정

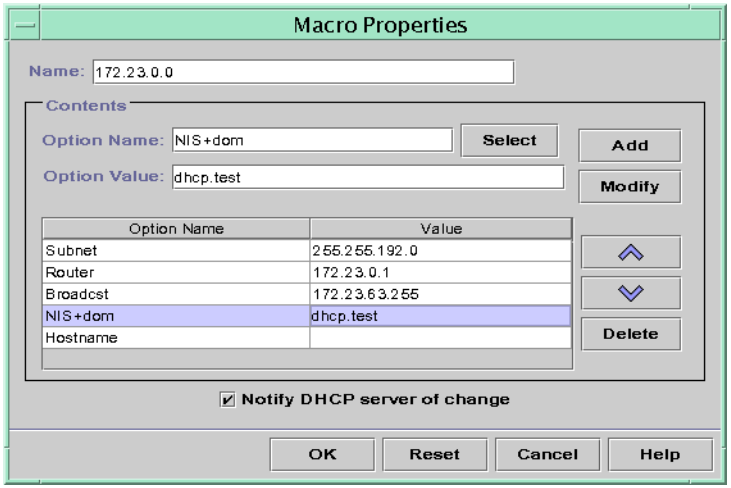
네트워크의 일부가 변경되고 하나 이상의 DHCP 클라이언트가 이 변경 사항에 대해 알아야 하는 경우 매크로를 수정해야 할 수 있습니다. 예를 들어, 라우터 또는 NIS 서버를 추가하거나, 새 서브넷을 만들거나, 임대 정책을 변경할 수 있습니다.

매크로를 수정하기 전에 변경, 추가 또는 삭제할 DHCP 옵션의 이름을 파악합니다. 표준 DHCP 옵션은 DHCP 관리자 도움말 및 `dhcp_inittab(4)` 매뉴얼 페이지를 참조하십시오.

`dhtadm -M -m` 명령 또는 DHCP 관리자를 사용하여 매크로를 수정할 수 있습니다. `dhtadm`에 대한 자세한 내용은 `dhtadm(1M)` 매뉴얼 페이지를 참조하십시오.

다음 그림은 DHCP 관리자의 Macro Properties(매크로 등록 정보) 대화 상자를 보여줍니다.

그림 15–15 DHCP 관리자의 Macro Properties(매크로 등록 정보) 대화 상자



▼ DHCP 매크로에서 옵션 값을 변경하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **변경할 매크로를 선택합니다.**

3 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**

Macro Properties(매크로 등록 정보) 대화 상자가 열립니다.

4 **옵션 테이블에서 변경할 옵션을 선택합니다.**

옵션 이름 및 값이 Option Name(옵션 이름) 및 Option Value(옵션 값) 필드에 표시됩니다.

5 **Option Value(옵션 값) 필드에서 옵션에 대해 이전 값을 선택하고 새 값을 입력합니다.**

6 **Modify(수정)를 누릅니다.**

새 값이 옵션 테이블에 표시됩니다.

7 **Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**

이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

8 **OK(확인)를 누릅니다.**

▼ DHCP 매크로에서 옵션 값을 변경하는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 옵션 값을 변경합니다.**

```
# dhtadm -M -m macroname -e 'option=value:option=value' -g
```

예를 들어, `bluenote` 매크로에서 임대 시간 및 Universal Time Offset을 변경하려면 다음을 입력합니다.

```
# dhtadm -M -m bluenote -e 'LeaseTim=43200:UTCOffst=28800' -g
```

▼ DHCP 매크로에 옵션을 추가하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **변경할 매크로를 선택합니다.**

3 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**

Macro Properties(매크로 등록 정보) 대화 상자가 열립니다.

4 **Option Name(옵션 이름) 필드에서 다음 방법 중 하나를 사용하여 옵션 이름을 지정합니다.**

- **Option Name(옵션 이름) 필드 옆의 Select(선택) 버튼을 눌러 매크로에 추가할 옵션을 선택합니다.**

  Select Option(옵션 선택) 대화 상자에는 표준 범주 옵션 이름 및 설명 목록이 사전순으로 표시됩니다. 표준 범주에 없는 옵션을 추가하려면 Category(범주) 목록을 사용하여 범주를 선택합니다.

  매크로 범주에 대한 자세한 내용을 290 페이지 “DHCP 매크로 정보”를 참조하십시오.

- **새 매크로에 기존 매크로에 대한 참조를 포함하려면 `Include`를 입력합니다.**

5 **Option Value(옵션 값) 필드에 옵션 값을 입력합니다.**

옵션 이름으로 **Include**를 입력한 경우 Option Value(옵션 값) 필드에 기존 매크로의 이름을 지정해야 합니다.

6 **Add(추가)를 누릅니다.**

옵션이 이 매크로의 옵션 목록 맨 아래에 추가됩니다. 매크로에서 옵션의 위치를 변경하려면 옵션을 선택하고 화살표 버튼을 눌러 목록에서 옵션을 위/아래로 이동합니다.

7 **Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**

이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

8 **OK(확인)를 누릅니다.**

▼ DHCP 매크로에 옵션을 추가하는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 매크로에 옵션을 추가합니다.**

```
# dhtadm -M -m macroname -e 'option=value' -g
```

예를 들어, `bluenote` 매크로에 임대 협상 기능을 추가하려면 다음 명령을 입력합니다.

```
# dhtadm -M -m bluenote -e 'LeaseNeg=_NULL_VALUE' -g
```

옵션에 값이 필요하지 않은 경우 옵션 값으로 `_NULL_VALUE`를 사용해야 합니다.

▼ DHCP 매크로에서 옵션을 삭제하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **변경할 매크로를 선택합니다.**

3 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**
Macro Properties(매크로 등록 정보) 대화 상자가 열립니다.

4 **매크로에서 제거할 옵션을 선택합니다.**

5 **Delete(삭제)를 누릅니다.**
이 매크로에 대한 옵션 목록에서 옵션이 제거됩니다.

6 **Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**
이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

7 **OK(확인)를 누릅니다.**

▼ DHCP 매크로에서 옵션을 삭제하는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**
DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 매크로에서 옵션을 삭제합니다.**

```
# dhtadm -M -m macroname -e 'option=' -g
```

예를 들어, `bluenote` 매크로에서 임대 협상 기능을 제거하려면 다음 명령을 입력합니다.

```
# dhtadm -M -m bluenote -e 'LeaseNeg=' -g
```

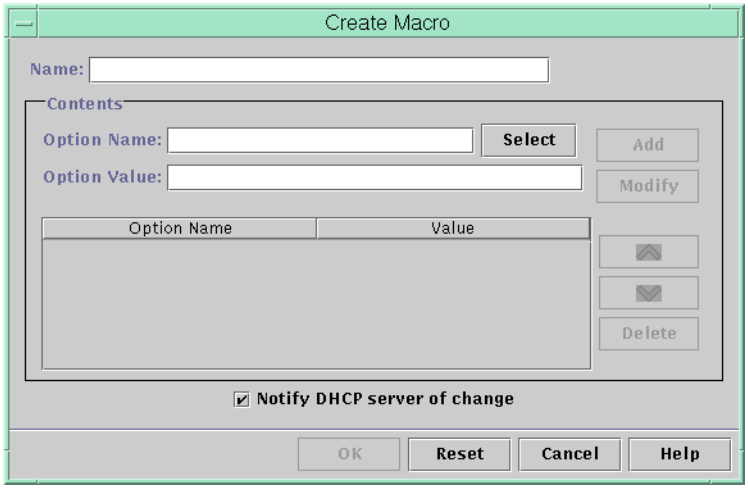
값이 없이 옵션이 지정된 경우 해당 옵션은 매크로에서 제거됩니다.

DHCP 매크로 만들기

특별한 요구 사항이 있는 클라이언트를 지원하기 위해 DHCP 서비스에 새 매크로를 추가할 수 있습니다. `dhtadm -A -m` 명령 또는 DHCP 관리자의 Create Macro(매크로 만들기) 대화 상자를 사용하여 매크로를 추가할 수 있습니다. `dhtadm` 명령에 대한 자세한 내용은 `dhtadm(1M)` 매뉴얼 페이지를 참조하십시오.

다음 그림은 DHCP 관리자의 Create Macro(매크로 만들기) 대화 상자를 보여줍니다.

그림 15–16 DHCP 관리자의 Create Macro(매크로 만들기) 대화 상자



▼ DHCP 매크로를 만드는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **Edit(편집) 메뉴에서 Create(만들기)를 선택합니다.**

Create Macro(매크로 만들기) 대화 상자가 열립니다.

3 **매크로의 고유 이름을 입력합니다.**

이름에는 최대 128개의 영숫자를 사용할 수 있습니다. 공급업체 클래스 식별자, 네트워크 주소 또는 클라이언트 ID와 일치하는 이름을 사용하는 경우 해당 클라이언트에 대해 매크로가 자동으로 처리됩니다. 다른 이름을 사용하는 경우 매크로가 자동으로 처리되지 않습니다. 이런 매크로는 특정 IP 주소에 지정되거나 자동으로 처리되는 다른 매크로 내부에 포함되어야 합니다. 자세한 내용은 290 페이지 “DHCP 서버의 매크로 처리”를 참조하십시오.

4 **Option Name(옵션 이름) 필드 옆의 Select(선택) 버튼을 누릅니다.**

Select Option(옵션 선택) 대화 상자에는 표준 범주 옵션 이름 및 해당 설명 목록이 사전순으로 표시됩니다. 표준 범주에 없는 옵션을 추가하려면 Category(범주) 목록을 사용합니다. Category(범주) 목록에서 범주를 선택합니다. 옵션 범주에 대한 자세한 내용은 290 페이지 “DHCP 옵션 정보”를 참조합니다.

5 **매크로에 추가할 옵션을 선택하고 OK(확인)를 누릅니다.**

Macro Properties(매크로 등록 정보) 대화 상자의 Option Name(옵션 이름) 필드에 선택된 옵션이 표시됩니다.

6 **Option Value(옵션 값) 필드에 옵션 값을 입력하고 Add(추가)를 누릅니다.**

옵션이 이 매크로의 옵션 목록 맨 아래에 추가됩니다. 매크로에서 옵션의 위치를 변경하려면 옵션을 선택하고 화살표 버튼을 눌러 목록에서 옵션을 위/아래로 이동합니다.

7 **매크로에 추가할 각 옵션에 대해 단계 5 및 단계 6을 반복합니다.**

8 **옵션 추가를 마치면 Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**

이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

9 **OK(확인)를 누릅니다.**

▼ DHCP 매크로를 만드는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 매크로를 만듭니다.**

```
# dhtadm -A -m macroname -d ':option=value:option=value:option=value:' -g
```

`-d`에 대한 인수에 포함할 수 있는 *option=value* 쌍의 개수에는 제한이 없습니다. 인수는 콜론으로 시작되고 끝나야 하며, 각 *option=value* 쌍 사이에도 콜론이 있어야 합니다. 전체 문자열은 따옴표로 묶어야 합니다.

예를 들어, `bluenote` 매크로를 만들려면 다음 명령을 입력합니다.

```
# dhtadm -A -m bluenote -d ':Router=10.63.6.121\
:LeaseNeg=_NULL_VALUE:DNSserv=10.63.28.12:' -g
```

옵션에 값이 필요하지 않은 경우 옵션 값으로 `_NULL_VALUE`를 사용해야 합니다.

DHCP 매크로 삭제

DHCP 서비스에서 매크로를 삭제해야 할 수 있습니다. 예를 들어, DHCP 서비스에서 네트워크를 삭제하는 경우 연관된 네트워크 매크로도 삭제할 수 있습니다.

`dhtadm -D -m` 명령 또는 DHCP 관리자를 사용하여 매크로를 삭제할 수 있습니다.

▼ DHCP 매크로를 삭제하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Macros(매크로) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.

2 **삭제할 매크로를 선택합니다.**

Delete Macro(매크로 삭제) 대화 상자에서는 지정된 매크로를 삭제할지 묻는 메시지가 표시됩니다.

3 **Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**

이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

4 **OK(확인)를 누릅니다.**

▼ DHCP 매크로를 삭제하는 방법(dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 매크로를 삭제합니다.**

```
# dhtadm -D -m macroname -g
```

예를 들어, `bluenote` 매크로를 삭제하려면 다음 명령을 입력합니다.

```
# dhtadm -D -m bluenote -g
```

DHCP 옵션 작업(작업 맵)

옵션은 DHCP 서버가 클라이언트에 전달할 수 있는 네트워크 구성 매개변수에 대한 키워드입니다. DHCP 서비스에서 표준 DHCP 옵션을 만들거나 삭제 또는 수정할 수 없습니다. 표준 옵션은 DHCP 프로토콜에 의해 정의되므로 변경할 수 없습니다. 사이트에 대해 사용자가 만든 옵션에 대해서만 작업을 수행할 수 있습니다. 이 때문에 DHCP 서비스를 처음 설정할 때 DHCP 관리자의 Options(옵션) 탭은 사용자가 사이트에 대한 옵션을 만들 때까지 비어 있습니다.

DHCP 서버에서 옵션을 만드는 경우 DHCP 클라이언트에서도 옵션에 대한 정보를 추가해야 합니다. DHCP 클라이언트의 경우 `/etc/dhcp/inittab` 파일을 편집하여 새 옵션에 대한 항목을 추가해야 합니다. 이 파일에 대한 자세한 내용은 `dhcp_inittab(4)` 매뉴얼 페이지를 참조하십시오.

Oracle Solaris 클라이언트가 아닌 DHCP 클라이언트가 있는 경우 해당 클라이언트 설명서에서 옵션 또는 기호 추가에 대한 정보를 참조하십시오. DHCP의 옵션에 대한 자세한 내용은 290 페이지 “DHCP 옵션 정보”를 참조하십시오.

DHCP 관리자 또는 `dhtadm` 명령을 사용하여 옵션을 만들거나, 수정 또는 삭제할 수 있습니다.

참고 – DHCP 관련 문맥에서 옵션은 **기호**로 불립니다. `dhtadm` 명령 및 관련 매뉴얼 페이지에서도 옵션은 기호로 표기됩니다.

다음 작업 맵은 DHCP 옵션을 만들거나 수정 및 삭제하기 위해 수행하는 작업을 나열합니다. 작업 맵에는 작업 절차에 대한 링크가 포함되어 있습니다.

작업	설명	수행 방법
DHCP 옵션을 만듭니다.	표준 DHCP 옵션에 포함되지 않은 정보에 대해 새 옵션을 추가합니다.	380 페이지 “DHCP 옵션을 만드는 방법(DHCP 관리자)” 381 페이지 “DHCP 옵션을 만드는 방법(dhtadm)” 385 페이지 “DHCP 클라이언트의 옵션 정보 수정”
DHCP 옵션을 수정합니다.	만들어진 DHCP 옵션의 등록 정보를 변경합니다.	382 페이지 “DHCP 옵션 등록 정보를 수정하는 방법(DHCP 관리자)” 383 페이지 “DHCP 옵션 등록 정보를 수정하는 방법 (dhtadm)”

작업	설명	수행 방법
DHCP 옵션을 삭제합니다.	만들어진 DHCP 옵션을 제거합니다.	384 페이지 “DHCP 옵션을 삭제하는 방법(DHCP 관리자)” 384 페이지 “DHCP 옵션을 삭제하는 방법(dhtadm)”

DHCP 옵션을 만들기 전에 다음 표에 나와 있는 옵션 등록 정보를 파악해야 합니다.

표 15–5 DHCP 옵션 등록 정보

옵션 등록 정보	설명
범주	옵션의 **범주**는 다음 중 하나여야 합니다. ■ Vendor(공급업체) – 클라이언트의 공급업체 플랫폼(하드웨어 또는 소프트웨어) 특정 옵션입니다. ■ Site(사이트) – 사이트 특정 옵션입니다. ■ Extend(확장) – DHCP 프로토콜에 추가되었지만 아직 DHCP 표준 옵션으로 구현되지 않은 새 옵션입니다.
코드	**코드**는 옵션에 지정하는 고유한 숫자입니다. 한 옵션 범주 내에서 동일한 코드를 다른 옵션에 사용할 수 없습니다. 다음과 같이 옵션 범주에 해당하는 코드를 사용해야 합니다. ■ Vendor(공급업체) – 각 공급업체 클래스에 대해 코드 값 1–254 ■ Site(사이트) – 코드 값 128–254 ■ Extend(확장) – 코드 값 77–127

표 15–5 DHCP 옵션 등록 정보 *(계속)*

옵션 등록 정보	설명
데이터 유형	**데이터 유형**은 옵션 값으로 지정할 수 있는 데이터 종류를 지정합니다. 다음 목록에는 유효한 데이터 유형이 나와 있습니다. ■ ASCII – 텍스트 문자열 값입니다. ■ BOOLEAN – BOOLEAN 데이터 유형에는 값이 연관되지 않습니다. 이 옵션이 있으면 조건이 true임을 나타내고 옵션이 없으면 조건이 false임을 나타냅니다. 예를 들어, `Hostname` 옵션은 BOOLEAN입니다. 매크로에 `Hostname`이 있으면 DHCP 서버가 지정된 주소와 연관된 호스트 이름을 조회합니다. ■ IP – 점으로 구분된 십진수 형식(*xxx.xxx.xxx.xxx*)의 하나 이상의 IP 주소입니다. ■ OCTET – 이진 데이터의 해석되지 않은 ASCII 표현입니다. 예를 들어, 클라이언트 ID는 OCTET 데이터 유형을 사용합니다. 유효한 문자는 0–9, A–F 및 a–f입니다. 8비트 분량을 나타내려면 두 개의 ASCII 문자가 필요합니다. ■ UNUMBER8, UNUMBER16, UNUMBER32, UNUMBER64, SNUMBER8, SNUMBER16, SNUMBER32 또는 SNUMBER64 – 숫자 값입니다. 맨 앞의 U 또는 S는 숫자가 부호 없는 숫자인지 부호 있는 숫자인지를 나타냅니다. 맨 뒷자리는 숫자에 포함된 비트 수를 나타냅니다.
단위	**단위**는 전체 옵션 값을 나타내는 데 필요한 데이터 유형의 "인스턴스" 수를 지정합니다. 예를 들어, 데이터 유형이 IP이고 단위가 2이면 옵션 값은 두 개의 IP 주소를 포함해야 합니다.
최대	옵션에 지정할 수 있는 값의 최대 수입니다. 예를 들어 최대가 2이고, 단위가 2이고, 데이터 유형이 IP라고 가정합니다. 이 경우 옵션 값은 최대 두 개의 IP 주소 쌍을 포함할 수 있습니다.

표 15–5 DHCP 옵션 등록 정보 *(계속)*

옵션 등록 정보	설명
공급업체 클라이언트 클래스	이 옵션은 옵션 범주가 Vendor(공급업체)인 경우에만 사용할 수 있습니다. 공급업체 클라이언트 클래스는 Vendor(공급업체) 옵션이 연관된 클라이언트 클래스를 식별합니다. 이 클래스는 클라이언트 컴퓨터 유형 또는 운영 체제를 나타내는 ASCII 문자열입니다. 예를 들어, Sun 워크스테이션 일부 모델의 클래스 문자열은 `SUNW.Sun-Blade-100`입니다. 이 유형의 옵션을 사용하면 동일한 클래스의 모든 클라이언트에 전달되고 다른 클래스의 클라이언트에는 전달되지 **않는** 구성 매개변수를 정의할 수 있습니다. 여러 개의 클라이언트 클래스를 지정할 수 있습니다. 지정한 클래스와 일치하는 클라이언트 클래스 값을 가진 DHCP 클라이언트만 해당 클래스에 의해 범위가 지정된 옵션을 받습니다. 클라이언트 클래스는 DHCP 클라이언트의 공급업체에 의해 결정됩니다. Oracle Solaris 클라이언트가 아닌 DHCP 클라이언트의 경우 클라이언트 클래스는 DHCP 클라이언트에 대한 공급업체 설명서를 참조하십시오. Oracle Solaris 클라이언트의 경우 클라이언트에서 `prtconf -b` 명령을 입력하면 공급업체 클라이언트 클래스를 얻을 수 있습니다. 공급업체 클라이언트 클래스를 지정하려면 `uname` 명령에서 반환된 문자열에서 쉼표를 마침표로 대체합니다. 예를 들어, `prtconf -b` 명령에서 `SUNW,Sun-Blade-100`이 반환된 경우 공급업체 클라이언트 클래스를 `SUNW.Sun-Blade-100`으로 지정해야 합니다.

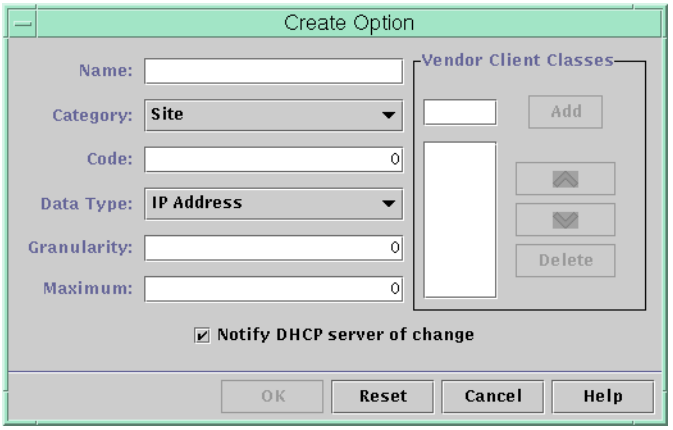
DHCP 옵션 만들기

DHCP 프로토콜에 기존 옵션이 없는 클라이언트 정보를 전달해야 하는 경우 옵션을 만들 수 있습니다. 자체 옵션을 만들기 전에 DHCP에 정의된 모든 옵션 목록을 확인하려면 `dhcp_inittab(4)` 매뉴얼 페이지를 참조하십시오.

`dhtadm -A -s` 명령 또는 DHCP 관리자의 Create Option(옵션 만들기) 대화 상자를 사용하여 새 옵션을 만들 수 있습니다.

다음 그림은 DHCP 관리자의 Create Option(옵션 만들기) 대화 상자를 보여줍니다.

그림 15–17 DHCP 관리자의 Create Option(옵션 만들기) 대화 상자



▼ DHCP 옵션을 만드는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Options(옵션) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **Edit(편집) 메뉴에서 Create(만들기)를 선택합니다.**

Create Option(옵션 만들기) 대화 상자가 열립니다.

3 **새 옵션에 대해 설명이 포함된 짧은 이름을 입력합니다.**

이름에는 최대 128개의 영숫자와 공백을 사용할 수 있습니다.

4 **대화 상자의 각 설정에 대해 값을 입력하거나 선택합니다.**

각 설정에 대한 정보나 DHCP 관리자 도움말을 보려면 표 15–5를 참조하십시오.

5 **옵션 만들기를 마치면 Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**

이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용합니다.

6 **OK(확인)를 누릅니다.**

이제 매크로에 옵션을 추가하고 클라이언트에 전달할 값을 옵션에 지정할 수 있습니다.

▼ DHCP 옵션을 만드는 방법(dhtadm)

1 수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 다음 형식으로 명령을 입력하여 DHCP 옵션을 만듭니다.

```
# dhtadm -A -s option-name -d 'category,code,data-type,granularity,maximum' -g
```

option-name 128자 이하의 영숫자 문자열입니다.

category `Site`, `Extend` 또는 `Vendor=`*list-of-classes* 중 하나입니다. *list-of-classes*는 옵션이 적용되는 공급업체 클라이언트 클래스를 공백으로 구분한 목록입니다. 공급업체 클라이언트 클래스를 결정하는 방법은 표 15–5를 참조하십시오.

code 표 15–5에 설명된 대로 옵션 범주에 적합한 숫자 값입니다.

data-type 표 15–5에 설명된 대로 옵션과 함께 전달되는 데이터 유형을 나타내는 키워드로 지정됩니다.

granularity 표 15–5에 설명된 대로 음수가 아닌 숫자로 지정됩니다.

maximum 표 15–5에 설명된 대로 음수가 아닌 숫자입니다.

예 15–3 dhtadm으로 DHCP 옵션 만들기

다음 명령은 Site(사이트) 범주 옵션인 `NewOpt`라는 옵션을 만듭니다. 옵션 코드는 130입니다. 옵션 값은 하나의 부호 없는 8비트 정수로 설정할 수 있습니다.

```
# dhtadm -A -s NewOpt -d 'Site,130,UNUMBER8,1,1' -g
```

다음 명령은 컴퓨터 유형이 `SUNW,Sun-Blade-100` 또는 `SUNW,Sun-Blade-1000`인 클라이언트에 적용되는 Vendor(공급업체) 범주 옵션인 `NewServ`라는 옵션을 만듭니다. 옵션 코드는 200입니다. 옵션 값은 하나의 IP 주소로 설정할 수 있습니다.

```
# dhtadm -A -s NewServ -d 'Vendor=SUNW.Sun-Blade-100 \
SUNW.Sun-Blade-1000,200,IP,1,1' -g
```

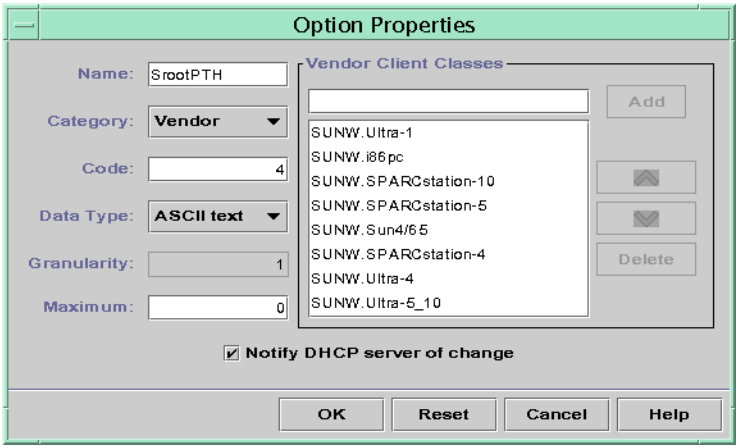
DHCP 옵션 수정

DHCP 서비스에 대해 옵션을 만든 경우 이러한 옵션에 대한 등록 정보를 변경할 수 있습니다. `dhtadm -M -s` 명령 또는 DHCP 관리자의 Option Properties(옵션 등록 정보) 대화 상자를 사용하여 옵션을 수정할 수 있습니다.

DHCP 서비스에 수정한 내용을 반영하도록 DHCP 클라이언트의 옵션 정보를 수정해야 합니다. 385 페이지 “DHCP 클라이언트의 옵션 정보 수정”을 참조하십시오.

다음 그림은 DHCP 관리자의 Option Properties(옵션 등록 정보) 대화 상자를 보여줍니다.

그림 15–18 DHCP 관리자의 Option Properties(옵션 등록 정보) 대화 상자



▼ DHCP 옵션 등록 정보를 수정하는 방법(DHCP 관리자)

1 **DHCP 관리자에서 Options(옵션) 탭을 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **수정할 옵션을 선택합니다.**

3 **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**

Option Properties(옵션 등록 정보) 대화 상자가 열립니다.

4 **필요에 따라 등록 정보를 편집합니다.**

등록 정보에 대한 자세한 내용 또는 DHCP 관리자 도움말을 보려면 표 15–5를 참조하십시오.

5 **옵션 작업을 마치면 Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**

변경 사항은 `dhcptab` 테이블에 적용됩니다. DHCP 서버는 `dhcptab` 테이블을 다시 읽어 변경 사항을 적용하라는 알림을 받습니다.

6 **OK(확인)를 누릅니다.**

▼ DHCP 옵션 등록 정보를 수정하는 방법 (dhtadm)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 “DHCP 명령에 사용자 액세스 설정”을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 옵션을 수정합니다.**

```
# dhtadm -M -s option-name -d 'category,code,data-type,granularity,maximum' -g
```

option-name 변경할 옵션 이름을 지정합니다.

category `Site`, `Extend` 또는 `Vendor=`*list-of-classes*일 수 있습니다. *list-of-classes*는 옵션이 적용되는 공급업체 클라이언트 클래스를 공백으로 구분한 목록입니다. 예를 들어, `SUNW.Sun-Blade-100 SUNW.Ultra-80 SUNWi86pc`가 여기에 해당합니다.

code 표 15–5에 설명된 대로 옵션 범주에 적합한 숫자 값을 지정합니다.

data-type 표 15–5에 설명된 대로 옵션과 함께 전달되는 데이터 유형을 나타내는 키워드를 지정합니다.

granularity 표 15–5에 설명된 대로 음수가 아닌 숫자입니다.

maximum 표 15–5에 설명된 대로 음수가 아닌 숫자입니다.

변경할 등록 정보 뿐이 아니라 `-d` 스위치를 사용하여 모든 DHCP 옵션 등록 정보를 지정해야 합니다.

예 15–4 dhtadm으로 DHCP 옵션 수정

다음 명령은 `NewOpt`라는 이름의 옵션을 수정합니다. 이 옵션은 Site(사이트) 범주 옵션입니다. 옵션 코드는 135입니다. 옵션 값은 하나의 부호 없는 8비트 정수로 설정할 수 있습니다.

```
# dhtadm -M -s NewOpt -d 'Site,135,UNUMBER8,1,1'
```

다음 명령은 Vendor(공급업체) 범주 옵션인 NewServ라는 옵션을 수정합니다. 이제 이 옵션은 컴퓨터 유형이 SUNW,Sun-Blade-100 또는 SUNW,i86pc인 클라이언트에 적용됩니다. 옵션 코드는 200입니다. 옵션 값은 하나의 IP 주소로 설정할 수 있습니다.

```
# dhtadm -M -s NewServ -d 'Vendor=SUNW.Sun-Blade-100 \
SUNW.i86pc,200,IP,1,1' -g
```

DHCP 옵션 삭제

표준 DHCP 옵션은 삭제할 수 없습니다. 하지만 DHCP 서비스에 대해 옵션을 정의한 경우 이러한 옵션은 DHCP 관리자 또는 dhtadm 명령을 사용하여 삭제할 수 있습니다.

▼ DHCP 옵션을 삭제하는 방법(DHCP 관리자)

1. **DHCP 관리자에서 Options(옵션) 탭을 선택합니다.**
   DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.
2. **삭제할 옵션을 선택합니다.**
3. **Edit(편집) 메뉴에서 Delete(삭제)를 선택합니다.**
   Delete Option(옵션 삭제) 대화 상자가 열립니다.
4. **옵션 삭제를 마치면 Notify DHCP Server of Change(DHCP 서버에 변경 사항을 알립니다)를 선택합니다.**
   이렇게 선택하면 OK(확인)를 누른 즉시 DHCP 서버가 dhcptab 테이블을 다시 읽어 변경 사항을 적용합니다.
5. **OK(확인)를 누릅니다.**

▼ DHCP 옵션을 삭제하는 방법(dhtadm)

1. **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**
   DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 DHCP 옵션을 삭제합니다.**

```
# dhtadm -D -s option-name -g
```

DHCP 클라이언트의 옵션 정보 수정

DHCP 서버에 새 DHCP 옵션을 추가하는 경우 각 DHCP 클라이언트의 옵션 정보에 보완 항목을 추가해야 합니다. DHCP 클라이언트가 아닌 클라이언트가 있는 경우 옵션 또는 기호 추가에 대한 정보는 해당 클라이언트 설명서를 참조하십시오.

DHCP 클라이언트에서 `/etc/dhcp/inittab` 파일을 편집하여 DHCP 서버에 추가한 각 옵션에 대한 항목을 추가해야 합니다. 나중에 서버에서 옵션을 수정하는 경우 클라이언트의 `/etc/dhcp/inittab` 파일에 있는 항목도 수정해야 합니다.

`/etc/dhcp/inittab` 파일의 구문에 대한 자세한 내용은 `dhcp_inittab(4)` 매뉴얼 페이지를 참조하십시오.

주 – 이전 Oracle Solaris 릴리스에서 `dhcptags` 파일에 DHCP 옵션을 추가한 경우 `/etc/dhcp/inittab` 파일에 해당 옵션을 추가해야 합니다. 자세한 내용은 450 페이지 "DHCP 옵션 정보"를 참조하십시오.

DHCP 서비스로 Oracle Solaris 네트워크 설치 지원

네트워크의 특정 클라이언트 시스템에서 DHCP를 사용하여 Oracle Solaris를 설치할 수 있습니다. Oracle Solaris 실행에 대한 하드웨어 요구 사항을 충족하는 `sun4u` 기반 시스템 및 x86 시스템에서만 이 기능을 사용할 수 있습니다. DHCP를 사용하여 부트 시 네트워크에 대해 클라이언트 시스템을 자동으로 구성하는 방법은 **Oracle Solaris 10 1/13 설치 설명서: 네트워크 기반 설치**의 2 장, "시스템 구성 정보 미리 구성(작업)"을 참조하십시오.

DHCP는 또한 HTTP를 사용하여 WAN(Wide Area Network)을 통해 서버에서 원격으로 부트 및 설치되는 Oracle Solaris 클라이언트 시스템을 지원합니다. 이러한 원격 부트 및 설치 방법을 *WAN* **부트 설치** 방법이라고 합니다. WAN 부트를 사용하면 신뢰할 수 없는 네트워크 기반구조를 가진 대규모 공개 네트워크에서 SPARC 기반 시스템에 Oracle Solaris를 설치할 수 있습니다. WAN 부트를 보안 기능과 함께 사용하여 데이터 기밀 및 설치 이미지 무결성을 보호할 수 있습니다.

DHCP를 사용하여 WAN 부트를 통해 원격으로 클라이언트 시스템을 부트 및 설치하려면 다음 정보를 클라이언트에 제공하도록 DHCP 서버를 구성해야 합니다.

- 프록시 서버의 IP 주소
- `wanboot-cgi` 프로그램의 위치

이러한 정보를 제공하도록 DHCP 서버를 구성하는 방법은 **Oracle Solaris 10 1/13 설치 설명서: 네트워크 기반 설치**의 2 장, "시스템 구성 정보 미리 구성(작업)"을 참조하십시오. WAN을 통해 DHCP 서버를 사용하여 클라이언트 시스템을 부트 및 설치하는 방법은 **Oracle Solaris 10 1/13 설치 설명서: 네트워크 기반 설치**의 10 장, "WAN 부트(개요)"를 참조하십시오.

디스크가 없는 클라이언트를 지원하는 방법은 386 페이지 "원격 부트 및 디스크가 없는 부트 클라이언트 지원(작업 맵)"을 참조하십시오.

원격 부트 및 디스크가 없는 부트 클라이언트 지원(작업 맵)

DHCP 서비스는 운영 체제 파일을 다른 컴퓨터(OS 서버)에서 원격으로 마운트하는 Oracle Solaris 클라이언트 시스템을 지원할 수 있습니다. 이러한 클라이언트를 대개 **디스크가 없는 클라이언트**라고 합니다. 디스크가 없는 클라이언트는 영구 원격 부트 클라이언트라고 생각할 수 있습니다. 디스크가 없는 클라이언트는 부트할 때마다 클라이언트 운영 체제 파일을 호스트하는 서버의 이름 및 IP 주소를 가져와야 합니다. 그런 다음 디스크가 없는 클라이언트는 이러한 파일에서 원격으로 부트할 수 있습니다.

디스크가 없는 각 클라이언트는 OS 서버에 클라이언트 호스트 이름에 공유되는 자체 루트 분할 영역을 가지고 있습니다. DHCP 서버는 디스크가 없는 클라이언트에 항상 동일한 IP 주소를 반환해야 합니다. 해당 주소는 DNS 등의 이름 서비스에서 동일한 호스트 이름에 매핑된 상태를 유지해야 합니다. 디스크가 없는 클라이언트가 일관된 IP 주소를 받으면 이 클라이언트는 일관된 호스트 이름을 사용하여 OS 서버의 루트 분할 영역에 액세스할 수 있습니다.

IP 주소 및 호스트 이름에 추가하여 DHCP 서버는 디스크가 없는 클라이언트의 운영 체제 파일 위치를 제공할 수 있습니다. 하지만 DHCP 메시지 패킷에 정보를 전달할 옵션 및 매크로를 만들어야 합니다.

다음 작업 맵은 디스크가 없는 클라이언트 또는 다른 영구 원격 부트 클라이언트를 지원하는 데 필요한 작업을 나열합니다. 작업 맵은 작업을 수행하는 데 도움이 되는 절차에 대한 링크도 제공합니다.

작업	설명	수행 방법
Oracle Solaris 서버에서 OS 서비스를 설정합니다.	`smosservice` 명령을 사용하여 클라이언트의 운영 체제 파일을 만듭니다.	**Oracle Solaris 관리: 기본 관리**의 7 장, "디스크가 없는 클라이언트 관리(작업)" `smosservice(1M)` 매뉴얼 페이지도 참조하십시오.
네트워크 부트 클라이언트를 지원하도록 DHCP 서비스를 설정합니다.	DHCP 관리자 또는 `dhtadm` 명령을 사용하여 DHCP 서버가 클라이언트에 부트 정보를 전달하는 데 사용할 수 있는 새 Vendor(공급업체) 옵션 및 매크로를 만듭니다. 네트워크 설치 클라이언트에 대한 옵션을 이미 만든 경우에는 디스크가 없는 클라이언트의 Vendor(공급업체) 클라이언트 유형에 대한 매크로만 만들면 됩니다.	**Oracle Solaris 10 1/13 설치 설명서: 네트워크 기반 설치**의 2 장, "시스템 구성 정보 미리 구성(작업)"
디스크가 없는 클라이언트에 예약된 IP 주소를 지정합니다.	DHCP 관리자를 사용하여 주소를 예약된 주소로 표시하거나 `pntadm` 명령을 사용하여 디스크가 없는 클라이언트에 대해 주소를 `MANUAL`로 표시합니다.	363 페이지 "예약된 IP 주소를 DHCP 클라이언트에 지정"
OS 서비스에 대해 디스크가 없는 클라이언트를 설정합니다.	`smdiskless` 명령을 사용하여 각 클라이언트에 대해 OS 서버에서 운영 체제 지원을 추가합니다. 각 클라이언트에 대해 예약한 IP 주소를 지정합니다.	**Oracle Solaris 관리: 기본 관리**의 7 장, "디스크가 없는 클라이언트 관리(작업)" `smdiskless(1M)` 매뉴얼 페이지도 참조하십시오.

정보만 수신하도록 DHCP 클라이언트 설정(작업 맵)

일부 네트워크에서는 DHCP 서비스가 클라이언트에 구성 정보만 제공해야 할 수 있습니다. 정보만 필요하고 임대는 필요하지 않은 클라이언트 시스템은 DHCP 클라이언트를 사용하여 `INFORM` 메시지를 생성할 수 있습니다. `INFORM` 메시지는 DHCP 서버에 적절한 구성 정보를 클라이언트에 전송하도록 요청합니다.

정보만 필요한 클라이언트를 지원하도록 DHCP 서버를 설정할 수 있습니다. 클라이언트를 호스트하는 네트워크에 해당하는 빈 네트워크 테이블을 만들어야 합니다. 이 테이블이 존재해야 DHCP 서버가 해당 네트워크에서 클라이언트에 응답할 수 있습니다.

다음 작업 맵은 정보 전용 클라이언트를 지원하는 데 필요한 작업을 나열합니다. 작업 맵에는 작업을 수행하는 데 도움이 되는 절차에 대한 링크도 포함되어 있습니다.

작업	설명	수행 방법
빈 네트워크 테이블을 만듭니다.	DHCP 관리자 또는 `pntadm` 명령을 사용하여 정보 전용 클라이언트의 네트워크에 대한 네트워크 테이블을 만듭니다.	340 페이지 “DHCP 네트워크 추가”
클라이언트에 필요한 정보를 포함하는 매크로를 만듭니다.	DHCP 관리자 또는 `dhtadm` 명령을 사용하여 필요한 정보를 클라이언트에 전달하는 매크로를 만듭니다.	372 페이지 “DHCP 매크로 만들기”
DHCP 클라이언트가 `INFORM` 메시지를 생성하게 합니다.	`ifconfig` *int* `dhcp inform` 명령을 사용하여 DHCP 클라이언트가 `INFORM` 메시지를 생성하게 합니다.	403 페이지 “DHCP 클라이언트 시작” 408 페이지 “DHCP 클라이언트에서 사용되는 `ifconfig` 명령 옵션” `ifconfig`(1M) 매뉴얼 페이지

새 DHCP 데이터 저장소로 변환

DHCP는 한 데이터 저장소에서 다른 데이터 저장소로 DHCP 구성 데이터를 변환하는 유틸리티를 제공합니다. 새 데이터 저장소로 변환하는 이유는 몇 가지가 있을 수 있습니다. 예를 들어, DHCP 클라이언트가 많아져서 DHCP 서비스의 성능 또는 용량이 더 필요할 수 있습니다. 또한 DHCP 서버가 하는 일을 여러 서버로 나누어야 할 수도 있습니다. 각 데이터 저장소 유형의 상대적인 장점 및 단점에 대한 비교는 298 페이지 “DHCP 데이터 저장소 선택”을 참조하십시오.

주 – Solaris 8 7/01 릴리스 이전 Oracle Solaris 릴리스에서 업그레이드하는 경우 이 주를 읽어야 합니다.

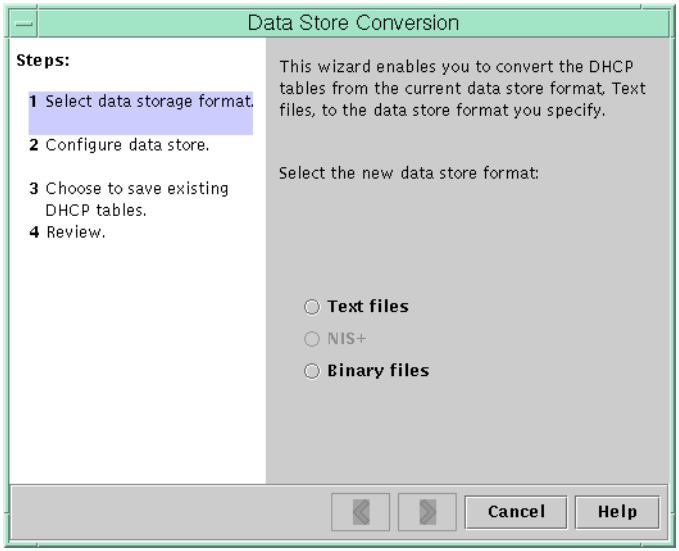
Oracle Solaris 설치 후 DHCP 도구를 실행할 때 새 데이터 저장소로 변환하라는 메시지가 표시됩니다. 변환이 필요한 이유는 파일 및 NIS+에 저장된 데이터 형식이 Solaris 8 7/01 릴리스에서 변경되었기 때문입니다. 새 데이터 저장소로 변환하지 않으면 DHCP 서버는 이전 데이터 테이블을 계속 읽습니다. 하지만 서버는 기존 클라이언트에 대해서만 임대를 연장할 수 있습니다. 새 DHCP 클라이언트를 등록할 수 없거나 이전 데이터 테이블에서 DHCP 관리 도구를 사용할 수 없습니다.

변환 유틸리티는 Sun이 제공하는 데이터 저장소에서 타사 데이터 저장소로 변환하는 사이트에서도 유용합니다. 변환 유틸리티는 기존 데이터 저장소의 항목을 조회하고 동일한 데이터를 포함하는 새 항목을 새 데이터 저장소에 추가합니다. 데이터 저장소 액세스는 각 데이터 저장소에 대해 별도의 모듈에 구현됩니다. 이러한 모듈식 접근 방식으로 인해 변환 유틸리티는 모든 데이터 저장소 형식에서 다른 모든 데이터 저장소 형식으로 DHCP 데이터를 변환할 수 있습니다. 각 데이터 저장소에는 DHCP 서비스가 사용할 수 있는 모듈이 있어야 합니다. 타사 데이터 저장소를 지원하는 모듈을 작성하는 방법은 **Solaris DHCP Service Developer's Guide** 를 참조하십시오.

데이터 저장소 변환은 DHCP 관리자에서 Data Store Conversion(데이터 저장소 변환) 마법사를 통해 수행하거나 `dhcpconfig -C` 명령을 통해 수행할 수 있습니다.

다음 그림은 Data Store Conversion(데이터 저장소 변환) 마법사의 초기 대화 상자를 보여줍니다.

그림 15–19 DHCP 관리자의 Data Store Conversion(데이터 저장소 변환) 마법사 대화 상자



변환을 시작하기 전에 이전 데이터 저장소의 테이블(`dhcptab` 및 네트워크 테이블)을 저장할지 여부를 지정해야 합니다. 그러면 변환 유틸리티는 DHCP 서버를 중지하고, 데이터 저장소를 변환하고, 변환이 성공적으로 완료되면 서버를 다시 시작합니다. 이전 테이블을 저장하도록 지정하지 않은 경우 유틸리티는 변환이 성공했는지 확인한 후 테이블을 삭제합니다. 변환 과정에는 시간이 많이 걸릴 수 있습니다. 변환은 백그라운드에서 실행되고 사용자에게는 진행률을 알려주는 측정기가 표시됩니다.

▼ DHCP 데이터 저장소를 변환하는 방법(DHCP 관리자)

1 **DHCP 관리자의 Service(서비스) 메뉴에서 Convert Data Store(데이터 저장소 변환)를 선택합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

Data Store Conversion(데이터 저장소 변환) 마법사가 열립니다.

2 **마법사에 메시지가 표시되면 대답합니다.**

요청된 정보를 제공하기 어려운 경우 Help(도움말)를 누르면 각 대화 상자에 대한 자세한 정보를 볼 수 있습니다.

3 **선택 사항을 검토한 다음 Finish(마침)를 눌러 데이터 저장소를 변환합니다.**

변환이 완료되면 DHCP 서버가 다시 시작됩니다. 서버는 즉시 새 데이터 저장소를 사용합니다.

▼ DHCP 데이터 저장소를 변환하는 방법(dhcpconfig -C)

1 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

2 **다음 형식으로 명령을 입력하여 데이터 저장소를 변환합니다.**

```
# /usr/sbin/dhcpconfig -C -r resource -p path
```

resource SUNWbinfiles와 같은 새 데이터 저장소 유형입니다.

path /var/dhcp와 같은 데이터에 대한 경로입니다.

변환 후 이전 데이터 저장소에 원래 데이터를 유지하려면 -k 옵션을 지정합니다. 예를 들어, 데이터 저장소를 SUNWbinfiles로 변환하고 이전 데이터 저장소를 저장하려면 다음을 입력합니다.

```
# /usr/sbin/dhcpconfig -C -r SUNWbinfiles -p /var/dhcp -k
```

dhcpconfig 유틸리티에 대한 자세한 내용은 dhcpconfig(1M) 매뉴얼 페이지를 참조하십시오.

DHCP 서버 간 구성 데이터 이동(작업 맵)

DHCP 관리자 및 dhcpconfig 유틸리티를 사용하면 한 DHCP 서버에서 다른 서버로 DHCP 구성 데이터 일부 또는 전부를 이동할 수 있습니다. 전체 네트워크 및 네트워크와 연관된 모든 IP 주소, 매크로 및 옵션을 이동할 수 있습니다. 또는 특정 IP 주소, 매크로 및 옵션을 이동하도록 선택할 수도 있습니다. 첫번째 서버에서 매크로 및 옵션을 제거하지 않고 복사할 수도 있습니다.

다음 작업 중 하나를 수행할 경우 데이터를 이동해야 할 수 있습니다.

- DHCP 작업을 공유하는 서버를 추가합니다.
- DHCP 서버의 시스템을 바꿉니다.
- 같은 데이터 저장소를 사용하면서 데이터 저장소의 경로를 변경합니다.

다음 작업 맵은 DHCP 구성 데이터를 이동할 때 수행해야 할 절차입니다. 맵에는 작업을 수행하는 절차에 대한 링크가 포함되어 있습니다.

작업	설명	수행 방법
1. 첫번째 서버에서 데이터를 내보냅니다.	다른 서버로 이동할 데이터를 선택하고 내보낸 데이터 파일을 만듭니다.	393 페이지 “DHCP 서버에서 데이터를 내보내는 방법(DHCP 관리자)” 393 페이지 “DHCP 서버에서 데이터를 내보내는 방법(dhcpconfig -X)”
2. 두번째 서버로 데이터를 가져옵니다.	내보낸 데이터를 다른 DHCP 서버의 데이터 저장소로 복사합니다.	394 페이지 “DHCP 서버에서 데이터를 가져오는 방법(DHCP 관리자)” 395 페이지 “DHCP 서버에서 데이터를 가져오는 방법(dhcpconfig -I)”
3. 새 서버 환경에 맞게 가져온 데이터를 수정합니다.	새 서버의 정보에 맞게 서버 특정 구성 데이터를 변경합니다.	395 페이지 “가져온 DHCP 데이터를 수정하는 방법(DHCP 관리자)” 396 페이지 “가져온 DHCP 데이터를 수정하는 방법(pntadm, dhtadm)”

DHCP 관리자에서 데이터 내보내기 마법사 및 데이터 가져오기 마법사를 사용하여 한 서버에서 다른 서버로 데이터를 이동합니다. 그런 다음 Macros(매크로) 탭에서 매크로를 수정합니다. 다음 그림은 마법사의 초기 대화 상자를 보여줍니다.

그림 15–20 DHCP 관리자의 Export Data Wizard(데이터 내보내기 마법사) 대화 상자

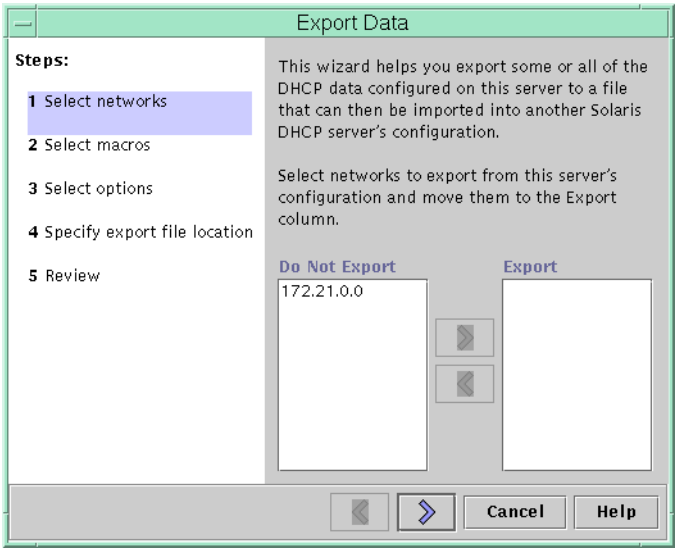
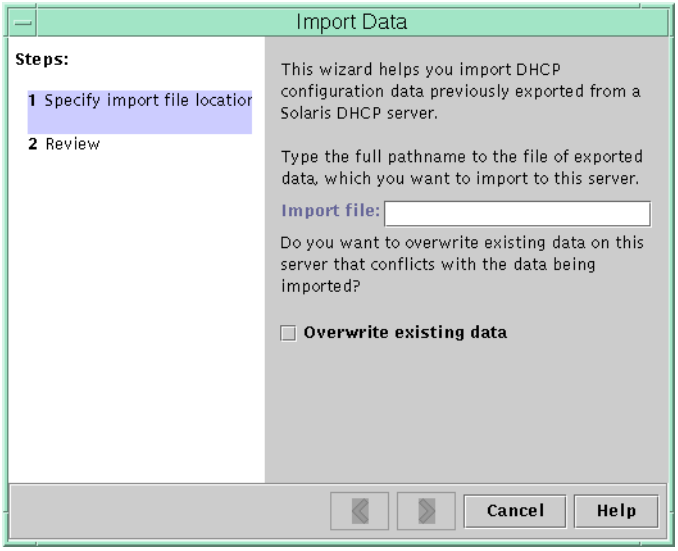


그림 15–21 DHCP 관리자의 Import Data Wizard(데이터 가져오기 마법사) 대화 상자



▼ DHCP 서버에서 데이터를 내보내는 방법(DHCP 관리자)

1 **데이터를 이동하거나 복사할 서버에서 DHCP 관리자를 시작합니다.**
DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.

2 **Service(서비스) 메뉴에서 Export Data(데이터 내보내기)를 선택합니다.**
그림 15–20에 보이는 대로 데이터 내보내기 마법사가 열립니다.

3 **마법사에 메시지가 표시되면 대답합니다.**
마법사가 표시하는 메시지에 대한 자세한 정보를 보려면 Help(도움말)을 누릅니다.

4 **데이터를 가져와야 하는 DHCP 서버가 액세스할 수 있는 파일 시스템으로 내보내기 파일을 이동합니다.**

참조 394 페이지 "DHCP 서버에서 데이터를 가져오는 방법(DHCP 관리자)"에 설명된 대로 데이터를 가져옵니다.

▼ DHCP 서버에서 데이터를 내보내는 방법(dhcpconfig -X)

1 **데이터를 이동하거나 복사할 서버에 로그인합니다.**

2 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**
DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

3 **데이터를 내보냅니다.**
모든 DHCP 데이터를 내보낼 수도 있고, 데이터의 특정 부분을 내보낼 수도 있습니다.

- **특정 주소, 매크로 및 옵션을 내보내려면 다음 형식으로 명령을 입력합니다.**

  # dhcpconfig -X *filename* **-a** *network-addresses* **-m** *macros* **-o** *options*

  *filename*은 압축된 내보낸 데이터를 저장하는 데 사용할 전체 경로 이름입니다. 특정 네트워크 주소, DHCP 매크로 및 DHCP 옵션을 쉼표로 구분된 목록으로 지정합니다. 다음 예는 특정 네트워크, 매크로 및 옵션을 내보내는 방법을 보여줍니다.

```
# dhcpconfig -X /var/dhcp/0dhcp1065_data \
-a 10.63.0.0,10.62.0.0 \
-m 10.63.0.0,10.62.0.0,SUNW.Sun-Blade-100 -o Sterm
```

- **모든 DHCP 데이터를 내보내려면 ALL 키워드를 사용하는 명령을 입력합니다.**

  ```
  # dhcpconfig -X filename -a ALL -m ALL -o ALL
  ```

  *filename*은 압축된 내보낸 데이터를 저장하는 데 사용할 전체 경로 이름입니다. ALL 키워드를 명령 옵션에서 사용하여 모든 네트워크 주소, 매크로 또는 옵션을 내보낼 수 있습니다. 다음 예는 ALL 키워드를 사용하는 방법을 보여줍니다.

  ```
  # dhcpconfig -X /var/dhcp/dhcp1065_data -a ALL -m ALL -o ALL
  ```

참고 – 특정 종류의 데이터를 내보내지 않으려면 해당 데이터 유형에 대해 dhcpconfig 명령 옵션을 지정하지 않으면 됩니다. 예를 들어, -m 옵션을 지정하지 않으면 DHCP 매크로를 내보내지 않습니다.

---

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

4 **데이터를 가져와야 하는 서버가 액세스할 수 있는 위치로 내보내기 파일을 이동합니다.**

참조 395 페이지 “DHCP 서버에서 데이터를 가져오는 방법(dhcpconfig -I)”에 설명된 대로 데이터를 가져옵니다.

▼ DHCP 서버에서 데이터를 가져오는 방법(DHCP 관리자)

1 **DHCP 서버에서 내보낸 데이터를 이동할 서버에서 DHCP 관리자를 시작합니다.**

DHCP 관리자에 대한 자세한 내용은 320 페이지 “DHCP 관리자를 시작 및 중지하는 방법”을 참조하십시오.

2 **Service(서비스) 메뉴에서 Import Data(데이터 가져오기)를 선택합니다.**

그림 15–21에 보이는 대로 데이터 가져오기 마법사가 열립니다.

3 **마법사에 메시지가 표시되면 대답합니다.**

마법사가 표시하는 메시지에 대한 자세한 정보를 보려면 Help(도움말)을 누릅니다.

4 **필요한 경우 가져온 데이터를 수정합니다.**

395 페이지 “가져온 DHCP 데이터를 수정하는 방법(DHCP 관리자)”을 참조하십시오.

▼ DHCP 서버에서 데이터를 가져오는 방법(dhcpconfig -I)

1. **데이터를 가져올 서버에 로그인합니다.**
2. **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**
   DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

   역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.
3. **다음 형식으로 명령을 입력하여 데이터를 가져옵니다.**
   `# dhcpconfig -I` *filename*

   *filename*은 내보낸 데이터를 포함하는 파일의 이름입니다.
4. **필요한 경우 가져온 데이터를 수정합니다.**
   396 페이지 "가져온 DHCP 데이터를 수정하는 방법(pntadm, dhtadm)"을 참조하십시오.

▼ 가져온 DHCP 데이터를 수정하는 방법(DHCP 관리자)

1. **데이터를 가져온 서버에서 DHCP 관리자를 시작합니다.**
   DHCP 관리자에 대한 자세한 내용은 320 페이지 "DHCP 관리자를 시작 및 중지하는 방법"을 참조하십시오.
2. **가져온 데이터에서 수정이 필요한 네트워크 특정 정보를 검토합니다.**
   예를 들어, 네트워크를 이동한 경우 Addresses(주소) 탭을 열고 가져온 네트워크에서 주소의 소유 서버를 변경해야 합니다. Macros(매크로) 탭을 열어 일부 매크로에서 NIS, NIS+ 또는 DNS에 대해 올바른 도메인 이름을 지정해야 할 수도 있습니다.
3. **Addresses(주소) 탭을 열고 가져온 네트워크를 선택합니다.**
4. **모든 주소를 선택하려면 첫번째 주소를 누르고 Shift 키를 누른 상태로 마지막 주소를 누릅니다.**
5. **Edit(편집) 메뉴에서 Properties(등록 정보)를 선택합니다.**
   Modify Multiple Addresses(복수 주소 수정) 대화 상자가 열립니다.
6. **Managing Server(관리 서버) 프롬프트에서 새 서버의 이름을 선택합니다.**

7 **Configuration Macro(구성 매크로) 프롬프트에서 이 네트워크의 모든 클라이언트에 사용되어야 하는 매크로를 선택한 다음 OK(확인)를 누릅니다.**

8 **Macros(매크로) 탭을 엽니다.**

9 **Find(찾기) 버튼을 사용하여 수정된 값이 필요할 수 있는 옵션을 찾습니다.**

Find(찾기) 버튼은 창의 맨 아래에 있습니다.

새 서버에서 수정이 필요할 수 있는 옵션의 예로는 `DNSdmain`, `DNSserv`, `NISservs`, `NIS+serv` 및 `NISdmain` 등이 있습니다.

10 **해당하는 매크로에서 옵션을 변경합니다.**

옵션 변경 절차는 382 페이지 "DHCP 옵션 등록 정보를 수정하는 방법(DHCP 관리자)"을 참조하십시오.

▼ 가져온 DHCP 데이터를 수정하는 방법(pntadm, dhtadm)

1 **데이터를 가져온 서버에 로그인합니다.**

2 **수퍼 유저가 되거나 DHCP 관리 프로파일에 할당된 역할이나 사용자 이름을 맡습니다.**

DHCP 관리 프로파일에 대한 자세한 내용은 321 페이지 "DHCP 명령에 사용자 액세스 설정"을 참조하십시오.

역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"를 참조하십시오.

3 **네트워크 테이블에서 수정해야 하는 데이터가 있는지 검토합니다.**

네트워크를 이동한 경우 `pntadm -P` *network-address* 명령을 사용하여 이동한 네트워크의 네트워크 테이블을 인쇄합니다.

4 **`pntadm` 명령을 사용하여 IP 주소 정보를 수정합니다.**

가져온 주소에 대해 소유 서버 및 구성 매크로를 변경해야 할 수 있습니다. 예를 들어, `10.63.0.2` 주소에 대해 소유 서버(`10.60.3.4`) 및 매크로(`dhcpsrv-1060`)를 변경하려면 다음 명령을 사용합니다.

```
pntadm -M 10.63.0.2 -s 10.60.3.4 -m dhcpsrv-1060 10.60.0.0
```

주소가 많은 경우 각 주소를 수정하는 명령이 포함된 스크립트를 만들어야 합니다. 일괄 처리 모드로 `pntadm`을 실행하는 `pntadm -B` 명령으로 스크립트를 실행합니다. `pntadm`(1M) 매뉴얼 페이지를 참조하십시오.

5 dhcptab 매크로에서 수정이 필요한 값을 가진 옵션이 있는지 검토합니다.

dhtadm -P 명령을 사용하여 전체 dhcptab 테이블을 화면에 표시합니다. grep 또는 다른 도구를 사용하여 변경할 옵션 또는 값을 검색합니다.

6 필요한 경우 dhtadm -M 명령을 사용하여 매크로의 옵션을 수정합니다.

예를 들어, NIS, NIS+ 또는 DNS에 대해 올바른 도메인 이름을 지정하기 위해 일부 매크로를 수정해야 할 수 있습니다. 예를 들어, 다음 명령은 mymacro 매크로에서 DNSdmain 및 DNSserv의 값을 변경합니다.

```
dhtadm -M -m mymacro -e 'DNSserv=dnssrv2:DNSdmain=example.net' -g
```

◆ ◆ ◆ 16장

16 DHCP 클라이언트 구성 및 관리

이 장에서는 Oracle Solaris에 속하는 DHCP(Dynamic Host Configuration Protocol) 클라이언트에 대해 설명합니다. 클라이언트의 DHCPv4 및 DHCPv6 프로토콜이 작동하는 방법과 클라이언트의 동작에 영향을 주는 방법을 설명합니다.

한 프로토콜인 DHCPv4는 Oracle Solaris에 오랫동안 속해 왔으며, 이를 사용하여 DHCP 서버는 IPv4 네트워크 주소와 같은 구성 매개변수를 IPv4 노드로 전달할 수 있습니다.

다른 프로토콜인 DHCPv6을 사용하여 DHCP 서버는 IPv6 네트워크 주소와 같은 구성 매개변수를 IPv6 노드로 전달할 수 있습니다. DHCPv6은 "IPv6 Stateless 주소 자동 구성"(RFC 2462)에 대응하는 Stateful 항목으로, 구성 매개변수를 얻기 위해 Stateless와 별도로 또는 동시에 사용할 수 있습니다.

이 장은 다음 정보를 포함합니다.

DHCP 클라이언트 정보

DHCP 클라이언트는 `dhcpagent` 데몬입니다. Oracle Solaris를 설치하면 DHCP를 사용하여 네트워크 인터페이스를 구성할지 묻는 메시지가 표시됩니다. DHCPv4에 대해 Yes(예)를 지정하면 Oracle Solaris 설치 중 해당 프로토콜이 시스템에서 사용으로 설정됩니다. DHCPv6에 대해서는 설치 시 옵션이 없습니다. 하지만 관련 질문은 IPv6에 대한 것입니다. IPv6을 사용으로 설정하면 DHCPv6을 지원하는 로컬 네트워크에서 DHCPv6도 사용으로 설정됩니다.

Oracle Solaris 클라이언트가 DHCP를 사용하기 위해 다른 필요한 일은 없습니다. DHCP 서버의 구성에 따라 DHCP 서비스를 사용하는 DHCP 클라이언트 시스템에 어떤 정보가 제공될지 결정됩니다.

클라이언트 시스템이 Oracle Solaris를 이미 실행 중이지만 DHCP를 사용 중이 아닌 경우 DHCP를 사용하도록 클라이언트 시스템을 재구성할 수 있습니다. 또한 DHCP 사용을 중지하고 정적 네트워크 정보를 사용하도록 DHCP 클라이언트 시스템을 재구성할 수도 있습니다. 자세한 내용은 407 페이지 “DHCP 클라이언트 사용 및 사용 안함”을 참조하십시오.

DHCPv6 서버

Oracle Solaris에 대해 Sun Microsystems를 통해 사용 가능한 DHCPv6 서버는 없습니다. 타사에서 제공된 서버는 Sun의 DHCPv6과 호환될 수 있고, 네트워크에 DHCPv6 서버가 있는 경우 Sun의 DHCPv6 클라이언트가 이를 사용합니다.

Sun DHCPv4 서버에 대한 자세한 내용은 284 페이지 “DHCP 서버”를 참조하십시오.

DHCPv4와 DHCPv6의 차이점

DHCPv4와 DHCPv6의 두 가지 주요 차이점은 다음과 같습니다.

- **관리 모델**
  - DHCPv4 – 관리자가 각 인터페이스마다 DHCP를 사용으로 설정합니다. 논리적 인터페이스 단위로 관리가 이루어집니다.
  - DHCPv6 – 명시적 구성이 필요하지 않습니다. 이 프로토콜은 주어진 물리적 인터페이스에 사용으로 설정됩니다.
- **프로토콜 세부 정보**
  - DHCPv4 – DHCP 서버가 각 주소에 대한 서브넷 마스크를 제공합니다. 호스트 이름 옵션이 시스템 차원의 노드 이름을 설정합니다.
  - DHCPv6 – DHCPv6 서버가 아닌, Router Advertisements에서 서브넷 마스크를 제공합니다. DHCPv6 호스트 이름 옵션이 없습니다.

DHCP 관리 모델

DHCPv4는 명시적 클라이언트 구성이 필요합니다. 필요한 경우 주소 지정을 위해 DHCPv4 시스템을 설정해야 합니다. 이러한 설정 작업은 대개 초기 시스템 설치 중에 수행되거나 `ifconfig(1M)` 옵션 사용을 통해 동적으로 수행됩니다.

DHCPv6은 명시적 클라이언트 구성이 필요하지 않습니다. 대신, DHCP 사용이 네트워크의 등록 정보이고 이를 사용하는 신호가 로컬 라우터에서 Router Advertisement 메시지에 전달됩니다. DHCP 클라이언트는 필요에 따라 자동으로 논리적 인터페이스를 만들고 제거합니다.

DHCPv6 방식은 기존의 IPv6 Stateless (자동) 주소 구성과 관리상 매우 비슷합니다. Stateless 주소 구성의 경우 로컬 라우터에 플래그를 설정하여 주어진 접두어 세트에 대해 각 클라이언트가 보급된 접두어에 로컬 인터페이스 토큰이나 난수를 더해서 자체에 주소를 자동으로 구성해야 합니다. DHCPv6의 경우 동일한 접두어가 필요하지만 주소가 "무작위로" 지정되는 대신 DHCPv6 서버를 통해 획득, 관리됩니다.

MAC 주소 및 클라이언트 ID

DHCPv4는 MAC 주소 및 주소 지정 목적으로 클라이언트를 식별하는 선택적 클라이언트 ID를 사용합니다. 동일한 클라이언트가 네트워크에 도착할 때마다 가능하면 동일한 주소를 얻습니다.

DHCPv6은 기본적으로 동일한 체계를 사용하지만 클라이언트 ID가 필수이고 거기에 구조를 강제 적용합니다. DHCPv6의 클라이언트 ID는 DUID(DHCP Unique Identifier) 및 IAID(Identity Association Identifier)의 두 부분으로 구성됩니다. DUID는 (DHCPv4에서처럼 단지 인터페이스가 아닌) 클라이언트 **시스템**을 식별하고 IAID는 해당 시스템의 인터페이스를 식별합니다.

RFC 3315에 기술된 대로, ID 연관은 서버 및 클라이언트에서 관련된 IPv6 주소 세트를 식별, 그룹화, 관리하기 위해 사용되는 수단입니다. 클라이언트는 적어도 하나의 별개의 IA를 각 네트워크 인터페이스와 연관시키고, 지정된 IA를 사용하여 해당 인터페이스의 서버에서 구성 정보를 얻어야 합니다. IA에 대한 추가 정보는 다음 절인 "프로토콜 세부 정보"를 참조하십시오.

DUID+IAID를 DHCPv4와 함께 사용할 수도 있습니다. 이들은 클라이언트 ID로 작동할 수 있도록 분명하게 서로 연결할 수 있습니다. 호환성 이유로 일반 IPv4 인터페이스에는 수행되지 않습니다. 그러나 논리적 인터페이스의 경우(`hme0:1`) 구성된 클라이언트 ID가 없으면 DUID+IAID가 사용됩니다.

IPv4 DHCP와 달리, DHCPv6은 “클라이언트 이름” 옵션을 제공하지 않으므로 DHCPv6 혼자만 기반으로 시스템에 이름을 지정할 방법이 없습니다. 대신, DHCPv6에서 제공된 주소와 어울리는 DNS 이름을 알아야 하는 경우 해당하는 이름 정보를 찾으려면 DNS 역분석(`getaddrinfo(3SOCKET)` 함수를 통해 주소-이름 질의)을 사용하십시오. 따라서 DHCPv6만 사용하는데 특정 이름을 가진 노드가 필요한 경우 시스템에서 `/etc/nodename`을 설정해야 한다는 것을 알 수 있습니다.

프로토콜 세부 정보

DHCPv4에서는 DHCP 서버가 지정된 주소에 사용할 서브넷 마스크를 제공합니다. DHCPv6에서는 서브넷 마스크(“접두어 길이”라고도 함)가 Router Advertisements로 지정되고 DHCP 서버에서 제어하지 않습니다.

DHCPv4는 시스템 차원의 노드 이름을 설정하는 데 사용되는 호스트 이름 옵션을 전달합니다. DHCPv6에는 해당 옵션이 없습니다.

DHCPv6용 클라이언트 ID를 구성하려면 시스템에서 자동 선택을 허용하기보다는 DUID를 지정해야 합니다. 이는 데몬에 대해 전역적으로 또는 인터페이스 단위로 수행할 수 있습니다. 다음 형식을 사용하여 전역 DUID를 설정합니다(처음의 점 주의).

.v6.CLIENT_ID=*DUID*

특정 인터페이스에서 주어진 DUID를 사용하도록 설정하려면(시스템에서 다중 독립 클라이언트가 DHCPv6 서버로 보임) 다음을 사용합니다.

hme0.v6.CLIENT ID=*DUID*

각 ID 연관(IA)은 한가지 유형의 주소를 보유합니다. 예를 들어, 임시 주소용 ID 연관(IA_TA)은 임시 주소를 보유하고 비임시 주소용 ID 연관(IA_NA)은 영구적인 지정된 주소를 전달합니다. 이 설명서에 기술된 DHCPv6 버전은 IA_NA 연관만 제공합니다.

Oracle Solaris는 요청 시 정확히 하나의 IAID를 각 인터페이스에 지정하고 IAID는 루트 파일 시스템의 파일에 저장되므로 시스템 전체 수명 동안 일정하게 유지됩니다.

논리적 인터페이스

DHCPv4 클라이언트에서 각 논리적 인터페이스는 독립적이며 관리 단위입니다. 0번째 논리적 인터페이스에 더해서(식별자로 인터페이스 MAC 주소가 기본 설정) 사용자는 `dhcpagent` 구성 파일에서 CLIENT_ID를 지정하여 특정 논리적 인터페이스에서 DHCP가 실행되도록 구성할 수 있습니다. 예를 들면 다음과 같습니다.

```
hme0:1.CLIENT_ID=orangutan
```

DHCPv6은 다르게 작동합니다. IPv4와 달리, IPv6 인터페이스의 0번째 논리적 인터페이스는 항상 링크 로컬입니다. 링크 로컬을 사용하면 DHCP 서버와 같은 사용 가능한 지정 방법이 없을 때 IP 네트워크의 장치에 IP 주소를 자동으로 지정할 수 있습니다. 0번째 논리적 인터페이스를 DHCP 통제하에 놓을 수 없으므로 DHCPv6이 0번째 논리적 인터페이스("물리적" 인터페이스라고도 함)에서 실행되더라도 0이 아닌 논리적 인터페이스에만 주소가 지정됩니다.

DHCPv6 클라이언트 요청에 대한 응답으로 DHCPv6 서버는 구성할 클라이언트에 대한 주소 목록을 반환합니다.

옵션 협상

DHCPv6에는 클라이언트가 선호하는 내용을 서버에 힌트로 알려주는 Option Request Option이 있습니다. 모든 가능한 옵션을 서버에서 클라이언트로 보낸 경우 그 중 일부가 클라이언트로 가능 도중에 삭제될 것이라는 정보를 보낼 수 있습니다. 서버는 힌트를

사용하여 회신에 포함할 옵션을 고를 수 있습니다. 다른 방법으로, 서버가 힌트를 무시하고 다른 항목을 고를 수 있습니다. 예를 들어, Oracle Solaris에서 선호 옵션이 Oracle Solaris DNS 주소 도메인 또는 NIS 주소 도메인을 포함할 수 있지만, net BIOS 서버를 포함하지는 않습니다.

DHCPv4에도 동일한 유형의 힌트가 제공되지만 특수한 Option Request Option이 없습니다. 대신, DHCPv4는 `/etc/default/dhcpagent`의 `PARAM_REQUEST_LIST`를 사용합니다.

구성 구문

`/etc/default/dhcpagent`를 사용하여 기존 DHCPv4 클라이언트와 동일한 방법으로 DHCPv6 클라이언트를 구성합니다.

구문의 인수는 인터페이스 이름(있는 경우)과 구성될 매개변수 사이에 ".v6" 표시자로 지정됩니다. 예를 들어, 전역 IPv4 옵션 요청 목록은 다음과 같이 설정됩니다.

```
PARAM_REQUEST_LIST=1,3,6,12,15,28,43
```

다음과 같이 개별 인터페이스에서 호스트 이름 옵션을 생략하도록 구성할 수 있습니다.

```
hme0.PARAM_REQUEST_LIST=1,3,6,15,28,43
```

DHCPv6의 전역 요청 목록을 설정하려면 선행 점에 주의하십시오.

```
.v6.PARAM_REQUEST_LIST=23,24
```

또는, 개별 인터페이스를 설정하려면 다음 예제를 따르십시오.

```
hme0.v6.PARAM_REQUEST_LIST=21,22,23,24
```

참조용으로 여기에 DHCPv6 구성의 실제 `/etc/default/dhcpagent` 파일이 있습니다.

```
# The default DHCPv6 parameter request list has preference (7), unicast (12),
# DNS addresses (23), DNS search list (24), NIS addresses (27), and
# NIS domain (29).  This may be changed by altering the following parameter-
# value pair.  The numbers correspond to the values defined in RFC 3315 and
# the IANA dhcpv6-parameters registry.
.v6.PARAM_REQUEST_LIST=7,12,23,24,27,29
```

DHCP 클라이언트 시작

대부분의 경우 DHCPv6 클라이언트 시작을 위해 아무것도 필요하지 않습니다. `in.ndpd` 데몬이 필요할 때 자동으로 DHCPv6을 시작합니다. `/etc/hostname6.$IFNAME`을 수정하여 부트 시 인터페이스가 IPv6에 연결되도록 구성해야 할 수 있습니다. 하지만 설치 시 시스템에서 IPv6을 사용으로 설정하는 경우에는 설치 프로그램에서 이를 자동으로 수행합니다.

그러나 DHCPv4의 경우 Oracle Solaris 설치 중에 수행되지 않았다면 클라이언트 시작을 요청해야 합니다. 407 페이지 “DHCP 클라이언트를 사용으로 설정하는 방법”을 참조하십시오.

dhcpagent 데몬은 시스템 부트와 관련한 다른 프로세스에서 필요한 구성 정보를 얻습니다. 이러한 이유로 시스템 시작 스크립트가 부트 프로세스에서 조기에 dhcpagent를 시작하고 DHCP 서버에서 네트워크 구성 정보가 도착할 때까지 기다립니다.

기본값은 DHCPv6을 실행하는 것이지만 DHCPv6이 실행되지 않도록 선택할 수 있습니다. DHCPv6이 실행되기 시작한 후 ifconfig 명령을 사용하여 이를 중지할 수 있습니다. /etc/inet/ndpd.conf 파일을 수정하여 DHCPv6이 재부트 시 시작되지 않도록 사용 안함으로 설정할 수도 있습니다.

다음 예는 hme0이라는 인터페이스에서 DHCPv6을 즉시 종료하는 방법을 보여줍니다.

```
ex# echo ifdefault StatefulAddrConf false >> /etc/inet/ndpd.conf
ex# pkill -HUP -x in.ndpd
ex# ifconfig hme0 inet6 dhcp release
```

/etc/dhcp.*interface* 파일(예를 들어, Sun Fire 880 시스템의 /etc/dhcp.ce0)의 존재는 지정된 인터페이스에서 DHCPv4가 사용될 것임을 시작 스크립트에 알립니다. dhcp.*interface* 파일을 찾으면 시작 스크립트는 dhcpagent를 시작합니다.

시작 후 dhcpagent는 네트워크 인터페이스를 구성하라는 지시를 받을 때까지 대기합니다. 시작 스크립트는 281 페이지 “DHCP의 작동 방식”에 설명된 대로 DHCPv4를 시작하도록 dhcpagent에 지시하는 ifconfig *interface* dhcp start 명령을 실행합니다. 명령이 dhcp.*interface* 파일에 포함되어 있는 경우 해당 명령은 ifconfig의 dhcp start 옵션에 추가됩니다. ifconfig *interface* dhcp 명령에서 사용되는 옵션에 대한 자세한 내용은 ifconfig(1M) 매뉴얼 페이지를 참조하십시오.

DHCPv6 통신

수동 구성으로 호출된 DHCPv4와 달리, DHCPv6은 RA(Router Advertisements)로 호출됩니다. 라우터 구성 방법에 따라 시스템이 Router Advertisement 메시지가 수신된 인터페이스에서 DHCPv6을 자동으로 호출하고 DHCP를 사용하여 주소나 기타 매개변수를 얻거나, 또는 시스템이 DHCPv6을 사용하여 주소 이외의 데이터(예: DNS 서버)만 요청합니다.

in.ndpd 데몬이 Router Advertisement 메시지를 수신합니다. 이는 시스템에서 IPv6용으로 배관된 모든 인터페이스에서 자동으로 수행됩니다. in.ndpd가 DHCPv6이 실행되도록 지정하는 RA를 발견하면 이를 호출합니다.

in.ndpd에서 DHCPv6이 시작하지 못하도록 하려면 /etc/inet/ndpd.conf 파일을 변경할 수 있습니다.

다음 버전의 ifconfig를 사용하여 DHCPv6이 시작된 후 중지할 수도 있습니다.

```
ifconfig <interface>inet6 dhcp drop
```

또는

```
ifconfig <interface> inet6 dhcp release
```

DHCP 클라이언트 프로토콜이 네트워크 구성 정보를 관리하는 방법

DHCPv4 및 DHCPv6 클라이언트 프로토콜은 여러 가지 방법으로 네트워크 구성 정보를 관리합니다. 주요 차이점은, DHCPv4에서는 단일 주소의 임대 및 이와 어울리는 옵션을 협상하는 것입니다. DHCPv6에서는 일괄 주소 및 일괄 옵션에 걸쳐 협상이 이루어집니다.

DHCPv4 클라이언트와 서버 간의 상호 작용에 대한 배경 정보는 12 장, “DHCP 정보(개요)”를 참조하십시오.

DHCPv4 클라이언트가 네트워크 구성 정보를 관리하는 방법

DHCP 서버에서 정보 패킷을 얻은 후에 dhcpagent가 네트워크 인터페이스를 구성하고 인터페이스를 가져옵니다. 데몬이 IP 주소에 대한 임대 기간 동안 인터페이스를 제어하고 내부 테이블에서 구성 데이터를 유지 관리합니다. 시스템 시작 스크립트가 dhcpinfo 명령을 사용하여 내부 테이블에서 구성 옵션 값을 추출합니다. 값을 사용하여 시스템을 구성하고 네트워크에서 통신이 가능합니다.

dhcpagent 데몬은 시간이 경과할 때까지(대개 임대 시간의 절반) 수동적으로 기다립니다. 그런 다음 데몬이 DHCP 서버에서 임대 연장을 요청합니다. 시스템에서 인터페이스가 작동 중지되거나 IP 주소가 변경되었다고 dhcpagent에 알리면 ifconfig 명령에서 별도로 지시할 때까지 데몬이 인터페이스를 제어하지 않습니다. 인터페이스가 작동 중이고 IP 주소가 변경되지 않았음을 dhcpagent가 알게 되면 데몬이 서버에 임대 갱신 요청을 보냅니다. 임대를 갱신할 수 없으면 dhcpagent가 임대 시간 끝에 인터페이스를 끌어내립니다.

dhcpagent가 임대에 관련된 조치를 실행할 때마다 데몬이 /etc/dhcp/eventhook라는 실행 파일을 찾습니다. 이 이름을 가진 실행 파일을 찾으면 dhcpagent가 실행 파일을 호출합니다. 이벤트 실행 파일 사용에 대한 자세한 내용은 418 페이지 “DHCP 클라이언트 이벤트 스크립트”를 참조하십시오.

DHCPv6 클라이언트가 네트워크 구성 정보를 관리하는 방법

클라이언트와 서버 간의 DHCPv6 통신은 클라이언트가 서버를 찾기 위해 Solicit 메시지를 발송하는 것으로 시작합니다. 응답에서 DHCP 서비스에 사용 가능한 모든 서버가 Advertise 메시지를 보냅니다. 서버 메시지는 여러 IA_NA(Identity Association Non-Temporary Address) 레코드와 기타 서버가 제공할 수 있는 옵션(예: DNS 서버 주소)을 포함합니다.

클라이언트가 Request 메시지에 고유의 IA_NA/IAADDR 레코드를 설정하여 특정 주소(및 이것의 배수)를 요청할 수 있습니다. 클라이언트는 일반적으로 이전 주소가 기록된 경우 특정 주소를 요청하고, 서버는 가능하면 똑같은 것을 제공합니다. 클라이언트가 무엇이든 관계없이(주소를 전혀 요청하지 않더라도) 단일 DHCPv6 트랜잭션에 대해 서버가 원하는 수의 주소를 클라이언트에 제공할 수 있습니다.

이것은 클라이언트와 서버 간에 발생하는 메시지 대화입니다.

- 클라이언트가 서버를 찾기 위해 Solicit 메시지를 보냅니다.
- 서버가 Advertise 메시지를 보내어 DHCP 서비스에 사용 가능함을 나타냅니다.
- 클라이언트가 Request 메시지를 보내어 가장 큰 선호 값으로 서버로부터 IP 주소를 포함한 구성 매개변수를 요청합니다. 서버 선호 값이 관리자에 의해 설정되고 하한값 0부터 상한값 255까지 확장됩니다.
- 서버가 주소 임대 및 구성 데이터를 포함하는 Reply 메시지를 보냅니다.

Advertise 메시지의 선호 값이 255이면 DHCPv6 클라이언트가 해당 서버를 즉시 선택합니다. 가장 선호되는 서버가 응답하지 않거나 Request 메시지에 성공적인 Reply를 실패하면 더 이상 Advertise 메시지를 구할 수 없을 때까지 (순서대로) 덜 선호되는 서버를 계속 찾습니다. 이 시점에서 클라이언트가 Solicit 메시지를 다시 보내어 시작합니다.

선택한 서버가 Solicit 또는 Request 메시지에 대한 응답으로 지정된 주소 및 구성 매개변수를 포함하는 Reply 메시지를 보냅니다.

DHCP 클라이언트 종료

종료 시, 클라이언트가 Release 메시지를 클라이언트에 주소를 지정한 서버에 보내어 클라이언트가 더 이상 하나 이상의 지정된 주소를 사용하지 않음을 나타냅니다. DHCPv4 클라이언트 시스템이 정상적으로 종료할 때 `dhcpagent`가 현재 구성 정보를 파일(있는 경우)에 작성합니다. DHCPv4의 파일 이름은 `/etc/dhcp/`*interface*`.dhc`이고 DHCPv6의 파일 이름은 `/etc/dhcp/`*interface*`.dh6`입니다. 기본적으로 임대는 해제가 아니라 저장되므로 DHCP 서버에서 IP 주소가 활성 사용 중이 아님을 감지할 수 없습니다. 따라서 클라이언트가 다음 부트 시 주소를 쉽게 되찾을 수 있습니다. 이 기본 작업은 `ifconfig <interface> dhcp drop` 명령과 동일합니다.

시스템을 재부트할 때 해당 파일의 임대가 여전히 유효하면 `dhcpagent`가 동일한 IP 주소 및 네트워크 구성 정보를 사용하도록 약식 요청을 보냅니다. DHCPv4의 경우 이것은 Request 메시지입니다. DHCPv6의 경우 Confirm 메시지입니다.

DHCP 서버가 이 요청을 허가하면 `dhcpagent`가 시스템을 종료할 때 디스크에 작성된 정보를 사용할 수 있습니다. 서버가 클라이언트의 정보 사용을 허가하지 않으면 `dhcpagent`가 281 페이지 “DHCP의 작동 방식”에 설명된 DHCP 프로토콜 시퀀스를 시작합니다. 그 결과, 클라이언트가 새 네트워크 구성 정보를 얻습니다.

DHCP 클라이언트 사용 및 사용 안함

Oracle Solaris를 이미 실행 중이고 DHCP를 사용 중이 아닌 시스템에서 DHCP 클라이언트를 사용으로 설정하려면 먼저 시스템 구성을 해제해야 합니다. 시스템을 부트할 때 시스템을 설정하고 DHCP 클라이언트를 사용으로 설정하려면 몇 가지 명령을 실행해야 합니다.

주 – 대부분의 배치에서 흔한 방법은 DHCP를 사용하기보다, 기반구조의 중요한 부분을 정적 IP 주소로 설정하는 것입니다. 네트워크의 어떤 장치(예: 라우터 및 특정 서버)가 클라이언트여야 하고 어떤 것이 안되는지 결정하는 것은, 이 설명서의 범위를 벗어납니다.

▼ DHCP 클라이언트를 사용으로 설정하는 방법

이 절차는 DHCPv4가 Oracle Solaris 설치 중 사용으로 설정되지 않은 경우에만 필요합니다. DHCPv6에는 필요 없습니다.

1 **클라이언트 시스템에 수퍼 유저로 로그인합니다.**

2 **이 시스템에서 대화형 구성 대신 사전 구성을 사용하는 경우 `sysidcfg` 파일을 편집합니다. `sysidcfg` 파일에서 `network_interface` 키워드에 `dhcp` 하위 키를 추가합니다.**

예를 들어, `network_interface=hme0 {dhcp}`와 같이 추가합니다. 자세한 내용은 `sysidcfg(4)`매뉴얼 페이지를 참조하십시오.

3 **시스템 구성을 해제하고 종료합니다.**

```
# sys-unconfig
```

이 명령에 의해 제거되는 구성 정보에 대한 자세한 내용은 `sys-unconfig(1M)` 매뉴얼 페이지를 참조하십시오.

4 **종료가 완료되면 시스템을 재부트합니다.**

시스템이 사전 구성을 사용하는 경우 `sysidcfg` 파일의 `dhcp` 하위 키는 부트할 때 DHCP 클라이언트를 사용하도록 시스템을 구성합니다.

시스템에서 사전 구성을 사용하지 않는 경우 시스템 재부트 시 `sysidtool` 프로그램에서 시스템 구성 정보를 입력하라는 메시지를 표시합니다. 자세한 내용은 `sysidtool(1M)` 매뉴얼 페이지를 참조하십시오.

5 **DHCP를 사용하여 네트워크 인터페이스를 구성할지 묻는 메시지가 표시되면 Yes(예)를 지정합니다.**

▼ DHCP 클라이언트를 사용 안함으로 설정하는 방법

1. **클라이언트 시스템에 수퍼 유저로 로그인합니다.**

2. **sysidcfg 파일을 사용하여 시스템을 사전 구성하는 경우 network_interface 키워드에서 dhcp 하위 키를 제거합니다.**

3. **시스템 구성을 해제하고 종료합니다.**

   ```
   # sys-unconfig
   ```

   이 명령에 의해 제거되는 구성 정보에 대한 자세한 내용은 sys-unconfig(1M) 매뉴얼 페이지를 참조하십시오.

4. **종료가 완료되면 시스템을 재부트합니다.**

   시스템에서 사전 구성을 사용하는 경우 구성 정보를 입력하라는 메시지가 표시되지 않고 DHCP 클라이언트가 구성되지 않습니다.

   시스템에서 사전 구성을 사용하지 않는 경우 시스템 재부트 시 sysidtool 프로그램에서 시스템 구성 정보를 입력하라는 메시지를 표시합니다. 자세한 내용은 sysidtool(1M) 매뉴얼 페이지를 참조하십시오.

5. **DHCP를 사용하여 네트워크 인터페이스를 구성할지 묻는 메시지가 표시되면 No(아니오)를 지정합니다.**

DHCP 클라이언트 관리

DHCP 클라이언트 소프트웨어는 정상적인 시스템 운영하에서 관리가 필요하지 않습니다. dhcpagent 데몬은 시스템을 부트할 때 자동으로 시작하고, 임대를 재협상하고, 시스템을 종료할 때 중지합니다. 직접 dhcpagent 데몬을 수동으로 시작 및 중지하면 안됩니다. 대신, 클라이언트 시스템에서 수퍼 유저로 ifconfig 명령을 사용하여 필요한 경우 dhcpagent의 네트워크 인터페이스 관리에 영향을 미칠 수 있습니다.

DHCP 클라이언트에서 사용되는 ifconfig 명령 옵션

이 절은 ifconfig(1M) 매뉴얼 페이지에 문서화된 명령 옵션을 요약합니다. 이 명령의 DHCPv4 버전과 DHCPv6 버전의 유일한 차이는 "inet6" 키워드입니다. DHCPv6의 경우 "inet6" 키워드를 포함하고 DHCPv4를 실행할 때는 제외하십시오.

ifconfig 명령을 통해 다음을 수행할 수 있습니다.

- **DHCP 클라이언트 시작** – ifconfig *interface* [inet6] dhcp start 명령은 dhcpagent와 DHCP 서버 간 IP 주소 및 새로운 구성 옵션 세트를 가져오기 위한 상호 작용을 시작합니다. 이 명령은 IP 주소를 추가하거나 서브넷 마스크를 변경할 때와 같이 클라이언트가 즉시 사용할 정보를 변경할 때 유용합니다.

- **네트워크 구성 정보만 요청** – `ifconfig` *interface* `[inet6] dhcp inform` 명령은 `dhcpagent`가 IP 주소를 제외한 네트워크 구성 매개변수 요청을 실행하도록 합니다. 이 명령은 네트워크 인터페이스에 정적 IP 주소가 있지만 클라이언트 시스템에 업데이트된 네트워크 옵션이 필요할 때 유용합니다. 예를 들어, 이 명령은 DHCP를 IP 주소 관리에 사용하지 않지만 네트워크의 호스트 구성에 사용하려는 경우 유용합니다.
- **임대 연장 요청** – `ifconfig` *interface* `[inet6] dhcp extendipadm refresh-addr` *dhcp-addrobj* 명령은 `dhcpagent`가 임대 갱신 요청을 실행하도록 합니다. 클라이언트가 자동으로 임대를 갱신하도록 요청합니다. 그러나 임대 시간을 변경한 후에 다음 임대 갱신 시도를 기다리지 않고 새 임대 시간을 즉시 사용하도록 하려면 이 명령을 사용할 수 있습니다.
- **IP 주소 해제** – `ifconfig` *interface* `[inet6] dhcp release` 명령은 `dhcpagent`가 네트워크 인터페이스에서 사용된 IP 주소를 양도하도록 합니다. IP 주소 해제는 임대가 만료될 때 자동으로 발생합니다. 랩탑에서 네트워크를 남겨 두고 새 네트워크에서 시스템을 시작하려고 할 때 이 명령을 실행할 수 있습니다. `/etc/default/dhcpagent` 구성 파일 `RELEASE_ON_SIGTERM` 등록 정보를 참조하십시오.
- **IP 주소 삭제** – `ifconfig` *interface* `[inet6] dhcp drop` 명령은 `dhcpagent`가 DHCP 서버에 알리지 않고 네트워크 인터페이스를 끌어내리고 파일 시스템에 임대를 캐시하도록 합니다. 이 명령으로 클라이언트는 재부트할 때 동일한 IP 주소를 사용할 수 있습니다.
- **네트워크 인터페이스 핑** – `ifconfig` *interface* `[inet6] dhcp ping` 명령을 사용하면 인터페이스가 DHCP의 제어를 받는지 여부를 파악할 수 있습니다.
- **네트워크 인터페이스의 DHCP 구성 상태 보기** – `ifconfig` *interface* `[inet6] dhcp status` 명령은 DHCP 클라이언트의 현재 상태를 표시합니다. 다음 항목이 표시됩니다.
  - IP 주소가 클라이언트에 바인드되었는지 여부
  - 전송, 수신 및 거부된 요청 수
  - 이 인터페이스가 기본 인터페이스인지 여부
  - 임대를 획득한 시간, 만료되는 시간 및 갱신 시도가 시작되도록 예약된 시간

  예를 들면 다음과 같습니다.

```
# ifconfig hme0 dhcp status
Interface  State         Sent  Recv  Declined  Flags
hme0       BOUND            1     1         0   [PRIMARY]
(Began,Expires,Renew)=(08/16/2005 15:27, 08/18/2005 13:31, 08/17/2005 15:24)

# ifconfig hme0 inet6 dhcp status
Interface  State         Sent  Recv  Declined  Flags
hme0       BOUND            1     0         0   [PRIMARY]
(Began,Expires,Renew)=(11/22/2006 20:39, 11/22/2006 20:41, 11/22/2006 20:40)
```

DHCP 클라이언트 구성 매개변수 설정

클라이언트 시스템의 /etc/default/dhcpagent 파일은 dhcpagent의 조정 가능한 매개변수를 포함합니다. 텍스트 편집기를 사용하여 클라이언트 운영에 영향을 주는 여러 매개변수를 변경할 수 있습니다. /etc/default/dhcpagent 파일은 잘 문서화되어 있으므로 자세한 내용은 이 파일과 dhcpagent(1M) 매뉴얼 페이지를 참조해야 합니다.

/etc/dhcp.*interface* 파일은 DHCP 클라이언트에 영향을 미치는 매개변수가 설정되는 다른 위치입니다. 이 파일에 설정된 매개변수는 ifconfig 명령을 통해 시스템 시작 스크립트에서 사용됩니다. 하지만 이 파일은 DHCPv4에만 영향을 줍니다. DHCPv6에는 여기에 해당하는 파일이 없습니다.

기본적으로 DHCP 클라이언트는 다음과 같이 구성됩니다.

DHCPv4의 경우

- 클라이언트 시스템에 특정 호스트 이름이 필요하지 않습니다.

  클라이언트가 특정 호스트 이름을 요청하도록 하려면 412 페이지 "DHCPv4 클라이언트 호스트 이름"을 참조하십시오.

- 클라이언트의 기본 요청은 /etc/default/dhcpagent 에 제공되고 DNS 서버, DNS 도메인 및 브로드캐스트 주소를 포함합니다.

  /etc/default/dhcpagent 파일의 PARAM_REQUEST_LIST 키워드에서 DHCP 클라이언트의 매개변수 파일이 더 많은 옵션을 요청하도록 설정할 수 있습니다. DHCP 서버가 특별히 요청되지 않은 옵션을 제공하도록 구성할 수 있습니다. DHCP 서버 매크로를 사용하여 클라이언트에 정보를 보내는 방법은 dhcpd(8) 매뉴얼 페이지 및 366 페이지 "DHCP 매크로 작업(작업 맵)"을 참조하십시오.

DHCPv4 및 DHCPv6

- 클라이언트 시스템이 하나의 물리적 네트워크 인터페이스에서 DHCP를 사용합니다.

  여러 개의 물리적 네트워크 인터페이스에서 DHCP를 사용하려면 411 페이지 "다중 네트워크 인터페이스의 DHCP 클라이언트 시스템"을 참조하십시오.

- DHCP 클라이언트가 Oracle Solaris 설치 후에 구성된 경우 이름 서비스 클라이언트로 자동으로 구성되지 않습니다.

  DHCP 클라이언트에서 이름 서비스 사용에 대한 자세한 내용은 413 페이지 "DHCP 클라이언트 시스템 및 이름 서비스"를 참조하십시오.

다중 네트워크 인터페이스의 DHCP 클라이언트 시스템

DHCP 클라이언트는 한 시스템에서 여러 다른 인터페이스를 동시에 관리할 수 있습니다. 인터페이스는 물리적 인터페이스 또는 논리적 인터페이스일 수 있습니다. 각 인터페이스에는 고유의 IP 주소 및 임대 시간이 있습니다. 여러 개의 네트워크 인터페이스가 DHCP용으로 구성된 경우 클라이언트가 이들을 구성하기 위해 별도의 요청을 실행합니다. 클라이언트는 각 인터페이스마다 별도의 네트워크 구성 매개변수를 유지 관리합니다. 매개변수가 별도로 저장되더라도 일부는 사실상 전역 매개변수입니다. 전역 매개변수는 특정 네트워크 인터페이스가 아닌 시스템에 전체적으로 적용됩니다.

전역 매개변수의 예로 호스트 이름, NIS 도메인 이름, 시간대 등이 있습니다. 전역 매개변수는 대개 각 인터페이스마다 다른 값을 가집니다. 그러나 각 시스템과 연관된 각 전역 매개변수에 대해 하나의 값만 사용할 수 있습니다. 전역 매개변수에 대한 질의 응답이 하나만 있도록 하려면 기본 네트워크 인터페이스의 매개변수만 사용됩니다. 기본 인터페이스로 지정하려는 인터페이스에 대해 `/etc/dhcp.` *interface* 파일에 `primary`라는 단어를 삽입할 수 있습니다. `primary` 키워드가 사용되지 않은 경우 사전순으로 첫번째 인터페이스가 기본 인터페이스로 간주됩니다.

DHCP 클라이언트는 논리적 인터페이스 및 물리적 인터페이스에 대한 임대를 동일하게 관리합니다. 단, 논리적 인터페이스에는 다음 제한 사항이 있습니다.

- DHCP 클라이언트가 논리적 인터페이스와 연관된 기본 경로를 관리하지 않습니다.

  Oracle Solaris 커널이 경로를 논리적 인터페이스가 아닌 물리적 인터페이스와 연관시킵니다. 물리적 인터페이스의 IP 주소가 설정된 경우 필요한 기본 경로가 경로 지정 테이블에 배치되어야 합니다. DHCP가 나중에 물리적 인터페이스와 연관된 논리적 인터페이스를 구성하는 경우 필요한 경로가 이미 제자리에 있어야 합니다. 논리적 인터페이스가 동일한 경로를 사용합니다.

  물리적 인터페이스에서 임대가 만료되면 DHCP 클라이언트가 인터페이스와 연관된 기본 경로를 제거합니다. 논리적 인터페이스에서 임대가 만료되면 DHCP 클라이언트가 논리적 인터페이스와 연관된 기본 경로를 제거하지 않습니다. 연관된 물리적 인터페이스 및 다른 가능한 논리적 인터페이스가 동일한 경로를 사용해야 할 수 있습니다.

  DHCP 제어 인터페이스와 연관된 기본 경로를 추가/제거해야 하는 경우 DHCP 클라이언트 이벤트 스크립트 방식을 사용할 수 있습니다. 418 페이지 “DHCP 클라이언트 이벤트 스크립트”를 참조하십시오.

DHCPv4 클라이언트 호스트 이름

기본적으로 DHCPv4 클라이언트는 DHCP 서버에서 호스트 이름을 제공할 것으로 기대하기 때문에 고유의 호스트 이름을 제공하지 않습니다. DHCPv4 서버는 기본적으로 DHCPv4 클라이언트에 호스트 이름을 제공하도록 구성됩니다. DHCPv4 클라이언트와 서버를 함께 사용할 때 이러한 기본값이 잘 작동합니다. 그러나 DHCPv4 클라이언트를 타사 DHCP 서버와 사용할 때 클라이언트가 서버에서 호스트 이름을 받지 못할 수 있습니다. DHCP 클라이언트가 DHCP를 통해 호스트 이름을 받지 않는 경우 클라이언트 시스템은 `/etc/nodename` 파일에서 호스트 이름으로 사용할 이름을 확인합니다. 파일이 비어 있으면 호스트 이름이 `unknown`으로 설정됩니다.

DHCP 서버가 DHCP `Hostname` 옵션에 이름을 제공하는 경우 `/etc/nodename` 파일에 다른 값이 있을지라도 클라이언트는 이 호스트 이름을 사용합니다. 클라이언트가 특정 호스트 이름을 사용하도록 하려면 클라이언트가 해당 이름을 요청하도록 설정할 수 있습니다. 다음 절차를 참조하십시오.

주 – 다음 절차는 모든 DHCP 서버와 작동하지 않습니다. 이 절차를 통해 클라이언트가 DHCP 서버에 특정 호스트 이름을 보내고 교대로 동일한 이름을 기대하도록 요구하게 됩니다.

그러나 DHCP 서버는 이 요청을 존중할 필요가 없으며 대부분 무시합니다. 간단히 다른 이름을 반환합니다.

▼ DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법

1. **클라이언트 시스템에서 수퍼 유저로 `/etc/default/dhcpagent` 파일을 편집합니다.**

2. **`/etc/default/dhcpagent` 파일에서 `REQUEST_HOSTNAME` 키워드를 찾아 다음과 같이 수정합니다.**

   ```
   REQUEST_HOSTNAME=yes
   ```

   주석 기호(#)가 `REQUEST_HOSTNAME` 앞에 있으면 #을 제거합니다. `REQUEST_HOSTNAME` 키워드가 존재하지 않으면 삽입합니다.

3. **클라이언트 시스템에서 `/etc/hostname.`*interface* 파일을 편집하여 다음 행을 추가합니다.**

   `inet` *hostname*

   *hostname*은 클라이언트가 사용할 이름입니다.

4 **클라이언트가 재부트 시 전체 DHCP 협상을 수행하도록 지정하려면 다음 명령을 입력합니다.**

```
# ifconfig interface dhcp release
# reboot
```

클라이언트에 캐시된 DHCP 데이터는 제거됩니다. 클라이언트는 새 호스트 이름을 포함하는 새 구성 정보를 요청하기 위해 프로토콜을 다시 시작합니다. DHCP 서버는 먼저 네트워크의 다른 시스템에서 호스트 이름을 사용하지 않는지 확인합니다. 그런 다음 서버는 호스트 이름을 클라이언트에 지정합니다. 구성에 따라 DHCP 서버는 클라이언트의 호스트 이름으로 이름 서비스를 업데이트할 수 있습니다.

나중에 호스트 이름을 변경하려면 단계 3 및 단계 4를 반복합니다.

DHCP 클라이언트 시스템 및 이름 서비스

Oracle Solaris 시스템은 이름 서비스로 DNS, NIS, NIS+ 및 로컬 파일 저장소(`/etc/inet/hosts`)를 지원합니다. 각 이름 서비스는 사용하기 전에 일부 구성이 필요합니다. 또한 사용될 이름 서비스를 나타내도록 이름 서비스 스위치 구성 파일(`nsswitch.conf(4)` 참조)을 적절히 설정해야 합니다.

DHCP 클라이언트 시스템이 이름 서비스를 사용하기 전에 시스템을 이름 서비스의 클라이언트로 구성해야 합니다. 기본적으로, 그리고 시스템 설치 중에 구성되지 않는 한 로컬 파일만 사용됩니다.

다음 표는 각 이름 서비스 및 DHCP에 관련된 문제를 요약한 것입니다. 이 표는 각 이름 서비스에 대한 클라이언트를 설정하는 데 도움이 되는 문서에 대한 상호 참조를 포함합니다.

표 16–1 DHCP 클라이언트 시스템에 대한 이름 서비스 클라이언트 설정 정보

이름 서비스	클라이언트 설정 정보

NIS	DHCP를 사용하여 Oracle Solaris 네트워크 설치 정보를 클라이언트 시스템으로 보내는 경우 `NISservs` 및 `NISdmain` 옵션을 포함하는 구성 매크로를 사용할 수 있습니다. 이러한 옵션은 NIS 서버의 IP 주소와 NIS 도메인 이름을 클라이언트로 전달합니다. 그러면 클라이언트가 자동으로 NIS 클라이언트가 됩니다.
	
	DHCP 클라이언트 시스템이 Oracle Solaris를 이미 실행 중인 경우 DHCP 서버가 NIS 정보를 클라이언트로 보낼 때 NIS 클라이언트가 해당 시스템에 자동으로 구성되지 않습니다.
	
	DHCP 서버가 NIS 정보를 DHCP 클라이언트 시스템으로 보내도록 구성된 경우 다음과 같이 클라이언트에서 `dhcpinfo` 명령을 사용하면 클라이언트에 제공된 값을 볼 수 있습니다.
	
	`# /usr/sbin/dhcpinfo NISdmain`
	
	`# /usr/sbin/dhcpinfo NISservs`
	
	주 – DHCPv6의 경우 다음과 같이 `-v6` 및 다른 프로토콜 키워드를 명령에 포함합니다.
	
	`# /usr/sbin/dhcpinfo -v6 NISDomain`
	
	`# /usr/sbin/dhcpinfo -v6 NISServers`
	
	시스템을 NIS 클라이언트로 설정할 때 NIS 도메인 이름 및 NIS 서버에 대한 반환된 값을 사용합니다.
	
	표준 방법으로 DHCP 클라이언트 시스템에 대해 NIS 클라이언트를 설정합니다. **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**의 5 장, “Setting Up and Configuring NIS Service”을 참조하십시오.
	
	참고 – `dhcpinfo` 및 `ypinit`를 사용하는 스크립트를 작성하여 DHCP 클라이언트 시스템에서 NIS 클라이언트 구성을 자동화할 수 있습니다.

NIS+	DHCP 클라이언트 시스템에 대한 NIS+ 클라이언트가 기존 방식으로 설정된 경우 DHCP 서버는 때에 따라 클라이언트에 다른 주소를 줄 수 있습니다. NIS+ 보안은 구성의 일부로 IP 주소를 포함하기 때문에 이렇게 되면 보안 문제가 발생할 수 있습니다. 클라이언트가 매번 동일한 주소를 받도록 하려면 DHCP 클라이언트 시스템에 대한 NIS+ 클라이언트를 표준이 아닌 방법(415 페이지 “DHCP 클라이언트를 NIS+ 클라이언트로 설정” 참조)으로 설정해야 합니다. DHCP 클라이언트 시스템에 수동으로 IP 주소가 지정된 경우 클라이언트의 주소는 항상 동일합니다. NIS+ 클라이언트를 표준 방법으로 설정할 수 있습니다(**System Administration Guide: Naming and Directory Services (NIS+)**의 “Setting Up NIS+ Client Machines” 참조).

표 16–1 DHCP 클라이언트 시스템에 대한 이름 서비스 클라이언트 설정 정보 *(계속)*

이름 서비스	클라이언트 설정 정보
/etc/inet/hosts	이름 서비스로 /etc/inet/hosts를 사용할 DHCP 클라이언트 시스템에 대해 /etc/inet/hosts 파일을 설정해야 합니다. DHCP 도구에 의해 DHCP 클라이언트 시스템의 호스트 이름이 고유의 /etc/inet/hosts 파일에 추가됩니다. 그러나 네트워크의 다른 시스템의 /etc/inet/hosts 파일에 호스트 이름을 수동으로 추가해야 합니다. DHCP 서버 시스템이 이름 분석에 /etc/inet/hosts를 사용하는 경우 시스템에서 클라이언트의 호스트 이름을 수동으로 추가해야 합니다.
DNS	DHCP 클라이언트 시스템이 DHCP를 통해 DNS 도메인 이름을 수신하는 경우 클라이언트 시스템의 /etc/resolv.conf 파일은 자동으로 구성됩니다. /etc/nsswitch.conf 파일도 자동으로 업데이트되어 검색 순서에서 다른 모든 이름 서비스 뒤의 hosts 행에 dns가 추가됩니다. DNS에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)**를 참조하십시오.

DHCP 클라이언트를 NIS+ 클라이언트로 설정

DHCP 클라이언트인 Oracle Solaris 시스템에서 NIS+ 이름 서비스를 사용할 수 있습니다. 하지만 DHCP 서버가 때에 따라 다른 주소를 제공한다면 NIS+의 보안 향상 기능 중 하나인 DES(Data Encryption Standard) 자격 증명을 만드는 기능이 일부 사용되지 못할 수 있습니다. 보안을 위해 항상 같은 주소를 제공하도록 DHCP 서버를 구성하십시오. DHCP를 사용하지 **않는** NIS+ 클라이언트를 설정하는 경우 클라이언트에 대한 고유한 DES 자격 증명을 NIS+ 서버에 추가합니다. nisclient 스크립트 또는 nisaddcred 명령을 사용하는 등 몇 가지 방법으로 자격 증명을 만들 수 있습니다.

NIS+ 자격 증명을 생성하려면 자격 증명을 만들고 저장할 수 있도록 클라이언트가 정적 호스트 이름을 가지고 있습니다. NIS+ 및 DHCP를 사용하려면 동일한 자격 증명을 만들어 DHCP 클라이언트의 모든 호스트 이름에 대해 사용되도록 해야 합니다. 이렇게 하면 DHCP 클라이언트는 어떤 IP 주소 및 연관된 호스트 이름을 받는지에 관계없이 동일한 DES 자격 증명을 사용할 수 있습니다.

다음 절차에서는 모든 DHCP 호스트 이름에 대해 동일한 자격 증명을 만드는 방법을 보여줍니다. 이 절차는 DHCP 클라이언트가 사용하는 호스트 이름을 아는 경우에만 유효합니다. 예를 들어, DHCP 서버가 호스트 이름을 생성할 때 클라이언트가 받을 수 있는 호스트 이름을 알고 있습니다.

▼ DHCP 클라이언트를 NIS+ 클라이언트로 설정하는 방법

NIS+ 클라이언트가 될 DHCP 클라이언트 시스템은 NIS+ 도메인의 다른 NIS+ 클라이언트 시스템에 속한 자격 증명을 사용해야 합니다. 이 절차에서는 시스템에 대한 자격 증명만 생성합니다. 이 자격 증명은 시스템에 로그인한 수퍼 유저에게만 적용됩니다. DHCP 클라이언트 시스템에 로그인하는 다른 사용자는 NIS+ 서버에 고유한 자체 자격 증명을 가지고 있어야 합니다. 이러한 자격 증명은 **System Administration Guide: Naming and Directory Services (NIS+)** 의 절차에 따라 만들어집니다.

1 **NIS+ 서버에서 다음 명령을 입력하여 클라이언트에 대한 자격 증명을 만듭니다.**

```
# nisgrep nisplus-client-name cred.org_dir > /tmp/file
```

이 명령은 NIS+ 클라이언트에 대한 `cred.org_dir` 테이블 항목을 임시 파일에 씁니다.

2 **`cat` 명령을 사용하여 임시 파일의 내용을 봅니다.**

또는 텍스트 편집기를 사용합니다.

3 **DHCP 클라이언트에 사용할 자격 증명을 복사합니다.**

콜론으로 구분된 숫자와 문자의 긴 문자열인 공개 키와 개인 키를 복사해야 합니다. 자격 증명은 다음 단계에서 실행되는 명령으로 붙여 넣게 됩니다.

4 **다음 명령을 입력하여 DHCP 클라이언트에 대한 자격 증명을 추가합니다.**

```
# nistbladm -a cname=" dhcp-client-name@nisplus-domain" auth_type=DES \
auth_name="unix.dhcp-client-name@nisplus-domain" \
public_data=copied-public-key \
private_data=copied-private-key
```

*copied-public-key*의 경우 임시 파일에서 복사한 공개 키 정보를 붙여 넣습니다.
*copied-private-key*의 경우 임시 파일에서 복사한 개인 키 정보를 붙여 넣습니다.

5 **DHCP 클라이언트 시스템에서 다음 명령을 입력하여 NIS+ 클라이언트 시스템의 파일을 DHCP 클라이언트 시스템으로 원격 복사합니다.**

```
# rcp nisplus-client-name:/var/nis/NIS_COLD_START /var/nis
# rcp nisplus-client-name:/etc/.rootkey /etc
# rcp nisplus-client-name:/etc/defaultdomain /etc
```

"권한이 거부되었습니다."라는 메시지가 표시되는 경우 시스템이 원격 복사를 허용하도록 설정되어 있지 않을 수 있습니다. 이 경우 일반 사용자로 파일을 중간 위치에 복사할 수 있습니다. 수퍼 유저로 중간 위치에서 DHCP 클라이언트 시스템의 적절한 위치로 파일을 복사합니다.

6 **DHCP 클라이언트 시스템에서 다음 명령을 입력하여 NIS+에 대한 올바른 이름 서비스 스위치 파일을 복사합니다.**

```
# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
```

7 **DHCP 클라이언트 시스템을 재부트합니다.**

이제 DHCP 클라이언트 시스템은 NIS+ 서비스를 사용할 수 있어야 합니다.

예 16–1 DHCP 클라이언트 시스템을 NIS+ 클라이언트로 설정

다음 예에서는 NIS+ 도메인 `dev.example.net`의 NIS+ 클라이언트인 `nisei` 시스템이 있다고 가정합니다. 또한 DHCP 클라이언트 시스템 `dhow`가 있으며 `dhow`를 NIS+ 클라이언트로 만들려고 한다고 가정합니다.

```
     (First log in as superuser on the NIS+ server)
# nisgrep nisei cred.org_dir > /tmp/nisei-cred
# cat /tmp/nisei-cred
nisei.dev.example.net.:DES:unix.nisei@dev.example.net:46199279911a84045b8e0
c76822179138173a20edbd8eab4:90f2e2bb6ffe7e3547346dda624ec4c7f0fe1d5f37e21cff63830
c05bc1c724b
# nistbladm -a cname="dhow@dev.example.net." \
auth_type=DES auth_name="unix.dhow@dev.example.net" \
public_data=46199279911a84045b8e0c76822179138173a20edbd8eab4 \
private_data=90f2e2bb6ffe7e3547346dda624ec4c7f0fe1d5f37e21cff63830\
c05bc1c724b
# rlogin dhow
     (Log in as superuser on dhow)
# rcp nisei:/var/nis/NIS_COLD_START /var/nis
# rcp nisei:/etc/.rootkey /etc
# rcp nisei:/etc/defaultdomain /etc
# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
# reboot
```

이제 DHCP 클라이언트 시스템 dhow는 NIS+ 서비스를 사용할 수 있어야 합니다.

예 16–2 스크립트를 사용하여 자격 증명 추가

많은 수의 DHCP 클라이언트 시스템을 NIS+로 설정해야 하는 경우 스크립트를 작성할 수 있습니다. 스크립트를 사용하면 cred.org_dir NIS+ 테이블에 항목을 빠르게 추가할 수 있습니다. 다음은 스크립트의 예를 보여줍니다.

```
#! /usr/bin/ksh
#
# Copyright (c) by Sun Microsystems, Inc. All rights reserved.
#
# Sample script for cloning a credential. Hosts file is already populated
# with entries of the form dhcp-[0-9][0-9][0-9]. The entry we're cloning
# is dhcp-001.
#
#
PUBLIC_DATA=6e72878d8dc095a8b5aea951733d6ea91b4ec59e136bd3b3
PRIVATE_DATA=3a86729b685e2b2320cd7e26d4f1519ee070a60620a93e48a8682c5031058df4
HOST="dhcp-"
DOMAIN="mydomain.example.com"

for
i in 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019
do
     print - ${HOST}${i}
     #nistbladm -r [cname="${HOST}${i}.${DOMAIN}."]cred.org_dir
     nistbladm -a cname="${HOST}${i}.${DOMAIN}." \
         auth_type=DES auth_name="unix.${HOST}${i}@${DOMAIN}" \
         public_data=${PUBLIC_DATA} private_data=${PRIVATE_DTA} cred.org_Dir
done

exit 0
```

DHCP 클라이언트 이벤트 스크립트

DHCP 클라이언트를 설정하여 클라이언트 시스템에 적절한 동작을 수행할 수 있는 실행 파일 프로그램 또는 스크립트를 실행할 수 있습니다. 프로그램 또는 스크립트는 **이벤트 스크립트**라고 하며, 특정 DHCP 임대 이벤트가 발생한 후 자동으로 실행됩니다. 이벤트 스크립트를 사용하여 특정 임대 이벤트에 대한 응답으로 다른 명령, 프로그램 또는 스크립트를 실행할 수 있습니다. 이 기능을 사용하려면 고유의 이벤트 스크립트를 제공해야 합니다.

다음 이벤트 키워드가 `dhcpagent`에서 DHCP 임대 이벤트를 구별하는 데 사용됩니다.

이벤트 키워드	설명
`BOUND` 및 `BOUND6`	DHCP용으로 인터페이스가 구성됩니다. 클라이언트가 DHCP 서버에서 확인 메시지(DHCPv4 `ACK` 또는 DHCPv6 `Reply`)를 수신하여 IP 주소에 대한 임대 요청을 부여합니다. 인터페이스를 성공적으로 구성한 후에 즉시 이벤트 스크립트가 호출됩니다.
`EXTEND` 및 `EXTEND6`	클라이언트가 임대를 성공적으로 연장합니다. 클라이언트가 DHCP 서버에서 갱신 요청에 대한 확인 메시지를 수신한 후에 즉시 이벤트 스크립트가 호출됩니다.
`EXPIRE` 및 `EXPIRE6`	임대 시간이 다 되었을 때 임대가 만료됩니다. DHCPv4의 경우, 임대된 주소가 인터페이스에서 제거되고 인터페이스가 작동 중지로 표시되기 전에 즉시 이벤트 스크립트가 호출됩니다. DHCPv6의 경우, 마지막 남은 임대된 주소가 인터페이스에서 제거되기 전에 바로 이벤트 스크립트가 호출됩니다.
`DROP` 및 `DROP6`	클라이언트가 임대를 취소하여 DHCP 컨트롤에서 인터페이스를 제거합니다. 인터페이스를 DHCP 제어에서 제거한 후에 즉시 이벤트 스크립트가 호출됩니다.
`RELEASE` 및 `RELEASE6`	클라이언트가 IP 주소를 양도합니다. 클라이언트가 인터페이스에서 주소를 해제하고 DHCPv4 `RELEASE` 또는 DHCPv6 `Release` 패킷을 DHCP 서버로 보내기 전에 즉시 이벤트 스크립트가 호출됩니다.
`INFORM` 및 `INFORM6`	인터페이스가 DHCPv4 `INFORM` 또는 DHCPv6 `Information-Request` 메시지를 통해 DHCP 서버에서 신규 또는 업데이트된 구성 정보를 획득합니다. 이러한 이벤트는 DHCP 클라이언트가 서버에서 구성 매개변수만 얻고 IP 주소 임대를 얻지 않을 때 발생합니다.
`LOSS6`	임대 만료 중 하나 이상의 유효한 임대가 계속 남아 있으면 만료된 주소가 제거되기 전에 바로 이벤트 스크립트가 호출됩니다. 이러한 제거 예정 항목은 `IFF_DEPRECATED` 플래그로 표시됩니다.

이러한 이벤트를 사용하여 dhcpagent는 다음 명령을 호출합니다.

```
/etc/dhcp/eventhook interface event
```

여기서 *interface*는 DHCP를 사용 중인 인터페이스이고 *event*는 이전에 설명된 이벤트 키워드 중 하나입니다. 예를 들어, ce0 인터페이스가 DHCP용으로 처음 구성될 때 다음과 같이 dhcpagent가 이벤트 스크립트를 호출합니다.

```
/etc/dhcp/eventhook net0 BOUND
```

이벤트 스크립트 기능을 사용하려면 다음을 수행해야 합니다.

- 실행 파일 이름을 /etc/dhcp/eventhook로 지정합니다.
- root가 될 파일의 소유자를 설정합니다.
- 사용 권한을 755(rwxr-xr-x)로 설정합니다.
- 스크립트 또는 프로그램을 작성하여 문서화된 이벤트의 응답으로 동작 순서를 수행합니다. Sun이 새 이벤트를 추가할 수 있으므로 인식할 수 없거나 조치가 필요하지 않은 이벤트를 프로그램이 자동으로 무시해야 합니다. 예를 들어, 프로그램 또는 스크립트는 이벤트가 RELEASE일 때 로그 파일에 작성하고 다른 모든 이벤트를 무시할 수 있습니다.
- 스크립트 또는 프로그램을 비대화식으로 만듭니다. 이벤트 스크립트를 호출하기 전에 stdin, stdout, stderr이 /dev/null에 연결됩니다. 출력 또는 오류를 보려면 파일로 재지정해야 합니다.

이벤트 스크립트가 dhcpagent로부터 프로그램 환경을 상속받고 root 권한으로 실행합니다. 스크립트가 dhcpinfo 유틸리티를 사용하여 필요한 경우 인터페이스에 대한 추가 정보를 얻을 수 있습니다. 자세한 내용은 dhcpinfo(1) 매뉴얼 페이지를 참조하십시오.

dhcpagent 데몬이 이벤트 스크립트가 모든 이벤트에서 종료되기를 기다립니다. 이벤트 스크립트가 55초 후에 종료되지 않으면 dhcpagent가 SIGTERM 신호를 스크립트 프로세스로 보냅니다. 추가 3초 후에도 여전히 프로세스가 종료되지 않으면 데몬이 프로세스를 종료하기 위해 SIGKILL 신호를 보냅니다.

dhcpagent(1M) 매뉴얼 페이지에 이벤트 스크립트의 예가 포함됩니다.

예 16–3은 DHCP 이벤트 스크립트를 사용하여 /etc/resolv.conf 파일의 내용을 최신 상태로 유지하는 방법을 보여줍니다. BOUND 및 EXTEND 이벤트가 발생하면 스크립트는 도메인 서버 및 이름 서버의 이름을 바꿉니다. EXPIRE, DROP 및 RELEASE 이벤트가 발생하면 스크립트는 파일에서 도메인 서버 및 이름 서버의 이름을 제거합니다.

주 – 스크립트 예에서는 DHCP가 도메인 서버 및 이름 서버의 이름에 대한 권한 있는 소스라고 가정합니다. 또한 이 스크립트에서는 DHCP의 제어를 받는 모든 인터페이스가 일관성있는 현재 정보를 반환한다고 가정합니다. 이러한 가정은 사용자 시스템의 조건을 반영하지 않을 수 있습니다.

예 16–3 /etc/resolv.conf 파일을 업데이트하는 이벤트 스크립트

```
#!/bin/ksh -p

PATH=/bin:/sbin export PATH
umask 0222

# Refresh the domain and name servers on /etc/resolv.conf

insert ()
{
    dnsservers=‘dhcpinfo -i $1 DNSserv‘
    if [ -n "$dnsservers" ]; then
        # remove the old domain and name servers
        if [ -f /etc/resolv.conf ]; then
            rm -f /tmp/resolv.conf.$$
            sed -e ’/^domain/d’ -e ’/^nameserver/d’ \
                /etc/resolv.conf > /tmp/resolv.conf.$$
        fi

        # add the new domain
        dnsdomain=‘dhcpinfo -i $1 DNSdmain‘
        if [ -n "$dnsdomain" ]; then
            echo "domain $dnsdomain" >> /tmp/resolv.conf.$$
        fi

        # add new name servers
        for name in $dnsservers; do
            echo nameserver $name >> /tmp/resolv.conf.$$
        done
        mv -f /tmp/resolv.conf.$$ /etc/resolv.conf
    fi
}

# Remove the domain and name servers from /etc/resolv.conf

remove ()
{
    if [ -f /etc/resolv.conf ]; then
        rm -f /tmp/resolv.conf.$$
        sed -e ’/^domain/d’ -e ’/^nameserver/d’ \
            /etc/resolv.conf > /tmp/resolv.conf.$$
        mv -f /tmp/resolv.conf.$$ /etc/resolv.conf
    fi
}

case $2 in
BOUND | EXTEND)
    insert $1
    exit 0
```

예 16–3 /etc/resolv.conf 파일을 업데이트하는 이벤트 스크립트 *(계속)*

```
    ;;
EXPIRE | DROP | RELEASE)
    remove
    exit 0
    ;;
*)
    exit 0
    ;;
esac
```

◆ ◆ ◆ 17 장

17 DHCP 문제 해결(참조)

이 장에서는 DHCP 서버나 클라이언트를 구성할 때 발생할 수 있는 문제를 해결하는 데 도움이 되는 정보를 제공합니다. 또한 구성을 완료한 후 DHCP 사용 시 겪을 수 있는 문제를 알려줍니다.

이 장은 다음 정보를 포함합니다.

DHCP 서버 구성에 대한 내용은 14 장, "DHCP 서비스 구성(작업)"을 참조하십시오. DHCP 클라이언트 구성에 대한 내용은 407 페이지 "DHCP 클라이언트 사용 및 사용 안함"을 참조하십시오.

DHCP 서버 문제 해결

서버를 구성할 때 발생할 수 있는 문제는 다음 범주로 분류됩니다.

NIS+ 문제 및 DHCP 데이터 저장소

NIS+를 DHCP 데이터 저장소로 사용하는 경우 발생할 수 있는 문제는 다음과 같이 분류될 수 있습니다.

NIS+를 DHCP 데이터 저장소로 선택할 수 없음

NIS+를 데이터 저장소로 사용하려고 하는 경우 DHCP 관리자가 데이터 저장소에 대한 선택 항목으로 NIS+를 제공하지 않을 수 있습니다. `dhcpconfig` 명령을 사용하는 경우 NIS+가 설치되어 실행되고 있지 않다는 메시지가 표시될 수 있습니다. 이러한 증상은 모두 NIS+가 네트워크에서 사용 중일지라도 이 서버에 대해 NIS+가 구성되지 않았음을 의미합니다. NIS+를 데이터 저장소로 선택하려면 서버 시스템을 NIS+ 클라이언트로 구성해야 합니다.

DHCP 서버 시스템을 NIS+ 클라이언트로 설정하려면 다음 조건이 성립해야 합니다.

- 도메인이 이미 구성되어 있음
- NIS+ 도메인의 마스터 서버가 실행되고 있음
- 마스터 서버의 테이블이 채워져 있음
- 호스트 테이블에 새 클라이언트 시스템인 DHCP 서버 시스템에 대한 항목이 있음

NIS+ 클라이언트 구성에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (NIS+)**의 "Setting Up NIS+ Client Machines"을 참조하십시오.

NIS+가 DHCP 데이터 저장소로 적절히 구성되어 있지 않음

DHCP에서 NIS+를 성공적으로 사용한 후 NIS+에 변경된 사항이 있는 경우 오류가 발생할 수 있습니다. 변경 사항으로 인해 구성 문제가 발생할 수 있습니다. 다음 문제 및 해결 방법에 대한 설명을 통해 구성 문제의 원인을 파악하십시오.

문제: 루트 객체가 NIS+ 도메인에 존재하지 않습니다.

해결: 다음 명령을 입력합니다.

```
/usr/lib/nis/nisstat
```

이 명령은 도메인에 대한 통계를 표시합니다. 루트 객체가 존재하지 않으면 통계가 반환되지 않습니다.

System Administration Guide: Naming and Directory Services (NIS+)를 사용하여 NIS+ 도메인을 설정합니다.

문제: NIS+가 `passwd` 및 `publickey` 정보에 대해 사용되지 않습니다.

해결: 다음 명령을 입력하여 이름 서비스 스위치에 대한 구성 파일을 봅니다.

```
cat /etc/nsswitch.conf
```

"nisplus" 키워드에 대해 `passwd` 및 `publickey` 항목을 확인합니다. 이름 서비스 스위치 구성에 대한 자세한 내용은 **System Administration Guide: Naming and Directory Services (NIS+)**를 참조하십시오.

문제: 도메인 이름이 비어 있습니다.

해결: 다음 명령을 입력합니다.

```
domainname
```

명령에서 빈 문자열을 나열하는 경우 도메인에 대해 설정된 도메인 이름이 없는 것입니다. 로컬 파일을 데이터 저장소로 사용하거나 네트워크에 대해 NIS+ 도메인을 설정합니다. **System Administration Guide: Naming and Directory Services (NIS+)** 를 참조하십시오.

문제: NIS_COLD_START 파일이 존재하지 않습니다.

해결: 서버 시스템에서 다음 명령을 입력하여 파일이 존재하는지 여부를 파악합니다.

```
cat /var/nis/NIS_COLD_START
```

로컬 파일을 데이터 저장소로 사용하거나 NIS+ 클라이언트를 만듭니다. **System Administration Guide: Naming and Directory Services (NIS+)** 를 참조하십시오.

DHCP 데이터 저장소에 대한 NIS+ 액세스 문제

NIS+ 액세스 문제로 인해 잘못된 DES 자격 증명, 또는 NIS+ 객체 또는 테이블을 업데이트할 수 있는 권한 부족에 대한 오류 메시지가 발생할 수 있습니다. 다음 문제 및 해결 방법에 대한 설명을 통해 NIS+ 액세스 오류가 표시되는 원인을 파악하십시오.

문제: DHCP 서버 시스템에 NIS+ 도메인의 org_dir 객체에 대한 만들기 권한이 없습니다.

해결: 다음 명령을 입력합니다.

nisls -ld org_dir

액세스 권한은 r---rmcdrmcdr---의 형식으로 나열됩니다. 여기서 권한은 각각 nobody, owner, group 및 world에 적용됩니다. 객체의 소유자가 다음에 나열됩니다.

일반적으로 org_dir 디렉토리 객체는 owner와 group에 전체 권한을 제공합니다. 전체 권한은 읽기, 수정, 만들기 및 삭제로 구성됩니다. org_dir 디렉토리 객체는 world와 nobody 클래스에 읽기 권한만 제공합니다.

DHCP 서버 이름은 org_dir 객체의 소유자로 나열되거나 그룹의 주체로 나열되어야 합니다. 그룹에는 만들기 권한이 있어야 합니다. 다음 명령으로 그룹을 나열합니다.

nisls -ldg org_dir

필요한 경우 nischmod 명령을 사용하여 org_dir에 대한 권한을 변경합니다. 예를 들어, 그룹에 만들기 권한을 추가하려면 다음 명령을 입력합니다.

nischmod g+c org_dir

자세한 내용은 nischmod(1) 매뉴얼 페이지를 참조하십시오.

문제: DHCP 서버에 org_dir 객체 아래 테이블을 만들 수 있는 액세스 권한이 없습니다.

대개 이 문제는 서버 시스템의 주체 이름이 org_dir 객체의 소유 그룹 멤버가 아니거나 소유 그룹이 존재하지 않음을 의미합니다.

해결: 다음 명령을 입력하여 소유 그룹 이름을 찾습니다.

niscat -o org_dir
다음과 유사한 행을 찾습니다.

```
Group : "admin.example.com."
```

다음 명령을 사용하여 그룹의 주체 이름을 나열합니다.

nisgrpadm -l *groupname*
예를 들어, 다음 명령은 admin.example.com 그룹의 주체 이름을 나열합니다.

```
nisgrpadm -l admin.example.com
```

서버 시스템의 이름은 그룹의 명시적 멤버로 나열되거나 그룹의 암시적 멤버로 포함되어야 합니다. 필요한 경우 nisgrpadm 명령을 사용하여 서버 시스템의 이름을 그룹에 추가합니다.

예를 들어, 서버 이름 pacific을 그룹 admin.example.com에 추가하려면 다음 명령을 입력합니다.

```
nisgrpadm -a admin.example.com pacific.example.com
```

자세한 내용은 nisgrpadm(1) 매뉴얼 페이지를 참조하십시오.

문제: DHCP 서버가 NIS+ cred 테이블에 유효한 DES(Data Encryption Standard) 자격 증명을 가지고 있지 않습니다.

해결: 자격 증명 문제가 있으면 사용자가 NIS+ 이름 서비스에 DES 자격 증명을 가지고 있지 않다는 오류 메시지가 표시됩니다.

nisaddcred 명령을 사용하여 DHCP 서버 시스템에 대한 보안 자격 증명을 추가합니다.

다음 예는 example.com 도메인의 mercury 시스템에 대한 DES 자격 증명을 추가하는 방법을 보여줍니다.

```
nisaddcred -p unix.mercury@example.com \
-P mercury.example.com. DES example.com.
```

이 명령은 암호화된 보안 키를 생성하는 데 필요한 루트 암호를 입력하라는 메시지를 표시합니다.

자세한 내용은 nisaddcred(1M) 매뉴얼 페이지를 참조하십시오.

DHCP의 IP 주소 할당 오류

클라이언트가 IP 주소를 얻거나 확인하려고 시도할 때 syslog 또는 서버 디버깅 모드 출력에 기록된 문제를 볼 수 있습니다. 다음 공통 오류 메시지 목록은 가능한 원인과 해결 방법을 나타냅니다.

`There is no` *n.n.n.n* `dhcp-network table for DHCP client's network`
원인: 클라이언트가 특정 IP 주소를 요청 중이거나 현재 IP 주소에 대한 임대를 연장하려고 합니다. DHCP 서버가 해당 주소에 대한 DHCP 네트워크 테이블을 찾을 수 없습니다.

해결책: DHCP 네트워크 테이블이 실수로 삭제되었을 수 있습니다. DHCP 관리자 또는 `dhcpconfig` 명령으로 네트워크를 다시 추가하여 네트워크 테이블을 다시 만들 수 있습니다.

`ICMP ECHO reply to OFFER candidate: n.n.n.n, disabling`

원인: DHCP 클라이언트 제공 대상으로 고려된 IP 주소가 이미 사용 중입니다. 이 문제는 여러 개의 DHCP 서버가 주소를 소유하는 경우 발생할 수 있습니다. 또한 비DHCP 네트워크 클라이언트에 대해 주소가 수동으로 구성된 경우에도 발생할 수 있습니다.

해결책: 적절한 주소 소유권을 결정합니다. DHCP 서버 데이터베이스 또는 호스트의 네트워크 구성을 수정하십시오.

`ICMP ECHO reply to OFFER candidate: n.n.n.n. No corresponding dhcp network record.`

원인: DHCP 클라이언트 제공 대상으로 고려된 IP 주소의 레코드가 네트워크 테이블에 없습니다. 이 오류는 IP 주소를 선택한 후에 DHCP 네트워크 테이블에서 해당 주소 레코드가 삭제되었음을 나타냅니다. 이 오류는 중복 주소 검사가 완료되기 전 짧은 기간에만 발생할 수 있습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 DHCP 네트워크 테이블을 확인합니다. IP 주소가 누락된 경우 DHCP 관리자에서 Address(주소) 탭의 Edit(편집) 메뉴에서 Create(만들기)를 선택하여 주소를 만듭니다. `pntadm`을 사용하여 IP 주소를 만들 수도 있습니다.

`DHCP network record for n.n.n.nis unavailable, ignoring request.`

원인: 요청된 IP 주소에 대한 레코드가 DHCP 네트워크 테이블에 없으므로 서버가 요청을 삭제하는 중입니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 DHCP 네트워크 테이블을 확인합니다. IP 주소가 누락된 경우 DHCP 관리자에서 Address(주소) 탭의 Edit(편집) 메뉴에서 Create(만들기)를 선택하여 주소를 만듭니다. `pntadm`을 사용하여 주소를 만들 수도 있습니다.

n.n.n.n `currently marked as unusable.`

원인: 요청된 IP 주소가 네트워크 테이블에 사용할 수 없음으로 표시되었으므로 주소를 제공할 수 없습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 주소를 사용할 수 있도록 만들 수 있습니다.

n.n.n.n `was manually allocated. No dynamic address will be allocated.`

원인: 클라이언트 ID가 수동으로 할당된 주소에 지정되었고, 해당 주소가 사용할 수 없음으로 표시되어 있습니다. 서버가 이 클라이언트에 다른 주소를 할당할 수 없습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 주소를 사용할 수 있도록 만들거나, 다른 주소를 클라이언트에 수동으로 할당할 수 있습니다.

`Manual allocation (`*n.n.n.n, client ID*`) has` *n* `other records. Should have 0.`
원인: 지정된 클라이언트 ID를 가진 클라이언트가 여러 개의 IP 주소에 수동으로 지정되었습니다. 클라이언트가 하나의 주소에만 지정되어야 합니다. 서버가 네트워크 테이블에서 발견된 마지막 수동 지정 주소를 선택합니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 추가 수동 할당이 제거되도록 IP 주소를 수정하십시오.

`No more IP addresses on` *n.n.n.n*`network.`
원인: 지정된 네트워크에서 현재 DHCP에서 관리되는 IP 주소가 모두 할당되었습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 이 네트워크에 대한 새 IP 주소를 만듭니다.

`Client:` *clientid* `lease on` *n.n.n.n* `expired.`
원인: 임대를 협상할 수 없고 시간이 초과되었습니다.

해결책: 클라이언트가 새 임대를 얻으려면 프로토콜을 자동으로 다시 시작해야 합니다.

`Offer expired for client:` *n.n.n.n*
원인: 서버가 클라이언트에 IP 주소를 제공했지만, 클라이언트가 응답하는 데 너무 오랜 시간이 걸려서 제공이 만료되었습니다.

해결책: 클라이언트가 다른 Discover 메시지를 자동으로 발행해야 합니다. 이 메시지도 시간 초과되면 DHCP 서버에 대한 캐시 제공 시간 초과를 늘리십시오. DHCP 관리자의 Service(서비스) 메뉴에서 Modify(수정)를 선택합니다.

`Client:` *clientid* `REQUEST is missing requested IP option.`
원인: 클라이언트의 요청이 제공된 IP 주소를 지정하지 않았으므로 DHCP 서버가 요청을 무시했습니다. 이 문제는 업데이트된 DHCP 프로토콜인 RFC 2131과 호환되지 않는 타사 DHCP 클라이언트를 사용하는 경우 발생할 수 있습니다.

해결책: 클라이언트 소프트웨어를 업데이트하십시오.

`Client:` *clientid* `is trying to renew` *n.n.n.n*`, an IP address it has not leased.`
원인: DHCP 네트워크 테이블에 있는 이 클라이언트의 IP 주소가 클라이언트의 갱신 요청에서 지정한 IP 주소와 일치하지 않습니다. DHCP 서버가 임대를 갱신하지 않습니다. 이 문제는 클라이언트가 IP 주소를 아직 사용 중인 동안 클라이언트의 레코드를 삭제할 경우 발생할 수 있습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 네트워크 테이블을 조사하고, 필요한 경우 클라이언트의 레코드를 수정하십시오. 클라이언트 ID가 지정된 IP 주소에 바인드되어야 합니다. 클라이언트 ID가 바인드되지 않은 경우 클라이언트 ID를 추가하도록 주소 등록 정보를 편집하십시오.

`Client: `*clientid*` is trying to verify unrecorded address: `*n.n.n.n*`, ignored.`

원인: 지정된 클라이언트가 이 주소로 DHCP 네트워크 테이블에 등록되지 않았으므로 이 DHCP 서버에서 요청을 무시합니다.

네트워크의 다른 DHCP 서버가 이 클라이언트에 주소를 지정했을 수 있습니다. 그러나 클라이언트가 IP 주소를 아직 사용 중인 동안 클라이언트의 레코드를 삭제했을 수도 있습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 이 서버의 네트워크 테이블을 조사하고 네트워크에 다른 DHCP 서버가 있는지 확인하십시오. 필요한 경우 내용을 수정하십시오.

아무것도 안하고 임대가 만료되도록 허용할 수도 있습니다. 클라이언트가 새 주소 임대를 자동으로 요청합니다.

클라이언트가 새 임대를 즉시 얻도록 하려면 다음 명령을 입력하여 클라이언트에서 DHCP 프로토콜을 다시 시작하십시오.

```
ifconfig interface dhcp release
ifconfig interface dhcp start
```

DHCP 클라이언트 구성 문제 해결

DHCP 클라이언트에 발생할 수 있는 문제는 다음 범주로 분류됩니다.

- 429 페이지 “DHCP 서버와 통신 문제”
- 438 페이지 “부정확한 DHCP 구성 정보 관련 문제”

DHCP 서버와 통신 문제

이 절에서는 DHCP 클라이언트를 네트워크에 추가할 때 발생할 수 있는 문제를 설명합니다.

클라이언트 소프트웨어를 사용으로 설정하고 시스템을 재부트한 후에 클라이언트가 네트워크 구성을 얻기 위해 DHCP 서버에 연결하려고 시도합니다. 클라이언트가 서버 연결을 실패하면 다음과 같은 오류 메시지가 나타날 수 있습니다.

```
DHCP or BOOTP server not responding
```

문제를 해결하려면 클라이언트와 서버 모두에서 진단 정보를 수집해야 합니다. 정보를 수집하려면 다음 작업을 수행할 수 있습니다.

1. 430 페이지 “디버깅 모드로 DHCP 클라이언트를 실행하는 방법”
2. 430 페이지 “디버깅 모드로 DHCP 서버를 실행하는 방법”
3. 431 페이지 “snoop를 사용하여 DHCP 네트워크 트래픽을 모니터하는 방법”

이러한 작업은 별도로 또는 동시에 수행할 수 있습니다.

수집한 정보를 통해 문제 원인이 클라이언트나 서버인지 또는 중계 에이전트인지 확인할 수 있습니다. 그런 다음 해결 방법을 찾을 수 있습니다.

▼ 디버깅 모드로 DHCP 클라이언트를 실행하는 방법

DHCP 클라이언트가 아닌 경우 디버깅 모드로 클라이언트를 실행하는 방법은 클라이언트 설명서를 참조하십시오.

DHCP 클라이언트인 경우 다음 단계를 사용하십시오.

1 DHCP 클라이언트 시스템에 수퍼 유저로 로그인합니다.

2 DHCP 클라이언트 데몬을 종료합니다.

```
# pkill -x dhcpagent
```

3 디버깅 모드로 데몬을 다시 시작합니다.

```
# /sbin/dhcpagent -d1 -f &
```

`-d` 스위치는 상세 정보 표시 레벨 1로 DHCP 클라이언트를 디버깅 모드에 넣습니다. `-f` 스위치는 `syslog` 대신 콘솔로 출력이 보내지도록 합니다.

4 DHCP 협상을 시작하도록 인터페이스를 구성합니다.

```
# ifconfig interface dhcp start
```

*interface*를 `ge0`과 같은 클라이언트의 네트워크 인터페이스 이름으로 바꿉니다.

디버깅 모드로 실행할 때 클라이언트 데몬이 DHCP 요청을 수행하는 동안 화면에 메시지를 표시합니다. 클라이언트 디버깅 모드 출력에 대한 내용은 431 페이지 “디버깅 모드에서 DHCP 클라이언트의 출력”을 참조하십시오.

▼ 디버깅 모드로 DHCP 서버를 실행하는 방법

1 서버 시스템에 수퍼 유저로 로그인합니다.

2 DHCP 서버를 일시적으로 중지합니다.

```
# svcadm disable -t svc:/network/dhcp-server
```

DHCP 관리자 또는 `dhcpconfig`를 사용하여 서버를 중지할 수도 있습니다.

3. **디버깅 모드로 데몬을 다시 시작합니다.**

   ```
   # /usr/lib/inet/in.dhcpd -d -v
   ```

   대개 데몬을 실행할 때 사용하는 `in.dhcpd` 명령줄 옵션도 사용해야 합니다. 예를 들어, BOOTP 중계 에이전트로 데몬을 실행하는 경우 `in.dhcpd -d -v` 명령에 `-r` 옵션을 넣으십시오.

   디버깅 모드로 실행할 때 데몬이 DHCP 또는 BOOTP 요청을 처리하는 동안 화면에 메시지를 표시합니다. 서버 디버깅 모드 출력에 대한 내용은 432 페이지 "디버깅 모드에서 DHCP 서버의 출력"을 참조하십시오.

▼ snoop를 사용하여 DHCP 네트워크 트래픽을 모니터하는 방법

1. **DHCP 서버 시스템에 수퍼 유저로 로그인합니다.**

2. **snoop를 시작하여 서버의 네트워크 인터페이스에서 네트워크 트래픽 추적을 시작합니다.**

   ```
   # /usr/sbin/snoop -d interface -o snoop-output-filename udp port 67 or udp port 68
   ```

   예를 들어, 다음 명령을 입력할 수 있습니다.

   ```
   #  /usr/sbin/snoop -d hme0 -o /tmp/snoop.output udp port 67 or udp port 68
   ```

   필요한 정보를 얻은 후에 Ctrl-C를 눌러 `snoop`를 중지할 때까지 `snoop`가 인터페이스를 계속 모니터합니다.

3. **클라이언트 시스템을 부트하거나 클라이언트 시스템에서 `dhcpagent`를 다시 시작합니다.**

   430 페이지 "디버깅 모드로 DHCP 클라이언트를 실행하는 방법"에서 `dhcpagent`를 다시 시작하는 방법을 설명합니다.

4. **서버 시스템에서 `snoop`를 사용하여 네트워크 패킷의 내용이 담긴 출력 파일을 표시합니다.**

   ```
   # /usr/sbin/snoop -i snoop-output-filename -x0 -v
   ```

   예를 들어, 다음 명령을 입력할 수 있습니다.

   ```
   # /usr/sbin/snoop -i /tmp/snoop.output -x0 -v
   ```

참조 출력 해석에 대한 내용은 435 페이지 "DHCP snoop 출력"을 참조하십시오.

디버깅 모드에서 DHCP 클라이언트의 출력

다음 예는 디버깅 모드의 DHCP 클라이언트가 DHCP 요청을 보내고 DHCP 서버에서 구성 정보를 수신할 때 정상 출력을 보여줍니다.

예 17–1 디버깅 모드에서 DHCP 클라이언트의 정상 출력

```
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x27fc8 (DHCP filter)
/sbin/dhcpagent: debug: init_ifs: initted interface hme0
/sbin/dhcpagent: debug: insert_ifs: hme0: sdumax 1500, optmax 1260, hwtype 1, hwlen 6
/sbin/dhcpagent: debug: insert_ifs: inserted interface hme0
/sbin/dhcpagent: debug: register_acknak: registered acknak id 5
/sbin/dhcpagent: debug: unregister_acknak: unregistered acknak id 5
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x26018 (ARP reply filter)
/sbin/dhcpagent: info: setting IP netmask on hme0 to 255.255.192.0
/sbin/dhcpagent: info: setting IP address on hme0 to 10.23.3.233
/sbin/dhcpagent: info: setting broadcast address on hme0 to 10.23.63.255
/sbin/dhcpagent: info: added default router 10.23.0.1 on hme0
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x28054 (blackhole filter)
/sbin/dhcpagent: debug: configure_if: bound ifsp->if_sock_ip_fd
/sbin/dhcpagent: info: hme0 acquired lease, expires Tue Aug 10 16:18:33 2006
/sbin/dhcpagent: info: hme0 begins renewal at Tue Aug 10 15:49:44 2006
/sbin/dhcpagent: info: hme0 begins rebinding at Tue Aug 10 16:11:03 2006
```

클라이언트가 DHCP 서버에 연결할 수 없으면 다음 예제에 표시된 출력과 비슷한 디버깅 모드 출력이 나타날 수 있습니다.

예 17–2 디버깅 모드에서 DHCP 클라이언트의 문제를 나타내는 출력

```
/sbin/dhcpagent: debug: set_packet_filter: set filter 0x27fc8 (DHCP filter)
/sbin/dhcpagent: debug: init_ifs: initted interface hme0
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
/sbin/dhcpagent: debug: select_best: no valid OFFER/BOOTP reply
```

이 메시지가 나타나면 클라이언트 요청이 서버에 도달하지 않았거나 서버가 클라이언트에 응답을 보낼 수 없습니다. 431 페이지 “snoop를 사용하여 DHCP 네트워크 트래픽을 모니터하는 방법”에 설명된 대로 서버에서 snoop를 실행하여 클라이언트의 패킷이 서버에 도달했는지 확인합니다.

디버깅 모드에서 DHCP 서버의 출력

정상적인 서버 디버깅 모드 출력은 데몬을 시작할 때 서버 구성 정보에 이어서 각 네트워크 인터페이스에 대한 정보를 보여줍니다. 데몬을 시작한 후에 디버깅 모드 출력은 데몬이 처리하는 요청에 대한 정보를 보여줍니다. 예 17–3은 방금 시작한 DHCP 서버에 대한 디버깅 모드 출력을 보여줍니다. 응답하지 않는 다른 DHCP 서버에서 소유한 주소를 사용 중인 클라이언트에 대해 임대를 연장합니다.

예 17–3 디버깅 모드에서 DHCP 서버의 정상 출력

```
Daemon Version: 3.1
Maximum relay hops: 4
Transaction logging to console enabled.
Run mode is: DHCP Server Mode.
Datastore: nisplus
Path: org_dir.dhcp.test..:dhcp.test..:$
```

예 17–3 디버깅 모드에서 DHCP 서버의 정상 출력 *(계속)*

```
DHCP offer TTL: 10
Ethers compatibility enabled.
BOOTP compatibility enabled.
ICMP validation timeout: 1000 milliseconds, Attempts: 2.
Monitor (0005/hme0) started...
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.21.255.255
Netmask: 255.255.0.0
Address: 10.21.0.2
Monitor (0006/nf0) started...
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352      Type: DLPI
Broadcast: 10.22.255.255
Netmask: 255.255.0.0
Address: 10.22.0.1
Monitor (0007/qfe0) started...
Thread Id: 0007 - Monitoring Interface: qfe0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.23.63.255
Netmask: 255.255.192.0
Address: 10.23.0.1
Read 33 entries from DHCP macro database on Tue Aug 10 15:10:27 2006
Datagram received on network device: qfe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qfe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qfe0
Client: 0800201DBA3A is requesting verification of address owned by 10.21.0.4
Datagram received on network device: qfe0
Client: 0800201DBA3A maps to IP: 10.23.3.233
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
DHCP EXTEND 0934312543 0934316143 10.23.3.233 10.21.0.2
          0800201DBA3A SUNW.Ultra-5_10 0800201DBA3A
```

예 17–4는 BOOTP 중계 에이전트로 시작한 DHCP 데몬에서 디버깅 모드 출력을 보여줍니다. 에이전트가 클라이언트에서 DHCP 서버로 요청을 중계하고, 서버의 응답을 클라이언트로 중계합니다.

예 17–4 디버깅 모드에서 BOOTP 중계의 정상 출력

```
Relay destination: 10.21.0.4 (blue-servr2)     network: 10.21.0.0
Daemon Version: 3.1
Maximum relay hops: 4
Transaction logging to console enabled.
Run mode is: Relay Agent Mode.
Monitor (0005/hme0) started...
Thread Id: 0005 - Monitoring Interface: hme0 *****
MTU: 1500      Type: DLPI
Broadcast: 10.21.255.255
Netmask: 255.255.0.0
Address: 10.21.0.2
Monitor (0006/nf0) started...
```

예 17–4 디버깅 모드에서 BOOTP 중계의 정상 출력 *(계속)*

```
Thread Id: 0006 - Monitoring Interface: nf0 *****
MTU: 4352       Type: DLPI
Broadcast: 10.22.255.255
Netmask: 255.255.0.0
Address: 10.22.0.1
Monitor (0007/qfe0) started...
Thread Id: 0007 - Monitoring Interface: qfe0 *****
MTU: 1500       Type: DLPI
Broadcast: 10.23.63.255
Netmask: 255.255.192.0
Address: 10.23.0.1
Relaying request 0800201DBA3A to 10.21.0.4, server port.
BOOTP RELAY-SRVR 0934297685 0000000000 0.0.0.0 10.21.0.4 0800201DBA3A
N/A 0800201DBA3A
Packet received from relay agent: 10.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
BOOTP RELAY-CLNT 0934297688 0000000000 10.23.0.1 10.23.3.233 0800201DBA3A
N/A 0800201DBA3A
Relaying request 0800201DBA3A to 10.21.0.4, server port.
BOOTP RELAY-SRVR 0934297689 0000000000 0.0.0.0 10.21.0.4 0800201DBA3A
N/A 0800201DBA3A
Packet received from relay agent: 10.23.0.1
Relaying reply to client 0800201DBA3A
Unicasting datagram to 10.23.3.233 address.
Adding ARP entry: 10.23.3.233 == 0800201DBA3A
```

DHCP에 문제가 있는 경우 디버깅 모드 출력에 경고 또는 오류 메시지가 표시될 수 있습니다. 다음과 같은 DHCP 서버 오류 메시지 목록을 사용하여 해결 방법을 찾으십시오.

`ICMP ECHO reply to OFFER candidate:` *ip_address* `disabling`

원인: DHCP 서버가 클라이언트에 IP 주소를 제공하기 전에 해당 주소를 핑하여 사용 중이 아닌지 확인합니다. 클라이언트가 회신하면 주소가 사용 중입니다.

해결책: 구성된 주소가 아직 사용 중이 아닌지 확인합니다. `ping` 명령을 사용할 수 있습니다. 자세한 내용은 `ping`(1M) 매뉴얼 페이지를 참조하십시오.

`No more IP addresses on` *network-address* `network.`

원인: 클라이언트의 네트워크와 연관된 DHCP 네트워크 테이블에서 사용 가능한 IP 주소가 없습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 IP 주소를 더 만듭니다. DHCP 데몬이 여러 서브넷을 모니터하는 경우 클라이언트가 위치한 서브넷에 대해 추가 주소가 있는지 확인합니다. 자세한 내용은 354 페이지 “DHCP 서비스에 IP 주소 추가”를 참조하십시오.

`No more IP addresses for` *network-address* `network when you are running the DHCP daemon in BOOTP compatibility mode.`

원인: BOOTP가 임대 시간을 사용하지 않으므로 DHCP 서버가 설정된 BOOTP 플래그로 여유 주소를 찾아서 BOOTP 클라이언트에 할당합니다.

해결책: DHCP 관리자를 사용하여 BOOTP 주소를 할당합니다. 347 페이지 "DHCP 서비스로 BOOTP 클라이언트 지원(작업 맵)"을 참조하십시오.

`Request to access nonexistent per network database:` *database-name* `in datastore:` *datastore.*

원인: DHCP 서버 구성 중 서브넷의 DHCP 네트워크 테이블이 생성되지 않았습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 DHCP 네트워크 테이블과 새 IP 주소를 만듭니다. 340 페이지 "DHCP 네트워크 추가"를 참조하십시오.

`There is no` *table-name* `dhcp-network table for DHCP client's network.`

원인: DHCP 서버 구성 중 서브넷의 DHCP 네트워크 테이블이 생성되지 않았습니다.

해결책: DHCP 관리자 또는 `pntadm` 명령을 사용하여 DHCP 네트워크 테이블과 새 IP 주소를 만듭니다. 340 페이지 "DHCP 네트워크 추가"를 참조하십시오.

`Client using non_RFC1048 BOOTP cookie.`

원인: 네트워크의 장치가 지원되지 않는 BOOTP 구현에 액세스하려고 합니다.

해결책: 이 장치를 구성할 필요가 없으면 이 메시지를 무시하십시오. 장치를 지원하려면 자세한 내용은 347 페이지 "DHCP 서비스로 BOOTP 클라이언트 지원(작업 맵)"을 참조하십시오.

DHCP snoop 출력

`snoop` 출력에서 DHCP 클라이언트 시스템과 DHCP 서버 시스템 사이에 패킷이 교환된다고 나타나야 합니다. 각 시스템에 대한 IP 주소가 각 패킷에 나타납니다. 패킷의 경로에는 라우터나 중계 에이전트의 IP 주소도 포함됩니다. 시스템이 패킷을 교환하지 않으면 클라이언트 시스템과 서버 시스템이 전혀 연락하지 못할 수 있습니다. 그러면 문제가 더 낮은 레벨에 있습니다.

`snoop` 출력을 평가하려면 예상된 동작이 무엇인지 알아야 합니다. 예를 들어, 요청이 BOOTP 중계 에이전트를 통과해야 하는지 알아야 합니다. 또한 MAC 주소 및 관여한 시스템의 IP 주소를 알아야 이러한 값이 예상대로 나타나는지 확인할 수 있습니다. 여러 개의 네트워크 인터페이스가 있는 경우 네트워크 인터페이스의 주소도 알아야 합니다.

다음 예는 `blue-servr2`의 DHCP 서버에서 MAC 주소가 `8:0:20:8e:f3:7e`인 클라이언트로 보낸 DHCP 확인 메시지에 대한 정상적인 `snoop` 출력을 보여줍니다. 메시지에서 서버가 클라이언트에 IP 주소 `192.168.252.6` 및 호스트 이름 `white-6`를 지정합니다. 또한 메시지에는 클라이언트에 대한 수많은 표준 네트워크 옵션과 여러 공급업체별 옵션이 포함됩니다.

예 17–5 하나의 패킷에 대한 샘플 snoop 출력

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 26 arrived at 14:43:19.14
ETHER:  Packet size = 540 bytes
ETHER:  Destination = 8:0:20:8e:f3:7e, Sun
ETHER:  Source      = 8:0:20:1e:31:c1, Sun
ETHER:  Ethertype = 0800 (IP)
ETHER:
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:   Total length = 526 bytes
IP:   Identification = 64667
IP:   Flags = 0x4 IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 254 seconds/hops
IP:   Protocol = 17 (UDP)
IP:   Header checksum = 157a
IP:   Source address = 10.21.0.4, blue-servr2
IP:   Destination address = 192.168.252.6, white-6
IP:   No options
IP:   UDP:  ----- UDP Header -----
UDP:
UDP:  Source port = 67
UDP:  Destination port = 68 (BOOTPC)
UDP:  Length = 506
UDP:  Checksum = 5D4C
UDP:
DHCP: ----- Dynamic Host Configuration Protocol -----
DHCP:
DHCP: Hardware address type (htype) =  1 (Ethernet (10Mb))
DHCP: Hardware address length (hlen) = 6 octets
DHCP: Relay agent hops = 0
DHCP: Transaction ID = 0x2e210f17
DHCP: Time since boot = 0 seconds
DHCP: Flags = 0x0000
DHCP: Client address (ciaddr) = 0.0.0.0
DHCP: Your client address (yiaddr) = 192.168.252.6
DHCP: Next server address (siaddr) = 10.21.0.2
DHCP: Relay agent address (giaddr) = 0.0.0.0
DHCP: Client hardware address (chaddr) = 08:00:20:11:E0:1B
DHCP:
DHCP: ----- (Options) field options -----
DHCP:
DHCP: Message type = DHCPACK
DHCP: DHCP Server Identifier = 10.21.0.4
DHCP: Subnet Mask = 255.255.255.0
DHCP: Router at = 192.168.252.1
DHCP: Broadcast Address = 192.168.252.255
DHCP: NISPLUS Domainname = dhcp.test
```

예 17–5 하나의 패킷에 대한 샘플 snoop 출력 *(계속)*

```
DHCP: IP Address Lease Time = 3600 seconds
DHCP: UTC Time Offset = -14400 seconds
DHCP: RFC868 Time Servers at = 10.21.0.4
DHCP: DNS Domain Name = sem.example.com
DHCP: DNS Servers at = 10.21.0.1
DHCP: Client Hostname = white-6
DHCP: Vendor-specific Options (166 total octets):
DHCP:   (02) 04 octets  0x8194AE1B (unprintable)
DHCP:   (03) 08 octets  "pacific"
DHCP:   (10) 04 octets  0x8194AE1B (unprintable)
DHCP:   (11) 08 octets  "pacific"
DHCP:   (15) 05 octets  "xterm"
DHCP:   (04) 53 octets  "/export/s2/base.s2s/latest/Solaris_8/Tools/Boot"
DHCP:   (12) 32 octets  "/export/s2/base.s2s/latest"
DHCP:   (07) 27 octets  "/platform/sun4u/kernel/unix"
DHCP:   (08) 07 octets  "EST5EDT"
  0: 0800 208e f37e 0800 201e 31c1 0800 4500    .. .ó~.. .1...E.
 16: 020e fc9b 4000 fe11 157a ac15 0004 c0a8    ....@....z......
 32: fc06 0043 0044 01fa 5d4c 0201 0600 2e21    ...C.D..]L.....!
 48: 0f17 0000 0000 0000 0000 c0a8 fc06 ac15    ................
 64: 0002 0000 0000 0800 2011 e01b 0000 0000    ........ .......
 80: 0000 0000 0000 0000 0000 0000 0000 0000    ................
 96: 0000 0000 0000 0000 0000 0000 0000 0000    ................
112: 0000 0000 0000 0000 0000 0000 0000 0000    ................
128: 0000 0000 0000 0000 0000 0000 0000 0000    ................
144: 0000 0000 0000 0000 0000 0000 0000 0000    ................
160: 0000 0000 0000 0000 0000 0000 0000 0000    ................
176: 0000 0000 0000 0000 0000 0000 0000 0000    ................
192: 0000 0000 0000 0000 0000 0000 0000 0000    ................
208: 0000 0000 0000 0000 0000 0000 0000 0000    ................
224: 0000 0000 0000 0000 0000 0000 0000 0000    ................
240: 0000 0000 0000 0000 0000 0000 0000 0000    ................
256: 0000 0000 0000 0000 0000 0000 0000 0000    ................
272: 0000 0000 0000 6382 5363 3501 0536 04ac    ......c.Sc5..6..
288: 1500 0401 04ff ffff 0003 04c0 a8fc 011c    ................
304: 04c0 a8fc ff40 0964 6863 702e 7465 7374    .....@.dhcp.test
320: 3304 0000 0e10 0204 ffff c7c0 0404 ac15    3...............
336: 0004 0f10 736e 742e 6561 7374 2e73 756e    ....sem.example.
352: 2e63 6f6d 0604 ac15 0001 0c07 7768 6974    com.........whit
368: 652d 362b a602 0481 94ae 1b03 0861 746c    e-6+.........pac
384: 616e 7469 630a 0481 94ae 1b0b 0861 746c    ific.........pac
400: 616e 7469 630f 0578 7465 726d 0435 2f65    ific...xterm.5/e
416: 7870 6f72 742f 7332 382f 6261 7365 2e73    xport/sx2/bcvf.s
432: 3238 735f 776f 732f 6c61 7465 7374 2f53    2xs_btf/latest/S
448: 6f6c 6172 6973 5f38 2f54 6f6f 6c73 2f42    olaris_x/Tools/B
464: 6f6f 740c 202f 6578 706f 7274 2f73 3238    oot. /export/s2x
480: 2f62 6173 652e 7332 3873 5f77 6f73 2f6c    /bcvf.s2xs_btf/l
496: 6174 6573 7407 1b2f 706c 6174 666f 726d    atest../platform
512: 2f73 756e 346d 2f6b 6572 6e65 6c2f 756e    /sun4u/kernel/un
528: 6978 0807 4553 5435 4544 54ff              ix..EST5EDT.
```

부정확한 DHCP 구성 정보 관련 문제

DHCP 클라이언트가 네트워크 구성 정보에 부정확한 정보를 수신하는 경우 DHCP 서버 데이터를 확인합니다. 이 클라이언트에 대해 DHCP 서버가 처리하는 매크로의 옵션 값을 조사해야 합니다. 부정확한 정보의 예로 잘못된 NIS 도메인 이름 또는 라우터 IP 주소가 있습니다.

다음 일반 지침을 사용하여 부정확한 정보의 소스를 확인할 수 있습니다.

- 367 페이지 "DHCP 서버에 정의된 매크로를 보는 방법(DHCP 관리자)"에 설명된 대로 서버에 정의된 매크로를 확인합니다. 291 페이지 "매크로 처리 순서"에서 정보를 검토하고, 이 클라이언트에 대해 어떤 매크로가 자동으로 처리되는지 확인합니다.
- 네트워크 테이블에서 어떤 매크로가 클라이언트의 IP 주소에 구성 매크로로 지정되는지 확인합니다. 자세한 내용은 350 페이지 "DHCP 서비스에서 IP 주소 작업(작업 맵)"을 참조하십시오.
- 여러 개의 매크로에 발생하는 옵션이 있는지 주의합니다. 옵션에 원하는 값이 마지막 처리된 매크로에 설정되었는지 확인합니다.
- 적절한 매크로를 편집하여 올바른 값이 클라이언트에 전달되도록 합니다. 368 페이지 "DHCP 매크로 수정"을 참조하십시오.

DHCP 클라이언트가 제공한 호스트 이름 관련 문제

이 절에서는 DNS에 등록될 고유의 호스트 이름을 제공하는 DHCP 클라이언트에 발생할 수 있는 문제를 설명합니다.

DHCP 클라이언트가 호스트 이름을 요청하지 않음

DHCP 클라이언트가 아닌 경우 호스트 이름을 요청하도록 클라이언트를 구성하는 방법은 클라이언트 설명서를 참조하십시오. DHCP 클라이언트인 경우 412 페이지 "DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법"을 참조하십시오.

DHCP 클라이언트가 요청된 호스트 이름을 가져오지 않음

다음 목록은 클라이언트가 요청된 호스트 이름을 가져올 때 발생 가능한 문제를 포함하고 제안된 해결 방법을 설명합니다.

문제: 클라이언트가 DNS 업데이트를 발행하지 않는 DHCP 서버에서 제공을 수락했습니다.

해결: 두 DHCP 서버가 클라이언트에 사용 가능한 경우 양쪽 서버 모두 DNS 업데이트를 제공하도록 구성해야 합니다. DHCP 서버 및 DNS 서버 구성에 대한 내용은 332 페이지 "DHCP 서버에 의한 동적 DNS 업데이트를 사용으로 설정"을 참조하십시오.

DHCP 서버가 DNS 업데이트를 제공하도록 구성되었는지 여부를 확인하려면 다음과 같이 하십시오.

1. 클라이언트의 DHCP 서버의 IP 주소를 확인합니다. 클라이언트 시스템에서 네트워크 패킷을 캡처하기 위해 `snoop` 또는 다른 응용 프로그램을 사용합니다. 431 페이지 “snoop를 사용하여 DHCP 네트워크 트래픽을 모니터하는 방법”을 참조하여 서버가 아닌 클라이언트에서 절차를 수행합니다. `snoop` 출력에서 DHCP 서버 식별자를 찾아서 서버의 IP 주소를 가져옵니다.
2. DHCP 서버 시스템에 로그인하여 시스템이 DNS 업데이트를 제공하도록 구성되었는지 확인합니다. 다음 명령을 수퍼 유저로 입력합니다.

   ```
   dhcpconfig -P
   ```

   `UPDATE_TIMEOUT`이 서버 매개변수로 나열된 경우 DHCP 서버가 DNS 업데이트를 제공하도록 구성된 것입니다.
3. DNS 서버에서 `/etc/named.conf` 파일을 확인합니다. 적절한 도메인의 `zone` 섹션에서 `allow-update` 키워드를 찾습니다. DHCP 서버에 의한 DNS 업데이트를 허용하는 경우 DHCP 서버의 IP 주소가 `allow-update` 키워드에 나열됩니다.

문제: 클라이언트가 호스트 이름을 지정하는 `FQDN` 옵션을 사용 중입니다. `FQDN` 옵션은 DHCP 프로토콜에 공식 포함되지 않으므로 DHCP에서 현재 지원되지 않습니다.

해결: 서버에서 네트워크 패킷을 캡처하기 위해 `snoop` 또는 다른 응용 프로그램을 사용합니다. 431 페이지 “snoop를 사용하여 DHCP 네트워크 트래픽을 모니터하는 방법”을 참조하십시오. `snoop` 출력에서, 클라이언트의 패킷에서 `FQDN` 옵션을 찾습니다.

`Hostname` 옵션을 사용하여 호스트 이름을 지정하도록 클라이언트를 구성합니다. `Hostname`은 옵션 코드 12입니다. 지침은 클라이언트 설명서를 참조하십시오.

Oracle Solaris 클라이언트인 경우 412 페이지 “DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법”을 참조하십시오.

문제: 클라이언트에 주소를 제공하는 DHCP 서버가 클라이언트의 DNS 도메인을 모릅니다.

해결: DHCP 서버에서 유효한 값으로 `DNSdmain` 옵션을 찾습니다. 이 클라이언트에 대해 처리되는 매크로에서 `DNSdmain` 옵션을 올바른 DNS 도메인 이름으로 설정합니다. `DNSdmain`은 대개 네트워크 매크로에 포함됩니다. 매크로의 옵션 값 변경에 대한 내용은 368 페이지 “DHCP 매크로 수정”을 참조하십시오.

문제: 클라이언트에서 요청된 호스트 이름이 DHCP 서버에서 관리되지 않는 IP 주소에 해당합니다. DHCP 서버는 관리하지 않는 IP 주소에 대해 DNS 업데이트를 수행하지 않습니다.

해결: DHCP 서버에서 다음 메시지 중 하나가 있는지 `syslog`를 확인하십시오.

- `There is no` *n.n.n.n* `dhcp-network table for DHCP client's network.`
- `DHCP network record for` *n.n.n.n* `is unavailable, ignoring request.`

다른 이름을 요청하도록 클라이언트를 구성합니다. 412 페이지 “DHCPv4 클라이언트가 특정 호스트 이름을 요청하도록 설정하는 방법”을 참조하십시오. DHCP 서버에서 관리되는 주소에 매핑된 이름을 선택합니다. DHCP 관리자의 Addresses(주소) 탭에서 주소 매핑을 볼 수 있습니다. 다른 방법으로, IP 주소에 매핑되지 않은 주소를 선택합니다.

문제: 클라이언트에서 요청된 호스트 이름이 현재 사용할 수 없는 IP 주소에 해당합니다. 주소가 사용 중이거나, 다른 클라이언트에 임대되었거나, 다른 클라이언트에 제공된 상태일 수 있습니다.

해결: DHCP 서버에서 다음 메시지가 있는지 `syslog`를 확인하십시오. `ICMP ECHO reply to OFFER candidate: n.n.n.n.`

다른 IP 주소에 해당하는 이름을 선택하도록 클라이언트를 구성합니다. 다른 방법으로, 주소를 사용하는 클라이언트에서 주소를 재생 이용합니다.

문제: DNS 서버가 DHCP 서버에서 업데이트를 수락하도록 구성되지 않았습니다.

해결: DNS 서버에서 `/etc/named.conf` 파일을 조사합니다. DHCP 서버 도메인에 대한 적절한 `zone` 섹션에서 `allow-update` 키워드로 DHCP 서버의 IP 주소를 찾습니다. IP 주소가 존재하지 않으면 DNS 서버가 DHCP 서버에서 업데이트를 수락하도록 구성되지 않았습니다.

DNS 서버 구성에 대한 내용은 333 페이지 “DHCP 클라이언트에 대한 동적 DNS 업데이트를 사용으로 설정하는 방법”을 참조하십시오.

DHCP 서버에 여러 인터페이스가 있는 경우 DHCP 서버의 모든 주소에서 업데이트를 수락하도록 DNS 서버를 구성해야 할 수 있습니다. DNS 서버에 디버깅을 사용으로 설정하여 DNS 서버에 업데이트가 도달하는지 여부를 확인합니다. DNS 서버가 업데이트 요청을 수신한 경우 디버깅 모드 출력을 조사하여 업데이트가 발생하지 않은 이유를 확인합니다. DNS 디버깅 모드에 대한 내용은 `in.named`.1M 매뉴얼 페이지를 참조하십시오.

문제: DNS 업데이트가 주어진 시간 안에 완료되지 않았을 수 있습니다. 구성된 시간 제한까지 DNS 업데이트가 완료되지 않은 경우 DHCP 서버가 클라이언트에 호스트 이름을 반환하지 않습니다. 그러나 DNS 업데이트를 완료하려는 시도를 계속합니다.

해결: `nslookup` 명령을 사용하여 업데이트가 성공적으로 완료되었는지 여부를 확인합니다. `nslookup(1M)` 매뉴얼 페이지를 참조하십시오.

예를 들어, DNS 도메인이 `hills.example.org`이고 DNS 서버의 IP 주소가 `10.76.178.11`이라고 가정해 보십시오. 클라이언트가 등록할 호스트 이름은 `cathedral`입니다. 다음 명령을 사용하여 `cathedral`이 DNS 서버에 등록되었는지 확인할 수 있습니다.

```
nslookup cathedral.hills.example.org 10.76.178.11
```

업데이트가 성공적으로 완료되었지만 주어진 시간을 초과한 경우 시간 초과 값을 늘려야 합니다. 333 페이지 “DHCP 클라이언트에 대한 동적 DNS 업데이트를 사용으로 설정하는 방법”을 참조하십시오. 이 절차에서 시간 초과 전에 DNS 서버에서 응답을 기다리는 시간(초)을 늘려야 합니다.

◆ ◆ ◆ 18장

18 DHCP 명령 및 파일(참조)

이 장에서는 DHCP 명령과 DHCP 파일 사이의 관계를 설명합니다. 그러나 명령 사용 방법은 설명하지 않습니다.

이 장은 다음 정보를 포함합니다.

DHCP 명령

다음 표는 네트워크에서 DHCP를 관리하는 데 사용할 수 있는 명령을 나열합니다.

표 18–1 DHCP에 사용된 명령

명령	설명
/usr/lib/inet/dhcpd	ISC DHCP 전용: ISC DHCP 서버 데몬입니다. 자세한 내용은 dhcpd(8) 매뉴얼 페이지를 참조하십시오.
/usr/lib/inet/dhcrelay	ISC DHCP 전용: DHCP 서버가 없는 네트워크의 클라이언트에서 다른 네트워크의 서버로 DHCP 및 BOOTP 요청을 중계하기 위한 수단입니다. 자세한 내용은 dhcrelay(8) 매뉴얼 페이지를 참조하십시오.
/usr/lib/inet/in.dhcpd	DHCP 서버 데몬입니다. 시스템을 시작할 때 데몬이 시작됩니다. 서버 데몬을 직접 시작하면 안됩니다. DHCP 관리자, svcadm 명령 또는 dhcpconfig를 사용하여 데몬을 시작 및 중지합니다. 문제 해결을 위해 디버그 모드로 서버를 실행하는 경우에만 데몬을 직접 호출해야 합니다. 자세한 내용은 in.dhcpd(1M) 매뉴얼 페이지를 참조하십시오.
/usr/sadm/admin/bin/dhcpmgr	DHCP 서비스 구성 및 관리에 사용되는 GUI(그래픽 사용자 인터페이스) 도구인 DHCP 관리자입니다. DHCP 관리자는 권장되는 DHCP 관리 도구입니다. 자세한 내용은 dhcpmgr(1M) 매뉴얼 페이지를 참조하십시오.

표 18–1 DHCP에 사용된 명령 (계속)

명령	설명
/usr/sbin/dhcpagent	DHCP 프로토콜의 클라이언트측을 구현하는 DHCP 클라이언트 데몬입니다. 자세한 내용은 dhcpagent(1M) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/dhcpconfig	DHCP 서버 및 BOOTP 중계 에이전트를 구성/구성 해제하는 데 사용됩니다. 또한 다른 데이터 저장소 형식으로 변환하고 DHCP 구성 데이터를 가져오고 내보내는 데 사용됩니다. 자세한 내용은 dhcpconfig(1M) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/dhcpinfo	Oracle Solaris 클라이언트 시스템의 시스템 시작 스크립트가 DHCP 클라이언트 데몬 dhcpagent에서 호스트 이름 등의 정보를 얻는 데 사용됩니다. 스크립트 또는 명령줄에서 dhcpinfo를 사용하여 지정된 매개변수 값을 얻을 수도 있습니다. 자세한 내용은 dhcpinfo(1) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/dhtadm	dhcptab 테이블의 옵션 및 매크로를 변경하는 데 사용됩니다. 이 명령은 DHCP 정보 변경을 자동화하기 위해 만드는 스크립트에 가장 유용합니다. dhcptab 테이블에서 특정 옵션 값을 검색하는 가장 빠른 방법은 dhtadm을 -P 옵션과 함께 사용하고 grep 명령을 통해 출력 결과를 파이프로 연결하는 것입니다. 자세한 내용은 dhtadm(1M) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/ifconfig	시스템 부트 시 네트워크 인터페이스에 IP 주소를 지정하거나, 네트워크 인터페이스 매개변수를 구성하거나 이 두 작업 모두를 수행하는 데 사용됩니다. DHCP 클라이언트에서 ifconfig가 DHCP를 시작하여 네트워크 인터페이스를 구성하는 데 필요한 매개변수(IP 주소 포함)를 얻습니다. 자세한 내용은 ifconfig(1M) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/omshell	ISC DHCP 전용: OMAPI(Object Management API)를 사용하여 ISC DHCP 서버의 상태를 질의하고 변경하는 방법을 제공합니다. 자세한 내용은 omshell(1) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/pntadm	클라이언트 ID를 IP 주소에 매핑하는 DHCP 네트워크 테이블을 변경하고, 선택적으로 구성 정보를 IP 주소와 연관시키는 데 사용됩니다. 자세한 내용은 pntadm(1M) 매뉴얼 페이지를 참조하십시오.
/usr/sbin/snoop	네트워크에서 전달되는 패킷의 내용을 캡처하고 표시하는 데 사용됩니다. snoop는 DHCP 서비스 관련 문제를 해결하는 데 유용합니다. 자세한 내용은 snoop(1M) 매뉴얼 페이지를 참조하십시오.

스크립트에서 DHCP 명령 실행

dhcpconfig, dhtadm 및 pntadm 명령은 스크립트에서 사용하도록 최적화되어 있습니다. 특히 pntadm 명령은 DHCP 네트워크 테이블에 많은 수의 IP 주소 항목을 만드는 데 유용합니다. 다음 스크립트 예에서는 일괄 처리 모드로 pntadm을 사용하여 IP 주소를 만듭니다.

예 18–1 pntadm 명령이 포함된 addclient.ksh 스크립트

```
#! /usr/bin/ksh
#
# This script utilizes the pntadm batch facility to add client entries
# to a DHCP network table. It assumes that the user has the rights to
# run pntadm to add entries to DHCP network tables.

#
# Based on the switch setting, query the netmasks table for a netmask.
# Accepts one argument, a dotted IP address.
#
get_netmask()
{
    MTMP=‘getent netmasks ${1} | awk ’{ print $2 }’‘
    if [ ! -z "${MTMP}" ]
    then
        print - ${MTMP}
    fi
}

#
# Based on the network specification, determine whether or not network is
# subnetted or supernetted.
# Given a dotted IP network number, convert it to the default class
# network.(used to detect subnetting). Requires one argument, the
# network number. (e.g. 10.0.0.0) Echos the default network and default
# mask for success, null if error.
#
get_default_class()
{
    NN01=${1%%.*}
    tmp=${1#*.}
    NN02=${tmp%%.*}
    tmp=${tmp#*.}
    NN03=${tmp%%.*}
    tmp=${tmp#*.}
    NN04=${tmp%%.*}
    RETNET=""
    RETMASK=""

    typeset -i16 ONE=10#${1%%.*}
    typeset -i10 X=$((${ONE}&16#f0))
    if [ ${X} -eq 224 ]
    then
        # Multicast
        typeset -i10 TMP=$((${ONE}&16#f0))
        RETNET="${TMP}.0.0.0"
        RETMASK="240.0.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#80))
    if [ -z "${RETNET}" -a ${X} -eq 0 ]
    then
        # Class A
        RETNET="${NN01}.0.0.0"
        RETMASK="255.0.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#c0))
    if [ -z "${RETNET}" -a ${X} -eq 128 ]
```

예 18–1 pntadm 명령이 포함된 addclient.ksh 스크립트 (계속)

```
    then
        # Class B
        RETNET="${NN01}.${NN02}.0.0"
        RETMASK="255.255.0.0"
    fi
    typeset -i10 X=$((${ONE}&16#e0))
    if [ -z "${RETNET}" -a ${X} -eq 192 ]
    then
        # Class C
        RETNET="${NN01}.${NN02}.${NN03}.0"
        RETMASK="255.255.255.0"
    fi
    print - ${RETNET} ${RETMASK}
    unset NNO1 NNO2 NNO3 NNO4 RETNET RETMASK X ONE
}

#
# Given a dotted form of an IP address, convert it to its hex equivalent.
#
convert_dotted_to_hex()
{
    typeset -i10 one=${1%%.*}
    typeset -i16 one=${one}
    typeset -Z2 one=${one}
    tmp=${1#*.}

    typeset -i10 two=${tmp%%.*}
    typeset -i16 two=${two}
    typeset -Z2 two=${two}
    tmp=${tmp#*.}

    typeset -i10 three=${tmp%%.*}
    typeset -i16 three=${three}
    typeset -Z2 three=${three}
    tmp=${tmp#*.}

    typeset -i10 four=${tmp%%.*}
    typeset -i16 four=${four}
    typeset -Z2 four=${four}

     hex=‘print - ${one}${two}${three}${four} | sed -e ’s/#/0/g’‘
     print - 16#${hex}
     unset one two three four tmp
}

#
# Generate an IP address given the network address, mask, increment.
#
get_addr()
{
    typeset -i16 net=‘convert_dotted_to_hex ${1}‘
    typeset -i16 mask=‘convert_dotted_to_hex ${2}‘
    typeset -i16 incr=10#${3}

    # Maximum legal value - invert the mask, add to net.
    typeset -i16 mhosts=~${mask}
```

예 18–1 pntadm 명령이 포함된 addclient.ksh 스크립트 *(계속)*

```
    typeset -i16 maxnet=${net}+${mhosts}

    # Add the incr value.
    let net=${net}+${incr}

    if [ $((${net} < ${maxnet})) -eq 1 ]
    then
        typeset -i16 a=${net}\&16#ff000000
        typeset -i10 a="${a}>>24"

        typeset -i16 b=${net}\&16#ff0000
        typeset -i10 b="${b}>>16"

        typeset -i16 c=${net}\&16#ff00
        typeset -i10 c="${c}>>8"

        typeset -i10 d=${net}\&16#ff
        print - "${a}.${b}.${c}.${d}"
    fi
    unset net mask incr mhosts maxnet a b c d
}

# Given a network address and client address, return the index.
client_index()
{
    typeset -i NNO1=${1%%.*}
    tmp=${1#*.}
    typeset -i NNO2=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO3=${tmp%%.*}
    tmp=${tmp#*.}
    typeset -i NNO4=${tmp%%.*}

    typeset -i16 NNF1
    let NNF1=${NNO1}
    typeset -i16 NNF2
    let NNF2=${NNO2}
    typeset -i16 NNF3
    let NNF3=${NNO3}
    typeset -i16 NNF4
    let NNF4=${NNO4}
    typeset +i16 NNF1
    typeset +i16 NNF2
    typeset +i16 NNF3
    typeset +i16 NNF4
    NNF1=${NNF1#16\#}
    NNF2=${NNF2#16\#}
    NNF3=${NNF3#16\#}
    NNF4=${NNF4#16\#}
    if [ ${#NNF1} -eq 1 ]
    then
        NNF1="0${NNF1}"
    fi
    if [ ${#NNF2} -eq 1 ]
    then
        NNF2="0${NNF2}"
```

예 18–1 pntadm 명령이 포함된 addclient.ksh 스크립트 (계속)

```
fi
if [ ${#NNF3} -eq 1 ]
then
    NNF3="0${NNF3}"
fi
if [ ${#NNF4} -eq 1 ]
then
    NNF4="0${NNF4}"
fi
typeset -i16 NN
let NN=16#${NNF1}${NNF2}${NNF3}${NNF4}
unset NNF1 NNF2 NNF3 NNF4

typeset -i NNO1=${2%%.*}
tmp=${2#*.}
typeset -i NNO2=${tmp%%.*}
tmp=${tmp#*.}
typeset -i NNO3=${tmp%%.*}
tmp=${tmp#*.}
typeset -i NNO4=${tmp%%.*}
typeset -i16 NNF1
let NNF1=${NNO1}
typeset -i16 NNF2
let NNF2=${NNO2}
typeset -i16 NNF3
let NNF3=${NNO3}
typeset -i16 NNF4
let NNF4=${NNO4}
typeset +i16 NNF1
typeset +i16 NNF2
typeset +i16 NNF3
typeset +i16 NNF4
NNF1=${NNF1#16\#}
NNF2=${NNF2#16\#}
NNF3=${NNF3#16\#}
NNF4=${NNF4#16\#}
if [ ${#NNF1} -eq 1 ]
then
    NNF1="0${NNF1}"
fi
if [ ${#NNF2} -eq 1 ]
then
    NNF2="0${NNF2}"
fi
if [ ${#NNF3} -eq 1 ]
then
    NNF3="0${NNF3}"
fi
if [ ${#NNF4} -eq 1 ]
then
    NNF4="0${NNF4}"
fi
typeset -i16 NC
let NC=16#${NNF1}${NNF2}${NNF3}${NNF4}
typeset -i10 ANS
let ANS=${NC}-${NN}
```

예 18–1 pntadm 명령이 포함된 addclient.ksh 스크립트 *(계속)*

```
    print - $ANS
}

#
# Check usage.
#
if [ "$#" != 3 ]
then
    print "This script is used to add client entries to a DHCP network"
    print "table by utilizing the pntadm batch facilty.\n"
    print "usage: $0 network start_ip entries\n"
    print "where: network is the IP address of the network"
        print "       start_ip is the starting IP address \n"
        print "       entries is the number of the entries to add\n"
    print "example: $0 10.148.174.0 10.148.174.1 254\n"
    return
fi

#
# Use input arguments to set script variables.
#
NETWORK=$1
START_IP=$2
typeset -i STRTNUM=‘client_index ${NETWORK} ${START_IP}‘
let ENDNUM=${STRTNUM}+$3
let ENTRYNUM=${STRTNUM}
BATCHFILE=/tmp/batchfile.$$
MACRO=‘uname -n‘

#
# Check if mask in netmasks table. First try
# for network address as given, in case VLSM
# is in use.
#
NETMASK=‘get_netmask ${NETWORK}‘
if [ -z "${NETMASK}" ]
then
    get_default_class ${NETWORK} | read DEFNET DEFMASK
    # use the default.
    if [ "${DEFNET}" != "${NETWORK}" ]
    then
        # likely subnetted/supernetted.
        print - "\n\n###\tWarning\t###\n"
        print - "Network ${NETWORK} is netmasked, but no entry was found  \n
              in the 'netmasks' table; please update the 'netmasks'  \n
              table in the appropriate nameservice before continuing. \n
              (See /etc/nsswitch.conf.) \n" >&2
        return 1
    else
        # use the default.
        NETMASK="${DEFMASK}"
    fi
fi

#
# Create a batch file.
```

예 18–1 pntadm 명령이 포함된 addclient.ksh 스크립트 (계속)

```
#
print -n "Creating batch file "
while [ ${ENTRYNUM} -lt ${ENDNUM} ]
do
    if [ $((${ENTRYNUM}-${STRTNUM}))%50 -eq 0 ]
    then
        print -n "."
    fi

    CLIENTIP=‘get_addr ${NETWORK} ${NETMASK} ${ENTRYNUM}‘
    print "pntadm -A ${CLIENTIP} -m ${MACRO} ${NETWORK}" >> ${BATCHFILE}
    let ENTRYNUM=${ENTRYNUM}+1
done
print " done.\n"

#
# Run pntadm in batch mode and redirect output to a temporary file.
# Progress can be monitored by using the output file.
#
print "Batch processing output redirected to ${BATCHFILE}"
print "Batch processing started."

pntadm -B ${BATCHFILE} -v > /tmp/batch.out 2 >&1

print "Batch processing completed."
```

DHCP 서비스에서 사용된 파일

다음 표는 DHCP와 연관된 파일을 나열합니다.

표 18–2 DHCP 데몬 및 명령에서 사용된 파일 및 테이블

파일 또는 테이블 이름	설명
dhcptab	레거시 Sun DHCP 전용: 옵션과 함께 지정된 값으로 기록된(이후 매크로로 그룹화됨) DHCP 구성 정보의 테이블을 지칭하는 일반 용어입니다. dhcptab 테이블의 이름과 위치는 DHCP 정보에 사용할 데이터 저장소에 의해 결정됩니다. 자세한 내용은 dhcptab(4) 매뉴얼 페이지를 참조하십시오.
DHCP 네트워크 테이블	레거시 Sun DHCP 전용: IP 주소를 클라이언트 ID 및 구성 옵션에 매핑합니다. 네트워크의 IP 주소(예: 10.21.32.0)에 따라 DHCP 네트워크 테이블 이름이 지정됩니다. dhcp_network라는 파일이 없습니다. DHCP 네트워크 테이블의 이름과 위치는 DHCP 정보에 사용할 데이터 저장소에 의해 결정됩니다. 자세한 내용은 dhcp_network(4) 매뉴얼 페이지를 참조하십시오.
/etc/dhcp/eventhook	레거시 Sun DHCP 전용: dhcpagent 데몬이 자동으로 실행할 수 있는 스크립트 또는 실행 파일입니다. 자세한 내용은 dhcpagent(1M) 매뉴얼 페이지를 참조하십시오.

표 18–2 DHCP 데몬 및 명령에서 사용된 파일 및 테이블 *(계속)*

파일 또는 테이블 이름	설명
/etc/inet/dhcpd4.conf /etc/inet/dhcpd6.conf	ISC DHCP 전용: ISC DHCP 서버 dhcpd에 대한 구성 정보를 포함합니다. 자세한 내용은 dhcpd.conf(5) 매뉴얼 페이지를 참조하십시오.
/etc/inet/dhcpsvc.conf	레거시 Sun DHCP 전용: DHCP 데몬의 시작 옵션 및 데이터 저장소 정보를 저장합니다. 이 파일은 수동으로 편집하면 안됩니다. dhcpconfig 명령을 사용하여 시작 옵션을 변경합니다. 자세한 내용은 dhcpsvc.conf(4) 매뉴얼 페이지를 참조하십시오.
nsswitch.conf	이름 서비스 데이터베이스의 위치 및 다양한 종류의 정보에 대해 이름 서비스를 검색하는 순서를 지정합니다. nsswitch.conf 파일은 DHCP 서버를 구성할 때 정확한 구성 정보를 가져오기 위해 읽습니다. 이 파일은 /etc 디렉토리에 있습니다. 자세한 내용은 nsswitch.conf(4) 매뉴얼 페이지를 참조하십시오.
resolv.conf	DNS 질의를 해결하는 데 사용되는 정보를 포함합니다. DHCP 서버 구성 중 DNS 도메인 및 DNS 서버에 대한 정보를 얻기 위해 이 파일이 참조됩니다. 이 파일은 /etc 디렉토리에 있습니다. 자세한 내용은 resolv.conf(4) 매뉴얼 페이지를 참조하십시오.
dhcp.*interface*	dhcp.*interface* 파일 이름에 지정된 클라이언트 네트워크 인터페이스에 DHCP가 사용됨을 나타냅니다. 예를 들어, dhcp.qe0이라는 파일의 존재는 DHCP가 qe0 인터페이스에서 사용될 것임을 나타냅니다. dhcp.*interface* 파일은 클라이언트에서 DHCP를 시작하는 데 사용되는 ifconfig 명령에 옵션으로 전달되는 명령을 포함할 수 있습니다. 이 파일은 DHCP 클라이언트 시스템의 /etc 디렉토리에 있습니다. 관련 매뉴얼 페이지가 없습니다. dhcp(5)를 참조하십시오.
/etc/dhcp/*interface*.dhc /etc/dhcp/*interface*.dh6	제공된 네트워크 인터페이스에 대해 DHCP에서 얻은 구성 매개변수를 포함합니다. DHCPv4의 경우 파일 이름이 dhc로 끝납니다. DHCPv6의 경우 파일 이름이 dh6으로 끝납니다. 인터페이스의 IP 주소 임대를 삭제할 때 /etc/dhcp/*interface* .dhc에 현재 구성 정보를 캐싱합니다. 예를 들어, DHCP가 qe0 인터페이스에 사용된 경우 dhcpagent가 /etc/dhcp/qe0.dhc에 구성 정보를 캐싱합니다. 다음에 인터페이스에서 DHCP를 시작할 때 임대가 만료되지 않았을 경우 클라이언트가 캐시된 구성을 사용하도록 요청합니다. DHCP 서버가 요청을 거부하면 클라이언트가 DHCP 임대 협상의 표준 프로세스를 시작합니다.
/etc/default/dhcpagent	dhcpagent 클라이언트 데몬에 대한 매개변수 값을 설정합니다. 매개변수에 대한 자세한 내용은 /etc/default/dhcpagent 파일 또는 dhcpagent(1M) 매뉴얼 페이지를 참조하십시오.

표 18–2 DHCP 데몬 및 명령에서 사용된 파일 및 테이블 *(계속)*

파일 또는 테이블 이름	설명
/etc/dhcp/inittab /etc/dhcp/inittab6	레거시 Sun DHCP 전용: 데이터 유형과 같은 DHCP 옵션 코드의 여러 측면을 정의하고 니모닉 레이블을 지정합니다. 파일 구문에 대한 자세한 내용은 dhcp_inittab(4) 매뉴얼 페이지를 참조하십시오. /etc/dhcp/inittab6은 DHCPv6 클라이언트에서 사용됩니다. 클라이언트에서 /etc/dhcp/inittab 파일의 정보를 dhcpinfo 명령에서 사용하여 정보 구독자에게 보다 의미있는 정보를 제공합니다. DHCP 서버 시스템에서 이 파일을 DHCP 데몬 및 관리 도구에서 사용하여 DHCP 옵션 정보를 얻습니다. /etc/dhcp/inittab 파일은 이전 릴리스에서 사용된 /etc/dhcp/dhcptags 파일을 대체합니다.
/var/db/isc-dhcp/dhcp4.leases /var/db/isc-dhcp/dhcp4.leases- /var/db/isc-dhcp/dhcp6.leases /var/db/isc-dhcp/dhcp6.leases-	ISC DHCP 전용: DHCPv4 및 DHCPv6 서버의 임대를 나열합니다. 파일 이름 끝에 "-"가 붙은 파일은 이전 복사본입니다.

DHCP 옵션 정보

DHCP 옵션 정보는 서버의 dhcptab 테이블, 클라이언트의 dhcptags 파일, 다양한 프로그램의 내부 테이블을 비롯한 여러 장소에 저장되어 왔습니다. Solaris 8 릴리스부터 옵션 정보는 /etc/dhcp/inittab 파일에 통합됩니다. 이 파일에 대한 자세한 내용은 dhcp_inittab(4) 매뉴얼 페이지를 참조하십시오.

DHCP 클라이언트는 DHCP inittab 파일로 dhcptags 파일을 대체합니다. 클라이언트는 이 파일을 사용하여 DHCP 패킷에 수신된 옵션 코드에 대한 정보를 가져옵니다. DHCP 서버의 in.dhcpd, snoop 및 dhcpmgr 프로그램에서도 inittab 파일을 사용합니다.

사이트가 영향을 받는지 여부 결정

DHCP를 사용하는 대부분의 사이트는 /etc/dhcp/inittab 파일로 전환해도 영향을 받지 **않습니다**. 다음 조건이 모두 성립되면 사이트가 영향을 받습니다.

- Solaris 8 릴리스보다 오래된 Oracle Solaris 릴리스에서 업그레이드하려고 계획합니다.
- 이전에 새 DHCP 옵션을 만들었습니다.
- /etc/dhcp/dhcptags 파일을 수정했으며 변경 사항을 유지하려고 합니다.

업그레이드할 때 업그레이드 로그가 dhcptags 파일이 수정되었으며 DHCP inittab 파일을 변경해야 한다고 알립니다.

dhcptags와 inittab 파일의 차이점

inittab 파일에는 dhcptags 파일보다 더 많은 정보가 들어 있습니다. 또한 inittab 파일은 다른 구문을 사용합니다.

dhcptags 항목의 예는 다음과 같습니다.

```
33 StaticRt - IPList Static_Routes
```

33은 DHCP 패킷에 전달되는 숫자 코드입니다. StaticRt는 옵션 이름입니다. IPList는 StaticRt의 데이터 유형이 IP 주소 목록이어야 함을 나타냅니다. Static_Routes는 더 많은 설명이 포함된 이름입니다.

inittab 파일은 각 옵션을 설명하는 한 행으로 된 레코드로 구성됩니다. 이 형식은 dhcptab의 기호를 정의하는 형식과 유사합니다. 다음 표는 inittab 파일의 구문을 설명합니다.

옵션	설명
option-name	옵션 이름입니다. 옵션 이름은 옵션 범주 내에서 고유해야 하며 Standard(표준), Site(사이트) 및 Vendor(공급업체) 범주의 옵션 이름과 겹치지 않아야 합니다. 예를 들어, 이름이 같은 두 개의 Site(사이트) 옵션을 가질 수 없으며 Standard(표준) 옵션과 이름이 같은 Site(사이트) 옵션을 만들 수 없습니다.
category	옵션이 속한 이름 공간을 식별합니다. Standard(표준), Site(사이트), Vendor(공급업체), Field(필드) 또는 Internal(내부) 중 하나여야 합니다.
code	네트워크를 통해 전송될 때 옵션을 식별합니다. 대부분의 경우 코드는 범주 없이 옵션을 고유하게 식별합니다. 하지만 Field(필드) 또는 Internal(내부)과 같은 내부 범주인 경우 코드를 다른 용도로 사용할 수 있습니다. 코드는 전역적으로 고유하지 않을 수 있습니다. 코드는 옵션 범주 내에서 고유해야 하며 Standard(표준) 및 Site(사이트) 필드의 코드와 겹쳐지지 않아야 합니다.
type	이 옵션과 연관된 데이터를 설명합니다. 유효한 유형은 IP, ASCII, Octet, Boolean, Unumber8, Unumber16, Unumber32, Unumber64, Snumber8, Snumber16, Snumber32 및 Snumber64입니다. 숫자인 경우 제일 앞의 U 또는 S는 숫자가 부호가 없는지 있는지를 나타냅니다. 맨 뒷자리는 숫자에 포함된 비트 수를 나타냅니다. 예를 들어, Unumber8은 부호 없는 8비트 숫자입니다. 유형은 대소문자를 구분하지 않습니다.
granularity	이 옵션에서 몇 개의 데이터 단위가 전체 값을 구성하는지 나타냅니다.
maximum	이 옵션에 허용되는 전체 값 개수를 나타냅니다. 0은 한도가 없음을 나타냅니다.
consumers	이 정보를 사용할 수 있는 프로그램을 설명합니다. consumers는 sdmi로 설정되어야 합니다. 여기에서 각 알파벳은 다음 프로그램을 나타냅니다.

```
s    snoop
d    in.dhcpd
m    dhcpmgr
i    dhcpinfo
```

다음은 inittab 항목의 예입니다.

```
StaticRt - Standard, 33, IP, 2, 0, sdmi
```

이 항목은 StaticRt라는 이름의 옵션을 설명합니다. 이 옵션은 Standard(표준) 범주에 속하며 옵션 코드는 33입니다. type이 IP이고, granularity가 2이고, maximum이 무한(0)이므로 예상되는 데이터는 한도가 없는 IP 주소 쌍입니다. 이 옵션의 consumers는 sdmi, 즉 snoop, in.dhcpd, dhcpmgr 및 dhcpinfo입니다.

dhcptags 항목을 inittab 항목으로 변환

이전에 dhcptags 파일에 항목을 추가한 경우 사이트에 추가한 옵션을 계속 사용하려면 새 inittab 파일에 해당 항목을 추가해야 합니다. 다음 예에서는 샘플 dhcptags 항목이 inittab 형식으로 어떻게 표현되는지 보여줍니다.

네트워크에 연결된 팩스에 대해 다음 dhcptags 항목을 추가했다고 가정합니다.

```
128 FaxMchn - IP Fax_Machine
```

코드 128은 옵션이 Site(사이트) 범주에 속해야 함을 의미합니다. 옵션 이름은 FaxMchn이고 데이터 유형은 IP입니다.

해당 inittab 항목은 다음과 같을 수 있습니다.

```
FaxMchn SITE, 128, IP, 1, 1, sdmi
```

granularity가 1이고 maximum이 1이므로 이 옵션에는 하나의 IP 주소가 예상됩니다.

제 4 부

IP 보안

이 절에서는 네트워크 보안을 중점적으로 다룹니다. IPsec(IP security architecture)는 패킷 레벨에서 네트워크를 보호합니다. IKE(Internet Key Exchange)는 IPsec에 대한 키를 관리합니다. Oracle Solaris의 IP 필터 기능은 방화벽을 제공합니다.

◆ ◆ ◆ 19 장

19 IP 보안 아키텍처(개요)

IPsec(IP Security Architecture)는 IPv4 및 IPv6 네트워크 패킷에서 IP 데이터그램에 대한 암호화 보호를 제공합니다.

이 장은 다음 정보를 포함합니다.

네트워크에서 IPsec를 구현하려면 20 장, “IPsec 구성(작업)”을 참조하십시오. 참조 정보는 21 장, “IP 보안 아키텍처(참조)”를 참조하십시오.

IPsec의 새로운 기능

Solaris 10 4/09: 이번 릴리스부터 SMF(서비스 관리 기능)에서 IPsec가 서비스 세트로 관리됩니다.

기본적으로 시스템 부트 시 두 개의 IPsec 서비스가 사용으로 설정됩니다.

- `svc:/network/ipsec/policy:default`
- `svc:/network/ipsec/ipsecalgs:default`

기본적으로 키 관리 서비스는 시스템 부트 시 사용 안함으로 설정됩니다.

- `svc:/network/ipsec/manual-key:default`
- `svc:/network/ipsec/ike:default`

SMF에서 IPsec 정책을 활성화하려면 다음 단계를 수행하십시오.

1. IPsec 정책 항목을 `ipsecinit.conf` 파일에 추가합니다.
2. IKE(인터넷 키 교환)를 구성하거나 키를 수동으로 구성합니다.
3. IPsec 정책 서비스를 새로 고칩니다.
4. 키 관리 서비스를 사용으로 설정합니다.

SMF에 대한 자세한 내용은 **Oracle Solaris 관리: 기본 관리**의 18 장, "서비스 관리(개요)"를 참조하십시오. 또한 `smf(5)` 및 `svcadm(1M)` 매뉴얼 페이지를 참조하십시오.

이번 릴리스부터 `ipsecconf` 및 `ipseckey` 명령에는 해당 구성 파일의 구문을 확인하기 위한 `-c` 옵션이 포함됩니다. 또한 IPsec 및 IKE 관리를 위해 Network IPsec Management 권한 프로파일이 제공됩니다.

Solaris 10 7/07: 이번 릴리스부터 IPsec는 터널을 터널 모드로 완전히 구현하며 터널을 지원하는 유틸리티가 수정되었습니다.

- IPsec는 VPN(가상 사설망)에 대한 터널 모드로 터널을 구현합니다. 터널 모드에서 IPsec는 단일 NAT 뒤에서 여러 클라이언트를 지원합니다. 터널 모드에서 IPsec는 다른 공급업체에서 제공하는 IP-in-IP 터널의 구현과 상호 운용됩니다. IPsec는 계속해서 전송 모드의 터널을 지원하므로 이전 Solaris 릴리스와도 호환됩니다.
- 터널을 만드는 구문이 간소화되었습니다. IPsec 정책 관리를 위해 `ipsecconf` 명령이 확장되었습니다. `ifconfig` 명령은 더 이상 IPsec 정책을 관리하는 데 사용되지 않습니다.
- 이번 릴리스부터 `/etc/ipnodes` 파일이 제거되었습니다. `/etc/hosts` 파일을 사용하여 네트워크 IPv6 주소를 구성하십시오.

Solaris 10 1/06: 이번 릴리스부터 IKE는 RFC 3947 및 RFC 3948에 설명된 대로 NAT 순회 지원과 완전히 호환됩니다. IKE 작업에서는 성능이 향상된 암호화 프레임워크의 PKCS #11 라이브러리가 사용됩니다.

암호화 프레임워크는 metaslot을 사용하는 응용 프로그램을 위해 소프트 토큰 키 저장소를 제공합니다. IKE에서 metaslot이 사용될 때는 디스크, 연결된 보드 또는 소프트 토큰 키 저장소에 키를 저장하도록 선택할 수 있습니다.

- 소프트 토큰 키 저장소를 사용하려면 `cryptoadm(1M)` 매뉴얼 페이지를 참조하십시오.
- 새로운 Oracle Solaris 기능의 전체 목록은 **Oracle Solaris 10 1/13 새로운 기능**을 참조하십시오.

IPsec 소개

IPsec는 패킷을 인증하거나 패킷을 암호화하거나 둘 다 수행하여 IP 패킷을 보호합니다. IPsec는 IP 모듈 내에서 수행됩니다. 따라서 인터넷 응용 프로그램에서는 IPsec를 사용하도록 구성할 필요 없이 IPsec를 활용할 수 있습니다. 제대로 사용되면 IPsec는 네트워크 트래픽을 보호하는 효과적인 도구가 될 수 있습니다.

IPsec 보호에는 다음 주요 구성 요소가 관련됩니다.

- **보안 프로토콜** – IP 데이터그램 보호 방식입니다. AH(인증 헤더)는 IP 패킷의 해시를 포함하고 무결성을 보장합니다. 데이터그램의 컨텐츠는 암호화되지 않지만, 수신자에게 패킷 컨텐츠가 변경되지 않았음을 보장합니다. 또한 패킷이 발신자에 의해 보내졌음을 수신자에게 보장합니다. ESP(보안 페이로드 캡슐화)는 IP 데이터를 암호화하므로 패킷 전송 중 컨텐츠를 숨깁니다. 또한 ESP는 인증 알고리즘 옵션을 통해 데이터 무결성을 보장할 수 있습니다.
- **SA(보안 연관)** – 네트워크 트래픽의 특정 플로우에 적용되는 암호화 매개변수 및 IP 보안 프로토콜입니다. 각 SA는 SPI(Security Parameters Index)라는 고유한 참조를 가집니다.
- **SADB(보안 연결 데이터베이스)** – 보안 프로토콜과 IP 대상 주소 및 색인화 번호를 연결하는 데이터베이스입니다. 색인화 번호는 SPI(보안 매개변수 색인)라고 합니다. 이러한 세 가지 요소(보안 프로토콜, 대상 주소 및 SPI)는 적법한 IPsec 패킷을 고유하게 식별합니다. 데이터베이스는 패킷 대상에 도달하는 보호된 패킷을 수신자가 인식할 수 있도록 합니다. 또한 수신자는 데이터베이스의 정보를 사용하여 통신을 해독하고, 패킷이 변경되지 않았음을 확인하며, 패킷을 재어셈블하고, 패킷을 최종 대상에 전달합니다.
- **키 관리** – 암호화 알고리즘 및 SPI에 대한 키 생성 및 배포입니다.
- **보안 방식** – IP 데이터그램에서 데이터를 보호하는 인증 및 암호화 알고리즘입니다.
- **SPD(보안 정책 데이터베이스)** – 패킷에 적용되는 보호 레벨을 지정하는 데이터베이스입니다. SPD는 IP 트래픽을 필터링하여 패킷이 어떻게 처리되어야 하는지 결정합니다. 패킷은 폐기할 수 있습니다. 패킷은 투명하게 전달할 수 있습니다. 또는 패킷은 IPsec로 보호할 수 있습니다. 아웃바운드 패킷에 대해 SPD 및 SADB는 적용할 보호 레벨을 결정합니다. 인바운드 패킷에 대해 SPD는 패킷에 대한 보호 레벨이 합당한지 여부를 결정하는 데 도움을 줍니다. 패킷이 IPsec로 보호되는 경우 패킷을 해독하고 확인한 후 SPD를 참조합니다.

IPsec는 IP 대상 주소로 이동하는 IP 데이터그램에 보안 방식을 적용합니다. 수신자는 SADB의 정보를 사용하여 도달한 패킷이 적법한지 확인하고 해독합니다. 응용 프로그램에서는 IPsec를 호출하여 소켓별 레벨에서도 IP 데이터그램에 보안 방식을 적용할 수 있습니다.

포트의 소켓이 연결되고 나중에 해당 포트에 IPsec 정책이 적용될 경우 해당 소켓을 사용하는 트래픽은 IPsec로 보호되지 않습니다. 물론, IPsec 정책이 포트에 적용된 **이후** 포트에서 열린 소켓은 IPsec 정책으로 보호됩니다.

IPsec RFC

IETF(Internet Engineering Task Force)는 IP 계층에 대한 보안 아키텍처를 설명하는 여러 RFC(Requests for Comment)를 게시했습니다. 모든 RFC는 Internet Society에 의해 암호화됩니다. RFC에 대한 링크는 http://www.ietf.org/를 참조하십시오. 다음 RFC 목록은 일반적인 IP 보안 참조를 다룹니다.

- RFC 2411, "IP Security Document Roadmap," 1998년 11월
- RFC 2401, "Security Architecture for the Internet Protocol," 1998년 11월
- RFC 2402, "IP Authentication Header," 1998년 11월
- RFC 2406, "IP Encapsulating Security Payload (ESP)," 1998년 11월
- RFC 2408, "Internet Security Association and Key Management Protocol (ISAKMP)," 1998년 11월
- RFC 2407, "The Internet IP Security Domain of Interpretation for ISAKMP," 1998년 11월
- RFC 2409, "The Internet Key Exchange (IKE)," 1998년 11월
- RFC 3554, "On the Use of Stream Control Transmission Protocol (SCTP) with IPsec," 2003년 7월 [Oracle Solaris 10 릴리스에 구현되지 않음]

IPsec 용어

IPsec RFC는 시스템에서 IPsec를 구현할 때 알아두면 유용한 많은 용어를 정의합니다. 다음 표에서는 IPsec 용어 및 일반적으로 사용되는 약어를 나열하고 각 용어를 정의합니다. 키 협상에서 사용되는 용어 목록은 표 22–1을 참조하십시오.

표 19–1 IPsec 용어, 약어 및 사용

IPsec 용어	머리글자어	정의
보안 연결	SA	네트워크 트래픽의 특정 플로우에 적용되는 암호화 매개변수 및 IP 보안 프로토콜입니다. SA는 보안 프로토콜, 고유한 SPI(보안 매개변수 색인), IP 대상, 이렇게 3중으로 정의됩니다.
보안 연결 데이터베이스	SADB	모든 활성 보안 연결을 포함하는 데이터베이스입니다.
보안 매개변수 색인	SPI	보안 연결에 대한 색인화 값입니다. SPI는 동일한 IP 대상 및 보안 프로토콜을 가지는 SA 사이에서 구분되는 32비트 값입니다.
보안 정책 데이터베이스	SPD	아웃바운드 패킷 및 인바운드 패킷이 지정된 보호 레벨을 가지는지 여부를 결정하는 데이터베이스입니다.
키 교환		비대칭 암호화 알고리즘을 사용하여 키를 생성하는 프로세스입니다. 두 가지 주요 방식은 RSA 및 Diffie-Hellman입니다.

표 19–1 IPsec 용어, 약어 및 사용 (계속)

IPsec 용어	머리글자어	정의
Diffie-Hellman	DH	키 생성 및 키 인증을 허용하는 키 교환 알고리즘입니다. **인증된 키 교환**이라고도 합니다.
RSA	RSA	키 생성 및 키 배포를 허용하는 키 교환 알고리즘입니다. 프로토콜 이름은 Rivest, Shamir, Adleman 등 3인의 저작자 이름에서 따왔습니다.
인터넷 보안 연결 및 키 관리 프로토콜	ISAKMP	SA 속성 형식 설정과 SA 협상, 수정 및 삭제를 위한 공통 프레임워크입니다. ISAKMP는 IKE 교환 처리를 위한 IETF 표준입니다.

IPsec 패킷 플로우

그림 19–1은 IPsec가 아웃바운드 패킷에서 호출될 때 IP 주소 지정된 패킷이 IP 데이터그램의 일부로 진행되는지 보여줍니다. 플로우 다이어그램은 AH(authentication header) 및 ESP(encapsulating security payload) 엔티티를 어디에서 패킷에 적용할 수 있는지 보여줍니다. 이러한 엔티티를 적용하는 방법 및 알고리즘을 선택하는 방법은 다음 절에서 설명합니다.

그림 19–2는 IPsec 인바운드 프로세스를 보여줍니다.

그림 19–1 아웃바운드 패킷 프로세스에 적용된 IPsec

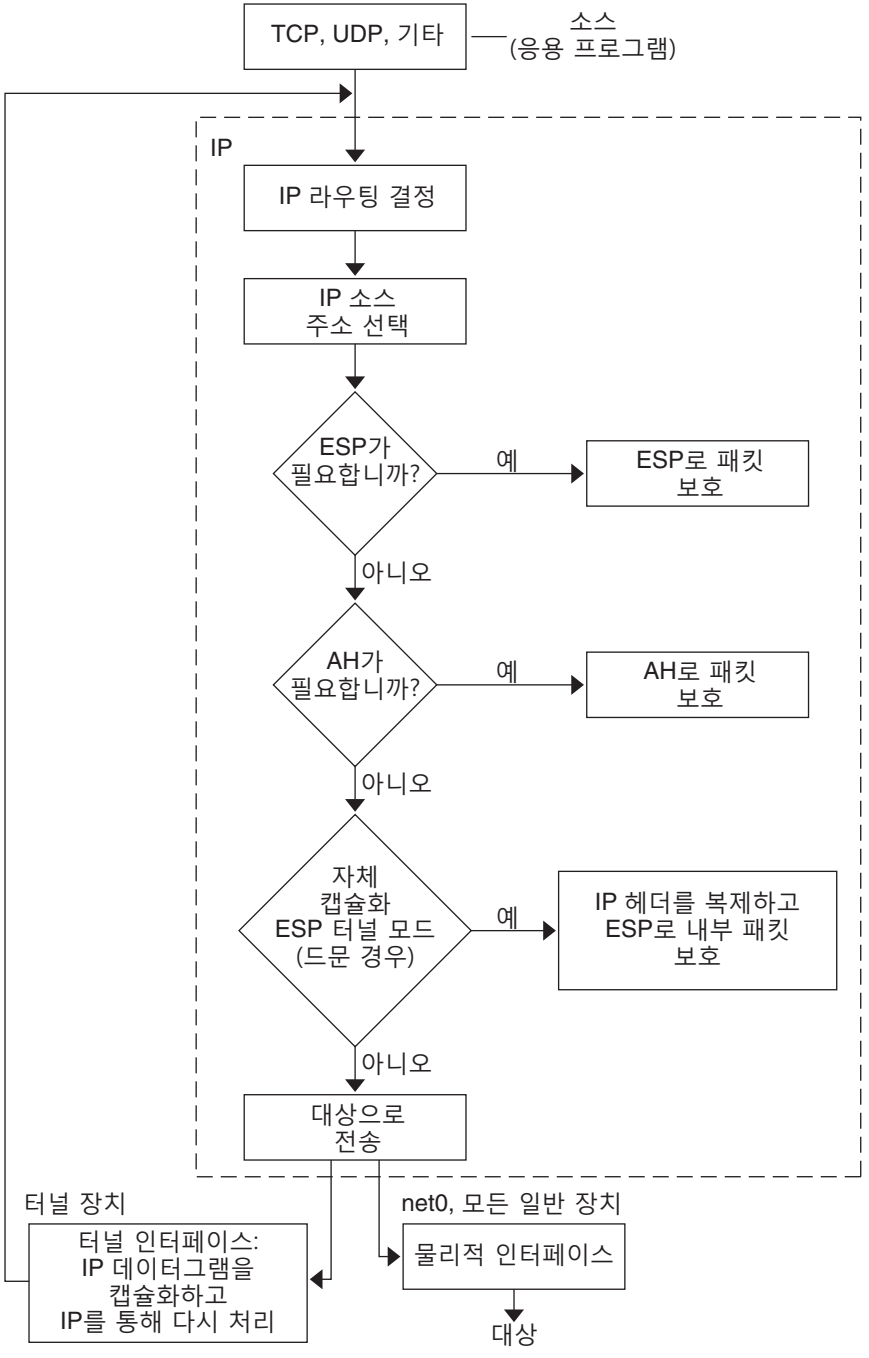
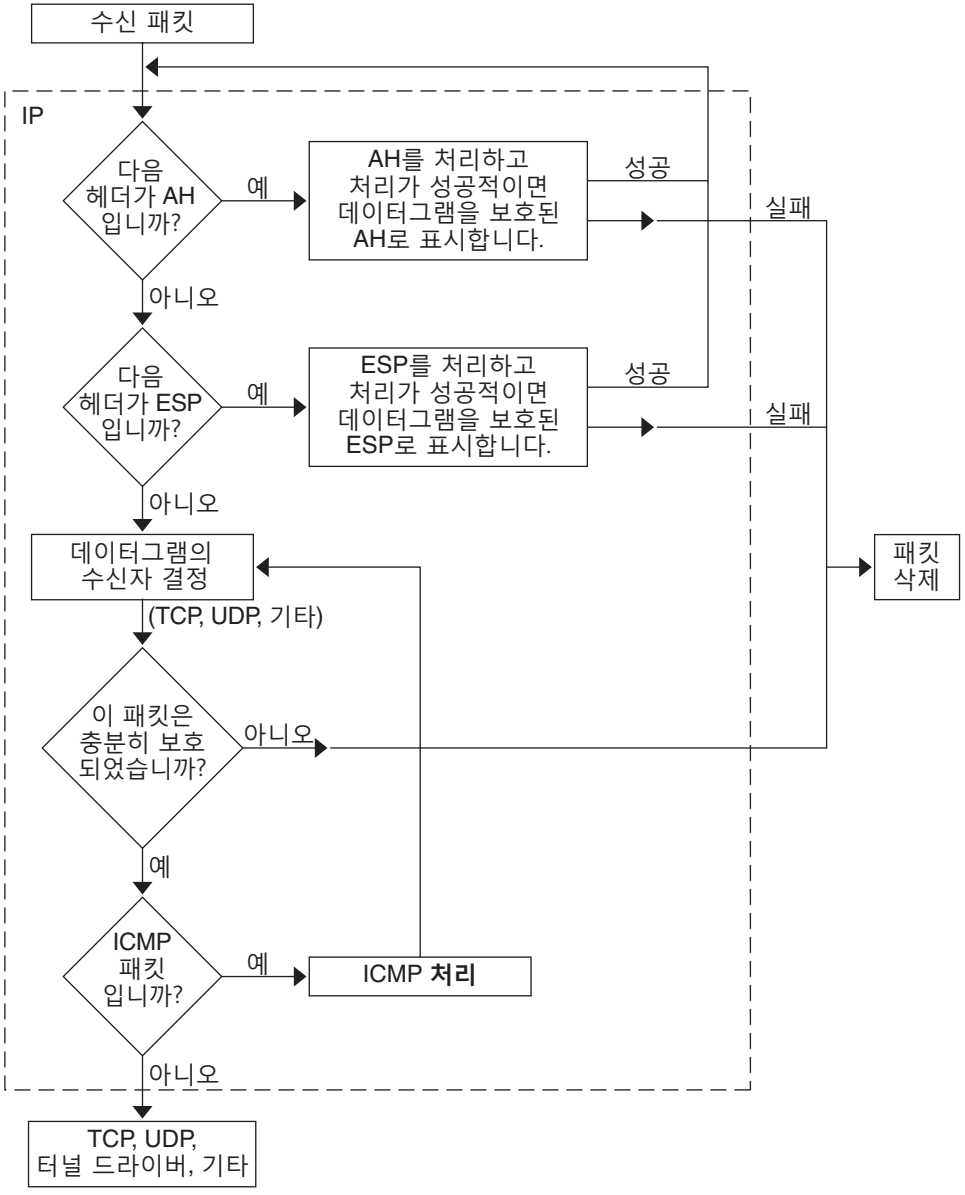


그림 19–2 인바운드 패킷 프로세스에 적용된 IPsec



IPsec 보안 연결

IPsec SA(**보안 연결**)는 통신 호스트에서 인식할 수 있는 보안 등록 정보를 지정합니다. 단일 SA는 한 방향의 데이터를 보호합니다. 단일 호스트 또는 그룹(멀티캐스트) 주소에 대한 보호입니다. 대부분의 통신은 피어 투 피어 또는 클라이언트-서버이므로 양방향에서 트래픽을 보호하려면 두 SA가 존재해야 합니다.

다음 세 가지 요소는 IPsec SA를 고유하게 식별합니다.

- 보안 프로토콜(AH 또는 ESP)
- 대상 IP 주소
- SPI(보안 매개변수 색인)

임의의 32비트 값인 SPI는 AH 또는 ESP 패킷으로 전송됩니다. `ipsecah(7P)` 및 `ipsecesp(7P)` 매뉴얼 페이지에서 AH 및 ESP로 보호되는 보호의 범위를 설명합니다. 무결성 체크섬 값은 패킷을 인증하는 데 사용됩니다. 인증을 실패할 경우 패킷은 삭제됩니다.

보안 연결은 SADB(**보안 연결 데이터베이스**)에 저장됩니다. 소켓 기반 관리 인터페이스인 `PF_KEY`는 권한이 부여된 응용 프로그램이 데이터베이스를 관리하도록 합니다. 예를 들어, IKE 응용 프로그램 및 `ipseckeys` 명령은 `PF_KEY` 소켓 인터페이스를 사용합니다.

- IPsec SADB에 대한 자세한 설명은 533 페이지 “IPsec에 대한 보안 연결 데이터베이스”를 참조하십시오.
- SADB를 관리하는 방법에 대한 자세한 내용은 `pf_key(7P)` 매뉴얼 페이지를 참조하십시오.

IPsec에서 키 관리

SA(보안 연결)에는 인증 및 암호화를 위한 키 입력 자료가 필요합니다. 이 키 입력 자료 관리를 **키 관리**라고 합니다. IKE(Internet Key Exchange) 프로토콜은 키 관리를 자동으로 처리합니다. 또한 `ipseckey` 명령을 사용하여 수동으로 키를 관리할 수 있습니다.

IPv4 및 IPv6 소켓에 대한 SA에서는 이러한 두 가지 키 관리 방식을 사용할 수 있습니다. 수동 키 관리를 사용해야 하는 분명한 이유가 없다면 IKE를 사용하는 것이 좋습니다.

Oracle Solaris의 SMF(서비스 관리 기능) 기능은 IPsec에 대한 다음 키 관리 서비스를 제공합니다.

- `svc:/network/ipsec/ike:default` 서비스 – 자동 키 관리를 위한 SMF 서비스입니다. `ike` 서비스는 `in.iked` 데몬을 실행하여 자동 키 관리를 제공합니다. IKE에 대한 설명은 22 장, “Internet Key Exchange(개요)”를 참조하십시오. `in.iked` 데몬에 대한 자세한 내용은 `in.iked(1M)` 매뉴얼 페이지를 참조하십시오. `ike` 서비스에 대한 자세한 내용은 589 페이지 “IKE 서비스”를 참조하십시오.
- `svc:/network/ipsec/manual-key:default` 서비스 – 수동 키 관리를 위한 SMF 서비스입니다. `manual-key` 서비스는 `ipseckey` 명령을 다양한 옵션과 함께 실행하여 키를 수동으로 관리합니다. `ipseckey` 명령에 대한 설명은 533 페이지 “IPsec에서 SA 생성을 위한 유틸리티”를 참조하십시오. `ipseckey` 명령 옵션에 대한 자세한 설명은 `ipseckey(1M)` 매뉴얼 페이지를 참조하십시오.

Solaris 10 4/09 이전 릴리스에서는 `in.iked` 및 `ipseckey` 명령으로 키 입력 자료가 관리됩니다.

- `in.iked` 데몬은 자동 키 관리를 제공합니다. IKE에 대한 설명은 22 장, “Internet Key Exchange(개요)”를 참조하십시오. `in.iked` 데몬에 대한 자세한 내용은 `in.iked(1M)` 매뉴얼 페이지를 참조하십시오.
- `ipseckey` 명령은 수동 키 관리를 제공합니다. 명령에 대한 설명은 533 페이지 “IPsec에서 SA 생성을 위한 유틸리티”를 참조하십시오. `ipseckey` 명령 옵션에 대한 자세한 설명은 `ipseckey(1M)` 매뉴얼 페이지를 참조하십시오.

IPsec 보호 방식

IPsec는 데이터 보호를 위한 두 가지 보안 프로토콜을 제공합니다.

- AH(인증 헤더)
- ESP(Encapsulating Security Payload)

AH는 인증 알고리즘으로 데이터를 보호합니다. ESP는 암호화 알고리즘으로 데이터를 보호합니다. ESP는 인증 방식과 함께 사용할 수 있으며 그렇게 사용해야 합니다. NAT를 통과하지 않는 경우 ESP와 AH를 결합할 수 있습니다. 그렇지 않은 경우 인증 알고리즘 및 암호화 방식을 ESP와 함께 사용할 수 있습니다.

인증 헤더

인증 헤더는 IP 데이터그램에 데이터 인증, 강력한 무결성 및 재생 보호 기능을 제공합니다. AH는 IP 데이터그램의 많은 부분을 보호합니다. 다음 그림에 나온 대로 AH는 IP 헤더와 전송 헤더 사이에 삽입됩니다.

전송 헤더는 TCP, UDP, SCTP 또는 ICMP가 될 수 있습니다. 터널이 사용되는 경우 전송 헤더는 다른 IP 헤더가 될 수 있습니다.

ESP(Encapsulating Security Payload)

ESP(보안 페이로드 캡슐화) 모듈은 ESP가 캡슐화하는 컨텐츠에 대한 기밀성을 제공합니다. 또한 ESP는 AH가 제공하는 서비스도 제공합니다. 하지만 ESP는 ESP가 캡슐화하는 데이터그램의 부분에 대해서만 보호 기능을 제공합니다. ESP는 보호된 패킷의 무결성을 위해 선택적 인증 서비스를 제공합니다. ESP는 암호화 지원 기술을 사용하므로 ESP를 제공하는 시스템은 가져오기 및 내보내기 제어 규칙에 종속될 수 있습니다.

ESP는 데이터를 캡슐화하므로 ESP는 다음 그림에 나온 대로 데이터그램에서 시작 이후의 데이터만 보호합니다.

암호화됨

TCP 패킷에서 ESP는 TCP 헤더 및 해당 데이터만 캡슐화합니다. 패킷이 IP-in-IP 데이터그램인 경우 ESP는 내부 IP 데이터그램을 보호합니다. 소켓별 정책에서는 **자체 캡슐화**를 허용하므로 ESP에서 필요할 때 ESP가 IP 옵션을 캡슐화할 수 있습니다.

자체 캡슐화가 설정되면 IP 헤더의 복사본이 IP-in-IP 데이터그램을 생성하게 됩니다. 예를 들어, 자체 캡슐화가 TCP 소켓에서 설정되지 않은 경우 데이터그램은 다음 형식으로 보내집니다.

```
[ IP(a -> b) options + TCP + data ]
```

자체 캡슐화가 TCP 소켓에서 설정된 경우 데이터그램은 다음 형식으로 보내집니다.

```
[ IP(a -> b) + ESP [ IP(a -> b) options + TCP + data ] ]
```

자세한 내용은 467 페이지 “IPsec의 전송 및 터널 모드”를 참조하십시오.

AH 및 ESP를 사용할 때 보안 고려 사항

다음 표는 AH 및 ESP에서 제공하는 보호 기능을 비교한 것입니다.

표 19–2 IPsec에서 AH 및 ESP로 제공되는 보호 기능

프로토콜	패킷 범위	보호	공격 방어
AH	IP 헤더에서 전송 헤더까지 패킷을 보호합니다.	강력한 무결성, 데이터 인증을 제공합니다. ■ 발신자가 보낸 컨텐츠를 그대로 수신자가 수신할 수 있도록 합니다. ■ AH에서 재생 보호를 사용으로 설정하지 않을 경우 재생 공격에 취약합니다.	재생, 잘라내기 및 붙여넣기
ESP	데이터그램에서 ESP 시작 이후의 패킷을 보호합니다.	암호화 옵션을 사용하여 IP 페이로드를 암호화합니다. 기밀성을 유지합니다.	도청
		인증 옵션을 사용하여 AH와 동일한 페이로드 보호 기능을 제공합니다.	재생, 잘라내기 및 붙여넣기
		두 옵션을 모두 사용하면 강력한 무결성, 데이터 인증 및 기밀성을 제공할 수 있습니다.	재생, 잘라내기 및 붙여넣기, 도청

IPsec의 인증 및 암호화 알고리즘

IPsec 보안 프로토콜에서는 인증 및 암호화의 두 가지 알고리즘 유형을 사용합니다. AH 모듈은 인증 알고리즘을 사용합니다. ESP 모듈은 인증 알고리즘과 함께 암호화를 사용할 수 있습니다. 시스템의 알고리즘 및 해당 등록 정보 목록은 `ipsecalgs` 명령을 사용하여 얻을 수 있습니다. 자세한 내용은 `ipsecalgs(1M)` 매뉴얼 페이지를 참조하십시오. 또한 `getipsecalgbyname(3NSL)` 매뉴얼 페이지에 설명된 기능을 사용하여 알고리즘의 등록 정보를 검색할 수 있습니다.

IPsec는 암호화 프레임워크를 사용하여 알고리즘에 액세스합니다. 암호화 프레임워크는 다른 서비스와 함께 알고리즘에 대한 중앙 저장소를 제공합니다. 프레임워크를 통해 IPsec는 높은 성능의 암호화 하드웨어 가속기를 활용할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- **System Administration Guide: Security Services**의 13 장, “Oracle Solaris Cryptographic Framework (Overview)”
- **Developer’s Guide to Oracle Solaris 10 Security**의 8 장, “Introduction to the Oracle Solaris Cryptographic Framework”

IPsec의 인증 알고리즘

인증 알고리즘은 데이터 및 키를 기반으로 하는 무결성 체크섬 값 또는 **다이제스트**를 생성합니다. AH 모듈은 인증 알고리즘을 사용합니다. ESP 모듈은 인증 알고리즘도 사용할 수 있습니다.

IPsec의 암호화 알고리즘

암호화 알고리즘은 키로 데이터를 암호화합니다. IPsec의 ESP 모듈은 암호화 알고리즘을 사용합니다. 알고리즘은 **블록 크기** 단위로 데이터에 작동합니다.

각 Oracle Solaris 릴리스마다 서로 다른 기본 암호화 알고리즘을 제공합니다.

Solaris 10 7/07 릴리스부터는 Solaris 암호화 키트 컨텐츠가 Solaris 설치 매체로 설치됩니다. 이 릴리스에는 SHA2 인증 알고리즘 sha256, sha384 및 sha512가 추가되었습니다. SHA2 구현은 RFC 4868 사양을 준수합니다. 이 릴리스에는 또한 더 큰 Diffie-Hellman 그룹인 2048비트(그룹 14), 3072비트(그룹 15) 및 4096비트(그룹 16)가 추가되었습니다. CoolThreads 기술이 포함된 Oracle Sun 시스템은 2048비트 그룹만 가속화합니다.

주의 – Solaris 10 7/07 릴리스부터는 Solaris 암호화 키트를 시스템에 추가하지 마십시오. 이 키트는 시스템의 암호화에 대한 패치 레벨을 다운그레이드합니다. 이 키트는 시스템의 암호화와 호환되지 않습니다.

IPsec 보호 정책

IPsec 보호 정책에서는 모든 보안 방식을 사용할 수 있습니다. IPsec 정책은 다음 레벨에서 적용할 수 있습니다.

- 시스템 전역 레벨
- 소켓별 레벨

IPsec는 아웃바운드 데이터그램 및 인바운드 데이터그램에 시스템 전역 정책을 적용합니다. 아웃바운드 데이터그램은 보호 기능과 함께 또는 보호 기능 없이 보낼 수 있습니다. 보호 기능이 적용된 경우 알고리즘은 특정 또는 비특정입니다. 시스템에서 알고 있는 추가 데이터로 인해 아웃바운드 데이터그램에 추가 규칙을 적용할 수 있습니다. 인바운드 데이터그램은 수용하거나 삭제할 수 있습니다. 인바운드 데이터그램의 삭제 또는 수용 결정은 때때로 겹치거나 충돌하는 여러 조건을 기준으로 합니다. 충돌은 먼저 구문 분석된 규칙을 결정하여 해결됩니다. 트래픽이 모든 기타 정책을 우회해야 하는 정책 항목 상태일 때를 제외하고 트래픽은 자동으로 수용됩니다.

일반적으로 데이터그램을 보호하는 정책은 우회할 수 있습니다. 시스템 전역 정책에서 예외 사항을 지정하거나 소켓별 정책에서 우회를 요청할 수 있습니다. 시스템 내부 트래픽의 경우 정책이 적용되지만 실제 보안 방식은 적용되지 않습니다. 대신 시스템간 패킷에 대한 아웃바운드 정책은 해당 방식이 적용된 인바운드 패킷으로 변환됩니다.

`ipsecinit.conf` 파일 및 `ipsecconf` 명령을 사용하여 IPsec 정책을 구성합니다. 자세한 내용 및 예는 `ipsecconf(1M)` 매뉴얼 페이지를 참조하십시오.

IPsec의 전송 및 터널 모드

IPsec 표준에서는 **전송 모드** 및 **터널 모드**의 두 가지 고유 IPsec 작업 모드를 정의합니다. 모드는 패킷의 인코딩에 영향을 주지 않습니다. 패킷은 각 모드에서 AH, ESP 또는 둘 다로 보호됩니다. 모드는 내부 패킷이 IP 패킷일 때 정책 적용 면에서 다음과 같이 다릅니다.

- 전송 모드에서 외부 헤더는 내부 IP 패킷을 보호하는 IPsec 정책을 결정합니다.
- 터널 모드에서 내부 IP 패킷은 해당 컨텐츠를 보호하는 IPsec 정책을 결정합니다.

전송 모드에서 외부 헤더, 다음 헤더 및 다음 헤더가 지원하는 모든 포트는 IPsec 정책을 결정하는 데 사용될 수 있습니다. 실제로 IPsec는 두 IP 주소 사이에 서로 다른 전송 모드 정책을 적용하여 단일 포트를 세분화할 수 있습니다. 예를 들어, 다음 헤더가 포트를 지원하는 TCP인 경우 IPsec 정책을 외부 IP 주소의 TCP 정책에 대해 설정할 수 있습니다. 마찬가지로 다음 헤더가 IP 헤더인 경우 외부 헤더 및 내부 IP 헤더를 사용하여 IPsec 정책을 결정할 수 있습니다.

터널 모드는 IP-in-IP 데이터그램에 대해서만 작동합니다. 터널 모드의 터널링은 집에 있는 컴퓨터 작업자가 중앙 컴퓨터 위치에 연결할 때 유용할 수 있습니다. 터널 모드에서 IPsec 정책은 내부 IP 데이터그램의 컨텐츠에 적용됩니다. 서로 다른 내부 IP 주소에 대해 서로 다른 IPsec 정책을 적용할 수 있습니다. 즉, 내부 IP 헤더, 다음 헤더 및 다음 헤더가 지원하는 포트가 정책을 적용할 수 있습니다. 전송 모드와 달리 터널 모드에서는 외부 IP 헤더가 내부 IP 데이터그램의 정책을 결정하지 않습니다.

따라서 터널 모드에서 IPsec 정책은 라우터 뒤의 LAN 서브넷 및 이러한 서브넷의 포트에 대해 지정할 수 있습니다. 또한 IPsec 정책은 이러한 서브넷에 있는 특정 IP 주소(즉, 호스트)에 대해 지정할 수도 있습니다. 이러한 호스트의 포트도 특정 IPsec 정책을 가질 수 있습니다. 하지만 동적 경로 지정 프로토콜이 터널을 통해 실행되는 경우 피어 네트워크의 네트워크 토폴로지에 대한 뷰가 변경될 수 있으므로 서브넷 선택이나 주소 선택을 사용하지 마십시오. 변경되면 정적 IPsec 정책이 무효화됩니다. 정적 경로 구성을 포함하는 터널링 절차의 예는 496 페이지 “IPsec를 사용하여 VPN 보호(작업 맵)”를 참조하십시오.

Oracle Solaris에서 터널 모드는 IP 터널링 네트워크 인터페이스에만 적용할 수 있습니다. `ipsecconf` 명령은 IP 터널링 네트워크 인터페이스를 선택하기 위한 `tunnel` 키워드를 제공합니다. `tunnel` 키워드가 규칙에 존재하는 경우 해당 규칙에서 지정된 모든 선택기가 내부 패킷에 적용됩니다.

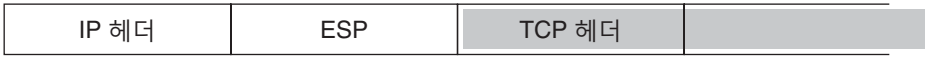
전송 모드에서는 ESP, AH 또는 둘 다 데이터그램을 보호할 수 있습니다.

다음 그림은 보호되지 않는 TCP 패킷의 IP 헤더를 보여줍니다.

그림 19–3 TCP 정보를 전달하는 보호되지 않는 IP 패킷

전송 모드에서 ESP가 다음 그림에 나온 대로 데이터를 보호합니다. 음영 영역은 패킷의 암호화된 부분을 나타냅니다.

그림 19–4 TCP 정보를 전달하는 보호된 IP 패킷



암호화됨

전송 모드에서 AH가 다음 그림에 나온 대로 데이터를 보호합니다.

그림 19–5 인증 헤더로 보호된 패킷

AH 보호는 전송 모드라도 IP 헤더의 대부분을 포함합니다.

터널 모드에서 전체 데이터그램은 IPsec 헤더의 보호 **내부**에 있습니다. 그림 19–3의 데이터그램은 다음 그림에 나온 대로 외부 IPsec 헤더(이 경우 ESP)로 터널 모드에서 보호됩니다.

그림 19–6 터널 모드에서 보호된 IPsec 패킷

암호화됨

`ipsecconf` 명령에는 터널을 터널 모드 또는 전송 모드로 설정하는 키워드가 포함되어 있습니다.

- 소켓별 정책에 대한 자세한 내용은 `ipsec`(7P) 매뉴얼 페이지를 참조하십시오.
- 소켓별 정책의 예는 480 페이지 “IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법”을 참조하십시오.

- 터널에 대한 자세한 내용은 ipsecconf(1M) 매뉴얼 페이지를 참조하십시오.
- 터널 구성의 예는 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"을 참조하십시오.

VPN(Virtual Private Networks) 및 IPsec

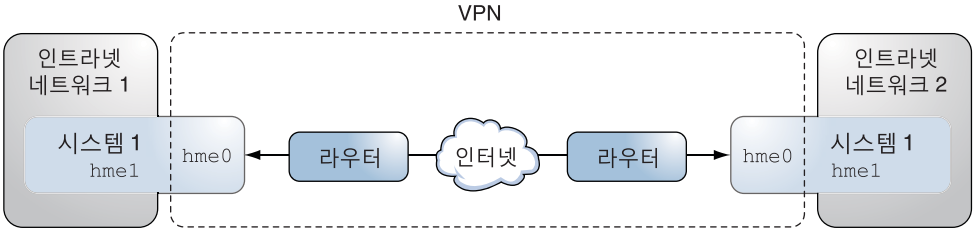
구성된 터널은 지점간 인터페이스입니다. 터널을 통해 한 IP 패킷을 다른 IP 패킷 내부에 캡슐화할 수 있습니다. 올바르게 구성된 터널에는 터널 소스와 터널 대상이 필요합니다. 자세한 내용은 tun(7M) 매뉴얼 페이지 및 IPv6 지원을 위한 터널 구성을 참조하십시오.

터널은 IP에 대한 분명한 물리적 인터페이스를 만듭니다. 물리적 링크의 무결성은 기본 보안 프로토콜에 의존합니다. SA(보안 연결)를 안전하게 설정할 경우 터널을 신뢰할 수 있습니다. 터널에서 나온 패킷은 터널 대상에 지정된 피어로부터 나왔어야 합니다. 이 신뢰가 존재할 경우 인터페이스별 IP 전달을 사용하여 VPN(가상 사설망)을 만들 수 있습니다.

IPsec 보호를 VPN에 추가할 수 있습니다. IPsec는 연결을 보호합니다. 예를 들어, VPN 기술을 사용하여 별도 네트워크의 사무실을 연결하는 조직에서는 IPsec를 추가하여 두 사무실 사이의 트래픽을 보호할 수 있습니다.

다음 그림은 IPsec를 사용하는 VPN의 두 사무실이 해당 네트워크 시스템에서 어떻게 배치되었는지 보여줍니다.

그림 19–7 VPN(가상 사설망)



설정 절차의 자세한 예는 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"을 참조하십시오.

IPv6 주소에 대한 유사한 예는 508 페이지 "IPv6을 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"을 참조하십시오.

IPsec 및 NAT 순회

IKE는 NAT 장치에 걸쳐 IPsec SA를 협상할 수 있습니다. 시스템이 NAT 장치 뒤에 있더라도 이 기능을 통해 시스템은 원격 네트워크에서 안전하게 연결할 수 있습니다. 예를 들어, 집에서 작업하거나 회의실에서 로그온하는 직원은 IPsec를 사용하여 트래픽을 보호할 수 있습니다.

NAT는 네트워크 주소 변환(network address translation)을 나타냅니다. NAT 장치를 사용하여 개인 내부 주소를 고유한 인터넷 주소로 변환할 수 있습니다. NAT는 호텔과 같은 인터넷 공용 액세스 지점에서 매우 일반적입니다. 자세한 내용은 606 페이지 “IP 필터의 NAT 기능 사용”을 참조하십시오.

NAT 장치가 통신 시스템 사이에 있을 때 IKE를 사용하는 기능을 NAT 순회 또는 NAT-T라고 합니다. Oracle Solaris 10 릴리스에서 NAT-T는 다음 제한 사항이 있습니다.

- NAT-T는 Sun Crypto Accelerator 4000 보드에서 제공하는 IPsec ESP 가속화를 활용할 수 없습니다. 하지만 Sun Crypto Accelerator 4000 보드의 IKE 가속화는 작동합니다.
- AH 프로토콜은 변경되지 않는 IP 헤더에 의존하므로 AH는 NAT-T와 함께 작동할 수 없습니다. ESP 프로토콜은 NAT-T와 함께 사용됩니다.
- NAT 장치는 특수 처리 규칙을 사용하지 않습니다. 특수한 IPsec 처리 규칙을 사용하는 NAT 장치는 NAT-T의 구현에 방해가 될 수 있습니다.
- NAT-T는 IKE 개시자가 NAT 장치의 뒤에 있는 시스템일 때만 작동합니다. 장치가 IKE 패킷을 장치 뒤의 해당 개별 시스템에 전달하도록 프로그래밍되지 않은 경우 IKE 응답자는 NAT 장치 뒤에 있을 수 없습니다.

다음 RFC는 NAT 기능 및 NAT-T의 제한 사항을 설명합니다. RFC의 사본은 `http://www.rfc-editor.org`에서 검색할 수 있습니다.

- RFC 3022, “Traditional IP Network Address Translator (Traditional NAT),” 2001년 1월
- RFC 3715, “IPsec-Network Address Translation (NAT) Compatibility Requirements,” 2004년 3월
- RFC 3947, “Negotiation of NAT-Traversal in the IKE,” 2005년 1월
- RFC 3948, “UDP Encapsulation of IPsec Packets,” 2005년 1월

NAT에 걸쳐 IPsec를 사용하려면 573 페이지 “모바일 시스템에 대한 IKE 구성(작업 맵)”을 참조하십시오.

IPsec 및 SCTP

Oracle Solaris는 SCTP(Streams Control Transmission Protocol)를 지원합니다. IPsec 정책 지정을 위한 SCTP 프로토콜 및 SCTP 포트 번호 사용은 지원되지만 안정적이지는 않습니다. RFC 3554에 지정된 SCTP에 대한 IPsec 확장 기능은 아직 구현되지 않았습니다. 이러한 제한 사항으로 인해 SCTP에 대한 IPsec 정책을 만들 때 복잡해질 수 있습니다.

SCTP는 단일 SCTP 연결 컨텍스트에서 여러 소스 및 대상 주소를 활용할 수 있습니다. IPsec 정책이 단일 소스나 단일 대상 주소에 적용된 경우 SCTP가 해당 연결의 소스나 대상 주소를 바꾸면 통신이 실패합니다. IPsec 정책은 원래 주소만 인식할 수 있습니다. SCTP에 대한 자세한 내용은 RFC 및 37 페이지 “SCTP 프로토콜”을 참조하십시오.

IPsec 및 Oracle Solaris 영역

공유 IP 영역의 경우, IPsec는 전역 영역에서 구성됩니다. IPsec 정책 구성 파일 `ipsecinit.conf`는 전역 영역에만 존재합니다. 파일에는 비전역 영역에 적용되는 항목과 전역 영역에 적용되는 항목이 있을 수 있습니다.

배타적 IP 영역의 경우, IPsec는 비전역 영역별로 구성됩니다.

영역에서 IPsec를 사용하는 방법에 대한 자세한 내용은 475 페이지 “IPsec를 사용하여 트래픽 보호(작업 맵)”를 참조하십시오. 영역에 대한 자세한 내용은 **System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones**의 16 장, “Introduction to Solaris Zones”를 참조하십시오.

IPsec 및 논리적 도메인

IPsec는 논리적 도메인에서 작동합니다. 논리적 도메인은 IPsec가 포함된 Oracle Solaris 버전(예: Oracle Solaris 10 릴리스)을 실행하고 있어야 합니다.

논리적 도메인을 만들려면 Oracle VM Server for SPARC(이전의 논리적 도메인)를 사용해야 합니다. 논리적 도메인 구성 방법에 대한 자세한 내용은 **Oracle VM Server for SPARC 2.2 관리 설명서** 또는 **Oracle VM Server for SPARC 2.0 Administration Guide**를 참조하십시오.

IPsec 유틸리티 및 파일

표 19–3에서는 IPsec를 구성하고 관리하는 데 사용되는 파일, 명령 및 서비스 식별자를 설명합니다. 전체성을 위해 표에는 키 관리 파일, 소켓 인터페이스 및 명령도 포함되어 있습니다.

Solaris 10 4/09 릴리스부터는 IPsec가 SMF에서 관리됩니다. 서비스 식별자에 대한 자세한 내용은 **Oracle Solaris 관리: 기본 관리**의 18 장, "서비스 관리(개요)"를 참조하십시오.

- 네트워크에서 IPsec 구현에 대한 지침은 475 페이지 "IPsec를 사용하여 트래픽 보호(작업 맵)"를 참조하십시오.
- IPsec 유틸리티 및 파일에 대한 자세한 내용은 21 장, "IP 보안 아키텍처(참조)"를 참조하십시오.

표 19–3 일부 IPsec 유틸리티 및 파일 목록

IPsec 유틸리티, 파일 또는 서비스	설명	매뉴얼 페이지
`svc:/network/ipsec/ipsecalgs`	현재 릴리스에서 IPsec 알고리즘을 관리하는 SMF 서비스입니다.	`ipsecalgs(1M)`
`svc:/network/ipsec/manual-key`	현재 릴리스에서 입력된 IPsec SA를 수동으로 관리하는 SMF 서비스입니다.	`ipseckey(1M)`
`svc:/network/ipsec/policy`	현재 릴리스에서, IPsec 정책을 관리하는 SMF 서비스입니다.	`smf(5)`, `ipsecconf(1M)`
`svc:/network/ipsec/ike`	현재 릴리스에서 IKE를 사용하여 IPsec SA의 자동 관리를 위한 SMF 서비스입니다.	`smf(5)`, `in.iked(1M)`
`/etc/inet/ipsecinit.conf` 파일	IPsec 정책 파일입니다. Solaris 10 4/09 이전 릴리스의 경우 이 파일이 존재하면 IPsec가 부트 시 활성화됩니다. 현재 릴리스에서 SMF `policy` 서비스에서는 이 파일을 사용하여 시스템 부트 시 IPsec 정책을 구성합니다.	`ipsecconf(1M)`
`ipsecconf` 명령	IPsec 정책 명령입니다. 현재 IPsec 정책을 보고 수정하며 테스트하는 데 유용합니다. Solaris 10 4/09 이전 릴리스에서 부트 스크립트는 `ipsecconf`를 사용하여 `/etc/inet/ipsecinit.conf` 파일을 읽고 IPsec를 활성화합니다. 현재 릴리스에서 `ipsecconf`는 SMF `policy` 서비스에서 시스템 부트 시 IPsec 정책을 구성하는 데 사용됩니다.	`ipsecconf(1M)`
`PF_KEY` 소켓 인터페이스	SADB(보안 연결 데이터베이스)에 대한 인터페이스입니다. 수동 키 관리 및 자동 키 관리를 처리합니다.	`pf_key(7P)`
`ipseckey` 명령	IPsec SA 키 입력 명령. `ipseckey`는 `PF_KEY` 인터페이스에 대한 명령줄 프론트 엔드입니다. `ipseckey`는 SA를 만들거나 삭제하거나 수정할 수 있습니다.	`ipseckey(1M)`

표 19–3 일부 IPsec 유틸리티 및 파일 목록 (계속)

IPsec 유틸리티, 파일 또는 서비스	설명	매뉴얼 페이지
/etc/inet/secret/ipseckeys 파일	수동으로 키를 입력한 SA가 포함됩니다. Solaris 10 4/09 이전 릴리스에서 ipsecinit.conf 파일이 존재하면 ipseckeys 파일이 부트 시 자동으로 읽혀집니다. 현재 릴리스에서 ipseckeys는 SMF manual-key 서비스에서 시스템 부트 시 SA를 수동으로 구성하는 데 사용됩니다.	
ipsecalgs 명령	IPsec 알고리즘 명령입니다. IPsec 알고리즘 및 해당 등록 정보 목록을 보고 수정하는 데 유용합니다. 현재 릴리스에서 SMF ipsecalgs 서비스에서 시스템 부트 시 알려진 IPsec 알고리즘을 커널과 동기화하는 데 사용됩니다.	ipsecalgs(1M)
/etc/inet/ipsecalgs 파일	구성된 IPsec 프로토콜 및 알고리즘 정의를 포함합니다. 이 파일은 ipsecalgs 명령으로 관리되며 수동으로 편집하면 안됩니다.	
/etc/inet/ike/config 파일	IKE 구성 및 정책 파일입니다. 기본적으로 이 파일은 존재하지 않습니다. Solaris 10 4/09 이전 릴리스에서 이 파일이 존재하면 IKE 데몬 in.iked가 자동 키 관리를 제공합니다. 키 관리는 /etc/inet/ike/config 파일의 규칙 및 전역 매개변수를 기준으로 합니다. 542 페이지 “IKE 유틸리티 및 파일”을 참조하십시오. 현재 릴리스에서 이 파일이 존재하면 svc:/network/ipsec/ike 서비스가 IKE 데몬 in.iked를 시작하여 자동 키 관리를 제공합니다.	ike.config(4)

Oracle Solaris 10 릴리스의 IPsec 변경 사항

새로운 Oracle Solaris 기능의 전체 목록은 **Oracle Solaris 10 1/13 새로운 기능**을 참조하십시오. Solaris 9 릴리스부터 IPsec에는 다음 기능이 포함됩니다.

- Sun Crypto Accelerator 4000 보드를 연결하면 보드가 보드의 이더넷 인터페이스를 사용하는 패킷에 대해 IPsec SA를 자동으로 캐시합니다. 또는 보드가 IPsec SA 처리를 가속화합니다.
- IPsec는 IPv6 네트워크에서 IKE를 사용하여 자동 키 관리를 활용할 수 있습니다. 자세한 내용은 22 장, “Internet Key Exchange(개요)”를 참조하십시오.

  새로운 IKE 기능은 543 페이지 “Oracle Solaris 10 릴리스의 IKE 변경 사항”을 참조하십시오.
- ipseckey 명령의 구문 분석기에서 명확한 도움말을 제공합니다. ipseckey monitor 명령은 각 이벤트에 대한 시간 기록을 작성합니다. 자세한 내용은 ipseckey(1M) 매뉴얼 페이지를 참조하십시오.

- 이제 IPsec 알고리즘은 Oracle Solaris의 암호화 프레임워크 기능인 중앙 저장소 위치에서 제공됩니다. `ipsecalgs(1M)` 매뉴얼 페이지에서는 사용 가능한 알고리즘의 특성에 대해 설명합니다. 알고리즘은 실행되는 아키텍처에 맞게 최적화됩니다. 암호화 프레임워크에 대한 자세한 내용은 **System Administration Guide: Security Services**의 13 장, "Oracle Solaris Cryptographic Framework (Overview)"를 참조하십시오.
- IPsec는 전역 영역에서 작동합니다. IPsec 정책은 비전역 영역의 경우 전역 영역에서 관리됩니다. 키 입력 자료가 만들어지고 비전역 영역에 대해 전역 영역에서 수동으로 관리됩니다. IKE는 비전역 영역에 대해 키를 생성하는 데 사용할 수 없습니다. 영역에 대한 자세한 내용은 **System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones**의 16 장, "Introduction to Solaris Zones"를 참조하십시오.
- IPsec 정책은 SCTP(Streams Control Transmission Protocol) 및 SCTP 포트와 작동할 수 있습니다. 하지만 구현은 완전하지 않습니다. RFC 3554에 지정된 SCTP에 대한 IPsec 확장 기능은 아직 구현되지 않았습니다. 이러한 제한 사항으로 인해 SCTP에 대한 IPsec 정책을 만들 때 복잡해질 수 있습니다. 자세한 내용은 RFC를 참조하십시오. 또한 471 페이지 "IPsec 및 SCTP" 및 37 페이지 "SCTP 프로토콜"을 참조하십시오.
- IPsec 및 IKE는 NAT 제품 뒤에서 발생하는 트래픽을 보호할 수 있습니다. 자세한 내용 및 제한 사항은 470 페이지 "IPsec 및 NAT 순회"를 참조하십시오. 절차는 573 페이지 "모바일 시스템에 대한 IKE 구성(작업 맵)"을 참조하십시오.

◆ ◆ ◆ 20 장

20 IPsec 구성(작업)

이 장에서는 네트워크에서 IPsec를 구현하기 위한 절차를 설명합니다. 관련 절차는 다음 작업 맵에서 설명합니다.

- 475 페이지 “IPsec를 사용하여 트래픽 보호(작업 맵)”
- 496 페이지 “IPsec를 사용하여 VPN 보호(작업 맵)”

IPsec에 대한 개요 정보는 19 장, “IP 보안 아키텍처(개요)”를 참조하십시오. IPsec에 대한 참조 정보는 21 장, “IP 보안 아키텍처(참조)”를 참조하십시오.

IPsec를 사용하여 트래픽 보호(작업 맵)

다음 작업 맵에서는 하나 이상의 시스템 사이에 IPsec를 설정하는 절차를 안내합니다. `ipsecconf`(1M), `ipseckey`(1M) 및 `ifconfig`(1M) 매뉴얼 페이지에서도 해당 예 절에서 유용한 절차를 설명합니다.

작업	설명	수행 방법
두 시스템 사이의 트래픽을 보호합니다.	한 시스템에서 다른 시스템으로의 패킷을 보호합니다.	477 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”
IPsec 정책을 사용하여 웹 서버를 보호합니다.	비웹 트래픽에서 IPsec를 사용하도록 합니다. 웹 클라이언트는 IPsec 검사를 우회하는 특정 포트로 식별됩니다.	480 페이지 “IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법”
IPsec 정책을 표시합니다.	현재 적용 중인 IPsec 정책을 해당 정책이 적용되는 순서대로 표시합니다.	483 페이지 “IPsec 정책을 표시하는 방법”
난수를 생성합니다.	수동으로 만든 보안 연관에 대한 키 입력 자료의 난수를 생성합니다.	484 페이지 “Oracle Solaris 시스템에서 난수를 생성하는 방법” **System Administration Guide: Security Services**의 “How to Generate a Symmetric Key by Using the pktool Command”

작업	설명	수행 방법
보안 연결을 수동으로 만들거나 바꿉니다.	보안 연결을 위한 원시 데이터를 제공합니다. ■ IPsec 알고리즘 이름 및 키 입력 자료 ■ SPI(보안 매개변수 색인) ■ IP 소스 및 대상 주소와 기타 매개변수	485 페이지 “수동으로 IPsec 보안 연관을 만드는 방법”
IPsec가 패킷을 보호하고 있는지 확인합니다.	IP 데이터그램이 어떻게 보호되는지 나타내는 특정 헤더에 대한 snoop 출력을 검사합니다.	490 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”
(선택 사항) 네트워크 보안 역할을 만듭니다.	보안 네트워크를 설정할 수 있지만 수퍼 유저 역할보다 권한이 적은 역할을 만듭니다.	491 페이지 “네트워크 보안에 대한 역할을 구성하는 방법”
IPsec 및 키 입력 자료를 SMF 서비스의 일부로 관리합니다.	서비스를 사용으로 설정, 사용 안함으로 설정, 새로 고침 및 다시 시작하는 명령을 언제, 어떻게 사용하는지 설명합니다. 또한 서비스의 등록 정보 값을 변경하는 명령을 설명합니다.	492 페이지 “IKE 및 IPsec 서비스를 관리하는 방법”
보안 VPN(virtual private network)을 설정합니다.	인터넷을 거치는 두 시스템 사이에 IPsec를 설정합니다.	496 페이지 “IPsec를 사용하여 VPN 보호(작업 맵)”

IPsec를 사용하여 트래픽 보호

이 절에서는 두 시스템 간의 트래픽을 보호하고 웹 서버의 보안을 유지할 수 있는 절차를 제공합니다. VPN을 보호하려면 496 페이지 “IPsec를 사용하여 VPN 보호(작업 맵)”를 참조하십시오. 추가 절차는 키 입력 자료 및 보안 연관을 제공하고 IPsec가 구성한 대로 작동 중인지 확인합니다.

다음 정보는 모든 IPsec 구성 작업에 적용됩니다.

- **IPsec 및 영역** – 공유 IP 비전역 영역에 대한 IPsec 정책 및 키를 관리하려면 전역 영역에서 IPsec 정책 파일을 만들고 전역 영역에서 IPsec 구성 명령을 실행합니다. 구성 중인 비보안 영역에 해당하는 소스 주소를 사용합니다. 또한 전역 영역에서 전역 영역의 IPsec 정책 및 키를 구성할 수 있습니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다. Solaris 10 7/07 릴리스부터 IKE를 사용하여 비전역 영역에서 키를 관리할 수 있습니다.
- **IPsec 및 RBAC** – 역할을 사용하여 IPsec를 관리하려면 **System Administration Guide: Security Services**의 9 장, “Using Role-Based Access Control (Tasks)”을 참조하십시오. 예는 491 페이지 “네트워크 보안에 대한 역할을 구성하는 방법”을 참조하십시오.
- **IPsec 및 SCTP** – IPsec는 SCTP(Streams Control Transmission Protocol) 연결을 보호하는 데 사용할 수 있지만 주의해야 합니다. 자세한 내용은 471 페이지 “IPsec 및 SCTP”를 참조하십시오.

▼ IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법

이 절차에서는 다음 설정을 가정합니다.

- 두 시스템의 이름은 `enigma` 및 `partym`입니다.
- 각 시스템에는 두 주소인 IPv4 주소와 IPv6 주소가 있습니다.
- 각 시스템에는 AES 알고리즘을 사용하는 ESP 암호화(128비트의 키 필요) 및 SHA1 메시지 다이제스트를 사용하는 ESP 인증(160비트의 키 필요)이 필요합니다.
- 각 시스템은 공유 보안 연결을 사용합니다.

  공유 SA를 사용하여 두 시스템을 보호하는 데 한 쌍의 SA만 필요합니다.

시작하기 전에 시스템 또는 공유 IP 영역에 대한 IPsec 정책을 구성하려면 전역 영역에 있어야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오. 예는 예 20–1를 참조하십시오.

---

2 각 시스템에서 호스트 항목을 확인합니다.

현재 릴리스에서 `/etc/inet/hosts` 파일에 호스트 항목을 추가합니다.

Solaris 10 7/07 이전 릴리스를 실행하는 시스템에서는 IPv4 및 IPv6 항목을 `/etc/inet/ipnodes` 파일에 추가합니다. 한 시스템에 대한 항목은 이 파일에서 모두 같이 있어야 합니다. 시스템 구성 파일에 대한 자세한 내용은 217 페이지 “TCP/IP 구성 파일” 및 11 장, “IPv6 세부 개요(참조)”를 참조하십시오.

IPv4 주소만 사용하여 시스템을 연결 중인 경우 `/etc/inet/hosts` 파일을 수정합니다. 이 예에서 연결 시스템은 이전 Solaris 릴리스를 실행 중이며 IPv6 주소를 사용 중입니다.

a. 이름이 `enigma`인 시스템에서 `hosts` 또는 `ipnodes` 파일에 다음을 입력합니다.

```
# Secure communication with partym
192.168.13.213 partym
2001::eeee:3333:3333 partym
```

b. 이름이 partym인 시스템에서 hosts 또는 ipnodes 파일에 다음을 입력합니다.

```
# Secure communication with enigma
192.168.116.16 enigma
2001::aaaa:6666:6666 enigma
```

심볼릭 이름에 대한 이름 지정 서비스를 사용하는 경우 안전하지 않습니다.

3 각 시스템에서 IPsec 정책 파일을 만듭니다.

파일 이름은 `/etc/inet/ipsecinit.conf`입니다. 예는 `/etc/inet/ipsecinit.sample` 파일을 참조하십시오.

4 IPsec 정책 항목을 ipsecinit.conf 파일에 추가합니다.

a. enigma 시스템에서 다음 정책을 추가합니다.

```
{laddr enigma raddr partym} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. partym 시스템에서 동일한 정책을 추가합니다.

```
{laddr partym raddr enigma} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

IPsec 정책 항목의 구문은 `ipsecconf(1M)` 매뉴얼 페이지를 참조하십시오.

5 각 시스템에서 두 시스템 사이에 IPsec SA 쌍을 추가합니다.

자동으로 SA를 만들도록 IKE(Internet Key Exchange)를 구성할 수 있습니다. 또한 수동으로 SA를 추가할 수 있습니다.

주 – 수동으로 키를 생성하고 유지 관리해야 하는 합당한 이유가 없는 경우 IKE를 사용해야 합니다. IKE 키 관리는 수동 키 관리보다 안전합니다.

- 545 페이지 “IKE 구성(작업 맵)”의 구성 절차 중 하나에 따라 IKE를 구성합니다. IKE 구성 파일의 구문은 `ike.config(4)` 매뉴얼 페이지를 참조하십시오.
- 수동으로 SA를 추가하려면 485 페이지 “수동으로 IPsec 보안 연관을 만드는 방법”을 참조하십시오.

6 IPsec 정책을 사용으로 설정합니다.

- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우에는 시스템을 재부트합니다.**

  ```
  # init 6
  ```

  그런 다음 490 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”으로 이동합니다.

- **Solaris 10 4/09 릴리스부터 IPsec 서비스를 새로 고치고 키 관리 서비스를 사용으로 설정합니다.**

  단계 7에서 단계 10까지의 단계를 완료합니다.

7 **IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

오류를 수정하고 파일의 구문을 확인한 다음 계속합니다.

8 **IPsec 정책을 새로 고칩니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

IPsec 정책은 기본적으로 사용으로 설정되므로 **새로 고칩니다**. IPsec 정책을 사용 안함으로 설정한 경우 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/policy:default
```

9 **IPsec에 대한 키를 활성화합니다.**

- **단계 5에서 IKE를 구성한 경우 다음 중 하나를 수행합니다.**
  - **ike 서비스가 사용으로 설정되지 않은 경우 사용으로 설정합니다.**

    ```
    # svcadm enable svc:/network/ipsec/ike:default
    ```

  - **ike 서비스가 사용으로 설정된 경우 다시 시작합니다.**

    ```
    # svcadm restart svc:/network/ipsec/ike:default
    ```

- **단계 5에서 수동으로 키를 구성한 경우 다음 중 하나를 수행합니다.**
  - **manual-key 서비스가 사용으로 설정되지 않은 경우 사용으로 설정합니다.**

    ```
    # svcadm enable svc:/network/ipsec/manual-key:default
    ```

  - **manual-key 서비스가 사용으로 설정된 경우 새로 고칩니다.**

    ```
    # svcadm refresh svc:/network/ipsec/manual-key:default
    ```

10 **패킷이 보호되고 있는지 확인합니다.**

절차는 490 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”을 참조하십시오.

예 20–1 ssh 연결을 사용할 때 IPsec 정책 추가

이 예에서 수퍼 유저인 관리자는 ssh 명령을 사용하여 두번째 시스템에 접근한 다음 두 시스템에서 IPsec 정책 및 키를 구성합니다. 자세한 내용은 ssh(1) 매뉴얼 페이지를 참조하십시오.

- 먼저 관리자는 위 절차의 단계 2 ~ 단계 5를 수행하여 첫번째 시스템을 구성합니다.
- 그런 다음 다른 터미널 창에서 관리자는 ssh 명령을 사용하여 두번째 시스템에 로그인합니다.

```
local-system # ssh other-system
other-system #
```

- ssh 세션의 터미널 창에서 관리자는 단계 2 ~ 단계 6을 완료하여 두번째 시스템의 IPsec 정책 및 키를 구성합니다.
- 그런 다음 관리자는 ssh 세션을 종료합니다.

```
other-system # exit
local-system #
```

- 마지막으로 관리자는 단계 6을 완료하여 첫번째 시스템에서 IPsec 정책을 사용으로 설정합니다.

ssh 연결 사용을 포함하여 다음에 두 시스템이 통신할 때 통신이 IPsec로 보호됩니다.

예 20–2 재부트 없이 IPsec를 사용하여 트래픽 보안

Solaris 10 4/09 이전 릴리스를 실행 중인 경우 다음 예가 유용합니다. 즉, 해당 릴리스에서 IPsec는 서비스로 관리되지 않습니다. 이 예에서는 테스트 환경에서 IPsec를 구현하는 방법을 설명합니다. 운용 환경에서는 ipsecconf 명령을 실행하는 것보다 재부트하는 것이 더 안전합니다. 보안 고려 사항은 이 예의 끝부분을 참조하십시오.

단계 6에서 재부트하는 대신 다음 옵션 중 하나를 선택합니다.

- IKE를 사용하여 키 입력 자료를 만든 경우 중지한 후 in.iked 데몬을 다시 시작합니다.

```
# pkill in.iked
# /usr/lib/inet/in.iked
```

- 수동으로 키를 추가한 경우 ipseckey 명령을 사용하여 데이터베이스에 SA를 추가합니다.

```
# ipseckey -c -f /etc/inet/secret/ipseckeys
```

그런 다음 ipsecconf 명령을 사용하여 IPsec 정책을 활성화합니다.

```
# ipsecconf -a /etc/inet/ipsecinit.conf
```

보안 고려 사항 – ipsecconf 명령을 실행할 때 경고를 읽으십시오. 이미 잠긴 소켓, 즉 이미 사용 중인 소켓은 시스템에 대한 비보안 백 도어를 제공합니다. 자세한 내용은 531 페이지 “ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항”을 참조하십시오.

▼ IPsec를 사용하여 비웹 트래픽에서 웹 서버를 보호하는 방법

보안 웹 서버를 통해 웹 클라이언트가 웹 서비스와 통신할 수 있습니다. 보안 웹 서버에서 웹 트래픽이 아닌 트래픽은 보안 검사를 **통과해야** 합니다. 다음 절차에는 웹 트래픽에 대한 우회가 포함됩니다. 또한 이 웹 서버는 비보안 DNS 클라이언트 요청을 할 수 있습니다. 기타 모든 트래픽에는 AES 및 SHA-1 알고리즘을 사용하는 ESP가 필요합니다.

시작하기 전에 IPsec 정책을 구성하려면 전역 영역에 있어야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다.

477 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”을
완료했으므로 다음 조건이 적용됩니다.

- 두 시스템 사이의 통신이 IPsec로 보호됩니다.
- 키 입력 자료가 수동으로 또는 IKE에 의해 생성됩니다.
- 패킷이 보호되고 있는지 확인했습니다.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 보안 정책 검사를 우회해야 하는 서비스를 결정합니다.

웹 서버의 경우 이러한 서비스에는 TCP 포트 80(HTTP) 및 443(보안 HTTP)이 포함됩니다. 웹 서버에서 DNS 이름 조회를 제공하는 경우 TCP 및 UDP 모두에 대해 포트 53이 서버에 포함되어야 할 수도 있습니다.

3 웹 서버에 대한 IPsec 정책을 만들고 사용으로 설정합니다.

- **Solaris 10 4/09 릴리스부터 단계 4에서 단계 7까지의 단계를 수행합니다.**
- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우 단계 8에서 단계 11까지의 단계를 수행합니다.**

단계 12는 모든 Oracle Solaris 릴리스에서 옵션입니다.

4 웹 서버 정책을 IPsec 정책 파일에 추가합니다.

다음 행을 `/etc/inet/ipsecinit.conf` 파일에 추가합니다.

```
# Web traffic that web server should bypass.
{lport  80 ulp tcp dir both} bypass {}
{lport 443 ulp tcp dir both} bypass {}

# Outbound DNS lookups should also be bypassed.
{rport 53 dir both} bypass {}

# Require all other traffic to use ESP with AES and SHA-1.
# Use a unique SA for outbound traffic from the port
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

이 구성은 단계 4에서 설명한 우회 예외 사항과 함께 보안 트래픽만 시스템에 액세스할 수 있도록 허용합니다.

5 **IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 **IPsec 정책을 새로 고칩니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

7 **IPsec에 대한 키를 새로 고칩니다.**

- **477 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법"의 단계 5에서 IKE를 구성한 경우 ike 서비스를 다시 시작합니다.**

  ```
  # svcadm restart svc:/network/ipsec/ike
  ```

- **477 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법"의 단계 5에서 키를 수동으로 구성한 경우 manual-key 서비스를 새로 고칩니다.**

  ```
  # svcadm refresh svc:/network/ipsec/manual-key:default
  ```

설정이 완료되었습니다. 선택적으로 단계 12를 수행할 수 있습니다.

8 **/etc/inet 디렉토리에서 웹 서버 정책에 대한 파일을 만듭니다.**

주 – 다음 단계에서는 Solaris 10 4/09 이전 릴리스를 실행 중인 웹 서버를 구성합니다.

파일에 해당 용도를 나타내는 이름을 지정합니다(예: `IPsecWebInitFile`). 이 파일에 다음 라인을 입력합니다.

```
# Web traffic that web server should bypass.
{lport  80 ulp tcp dir both} bypass {}
{lport 443 ulp tcp dir both} bypass {}

# Outbound DNS lookups should also be bypassed.
{rport 53 dir both} bypass {}

# Require all other traffic to use ESP with AES and SHA-1.
# Use a unique SA for outbound traffic from the port
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

이 구성은 단계 4에서 설명한 우회 예외 사항과 함께 보안 트래픽만 시스템에 액세스할 수 있도록 허용합니다.

9 **단계 8에서 만든 파일의 내용을 /etc/inet/ipsecinit.conf 파일에 복사합니다.**

10 **읽기 전용 권한으로 IPsecWebInitFile 파일을 보호합니다.**

```
# chmod 400 IPsecWebInitFile
```

11 **재부트 없이 웹 서버의 보안을 유지합니다.**

다음 옵션 중 하나를 선택합니다.

- 키 관리를 위해 IKE를 사용 중인 경우 `in.iked` 데몬을 중지한 후 다시 시작합니다.

```
# pkill in.iked
# /usr/lib/inet/in.iked
```

- 수동으로 키를 관리 중인 경우 ipseckey 및 ipsecconf 명령을 사용합니다.

  IPsecWebInitFile을 ipsecconf 명령에 대한 인수로 사용합니다. ipsecinit.conf 파일을 인수로 사용하는 경우 파일의 정책이 이미 시스템에 구현된 상태이면 ipsecconf 명령은 오류를 발생시킵니다.

```
# ipseckey -c -f /etc/inet/secret/ipseckeys
# ipsecconf -a /etc/inet/IPsecWebInitFile
```

주의 – ipsecconf 명령을 실행할 때 경고를 읽으십시오. 이미 잠긴 소켓, 즉 이미 사용 중인 소켓은 시스템에 대한 비보안 백 도어를 제공합니다. 자세한 내용은 531 페이지 “ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항”을 참조하십시오. in.iked 데몬을 다시 시작하는 경우에도 동일한 경고가 적용됩니다.

또한 재부트할 수 있습니다. 재부트하면 모든 TCP 연결에 IPsec 정책이 적용됩니다. 재부트 시 TCP 연결이 IPsec 정책 파일의 정책을 사용합니다.

12 (옵션) 원격 시스템이 비웹 트래픽에 대해 웹 서버와 통신할 수 있도록 설정합니다.

다음 정책을 원격 시스템의 ipsecinit.conf 파일에 입력합니다.

```
# Communicate with web server about nonweb stuff
#
{laddr webserver} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

원격 시스템은 시스템의 IPsec 정책이 일치할 경우에만 비웹 트래픽에 대해 웹 서버와 안전하게 통신할 수 있습니다.

▼ IPsec 정책을 표시하는 방법

ipsecconf 명령을 인수 없이 실행하면 시스템에서 구성된 정책을 볼 수 있습니다.

시작하기 전에 ipsecconf 명령은 전역 영역에서 실행해야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 ipsecconf 명령을 실행합니다.

1 네트워크 IPsec 관리 프로파일이 포함된 역할 또는 수퍼 유저로 로그인합니다.

Solaris 10 4/09 이전 릴리스를 실행 중인 경우 네트워크 IPsec 관리 프로파일을 사용할 수 없습니다. 네트워크 보안 프로파일을 사용합니다.

네트워크 보안 프로파일을 포함하는 역할을 만들고 해당 역할을 사용자에게 지정하려면 491 페이지 “네트워크 보안에 대한 역할을 구성하는 방법”을 참조하십시오.

2 IPsec 정책을 표시합니다.

a. 항목이 추가된 순서대로 전역 IPsec 정책 항목을 표시합니다.

```
$ ipsecconf
```

명령은 **색인** 다음에 번호와 함께 각 항목을 표시합니다.

b. 일치하는 순서대로 IPsec 정책 항목을 표시합니다.

```
$ ipsecconf -l -n
```

c. 터널별 항목을 포함하여 일치하는 순서대로 IPsec 정책 항목을 표시합니다.

```
$ ipsecconf -L -n
```

▼ Oracle Solaris 시스템에서 난수를 생성하는 방법

수동으로 키를 지정하는 경우 키 입력 자료는 임의여야 합니다. IPsec 키에 대한 키 입력 자료 형식은 16진수입니다. 다른 운영 체제에서는 ASCII 키 입력 자료가 필요할 수 있습니다. ASCII가 필요한 운영 체제와 통신 중인 Oracle Solaris 시스템에 대한 키 입력 자료를 생성하려면 예 23–1을 참조하십시오.

사이트에 난수 생성기가 있는 경우 생성기를 사용합니다. 또는 `od` 명령을 `/dev/random` 장치와 함께 입력으로 사용할 수 있습니다. 자세한 내용은 od(1) 매뉴얼 페이지를 참조하십시오.

Solaris 10 4/09 릴리스에서 `pktool` 명령을 사용할 수도 있습니다. 이 명령 구문은 `od` 명령 구문보다 간단합니다. 자세한 내용은 **System Administration Guide: Security Services**의 "How to Generate a Symmetric Key by Using the pktool Command"을 참조하십시오.

1 16진수 형식의 난수를 생성합니다.

```
% od -x|-X -A n file | head -n
```

`-x` 8진수 덤프를 16진수 형식으로 표시합니다. 16진수 형식은 키 입력 자료에 유용합니다. 16진수가 4자 청크로 인쇄됩니다.

`-X` 8진수 덤프를 16진수 형식으로 표시합니다. 16진수가 8자 청크로 인쇄됩니다.

`-A n` 표시에서 입력 오프셋 기준을 제거합니다.

file 난수의 소스로 사용됩니다.

`head -`*n* 출력의 처음 *n*개 라인만 표시되도록 제한합니다.

2 출력을 결합하여 적합한 길이의 키를 만듭니다.

한 라인의 숫자 간 공백을 제거하여 32자 키를 만듭니다. 32자 키는 128비트입니다. SPI(보안 매개변수 색인)의 경우 8자 키를 사용해야 합니다. 키는 `0x` 접두어를 사용해야 합니다.

예 20–3 IPsec에 대한 키 자료 생성

다음 예에서는 각각 8개의 16진수 문자 그룹으로 구성된 키를 두 라인으로 표시합니다.

```
% od -X -A n /dev/random | head -2
        d54d1536 4a3e0352 0faf93bd 24fd6cad
        8ecc2670 f3447465 20db0b0c c83f5a4b
```

첫 라인에서 숫자 4개를 결합하여 32자 키를 만들 수 있습니다. `0x`로 시작되는 숫자 8자는 적합한 SPI 값을 제공합니다(예: `0xf3447465`).

다음 예에서는 각각 4개의 16진수 문자 그룹으로 구성된 키를 두 라인으로 표시합니다.

```
% od -x -A n /dev/random | head -2
        34ce 56b2 8b1b 3677 9231 42e9 80b0 c673
        2f74 2817 8026 df68 12f4 905a db3d ef27
```

첫 라인에서 숫자 8개를 결합하여 32자 키를 만들 수 있습니다.

▼ 수동으로 IPsec 보안 연관을 만드는 방법

다음 절차에서는 477 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법" 절차에 필요한 키 입력 자료를 제공합니다. `partym` 및 `enigma`의 두 시스템에 대한 키를 생성합니다. 한 시스템에서 키를 생성한 다음 첫번째 시스템의 키를 두 시스템에서 모두 사용합니다.

시작하기 전에 공유 IP 영역에 대한 키 입력 자료를 수동으로 관리하려면 전역 영역에 있어야 합니다.

1 SA에 대한 키 입력 자료를 생성합니다.

아우바운드 트래픽에 대한 3개의 16진수 난수 및 인바운드 트래픽에 대한 3개의 16진수 난수가 필요합니다.

따라서 한 시스템에서 다음 숫자를 생성해야 합니다.

- `spi` 키워드에 대한 값으로 2개의 16진수 임의 숫자. 하나는 아웃바운드 트래픽용입니다. 다른 하나는 인바운드 트래픽용입니다. 각 숫자 모두 최대 8자까지만 허용됩니다.
- 인증의 SHA1 알고리즘에 대한 2개의 16진수 난수. 160비트 키의 경우 각 숫자의 길이는 40자여야 합니다. 하나는 `dst enigma`용입니다. 다른 하나는 `dst partym`용입니다.
- ESP 암호화의 AES 알고리즘에 대한 2개의 16진수 난수. 256비트 키의 경우 각 숫자의 길이는 64자여야 합니다. 하나는 `dst enigma`용입니다. 다른 하나는 `dst partym`용입니다.

사이트에 임의 숫자 생성기가 있을 경우 생성기를 사용하십시오. 또한 `od` 명령을 사용할 수 있습니다. 절차는 484 페이지 "Oracle Solaris 시스템에서 난수를 생성하는 방법"을 참조하십시오.

2 **시스템 중 한 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

3 **SA를 만듭니다.**

- **Solaris 10 4/09 릴리스부터 단계 8에서 단계 10까지의 단계를 수행합니다.**
- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우 단계 4에서 단계 9까지의 단계를 수행합니다.**

4 **`ipseckey` 명령 모드를 사용으로 설정합니다.**

```
# ipseckey

>
```

> 프롬프트는 `ipseckey` 명령 모드임을 나타냅니다.

5 **기존 SA를 바꾸는 중인 경우 현재 SA를 비웁니다.**

```
> flush
>
```

공격자가 SA를 끊지 않도록 하려면 키 입력 자료를 바꿔야 합니다.

주 – 통신 시스템에서 키 바꾸기를 조정해야 합니다. 한 시스템에서 SA를 바꿀 때 원격 시스템에서도 SA를 바꿔야 합니다.

---

6 **SA를 만들려면 다음 명령을 입력합니다.**

```
> add protocol spi random-hex-string \
src addr dst addr2 \
protocol-prefix_alg protocol-algorithm   \
protocol-prefixkey random-hex-string-of-algorithm-specified-length
```

또한 이 구문을 사용하여 방금 비운 SA를 바꿉니다.

protocol
`esp` 또는 `ah`를 지정합니다.

random-hex-string
최대 8자의 난수를 16진수 형식으로 지정합니다. 문자 앞에 `0x` 접두어를 사용합니다. SPI(보안 매개변수 색인)가 허용하는 숫자보다 더 많이 입력하는 경우 시스템에서는 남은 숫자를 무시합니다. SPI가 허용하는 숫자보다 더 적게 입력하는 경우 시스템에서는 숫자를 더 추가합니다.

addr
한 시스템의 IP 주소를 지정합니다.

addr2
*addr*의 피어 시스템에 대한 IP 주소를 지정합니다.

protocol-prefix
`encr` 또는 `auth` 중 하나를 지정합니다. `encr` 접두어는 `esp` 프로토콜과 함께 사용됩니다. `auth` 접두어는 `ah` 프로토콜과 함께 사용되며 `esp` 프로토콜을 인증하는 데 사용됩니다.

protocol-algorithm
ESP 또는 AH에 대한 알고리즘을 지정합니다. 각 알고리즘에는 특정 길이의 키가 필요합니다.

인증 알고리즘에는 MD5 및 SHA1이 포함됩니다. Solaris 10 4/09 릴리스부터 SHA256 및 SHA512가 지원됩니다. 암호화 알고리즘에는 DES, 3DES, AES 및 Blowfish가 포함됩니다.

random-hex-string-of-algorithm-specified-length
알고리즘에 필요한 길이로 16진수의 난수를 지정합니다. 예를 들어, MD5 알고리즘에는 128비트 키의 32자 문자열이 필요합니다. 3DES 알고리즘에는 192비트 키의 48자 문자열이 필요합니다.

a. 예를 들어, `enigma` 시스템에서 아웃바운드 패킷을 보호합니다.

단계 1에서 생성한 난수를 사용합니다.

Solaris 10 1/06의 경우:

```
> add esp spi 0x8bcd1407 \
src 192.168.116.16 dst 192.168.13.213 \
encr_alg aes \
auth_alg sha1 \
encrkey c0c65b888c2ee301c84245c3da63127e92b2676105d5330e85327c1442f37d49 \
authkey 6fab07fec4f2895445500ed992ab48835b9286ff
>
```

주 – 피어 시스템은 동일한 키 입력 자료 및 동일한 SPI를 사용해야 합니다.

b. `enigma` 시스템의 `ipseckey` 명령 모드에서는 계속 인바운드 패킷을 보호합니다.

다음 명령을 입력하여 패킷을 보호합니다.

```
> add esp spi 0x122a43e4 \
src 192.168.13.213 dst 192.168.116.16 \
```

```
encr_alg aes \
auth_alg sha1 \
encrkey a2ea934cd62ca7fa14907cb2ad189b68e4d18c976c14f22b30829e4b1ea4d2ae \
authkey c80984bc4733cc0b7c228b9b74b988d2b7467745
>
```

주 – 각 SA에 대해 키 및 SPI가 서로 다를 수 있습니다. 각 SA에 대해 서로 다른 키 및 서로 다른 SPI를 지정**해야** 합니다.

7 ipseckey 명령 모드를 종료하려면 Control-D를 누르거나 quit를 입력합니다.

8 /etc/inet/secret/ipseckeys 파일에 키 입력 자료를 추가합니다.

Solaris 10 4/09 이전 릴리스에서 이 단계는 재부트 시 IPsec에서 키 입력 자료를 사용할 수 있도록 합니다.

/etc/inet/secret/ipseckeys 파일의 라인은 ipseckey 명령줄 언어와 동일합니다.

a. 예를 들어, enigma 시스템의 /etc/inet/secret/ipseckeys 파일은 다음과 유사하게 표시됩니다.

```
# ipseckeys - This file takes the file format documented in
#   ipseckey(1m).
#   Note that naming services might not be available when this file
#   loads, just like ipsecinit.conf.
#
# for outbound packets on enigma
add esp spi 0x8bcd1407 \
   src 192.168.116.16 dst 192.168.13.213  \
   encr_alg aes \
   auth_alg sha1  \
   encrkey  c0c65b888c2ee301c84245c3da63127e92b2676105d5330e85327c1442f37d49 \
   authkey  6fab07fec4f2895445500ed992ab48835b9286ff
#
# for inbound packets
add esp spi 0x122a43e4 \
   src 192.168.13.213 dst 192.168.116.16 \
   encr_alg aes \
   auth_alg sha1  \
   encrkey a2ea934cd62ca7fa14907cb2ad189b68e4d18c976c14f22b30829e4b1ea4d2ae \
   authkey c80984bc4733cc0b7c228b9b74b988d2b7467745
```

b. 읽기 전용 권한으로 파일을 보호합니다.

```
# chmod 400 /etc/inet/secret/ipseckeys
```

9 partym 시스템에서 이 절차를 반복합니다.

enigma에서 사용된 동일한 키 입력 자료를 사용합니다.

두 시스템의 키 입력 자료는 **동일해야** 합니다. 다음 예와 같이 ipseckeys 파일의 설명만 다릅니다. dst enigma이 enigma 시스템에서 인바운드되고 partym 시스템에서 아웃바운드되므로 설명이 다릅니다.

```
# partym ipseckeys file
#
```

```
# for inbound packets
add esp spi 0x8bcd1407 \
   src 192.168.116.16 dst 192.168.13.213  \
   encr_alg aes \
   auth_alg sha1  \
   encrkey  c0c65b888c2ee301c84245c3da63127e92b2676105d5330e85327c1442f37d49 \
   authkey  6fab07fec4f2895445500ed992ab48835b9286ff
#
# for outbound packets
add esp spi 0x122a43e4 \
   src 192.168.13.213 dst 192.168.116.16 \
   encr_alg aes \
   auth_alg sha1  \
   encrkey a2ea934cd62ca7fa14907cb2ad189b68e4d18c976c14f22b30829e4b1ea4d2ae \
   authkey c80984bc4733cc0b7c228b9b74b988d2b7467745
```

10 **manual-key 서비스를 사용으로 설정합니다.**

```
# svcadm enable svc:/network/ipsec/manual-key
```

현재 릴리스에서 키를 바꾸려면 예 20–4를 참조하십시오.

예 20–4 IPsec SA 바꾸기

이 예에서 관리자는 현재 Oracle Solaris 10 릴리스를 실행 중인 시스템을 구성하고 있습니다. 관리자는 새 키를 생성하고 ipseckeys 파일에서 키 입력 정보를 변경한 다음 서비스를 다시 시작합니다.

- 먼저 관리자는 484 페이지 “Oracle Solaris 시스템에서 난수를 생성하는 방법”을 완료하여 키를 생성합니다.
- 그런 다음 관리자는 /etc/inet/secret/ipseckeys 파일에서 생성된 키를 사용합니다.

  관리자는 동일한 알고리즘을 사용했습니다. 따라서 관리자는 SPI, encrkey 및 authkey 값만 변경하면 됩니다.

  ```
  add esp spi 0x8xzy1492 \
     src 192.168.116.16 dst 192.168.13.213  \
     encr_alg aes \
     auth_alg sha1  \
     encrkey 0a1f3886b06ebd7d39f6f89e4c29c93f2741c6fa598a38af969907a29ab1b42a \
     authkey a7230aabf513f35785da73e33b064608be41f69a
  #
  # add esp spi 0x177xce34\
     src 192.168.13.213 dst 192.168.116.16 \
     encr_alg aes \
     auth_alg sha1  \
     encrkey 4ef5be40bf93498017b2151d788bb37e372f091add9b11149fba42435fefe328 \
     authkey 0e1875d9ff8e42ab652766a5cad49f38c9152821
  ```

- 마지막으로 관리자는 manual-key 서비스를 다시 시작합니다. 다시 시작 명령은 새 키를 추가하기 전에 기존 키를 비웁니다.

  ```
  # svcadm restart manual-key
  ```

▼ IPsec로 패킷이 보호되는지 확인하는 방법

패킷이 보호되는지 확인하려면 snoop 명령을 사용하여 연결을 테스트합니다. 다음 접두어가 snoop 출력에 나타날 수 있습니다.

- AH: 접두어는 AH가 헤더를 보호하고 있음을 나타냅니다. auth_alg를 사용하여 트래픽을 보호하는 경우 AH:를 보게 됩니다.
- ESP: 접두어는 암호화된 데이터가 보내지고 있음을 나타냅니다. encr_auth_alg 또는 encr_alg를 사용하여 트래픽을 보호하는 경우 ESP:를 보게 됩니다.

시작하기 전에 snoop 출력을 만들려면 수퍼 유저 또는 동등한 역할로 로그인해야 합니다. 연결을 테스트하려면 두 시스템에 대한 액세스 권한이 있어야 합니다.

1 partym와 같은 한 시스템에서 수퍼 유저로 로그인합니다.

```
% su -
Password:       Type root password
#
```

2 partym 시스템에서 원격 시스템으로부터 패킷 스누핑을 준비합니다.

partym의 터미널 창에서 enigma 시스템으로부터 패킷을 스누핑합니다.

```
# snoop -d hme0 -v enigma
Using device /dev/hme (promiscuous mode)
```

3 원격 시스템에서 패킷을 보냅니다.

다른 터미널 창에서 enigma 시스템에 원격으로 로그인합니다. 암호를 제공합니다. 그런 다음 수퍼 유저로 로그인하고 enigma 시스템에서 partym 시스템으로 패킷을 보냅니다. 패킷은 snoop -v enigma 명령으로 캡처해야 합니다.

```
% ssh enigma
Password:       Type your password
% su -
Password:       Type root password
# ping partym
```

4 snoop 출력을 검사합니다.

partym 시스템에서 초기 IP 헤더 정보 이후 AH 및 ESP 정보가 포함된 출력을 볼 수 있어야 합니다. 다음과 유사한 AH 및 ESP 정보는 패킷이 보호되고 있음을 나타냅니다.

```
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 51 (AH)
IP:   Header checksum = 4e0e
IP:   Source address = 192.168.116.16, enigma
IP:   Destination address = 192.168.13.213, partym
IP:   No options
IP:
AH:  ----- Authentication Header -----
AH:
AH:  Next header = 50 (ESP)
```

```
AH:  AH length = 4 (24 bytes)
AH:  <Reserved field = 0x0>
AH:  SPI = 0xb3a8d714
AH:  Replay = 52
AH:  ICV = c653901433ef5a7d77c76eaa
AH:
ESP:  ----- Encapsulating Security Payload -----
ESP:
ESP:  SPI = 0xd4f40a61
ESP:  Replay = 52
ESP:     ....ENCRYPTED DATA....

ETHER:  ----- Ether Header -----
...
```

▼ 네트워크 보안에 대한 역할을 구성하는 방법

RBAC(역할 기반 액세스 제어)를 사용하여 시스템을 관리 중인 경우 이 절차에 따라 네트워크 관리 역할 또는 네트워크 보안 역할을 제공합니다.

1 로컬 `prof_attr` 데이터베이스에서 Network 권한 프로파일을 찾습니다.

현재 릴리스에서 출력은 다음과 같이 표시됩니다.

```
% cd /etc/security
% grep Network prof_attr
Network IPsec Management:::Manage IPsec and IKE...
Network Link Security:::Manage network link security...
Network Management:::Manage the host and network configuration...
Network Security:::Manage network and host security...
Network Wifi Management:::Manage wifi network configuration...
Network Wifi Security:::Manage wifi network security...
```

Solaris 10 4/09 이전 릴리스를 실행 중인 경우 출력은 다음과 같이 표시됩니다.

```
% cd /etc/security
% grep Network prof_attr
Network Management:::Manage the host and network configuration
Network Security:::Manage network and host security
System Administrator::: Network Management
```

Network Management 프로파일은 System Administrator 프로파일의 보조 프로파일입니다. 역할에 System Administrator 권한 프로파일을 포함시킨 경우 해당 역할은 Network Management 프로파일의 명령을 실행할 수 있습니다.

2 Network Management 권한 프로파일의 명령을 결정합니다.

```
% grep "Network Management" /etc/security/exec_attr
Network Management:solaris:cmd:::/usr/sbin/ifconfig:privs=sys_net_config
...
Network Management:suser:cmd:::/usr/sbin/snoop:uid=0
```

`solaris` 정책 명령은 권한(`privs=sys_net_config`)으로 실행됩니다. `suser` 정책 명령은 수퍼 유저(`uid=0`)로 실행됩니다.

3 **사이트에서 네트워크 보안 역할의 범위를 결정합니다.**

단계 1의 권한 프로파일 정의를 사용하여 결정합니다.

- **모든 네트워크 보안을 처리하는 역할을 만들려면 Network Security 권한 프로파일을 사용합니다.**
- **현재 릴리스에서 IPsec 및 IKE만 처리하는 역할을 만들려면 Network IPsec Management 권한 프로파일을 사용합니다.**

4 **Network Management 권한 프로파일을 포함하는 네트워크 보안 역할을 만듭니다.**

Network Management 권한 프로파일과 함께 Network Security 또는 Network IPsec Management 권한 프로파일을 가진 역할은 대표적으로 해당 권한으로 `ifconfig`, `snoop`, `ipsecconf` 및 `ipseckey` 명령을 실행할 수 있습니다.

역할을 만들고, 사용자에게 역할을 지정하고, 이름 서비스에 변경 사항을 등록하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

예 20–5 역할 간 네트워크 보안 책임 구분

이 예에서는 관리자가 두 역할 간에 네트워크 보안 책임을 구분합니다. 한 역할은 Wifi 및 링크 보안을 관리하고, 다른 역할은 IPsec 및 IKE를 관리합니다. 각 역할은 교대당 한 사람씩 세 명의 사용자에게 지정됩니다.

역할은 관리자가 다음과 같이 만듭니다.

- 관리자는 첫번째 역할 이름을 LinkWifi로 지정합니다.
  - 관리자는 Network Wifi, Network Link Security 및 Network Management 권한 프로파일을 역할에 지정합니다.
  - 그런 다음 관리자는 LinkWifi 역할을 해당 사용자에게 지정합니다.
- 관리자는 두번째 역할 이름을 IPsec Administrator로 지정합니다.
  - 관리자는 Network IPsec Management 및 Network Management 권한 프로파일을 역할에 지정합니다.
  - 그런 다음 관리자는 IPsec Administrator 역할을 해당 사용자에게 지정합니다.

▼ IKE 및 IPsec 서비스를 관리하는 방법

다음 단계에서는 IPsec, IKE 및 수동 키 관리에 대한 SMF 서비스의 가장 일반적인 사용을 제공합니다. 기본적으로 `policy` 및 `ipsecalgs` 서비스는 사용으로 설정됩니다. 또한 기본적으로 `ike` 및 `manual-key` 서비스는 사용 안함으로 설정됩니다.

1 IPsec 정책을 관리하려면 다음 중 하나를 수행합니다.

- **ipsecinit.conf** 파일에 새 정책을 추가한 후 **policy** 서비스를 새로 고칩니다.

```
# svcadm refresh svc:/network/ipsec/policy
```

- 서비스 등록 정보의 값을 변경한 후 등록 정보 값을 확인한 다음 **policy** 서비스를 새로 고치고 다시 시작합니다.

```
# svccfg -s policy setprop config/config_file=/etc/inet/MyIpsecinit.conf
# svcprop -p config/config_file policy
/etc/inet/MyIpsecinit.conf
# svcadm refresh svc:/network/ipsec/policy
# svcadm restart svc:/network/ipsec/policy
```

2 키를 자동으로 관리하려면 다음 중 하나를 수행합니다.

- **/etc/inet/ike/config** 파일에 항목을 추가한 후 **ike** 서비스를 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/ike
```

- **/etc/inet/ike/config** 파일에서 항목을 변경한 후 **ike** 서비스를 다시 시작합니다.

```
# svcadm restart svc:/network/ipsec/ike
```

- 서비스 등록 정보의 값을 변경한 후 등록 정보 값을 확인한 다음 서비스를 새로 고치고 다시 시작합니다.

```
# svccfg -s ike setprop config/admin_privilege=modkeys
# svcprop -p config/admin_privilege ike
modkeys
# svcadm refresh svc:/network/ipsec/ike
# svcadm restart svc:/network/ipsec/ike
```

- **ike** 서비스를 중지하려면 사용 안함으로 설정합니다.

```
# svcadm disable svc:/network/ipsec/ike
```

3 키를 수동으로 관리하려면 다음 중 하나를 수행합니다.

- **/etc/inet/secret/ipseckeys** 파일에 항목을 추가한 후 **manual-key** 서비스를 사용으로 설정합니다.

```
# svcadm enable svc:/network/ipsec/manual-key
```

- **ipseckeys** 파일을 변경한 수 서비스를 새로 고칩니다.

```
# svcadm refresh manual-key
```

- 서비스 등록 정보의 값을 변경한 후 등록 정보 값을 확인한 다음 서비스를 새로 고치고 다시 시작합니다.

```
# svccfg -s manual-key setprop config/config_file=/etc/inet/secret/MyIpseckeyfile
# svcprop -p config/config_file manual-key
/etc/inet/secret/MyIpseckeyfile
# svcadm refresh svc:/network/ipsec/manual-key
```

```
# svcadm restart svc:/network/ipsec/manual-key
```

- **수동 키 관리를 막으려면 `manual-key` 서비스를 사용 안함으로 설정합니다.**

  ```
  # svcadm disable svc:/network/ipsec/manual-key
  ```

4 **IPsec 프로토콜 및 알고리즘 테이블을 수정할 경우 `ipsecalgs` 서비스를 새로 고칩니다.**

```
# svcadm refresh svc:/network/ipsec/ipsecalgs
```

일반 오류 `svcs` *service* 명령을 사용하여 서비스의 상태를 찾습니다. 서비스가 `maintenance` 모드인 경우 `svcs -x` *service* 명령 출력의 디버깅 제안을 따릅니다.

IPsec를 사용하여 VPN 보호

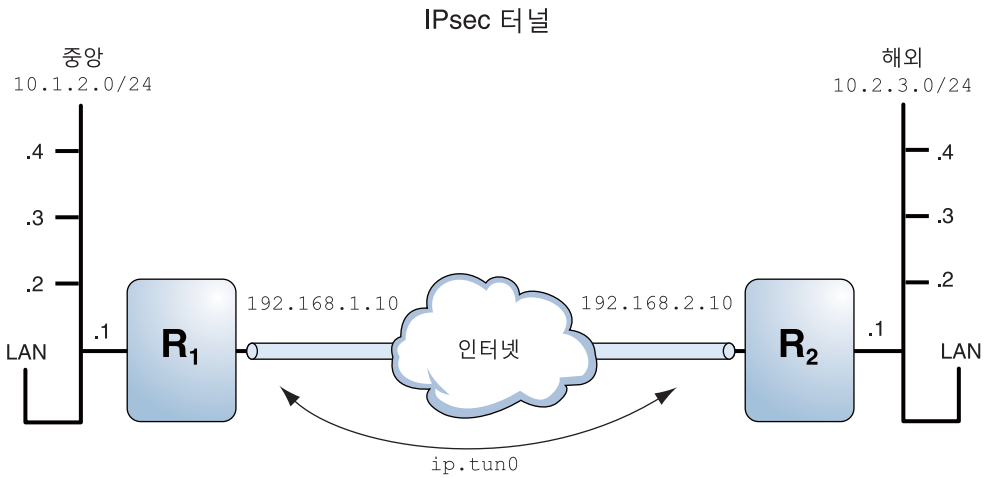
Oracle Solaris는 IPsec로 보호되는 VPN을 구성할 수 있습니다. 터널은 **터널 모드** 또는 **전송 모드**에서 만들 수 있습니다. **터널 모드**는 다른 공급업체의 IPsec 구현과 상호 운용합니다. **전송 모드**는 이전 버전의 Solaris OS와 상호 운용합니다. 터널 모드에 대한 자세한 내용은 467 페이지 “IPsec의 전송 및 터널 모드”를 참조하십시오.

터널 모드의 IPsec는 트래픽을 좀 더 세부적으로 제어합니다. 터널 모드에서 내부 IP 주소의 경우, 단일 포트까지 원하는 특정 보호를 지정할 수 있습니다.

- 터널 모드에서 터널에 대한 IPsec 정책의 예는 494 페이지 “터널 모드를 사용하여 IPsec로 VPN을 보호하는 예”를 참조하십시오.
- VPN을 보호하는 절차는 496 페이지 “IPsec를 사용하여 VPN 보호(작업 맵)”를 참조하십시오.

터널 모드를 사용하여 IPsec로 VPN을 보호하는 예

그림 20–1 IPsec 터널 다이어그램



다음 예에서는 터널이 LAN의 모든 서브넷에 대해 구성되어 있다고 가정합니다.

```
## Tunnel configuration ##
# Tunnel name is ip.tun0
# Intranet point for the source is 10.1.2.1
# Intranet point for the destination is 10.2.3.1
# Tunnel source is 192.168.1.10
# Tunnel destination is 192.168.2.10
```

예 20–6 모든 서브넷에서 사용할 수 있는 터널 만들기

이 예에서는 그림 20–1과 같이 중앙 LAN의 로컬 LAN에서 모든 트래픽이 라우터 1을 거쳐 라우터 2로 터널링된 다음 해외 LAN의 모든 로컬 LAN에 전달될 수 있습니다. 이 트래픽은 AES로 암호화됩니다.

```
## IPsec policy ##
{tunnel ip.tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

예 20–7 두 서브넷만 연결하는 터널 만들기

이 예에서는 중앙 LAN의 서브넷 10.1.2.0/24와 해외 LAN의 서브넷 10.2.3.0/24 사이의 트래픽만 터널링되고 암호화됩니다. 중앙에 대한 다른 IPsec 정책이 없을 때 중앙 LAN에서 이 터널을 통해 다른 LAN에 대한 트래픽을 경로 지정하려고 시도하면 트래픽이 라우터 1에서 삭제됩니다.

```
## IPsec policy ##
{tunnel ip.tun0 negotiate tunnel laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs aes encr_auth_algs sha1 shared}
```

예 20–8 두 서브넷 간의 sendmail 트래픽에 대한 터널만 만들기

이 예에서는 sendmail 트래픽에 대한 터널만 만듭니다. 트래픽은 중앙 LAN의 서브넷 10.1.2.0/24에서 해외 LAN의 서브넷 10.2.3.0/24에 있는 전자 메일 서버로 전달됩니다. 전자 메일은 Blowfish를 사용하여 암호화됩니다. 정책은 원격 및 로컬 전자 메일 포트에 적용됩니다. rport 정책은 중앙이 해외의 원격 전자 메일 포트에 보내는 전자 메일을 보호합니다. lport 정책은 중앙에서 해외로부터 로컬 포트 25에서 수신하는 전자 메일을 보호합니다.

```
## IPsec policy for email from Central to Overseas ##
{tunnel ip.tun0 negotiate tunnel ulp tcp rport 25
 laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}

## IPsec policy for email from Overseas to Central ##
{tunnel ip.tun0 negotiate tunnel ulp tcp lport 25
 laddr 10.1.2.0/24 raddr 10.2.3.0/24}
 ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}
```

예 20–9 모든 서브넷에 대한 FTP 트래픽의 터널 만들기

이 예에서 IPsec 정책은 중앙 LAN의 모든 서브넷에서 해외 LAN의 모든 서브넷까지 AES를 사용하여 그림 20–1의 FTP 포트를 보호합니다. 이 구성은 FTP의 활성 모드에 대해 작동합니다.

```
## IPsec policy for outbound FTP from Central to Overseas ##
{tunnel ip.tun0 negotiate tunnel ulp tcp rport 21}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
{tunnel ip.tun0 negotiate tunnel ulp tcp lport 20}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## IPsec policy for inbound FTP from Central to Overseas ##
{tunnel ip.tun0 negotiate tunnel ulp tcp lport 21}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
{tunnel ip.tun0 negotiate tunnel ulp tcp rport 20}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

IPsec를 사용하여 VPN 보호(작업 맵)

다음 작업 맵에서는 인터넷을 통해 트래픽을 보호하도록 IPsec를 구성하는 절차를 안내합니다. 이러한 절차는 인터넷을 거치는 두 시스템 사이에 보안 VPN(가상 사설망)을 설정합니다. 이러한 기술은 일반적으로 원격 사무실을 인터넷을 통해 회사 네트워크에 안전하게 연결하는 데 사용됩니다.

작업	설명	수행 방법
IPv4를 통해 터널 모드로 터널 트래픽을 보호합니다.	두 Solaris 10 시스템 사이 또는 Solaris 10 시스템과 Oracle Solaris 11 시스템 사이에서 터널 모드로 트래픽을 보호합니다. Solaris 10 시스템은 Solaris 10 7/07 릴리스 이상을 실행 중이어야 합니다. 또한 Solaris 10 시스템 또는 Oracle Solaris 11 시스템과 다른 플랫폼에서 실행 중인 시스템 사이에서 터널 모드로 트래픽을 보호합니다. Solaris 10 시스템은 Solaris 10 7/07 릴리스 이상을 실행 중이어야 합니다.	499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”
IPv6을 통해 터널 모드로 터널 트래픽을 보호합니다.	IPv6 프로토콜을 사용 중인 두 Oracle Solaris 시스템 사이에서 터널 모드로 트래픽을 보호합니다.	508 페이지 “IPv6을 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”
IPv4를 통해 전송 모드로 터널 트래픽을 보호합니다.	두 Solaris 10 시스템 사이 또는 Solaris 10 시스템과 Oracle Solaris 시스템 사이에서 전송 모드로 트래픽을 보호합니다. Solaris 10 시스템은 Solaris 10 7/07 릴리스 이상을 실행 중이어야 합니다. 또한 Solaris OS 및 Solaris 10의 이전 버전을 실행 중인 시스템 또는 Oracle Solaris 시스템 사이에서 전송 모드로 트래픽을 보호합니다. Solaris 10 시스템은 Solaris 10 7/07 릴리스 이상을 실행 중이어야 합니다.	514 페이지 “IPv4를 사용하여 전송 모드의 IPsec 터널로 VPN을 보호하는 방법”
	더 이상 사용되지 않는 이전 구문을 사용하여 트래픽을 보호합니다. 이 방법은 이전 버전의 Solaris OS를 실행 중인 시스템과 통신할 때 유용합니다. 이 방법을 사용하면 두 시스템의 구성 파일을 간단하게 비교할 수 있습니다.	예 20–11 예 20–16
IPv6을 통해 전송 모드로 터널 트래픽을 보호합니다.	IPv6 프로토콜을 사용 중인 두 Oracle Solaris 시스템 사이에서 전송 모드로 트래픽을 보호합니다.	520 페이지 “IPv6을 사용하여 전송 모드의 IPsec 터널로 VPN을 보호하는 방법”
IP 속임수를 방지합니다.	시스템에서 패킷의 암호를 해독하지 않고 VPN을 통과하여 패킷을 전달하지 않도록 SMF 서비스를 만듭니다.	526 페이지 “IP 속임수를 방지하는 방법”

VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명

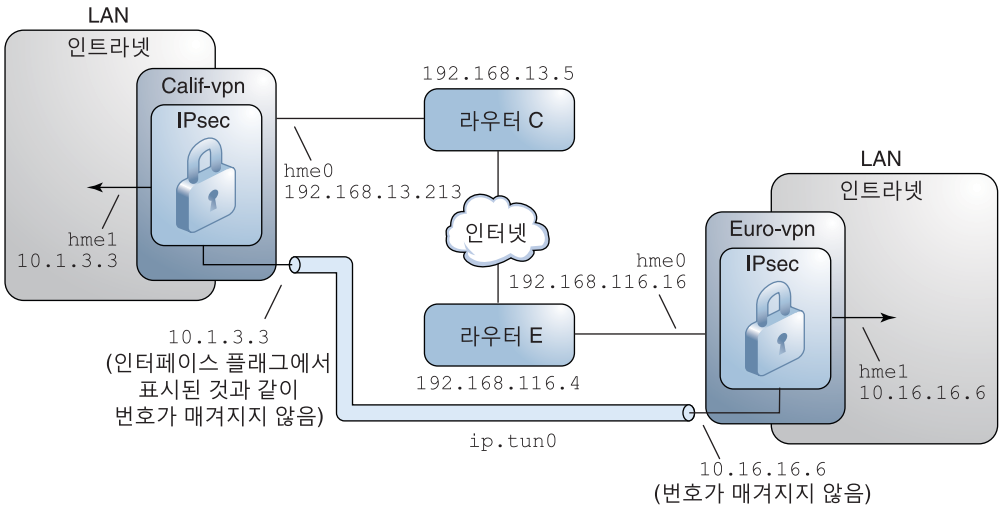
이 절에 나오는 절차에서는 다음 설정을 가정합니다. 네트워크 그림은 그림 20–2를 참조하십시오.

- 각 시스템은 IPv4 주소 공간을 사용합니다.

  IPv6 주소에 대한 유사한 예는 508 페이지 “IPv6을 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”을 참조하십시오.

- 각 시스템에는 두 개의 인터페이스가 있습니다. `hme0` 인터페이스는 인터넷에 연결됩니다. 이 예에서 인터넷 IP 주소는 `192.168`로 시작됩니다. `hme1` 인터페이스는 회사의 LAN(인트라넷)에 연결됩니다. 이 예에서는 인트라넷 IP 주소가 숫자 `10`으로 시작됩니다.
- 각 시스템에는 SHA-1 알고리즘을 사용하는 ESP 인증이 필요합니다. SHA-1 알고리즘에는 160비트 키가 필요합니다.
- 각 시스템에는 AES 알고리즘을 사용하는 ESP 암호화가 필요합니다. AES 알고리즘은 128비트 또는 256비트 키를 사용합니다.
- 각 시스템은 인터넷에 직접 액세스되는 라우터에 연결할 수 있습니다.
- 각 시스템은 공유 보안 연결을 사용합니다.

그림 20–2 인터넷으로 구분된 사무실 사이의 샘플 VPN



위의 그림에 나온 대로 IPv4 네트워크에 대한 절차에서는 다음 구성 매개변수를 사용합니다.

매개변수	유럽	캘리포니아
시스템 이름	`enigma`	`partym`
시스템 인트라넷 인터페이스	`hme1`	`hme1`
시스템 인트라넷 주소(또한 단계 7의 *-point* 주소)	`10.16.16.6`	`10.1.3.3`
시스템 인터넷 인터페이스	`hme0`	`hme0`
시스템 인터넷 주소(또한 단계 7의 *tsrc* 주소)	`192.168.116.16`	`192.168.13.213`

매개변수	유럽	캘리포니아
인터넷 라우터의 이름	router-E	router-C
인터넷 라우터의 주소	192.168.116.4	192.168.13.5
터널 이름	ip.tun0	ip.tun0

다음 IPv6 주소가 절차에 사용됩니다. 터널 이름은 동일합니다.

매개변수	유럽	캘리포니아
시스템 인트라넷 주소	6000:6666::aaaa:1116	6000:3333::eeee:1113
시스템 인터넷 주소	2001::aaaa:6666:6666	2001::eeee:3333:3333
인터넷 라우터의 주소	2001::aaaa:0:4	2001::eeee:0:1

▼ IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법

터널 모드에서 내부 IP 패킷은 해당 컨텐츠를 보호하는 IPsec 정책을 결정합니다.

이 절차는 절차 477 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”을 확장합니다. 설정은 497 페이지 “VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명”에 설명되어 있습니다.

주 – 두 시스템에서 이 절차의 단계를 수행하십시오.

두 시스템 연결과 함께 이러한 두 시스템에 연결되는 두 인트라넷을 연결하게 됩니다. 이 절차에서 시스템은 게이트웨이로 작동합니다.

시작하기 전에 시스템 또는 공유 IP 영역에 대한 IPsec 정책을 구성하려면 전역 영역에 있어야 합니다. 배타적 IP 영역의 경우 비전역 영역에서 IPsec 정책을 구성합니다.

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

2 IPsec를 구성하기 전에 패킷의 플로우를 제어합니다.

a. IP 전달 및 IP 동적 경로 지정이 사용 안함으로 설정되도록 합니다.

```
 # routeadm
Configuration   Current             Current
       Option   Configuration       System State
---------------------------------------------------------------
IPv4 routing      default (enabled)    enabled
IPv4 forwarding   disabled             disabled
...
```

IP 전달 및 IP 동적 경로 지정이 사용으로 설정된 경우 사용 안함으로 설정합니다.

```
# routeadm -d ipv4-routing -d ipv4-forwarding
# routeadm -u
```

IP 전달을 해제하면 패킷이 이 시스템을 통해 한 네트워크에서 다른 네트워크로 전달되지 않습니다. `routeadm` 명령에 대한 설명은 `routeadm(1M)` 매뉴얼 페이지를 참조하십시오.

b. 엄격한 IP 대상 멀티홈을 설정합니다.

```
# ndd -set /dev/ip ip_strict_dst_multihoming 1
```

엄격한 IP 대상 멀티홈을 설정하면 시스템의 대상 주소 중 하나에 대한 패킷이 올바른 대상 주소에 도달해야 합니다.

엄격한 대상 멀티홈이 사용으로 설정되면 특정 인터페이스에 도착하는 패킷의 주소를 해당 인터페이스의 로컬 IP 주소 중 하나로 지정해야 합니다. 기타 모든 패킷은 시스템의 다른 로컬 주소로 지정된 패킷이라도 삭제됩니다.

주의 – 시스템을 부트하면 멀티홈 값이 기본값으로 돌아갑니다. 변경된 값을 지속 값으로 만들려면 526 페이지 “IP 속임수를 방지하는 방법”을 참조하십시오.

c. 대부분의 네트워크 서비스 및 가능한 모든 네트워크 서비스를 사용 안함으로 설정합니다.

주 – “제한된” SMF 프로파일을 사용하여 시스템이 설치된 경우 이 단계를 건너뛸 수 있습니다. Oracle Solaris의 Secure Shell 기능은 제외한 네트워크 서비스가 사용 안함으로 설정됩니다.

네트워크 서비스를 사용 안함으로 설정하면 IP 패킷이 시스템에 손상을 주지 않습니다. 예를 들어 SNMP 데몬, `telnet` 연결 또는 `rlogin` 연결이 악용될 수 있습니다.

다음 옵션 중 하나를 선택합니다.

- Solaris 10 11/06 릴리스 또는 이후 릴리스를 실행 중인 경우 "제한된" SMF 프로파일을 실행합니다.

```
# netservices limited
```

- 그렇지 않으면 네트워크 서비스를 개별적으로 사용 안함으로 설정합니다.

```
# svcadm disable network/ftp:default
# svcadm disable network/finger:default
# svcadm disable network/login:rlogin
# svcadm disable network/nfs/server:default
# svcadm disable network/rpc/rstat:default
# svcadm disable network/smtp:sendmail
# svcadm disable network/telnet:default
```

d. 대부분의 네트워크 서비스가 사용 안함으로 설정되었는지 확인합니다.

루프백 마운트 및 ssh 서비스가 실행 중인지 확인합니다.

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

3 두 시스템 사이에 SA 쌍을 추가합니다.

다음 옵션 중 하나를 선택합니다.

- SA에 대한 키를 관리하도록 IKE를 구성합니다. 545 페이지 "IKE 구성(작업 맵)"의 절차 중 하나를 사용하여 VPN에 대한 IKE를 구성합니다.
- 수동으로 키를 관리해야 하는 분명한 이유가 있다면 485 페이지 "수동으로 IPsec 보안 연관을 만드는 방법"을 참조하십시오.

4 IPsec 정책을 추가합니다.

/etc/inet/ipsecinit.conf 파일을 편집하여 VPN에 대한 IPsec 정책을 추가합니다. 정책을 강화하려면 예 20–12를 참조하십시오. 추가 예는 494 페이지 "터널 모드를 사용하여 IPsec로 VPN을 보호하는 예"를 참조하십시오.

이 정책에서 로컬 LAN의 시스템과 게이트웨이의 내부 IP 주소 사이에는 IPsec 보호가 필요하지 않으므로 bypass 명령문이 추가됩니다.

a. enigma 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.16.16.6 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. partym 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}
```

```
# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate tunnel}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

5 **(옵션) IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 **IPsec를 사용하여 터널을 구성하고 보호하려면 Oracle Solaris 릴리스에 따라 단계를 수행합니다.**

- **Solaris 10 4/09 릴리스부터 단계 7에서 단계 13까지의 단계를 수행한 다음 단계 22의 경로 지정 프로토콜을 실행합니다.**
- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우 단계 14에서 단계 22까지의 단계를 수행합니다.**

7 **/etc/hostname.ip.tun0 파일에서 터널 ip.tun0을 구성합니다.**

파일 구문은 다음과 같습니다.

system1-point system2-point `tsrc` *system1-taddr* `tdst` *system2-taddr* `router up`

a. **enigma 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.**

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 tdst 192.168.13.213 router up
```

b. **partym 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.**

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 tdst 192.168.116.16 router up
```

8 **만든 IPsec 정책을 사용하여 터널을 보호합니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 **터널 구성 파일의 내용을 커널로 읽어오려면 네트워크 서비스를 다시 시작합니다.**

```
# svcadm restart svc:/network/initial:default
```

10 **hme1 인터페이스에 대한 IP 전달을 설정합니다.**

a. **enigma 시스템에서 /etc/hostname.hme1 파일에 라우터 항목을 추가합니다.**

```
192.168.116.16 router
```

b. **partym 시스템에서 /etc/hostname.hme1 파일에 라우터 항목을 추가합니다.**

```
192.168.13.213 router
```

IP 전달은 다른 곳에서 도달한 패킷을 전달할 수 있음을 의미합니다. 또한 IP 전달은 이 인터페이스에서 떠난 패킷이 다른 곳에서 왔을 수 있음을 의미합니다. 패킷을 성공적으로 전달하려면 수신 인터페이스와 전송 인터페이스에 모두 IP 전달이 설정되어 있어야 합니다.

hme1 인터페이스는 인트라넷 **내부**에 있으므로 hme1에 대해 IP 전달이 설정되어 있어야 합니다. ip.tun0은 인터넷을 통해 두 시스템을 연결하므로 ip.tun0에 대해 IP 전달이 설정되어 있어야 합니다.

hme0 인터페이스의 경우 **외부** 공격자가 보호된 인트라넷에 패킷을 주입하지 못하도록 IP 전달이 해제되어 있습니다. **외부**는 인터넷을 의미합니다.

11 경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.

a. enigma 시스템에서 /etc/hostname.hme0 파일에 private 플래그를 추가합니다.

```
10.16.16.6 private
```

b. partym 시스템에서 /etc/hostname.hme0 파일에 private 플래그를 추가합니다.

```
10.1.3.3 private
```

hme0에 IP 전달이 해제되어 있더라도 경로 지정 프로토콜 구현은 여전히 인터페이스를 알릴 수 있습니다. 예를 들어, in.routed 프로토콜은 hme0을 인트라넷 내부의 피어에 패킷을 전달하는 데 사용할 수 있음을 알릴 수 있습니다. 인터페이스의 **개인** 플래그를 설정하여 알림을 막을 수 있습니다.

12 hme0 인터페이스를 통한 기본 경로를 수동으로 추가합니다.

기본 경로는 인터넷에 직접 액세스되는 라우터에 있어야 합니다.

a. enigma 시스템에서 다음 경로를 추가합니다.

```
# route add default 192.168.116.4
```

b. partym 시스템에서 다음 경로를 추가합니다.

```
# route add default 192.168.13.5
```

hme0 인터페이스는 인트라넷의 일부가 아니지만 hme0은 인터넷을 거쳐 피어 시스템에 도달할 필요가 없습니다. 피어를 찾으려면 hme0은 인터넷 경로 지정에 대한 정보가 필요합니다. VPN 시스템은 나머지 인터넷에 라우터가 아닌 호스트로 나타납니다. 따라서 기본 라우터를 사용하거나 라우터 검색 프로토콜을 실행하여 피어 시스템을 찾을 수 있습니다. 자세한 내용은 route(1M) 및 in.routed(1M) 매뉴얼 페이지를 참조하십시오.

13 절차를 완료하려면 단계 22로 이동하여 경로 지정 프로토콜을 실행합니다.

14 터널 ip.tun0을 구성합니다.

주 – 다음 단계에서는 Solaris 10 4/09 이전 릴리스를 실행 중인 시스템에서 터널을 구성합니다.

---

ifconfig 명령을 사용하여 지점 간 인터페이스를 만듭니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 system1-point system2-point \
tsrc system1-taddr tdst system2-taddr
```

a. enigma 시스템에서 다음 명령을 입력합니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213
```

b. partym 시스템에서 다음 명령을 입력합니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.1.3.3 10.16.16.6  \
tsrc 192.168.13.213 tdst 192.168.116.16
```

15 만든 IPsec 정책을 사용하여 터널을 보호합니다.

```
# ipsecconf
```

16 터널에 대한 라우터를 가져옵니다.

```
# ifconfig ip.tun0 router up
```

17 hme1 인터페이스에 대한 IP 전달을 설정합니다.

```
# ifconfig hme1 router
```

IP 전달은 다른 곳에서 도달한 패킷을 전달할 수 있음을 의미합니다. 또한 IP 전달은 이 인터페이스에서 떠난 패킷이 다른 곳에서 왔을 수 있음을 의미합니다. 패킷을 성공적으로 전달하려면 수신 인터페이스와 전송 인터페이스에 모두 IP 전달이 설정되어 있어야 합니다.

hme1 인터페이스는 인트라넷 **내부**에 있으므로 hme1에 대해 IP 전달이 설정되어 있어야 합니다. ip.tun0은 인터넷을 통해 두 시스템을 연결하므로 ip.tun0에 대해 IP 전달이 설정되어 있어야 합니다.

hme0 인터페이스의 경우 **외부** 공격자가 보호된 인트라넷에 패킷을 주입하지 못하도록 IP 전달이 해제되어 있습니다. **외부**는 인터넷을 의미합니다.

18 경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.

```
# ifconfig hme0 private
```

hme0에 IP 전달이 해제되어 있더라도 경로 지정 프로토콜 구현은 여전히 인터페이스를 알릴 수 있습니다. 예를 들어, in.routed 프로토콜은 hme0이 인트라넷 내부의 피어에 패킷을 전달할 수 있음을 알릴 수 있습니다. 인터페이스의 **개인** 플래그를 설정하여 알림을 막을 수 있습니다.

19 hme0을 통한 기본 경로를 수동으로 추가합니다.

기본 경로는 인터넷에 직접 액세스되는 라우터에 있어야 합니다.

a. enigma 시스템에서 다음 경로를 추가합니다.

```
# route add default 192.168.116.4
```

b. partym 시스템에서 다음 경로를 추가합니다.

```
# route add default 192.168.13.5
```

hme0 인터페이스는 인트라넷의 일부가 아니지만 hme0은 인터넷을 거쳐 피어 시스템에 도달할 필요가 없습니다. 피어를 찾으려면 hme0은 인터넷 경로 지정에 대한 정보가 필요합니다. VPN 시스템은 나머지 인터넷에 라우터가 아닌 호스트로 나타납니다. 따라서 기본 라우터를 사용하거나 라우터 검색 프로토콜을 실행하여 피어 시스템을 찾을 수 있습니다. 자세한 내용은 route(1M) 및 in.routed(1M) 매뉴얼 페이지를 참조하십시오.

20 /etc/hostname.ip.tun0 파일에 항목을 추가하여 재부트 이후 VPN이 시작하도록 합니다.

system1-point system2-point tsrc *system1-taddr* tdst *system2-taddr* router up

a. enigma 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 tdst 192.168.13.213 router up
```

b. partym 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 tdst 192.168.116.16 router up
```

21 인터페이스 파일에서 올바른 매개변수를 경로 지정 데몬에 전달하도록 구성합니다.

a. enigma 시스템에서 /etc/hostname. *interface* **파일을 수정합니다.**

```
# cat /etc/hostname.hme0
## enigma
10.16.16.6 private

# cat /etc/hostname.hme1
## enigma
192.168.116.16 router
```

b. partym 시스템에서 /etc/hostname. *interface* **파일을 수정합니다.**

```
# cat /etc/hostname.hme0
## partym
10.1.3.3 private

# cat /etc/hostname.hme1
## partym
192.168.13.213 router
```

22 경로 지정 프로토콜을 실행합니다.

```
# routeadm -e ipv4-routing
# routeadm -u
```

경로 지정 프로토콜을 실행하기 전에 경로 지정 프로토콜을 구성해야 합니다. 자세한 내용은 236 페이지 “Oracle Solaris의 경로 지정 프로토콜”을 참조하십시오. 절차는 112 페이지 “IPv4 라우터 구성 방법”을 참조하십시오.

예 20–10 테스트 시 임시 터널 만들기

이 예에서 관리자는 Solaris 10 4/09 시스템에서 터널 만들기를 테스트합니다. 나중에 관리자는 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법” 절차를 사용하여 터널을 영구 터널로 만듭니다. 테스트 시 관리자는 시스템 `system1` 및 `system2`에서 다음과 같은 일련의 단계를 수행합니다.

- 두 시스템에서 관리자는 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”의 처음 5개 단계를 수행합니다.
- 관리자는 `ifconfig` 명령을 사용하여 임시 터널을 연결하고 구성합니다.

```
system1 # ifconfig ip.tun0 plumb
system1 # ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
          tsrc 192.168.116.16 tdst 192.168.13.213

# ssh system2
Password:      admin-password-on-system2
system2 # ifconfig ip.tun0 plumb
system2 # ifconfig ip.tun0  10.1.3.3 10.16.16.6 \
          tsrc 192.168.13.213 tdst 192.168.116.16
```

- 관리자는 터널에 대한 IPsec 정책을 사용으로 설정합니다. 정책은 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”의 단계 4에서 만들어졌습니다.

```
system1 # svcadm refresh svc:/network/ipsec/policy:default
system2 # svcadm refresh svc:/network/ipsec/policy:default
```

- 관리자는 인터넷 인터페이스를 라우터로 만들고 경로 지정 프로토콜이 인트라넷 인터페이스를 통해 이동하지 못하도록 합니다.

```
system1 # ifconfig hme1 router ; ifconfig hme0 private

system2 # ifconfig hme1 router ; ifconfig hme0 private
```

- 관리자는 두 시스템에서 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”의 단계 12 및 단계 22를 수행하여 수동으로 경로 지정을 추가하고 경로 지정 프로토콜을 실행합니다.

예 20–11 명령줄을 사용하여 이전 버전의 Solaris 시스템에 대한 터널 만들기

Solaris 10 7/07 릴리스에서 `ifconfig` 명령 구문이 간단해졌습니다. 이 예에서 관리자는 Solaris 10 7/07 이전 릴리스인 Solaris 버전을 실행 중인 시스템에 대한 터널 만들기를

테스트합니다. 관리자는 ifconfig 명령의 원래 구문을 사용하여 두 통신 시스템에서 동일한 명령을 사용할 수 있습니다. 나중에 관리자는 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"을 사용하여 터널을 영구 터널로 만듭니다.

테스트 시 관리자는 시스템 system1 및 system2에서 다음 단계를 수행합니다.

- 두 시스템에서 관리자는 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"의 처음 5개 단계를 수행합니다.
- 관리자는 터널을 연결하고 구성합니다.

```
system1 # ifconfig ip.tun0 plumb
system1 # ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
          tsrc 192.168.116.16 tdst 192.168.13.213 \
          encr_algs aes encr_auth_algs sha1
system1 # ifconfig ip.tun0 router up

# ssh system2
Password:      admin-password-on-system2
system2 # ifconfig ip.tun0 plumb
system2 # ifconfig ip.tun0 10.1.3.3  10.16.16.6 \
          tsrc 192.168.13.213 tdst 192.168.116.16 \
          encr_algs aes encr_auth_algs sha1
system2 # ifconfig ip.tun0 router up
```

- 관리자는 터널에 대한 IPsec 정책을 사용으로 설정합니다. 정책은 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"의 단계 4에서 만들어졌습니다.

```
system1 # svcadm refresh svc:/network/ipsec/policy:default
system2 # svcadm refresh svc:/network/ipsec/policy:default
```

- 관리자는 인터넷 인터페이스를 라우터로 만들고 경로 지정 프로토콜이 인트라넷 인터페이스를 통해 이동하지 못하도록 합니다.

```
system1 # ifconfig hme1 router ; ifconfig hme0 private
system2 # ifconfig hme1 router ; ifconfig hme0 private
```

- 관리자는 두 시스템에서 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"의 단계 12 및 단계 22를 수행하여 경로 지정을 추가합니다.

예 20–12 LAN의 모든 시스템에 대한 IPsec 정책 필요

이 예에서 관리자는 단계 4에서 구성된 bypass 정책을 주석 처리하여 보호를 강화합니다. 이 정책 구성을 사용하면 LAN의 각 시스템은 라우터와 통신하기 위해 IPsec를 활성화해야 합니다.

```
# LAN traffic must implement IPsec.
# {laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate tunnel} ipsec {encr_algs aes encr_auth_algs sha1}
```

예 20–13 IPsec를 사용하여 SMTP 트래픽과 다르게 Telnet 트래픽 보호

이 예에서 첫번째 규칙은 Blowfish 및 SHA-1을 사용하여 포트 23에 대한 `telnet` 트래픽을 보호합니다. 두번째 규칙은 AES 및 MD5를 사용하여 포트 25에 대한 SMTP 트래픽을 보호합니다.

```
{laddr 10.1.3.3 ulp tcp dport 23 dir both}
  ipsec {encr_algs blowfish encr_auth_algs sha1 sa unique}
{laddr 10.1.3.3 ulp tcp dport 25 dir both}
 ipsec {encr_algs aes encr_auth_algs md5 sa unique}
```

예 20–14 터널 모드의 IPsec 터널을 사용하여 다른 네트워크 트래픽과 다르게 서브넷 보호

다음 터널 구성은 터널을 통해 서브넷 `10.1.3.0/24`의 모든 트래픽을 보호합니다.

```
{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

다음 터널 구성은 터널을 통해 서브넷 `10.1.3.0/24`에서 다른 서브넷으로의 트래픽을 보호합니다. `10.2.`*x.x*로 시작하는 서브넷은 터널을 통과합니다.

```
{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24 raddr 10.2.1.0/24}
  ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}

{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24 raddr 10.2.2.0/24}
  ipsec {encr_algs blowfish encr_auth_algs sha1 sa shared}

{tunnel ip.tun0 negotiate tunnel laddr 10.1.3.0/24 raddr 10.2.3.0/24}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

▼ IPv6을 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법

IPv6 네트워크에서 VPN을 설정하려면 IPv4 네트워크의 단계를 동일하게 수행합니다. 그러나 명령 구문은 약간 다릅니다. 특정 명령을 실행하는 이유에 대한 자세한 설명은 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”에서 해당하는 단계를 참조하십시오.

주 – 두 시스템에서 이 절차의 단계를 수행하십시오.

이 절차에서는 다음 구성 매개변수를 사용합니다.

매개변수	유럽	캘리포니아
시스템 이름	enigma	partym
시스템 인트라넷 인터페이스	hme1	hme1
시스템 인터넷 인터페이스	hme0	hme0
시스템 인트라넷 주소	6000:6666::aaaa:1116	6000:3333::eeee:1113
시스템 인터넷 주소	2001::aaaa:6666:6666	2001::eeee:3333:3333
인터넷 라우터의 이름	router-E	router-C
인터넷 라우터의 주소	2001::aaaa:0:4	2001::eeee:0:1
터널 이름	ip6.tun0	ip6.tun0

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

2 IPsec를 구성하기 전에 패킷의 플로우를 제어합니다.

이러한 명령의 효과는 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”의 단계 2를 참조하십시오.

a. IP 전달 및 IP 동적 경로 지정이 사용 안함으로 설정되도록 합니다.

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

IP 전달 및 IP 동적 경로 지정이 사용으로 설정된 경우 다음을 입력하여 사용 안함으로 설정할 수 있습니다.

```
# routeadm -d ipv6-forwarding -d ipv6-routing
# routeadm -u
```

b. 엄격한 IP 대상 멀티홈을 설정합니다.

```
# ndd -set /dev/ip ip6_strict_dst_multihoming 1
```

주의 – 시스템을 부트하면 `ip6_strict_dst_multihoming` 값이 기본값으로 돌아갑니다. 변경된 값을 지속 값으로 만들려면 526 페이지 “IP 속임수를 방지하는 방법”을 참조하십시오.

c. 대부분의 네트워크 서비스 및 가능한 모든 네트워크 서비스를 사용 안함으로 설정합니다.

주 – "제한된" SMF 프로파일을 사용하여 시스템이 설치되어 있는 경우 이 단계를 건너뛸 수 있습니다. Secure Shell은 제외한 네트워크 서비스가 사용 안함으로 설정됩니다.

네트워크 서비스를 사용 안함으로 설정하면 IP 패킷이 시스템에 손상을 주지 않습니다. 예를 들어 SNMP 데몬, `telnet` 연결 또는 `rlogin` 연결이 악용될 수 있습니다.

다음 옵션 중 하나를 선택합니다.

- Solaris 10 11/06 릴리스 또는 이후 릴리스를 실행 중인 경우 “제한된” SMF 프로파일을 실행합니다.

```
# netservices limited
```

- 그렇지 않으면 네트워크 서비스를 개별적으로 사용 안함으로 설정합니다.

```
# svcadm disable network/ftp:default
# svcadm disable network/finger:default
# svcadm disable network/login:rlogin
# svcadm disable network/nfs/server:default
# svcadm disable network/rpc/rstat:default
# svcadm disable network/smtp:sendmail
# svcadm disable network/telnet:default
```

d. 대부분의 네트워크 서비스가 사용 안함으로 설정되었는지 확인합니다.

루프백 마운트 및 `ssh` 서비스가 실행 중인지 확인합니다.

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

3 두 시스템 사이에 SA 쌍을 추가합니다.

다음 옵션 중 하나를 선택합니다.

- SA에 대한 키를 관리하도록 IKE를 구성합니다. 545 페이지 “IKE 구성(작업 맵)”의 절차 중 하나를 사용하여 VPN에 대한 IKE를 구성합니다.
- 수동으로 키를 관리해야 하는 분명한 이유가 있다면 485 페이지 “수동으로 IPsec 보안 연관을 만드는 방법”을 참조하십시오.

4 **VPN에 대한 IPsec 정책을 추가합니다.**

`/etc/inet/ipsecinit.conf` 파일을 편집하여 VPN에 대한 IPsec 정책을 추가합니다.

a. **enigma 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.**

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic to and from this host can bypass IPsec.
{laddr 6000:6666::aaaa:1116 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate tunnel}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. **partym 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.**

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic to and from this host can bypass IPsec.
{laddr 6000:3333::eeee:1113 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate tunnel}
  ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

5 **(옵션) IPsec 정책 파일의 구문을 확인합니다.**

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 **IPsec를 사용하여 터널을 구성하고 보호하려면 Oracle Solaris 릴리스에 따라 단계를 수행합니다.**

- **Solaris 10 4/09 릴리스부터 단계 7에서 단계 13까지의 단계를 수행한 다음 단계 22의 경로 지정 프로토콜을 실행합니다.**
- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우 단계 14에서 단계 22까지의 단계를 수행합니다.**

7 **/etc/hostname.ip6.tun0 파일에서 터널 ip6.tun0을 구성합니다.**

a. **enigma 시스템에서 다음 항목을 hostname.ip6.tun0 파일에 추가합니다.**

```
6000:6666::aaaa:1116 6000:3333::eeee:1113 tsrc 2001::aaaa:6666:6666 tdst 2001::eeee:3333:3333 router up
```

b. **partym 시스템에서 다음 항목을 hostname.ip6.tun0 파일에 추가합니다.**

```
6000:3333::eeee:1113  6000:6666::aaaa:1116 tsrc 2001::eeee:3333:3333 tdst 2001::aaaa:6666:6666 router up
```

8 **만든 IPsec 정책을 사용하여 터널을 보호합니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 **터널 구성 파일의 내용을 커널로 읽어오려면 네트워크 서비스를 다시 시작합니다.**

```
# svcadm restart svc:/network/initial:default
```

10 **hme1 인터페이스에 대한 IP 전달을 설정합니다.**

a. **enigma 시스템에서 라우터 항목을 /etc/hostname6.hme1 파일에 추가합니다.**

```
2001::aaaa:6666:6666 inet6 router
```

b. **partym 시스템에서 라우터 항목을 /etc/hostname6.hme1 파일에 추가합니다.**

```
2001::eeee:3333:3333 inet6 router
```

11 **경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.**

a. **enigma 시스템에서 private 플래그를 /etc/hostname6.hme0 파일에 추가합니다.**

```
6000:6666::aaaa:1116 inet6 private
```

b. **partym 시스템에서 private 플래그를 /etc/hostname6.hme0 파일에 추가합니다.**

```
6000:3333::eeee:1113 inet6 private
```

12 **hme0을 통한 기본 경로를 수동으로 추가합니다.**

a. **enigma 시스템에서 다음 경로를 추가합니다.**

```
# route add -inet6 default 2001::aaaa:0:4
```

b. **partym 시스템에서 다음 경로를 추가합니다.**

```
# route add -inet6 default 2001::eeee:0:1
```

13 **절차를 완료하려면 단계 22로 이동하여 경로 지정 프로토콜을 실행합니다.**

14 **보안 터널 ip6.tun0을 구성합니다.**

주 – 다음 단계에서는 Solaris 10 4/09 이전 릴리스를 실행 중인 시스템에서 터널을 구성합니다.

---

a. **enigma 시스템에서 다음 명령을 입력합니다.**

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333
```

b. **partym 시스템에서 다음 명령을 입력합니다.**

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:3333::eeee:1113 6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666
```

15 **만든 IPsec 정책을 사용하여 터널을 보호합니다.**

```
# ipsecconf
```

16 **터널에 대한 라우터를 가져옵니다.**

```
# ifconfig ip6.tun0 router up
```

17 **각 시스템에서 hme1 인터페이스에 대한 IP 전달을 설정합니다.**

```
# ifconfig hme1 router
```

18 **경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.**

```
# ifconfig hme0 private
```

19 **hme0을 통한 기본 경로를 수동으로 추가합니다.**

기본 경로는 인터넷에 직접 액세스되는 라우터에 있어야 합니다.

a. **enigma 시스템에서 다음 경로를 추가합니다.**

```
# route add -inet6 default 2001::aaaa:0:4
```

b. **partym 시스템에서 다음 경로를 추가합니다.**

```
# route add -inet6 default 2001::eeee:0:1
```

20 **/etc/hostname6.ip6.tun0 파일에 항목을 추가하여 재부트한 후 VPN이 시작하도록 합니다.**

항목은 단계 14의 `ifconfig` 명령에 전달된 매개변수를 복제합니다.

a. **enigma 시스템에서 다음 항목을 hostname6.ip6.tun0 파일에 추가합니다.**

```
6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666  tdst 2001::eeee:3333:3333 router up
```

b. **partym 시스템에서 다음 항목을 hostname6.ip6.tun0 파일에 추가합니다.**

```
6000:3333::eeee:1113 6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666  router up
```

21 **각 시스템에서 올바른 매개변수가 경로 지정 데몬에 전달되도록 인터페이스 파일을 구성합니다.**

a. **enigma 시스템에서 /etc/hostname6.** *interface* **파일을 수정합니다.**

```
# cat /etc/hostname6.hme0
## enigma
6000:6666::aaaa:1116 inet6 private

#  cat /etc/hostname6.hme1
## enigma
2001::aaaa:6666:6666 inet6 router
```

b. partym 시스템에서 /etc/hostname6.*interface* 파일을 수정합니다.

```
# cat /etc/hostname6.hme0
## partym
6000:3333::eeee:1113 inet6 private

# cat /etc/hostname6.hme1
## partym
2001::eeee:3333:3333 inet6 router
```

22 경로 지정 프로토콜을 실행합니다.

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

경로 지정 프로토콜을 실행하기 전에 경로 지정 프로토콜을 구성해야 합니다. 자세한 내용은 236 페이지 "Oracle Solaris의 경로 지정 프로토콜"을 참조하십시오. 절차는 165 페이지 "IPv6 라우터 구성"을 참조하십시오.

▼ IPv4를 사용하여 전송 모드의 IPsec 터널로 VPN을 보호하는 방법

전송 모드에서 외부 헤더는 내부 IP 패킷을 보호하는 IPsec 정책을 결정합니다.

이 절차는 절차 477 페이지 "IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법"을 확장합니다. 두 시스템 연결과 함께 이러한 두 시스템에 연결되는 두 인트라넷을 연결하게 됩니다. 이 절차에서 시스템은 게이트웨이로 작동합니다.

이 절차는 497 페이지 "VPN을 보호하기 위한 IPsec 작업에 대한 네트워크 토폴로지 설명"에 설명되어 있는 설정을 사용합니다. 특정 명령을 실행하는 이유에 대한 자세한 설명은 499 페이지 "IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법"에서 해당하는 단계를 참조하십시오.

주 – 두 시스템에서 이 절차의 단계를 수행하십시오.

---

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 ssh 명령을 사용하십시오.

---

2 IPsec를 구성하기 전에 패킷의 플로우를 제어합니다.

a. IP 전달 및 IP 동적 경로 지정이 사용 안함으로 설정되도록 합니다.

```
# routeadm
Configuration       Current         Current
       Option       Configuration  System State
---------------------------------------------------
IPv4 forwarding     disabled           disabled
   IPv4 routing     default (enabled)   enabled
...
```

IP 전달 및 IP 동적 경로 지정이 사용으로 설정된 경우 다음을 입력하여 사용 안함으로 설정할 수 있습니다.

```
# routeadm -d ipv4-routing -d ipv4-forwarding
# routeadm -u
```

b. 엄격한 IP 대상 멀티홈을 설정합니다.

```
# ndd -set /dev/ip ip_strict_dst_multihoming 1
```

주의 – 시스템을 부트하면 `ip_strict_dst_multihoming` 값이 기본값으로 돌아갑니다. 변경된 값을 지속 값으로 만들려면 526 페이지 “IP 속임수를 방지하는 방법”을 참조하십시오.

c. 대부분의 네트워크 서비스 및 가능한 모든 네트워크 서비스를 사용 안함으로 설정합니다.

주 – "제한된" SMF 프로파일을 사용하여 시스템이 설치되어 있는 경우 이 단계를 건너뛸 수 있습니다. Secure Shell은 제외한 네트워크 서비스가 사용 안함으로 설정됩니다.

네트워크 서비스를 사용 안함으로 설정하면 IP 패킷이 시스템에 손상을 주지 않습니다. 예를 들어 SNMP 데몬, `telnet` 연결 또는 `rlogin` 연결이 악용될 수 있습니다.

다음 옵션 중 하나를 선택합니다.

- Solaris 10 11/06 릴리스 또는 이후 릴리스를 실행 중인 경우 “제한된” SMF 프로파일을 실행합니다.

  ```
  # netservices limited
  ```

- 그렇지 않으면 네트워크 서비스를 개별적으로 사용 안함으로 설정합니다.

  ```
  # svcadm disable network/ftp:default
  # svcadm disable network/finger:default
  # svcadm disable network/login:rlogin
  # svcadm disable network/nfs/server:default
  # svcadm disable network/rpc/rstat:default
  # svcadm disable network/smtp:sendmail
  # svcadm disable network/telnet:default
  ```

d. 대부분의 네트워크 서비스가 사용 안함으로 설정되었는지 확인합니다.

루프백 마운트 및 ssh 서비스가 실행 중인지 확인합니다.

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

3 두 시스템 사이에 SA 쌍을 추가합니다.

다음 옵션 중 하나를 선택합니다.

- SA에 대한 키를 관리하도록 IKE를 구성합니다. 545 페이지 “IKE 구성(작업 맵)”의 절차 중 하나를 사용하여 VPN에 대한 IKE를 구성합니다.
- 수동으로 키를 관리해야 하는 분명한 이유가 있다면 485 페이지 “수동으로 IPsec 보안 연관을 만드는 방법”을 참조하십시오.

4 IPsec 정책을 추가합니다.

/etc/inet/ipsecinit.conf 파일을 편집하여 VPN에 대한 IPsec 정책을 추가합니다. 정책을 강화하려면 예 20–15를 참조하십시오.

a. enigma 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.16.16.6 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. partym 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.

```
# LAN traffic to and from this host can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

5 (옵션) IPsec 정책 파일의 구문을 확인합니다.

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 IPsec를 사용하여 터널을 구성하고 보호하려면 Oracle Solaris 릴리스에 따라 단계를 수행합니다.

- **Solaris 10 4/09 릴리스부터 단계 7에서 단계 13까지의 단계를 수행한 다음 단계 22의 경로 지정 프로토콜을 실행합니다.**
- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우 단계 14에서 단계 22까지의 단계를 수행합니다.**

7 **/etc/hostname.ip.tun0 파일에서 터널 ip.tun0을 구성합니다.**

a. **enigma 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.**

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 tdst 192.168.13.213 router up
```

b. **partym 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.**

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 tdst 192.168.116.16 router up
```

8 **만든 IPsec 정책을 사용하여 터널을 보호합니다.**

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 **hostname.ip.tun0 파일의 내용을 커널로 읽어오려면 네트워크 서비스를 다시 시작합니다.**

```
# svcadm restart svc:/network/initial:default
```

10 **hme1 인터페이스에 대한 IP 전달을 설정합니다.**

a. **enigma 시스템에서 라우터 항목을 /etc/hostname.hme1 파일에 추가합니다.**

```
192.168.116.16 router
```

b. **partym 시스템에서 라우터 항목을 /etc/hostname.hme1 파일에 추가합니다.**

```
192.168.13.213 router
```

11 **경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.**

a. **enigma 시스템에서 private 플래그를 /etc/hostname.hme0 파일에 추가합니다.**

```
10.16.16.6 private
```

b. **partym 시스템에서 private 플래그를 /etc/hostname.hme0 파일에 추가합니다.**

```
10.1.3.3 private
```

12 **hme0를 통한 기본 경로를 수동으로 추가합니다.**

a. **enigma 시스템에서 다음 경로를 추가합니다.**

```
# route add default 192.168.116.4
```

b. **partym 시스템에서 다음 경로를 추가합니다.**

```
# route add default 192.168.13.5
```

13 **절차를 완료하려면 단계 22로 이동하여 경로 지정 프로토콜을 실행합니다.**

14 **터널 ip.tun0을 구성합니다.**

주 – 다음 단계에서는 Solaris 10 4/09 이전 릴리스를 실행 중인 시스템에서 터널을 구성합니다.

`ifconfig` 명령을 사용하여 지점 간 인터페이스를 만듭니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 system1-point system2-point \
tsrc system1-taddr tdst system2-taddr
```

a. `enigma` 시스템에서 다음 명령을 입력합니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213
```

b. `partym` 시스템에서 다음 명령을 입력합니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.1.3.3 10.16.16.6  \
tsrc 192.168.13.213 tdst 192.168.116.16
```

15 만든 IPsec 정책을 사용하여 터널을 보호합니다.

```
# ipsecconf
```

16 터널에 대한 라우터를 가져옵니다.

```
# ifconfig ip.tun0 router up
```

17 `hme1` 인터페이스에 대한 IP 전달을 설정합니다.

```
# ifconfig hme1 router
```

18 경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.

```
# ifconfig hme0 private
```

19 `hme0`을 통한 기본 경로를 수동으로 추가합니다.

기본 경로는 인터넷에 직접 액세스되는 라우터에 있어야 합니다.

```
# route add default router-on-hme0-subnet
```

a. `enigma` 시스템에서 다음 경로를 추가합니다.

```
# route add default 192.168.116.4
```

b. `partym` 시스템에서 다음 경로를 추가합니다.

```
# route add default 192.168.13.5
```

20 **/etc/hostname.ip.tun0 파일에 항목을 추가하여 재부트 이후 VPN이 시작하도록 합니다.**

system1-point system2-point `tsrc` *system1-taddr* `\`
`tdst` *system2-taddr* `encr_algs aes encr_auth_algs sha1 router up`

a. enigma 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 \
tdst 192.168.13.213 router up
```

b. partym 시스템에서 hostname.ip.tun0 파일에 다음 항목을 추가합니다.

```
10.1.3.3 10.16.16.6 tsrc 192.168.13.213 \
tdst 192.168.116.16 router up
```

21 **인터페이스 파일에서 올바른 매개변수를 경로 지정 데몬에 전달하도록 구성합니다.**

a. enigma 시스템에서 /etc/hostname. *interface* **파일을 수정합니다.**

```
# cat /etc/hostname.hme0
## enigma
10.16.16.6 private

# cat /etc/hostname.hme1
## enigma
192.168.116.16 router
```

b. partym 시스템에서 /etc/hostname. *interface* **파일을 수정합니다.**

```
# cat /etc/hostname.hme0
## partym
10.1.3.3 private

# cat /etc/hostname.hme1
## partym
192.168.13.213 router
```

22 **경로 지정 프로토콜을 실행합니다.**

```
# routeadm -e ipv4-routing
# routeadm -u
```

예 20–15 전송 모드의 모든 시스템에 대한 IPsec 정책 필요

이 예에서 관리자는 단계 4에서 구성된 `bypass` 정책을 주석 처리하여 보호를 강화합니다. 이 정책 구성을 사용하면 LAN의 각 시스템은 라우터와 통신하기 위해 IPsec를 활성화해야 합니다.

```
# LAN traffic must implement IPsec.
# {laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip.tun0 negotiate transport} ipsec {encr_algs aes encr_auth_algs sha1}
```

예 20–16 제거된 구문을 사용하여 전송 모드의 IPsec 터널 구성

이 예에서 관리자는 Solaris 10 7/07 시스템을 Oracle Solaris 10 릴리스를 실행 중인 시스템에 연결하고 있습니다. 따라서 관리자는 구성 파일의 Solaris 10 구문을 사용하고 `ifconfig` 명령에 IPsec 알고리즘을 포함시킵니다.

관리자는 구문의 다음 변경 사항과 함께 514 페이지 “IPv4를 사용하여 전송 모드의 IPsec 터널로 VPN을 보호하는 방법” 절차를 수행합니다.

- 단계 4의 경우 `ipsecinit.conf` 파일 구문은 다음과 같습니다.

```
# LAN traffic to and from this address can bypass IPsec.
{laddr 10.1.3.3 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{} ipsec {encr_algs aes encr_auth_algs sha1}
```

- 단계 14에서 단계 16까지의 경우 보안 터널을 구성하는 구문은 다음과 같습니다.

```
# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213 \
encr_algs aes encr_auth_algs sha1

# ifconfig ip.tun0 router up

# ifconfig ip.tun0 plumb

# ifconfig ip.tun0 10.16.16.6 10.1.3.3 \
tsrc 192.168.116.16 tdst 192.168.13.213 \
encr_algs aes encr_auth_algs sha1
```

`ifconfig` 명령에 전달된 IPsec 정책은 `ipsecinit.conf` 파일의 IPsec 정책과 동일해야 합니다. 재부트 시 각 시스템은 해당 정책에 대한 `ipsecinit.conf` 파일을 읽습니다.

- 단계 20의 경우 `hostname.ip.tun0` 파일 구문은 다음과 같습니다.

```
10.16.16.6 10.1.3.3 tsrc 192.168.116.16 \
tdst 192.168.13.213 encr_algs aes encr_auth_algs sha1 router up
```

▼ IPv6을 사용하여 전송 모드의 IPsec 터널로 VPN을 보호하는 방법

IPv6 네트워크에서 VPN을 설정하려면 IPv4 네트워크의 단계를 동일하게 수행합니다. 그러나 명령 구문은 약간 다릅니다. 특정 명령을 실행하는 이유에 대한 자세한 설명은 499 페이지 “IPv4를 사용하여 터널 모드의 IPsec 터널로 VPN을 보호하는 방법”에서 해당하는 단계를 참조하십시오.

주 – 두 시스템에서 이 절차의 단계를 수행하십시오.

이 절차에서는 다음 구성 매개변수를 사용합니다.

매개변수	유럽	캘리포니아
시스템 이름	enigma	partym
시스템 인트라넷 인터페이스	hme1	hme1
시스템 인터넷 인터페이스	hme0	hme0
시스템 인트라넷 주소	6000:6666::aaaa:1116	6000:3333::eeee:1113
시스템 인터넷 주소	2001::aaaa:6666:6666	2001::eeee:3333:3333
인터넷 라우터의 이름	router-E	router-C
인터넷 라우터의 주소	2001::aaaa:0:4	2001::eeee:0:1
터널 이름	ip6.tun0	ip6.tun0

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

2 IPsec를 구성하기 전에 패킷의 플로우를 제어합니다.

a. IP 전달 및 IP 동적 경로 지정이 사용 안함으로 설정되도록 합니다.

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

IP 전달 및 IP 동적 경로 지정이 사용으로 설정된 경우 다음을 입력하여 사용 안함으로 설정할 수 있습니다.

```
# routeadm -d ipv6-forwarding -d ipv6-routing
# routeadm -u
```

b. 엄격한 IP 대상 멀티홈을 설정합니다.

```
# ndd -set /dev/ip ip6_strict_dst_multihoming 1
```

주의 – 시스템을 부트하면 ip6_strict_dst_multihoming 값이 기본값으로 돌아갑니다. 변경된 값을 지속 값으로 만들려면 526 페이지 “IP 속임수를 방지하는 방법”을 참조하십시오.

c. 대부분의 네트워크 서비스가 사용 안함으로 설정되었는지 확인합니다.

루프백 마운트 및 ssh 서비스가 실행 중인지 확인합니다.

```
# svcs | grep network
online         Aug_02   svc:/network/loopback:default
...
online         Aug_09   svc:/network/ssh:default
```

3 두 시스템 사이에 SA 쌍을 추가합니다.

다음 옵션 중 하나를 선택합니다.

- SA에 대한 키를 관리하도록 IKE를 구성합니다. 545 페이지 “IKE 구성(작업 맵)”의 절차 중 하나를 사용하여 VPN에 대한 IKE를 구성합니다.
- 수동으로 키를 관리해야 하는 분명한 이유가 있다면 485 페이지 “수동으로 IPsec 보안 연관을 만드는 방법”을 참조하십시오.

4 IPsec 정책을 추가합니다.

/etc/inet/ipsecinit.conf 파일을 편집하여 VPN에 대한 IPsec 정책을 추가합니다.

a. enigma 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic can bypass IPsec.
{laddr 6000:6666::aaaa:1116 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1}
```

b. partym 시스템에서 다음 항목을 ipsecinit.conf 파일에 입력합니다.

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}

# LAN traffic can bypass IPsec.
{laddr 6000:3333::eeee:1113 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{tunnel ip6.tun0 negotiate transport}
 ipsec {encr_algs aes encr_auth_algs sha1}
```

5 (옵션) IPsec 정책 파일의 구문을 확인합니다.

```
# ipsecconf -c -f /etc/inet/ipsecinit.conf
```

6 IPsec를 사용하여 터널을 구성하고 보호하려면 Oracle Solaris 릴리스에 따라 단계를 수행합니다.

- Solaris 10 4/09 릴리스부터 단계 7에서 단계 13까지의 단계를 수행한 다음 단계 22의 경로 지정 프로토콜을 실행합니다.
- Solaris 10 4/09 이전 릴리스를 실행 중인 경우 단계 14에서 단계 22까지의 단계를 수행합니다.

7 /etc/hostname.ip6.tun0 파일에서 터널 ip6.tun0을 구성합니다.

a. enigma 시스템에서 다음 항목을 hostname.ip6.tun0 파일에 추가합니다.

```
6000:6666::aaaa:1116 6000:3333::eeee:1113 tsrc 2001::aaaa:6666:6666 tdst 2001::eeee:3333:3333 router up
```

b. partym 시스템에서 다음 항목을 hostname.ip6.tun0 파일에 추가합니다.

```
6000:3333::eeee:1113  6000:6666::aaaa:1116 tsrc 2001::eeee:3333:3333 tdst 2001::aaaa:6666:6666 router up
```

8 만든 IPsec 정책을 사용하여 터널을 보호합니다.

```
# svcadm refresh svc:/network/ipsec/policy:default
```

9 hostname.ip6.tun0 파일의 내용을 커널로 읽어오려면 네트워크 서비스를 다시 시작합니다.

```
# svcadm restart svc:/network/initial:default
```

10 hme1 인터페이스에 대한 IP 전달을 설정합니다.

a. enigma 시스템에서 라우터 항목을 /etc/hostname6.hme1 파일에 추가합니다.

```
2001::aaaa:6666:6666 inet6 router
```

b. partym 시스템에서 라우터 항목을 /etc/hostname6.hme1 파일에 추가합니다.

```
2001::eeee:3333:3333 inet6 router
```

11 경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.

a. enigma 시스템에서 private 플래그를 /etc/hostname6.hme0 파일에 추가합니다.

```
6000:6666::aaaa:1116 inet6 private
```

b. partym 시스템에서 private 플래그를 /etc/hostname6.hme0 파일에 추가합니다.

```
6000:3333::eeee:1113 inet6 private
```

12 hme0을 통한 기본 경로를 수동으로 추가합니다.

a. enigma 시스템에서 다음 경로를 추가합니다.

```
# route add -inet6 default 2001::aaaa:0:4
```

b. partym 시스템에서 다음 경로를 추가합니다.

```
# route add -inet6 default 2001::eeee:0:1
```

13 절차를 완료하려면 단계 22로 이동하여 경로 지정 프로토콜을 실행합니다.

14 보안 터널 ip6.tun0을 구성합니다.

주 – 다음 단계에서는 Solaris 10 4/09 이전 릴리스를 실행 중인 시스템에서 터널을 구성합니다.

---

a. enigma 시스템에서 다음 명령을 입력합니다.

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333
```

b. partym 시스템에서 다음 명령을 입력합니다.

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6  6000:3333::eeee:1113  6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666
```

15 만든 IPsec 정책을 사용하여 터널을 보호합니다.

```
# ipsecconf
```

16 터널에 대한 라우터를 가져옵니다.

```
# ifconfig ip6.tun0 router up
```

17 hme1 인터페이스에 대한 IP 전달을 설정합니다.

```
# ifconfig hme1 router
```

18 경로 지정 프로토콜이 인트라넷 내에서 기본 경로를 알리지 않도록 합니다.

```
# ifconfig hme0 private
```

19 각 시스템에서 hme0을 통한 기본 경로를 수동으로 추가합니다.

기본 경로는 인터넷에 직접 액세스되는 라우터에 있어야 합니다.

a. enigma 시스템에서 다음 경로를 추가합니다.

```
# route add -inet6 default 2001::aaaa:0:4
```

b. partym 시스템에서 다음 경로를 추가합니다.

```
# route add -inet6 default 2001::eeee:0:1
```

20 **각 시스템에서 `/etc/hostname6.ip6.tun0` 파일에 항목을 추가하여 재부트한 후 VPN이 시작하도록 합니다.**

항목은 단계 14의 `ifconfig` 명령에 전달된 매개변수를 복제합니다.

a. `enigma` 시스템에서 다음 항목을 `hostname6.ip6.tun0` 파일에 추가합니다.

```
6000:6666::aaaa:1116  6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333  router up
```

b. `partym` 시스템에서 다음 항목을 `hostname6.ip6.tun0` 파일에 추가합니다.

```
6000:3333::eeee:1113  6000:6666::aaaa:1116 \
tsrc 2001::eeee:3333:3333   tdst 2001::aaaa:6666:6666  router up
```

21 **인터페이스 파일에서 올바른 매개변수를 경로 지정 데몬에 전달하도록 구성합니다.**

a. `enigma` 시스템에서 `/etc/hostname6.`*interface* 파일을 수정합니다.

```
# cat /etc/hostname6.hme0
## enigma
6000:6666::aaaa:1116 inet6 private

#  cat /etc/hostname6.hme1
## enigma
2001::aaaa:6666:6666 inet6 router
```

b. `partym` 시스템에서 `/etc/hostname6.`*interface* 파일을 수정합니다.

```
# cat /etc/hostname6.hme0
## partym
6000:3333::eeee:1113 inet6 private

# cat /etc/hostname6.hme1
##
partym2001::eeee:3333:3333 inet6 router
```

22 **경로 지정 프로토콜을 실행합니다.**

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

예 20–17 제거된 구문을 사용하여 IPv6으로 전송 모드의 IPsec 구성

이 예에서 관리자는 Solaris 10 7/07 시스템을 Oracle Solaris 10 릴리스를 실행 중인 시스템에 연결하고 있습니다. 따라서 관리자는 구성 파일의 Solaris 10 구문을 사용하고 `ifconfig` 명령에 IPsec 알고리즘을 포함시킵니다.

관리자는 구문의 다음 변경 사항과 함께 520 페이지 “IPv6을 사용하여 전송 모드의 IPsec 터널로 VPN을 보호하는 방법” 절차를 수행합니다.

- 단계 4의 경우 `ipsecinit.conf` 파일 구문은 다음과 같습니다.

```
# IPv6 Neighbor Discovery messages bypass IPsec.
{ulp ipv6-icmp type 133-137 dir both} pass {}
```

```
# LAN traffic can bypass IPsec.
{laddr 6000:3333::eeee:1113 dir both} bypass {}

# WAN traffic uses ESP with AES and SHA-1.
{} ipsec {encr_algs aes encr_auth_algs sha1}
```

- 단계 14에서 단계 17까지의 경우 보안 터널을 구성하는 구문은 다음과 같습니다.

```
# ifconfig ip6.tun0 inet6 plumb

# ifconfig ip6.tun0 inet6 6000:6666::aaaa:1116 6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666 tdst 2001::eeee:3333:3333 \
encr_algs aes encr_auth_algs sha1

# ifconfig ip6.tun0 inet6 router up
```

  `ifconfig` 명령에 전달된 IPsec 정책은 `ipsecinit.conf` 파일의 IPsec 정책과 동일해야 합니다. 재부트 시 각 시스템은 해당 정책에 대한 `ipsecinit.conf` 파일을 읽습니다.

- 단계 20의 경우 `hostname6.ip6.tun0` 파일 구문은 다음과 같습니다.

```
6000:6666::aaaa:1116  6000:3333::eeee:1113 \
tsrc 2001::aaaa:6666:6666   tdst 2001::eeee:3333:3333 \
encr_algs aes encr_auth_algs sha1 router up
```

▼ IP 속임수를 방지하는 방법

시스템에서 패킷을 해독하지 않고 다른 인터페이스에 패킷을 전달하지 못하도록 지정하려면 IP 속임수를 확인해야 합니다. 한 가지 방지 방법은 `ndd` 명령을 사용하여 엄격한 IP 대상 멀티홈 매개변수를 설정하는 것입니다. 이 매개변수가 SMF 매니페스트에 설정되어 있는 경우 시스템이 재부트되면 매개변수가 설정됩니다.

주 – 두 시스템에서 이 절차의 단계를 수행하십시오.

---

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 사이트 특정 SMF 매니페스트를 만들어 IP 속임수를 확인합니다.

다음 샘플 스크립트 `/var/svc/manifest/site/spoof_check.xml`을 사용합니다.

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type='manifest' name='Custom:ip_spoof_checking'>

<!--    This is a custom smf(5) manifest for this system. Place this
        file in /var/svc/manifest/site, the directory for local
        system customizations. The exec method uses an unstable
```

```
        interface to provide a degree of protection against IP
        spoofing attacks when this system is acting as a router.

        IP spoof protection can also be achieved by using ipfilter(5).
        If ipfilter is configured, this service can be disabled.

        Note: Unstable interfaces might be removed in later
        releases.  See attributes(5).
-->

<service
        name='site/ip_spoofcheck'
        type='service'
        version='1'>

        <create_default_instance enabled='false' />
        <single_instance />

        <!--    Don't enable spoof protection until the
                network is up.
        -->
        <dependency
                name='basic_network'
                grouping='require_all'
                restart_on='none'
                type='service'>
        <service_fmri value='svc:/milestone/network' />
        </dependency>

        <exec_method
                type='method'
                name='start'
                exec='/usr/sbin/ndd -set /dev/ip ip_strict_dst_multihoming 1'
<!--
     For an IPv6 network, use the IPv6 version of this command, as in:
                exec='/usr/sbin/ndd -set /dev/ip ip6_strict_dst_multihoming 1
-->
                timeout_seconds='60'
        />

        <exec_method
                type='method'
                name='stop'
                exec=':true'
                timeout_seconds='3'
        />

        <property_group name='startd' type='framework'>
                <propval
                        name='duration'
                        type='astring'
                        value='transient'
                />
        </property_group>

        <stability value='Unstable' />

</service>
</service_bundle>
```

3 **이 매니페스트를 SMF 저장소로 가져옵니다.**

```
# svccfg import /var/svc/manifest/site/spoof_check.xml
```

4 **ip_spoofcheck 서비스를 사용으로 설정합니다.**

매니페스트 /site/ip_spoofcheck에 정의된 이름을 사용합니다.

```
# svcadm enable /site/ip_spoofcheck
```

5 **ip_spoofcheck 서비스가 온라인 상태인지 확인합니다.**

```
# svcs /site/ip_spoofcheck
```

◆ ◆ ◆ 21 장

21 IP 보안 아키텍처(참조)

이 장에는 다음 참조 정보가 포함되어 있습니다.

네트워크에서 IPsec를 구현하는 방법에 대한 지침은 20 장, “IPsec 구성(작업)”을 참조하십시오. IPsec의 개요는 19 장, “IP 보안 아키텍처(개요)”를 참조하십시오.

IPsec 서비스

SMF(서비스 관리 기능)는 IPsec에 대한 다음 서비스를 제공합니다.

- svc:/network/ipsec/policy 서비스 – IPsec 정책을 관리합니다. 기본적으로 이 서비스는 사용으로 설정됩니다. config_file 등록 정보의 값은 ipsecinit.conf 파일의 위치를 결정합니다. 초기 값은 /etc/inet/ipsecinit.conf입니다.
- svc:/network/ipsec/ipsecalgs 서비스 – IPsec에 사용 가능한 알고리즘을 관리합니다. 기본적으로 이 서비스는 사용으로 설정됩니다.
- svc:/network/ipsec/manual-key 서비스 – 수동 키 관리를 활성화합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. config_file 등록 정보의 값은 ipseckeys 구성 파일의 위치를 결정합니다. 초기 값은 /etc/inet/secret/ipseckeys입니다.
- svc:/network/ipsec/ike 서비스 – IKE를 관리합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 구성 가능한 등록 정보는 589 페이지 “IKE 서비스”를 참조하십시오.

SMF에 대한 자세한 내용은 **Oracle Solaris 관리: 기본 관리**의 18 장, “서비스 관리(개요)”를 참조하십시오. 또한 smf(5), svcadm(1M), svccfg(1M) 매뉴얼 페이지를 참조하십시오.

ipsecconf 명령

ipsecconf 명령을 사용하여 호스트에 대한 IPsec 정책을 구성합니다. 명령을 실행하여 정책을 구성할 때 시스템은 커널에 IPsec 정책 항목을 만듭니다. 시스템은 이러한 항목을 사용하여 모든 인바운드 및 아웃바운드 IP 데이터그램에 대한 정책을 확인합니다. 전달된 데이터그램은 이 명령을 사용하여 추가된 정책 확인에 종속되지 않습니다. ipsecconf 명령은 SPD(보안 정책 데이터베이스)도 구성합니다.

- 전달된 패킷을 보호하는 방법에 대한 자세한 내용은 ifconfig(1M) 및 tun(7M) 매뉴얼 페이지를 참조하십시오.
- IPsec 정책 옵션은 ipsecconf(1M) 매뉴얼 페이지를 참조하십시오.

ipsecconf 명령을 호출하려면 수퍼 유저로 로그인하거나 그에 상응하는 역할로 전환해야 합니다. 명령은 양방향에서 트래픽을 보호하는 항목을 허용합니다. 또한 명령은 한 방향에서만 트래픽을 보호하는 항목도 허용합니다.

로컬 주소 및 원격 주소 형식의 정책 항목은 단일 정책 항목으로 양방향에서 트래픽을 보호할 수 있습니다. 예를 들어, laddr host1 및 raddr host2 패턴을 포함하는 항목은 이름 지정된 호스트에 대해 지정된 방향이 없더라도 양방향에서 트래픽을 보호합니다. 따라서 각 호스트에 대해 하나의 정책 항목만 필요합니다.

소스 주소에서 대상 주소의 형식으로 된 정책 항목은 한 방향으로만 트래픽을 보호합니다. 예를 들어, saddr host1 daddr host2 패턴의 정책 항목은 인바운드 트래픽 또는 아웃바운드 트래픽만 보호하고 양방향을 모두 보호하지는 않습니다. 따라서 양방향의 트래픽을 보호하려면 saddr host2 daddr host1과 같이 ipsecconf 명령에 다른 항목을 전달해야 합니다.

시스템이 부트될 때 IPsec 정책이 활성화되도록 보장하려면 IPsec 정책 파일 /etc/inet/ipsecinit.conf를 만들 수 있습니다. 이 파일은 네트워크 서비스가 시작될 때 읽혀집니다. IPsec 정책 파일을 만드는 방법에 대한 자세한 내용은 475 페이지 “IPsec를 사용하여 트래픽 보호(작업 맵)”를 참조하십시오.

Solaris 10 4/09 릴리스부터 -c 옵션과 함께 ipsecconf 명령을 실행하면 인수로 제공하는 IPsec 정책 파일의 구문을 검사합니다.

ipsecconf 명령으로 추가된 정책 항목은 시스템을 재부트하면 없어집니다. 시스템이 부트할 때 IPsec 정책이 활성화되도록 하려면 정책 항목을 /etc/inet/ipsecinit.conf 파일에 추가합니다. 현재 릴리스에서는 policy 서비스를 새로 고치거나 사용으로 설정합니다. Solaris 10 4/09 이전 릴리스에서는 재부트하거나 ipsecconf 명령을 사용합니다. 예는 475 페이지 “IPsec를 사용하여 트래픽 보호(작업 맵)”를 참조하십시오.

ipsecinit.conf 파일

Oracle Solaris를 시작할 때 IPsec 보안 정책을 사용으로 설정하려면 구성 파일을 만들어 특정 IPsec 정책 항목으로 IPsec를 초기화합니다. 이 파일에 대한 기본 이름은 /etc/inet/ipsecinit.conf입니다. 정책 항목 및 해당 형식에 대한 자세한 내용은 ipsecconf(1M) 매뉴얼 페이지를 참조하십시오. 정책이 구성된 다음 ipsecconf 명령을 사용하여 기존 구성을 보거나 수정할 수 있습니다. Solaris 10 4/09 릴리스부터 기존 구성을 수정하려면 policy 서비스를 새로 고칩니다.

샘플 ipsecinit.conf 파일

Oracle Solaris 소프트웨어에는 샘플 IPsec 정책 파일 ipsecinit.sample이 포함되어 있습니다. 이 파일을 템플리트로 사용하여 자신의 ipsecinit.conf 파일을 만들 수 있습니다. ipsecinit.sample 파일에는 다음 예가 포함되어 있습니다.

```
#
# For example,
#
#     {rport 23} ipsec {encr_algs des encr_auth_algs md5}
#
# will protect the telnet traffic originating from the host with ESP using
# DES and MD5. Also:
#
#     {raddr 10.5.5.0/24} ipsec {auth_algs any}
#
# will protect traffic to or from the 10.5.5.0 subnet with AH
# using any available algorithm.
#
#
# To do basic filtering, a drop rule may be used. For example:
#
#     {lport 23 dir in} drop {}
#     {lport 23 dir out} drop {}
# will disallow any remote system from telnetting in.
#
# If you are using IPv6, it may be useful to bypass neighbor discovery
# to allow in.iked to work properly with on-link neighbors. To do that,
# add the following lines:
#
#        {ulp ipv6-icmp type 133-137 dir both } pass { }
#
# This will allow neighbor discovery to work normally.
```

ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항

네트워크에서 ipsecinit.conf 파일의 복사본을 전송할 때는 특히 주의하십시오. 공격자는 파일이 읽혀질 때 네트워크 마운트 파일을 읽을 수 있습니다. 예를 들어, /etc/inet/ipsecinit.conf 파일을 액세스하거나 NFS로 마운트된 파일 시스템에서 복사할 경우, 공격자가 파일에 포함된 정책을 변경할 수 있습니다.

IPsec 정책은 설정된 연결에 대해 변경할 수 없습니다. 정책을 변경할 수 없는 소켓을 **잠긴 소켓**이라고 합니다. 새 정책 항목은 이미 잠긴 소켓을 보호하지 않습니다. 자세한 내용은 connect(3SOCKET) 및 accept(3SOCKET) 매뉴얼 페이지를 참조하십시오. 의심스러운 경우 연결을 다시 시작하십시오.

이름 지정 시스템을 보호합니다. 다음 두 조건이 충족될 경우 호스트 이름을 더 이상 신뢰할 수 없습니다.

- 소스 주소가 네트워크를 통해 조회할 수 있는 호스트입니다.
- 이름 지정 시스템이 침해되었습니다.

보안 취약성은 실제 도구가 도구의 오용으로 인해 발생하기도 합니다. ipsecconf 명령을 사용할 때는 주의해야 합니다. 가장 안전한 작업 모드를 위해서는 콘솔 또는 기타 하드 연결된 TTY를 사용합니다.

ipsecalgs 명령

암호화 프레임워크는 IPsec에 인증 및 암호화 알고리즘을 제공합니다. ipsecalgs 명령은 각 IPsec 프로토콜이 지원하는 알고리즘을 나열할 수 있습니다. ipsecalgs 구성은 /etc/inet/ipsecalgs 파일에 저장됩니다. 일반적으로 이 파일은 수정할 필요가 없습니다. 하지만 파일을 수정해야 하는 경우 ipsecalgs 명령을 사용합니다. 파일을 직접 편집하면 안됩니다. 현재 릴리스에서 지원되는 알고리즘은 시스템 부트 시 svc:/network/ipsec/ipsecalgs:default 서비스로 커널과 동기화됩니다.

유효한 IPsec 프로토콜 및 알고리즘은 RFC 2407에 포함된 ISAKMP DOI(Domain of Interpretation)에 설명되어 있습니다. 일반적으로 DOI는 데이터 형식, 네트워크 트래픽 교환 유형 및 보안 관련 정보의 이름 지정 규칙을 정의합니다. 보안 관련 정보의 예로 보안 정책, 암호화 알고리즘, 암호화 모드 등이 있습니다.

구체적으로 ISAKMP DOI는 유효한 IPsec 알고리즘 및 해당 프로토콜(PROTO_IPSEC_AH 및 PROTO_IPSEC_ESP)에 대한 이름 지정 및 번호 지정 규칙을 정의합니다. 각 알고리즘은 정확히 하나의 프로토콜과 연결됩니다. 이러한 ISAKMP DOI 정의는 /etc/inet/ipsecalgs 파일에 있습니다. 알고리즘 및 프로토콜 번호는 IANA(Internet Assigned Numbers Authority)에 의해 정의됩니다. ipsecalgs 명령은 IPsec에 대한 알고리즘 목록을 확장할 수 있도록 합니다.

알고리즘에 대한 자세한 내용은 ipsecalgs(1M) 매뉴얼 페이지를 참조하십시오. 암호화 프레임워크에 대한 자세한 내용은 **System Administration Guide: Security Services**의 13 장, “Oracle Solaris Cryptographic Framework (Overview)”를 참조하십시오.

IPsec에 대한 보안 연결 데이터베이스

IPsec 보안 서비스에 대한 키 자료 정보는 보안 연결 데이터베이스(SADB)에서 유지 관리됩니다. SA(보안 연결)는 인바운드 패킷 및 아웃바운드 패킷을 보호합니다. SADB는 특수한 종류의 소켓을 통해 메시지를 보내는 사용자 프로세스 또는 여러 동시 작업 프로세스로 유지 관리됩니다. 이 SADB 유지 관리 방식은 route(7P) 매뉴얼 페이지에 설명된 방식과 유사합니다. 수퍼 유저 또는 동등한 역할을 맡은 사용자만 데이터베이스에 액세스할 수 있습니다.

in.iked 데몬 및 ipseckey 명령은 PF_KEY 소켓 인터페이스를 사용하여 SADB를 유지 관리합니다. SADB가 요청 및 메시지를 처리하는 방법에 대한 자세한 내용은 pf_key(7P) 매뉴얼 페이지를 참조하십시오.

IPsec에서 SA 생성을 위한 유틸리티

IKE 프로토콜은 IPv4 및 IPv6 데이터베이스에 대한 자동 키 관리를 제공합니다. IKE를 설정하는 방법에 대한 지침은 23 장, “IKE 구성(작업)”을 참조하십시오. 수동 키 입력 유틸리티는 ipseckey 명령이며, 이 명령은 ipseckey(1M) 매뉴얼 페이지에 설명되어 있습니다.

ipseckey 명령을 사용하여 SADB(보안 연결 데이터베이스)를 수동으로 채웁니다. 일반적으로 수동 SA 생성은 사정상 IKE를 사용할 수 없을 때 사용됩니다. 하지만 SPI 값이 고유한 경우 수동 SA 생성과 IKE를 동시에 사용할 수 있습니다.

ipseckey 명령은 키가 수동으로 또는 IKE로 추가되었는지 여부에 상관 없이 시스템에 알려진 모든 SA를 보는 데 사용할 수 있습니다. Solaris 10 4/09 릴리스부터는 -c 옵션으로 ipseckey 명령을 실행하면 인수로 제공하는 키 파일의 구문을 검사합니다.

ipseckey 명령으로 추가된 IPsec SA는 시스템을 재부트하면 없어집니다. 현재 릴리스에서 시스템 부트 시 수동으로 추가한 SA를 사용으로 설정하려면 항목을 /etc/inet/secret/ipseckeys 파일에 추가한 다음 svc:/network/ipsec/manual-key:default 서비스를 사용으로 설정합니다. 절차는 485 페이지 “수동으로 IPsec 보안 연관을 만드는 방법”을 참조하십시오.

ipseckey 명령에는 제한된 수의 일반 옵션만 있지만 명령은 풍부한 명령 언어를 지원합니다. 수동 키 입력에 대한 프로그래밍 인터페이스로 해당 요청이 전달되도록 지정할 수 있습니다. 추가 정보는 pf_key(7P) 매뉴얼 페이지를 참조하십시오.

ipseckey에 대한 보안 고려 사항

ipseckey 명령은 수퍼 유저나 Network Security 또는 Network IPsec Management 권한 프로파일을 가진 역할이 민감한 암호화 키 입력 정보를 입력할 수 있도록 합니다. 공격자가 이 정보에 대한 액세스 권한을 획득할 경우 IPsec 트래픽의 보안을 침해할 수 있습니다.

주 – 가능한 경우 ipseckey로 수동 키 입력이 아닌 IKE를 사용합니다.

키 입력 자료를 처리하고 ipseckey 명령을 사용할 때 다음 사항을 고려해야 합니다.

- 키 입력 자료를 새로 고쳤습니까? 정기적인 키 새로 고침은 기본적인 보안 방식입니다. 키 새로 고침은 잠재적인 알고리즘 및 키 취약성으로부터 보호하고 노출된 키의 손상을 제한합니다.
- TTY가 네트워크를 통해 이동합니까? ipseckey 명령이 대화식 모드입니까?
  - 대화식 모드에서는 키 입력 자료의 보안이 이 TTY의 트래픽에 대한 네트워크 경로의 보안입니다. 일반 텍스트 텔넷 또는 rlogin 세션을 통해 ipseckey 명령을 사용하는 것을 피해야 합니다.
  - 로컬 창이라도 창 이벤트를 읽는 숨겨진 프로그램의 공격 대상이 될 수 있습니다.
- -f 옵션을 사용했습니까? 파일이 네트워크를 통해 액세스합니까? 파일을 누구나 읽을 수 있습니까?
  - 공격자는 파일이 읽혀질 때 네트워크 마운트 파일을 읽을 수 있습니다. 키 입력 자료가 포함된 누구나 읽을 있는 파일 사용을 피해야 합니다.
  - 이름 지정 시스템을 보호합니다. 다음 두 조건이 충족될 경우 호스트 이름을 더 이상 신뢰할 수 없습니다.
    - 소스 주소가 네트워크를 통해 조회할 수 있는 호스트입니다.
    - 이름 지정 시스템이 침해되었습니다.

보안 취약성은 실제 도구가 도구의 오용으로 인해 발생하기도 합니다. ipseckey 명령을 사용할 때는 주의해야 합니다. 가장 안전한 작업 모드를 위해서는 콘솔 또는 기타 하드 연결된 TTY를 사용합니다.

다른 유틸리티에 대한 IPsec 확장

ifconfig 명령에는 터널 인터페이스에서 IPsec 정책을 관리하기 위한 옵션이 포함됩니다. snoop 명령은 AH 및 ESP 헤더를 구문 분석할 수 있습니다.

ifconfig 명령 및 IPsec

Solaris 10, Solaris 10 7/05, Solaris 10 1/06 및 Solaris 10 11/06 릴리스: IPsec를 지원하기 위해 ifconfig 명령에서 다음 보안 옵션이 제공됩니다. 이러한 보안 옵션은 Solaris 10 7/07 릴리스의 ipsecconf 명령으로 처리됩니다.

- auth_algs
- encr_auth_algs
- encr_algs

한 번의 호출에서 터널에 대한 모든 IPsec 보안 옵션을 지정해야 합니다. 예를 들어, 트래픽 보호를 위해 ESP만 사용할 경우 다음과 같이 두 보안 옵션이 모두 포함된 `ip.tun0` 터널을 구성해야 합니다.

```
# ifconfig ip.tun0 encr_algs aes encr_auth_algs md5
```

마찬가지로 `ipsecinit.conf` 항목은 다음과 같이 두 보안 옵션이 모두 포함된 터널을 구성합니다.

```
# WAN traffic uses ESP with AES and MD5.
   {} ipsec {encr_algs aes encr_auth_algs md5}
```

auth_algs 보안 옵션

이 옵션은 지정된 인증 알고리즘에서 터널에 대한 IPsec AH를 사용으로 설정합니다. `auth_algs` 옵션의 형식은 다음과 같습니다.

`auth_algs` *authentication-algorithm*

알고리즘에 대해 특정 알고리즘 기본 설정을 표현하지 않으려면 *any* 매개변수를 포함하여 숫자 또는 알고리즘 이름을 지정할 수 있습니다. 터널 보안을 사용 안함으로 설정하려면 다음 옵션을 지정합니다.

`auth_algs none`

사용 가능한 인증 알고리즘 목록을 보려면 `ipsecalgs` 명령을 실행하십시오.

주 – `auth_algs` 옵션은 NAT 순회와 작동하지 않습니다. 자세한 내용은 470 페이지 “IPsec 및 NAT 순회”를 참조하십시오.

---

encr_auth_algs 보안 옵션

이 옵션은 지정된 인증 알고리즘에서 터널에 대한 IPsec ESP를 사용으로 설정합니다. `encr_auth_algs` 옵션의 형식은 다음과 같습니다.

`encr_auth_algs` *authentication-algorithm*

알고리즘에 대해 특정 알고리즘 기본 설정을 표현하지 않으려면 *any* 매개변수를 포함하여 숫자 또는 알고리즘 이름을 지정할 수 있습니다. ESP 암호화 알고리즘을 지정했지만 인증 알고리즘을 지정하지 않을 경우, ESP 인증 알고리즘 값이 기본적으로 *any* 매개변수로 설정됩니다.

사용 가능한 인증 알고리즘 목록을 보려면 `ipsecalgs` 명령을 실행하십시오.

encr_algs 보안 옵션

이 옵션은 지정된 암호화 알고리즘에서 터널에 대한 IPsec ESP를 사용으로 설정합니다. `encr_algs` 옵션의 형식은 다음과 같습니다.

`encr_algs` *encryption-algorithm*

알고리즘에 대해서는 번호 또는 알고리즘 이름을 지정할 수 있습니다. 터널 보안을 사용 안함으로 설정하려면 다음 옵션을 지정합니다.

`encr_algs none`

ESP 인증 알고리즘을 지정했지만 암호화 알고리즘을 지정하지 않을 경우, ESP의 암호화 값이 기본적으로 *null* 매개변수로 설정됩니다.

사용 가능한 암호화 알고리즘 목록을 보려면 `ipsecalgs` 명령을 실행하십시오.

snoop 명령 및 IPsec

`snoop` 명령은 AH 및 ESP 헤더를 구문 분석할 수 있습니다. ESP는 데이터를 암호화하므로 `snoop` 명령은 ESP로 보호된 암호화된 헤더를 볼 수 없습니다. AH는 데이터를 암호화하지 않습니다. 따라서 AH로 보호된 트래픽은 `snoop` 명령으로 검사할 수 있습니다. 명령에 대한 `-V` 옵션은 AH가 패킷에서 언제 사용되었는지 표시합니다. 자세한 내용은 snoop(1M) 매뉴얼 페이지를 참조하십시오.

보호된 패킷에 대한 상세 정보 `snoop` 출력의 예는 490 페이지 “IPsec로 패킷이 보호되는지 확인하는 방법”을 참조하십시오.

◆ ◆ ◆ 22 장

22 Internet Key Exchange(개요)

IKE(Internet Key Exchange)는 IPsec의 키 관리를 자동화합니다. Oracle Solaris는 IKEv1을 구현합니다. 이 장은 IKE에 대한 다음 정보를 포함합니다.

IKE 구현 지침은 23 장, “IKE 구성(작업)”을 참조하십시오. 참고 사항은 24 장, “Internet Key Exchange(참조)”를 참조하십시오. IPsec에 대한 내용은 19 장, “IP 보안 아키텍처(개요)”를 참조하십시오.

IKE의 새로운 기능

Solaris 10 4/09: 이 릴리스부터 SMF(서비스 관리 기능)에서 IKE를 서비스로 관리합니다. 기본적으로 `svc:/network/ipsec/ike:default` 서비스는 사용 안함으로 설정됩니다. 또한 이 릴리스에서는 IPsec 및 IKE 관리를 위해 Network IPsec Management 권한 프로파일이 제공됩니다.

Solaris 10 7/07: 이 릴리스부터 IKE는 AES 알고리즘을 사용할 수 있으며 비전역 영역에서 사용하도록 전역 영역에서 구성할 수 있습니다.

- `SO_ALLZONES` 소켓 옵션을 사용으로 설정하면 IKE는 트래픽을 비전역 영역에서 처리할 수 있습니다.
- 새로운 Oracle Solaris 기능에 대한 전체 목록 및 Solaris 릴리스에 대한 설명은 **Oracle Solaris 10 1/13 새로운 기능**을 참조하십시오.

IKE로 키 관리

IPsec 보안 연관(SA)에 대한 키 입력 자료를 관리하는 것을 **키 관리**라고 합니다. 자동 키 관리를 위해서는 키 생성, 인증, 교환을 위한 통신 보안 채널이 필요합니다. Oracle Solaris는 IKE(Internet Key Exchange) 버전 1을 사용하여 키 관리를 자동화합니다. IKE는 대용량 트래픽에 보안 채널을 제공하도록 쉽게 확장됩니다. IPv4 및 IPv6 패킷의 IPsec SA는 IKE를 활용할 수 있습니다.

IKE는 사용 가능한 하드웨어 가속 및 하드웨어 저장소를 활용할 수 있습니다. 하드웨어 가속기를 사용하여 집중적인 키 작업을 시스템에서 처리할 수 있습니다. 하드웨어의 키 저장소는 추가적 보호 계층을 제공합니다.

IKE 키 협상

IKE 데몬 `in.iked`는 IPsec SA에 대한 키 입력 자료를 안전한 방식으로 협상하고 인증합니다. 데몬은 OS에서 제공된 내부 함수에서 키의 무작위 시드를 사용합니다. IKE는 PFS(완전 순방향 비밀성)를 제공합니다. PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송 키를 만드는 데 사용된 시드는 재사용되지 않습니다. `in.iked(1M)` 매뉴얼 페이지를 참조하십시오.

IKE 키 용어

다음 표는 키 협상에 사용되는 용어를 나열하고 흔히 사용되는 머리글자어를 제공하며 각 용어에 대한 정의 및 사용을 제시합니다.

표 22–1 키 협상 용어, 머리글자어 및 사용

키 협상 용어	머리글자어	정의 및 사용
키 교환		비대칭 암호화 알고리즘에 대한 키를 생성하는 프로세스입니다. 두 가지 주요 방법은 RSA 및 Diffie-Hellman 프로토콜입니다.
Diffie-Hellman 알고리즘	DH	키 생성 및 키 인증을 제공하는 키 교환 알고리즘입니다. **인증된 키 교환**이라고도 합니다.
RSA 알고리즘	RSA	키 생성 및 키 전송을 제공하는 키 교환 알고리즘입니다. 프로토콜 이름은 Rivest, Shamir, Adleman 등 3인의 저작자 이름에서 따왔습니다.
완전 순방향 비밀성	PFS	인증된 키 교환에만 적용됩니다. PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송을 보호하는 키의 소스도 추가 키를 파생하는 데 사용되지 않습니다.

표 22–1 키 협상 용어, 머리글자어 및 사용 (계속)

키 협상 용어	머리글자어	정의 및 사용
Oakley 그룹		안전한 방식으로 Phase 2의 키를 설정하는 방법입니다. Oakley 그룹은 PFS를 협상하는 데 사용됩니다. The Internet Key Exchange (IKE) (http://www.faqs.org/rfcs/rfc2409.html)의 6절을 참조하십시오.

IKE Phase 1 교환

Phase 1 교환을 **기본 모드**라고 합니다. Phase 1 교환에서 IKE는 공개 키 암호화 방법을 사용하여 피어 IKE 엔티티로 자체 인증합니다. 그 결과는 ISAKMP(Internet Security Association and Key Management Protocol) 보안 연관(SA)입니다. ISAKMP SA는 IP 데이터그램에 대한 키 입력 자료를 협상하기 위한 IKE의 보안 채널입니다. IPsec SA와 달리, ISAKMP SA는 양방향이므로 하나의 보안 연관만 필요합니다.

IKE가 Phase 1 교환에서 키 입력 자료를 협상하는 방법을 구성할 수 있습니다. IKE는 `/etc/inet/ike/config` 파일에서 구성 정보를 읽습니다. 구성 정보는 다음과 같습니다.

- 공개 키 인증서 이름과 같은 전역 매개변수
- PFS(완전 순방향 비밀성)의 사용 여부
- 영향을 받는 인터페이스
- 보안 프로토콜 및 해당 알고리즘
- 인증 방법

두 가지 인증 방법은 미리 공유한 키와 공개 키 인증서입니다. 공개 키 인증서는 자체 서명할 수 있습니다. 또는 공개 키 기반구조(PKI) 조직에서 CA(인증 기관)에 의해 인증서를 발행할 수 있습니다.

IKE Phase 2 교환

Phase 2 교환을 **빠른 모드**라고 합니다. Phase 2 교환에서 IKE는 IKE 데몬을 실행 중인 시스템 간에 IPsec SA를 만들고 관리합니다. IKE는 Phase 1 교환에서 만든 보안 채널을 사용하여 키 입력 자료의 전송을 보호합니다. IKE 데몬은 `/dev/random` 장치를 사용하여 난수 생성기로부터 키를 만듭니다. 데몬이 구성 가능한 비율로 키를 새로 고칩니다. IPsec 정책용 구성 파일인 `ipsecinit.conf`에 지정된 알고리즘에서 키 입력 자료를 사용할 수 있습니다.

IKE 구성 선택

/etc/inet/ike/config 구성 파일은 IKE 정책 항목을 포함합니다. 두 IKE 데몬이 서로 인증하려면 항목이 유효해야 합니다. 또한 키 입력 자료를 사용할 수 있어야 합니다. 구성 파일의 항목에 따라 키 입력 자료를 사용하여 Phase 1 교환을 인증하는 방법이 결정됩니다. 미리 공유한 키 또는 공개 키 인증서를 선택할 수 있습니다.

auth_method preshared 항목은 미리 공유한 키가 사용됨을 나타냅니다. preshared가 아닌 auth_method의 값은 공개 키 인증서가 사용될지 나타냅니다. 공개 키 인증서를 자체 서명할 수도 있고, PKI 조직에서 인증서를 설치할 수도 있습니다. 자세한 내용은 ike.config(4) 매뉴얼 페이지를 참조하십시오.

IKE와 미리 공유한 키 인증

미리 공유한 키는 두 개 이상의 피어 시스템을 인증하는 데 사용됩니다. 미리 공유한 키는 한 시스템에서 관리자가 만든 16진수 또는 ASCII 문자열입니다. 그런 다음 대역 외 연결에서 피어 시스템의 관리자와 키를 공유합니다. 악의적 사용자가 미리 공유한 키를 가로채면 피어 시스템 중 하나로 가장할 수 있습니다.

이 인증 방법을 사용하는 피어에서 미리 공유한 키는 동일해야 합니다. 키는 특정 IP 주소와 연결되어 있습니다. 각 시스템의 /etc/inet/secret/ike.preshared 파일에 키가 저장됩니다. ike.preshared 파일은 IKE용이고 ipseckeys 파일은 IPsec용입니다. ike.preshared 파일의 키가 손상되면 모든 전송에 영향을 줍니다. 키는 한 관리자가 통신 시스템을 제어할 때 가장 안전합니다. 자세한 내용은 ike.preshared(4) 매뉴얼 페이지를 참조하십시오.

IKE와 공개 키 인증서

공개 키 인증서를 사용하면 통신 시스템이 대역 외에서 보안 키 입력 자료를 공유할 필요가 없습니다. 공개 키는 키 인증 및 협상을 위해 Diffie-Hellman 알고리즘(DH)을 사용합니다. 공개 키 인증서는 두 종류로 나뉩니다. 인증서를 자체 서명할 수도 있고, CA(인증 기관)에서 인증서를 공인할 수도 있습니다.

자체 서명된 공개 키 인증서는 관리자 스스로 만듭니다. ikecert certlocal -ks 명령은 시스템의 공개-개인 키 쌍 중 개인 부분을 만듭니다. 그런 다음 원격 시스템에서 X.509 형식의 자체 서명된 인증서 출력을 가져옵니다. 키 쌍의 공개 부분을 위해 원격 시스템의 인증서가 ikecert certdb 명령에 입력됩니다. 자체 서명된 인증서는 통신 시스템의 /etc/inet/ike/publickeys 디렉토리에 상주합니다. -T 옵션을 사용하면 인증서가 연결된 하드웨어에 상주합니다.

자체 서명된 인증서는 미리 공유한 키와 CA 사이의 중간 지점입니다. 미리 공유한 키와 달리, 자체 서명된 인증서는 모바일 시스템이나 번호를 다시 매길 수 있는 시스템에서 사용할 수 있습니다. 고정 번호 없이 시스템에 인증서를 자체 서명하려면 DNS(www.example.org) 또는 email(root@domain.org) 대체 이름을 사용하십시오.

PKI 또는 CA 조직에서 공개 키를 전달할 수 있습니다. /etc/inet/ike/publickeys 디렉토리에 공개 키와 동반 CA를 설치합니다. -T 옵션을 사용하면 인증서가 연결된 하드웨어에 상주합니다. 또한 공급업체가 CRL(인증서 해지 목록)을 발행합니다. 관리자는 키 및 CA 설치와 함께 /etc/inet/ike/crls 디렉토리에 CRL을 설치할 책임이 있습니다.

CA는 사이트 관리자가 아닌 외부 조직에서 공인된다는 장점이 있습니다. 어떤 의미에서 CA는 공증된 인증서입니다. 자체 서명된 인증서와 마찬가지로, CA는 모바일 시스템이나 번호를 다시 매길 수 있는 시스템에서 사용할 수 있습니다. 자체 서명된 인증서와 달리, CA는 많은 수의 통신 시스템을 보호하도록 매우 쉽게 확장할 수 있습니다.

IKE 및 하드웨어 가속

IKE 알고리즘은 계산하는 데 비용이 많이 들며 Phase 1 교환 시 더욱 그렇습니다. 많은 교환을 처리하는 시스템은 Sun Crypto Accelerator 1000 또는 Sun Crypto Accelerator 6000 보드를 사용하여 공개 키 작업을 처리합니다. Sun Crypto Accelerator 6000 및 Sun Crypto Accelerator 4000 보드를 사용하여 비용이 많이 드는 Phase 1 계산을 처리할 수도 있습니다.

IKE 계산을 가속기 보드로 오프로드하도록 IKE를 구성하는 방법에 대한 자세한 내용은 582 페이지 “Sun Crypto Accelerator 4000 보드를 찾도록 IKE를 구성하는 방법”을 참조하십시오. 키를 저장하는 방법에 대한 자세한 내용은 582 페이지 “Sun Crypto Accelerator 4000 보드를 찾도록 IKE를 구성하는 방법” 및 cryptoadm(1M) 매뉴얼 페이지를 참조하십시오.

IKE 및 하드웨어 저장소

<공개 키 인증서, 개인 키 및 공개 키는 Sun Crypto Accelerator 6000 또는 Sun Crypto Accelerator 4000 보드에 저장할 수 있습니다. RSA 암호화의 경우 이러한 보드는 2048비트까지 키를 지원합니다. DSA 암호화의 경우 보드는 1024비트까지 키를 지원합니다. Sun Crypto Accelerator 6000 보드는 SHA-512 및 ECC 알고리즘을 지원합니다.

보드에 액세스하도록 IKE를 구성하는 방법에 대한 자세한 내용은 582 페이지 “Sun Crypto Accelerator 4000 보드를 찾도록 IKE를 구성하는 방법”을 참조하십시오. 보드에 인증서 및 공개 키를 추가하는 방법에 대한 자세한 내용은 567 페이지 “공개 키 인증서를 생성하여 하드웨어에 저장하는 방법”을 참조하십시오.

IKE 유틸리티 및 파일

다음 표는 IKE 정책의 구성 파일, IKE 키의 저장소 위치 및 IKE를 구현하는 다양한 명령과 서비스를 요약합니다. 서비스에 대한 자세한 내용은 **Oracle Solaris 관리: 기본 관리**의 18 장, "서비스 관리(개요)"를 참조하십시오.

표 22–2 IKE 구성 파일, 키 저장소 위치, 명령 및 서비스

파일, 위치, 명령 또는 서비스	설명	매뉴얼 페이지
`svc:/network/ipsec/ike`	현재 릴리스에서는 IKE를 관리하는 SMF 서비스입니다.	`smf(5)`
`/usr/lib/inet/in.iked`	IKE(Internet Key Exchange) 데몬입니다. 자동 키 관리를 활성화합니다. 현재 릴리스에서는 `ike` 서비스가 이 데몬을 사용으로 설정합니다. 이전 릴리스에서는 `in.iked` 명령이 사용되었습니다.	`in.iked(1M)`
`/usr/sbin/ikeadm`	IKE 정책을 보고 수정하기 위한 IKE 관리 명령입니다.	`ikeadm(1M)`
`/usr/sbin/ikecert`	공개 키 인증서를 보유하는 로컬 데이터베이스를 조작하기 위한 인증서 데이터베이스 관리 명령입니다. 데이터베이스를 연결된 하드웨어에 저장할 수도 있습니다.	`ikecert(1M)`
`/etc/inet/ike/config`	IKE 정책의 기본 구성 파일입니다. 인바운드 IKE 요청을 일치시키고 아웃바운드 IKE 요청을 준비하기 위한 사이트 규칙을 포함합니다. 현재 릴리스에서는 이 파일이 있으면 `ike` 서비스가 사용으로 설정될 때 `in.iked` 데몬이 시작됩니다. 이 파일의 위치는 `svccfg` 명령으로 변경할 수 있습니다.	`ike.config(4)`
`ike.preshared`	`/etc/inet/secret` 디렉토리의 미리 공유한 키 파일입니다. Phase 1 교환에서 인증을 위한 보안 키 입력 자료를 포함합니다. 미리 공유한 키로 IKE를 구성할 때 사용됩니다.	`ike.preshared(4)`
`ike.privatekeys`	`/etc/inet/secret` 디렉토리의 개인 키 디렉토리입니다. 공개-개인 키 쌍의 일부인 개인 키를 포함합니다.	`ikecert(1M)`
`publickeys` 디렉토리	공개 키 및 인증서 파일을 보유하는 `/etc/inet/ike` 디렉토리 안의 디렉토리입니다. 공개-개인 키 쌍 중 공개 키 부분을 포함합니다.	`ikecert(1M)`
`crls` 디렉토리	공개 키 및 인증서 파일에 대한 해지 목록을 보유하는 `/etc/inet/ike` 디렉토리 안의 디렉토리입니다.	`ikecert(1M)`
Sun Crypto Accelerator 1000 보드	운영 체제에서 작업 부담을 덜어서 공개 키 작업을 가속화하는 하드웨어입니다.	`ikecert(1M)`
Sun Crypto Accelerator 4000 보드	운영 체제에서 작업 부담을 덜어서 공개 키 작업을 가속화하는 하드웨어입니다. 또한 공개 키, 개인 키 및 공개 키 인증서를 저장합니다. Sun Crypto Accelerator 6000 보드는 레벨 3의 FIPS 140-2 공인 장치입니다.	`ikecert(1M)`

Oracle Solaris 10 릴리스의 IKE 변경 사항

Solaris 9 릴리스부터 IKE에는 다음 기능이 포함됩니다.

- IKE를 사용하여 IPv6 네트워크를 통한 IPsec의 키 교환을 자동화합니다. 자세한 내용은 538 페이지 "IKE로 키 관리"를 참조하십시오.

  주 – IKE는 비전역 영역의 IPsec 키를 관리하는 데 사용할 수 없습니다.

- IKE의 공개 키 작업은 Sun Crypto Accelerator 1000 보드 또는 Sun Crypto Accelerator 4000 보드에 의해 가속화될 수 있습니다. 작업이 보드로 오프로드됩니다. 오프로드는 암호화를 가속화하므로 운영 체제 리소스의 수요가 감소됩니다. 자세한 내용은 541 페이지 "IKE 및 하드웨어 가속"를 참조하십시오. 절차는 581 페이지 "연결된 하드웨어를 찾도록 IKE 구성"을 참조하십시오.
- 공개 키 인증서, 개인 키 및 공개 키는 Sun Crypto Accelerator 4000 보드에 저장할 수 있습니다. 키 저장소에 대한 자세한 내용은 541 페이지 "IKE 및 하드웨어 저장소"를 참조하십시오.
- IKE를 사용하여 NAT 박스 뒤에서 IPsec 키 교환을 자동화할 수 있습니다. 그러나 NAT를 순회하는 IPsec ESP 키는 하드웨어를 통해 가속화할 수 없습니다. 자세한 내용은 470 페이지 "IPsec 및 NAT 순회"를 참조하십시오. 절차는 573 페이지 "모바일 시스템에 대한 IKE 구성(작업 맵)"을 참조하십시오.
- 재전송 매개변수 및 패킷 시간 초과 매개변수가 `/etc/inet/ike/config` 파일에 추가되었습니다. 이러한 매개변수는 IKE Phase 1(주 모드) 협상을 조정하여 네트워크 간섭, 과도한 네트워크 트래픽 및 IKE 프로토콜이 다르게 구현된 플랫폼과의 상호 운용을 처리합니다. 매개변수에 대한 자세한 내용은 `ike.config(4)` 매뉴얼 페이지를 참조하십시오. 절차는 584 페이지 "IKE 전송 매개변수 변경(작업 맵)"을 참조하십시오.

◆ ◆ ◆ 23 장

23 IKE 구성(작업)

이 장에서는 시스템의 인터넷 키 교환(IKE) 구성 방법에 대해 설명합니다. IKE가 구성되면 네트워크의 IPsec에 대한 키 입력 도구가 자동으로 생성됩니다.

이 장은 다음 정보를 포함합니다.

IKE에 대한 개요 정보는 22 장, “Internet Key Exchange(개요)”를 참조하십시오. IKE에 대한 참조 정보는 24 장, “Internet Key Exchange(참조)”를 참조하십시오. 자세한 절차는 `ikeadm`(1M), `ikecert`(1M) 및 `ike.config`(4) 매뉴얼 페이지의 Examples 절을 참조하십시오.

IKE 구성(작업 맵)

미리 공유한 키, 자체 서명된 인증서 및 인증 기관(CA)의 인증서를 사용하여 IKE를 인증할 수 있습니다. 규칙은 보호되고 있는 끝점에 특정 IKE 인증 방법을 연결합니다. 따라서 시스템에서 IKE 인증 방법 중 하나 또는 전체를 사용할 수 있습니다. PKCS #11 라이브러리에 대한 포인터를 통해 인증서는 연결된 하드웨어 가속기를 사용으로 설정할 수 있습니다.

IKE를 구성한 후에는 IKE 구성을 사용하는 IPsec 작업을 완료합니다. 다음 표에서는 특정 IKE 구성을 중점적으로 다루는 작업 맵에 대해 설명합니다.

작업	설명	수행 방법
미리 공유한 키로 IKE를 구성합니다.	보안 키를 공유하는 시스템 간의 통신을 보호합니다.	546 페이지 “미리 공유한 키로 IKE 구성(작업 맵)”
공개 키 인증서로 IKE를 구성합니다.	공개 키 인증서로 통신을 보호합니다. 인증서는 자체 서명될 수도 있고, PKI 조직에 의해 보장될 수도 있습니다.	556 페이지 “공개 키 인증서로 IKE 구성(작업 맵)”
NAT 경계를 벗어납니다.	모바일 시스템과 통신하도록 IPsec 및 IKE를 구성합니다.	573 페이지 “모바일 시스템에 대한 IKE 구성(작업 맵)”
IKE를 구성하여 연결된 하드웨어에 공개 키 인증서를 생성 및 저장합니다.	Sun Crypto Accelerator 1000 보드 또는 Sun Crypto Accelerator 4000 보드를 사용으로 설정하면 IKE 작업 속도를 향상시킬 수 있습니다. 또한 Sun Crypto Accelerator 4000 보드를 사용으로 설정하면 공개 키 인증서를 저장할 수 있습니다.	581 페이지 “연결된 하드웨어를 찾도록 IKE 구성”
Phase 1 키 협상 매개변수를 조정합니다.	IKE 키 협상 타이밍을 변경합니다.	584 페이지 “IKE 전송 매개변수 변경(작업 맵)”

미리 공유한 키로 IKE 구성(작업 맵)

다음 표에서는 미리 공유한 키로 IKE를 구성 및 유지 관리하는 절차에 대해 설명합니다.

작업	설명	수행 방법
미리 공유한 키로 IKE를 구성합니다.	IKE 정책 파일과 공유할 키 하나를 만듭니다.	547 페이지 “미리 공유한 키로 IKE를 구성하는 방법”
실행 중인 IKE 시스템에서 미리 공유한 키를 새로 고칩니다.	통신 시스템에서 IKE에 대한 새로운 키 입력 자료를 추가합니다.	550 페이지 “IKE 미리 공유한 키를 새로 고치는 방법”
실행 중인 IKE 시스템에 미리 공유한 키를 추가합니다.	현재 IKE 정책을 적용 중인 시스템에 새 IKE 정책 항목 및 새 키 입력 도구를 추가합니다.	552 페이지 “`ipsecinit.conf`으로 새 정책 항목에 대해 IKE 미리 공유한 키를 추가하는 방법”
미리 공유한 키가 동일한지 확인합니다.	두 시스템에서 미리 공유한 키를 표시하여 키가 동일한지 확인합니다.	555 페이지 “IKE 미리 공유한 키가 동일한지 확인하는 방법”

미리 공유한 키로 IKE 구성

미리 공유한 키는 가장 간단한 IKE 인증 방법입니다. IKE를 사용하도록 두 시스템을 구성 중이며 두 시스템의 관리자라면 미리 공유한 키를 사용하는 것이 좋습니다. 단, 공개 키 인증서와 달리 미리 공유한 키는 특정 IP 주소와 연관되어 있습니다. 미리 공유한 키는 모바일 시스템 또는 번호가 재지정될 수 있는 시스템에서 사용할 수 없습니다.

▼ 미리 공유한 키로 IKE를 구성하는 방법

IKE 구현은 키 길이가 다양한 알고리즘을 제공합니다. 키 길이는 사이트 보안에 따라 선택할 수 있습니다. 일반적으로 길이가 긴 키는 길이가 짧은 키에 비해 더 강력한 보안을 제공합니다.

이 절차에서는 `enigma` 및 `partym` 시스템 이름을 사용합니다. `enigma` 및 `partym` 이름을 사용자의 현재 시스템 이름으로 대체하십시오.

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 **각 시스템에서 `/etc/inet/ike/config.sample` 파일을 `/etc/inet/ike/config` 파일에 복사합니다.**

3 **각 시스템의 `ike/config` 파일에 규칙 및 전역 매개변수를 입력합니다.**

이 파일의 규칙 및 전역 매개변수는 시스템의 `ipsecinit.conf` 파일에 설정되어 있는 IPsec 정책이 성공하도록 허용해야 합니다. 다음 `ike/config` 예는 477 페이지 “IPsec를 사용하여 두 시스템 사이의 트래픽을 보호하는 방법”의 `ipsecinit.conf` 예와 함께 작동합니다.

a. **예를 들어, `enigma` 시스템에서 `/etc/inet/ike/config` 파일을 수정합니다.**

```
### ike/config file on enigma, 192.168.116.16

## Global parameters
#
## Phase 1 transform defaults
p1_lifetime_secs 14400
p1_nonce_len 40
#
## Defaults that individual rules can override.
```

```
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2
#
## The rule to communicate with partym
#  Label must be unique
{ label "enigma-partym"
  local_addr 192.168.116.16
  remote_addr 192.168.13.213
  p1_xform
    { auth_method preshared oakley_group 5 auth_alg sha1 encr_alg aes }
  p2_pfs 5
}
```

b. partym 시스템에서 /etc/inet/ike/config 파일을 수정합니다.

```
### ike/config file on partym, 192.168.13.213
## Global Parameters
#
p1_lifetime_secs 14400
p1_nonce_len 40
#
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 2

## The rule to communicate with enigma
#  Label must be unique
{ label "partym-enigma"
  local_addr 192.168.13.213
  remote_addr 192.168.116.16
p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha1 encr_alg aes }
p2_pfs 5
}
```

4 각 시스템에서 파일의 구문을 확인합니다.

```
# /usr/lib/inet/in.iked -c -f /etc/inet/ike/config
```

5 키 입력 자료로 사용할 난수를 생성합니다.

사이트에 난수 생성기가 있는 경우 생성기를 사용합니다. Oracle Solaris 10 시스템에서 od 명령을 사용할 수 있습니다. 예를 들어, 다음 명령은 두 라인의 16진수를 출력합니다.

```
% od -X -A n /dev/random | head -2
         f47cb0f4 32e14480 951095f8 2b735ba8
         0a9467d0 8f92c880 68b6a40e 0efe067d
```

od 명령에 대한 설명은 484 페이지 “Oracle Solaris 시스템에서 난수를 생성하는 방법” 및 od(1) 매뉴얼 페이지를 참조하십시오.

주 – 다른 운영 체제에서는 ASCII 키 입력 자료가 필요할 수 있습니다. 16진수와 ASCII 형식으로 동일한 키를 생성하려면 예 23–1을 참조하십시오.

---

6 **단계 5 출력에서 키 하나를 생성합니다.**

```
f47cb0f432e14480951095f82b735ba80a9467d08f92c88068b6a40e
```

이 절차의 인증 알고리즘은 단계 3과 같이 SHA–1입니다. 해시 크기, 즉 인증 알고리즘의 출력 크기에 따라 미리 공유한 키의 최소 권장 크기가 결정됩니다. SHA–1 알고리즘의 출력은 160비트 또는 40자입니다. 예에서의 키 길이는 56자이며 IKE에서 사용할 추가 키 입력 자료를 제공합니다.

7 **각 시스템에서 `/etc/inet/secret/ike.preshared` 파일을 만듭니다.**

각 파일에 미리 공유한 키를 삽입합니다.

a. 예를 들어, `enigma` 시스템에서 `ike.preshared` 파일이 다음과 유사하게 표시됩니다.

```
# ike.preshared on enigma, 192.168.116.16
#...
{ localidtype IP
    localid 192.168.116.16
    remoteidtype IP
    remoteid 192.168.13.213
    # enigma and partym's shared key in hex (192 bits)
    key f47cb0f432e14480951095f82b735ba80a9467d08f92c88068b6a40e
    }
```

b. `partym` 시스템에서 `ike.preshared` 파일이 다음과 유사하게 표시됩니다.

```
# ike.preshared on partym, 192.168.13.213
#...
{ localidtype IP
    localid 192.168.13.213
    remoteidtype IP
    remoteid 192.168.116.16
    # partym and enigma's shared key in hex (192 bits)
    key f47cb0f432e14480951095f82b735ba80a9467d08f92c88068b6a40e
    }
```

주 – 각 시스템의 미리 공유한 키는 동일해야 합니다.

예 23–1 운영 체제가 다른 두 시스템에 대해 동일한 키 입력 자료 생성

Oracle Solaris의 IPsec 기능은 다른 운영 체제의 IPsec과 상호 운용됩니다. 시스템이 ASCII 미리 공유한 키가 필요한 시스템과 통신하는 경우 16진수와 ASCII 두 가지 형식으로 된 키 하나를 생성해야 합니다.

이 예에서 Oracle Solaris 시스템 관리자는 56자의 키 입력 자료가 필요합니다. 관리자는 다음 명령을 사용하여 ASCII 문장암호에서 16진수 키를 생성합니다. `-tx1` 옵션은 모든 Oracle Solaris 시스템에서 한 번에 하나씩 바이트를 출력합니다.

```
# /bin/echo "papiermache with cashews and\c" | od -tx1 | cut -c 8-55 | \
tr -d '\n' | tr -d ' ' | awk '{print}'
7061706965726d6163686520776974682063617368657773206 16e64
```

오프셋을 제거하고 16진수 출력을 연결하면 Oracle Solaris 시스템의 16진수 키는 `7061706965726d6163686520776974682063617368657773206 16e64`입니다. 관리자는 이 값을 Oracle Solaris 시스템의 `ike.preshared` 파일에 지정합니다.

```
# Shared key in hex (192 bits)
key 7061706965726d6163686520776974682063617368657773206 16e64
```

ASCII 미리 공유한 키가 필요한 시스템에서 문장암호는 미리 공유한 키입니다. Oracle Solaris 시스템 관리자는 문장암호 `papiermache with cashews and`를 소유한 다른 관리자에게 전화를 겁니다.

▼ IKE 미리 공유한 키를 새로 고치는 방법

이 절차에서는 미리 공유된 기존 키를 사용자가 정기적으로 바꾸려고 한다고 가정합니다.

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 **난수를 생성하고 적절한 길이의 키를 구성합니다.**

자세한 내용은 484 페이지 “Oracle Solaris 시스템에서 난수를 생성하는 방법”을 참조하십시오. Oracle Solaris용 미리 공유한 키를 생성하는 경우 ASCII가 필요한 운영 체제와 통신하는 시스템에 대해서는 예 23–1을 참조하십시오.

3 **현재 키를 새 키로 바꿉니다.**

예를 들어, `enigma` 및 `partym` 호스트에서 `/etc/inet/secret/ike.preshared` 파일의 `key` 값을 같은 길이의 새 숫자로 바꿉니다.

4 **새 키를 커널로 읽습니다.**

- **Solaris 10 4/09 릴리스부터는 `ike` 서비스를 다시 시작합니다.**

  ```
  # svcadm enable ike
  ```

- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우 in.iked 데몬을 강제 종료한 후 다시 시작합니다.**

  a. **in.iked 데몬의 권한 레벨을 확인합니다.**

  ```
  # /usr/sbin/ikeadm get priv
  Current privilege level is 0x0, base privileges enabled
  ```

  명령이 0x1 또는 0x2 권한 레벨을 반환한 경우 키 입력 자료를 변경할 수 있습니다. 레벨이 0x0인 경우에는 키 입력 자료를 수정하거나 볼 수 없습니다. 기본적으로 in.iked 데몬은 0x0 권한 레벨에서 실행됩니다.

  b. **권한 레벨이 0x0인 경우 데몬을 강제 종료한 후 다시 시작합니다.**

  데몬을 다시 시작하면 새 버전의 ike.preshared 파일을 읽습니다.

  ```
  # pkill in.iked
  # /usr/lib/inet/in.iked
  ```

  c. **권한 레벨이 0x1 또는 0x2인 경우에는 새 버전의 ike.preshared 파일에서 읽습니다.**

  ```
  # ikeadm read preshared
  ```

▼ IKE 미리 공유한 키를 보는 방법

기본적으로 ikeadm 명령은 Phase 1 SA의 덤프에서 실제 키를 보지 못하도록 합니다. 디버깅 중에는 키를 보는 것이 유용합니다.

실제 키를 보려면 데몬의 권한 레벨을 향상시켜야 합니다. 권한 레벨에 대한 설명은 591 페이지 “ikeadm 명령”을 참조하십시오.

주 – Solaris 10 4/09 이전 릴리스에서 이 절차를 수행하려면 예 23–2를 참조하십시오.

시작하기 전에 IKE가 구성되고 ike 서비스가 실행 중입니다.

1. **IKE 미리 공유한 키를 봅니다.**

```
# ikeadm
ikeadm> dump preshared
```

2. **오류가 발생하면 in.iked 데몬의 권한 레벨을 향상시킵니다.**

  a. **SMF 저장소에서 in.iked 데몬의 권한 레벨을 향상시킵니다.**

  ```
  # svcprop -p config/admin_privilege ike
  base
  # svccfg -s ike setprop config/admin_privilege=keymat
  ```

  b. **실행 중인 in.iked 데몬의 권한 레벨을 향상시킵니다.**

  ```
  # svcadm refresh ike ; svcadm restart ike
  ```

c. **(옵션) 권한 레벨이 keymat인지 확인합니다.**

```
# svcprop -p config/admin_privilege ike
keymat
```

d. **단계 1을 다시 실행하여 키를 봅니다.**

3 **IKE 데몬을 기본 권한 레벨로 되돌립니다.**

a. **키를 본 다음 권한 레벨을 기본값으로 되돌립니다.**

```
# svccfg -s ike setprop config/admin_privilege=base
```

b. **IKE를 새로 고친 후 다시 시작합니다.**

```
# svcadm refresh ike ; svcadm restart ike
```

예 23–2 Solaris 10 4/09 이전 릴리스에서 IKE 미리 공유한 키 확인

다음 예에서 관리자는 현재 Oracle Solaris 10 릴리스에서 실행되고 있지 않은 Solaris 시스템의 키를 보고 있습니다. 관리자는 이 시스템의 키가 통신 시스템의 키와 동일한지 확인하려고 합니다. 두 시스템의 키가 동일한지 확인한 후 관리자는 권한 레벨을 0으로 복원합니다.

- 먼저 관리자는 `in.iked` 데몬의 권한 레벨을 확인합니다.

```
adm1 # /usr/sbin/ikeadm get priv
Current privilege level is 0x0, base privileges enabled
```

- 권한 레벨이 0x1 또는 0x2가 아니므로 관리자는 `in.iked` 데몬을 중지한 후 권한 레벨을 2로 향상시킵니다.

```
adm1 # pkill in.iked
adm1 # /usr/lib/inet/in.iked -p 2
Setting privilege level to 2
```

- 관리자가 키를 표시합니다.

```
adm1 # ikeadm dump preshared
PSKEY: Preshared key (24 bytes): f47cb.../192
LOCIP: AF_INET: port 0, 192.168.116.16 (adm1).
REMIP: AF_INET: port 0, 192.168.13.213 (com1).
```

- 관리자는 원격으로 통신 시스템에 로그인하여 키가 동일한지 확인합니다.
- 그런 다음 기본 권한 레벨을 복원합니다.

```
# ikeadm set priv base
```

▼ ipsecinit.conf으로 새 정책 항목에 대해 IKE 미리 공유한 키를 추가하는 방법

같은 피어 간의 작업 구성에 IPsec 정책 항목을 추가할 경우에는 IPsec 정책 서비스를 새로 고쳐야 합니다. IKE는 재구성하거나 다시 시작하지 않아도 됩니다.

IPsec 정책에 새 피어를 추가할 경우 IPsec 변경 외에 IKE 구성도 수정해야 합니다.

주 – Solaris 10 4/09 이전 릴리스에서 이 절차를 수행하려면 예 23–3을 참조하십시오.

시작하기 전에 `ipsecinit.conf` 파일을 업데이트했으며 피어 시스템에 대한 IPsec 정책을 새로 고쳤습니다.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 이 시스템에서 난수를 생성하고 64-448비트의 키를 구성합니다.

자세한 내용은 484 페이지 “Oracle Solaris 시스템에서 난수를 생성하는 방법”을 참조하십시오. Oracle Solaris용 미리 공유한 키를 생성하는 경우 ASCII가 필요한 운영 체제와 통신하는 시스템에 대해서는 예 23–1을 참조하십시오.

3 원격 시스템의 관리자에게 키를 전송합니다.

두 사람 모두 동일한 미리 공유한 키를 동시에 추가해야 합니다. 키의 보안은 전송 방식의 보안에 따라 달라집니다. 등록된 메일 또는 보호된 팩스와 같은 대역 외 방식이 가장 좋습니다. 또한 `ssh` 세션을 사용하여 두 시스템을 관리할 수도 있습니다.

4 `enigma` 및 새 피어인 `ada`의 키를 관리하는 IKE용 규칙을 만듭니다.

a. `enigma` 시스템에서 `/etc/inet/ike/config` 파일에 다음 규칙을 추가합니다.

```
### ike/config file on enigma, 192.168.116.16

## The rule to communicate with ada

{label "enigma-to-ada"
 local_addr 192.168.116.16
 remote_addr 192.168.15.7
 p1_xform
 {auth_method preshared oakley_group 5 auth_alg sha1 encr_alg blowfish}
 p2_pfs 5
    }
```

b. `ada` 시스템에서 다음 규칙을 추가합니다.

```
### ike/config file on ada, 192.168.15.7

## The rule to communicate with enigma
```

```
{label "ada-to-enigma"
 local_addr 192.168.15.7
 remote_addr 192.168.116.16
 p1_xform
 {auth_method preshared oakley_group 5 auth_alg sha1 encr_alg blowfish}
 p2_pfs 5
}
```

5 재부트 시 IKE 미리 공유한 키를 사용할 수 있는지 확인합니다.

a. enigma 시스템에서 /etc/inet/secret/ike.preshared 파일에 다음 정보를 추가합니다.

```
# ike.preshared on enigma for the ada interface
#
{ localidtype IP
  localid 192.168.116.16
  remoteidtype IP
  remoteid 192.168.15.7
  # enigma and ada's shared key in hex (32 - 448 bits required)
  key 8d1fb4ee500e2bea071deb2e781cb48374411af5a9671714672bb1749ad9364d
}
```

b. ada 시스템에서 ike.preshared 파일에 다음 정보를 추가합니다.

```
# ike.preshared on ada for the enigma interface
#
{ localidtype IP
  localid 192.168.15.7
  remoteidtype IP
  remoteid 192.168.116.16
  # ada and enigma's shared key in hex (32 - 448 bits required)
  key 8d1fb4ee500e2bea071deb2e781cb48374411af5a9671714672bb1749ad9364d
}
```

6 각 시스템에서 ike 서비스를 새로 고칩니다.

```
# svcadm refresh ike
```

7 시스템에서 통신할 수 있는지 확인합니다.

555 페이지 "IKE 미리 공유한 키가 동일한지 확인하는 방법"을 참조하십시오.

예 23–3 새 IPsec 정책 항목에 대해 IKE 미리 공유한 키 추가

다음 예에서 관리자는 현재 Oracle Solaris 10 릴리스에서 실행되고 있지 않은 Solaris 시스템에 미리 공유한 키를 추가하고 있습니다. 관리자는 앞의 절차에 따라 `ike/config` 및 `ike.preshared` 파일을 수정하고, 키를 생성하고, 원격 시스템에 연결합니다.

- 새 키를 생성하기 전에 관리자는 `in.iked` 데몬의 권한 레벨을 2로 설정합니다.

```
# pkill in.iked
# /usr/lib/inet/in.iked -p 2
Setting privilege level to 2
```

- 키를 다른 시스템으로 보내고 시스템에 새 키를 추가한 후 관리자는 권한 레벨을 내립니다.

  ```
  # ikeadm set priv base
  ```
- 마지막으로 관리자는 새 IKE 규칙을 커널로 읽습니다.

  ```
  # ikeadm read rules
  ```

다음 순서 IPsec 정책 설정을 완료하지 않았으면 IPsec 정책을 사용으로 설정하거나 새로 고치는 IPsec 절차로 돌아가십시오.

▼ IKE 미리 공유한 키가 동일한지 확인하는 방법

통신 시스템에서 미리 공유한 키가 동일하지 않은 경우 시스템을 인증할 수 없습니다.

시작하기 전에 IPsec가 구성되고 테스트 중인 두 시스템 간에 사용으로 설정됩니다. 현재 Oracle Solaris 10 릴리스가 실행 중입니다.

주 – Solaris 10 4/09 이전 릴리스에서 이 절차를 수행하려면 예 23–2를 참조하십시오.

---

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 각 시스템에서 `in.iked` 데몬의 권한 레벨을 확인합니다.

```
# svcprop -p config/admin_privilege ike
base
```

- **권한 레벨이 `keymat`인 경우 단계 3을 진행합니다.**
- **권한 레벨이 `base` 또는 `modkeys`인 경우 권한 레벨을 향상시킵니다.**

  `ike` 서비스를 새로 고치고 다시 시작합니다.

  ```
  # svccfg -s ike setprop config/admin_privilege=keymat
  # svcadm refresh ike ; svcadm restart ike
  # svcprop -p config/admin_privilege ike
  keymat
  ```

3 **각 시스템에서 미리 공유한 키 정보를 봅니다.**

```
# ikeadm dump preshared
PSKEY: Preshared key (24 bytes): f47cb.../192
LOCIP: AF_INET: port 0, 192.168.116.16 (enigma).
REMIP: AF_INET: port 0, 192.168.13.213 (partym).
```

4 **두 덤프를 비교합니다.**

미리 공유한 키가 동일하지 않으면 한 키를 `/etc/inet/secret/ike.preshared` 파일의 다른 키로 바꿉니다.

5 **확인이 완료되면 각 시스템의 권한 레벨을 기본값으로 되돌립니다.**

```
# svccfg -s ike setprop config/admin_privilege=base
# svcadm restart ike
```

공개 키 인증서로 IKE 구성(작업 맵)

다음 표에서는 IKE에 대한 공개 키 인증서를 만드는 절차에 대해 설명합니다. 이 절차에서는 인증서를 빠르게 만들고 연결된 하드웨어에 저장하는 방법을 설명합니다.

공개 인증서는 고유해야 하므로 공개 키 인증서 작성자는 인증서의 이름을 임의적으로 고유한 이름으로 생성합니다. 일반적으로 X.509 식별 이름이 사용됩니다. 식별을 위해 대체 이름을 사용할 수도 있습니다. 이러한 이름의 형식은 *tag=value*입니다. 이 값은 임의적이지만 값의 형식은 태그 유형에 적합해야 합니다. 예를 들어, `email` 태그의 형식은 *name@domain.suffix*입니다.

작업	설명	수행 방법
자체 서명된 공개 키 인증서로 IKE를 구성합니다.	다음 두 개의 인증서를 만들어 각 시스템에 배치합니다. ■ 자체 서명된 인증서 ■ 원격 시스템의 공개 키 인증서	557 페이지 "자체 서명된 공개 키 인증서로 IKE를 구성하는 방법"
PKI 인증 기관으로 IKE를 구성합니다.	인증서 요청을 만들고 각 시스템에 다음 세 개의 인증서를 배치합니다. ■ 인증 기관(CA)이 요청에 따라 만든 인증서 ■ CA의 공개 키 인증서 ■ CA의 CRL	562 페이지 "CA가 서명한 인증서로 IKE를 구성하는 방법"
로컬 하드웨어에서 공개 키 인증서를 구성합니다.	다음 작업 중 하나를 수행합니다. ■ 로컬 하드웨어에서 자체 서명된 인증서를 생성한 다음 원격 시스템의 공개 키를 하드웨어에 추가합니다. ■ 로컬 하드웨어에서 인증서 요청을 생성한 다음 CA의 공개 키 인증서를 하드웨어에 추가합니다.	567 페이지 "공개 키 인증서를 생성하여 하드웨어에 저장하는 방법"

작업	설명	수행 방법
PKI에서 인증서 해지 목록(CRL)을 업데이트합니다.	중앙 배포 지점에서 CRL에 액세스합니다.	571 페이지 "인증서 해지 목록 처리 방법"

공개 키 인증서로 IKE 구성

공개 키 인증서를 사용하면 통신하는 시스템이 대역 외 연결에서 보안 키 입력 도구를 공유할 필요가 없습니다. 미리 공유한 키와 달리 공개 키 인증서는 모바일 시스템 또는 번호가 재지정될 수 있는 시스템에서 사용할 수 있습니다.

공개 키 인증서를 연결된 하드웨어에 저장할 수도 있습니다. 절차는 581 페이지 "연결된 하드웨어를 찾도록 IKE 구성"을 참조하십시오.

▼ 자체 서명된 공개 키 인증서로 IKE를 구성하는 방법

이 절차에서는 인증서 쌍을 만듭니다. 개인 키는 로컬 인증서 데이터베이스의 디스크에 저장되며 `certlocal` 하위 명령을 사용하여 참조할 수 있습니다. 인증서 쌍의 공개 부분은 공개 인증서 데이터베이스에 저장됩니다. 이는 `certdb` 하위 명령을 사용하여 참조할 수 있습니다. 피어 시스템과 공개 부분을 교환합니다. 두 인증서의 조합은 IKE 전송 인증에 사용됩니다.

자체 서명된 인증서는 CA의 공개 인증서보다 오버헤드가 적지만 확장이 어렵습니다. CA에서 발급한 인증서와 달리 자체 서명된 인증서는 대역 외 연결에서 확인해야 합니다.

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 **`ike.privatekeys` 데이터베이스에 자체 서명된 인증서를 만듭니다.**

```
# ikecert certlocal -ks|-kc -m keysize -t keytype \
-D dname -A altname \
[-S validity-start-time] [-F validity-end-time] [-T token-ID]
```

`-ks` 자체 서명된 인증서를 만듭니다.

`-kc` 인증서 요청을 만듭니다. 절차는 562 페이지 "CA가 서명한 인증서로 IKE를 구성하는 방법"을 참조하십시오.

`-m` *keysize*	키의 크기입니다. *keysize*는 512, 1024, 2048, 3072 또는 4096일 수 있습니다.
`-t` *keytype*	사용할 알고리즘의 유형을 지정합니다. *keytype*은 `rsa-sha1`, `rsa-md5` 또는 `dsa-sha1`일 수 있습니다.
`-D` *dname*	인증서 주체에 대한 X.509 식별 이름입니다. *dname*의 일반적인 형식은 C=국가, O=조직, OU=조직 구성 단위, CN=공통 이름입니다. 유효한 태그는 C, O, OU 및 CN입니다.
`-A` *altname*	인증서의 대체 이름입니다. *altname*의 형식은 `tag=value`입니다. 유효한 태그는 `IP`, `DNS`, `email` 및 `DN`입니다.
`-S` *validity-start-time*	인증서 시작 시간을 유효한 절대 또는 상대 시작 시간으로 지정합니다.
`-F` *validity-end-time*	인증서 종료 시간을 유효한 절대 또는 상대 종료 시간으로 지정합니다.
`-T` *token-ID*	PKCS #11 하드웨어 토큰이 키를 생성할 수 있도록 합니다. 그러면 인증서가 하드웨어에 저장됩니다.

a. 예를 들어, `partym` 시스템의 명령은 다음과 유사하게 표시됩니다.

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=US, O=PartyCo, OU=US-Partym, CN=Partym" \
-A IP=192.168.13.213
Creating software private keys.
  Writing private key to file /etc/inet/secret/ike.privatekeys/0.
Enabling external key providers - done.
Acquiring private keys for signing - done.
Certificate:
 Proceeding with the signing operation.
 Certificate generated successfully (.../publickeys/0)
Finished successfully.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIICLTCCAZagAwIBAgIBATANBgkqhkiG9w0BAQQFADBNMQswCQYDVQQGEwJVUzEX
...
6sKTxpg4GP3GkQGcd0r1rhW/3yaWBkDwOdFCqEUyffzU
-----END X509 CERTIFICATE-----
```

주 – `-D` 및 `-A` 옵션의 값은 임의의 값입니다. 이 값은 인증서를 식별하는 데만 사용됩니다. 192.168.13.213 등의 시스템을 식별하는 데는 사용되지 않습니다. 실제로 이러한 값은 고유하므로 피어 시스템에 올바른 인증서가 설치되어 있는지 대역 외 연결에서 확인해야 합니다.

---

b. `enigma` 시스템의 명령은 다음과 유사하게 표시됩니다.

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax" \
-A IP=192.168.116.16
```

```
Creating software private keys.
  ...
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
MIICKDCCAZGgAwIBAgIBATANBgkqhkiG9w0BAQQFADBJMQswCQYDVQQGEwJVUzEV
...
jpxfLM98xyFVyLCbkr3dZ3Tvxvi732BXePKF2A==
-----END X509 CERTIFICATE-----
```

3 **인증서를 저장하여 원격 시스템으로 보냅니다.**

이 인증서는 전자 메일에 첨부할 수 있습니다.

출력은 인증서 공개 부분의 인코딩된 버전입니다. 이 인증서는 전자 메일에 안전하게 첨부할 수 있습니다. 수신자는 단계 5와 같이 올바른 인증서를 설치했는지 대역 외 연결에서 확인해야 합니다.

a. **예를 들어, 다음 `partym` 인증서의 공개 부분을 `enigma` 관리자에게 보냅니다.**

```
To: admin@ja.enigmaexample.com
From: admin@us.partyexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MIICLTCCAZagAwIBAgIBATANBgkqhkiG9w0BAQQFADBNMQswCQYDVQQGEwJVUzEX
...
6sKTxpg4GP3GkQGcd0r1rhW/3yaWBkDwOdFCqEUyffzU
-----END X509 CERTIFICATE-----
```

b. **`enigma` 관리자로부터 다음 `enigma` 인증서의 공개 부분을 받습니다.**

```
To: admin@us.partyexample.com
From: admin@ja.enigmaexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MIICKDCCAZGgAwIBAgIBATANBgkqhkiG9w0BAQQFADBJMQswCQYDVQQGEwJVUzEV
...
jpxfLM98xyFVyLCbkr3dZ3Tvxvi732BXePKF2A==
-----END X509 CERTIFICATE-----
```

4 **각 시스템에서 받은 인증서를 추가합니다.**

a. **관리자의 전자 메일에서 공개 키를 복사합니다.**

b. **`ikecert certdb -a` 명령을 입력하고 Return 키를 누릅니다.**

Return 키를 누르면 프롬프트가 표시되지 않습니다.

```
# ikecert certdb -a       Press the Return key
```

c. **공개 키를 붙여 넣습니다. 그런 다음 Return 키를 누릅니다. 입력을 마치려면 Control-D를 누릅니다.**

```
-----BEGIN X509 CERTIFICATE-----
MIIC...
...
----END X509 CERTIFICATE-----       Press the Return key
<Control>-D
```

5 다른 관리자가 이 인증서를 보낸 것인지 해당 관리자에게 확인합니다.

예를 들어, 다른 관리자와 전화 통화를 통해 수신한 공개 인증서의 해시가 해당 관리자만 가진 개인 인증서의 해시와 일치하는지 확인할 수 있습니다.

a. partym에 저장된 인증서를 나열합니다.

다음 예에서 Note 1은 슬롯 0에 있는 인증서의 식별 이름(DN)을 나타냅니다. 슬롯 0에 있는 개인 인증서가 동일한 해시(주 3 참조)를 가지므로 이러한 인증서는 동일한 인증서 쌍입니다. 공개 인증서가 작동하려면 일치 쌍이 있어야 합니다. `certdb` 하위 명령은 공개 부분을 나열하며 `certlocal` 하위 명령은 개인 부분을 나열합니다.

```
partym # ikecert certdb -l
Certificate Slot Name: 0   Type: rsa-sha1
    Subject Name: <C=US, O=PartyCo, OU=US-Partym, CN=Partym>     Note 1
    Key Size: 1024
    Public key hash: 2239A6A127F88EE0CB40F7C24A65B818

Certificate Slot Name: 1   Type: rsa-sha1
    (Private key in certlocal slot 0)
    Subject Name: <C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax>
    Key Size: 1024
    Public key hash: B2BD13FCE95FD27ECE6D2DCD0DE760E2
partym # ikecert certlocal -l
Local ID Slot Name: 0   Key Type: rsa-sha1
    Key Size: 1024
    Public key hash: 2239A6A127F88EE0CB40F7C24A65B818     Note 3

Local ID Slot Name: 1   Key Type: rsa-sha1
        Key Size: 1024
        Public key hash: FEA65C5387BBF3B2C8F16C019FEBC388
...
```

이 검사에서 `partym` 시스템에 유효한 인증서 쌍이 있는 것이 확인되었습니다.

b. enigma 시스템에 partym의 공개 인증서가 있는지 확인합니다.

전화를 통해 공개 키 해시를 확인할 수 있습니다.

이전 단계에서 확인된 `partym`의 Note 3 해시를 `enigma`의 Note 4와 비교합니다.

```
enigma # ikecert certdb -l
Certificate Slot Name: 4   Type: rsa-sha1
    Subject Name: <C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax>
    Key Size: 1024
    Public key hash: DF3F108F6AC669C88C6BD026B0FCE3A0

Certificate Slot Name: 5   Type: rsa-sha1
    Subject Name: <C=US, O=PartyCo, OU=US-Partym, CN=Partym>
    Key Size: 1024
    Public key hash: 2239A6A127F88EE0CB40F7C24A65B818     Note 4
```

enigma의 공개 인증서 데이터베이스에 저장된 마지막 인증서의 공개 키 해시 및 주체 이름이 이전 단계의 partym에 대한 개인 인증서와 일치합니다.

6 각 시스템에서 두 인증서를 인증합니다.

인증서가 인식되도록 /etc/inet/ike/config 파일을 편집합니다.

원격 시스템의 관리자가 cert_trust, remote_addr 및 remote_id 매개변수에 대한 값을 제공합니다.

a. 예를 들어, partym 시스템에서 ike/config 파일은 다음과 유사하게 표시됩니다.

```
# Explicitly trust the self-signed certs
# that we verified out of band. The local certificate
# is implicitly trusted because we have access to the private key.
cert_trust "192.168.116.16"       Remote system's certificate Subject Alt Name

## Parameters that may also show up in rules.

p1_xform
  { auth_method preshared oakley_group 5 auth_alg sha encr_alg 3des }
p2_pfs 5

{
 label "US-partym to JA-enigmax"
 local_id_type dn
 local_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"
 remote_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"

 local_addr  192.168.13.213

# We could explicitly enter the peerâs IP address here, but we donât need
# to do this with certificates, so use a wildcard address. The wildcard
# allows the remote device to be mobile or behind a NAT box.
# remote_addr 192.168.116.16
 remote_addr 0.0.0.0/0

 p1_xform
   { auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes }
}
```

b. enigma 시스템의 ike/config 파일에서 로컬 매개변수에 대한 enigma 값을 추가합니다.

원격 매개변수의 경우 partym 값을 사용합니다. label 키워드가 로컬 시스템에서 고유한지 확인합니다.

```
...
{
 label "JA-enigmax to US-partym"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
```

```
    remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

    local_addr  192.168.116.16
    remote_addr 0.0.0.0/0
   ...
```

예 23–4 인증서의 시작 시간 및 종료 시간 지정

이 예에서 partym 시스템의 관리자는 인증서의 유효 기간을 지정합니다. 인증서는 2.5일 전까지 소급 적용되며 생성된 날짜로부터 4년 6개월까지 유효합니다.

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
-A IP=192.168.13.213 \
-S -2d12h -F +4y6m
```

enigma 시스템의 관리자는 인증서의 유효 기간을 지정합니다. 인증서는 2일 전까지 소급 적용되며 2010년 12월 31일 자정까지 유효합니다.

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
-D "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax" \
-A IP=192.168.116.16 \
-S -2d -F "12/31/2010 12:00 AM"
```

▼ CA가 서명한 인증서로 IKE를 구성하는 방법

인증 기관(CA)의 공개 인증서를 사용하려면 외부 조직과의 협상이 필요합니다. 간편한 인증서 확장을 통해 통신하는 여러 시스템을 보호할 수 있습니다.

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 ssh 명령을 사용하십시오.

---

2 ikecert certlocal -kc 명령을 사용하여 인증서 요청을 만듭니다.

명령 인수에 대한 설명은 557 페이지 "자체 서명된 공개 키 인증서로 IKE를 구성하는 방법"의 단계 2를 참조하십시오.

```
# ikecert certlocal -kc -m keysize -t keytype \
-D dname -A altname
```

a. 예를 들어, 다음 명령은 partym 시스템에서 인증서 요청을 만듭니다.

```
# ikecert certlocal -kc -m 1024 -t rsa-sha1 \
> -D "C=US, O=PartyCompany\, Inc., OU=US-Partym, CN=Partym" \
> -A "DN=C=US, O=PartyCompany\, Inc., OU=US-Partym"
Creating software private keys.
  Writing private key to file /etc/inet/secret/ike.privatekeys/2.
Enabling external key providers - done.
Certificate Request:
  Proceeding with the signing operation.
  Certificate request generated successfully (.../publickeys/0)
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIByjCCATMCAQAwUzELMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGVDb21w
...
lcM+tw0ThRrfuJX9t/Qa1R/KxRlMA3zckO80mO9X
-----END CERTIFICATE REQUEST-----
```

b. 다음 명령은 enigma 시스템에서 인증서 요청을 만듭니다.

```
# ikecert certlocal -kc -m 1024 -t rsa-sha1 \
> -D "C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax, CN=Enigmax" \
> -A "DN=C=JA, O=EnigmaCo\, Inc., OU=JA-Enigmax"
Creating software private keys.
...
Finished successfully.
-----BEGIN CERTIFICATE REQUEST-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
...
8qlqdjaStLGfhDOO
-----END CERTIFICATE REQUEST-----
```

3 PKI 조직에 인증서 요청을 제출합니다.

PKI 조직에서 인증서 요청 제출 방법을 제공할 수 있습니다. 대부분 조직에는 제출 양식을 제공하는 웹 사이트가 있습니다. 양식을 사용하려면 제출이 적합한지 증명해야 합니다. 일반적으로 양식에 인증서 요청을 붙여 넣습니다. 요청을 확인한 조직에서는 다음 두 개의 인증서 객체와 해지된 인증서 목록을 발급합니다.

- 공개 키 인증서 – 이 인증서는 사용자가 해당 조직에 제출한 요청을 기반으로 합니다. 제출한 요청은 이 공개 키 인증서의 일부입니다. 인증서는 사용자를 고유하게 식별합니다.
- 인증 기관 – 조직의 서명입니다. CA는 공개 키 인증서가 적합한지 확인합니다.
- 인증서 해지 목록(CRL) – 조직에서 해지한 최신 인증서 목록입니다. CRL에 대한 액세스 권한이 공개 키 인증서에 포함된 경우 CRL이 인증서 객체로 별도로 전송되지 않습니다.

CRL에 대한 URI가 공개 키 인증서에 포함된 경우 IKE가 자동으로 CRL을 검색할 수 있습니다. 마찬가지로 DN(LDAP 서버의 디렉토리 이름) 항목이 공개 키 인증서에 포함된 경우 IKE가 지정된 LDAP 서버에서 CRL을 검색하여 캐시할 수 있습니다.

공개 키 인증서에 포함된 URI 및 포함된 DN 항목의 예는 571 페이지 "인증서 해지 목록 처리 방법"을 참조하십시오.

4 시스템에 각 인증서를 추가합니다.

`ikecert certdb -a`에 대한 `-a` 옵션은 붙여 넣은 객체를 시스템의 적합한 인증서 데이터베이스에 추가합니다. 자세한 내용은 540 페이지 "IKE와 공개 키 인증서"를 참조하십시오.

a. 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

b. PKI 조직에서 수신한 공개 키 인증서를 추가합니다.

```
# ikecert certdb -a
      Press the Return key
      Paste the certificate:
-----BEGIN X509 CERTIFICATE-----
...
-----END X509 CERTIFICATE----
      Press the Return key
<Control>-D
```

c. PKI 조직의 CA를 추가합니다.

```
# ikecert certdb -a
      Press the Return key
      Paste the CA:
-----BEGIN X509 CERTIFICATE-----
...
-----END X509 CERTIFICATE----
      Press the Return key
<Control>-D
```

d. PKI 조직에서 해지된 인증서 목록을 보낸 경우 `certrldb` 데이터베이스에 CRL을 추가합니다.

```
# ikecert certrldb -a
      Press the Return key
      Paste the CRL:
-----BEGIN CRL-----
...
-----END CRL----
      Press the Return key
<Control>-D
```

5 cert_root 키워드를 사용하여 /etc/inet/ike/config 파일에서 PKI 조직을 식별합니다.
PKI 조직에서 제공한 이름을 사용합니다.

a. 예를 들어, partym 시스템의 ike/config 파일은 다음과 유사하게 표시될 수 있습니다.

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

## Parameters that may also show up in rules.

p1_xform
  { auth_method rsa_sig oakley_group 1 auth_alg sha1 encr_alg 3des }
p2_pfs 2

{
 label "US-partym to JA-enigmax - Example PKI"
 local_id_type dn
 local_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"
 remote_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"

 local_addr  192.168.13.213
 remote_addr 192.168.116.16

 p1_xform
   { auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes }
}
```

주 – auth_method 매개변수에 대한 모든 인수는 동일한 행에 있어야 합니다.

b. enigma 시스템에서 유사한 파일을 만듭니다.
특히 enigma ike/config 파일은 다음을 따라야 합니다.

- 동일한 cert_root 값을 포함합니다.
- 로컬 매개변수에 enigma 값을 사용합니다.
- 원격 매개변수에 partym 값을 사용합니다.
- label 키워드에 고유한 값을 만듭니다. 이 값은 원격 시스템의 label 값과 달라야 합니다.

```
...
cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"
...
{
 label "JA-enigmax to US-partym - Example PKI"
 local_id_type dn
 local_id   "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
```

```
 remote_addr 192.168.13.213
...
```

6 IKE에 CRL 처리 방법을 알립니다.

적합한 옵션을 선택합니다.

- **사용 가능한 CRL 없음**

  PKI 조직에서 CRL을 제공하지 않을 경우 `ignore_crls` 키워드를 `ike/config` 파일에 추가합니다.

```
# Trusted root cert
...
cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example,...
ignore_crls
...
```

  `ignore_crls` 키워드는 IKE에 CRL을 검색하지 않도록 알립니다.

- **사용 가능한 CRL 있음**

  PKI 조직에서 CRL에 대한 중앙 배포 지점을 제공할 경우 해당 위치를 가리키도록 `ike/config` 파일을 수정할 수 있습니다.

  예는 571 페이지 “인증서 해지 목록 처리 방법”을 참조하십시오.

예 23–5 IKE 구성 시 rsa_encrypt 사용

`ike/config` 파일의 `auth_method rsa_encrypt`를 사용할 경우 `publickeys` 데이터베이스에 피어의 인증서를 추가해야 합니다.

1. 원격 시스템의 관리자에게 인증서를 보냅니다.

   이 인증서는 전자 메일에 첨부할 수 있습니다.

   예를 들어, `partym` 관리자가 다음 전자 메일을 보냅니다.

```
To: admin@ja.enigmaexample.com
From: admin@us.partyexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MII...
----END X509 CERTIFICATE-----
```

   `enigma` 관리자가 다음 전자 메일을 보냅니다.

```
To: admin@us.partyexample.com
From: admin@ja.enigmaexample.com
Message: -----BEGIN X509 CERTIFICATE-----
MII
...
-----END X509 CERTIFICATE-----
```

2. 각 시스템에서 로컬 `publickeys` 데이터베이스에 전자 메일을 통해 전송된 인증서를 추가합니다.

```
# ikecert certdb -a
      Press the Return key
-----BEGIN X509 CERTIFICATE-----
MII...
-----END X509 CERTIFICATE-----
      Press the Return key
<Control>-D
```

RSA 암호화에 대한 인증 방법은 IKE에서 도청자에게 ID를 숨깁니다. rsa_encrypt 메소드는 피어의 ID를 숨기므로 IKE는 피어의 인증서를 검색할 수 없습니다. 즉, rsa_encrypt 메소드를 사용하려면 IKE 피어가 상대의 공개 키를 알고 있어야 합니다.

따라서 /etc/inet/ike/config 파일에 있는 rsa_encrypt의 auth_method를 사용할 경우 publickeys 데이터베이스에 피어의 인증서를 추가해야 합니다. 그러면 publickeys 데이터베이스가 통신하는 시스템 쌍의 각각에 대해 다음 세 개의 인증서를 보유합니다.

- 공개 키 인증서
- CA 인증서
- 피어의 공개 키 인증서

문제 해결 – 세 개의 인증서를 포함하는 IKE 페이로드는 너무 커서 rsa_encrypt를 통해 암호화하지 못할 수 있습니다. “authorization failed”, “malformed payload” 등의 오류는 rsa_encrypt 메소드가 전체 페이로드를 암호화할 수 없음을 나타내는 것일 수 있습니다. 두 개의 인증서만 필요로 하는 rsa_sig 등의 메소드를 사용하여 페이로드 크기를 줄이십시오.

▼ 공개 키 인증서를 생성하여 하드웨어에 저장하는 방법

공개 키 인증서를 생성하여 하드웨어에 저장하는 작업은 시스템에서 공개 키 인증서를 생성하여 저장하는 작업과 유사합니다. 하드웨어에서 ikecert certlocal 및 ikecert certdb 명령이 하드웨어를 식별해야 합니다. 토큰 ID를 사용하는 -T 옵션은 명령에 대한 하드웨어를 식별합니다.

시작하기 전에

- 하드웨어가 구성되어 있어야 합니다.
- /etc/inet/ike/config 파일의 pkcs11_path 키워드가 다른 라이브러리를 가리키지 않을 경우 하드웨어는 /usr/lib/libpkcs11.so 라이브러리를 사용합니다. RSA Security Inc. PKCS #11 암호화 토큰 인터페이스(Cryptoki), 즉 PKCS #11 라이브러리 표준에 따라 라이브러리가 구성되어 있어야 합니다.

  설정 지침은 582 페이지 “Sun Crypto Accelerator 4000 보드를 찾도록 IKE를 구성하는 방법”을 참조하십시오.

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 **자체 서명된 인증서 또는 인증서 요청을 생성하고 토큰 ID를 지정합니다.**

다음 옵션 중 하나를 선택합니다.

주 – Sun Crypto Accelerator 4000 및 Sun Crypto Accelerator 6000 보드는 RSA에 대해 최대 2048비트의 키를 지원합니다. DSA의 경우 이 보드는 최대 1024비트의 키를 지원합니다.

- **자체 서명된 인증서의 경우 다음 구문을 사용합니다.**

```
# ikecert certlocal -ks -m 1024 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

`-T` 옵션에 대한 인수는 연결된 보드의 토큰 ID입니다.

- **인증서 요청의 경우 다음 구문을 사용합니다.**

```
# ikecert certlocal -kc -m 1024 -t rsa-sha1 \
> -D "C=US, O=PartyCompany, OU=US-Partym, CN=Partym" \
> -a -T dca0-accel-stor IP=192.168.116.16
Creating hardware private keys.
Enter PIN for PKCS#11 token:      Type user:password
```

`ikecert` 명령 인수에 대한 설명은 `ikecert(1M)` 매뉴얼 페이지를 참조하십시오.

3 **PIN에 대한 프롬프트에서 보드 사용자, 콜론 및 사용자 암호를 입력합니다.**

보드에 암호가 `rgm4tigt`인 사용자 `ikemgr`이 있을 경우 다음을 입력합니다.

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
```

주 – PIN 응답은 디스크에 **일반 텍스트로** 저장됩니다.

암호를 입력하면 인증서가 다음과 같이 출력됩니다.

```
Enter PIN for PKCS#11 token: ikemgr:rgm4tigt
-----BEGIN X509 CERTIFICATE-----
MIIBuDCCASECAQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDFBhcnR5Q29tcGFu
```

```
...
oKUDBbZ9O/pLWYGr
-----END X509 CERTIFICATE-----
```

4 **상대방이 사용할 인증서를 보냅니다.**

다음 옵션 중 하나를 선택합니다.

- **원격 시스템에 자체 서명된 인증서를 보냅니다.**

  이 인증서는 전자 메일에 첨부할 수 있습니다.

- **PKI를 처리하는 조직에 인증서 요청을 보냅니다.**

  PKI 조직의 지침에 따라 인증서 요청을 제출합니다. 자세한 설명은 562 페이지 “CA가 서명한 인증서로 IKE를 구성하는 방법”의 단계 3을 참조하십시오.

5 **시스템에서 인증서가 인식되도록 `/etc/inet/ike/config` 파일을 편집합니다.**

다음 옵션 중 하나를 선택합니다.

- **자체 서명된 인증서**

  원격 시스템의 관리자가 `cert_trust`, `remote_id` 및 `remote_addr` 매개변수에 대해 제공하는 값을 사용합니다. 예를 들어, `enigma` 시스템에서 `ike/config` 파일은 다음과 유사하게 표시됩니다.

```
# Explicitly trust the following self-signed certs
# Use the Subject Alternate Name to identify the cert

cert_trust "192.168.116.16"       Local system's certificate Subject Alt Name
cert_trust "192.168.13.213"       Remote system's certificate Subject Alt name


# Solaris 10 1/06 release: default path does not have to be typed in #pkcs11_path
"/usr/lib/libpkcs11.so"      Hardware connection

# Solaris 10 release: use this path
#pkcs11_path "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
...
{
 label "JA-enigmax to US-partym"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213

 p1_xform
```

```
  {auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes}
}
```

- **인증서 요청**

  PKI 조직에서 `cert_root` 키워드에 대한 값으로 제공하는 이름을 입력합니다. 예를 들어, `enigma` 시스템의 `ike/config` 파일은 다음과 유사하게 표시될 수 있습니다.

```
# Trusted root cert
# This certificate is from Example PKI
# This is the X.509 distinguished name for the CA that it issues.

cert_root "C=US, O=ExamplePKI\, Inc., OU=PKI-Example, CN=Example PKI"

# Solaris 10 1/06 release: default path does not have to be typed in #pkcs11_path
"/usr/lib/libpkcs11.so"      Hardware connection


# Solaris 10 release: use this path
#pkcs11_path "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
...
{
 label "JA-enigmax to US-partym - Example PKI"
 local_id_type dn
 local_id "C=JA, O=EnigmaCo, OU=JA-Enigmax, CN=Enigmax"
 remote_id  "C=US, O=PartyCompany, OU=US-Partym, CN=Partym"

 local_addr  192.168.116.16
 remote_addr 192.168.13.213

 p1_xform
  {auth_method rsa_sig oakley_group 2 auth_alg sha1 encr_alg aes}
}
```

6 하드웨어에서 상대방의 인증서를 배치합니다.

단계 3에서 응답한 대로 PIN 요청에 응답합니다.

주 – 반드시 개인 키를 생성한 것과 동일한 연결된 하드웨어에 공개 키 인증서를 추가해야 합니다.

- **자체 서명된 인증서**

  원격 시스템의 자체 서명된 인증서를 추가합니다. 이 예에서는 인증서가 `DCA.ACCEL.STOR.CERT` 파일에 저장됩니다.

```
# ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
Enter PIN for PKCS#11 token:      Type user:password
```

  자체 서명된 인증서가 `rsa_encrypt`를 `auth_method` 매개변수에 대한 값으로 사용한 경우 하드웨어 저장소에 피어의 인증서를 추가합니다.

- **PKI 조직의 인증서**

  인증서 요청에 따라 조직에서 생성한 인증서를 추가하고 인증 기관(CA)을 추가합니다.

  ```
  # ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CERT
  Enter PIN for PKCS#11 token:      Type user:password

  # ikecert certdb -a -T dca0-accel-stor < DCA.ACCEL.STOR.CA.CERT
  Enter PIN for PKCS#11 token:      Type user:password
  ```

  PKI 조직의 인증서 해지 목록(CRL)을 추가하려면 571 페이지 “인증서 해지 목록 처리 방법”을 참조하십시오.

▼ 인증서 해지 목록 처리 방법

인증서 해지 목록(CRL)에는 인증 기관의 오래되거나 손상된 인증서가 포함됩니다. 네 가지 방법으로 CRL을 처리할 수 있습니다.

- CA 조직에서 CRL을 발급하지 않은 경우 CRL을 무시하도록 IKE에 알려야 합니다. 이 옵션은 562 페이지 “CA가 서명한 인증서로 IKE를 구성하는 방법”의 단계 6에서 설명됩니다.
- CA의 공개 키 인증서에 주소가 포함된 URI(Uniform Resource Indicator)의 CRL에 액세스하도록 IKE에 알릴 수 있습니다.
- CA의 공개 키 인증서에 디렉토리 이름(DN) 항목이 포함된 LDAP 서버의 CRL에 액세스하도록 IKE에 알릴 수 있습니다.
- `ikecert certrldb` 명령에 대한 인수로 CRL을 제공할 수 있습니다. 예는 예 23–6를 참조하십시오.

다음 절차에서는 중앙 배포 지점의 CRL을 사용하도록 IKE에 알리는 방법에 대해 설명합니다.

1 CA에서 수신한 인증서를 표시합니다.

```
# ikecert certdb -lv certspec
```

`-l` IKE 인증서 데이터베이스의 인증서를 나열합니다.

`-v` 상세 정보 표시 모드로 인증서를 나열합니다. 이 옵션은 주의해서 사용하십시오.

certspec IKE 인증서 데이터베이스의 인증서와 일치하는 패턴입니다.

예를 들어, Oracle에서 발급한 인증서는 다음과 같습니다. 세부 정보는 변경되었습니다.

```
# ikecert certdb -lv example-protect.oracle.com
Certificate Slot Name: 0   Type: dsa-sha1
   (Private key in certlocal slot 0)
 Subject Name: <O=Oracle, CN=example-protect.oracle.com>
```

```
Issuer Name: <CN=Oracle CA (Cl B), O=Oracle>
SerialNumber: 14000D93
  Validity:
     Not Valid Before: 2002 Jul 19th, 21:11:11 GMT
     Not Valid After:  2005 Jul 18th, 21:11:11 GMT
  Public Key Info:
     Public Modulus  (n) (2048 bits): C575A...A5
     Public Exponent (e) (  24 bits): 010001
  Extensions:
     Subject Alternative Names:
            DNS = example-protect.oracle.com
     Key Usage: DigitalSignature KeyEncipherment
     [CRITICAL]
  CRL Distribution Points:
     Full Name:
        URI = #Ihttp://www.oracle.com/pki/pkismica.crl#i
        DN = <CN= Oracle CA (Cl B), O=Oracle>
     CRL Issuer:
     Authority Key ID:
     Key ID:              4F ... 6B
     SubjectKeyID:        A5 ... FD
     Certificate Policies
     Authority Information Access
```

CRL Distribution Points 항목을 확인합니다. URI 항목은 이 조직의 CRL을 웹에서 사용할 수 있음을 나타냅니다. DN 항목은 CRL을 LDAP 서버에서 사용할 수 있음을 나타냅니다. IKE가 액세스한 CRL은 나중에 사용할 수 있도록 캐시됩니다.

CRL에 액세스하려면 배포 지점에 연결해야 합니다.

2 중앙 배포 지점에서 CRL에 액세스하는 데 사용할 다음 방법 중 하나를 선택합니다.

- **URI 사용**

  use_http 키워드를 호스트의 /etc/inet/ike/config 파일에 추가합니다. 예를 들어, ike/config 파일은 다음과 유사하게 표시됩니다.

  ```
  # Use CRL from organization's URI
  use_http
  ...
  ```

- **웹 프록시 사용**

  proxy 키워드를 ike/config 파일에 추가합니다. proxy 키워드는 다음에서와 같이 URL을 인수로 사용합니다.

  ```
  # Use own web proxy
  proxy "http://proxy1:8080"
  ```

- **LDAP 서버 사용**

  호스트의 `/etc/inet/ike/config` 파일에서 LDAP 서버를 `ldap-list` 키워드에 대한 인수로 지정합니다. 조직에서 LDAP 서버의 이름을 제공합니다. `ike/config` 파일의 항목은 다음과 유사하게 표시됩니다.

  ```
  # Use CRL from organization's LDAP
  ldap-list "ldap1.oracle.com:389,ldap2.oracle.com"
  ...
  ```

IKE가 CRL을 검색하고 인증서가 만료될 때까지 CRL을 캐시합니다.

예 23–6 로컬 certrldb 데이터베이스에 CRL 붙여넣기

중앙 배포 지점에서 PKI 조직의 CRL을 사용할 수 없을 경우 수동으로 로컬 `certrldb` 데이터베이스에 CRL을 추가할 수 있습니다. PKI 조직의 지침에 따라 CRL을 파일에 추출한 다음 `ikecert certrldb -a` 명령을 사용하여 데이터베이스에 CRL을 추가합니다.

```
# ikecert certrldb -a < Oracle.Cert.CRL
```

모바일 시스템에 대한 IKE 구성(작업 맵)

다음 표에서는 원격으로 중앙 사이트에 로그인한 시스템을 처리하도록 IKE를 구성하는 절차에 대해 설명합니다.

작업	설명	수행 방법
오프사이트의 중앙 사이트와 통신합니다.	오프사이트 시스템이 중앙 사이트와 통신할 수 있도록 합니다. 오프사이트 시스템은 모바일일 수 있습니다.	574 페이지 "오프사이트 시스템에 대한 IKE 구성 방법"
모바일 시스템의 트래픽을 승인하는 중앙 시스템에서 CA의 공개 인증서 및 IKE를 사용합니다.	고정 IP 주소가 없는 시스템의 IPsec 트래픽을 승인하도록 게이트웨이 시스템을 구성합니다.	예 23–7
고정 IP 주소가 없는 시스템에서 CA의 공개 인증서 및 IKE를 사용합니다.	회사 본사 등의 중앙 사이트에 대한 트래픽을 보호하도록 모바일 시스템을 구성합니다.	예 23–8
모바일 시스템의 트래픽을 승인하는 중앙 시스템에서 자체 서명된 인증서 및 IKE를 사용합니다.	모바일 시스템의 IPsec 트래픽을 승인하도록 자체 서명된 인증서로 게이트웨이 시스템을 구성합니다.	예 23–9
고정 IP 주소가 없는 시스템에서 자체 서명된 인증서 및 IKE를 사용합니다.	중앙 사이트에 대한 트래픽을 보호하도록 자체 서명된 인증서로 모바일 시스템을 구성합니다.	예 23–10

모바일 시스템에 대한 IKE 구성

제대로 구성된 경우 자택 근무 시, 그리고 모바일 랩탑에서 IPsec 및 IKE를 사용하여 회사의 중앙 컴퓨터와 통신할 수 있습니다. 공개 키 인증 방법과 결합된 총괄 IPsec 정책을 통해 오프사이트 시스템은 중앙 시스템에 대한 트래픽을 보호할 수 있습니다.

▼ 오프사이트 시스템에 대한 IKE 구성 방법

IPsec 및 IKE에는 소스 및 대상을 식별할 고유한 ID가 필요합니다. 고유한 IP 주소가 없는 오프사이트 또는 모바일 시스템의 경우 다른 ID 유형을 사용해야 합니다. `DNS`, `DN`, `email` 등의 ID 유형을 사용하여 시스템을 고유하게 식별할 수 있습니다.

고유한 IP 주소가 있는 오프사이트 또는 모바일 시스템은 다른 ID 유형으로 구성하는 것이 좋습니다. 예를 들어, 시스템이 NAT 박스 뒤에 있는 중앙 사이트에 연결하려고 시도할 경우 고유한 주소가 사용되지 않습니다. NAT 박스는 중앙 시스템에서 인식할 수 없는 임의적인 IP 주소를 지정합니다.

미리 공유한 키도 모바일 시스템에 대한 인증 방식으로 작동하지 않습니다. 미리 공유한 키에는 고정 IP 주소가 필요하기 때문입니다. 모바일 시스템은 자체 서명된 인증서 또는 PKI의 인증서를 통해 중앙 사이트와 통신할 수 있습니다.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 모바일 시스템을 인식하도록 중앙 시스템을 구성합니다.

a. `ipsecinit.conf` 파일을 설정합니다.

중앙 시스템에는 광범위한 IP 주소를 허용하는 정책이 필요합니다. 나중에 IKE 정책의 인증서를 사용하면 연결하는 시스템이 적합한 것으로 보장됩니다.

```
# /etc/inet/ipsecinit.conf on central
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

b. `ike.config` 파일을 설정합니다.

DNS가 중앙 시스템을 식별합니다. 인증서는 시스템을 인증하는 데 사용됩니다.

```
## /etc/inet/ike/ike.config on central
# Global parameters
#
```

```
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://somecache.domain:port/"
#
# Use LDAP server
ldap_server    "ldap-server1.domain.org,ldap2.domain.org:port"
#
# List CA-signed certificates
cert_root    "C=US, O=Domain  Org, CN=Domain STATE"
#
# List self-signed certificates - trust server and enumerated others
#cert_trust    "DNS=central.domain.org"
#cert_trust    "DNS=mobile.domain.org"
#cert_trust    "DN=CN=Domain Org STATE (CLASS), O=Domain Org
#cert_trust    "email=root@central.domain.org"
#cert_trust    "email=user1@mobile.domain.org"
#

# Rule for mobile systems with certificate
{
  label "Mobile systems with certificate"
  local_id_type DNS

# CA's public certificate ensures trust,
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

3 각 모바일 시스템에 로그인하고 중앙 시스템을 찾도록 시스템을 구성합니다.

a. /etc/hosts 파일을 설정합니다.

/etc/hosts 파일은 모바일 시스템의 주소를 필요로 하지 않지만 제공할 수 있습니다. 파일에는 중앙 시스템에 대한 공용 IP 주소가 포함되어야 합니다.

```
# /etc/hosts on mobile
central 192.xxx.xxx.x
```

b. ipsecinit.conf 파일을 설정합니다.

모바일 시스템이 공용 IP 주소로 중앙 시스템을 찾아야 합니다. 시스템은 동일한 IPsec 정책을 구성해야 합니다.

```
# /etc/inet/ipsecinit.conf on mobile
# Find central
{raddr 192.xxx.xxx.x} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}
```

c. ike.config 파일을 설정합니다.

IP 주소는 식별자일 수 없습니다. 모바일 시스템에 유효한 식별자는 다음과 같습니다.

- DN=*ldap-directory-name*
- DNS=*domain-name-server-address*
- email=*email-address*

인증서는 모바일 시스템을 인증하는 데 사용됩니다.

```
## /etc/inet/ike/ike.config on mobile
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://somecache.domain:port/"
#
# Use LDAP server
ldap_server    "ldap-server1.domain.org,ldap2.domain.org:port"
#
# List CA-signed certificates
cert_root     "C=US, O=Domain  Org, CN=Domain STATE"
#
# Self-signed certificates - trust me and enumerated others
#cert_trust     "DNS=mobile.domain.org"
#cert_trust     "DNS=central.domain.org"
#cert_trust     "DN=CN=Domain Org STATE (CLASS), O=Domain Org
#cert_trust     "email=user1@domain.org"
#cert_trust     "email=root@central.domain.org"
#
# Rule for off-site systems with root certificate
{
    label "Off-site mobile with certificate"
    local_id_type DNS

# NAT-T can translate local_addr into any public IP address
# central knows me by my DNS

    local_id "mobile.domain.org"
    local_addr 0.0.0.0/0

# Find central and trust the root certificate
    remote_id "central.domain.org"
    remote_addr 192.xxx.xxx.x

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

4 IKE 구성을 커널로 읽습니다.

- **Solaris 10 4/09 릴리스부터 ike 서비스를 사용으로 설정합니다.**

  ```
  # svcadm enable svc:/network/ipsec/ike
  ```

- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우에는 시스템을 재부트합니다.**

  ```
  # init 6
  ```

  또는 `in.iked` 데몬을 중지한 후 시작합니다.

예 23–7 모바일 시스템의 IPsec 트래픽을 승인하도록 중앙 컴퓨터 구성

IKE는 NAT 박스 뒤에서 협상을 시작할 수 있습니다. 하지만 적합한 IKE 설정은 개입하는 NAT 박스가 없는 것입니다. 다음 예에서는 CA의 공개 인증서가 모바일 시스템 및 중앙 시스템에 배치되었습니다. 중앙 시스템이 NAT 박스 뒤에 있는 시스템의 IPsec 협상을 승인합니다. `main1`은 오프사이트 시스템의 연결을 승인할 수 있는 회사 시스템입니다. 오프사이트 시스템을 설정하려면 예 23–8을 참조하십시오.

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
  label "Off-site system with root certificate"
  local_id_type DNS
  local_id "main1.domain.org"
  local_addr 192.168.0.100

# CA's public certificate ensures trust,
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha1}
```

```
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1}
p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg aes auth_alg sha1}
}
```

예 23–8 IPsec로 NAT 뒤에 있는 시스템 구성

다음 예에서는 CA의 공개 인증서가 모바일 시스템 및 중앙 시스템에 배치됩니다. `mobile1`은 자택에서 회사 본사에 연결하고 있습니다. 인터넷 서비스 제공업체(ISP) 네트워크는 NAT 박스를 사용하여 ISP가 `mobile1`에 개인 주소를 지정할 수 있도록 합니다. 그러면 NAT 박스는 다른 ISP 네트워크 노드와 공유되는 공용 IP 주소로 개인 주소를 변환합니다. 회사 본사는 NAT 뒤에 없습니다. 회사 본사에서 컴퓨터를 설정하려면 예 23–7을 참조하십시오.

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters
#
# Find CRLs by URI, URL, or LDAP
# Use CRL from organization's URI
use_http
#
# Use web proxy
proxy "http://cache1.domain.org:8080/"
#
# Use LDAP server
ldap_server   "ldap1.domain.org,ldap2.domain.org:389"
#
# List CA-signed certificate
cert_root "C=US, O=ExamplePKI Inc, OU=PKI-Example, CN=Example PKI"
#
# Rule for off-site systems with root certificate
{
  label "Off-site mobile1 with root certificate"
  local_id_type DNS
  local_id "mobile1.domain.org"
  local_addr 0.0.0.0/0

# Find main1 and trust the root certificate
  remote_id "main1.domain.org"
  remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

예 23–9 모바일 시스템의 자체 서명된 인증서 승인

다음 예에서는 자체 서명된 인증서가 발급되었으며 모바일 및 중앙 시스템에 배치됩니다. main1은 오프사이트 시스템의 연결을 승인할 수 있는 회사 시스템입니다. 오프사이트 시스템을 설정하려면 예 23–10을 참조하십시오.

```
## /etc/hosts on main1
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on main1
# Keep everyone out unless they use this IPsec policy:
{} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on main1
# Global parameters
#
# Self-signed certificates - trust me and enumerated others
cert_trust    "DNS=main1.domain.org"
cert_trust    "jdoe@domain.org"
cert_trust    "user2@domain.org"
cert_trust    "user3@domain.org"
#
# Rule for off-site systems with trusted certificate
{
  label "Off-site systems with trusted certificates"
  local_id_type DNS
  local_id "main1.domain.org"
  local_addr 192.168.0.100

# Trust the self-signed certificates
# so allow any remote_id and any remote IP address.
  remote_id ""
  remote_addr 0.0.0.0/0

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

예 23–10 자체 서명된 인증서를 사용하여 중앙 시스템에 연결

다음 예에서는 mobile1이 자택에서 회사 본사에 연결하고 있습니다. 인증서가 발급되었으며 모바일 및 중앙 시스템에 배치됩니다. ISP 네트워크는 NAT 박스를 사용하여 ISP가 mobile1에 개인 주소를 지정할 수 있도록 합니다. 그러면 NAT 박스는 다른 ISP 네트워크 노드와 공유되는 공용 IP 주소로 개인 주소를 변환합니다. 회사 본사는 NAT 뒤에 없습니다. 회사 본사에서 컴퓨터를 설정하려면 예 23–9를 참조하십시오.

```
## /etc/hosts on mobile1
mobile1 10.1.3.3
main1 192.168.0.100

## /etc/inet/ipsecinit.conf on mobile1
# Find main1
```

```
{raddr 192.168.0.100} ipsec {encr_algs aes encr_auth_algs sha1 sa shared}

## /etc/inet/ike/ike.config on mobile1
# Global parameters

# Self-signed certificates - trust me and the central system
cert_trust    "jdoe@domain.org"
cert_trust    "DNS=main1.domain.org"
#
# Rule for off-site systems with trusted certificate
{
  label "Off-site mobile1 with trusted certificate"
  local_id_type email
  local_id "jdoe@domain.org"
  local_addr 0.0.0.0/0

# Find main1 and trust the certificate
  remote_id "main1.domain.org"
  remote_addr 192.168.0.100

p2_pfs 5

p1_xform
{auth_method rsa_sig oakley_group 5 encr_alg blowfish auth_alg sha1 }
}
```

연결된 하드웨어를 찾도록 IKE 구성(작업 맵)

다음 표에서는 연결된 하드웨어에 대한 정보를 IKE에 알리는 절차에 대해 설명합니다. IKE에서 하드웨어를 사용할 수 있으려면 연결된 하드웨어에 대한 정보를 IKE에 알려야 합니다. 하드웨어를 사용하려면 557 페이지 “공개 키 인증서로 IKE 구성”의 하드웨어 절차를 따릅니다.

주 – 칩 내장 하드웨어에 대해서는 IKE에 알릴 필요가 없습니다. 예를 들어, UltraSPARC T2 프로세서는 암호화 가속을 제공합니다. 칩 내장 가속기를 찾도록 IKE를 구성할 필요가 없습니다.

작업	설명	수행 방법
Sun Crypto Accelerator 1000 보드로 IKE 키 작업을 오프로드합니다.	IKE를 PKCS #11 라이브러리에 연결합니다.	581 페이지 “Sun Crypto Accelerator 1000 보드를 찾도록 IKE를 구성하는 방법”
IKE 키 작업을 오프로드하고 Sun Crypto Accelerator 4000 보드에 키를 저장합니다.	IKE를 PKCS #11 라이브러리에 연결하고 연결된 하드웨어 이름을 나열합니다.	582 페이지 “Sun Crypto Accelerator 4000 보드를 찾도록 IKE를 구성하는 방법”

연결된 하드웨어를 찾도록 IKE 구성

공개 키 인증서도 연결된 하드웨어에 저장할 수 있습니다. Sun Crypto Accelerator 1000 보드는 저장소만 제공합니다. Sun Crypto Accelerator 4000 및 Sun Crypto Accelerator 6000 보드는 저장소를 제공하고 공개 키 작업을 사용으로 설정하면 시스템에서 보드로 오프로드할 수 있습니다.

▼ Sun Crypto Accelerator 1000 보드를 찾도록 IKE를 구성하는 방법

시작하기 전에 다음 절차에서는 Sun Crypto Accelerator 1000 보드가 시스템에 연결된 것으로 간주합니다. 또한 절차에서는 보드용 소프트웨어가 설치되었으며 소프트웨어가 구성된 것으로 간주합니다. 지침은 **Sun Crypto Accelerator 1000 Board Version 2.0 Installation and User's Guide** (`http://download.oracle.com/docs/cd/E19412-01/819-0425-11/819-0425-11.pdf`)를 참조하십시오.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 PKCS #11 라이브러리가 연결되어 있는지 확인합니다.

다음 명령을 입력하여 PKCS #11 라이브러리가 연결되었는지 여부를 확인합니다.

```
# ikeadm get stats
Phase 1 SA counts:
Current:   initiator:          0   responder:          0
Total:     initiator:          0   responder:          0
Attempted: initiator:          0   responder:          0
Failed:    initiator:          0   responder:          0
           initiator fails include 0 time-out(s)
PKCS#11 library linked in from /usr/lib/libpkcs11.so
#
```

3 Solaris 10 1/06: 이 릴리스부터 키를 소프트 토큰 키 저장소에 저장할 수 있습니다.

암호화 프레임워크에서 제공하는 키 저장소에 대한 자세한 내용은 `cryptoadm(1M)` 매뉴얼 페이지를 참조하십시오. 키 저장소의 사용 예는 Example 23–11을 참조하십시오.

▼ Sun Crypto Accelerator 4000 보드를 찾도록 IKE를 구성하는 방법

시작하기 전에 다음 절차에서는 Sun Crypto Accelerator 4000 보드가 시스템에 연결된 것으로 간주합니다. 또한 절차에서는 보드용 소프트웨어가 설치되었으며 소프트웨어가 구성된 것으로 간주합니다. 지침은 **Sun Crypto Accelerator 4000 Board Version 1.1 Installation and User's Guide** (`http://download.oracle.com/docs/cd/E19877-01/817-3693-10/817-3693-10.pdf`)를 참조하십시오.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 PKCS #11 라이브러리가 연결되어 있는지 확인합니다.

IKE는 라이브러리의 루틴을 사용하여 Sun Crypto Accelerator 4000 보드에서의 키 생성 및 키 저장을 처리합니다. 다음 명령을 입력하여 PKCS #11 라이브러리가 연결되었는지 여부를 확인합니다.

```
$ ikeadm get stats
...
PKCS#11 library linked in from /usr/lib/libpkcs11.so
$
```

주 – Sun Crypto Accelerator 4000 보드는 RSA에 대해 최대 2048비트의 키를 지원합니다. DSA의 경우 이 보드는 최대 1024비트의 키를 지원합니다.

---

3 연결된 Sun Crypto Accelerator 4000 보드에 대한 토큰 ID를 찾습니다.

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                        "
```

라이브러리가 32자의 토큰 ID(키 저장소 이름이라고도 함)를 반환합니다. 이 예에서는 `ikecert` 명령에 `Sun Metaslot` 토큰을 사용하여 IKE 키를 저장하고 속도를 향상시킬 수 있습니다.

토큰 사용 방법에 대한 지침은 567 페이지 "공개 키 인증서를 생성하여 하드웨어에 저장하는 방법"을 참조하십시오.

`ikecert` 명령을 통해 자동으로 후행 공백이 채워집니다.

예 23–11 Metaslot 토큰 찾기 및 사용

토큰은 디스크, 연결된 보드 또는 암호화 프레임워크가 제공하는 소프트 토큰 키 저장소에 저장할 수 있습니다. 소프트 토큰 키 저장소 토큰 ID는 다음과 유사할 수 있습니다.

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                    "
```

소프트 토큰 키 저장소에 대한 문장암호를 만들려면 `pktool(1)` 매뉴얼 페이지를 참조하십시오.

다음과 유사한 명령이 소프트 토큰 키 저장소에 인증서를 추가합니다. `Sun.Metaslot.cert`는 CA 인증서가 포함된 파일입니다.

```
# ikecert certdb -a -T "Sun Metaslot" < Sun.Metaslot.cert
Enter PIN for PKCS#11 token:       Type user:passphrase
```

▼ Sun Crypto Accelerator 6000 보드를 찾도록 IKE를 구성하는 방법

시작하기 전에 다음 절차에서는 Sun Crypto Accelerator 6000 보드가 시스템에 연결된 것으로 간주합니다. 또한 절차에서는 보드용 소프트웨어가 설치되었으며 소프트웨어가 구성된 것으로 간주합니다. 지침은 **Sun Crypto Accelerator 6000 Board Version 1.1 사용자 설명서** (`http://download.oracle.com/docs/cd/E19321-01/820-4144-12/820-4144-12.pdf`)를 참조하십시오.

1 시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 PKCS #11 라이브러리가 연결되어 있는지 확인합니다.

IKE는 라이브러리의 루틴을 사용하여 Sun Crypto Accelerator 6000 보드에서의 키 생성 및 키 저장을 처리합니다. 다음 명령을 입력하여 PKCS #11 라이브러리가 연결되었는지 여부를 확인합니다.

```
$ ikeadm get stats
...
```

```
PKCS#11 library linked in from /usr/lib/libpkcs11.so
$
```

3 연결된 Sun Crypto Accelerator 6000 보드에 대한 토큰 ID를 찾습니다.

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                    "
```

라이브러리가 32자의 토큰 ID(키 저장소 이름이라고도 함)를 반환합니다. 이 예에서는 `ikecert` 명령에 `Sun Metaslot` 토큰을 사용하여 IKE 키를 저장하고 속도를 향상시킬 수 있습니다.

토큰 사용 방법에 대한 지침은 567 페이지 “공개 키 인증서를 생성하여 하드웨어에 저장하는 방법”을 참조하십시오.

`ikecert` 명령을 통해 자동으로 후행 공백이 채워집니다.

예 23–12 Metaslot 토큰 찾기 및 사용

토큰은 디스크, 연결된 보드 또는 암호화 프레임워크가 제공하는 소프트 토큰 키 저장소에 저장할 수 있습니다. 소프트 토큰 키 저장소 토큰 ID는 다음과 유사할 수 있습니다.

```
$ ikecert tokens
Available tokens with library "/usr/lib/libpkcs11.so":

"Sun Metaslot                    "
```

소프트 토큰 키 저장소에 대한 문장암호를 만들려면 `pktool(1)` 매뉴얼 페이지를 참조하십시오.

다음과 유사한 명령이 소프트 토큰 키 저장소에 인증서를 추가합니다. `Sun.Metaslot.cert`는 CA 인증서가 포함된 파일입니다.

```
# ikecert certdb -a -T "Sun Metaslot" < Sun.Metaslot.cert
Enter PIN for PKCS#11 token:       Type user:passphrase
```

IKE 전송 매개변수 변경(작업 맵)

다음 표에서는 IKE용 전송 매개변수를 구성하는 절차에 대해 설명합니다.

작업	설명	수행 방법
키 협상의 효율성을 높입니다.	키 협상 매개변수를 변경합니다.	585 페이지 “Phase 1 IKE 키 협상 지속 시간을 변경하는 방법”

작업	설명	수행 방법
키 협상을 구성하여 전송의 지연을 허용합니다.	키 협상 매개변수를 연장합니다.	예 23–13
키 협상을 구성하여 성공 속도를 높이거나 실패를 신속하게 표시합니다.	키 협상 매개변수를 단축합니다.	예 23–14

IKE 전송 매개변수 변경

IKE가 키를 협상할 때 전송 속도가 협상의 성공 여부에 영향을 줄 수 있습니다. 일반적으로는 IKE 전송 매개변수의 기본값을 변경할 필요가 없습니다. 그러나 더티 회선을 통한 키 협상을 최적화하거나 문제를 재현할 때는 전송 값을 변경할 수 있습니다.

지속 시간이 길어지면 IKE를 사용으로 설정하여 불안정한 전송 회선을 통해 키를 협상할 수 있습니다. 초기 시도가 성공할 수 있도록 특정 매개변수를 연장할 수 있습니다. 초기 시도가 실패한 경우 다음에 시도할 때 약간의 간격을 두고 시간을 더 많이 제공하여 성공 확률을 높일 수 있습니다.

지속 시간이 짧으면 안정적인 전송 회선을 사용하도록 설정할 수 있습니다. 실패한 협상을 더 신속하게 재시도하여 협상 속도를 높일 수 있습니다. 문제가 진단되면 빨리 실패할 수 있도록 협상의 속도를 높이는 것이 좋습니다. 지속 시간이 짧으면 수명 주기 동안 Phase 1 SA를 사용하도록 설정할 수도 있습니다.

▼ Phase 1 IKE 키 협상 지속 시간을 변경하는 방법

1 **시스템 콘솔에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

주 – 원격으로 로그인하면 보안이 중요한 트래픽이 도청될 수 있습니다. 원격 로그인을 보호해도 시스템의 보안은 원격 로그인 세션 보안으로 약해집니다. 원격 로그인을 보호하려면 `ssh` 명령을 사용하십시오.

---

2 **각 시스템의 전역 전송 매개변수의 기본값을 변경합니다.**

각 시스템에서 `/etc/inet/ike/config` 파일의 Phase 1 지속 기간 매개변수를 수정합니다.

```
### ike/config file on      system

## Global parameters
#
## Phase 1 transform defaults
```

```
#
#expire_timer      300
#retry_limit         5
#retry_timer_init    0.5 (integer or float)
#retry_timer_max    30   (integer or float)
```

`expire_timer` 협상 시도가 삭제되기 전까지 아직 완료되지 않은 IKE Phase I 협상이 유지되는 시간(초)입니다. 기본적으로 시도는 30초 동안 유지됩니다.

`retry_limit` IKE 협상이 중지되기 전까지 재전송되는 횟수입니다. 기본적으로 IKE는 5번 시도합니다.

`retry_timer_init` 재전송의 초기 간격입니다. 이 간격은 `retry_timer_max` 값에 도달할 때까지 2배씩 늘어납니다. 초기 간격은 0.5초입니다.

`retry_timer_max` 재전송의 최대 간격(초)입니다. 재전송 간격은 이 제한 시간까지 증가하다가 중지합니다. 기본적으로 제한 시간은 30초입니다.

3 변경된 구성을 커널로 읽습니다.

- **Solaris 10 4/09 릴리스부터 `ike` 서비스를 새로 고칩니다.**

  ```
  # svcadm refresh svc:/network/ipsec/ike
  ```

- **Solaris 10 4/09 이전 릴리스를 실행 중인 경우에는 시스템을 재부트합니다.**

  ```
  # init 6
  ```

  또는 `in.iked` 데몬을 중지한 후 시작합니다.

예 23–13 IKE Phase 1 협상 시간 연장

다음 예에서 시스템은 트래픽이 많은 전송 회선을 통해 해당 IKE 피어에 연결됩니다. 원래 설정은 파일의 주석에 있습니다. 새 설정은 협상 시간을 연장합니다.

```
### ike/config file on partym
## Global Parameters
#
## Phase 1 transform defaults
#expire_timer   300
#retry_limit      5
#retry_timer_init 0.5 (integer or float)
#retry_timer_max 30   (integer or float)
#
expire_timer  600
retry_limit  10
retry_timer_init  2.5
retry_timer_max  180
```

예 23–14 IKE Phase 1 협상 시간 단축

다음 예에서 시스템은 트래픽이 적은 고속 회선을 통해 해당 IKE 피어에 연결됩니다. 원래 설정은 파일의 주석에 있습니다. 새 설정은 협상 시간을 단축합니다.

```
### ike/config file on partym
## Global Parameters
#
## Phase 1 transform defaults
#expire_timer   300
#retry_limit      5
#retry_timer_init 0.5 (integer or float)
#retry_timer_max 30   (integer or float)
#
expire_timer  120
retry_timer_init  0.20
```

◆ ◆ ◆ 24 장

24 Internet Key Exchange(참조)

이 장은 IKE에 대한 다음 참조 정보를 포함합니다.

IKE 구현 지침은 23 장, “IKE 구성(작업)”을 참조하십시오. 개요 정보는 22 장, “Internet Key Exchange(개요)”를 참조하십시오.

IKE 서비스

svc:/network/ipsec/ike:default **서비스** – SMF(서비스 관리 기능)는 IKE를 관리하기 위해 ike 서비스를 제공합니다. 기본적으로 이 서비스는 사용 안함으로 설정됩니다. 이 서비스를 사용으로 설정하기 전에 IKE 구성 파일 /etc/inet/ike/config를 만들어야 합니다.

다음 ike 서비스 등록 정보를 구성할 수 있습니다.

- config_file **등록 정보** – IKE 구성 파일의 위치입니다. 초기 값은 /etc/inet/ike/config입니다.
- debug_level **등록 정보** – in.iked 데몬의 디버깅 레벨입니다. 초기 값은 op 또는 operational입니다. 가능한 값은 ikeadm(1M) 매뉴얼 페이지에서 **객체 유형** 아래의 디버그 레벨 테이블을 참조하십시오.
- admin_privilege **등록 정보** – in.iked 데몬의 권한 레벨입니다. 초기 값은 base입니다. 다른 값으로 modkeys 및 keymat가 있습니다. 세부 정보는 591 페이지 “ikeadm 명령”을 참조하십시오.

SMF에 대한 자세한 내용은 **Oracle Solaris 관리: 기본 관리**의 18 장, “서비스 관리(개요)”를 참조하십시오. 또한 smf(5), svcadm(1M), svccfg(1M) 매뉴얼 페이지를 참조하십시오.

IKE 데몬

in.iked 데몬은 Oracle Solaris 시스템에서 IPsec에 대한 암호화 키 관리를 자동화합니다. 데몬은 동일한 프로토콜을 실행 중인 원격 시스템과 협상하여 보안 연관(SA)에 대한 인증된 키 입력 자료를 안전한 방식으로 제공합니다. 안전하게 통신하려는 모든 시스템에서 데몬을 실행 중이어야 합니다.

기본적으로 svc:/network/ipsec/ike:default 서비스는 사용으로 설정되지 않습니다. /etc/inet/ike/config 파일을 구성하고 ike 서비스를 사용으로 설정한 후에 시스템 부트 시 in.iked 데몬이 실행됩니다.

IKE 데몬을 실행할 때 시스템이 Phase 1 교환에서 피어 IKE 엔티티로 자체 인증합니다. 피어는 인증 방법과 마찬가지로 IKE 정책 파일에 정의됩니다. 그런 다음 데몬이 Phase 2 교환에 대한 키를 설정합니다. 정책 파일에 지정된 간격으로 IKE 키를 자동으로 새로 고칩니다. in.iked 데몬이 네트워크에서 들어오는 IKE 요청과 PF_KEY 소켓을 통과하는 아웃바운드 트래픽 요청을 수신합니다. 자세한 내용은 pf_key(7P) 매뉴얼 페이지를 참조하십시오.

두 가지 명령이 IKE 데몬을 지원합니다. ikeadm 명령을 사용하여 IKE 정책을 확인하고 일시적으로 수정할 수 있습니다. IKE 정책을 영구적으로 수정하려면 ike 서비스의 등록 정보를 수정합니다. IKE 서비스의 등록 정보를 수정하려면 492 페이지 “IKE 및 IPsec 서비스를 관리하는 방법”을 참조하십시오.

ikecert 명령을 사용하여 공개 키 데이터베이스를 보고 관리할 수 있습니다. 이 명령은 로컬 데이터베이스인 ike.privatekeys 및 publickeys를 관리합니다. 또한 이 명령은 공개 키 작업 및 하드웨어의 공개 키 저장소를 관리합니다.

IKE 구성 파일

IKE 구성 파일 /etc/inet/ike/config는 IPsec 정책 파일 /etc/inet/ipsecinit.conf에서 보호되는 인터페이스의 키를 관리합니다.

IKE의 키 관리에는 규칙 및 전역 매개변수가 관여합니다. IKE 규칙은 키 입력 자료를 보안하는 시스템 또는 네트워크를 식별합니다. 또한 규칙은 인증 방법을 지정합니다. 전역 매개변수에는 연결된 하드웨어 가속기의 경로와 같은 항목이 포함됩니다. IKE 정책 파일의 예는 546 페이지 “미리 공유한 키로 IKE 구성(작업 맵)”을 참조하십시오. IKE 정책 항목의 예제 및 설명은 ike.config(4) 매뉴얼 페이지를 참조하십시오.

IKE가 지원하는 IPsec SA는 IPsec 구성 파일 /etc/inet/ipsecinit.conf의 정책에 따라 IP 데이터그램을 보호합니다. IKE 정책 파일은 IPsec SA를 만들 때 PFS(완전 순방향 비밀성)의 사용 여부를 결정합니다.

/etc/inet/ike/config 파일은 RSA Security Inc.의 PKCS #11 암호화 토큰 인터페이스(Cryptoki) 표준에 따라 구현되는 라이브러리의 경로를 포함할 수 있습니다. IKE는 이 PKCS #11 라이브러리를 사용하여 키 가속 및 키 저장을 위한 하드웨어에 액세스합니다.

ike/config 파일에 대한 보안 고려 사항은 ipsecinit.conf 파일의 고려 사항과 비슷합니다. 세부 정보는 531 페이지 “ipsecinit.conf 및 ipsecconf에 대한 보안 고려 사항”을 참조하십시오.

ikeadm 명령

ikeadm 명령을 사용하여 다음을 수행할 수 있습니다.

- IKE 데몬 프로세스의 여러 측면을 봅니다.
- IKE 데몬으로 전달되는 매개변수를 변경합니다.
- Phase 1 교환 중 SA 생성에 대한 통계를 표시합니다.
- IKE 프로세스를 디버깅합니다.
- IKE 상태의 여러 측면을 봅니다.
- IKE 데몬의 등록 정보를 변경합니다.
- Phase 1 교환 중 SA 생성에 대한 통계를 표시합니다.
- IKE 프로토콜 교환을 디버그합니다.

이 명령의 옵션에 대한 예제 및 전체 설명은 ikeadm(1M) 매뉴얼 페이지를 참조하십시오.

실행 중인 IKE 데몬의 권한 레벨에 따라 IKE 데몬의 어떤 측면을 보고 수정할 수 있는지 결정됩니다. 3단계 권한 레벨이 가능합니다.

base 레벨	키 입력 자료를 보거나 수정할 수 없습니다. base 레벨이 기본 권한 레벨입니다.
modkeys 레벨	미리 공유한 키를 제거, 변경, 추가할 수 있습니다.
keymat 레벨	ikeadm 명령을 사용하여 실제 키 입력 자료를 볼 수 있습니다.

일시적 권한 변경은 ikeadm 명령을 사용할 수 있습니다. 영구적 변경은 ike 서비스의 admin_privilege 등록 정보를 변경합니다. 절차는 492 페이지 “IKE 및 IPsec 서비스를 관리하는 방법”을 참조하십시오.

ikeadm 명령에 대한 보안 고려 사항은 ipseckey 명령의 고려 사항과 비슷합니다. 세부 정보는 533 페이지 “ipseckey에 대한 보안 고려 사항”을 참조하십시오.

IKE 미리 공유한 키 파일

미리 공유한 키를 수동으로 만들 때 /etc/inet/secret 디렉토리의 파일에 키가 저장됩니다. ike.preshared 파일은 ISAKMP(Internet Security Association and Key Management Protocol) SA에 대한 미리 공유한 키를 포함합니다. ipseckeys 파일은 IPsec SA에 대한 미리 공유한 키를 포함합니다. 파일은 0600에서 보호됩니다. secret 디렉토리는 0700에서 보호됩니다.

- ike/config 파일에서 미리 공유한 키를 요구하도록 구성할 때 ike.preshared 파일을 만듭니다. ike.preshared 파일에 IKE 인증인 ISAKMP SA에 대한 키 입력 자료를 입력합니다. 미리 공유한 키를 사용하여 Phase 1 교환을 인증하므로 in.iked 데몬을 시작하기 전에 파일이 유효해야 합니다.
- ipseckeys 파일은 IPsec SA에 대한 키 입력 자료를 포함합니다. 파일 수동 관리의 예는 485 페이지 "수동으로 IPsec 보안 연관을 만드는 방법"을 참조하십시오. IKE 데몬은 이 파일을 사용하지 않습니다. IPsec SA에 대해 IKE가 생성하는 키 입력 자료는 커널에 저장됩니다.

IKE 공개 키 데이터베이스 및 명령

ikecert 명령은 로컬 시스템의 공개 키 데이터베이스를 조작합니다. ike/config 파일에 공개 키 인증서가 필요할 때 이 명령을 사용합니다. IKE는 이러한 데이터베이스를 사용하여 Phase 1 교환을 인증하므로 in.iked 데몬을 활성화하기 전에 데이터베이스를 채워야 합니다. 세 가지 하위 명령 certlocal, certdb, certrldb가 각각 세 데이터베이스를 처리합니다.

ikecert 명령은 키 저장소도 처리합니다. 키는 디스크, 연결된 Sun Crypto Accelerator 6000/Sun Crypto Accelerator 4000 보드 또는 소프트 토큰 키 저장소에 저장할 수 있습니다. 암호화 프레임워크의 metaslot을 사용하여 하드웨어 장치와 통신할 때 softtoken 키 저장소를 사용할 수 있습니다. ikecert 명령은 PKCS #11 라이브러리를 사용하여 키 저장소를 찾습니다.

- **Solaris 10 1/06**: 이 릴리스부터 라이브러리를 지정할 필요가 없습니다. 기본적으로 PKCS #11 라이브러리는 /usr/lib/libpkcs11.so입니다.
- **Solaris 10**: 이 릴리스에서는 PKCS #11 항목을 지정해야 합니다. 그렇지 않으면 ikecert 명령의 -T 옵션을 사용할 수 없습니다. 항목은 다음과 같이 표시됩니다.

  ```
  pkcs11_path "/opt/SUNWconn/cryptov2/lib/libvpkcs11.so"
  ```

자세한 내용은 ikecert(1M) 매뉴얼 페이지를 참조하십시오. metaslot 및 softtoken 키 저장소에 대한 내용은 cryptoadm(1M) 매뉴얼 페이지를 참조하십시오.

ikecert tokens 명령

tokens 인수는 사용 가능한 토큰 ID를 나열합니다. 토큰 ID를 통해 ikecert certlocal 및 ikecert certdb 명령에서 공개 키 인증서 및 인증서 요청을 생성할 수 있습니다. 또한 암호화 프레임워크에서 소프트 토큰 키 저장소 또는 연결된 Sun Crypto Accelerator 6000/Sun Crypto Accelerator 4000 보드에 인증서 및 인증서 요청을 저장할 수 있습니다. ikecert 명령은 PKCS #11 라이브러리를 사용하여 인증서 저장소를 찾습니다.

ikecert certlocal 명령

certlocal 하위 명령은 개인 키 데이터베이스를 관리합니다. 이 하위 명령의 옵션을 사용하여 개인 키를 추가, 보기, 제거할 수 있습니다. 또한 이 하위 명령은 자체 서명된 인증서 또는 인증서 요청을 만듭니다. -ks 옵션은 자체 서명된 인증서를 만듭니다. -kc 옵션은 인증서 요청을 만듭니다. 키는 /etc/inet/secret/ike.privatekeys 디렉토리에서 시스템에 저장되거나, -T 옵션을 사용하여 연결된 하드웨어에 저장됩니다.

개인 키를 만들 때 ikecert certlocal 명령의 옵션이 ike/config 파일의 항목과 관련을 맺어야 합니다. ikecert 옵션과 ike/config 항목 사이의 관련성이 다음 표에 표시됩니다.

표 24–1 ikecert 옵션과 ike/config 항목 사이의 관련성

ikecert 옵션	ike/config 항목	설명
-A *subject-alternate-name*	cert_trust *subject-alternate-name*	인증서를 고유하게 식별하는 별명입니다. 가능한 값은 IP 주소, 전자 메일 주소 또는 도메인 이름입니다.
-D *X.509-distinguished-name*	*X.509-distinguished-name*	국가(C), 조직 이름(ON), 조직 단위(OU), 공통 이름(CN)을 포함하는 인증 기관의 전체 이름입니다.
-t dsa-sha1	auth_method dsa_sig	RSA보다 약간 느린 인증 방법입니다.
-t rsa-md5 및 -t rsa-sha1	auth_method rsa_sig	DSA보다 약간 빠른 인증 방법입니다. RSA 공개 키는 가장 큰 페이로드를 암호화할 만큼 충분히 커야 합니다. 일반적으로 X.509 식별 이름과 같은 신원 페이로드가 가장 큰 페이로드입니다.
-t rsa-md5 및 -t rsa-sha1	auth_method rsa_encrypt	RSA 암호화는 도청자로부터 IKE의 신원을 숨기지만 IKE 피어가 서로의 공개 키를 알아야 합니다.
-T	pkcs11_path	PKCS #11 라이브러리는 Sun Crypto Accelerator 1000 보드, Sun Crypto Accelerator 6000 보드 및 Sun Crypto Accelerator 4000 보드에서 키 가속화를 처리합니다. 이 라이브러리는 Sun Crypto Accelerator 6000 및 Sun Crypto Accelerator 4000 보드의 키 저장소를 처리하는 토큰도 제공합니다.

ikecert certlocal -kc 명령으로 인증서 요청을 발행하면 명령의 출력을 PKI 조직이나 인증 기관(CA)으로 보냅니다. 회사에서 고유의 PKI를 실행하는 경우 PKI 관리자에게 출력을 보냅니다. 그런 다음 PKI 조직, CA 또는 PKI 관리자가 인증서를 만듭니다. PKI 또는 CA가 반환하는 인증서는 certdb 하위 명령으로 입력됩니다. PKI가 반환하는 CRL(인증서 해지 목록)은 certrldb 하위 명령으로 입력됩니다.

ikecert certdb 명령

certdb 하위 명령은 공개 키 데이터베이스를 관리합니다. 이 하위 명령의 옵션을 사용하여 인증서 및 공개 키를 추가, 보기, 제거할 수 있습니다. 이 명령은 원격 시스템에서 ikecert certlocal -ks 명령으로 생성된 인증서를 입력으로 받아들입니다. 절차는 557 페이지 "자체 서명된 공개 키 인증서로 IKE를 구성하는 방법"을 참조하십시오. 또한 이 명령은 PKI 또는 CA로부터 받은 인증서를 입력으로 받아들입니다. 절차는 562 페이지 "CA가 서명한 인증서로 IKE를 구성하는 방법"을 참조하십시오.

인증서 및 공개 키는 /etc/inet/ike/publickeys 디렉토리에서 시스템에 저장됩니다. -T 옵션은 연결된 하드웨어에 인증서, 개인 키, 공개 키를 저장합니다.

ikecert certrldb 명령

certrldb 하위 명령은 CRL(인증서 해지 목록) 데이터베이스인 /etc/inet/ike/crls를 관리합니다. CRL 데이터베이스는 공개 키에 대한 해지 목록을 유지 관리합니다. 이 목록에는 더 이상 유효하지 않은 인증서가 있습니다. PKI에서 CRL을 제공할 때 ikecert certrldb 명령을 사용하여 CRL 데이터베이스에 CRL을 설치할 수 있습니다. 절차는 571 페이지 "인증서 해지 목록 처리 방법"을 참조하십시오.

/etc/inet/ike/publickeys 디렉토리

/etc/inet/ike/publickeys 디렉토리는 공개-개인 키 쌍의 공개 부분과 해당 인증서를 파일이나 **슬롯**에 넣습니다. 디렉토리는 0755에서 보호됩니다. ikecert certdb 명령은 디렉토리를 채웁니다. -T 옵션은 publickeys 디렉토리가 아닌 Sun Crypto Accelerator 6000 또는 Sun Crypto Accelerator 4000 보드에 키를 저장합니다.

슬롯은 다른 시스템에서 생성된 인증서의 X.509 식별 이름을 인코딩된 형태로 포함합니다. 자체 서명된 인증서를 사용하는 경우 원격 시스템의 관리자로부터 받은 인증서를 명령의 입력으로 사용합니다. CA의 인증서를 사용하는 경우 CA에서 서명한 두 인증서를 이 데이터베이스로 설치합니다. CA로 보낸 인증서 서명 요청에 준하는 인증서를 설치합니다. 또한 CA의 인증서를 설치합니다.

/etc/inet/secret/ike.privatekeys 디렉토리

/etc/inet/secret/ike.privatekeys 디렉토리는 공개-개인 키 쌍의 일부인 개인 키 파일을 보유합니다. 디렉토리는 0700에서 보호됩니다. ikecert certlocal 명령은 ike.privatekeys 디렉토리를 채웁니다. 대응하는 공개 키, 자체 서명된 인증서 또는 CA를 설치할 때까지 개인 키는 효과가 없습니다. 대응하는 공개 키는 /etc/inet/ike/publickeys 디렉토리 또는 Sun Crypto Accelerator 6000/Sun Crypto Accelerator 4000 보드에 저장됩니다.

/etc/inet/ike/crls 디렉토리

/etc/inet/ike/crls 디렉토리는 CRL(인증서 해지 목록) 파일을 포함합니다. 각 파일은 /etc/inet/ike/publickeys 디렉토리의 공개 인증서 파일에 해당합니다. PKI 조직은 해당 인증서에 대한 CRL을 제공합니다. ikecert certrldb 명령을 사용하여 데이터베이스를 채울 수 있습니다.

◆ ◆ ◆ 25 장

25 Oracle Solaris의 IP 필터(개요)

이 장에서는 Oracle Solaris 기능인 IP 필터의 개요를 제공합니다. IP 필터 작업은 26 장, “IP 필터(작업)”를 참조하십시오.

이 장은 다음 정보를 포함합니다.

IP 필터의 새로운 기능

이 절에서는 IP 필터의 새로운 기능에 대해 설명합니다.

Oracle Solaris 릴리스의 새로운 기능에 대한 전체 목록 및 설명은 **Oracle Solaris 10 1/13 새로운 기능**을 참조하십시오.

패킷 필터링을 위한 패킷 필터 후크

Solaris 10 7/07 릴리스부터: 이제 Oracle Solaris의 네트워크 패킷을 필터링하는 데 패킷 필터 후크가 사용됩니다. 이 기능은 시스템 관리에 다음과 같은 이점을 제공합니다.

- 패킷 필터 후크는 IP 필터의 구성을 간소화합니다.
- 영역 간 패킷 필터링이 지원됩니다.

- 필터 후크를 사용하면 IP 필터의 성능을 향상시킬 수 있습니다.

이러한 후크에 대한 자세한 내용은 608 페이지 “패킷 필터 후크”를 참조하십시오. 패킷 필터 후크와 관련된 작업에 대해서는 26 장, “IP 필터(작업)”를 참조하십시오.

IP 필터용 IPv6 패킷 필터링

Solaris 6/06 OS: 전체 또는 일부 네트워크 기반구조가 IPv6으로 구성된 시스템의 관리자를 위해 IPv6 패킷 필터링을 포함하도록 IP 필터가 향상되었습니다. IPv6 패킷 필터링은 소스/대상 IPv6 주소, IPv6 주소를 포함하는 풀 및 IPv6 확장 헤더를 기준으로 필터링을 수행할 수 있습니다.

IPv6과 함께 사용하도록 `-6` 옵션이 `ipf` 명령 및 `ipfstat` 명령에 추가되었습니다. `ipmon` 및 `ippool` 명령에 대한 명령줄 인터페이스에는 변경 사항이 없지만 이 명령도 IPv6을 지원합니다. IPv6 패킷의 로깅을 위해 `ipmon` 명령이 향상되었으며 `ippool` 명령은 풀의 IPv6 주소를 포함하도록 지원합니다.

자세한 내용은 IP 필터용 IPv6을 참조하십시오. IPv6 패킷 필터링과 관련된 작업에 대해서는 26 장, “IP 필터(작업)”를 참조하십시오.

또한 IP 필터의 NAT(Network Address Translation) 기능에서도 IPv6이 지원됩니다. NAT에 대한 자세한 내용은 606 페이지 “IP 필터의 NAT 기능 사용”을 참조하십시오.

IP 필터 소개

Oracle Solaris의 IP 필터 기능은 OS의 SunScreen 방화벽을 대체합니다. SunScreen 방화벽과 마찬가지로 IP 필터는 Stateful 패킷 필터링 및 NAT(Network Address Translation)를 제공합니다. IP 필터에는 Stateless 패킷 필터링을 비롯하여 주소 풀 생성 및 관리 기능도 포함되어 있습니다.

패킷 필터링은 네트워크 기반 공격에 대비한 기본적인 보호를 제공합니다. IP 필터는 IP 주소, 포트, 프로토콜, 네트워크 인터페이스 및 트래픽 방향을 기준으로 필터링을 수행할 수 있습니다. 개별 소스 IP 주소, 대상 IP 주소, IP 주소 범위 또는 주소 풀을 기준으로도 필터링을 수행할 수 있습니다.

IP 필터는 오픈 소스 IP 필터 소프트웨어에서 파생되었습니다. 오픈 소스 IP 필터에 대한 라이센스 약관, 직권 및 저작권 설명을 볼 수 있는 기본 경로는 `/usr/lib/ipf/IPFILTER.LICENCE`입니다. Oracle Solaris가 기본 경로 이외의 다른 경로에 설치된 경우 설치된 위치의 파일에 액세스할 수 있도록 지정된 경로를 수정하십시오.

오픈 소스 IP 필터에 대한 정보 소스

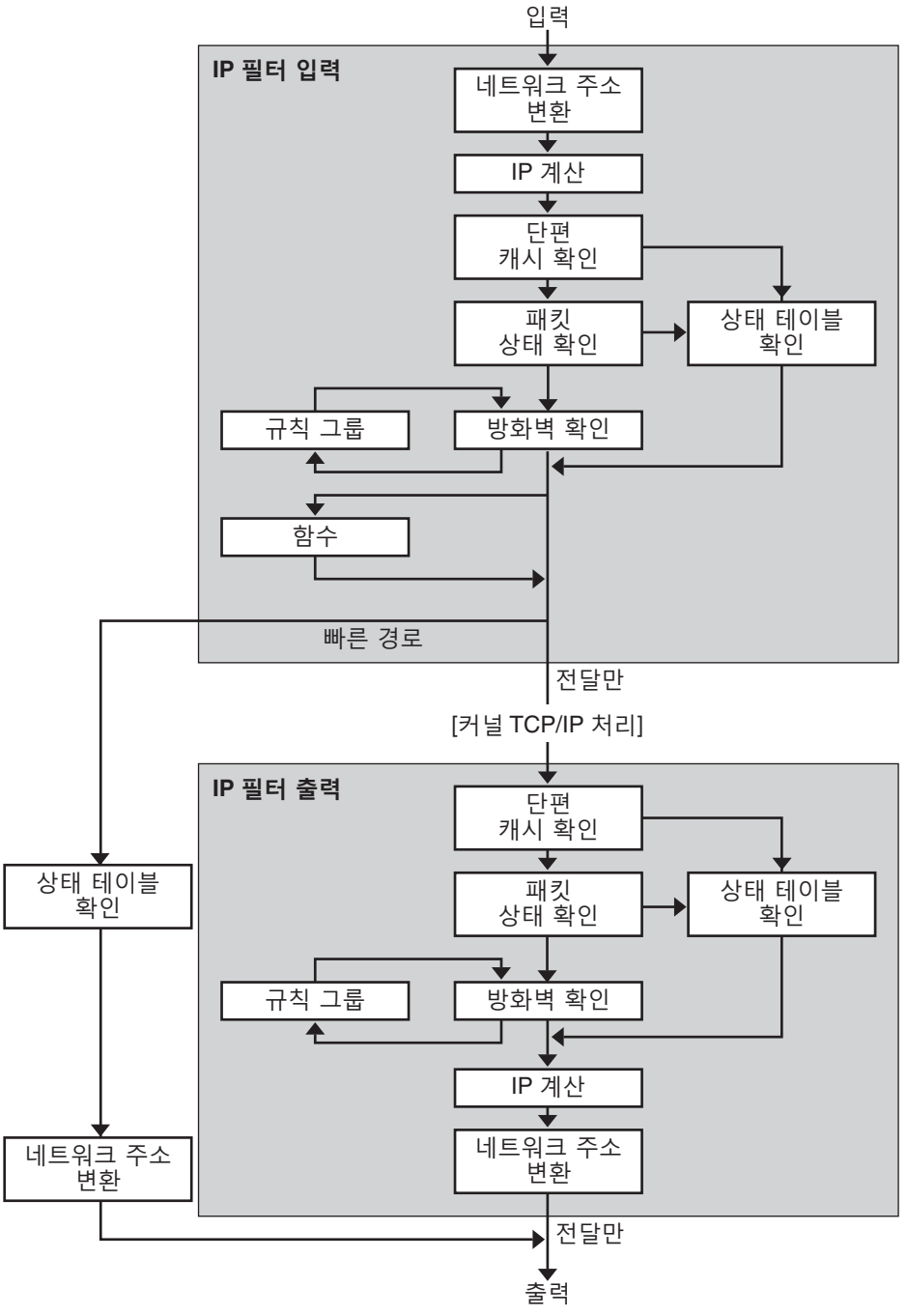
Darren Reed의 오픈 소스 IP 필터 소프트웨어 홈 페이지는 `http://coombs.anu.edu.au/~avalon/ip-filter.html`에서 확인할 수 있습니다. 이 사이트에서는 “IP Filter Based Firewalls HOWTO”(Brendan Conoboy and Erik Fichtner, 2002) 자습서에 대한 링크를

비롯하여 오픈 소스 IP 필터에 대한 정보를 제공합니다. 이 자습서는 BSD UNIX 환경에서 방화벽을 구축하는 단계별 지침을 제공합니다. 자습서는 BSD UNIX 환경에 대해 작성된 것이기는 하지만 IP 필터 기능 구성과도 관련이 있습니다.

IP 필터 패킷 처리

IP 필터는 패킷이 처리되는 일련의 단계를 실행합니다. 다음 다이어그램에서는 패킷 처리 단계 및 필터링과 TCP/IP 프로토콜 스택의 통합 방법을 보여 줍니다.

그림 25–1 패킷 처리 순서



패킷 처리 순서는 다음과 같습니다.

- **NAT(Network Address Translation)**

  개인 IP 주소를 다른 공용 주소로 변환하거나 다중 개인 주소의 별칭을 단일 공용 주소로 변환합니다. 기존 네트워크가 있으며 인터넷에 액세스해야 하는 조직에서는 NAT를 통해 IP 주소 소모 문제를 해결할 수 있습니다.

- **IP 계산**

  통과하는 바이트 수를 기록하여 입력 및 출력 규칙을 별도로 설정할 수 있습니다. 규칙 일치가 발생할 때마다 패킷 바이트 수가 규칙에 추가되므로 연속 통계를 수집할 수 있습니다.

- **단편 캐시 확인**

  현재 트래픽의 다음 패킷이 단편이고 이전 패킷이 허용된 경우 상태 테이블 및 규칙 확인이 무시되어 패킷 단편도 허용됩니다.

- **패킷 상태 확인**

  `keep state`가 규칙에 포함된 경우 규칙이 `pass`를 의미하는지 아니면 `block`을 의미하는지에 따라 지정된 세션의 모든 패킷이 자동으로 전달 또는 차단됩니다.

- **방화벽 확인**

  IP 필터를 통해 패킷이 허용될지 여부에 따라 커널 TCP/IP 루틴으로 들어오거나 네트워크를 통해 나가는 입력 및 출력 규칙을 별도로 설정할 수 있습니다.

- **그룹**

  그룹을 통해 트리 형식으로 규칙 세트를 작성할 수 있습니다.

- **함수**

  함수는 수행할 작업입니다. 가능한 함수로는 `block`, `pass`, `literal` 및 `send ICMP response`가 있습니다.

- **빠른 경로**

  빠른 경로는 경로 지정을 위해 패킷이 UNIX IP 스택으로 전달되지 않도록 IP 필터에 신호를 보냅니다. 해당 스택으로 전달될 경우 TTL이 줄어듭니다.

- **IP 인증**

  이중 처리를 방지하기 위해 인증된 패킷은 방화벽 루프를 통해 한 번만 전달됩니다.

IP 필터 사용 지침

- IP 필터는 SMF 서비스 svc:/network/pfil 및 svc:/network/ipfilter를 통해 관리됩니다. SMF의 전체 개요는 **Oracle Solaris 관리: 기본 관리**의 18 장, “서비스 관리(개요)”를 참조하십시오. SMF와 관련된 단계별 절차에 대한 자세한 내용은 **Oracle Solaris 관리: 기본 관리**의 19 장, “서비스 관리(작업)”를 참조하십시오.
- IP 필터를 사용하려면 구성 파일을 직접 편집해야 합니다.

- IP 필터는 Oracle Solaris의 일부로 설치됩니다. 기본적으로 새 설치 후 IP 필터가 활성화되지 않습니다. 필터링을 구성하려면 구성 파일을 편집하고 수동으로 IP 필터를 활성화해야 합니다. 시스템을 재부트하거나 `ifconfig` 명령으로 인터페이스를 연결하여 필터링을 활성화할 수 있습니다. 자세한 내용은 `ifconfig`(1M) 매뉴얼 페이지를 참조하십시오. IP 필터를 사용으로 설정하는 것과 관련된 작업은 613 페이지 "IP 필터 구성"을 참조하십시오.
- IP 필터를 관리하려면 IP 필터 관리 권한 프로파일이 포함된 역할의 사용자 또는 수퍼유저로 로그인할 수 있어야 합니다. 만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.
- IPMP(IP Network Multipathing)는 Stateless 필터링만 지원합니다.

  IP 필터가 IPMP 그룹과 주고 받는 트래픽에 대해 Stateless 필터링을 수행하도록 하려면 `ipmp_hook_emulation` 매개변수를 설정해야 합니다. 기본적으로 매개변수는 `0`으로 설정되어 있습니다. 기본값을 사용할 경우 IP 필터가 IPMP 그룹에 속하는 물리적 인터페이스에서 트래픽에 대해 Stateful 패킷 검사를 수행할 수 없습니다. IPMP 패킷 필터링을 사용으로 설정하려면 다음 명령을 실행하십시오.

  ```
  ndd -set /dev/ip ipmp_hook_emulation 1
  ```
- Oracle Solaris Cluster 소프트웨어의 경우 확장 가능한 서비스에 대해서는 IP 필터를 통한 필터링을 지원하지 않지만 페일오버 서비스에 대해서는 IP 필터를 지원합니다. 클러스터에서 IP 필터를 구성하는 경우 지침 및 제한 사항은 **Oracle Solaris Cluster 소프트웨어 설치 설명서**의 "Oracle Solaris OS 기능 제한 사항"을 참조하십시오.
- 시스템의 다른 영역에 대한 가상 라우터로 작동하는 영역에서 IP 필터 규칙이 구현된 경우 영역 간의 필터링이 지원됩니다.

IP 필터 구성 파일 사용

IP 필터를 사용하여 방화벽 서비스 또는 NAT(Network Address Translation)를 제공할 수 있습니다. 로드 가능한 구성 파일을 통해 IP 필터를 구현할 수 있습니다. IP 필터에는 `/etc/ipf`라는 디렉토리가 있습니다. `ipf.conf`, `ipnat.conf` 및 `ippool.conf`라는 구성 파일을 만들어 `/etc/ipf` 디렉토리에 저장할 수 있습니다. 이러한 파일은 `/etc/ipf` 디렉토리에 상주한 경우 부트 프로세스 중 자동으로 로드됩니다. 구성 파일을 다른 위치에 저장하고 수동으로 파일을 로드할 수도 있습니다. 구성 파일 예는 644 페이지 "IP 필터 구성 파일 만들기 및 편집"을 참조하십시오.

IP 필터 규칙 세트 사용

방화벽을 관리하려면 IP 필터를 사용하여 네트워크 트래픽 필터링에 사용할 규칙 세트를 지정하십시오. 다음 유형의 규칙 세트를 만들 수 있습니다.

- 패킷 필터링 규칙 세트
- NAT(Network Address Translation) 규칙 세트

또한 IP 주소 그룹을 참조할 주소 풀을 만들 수 있습니다. 그런 다음 나중에 규칙 세트에서 이러한 풀을 사용할 수 있습니다. 주소 풀을 사용하면 규칙 처리 속도가 빨라집니다. 또한 주소 풀을 사용하면 큰 주소 그룹을 간편하게 관리할 수 있습니다.

IP 필터의 패킷 필터링 기능 사용

패킷 필터링 규칙 세트를 사용하여 패킷 필터링을 설정합니다. `ipf` 명령을 사용하여 패킷 필터링 규칙 세트와 관련된 작업을 수행할 수 있습니다. `ipf` 명령에 대한 자세한 내용은 `ipf(1M)` 명령을 참조하십시오.

명령줄에서 `ipf` 명령을 사용하거나 패킷 필터링 구성 파일에서 패킷 필터링 규칙을 만들 수 있습니다. 부트 시 패킷 필터링 규칙이 로드되도록 하려면 패킷 필터링 규칙을 배치할 `/etc/ipf/ipf.conf`라는 구성 파일을 만듭니다. 부트 시 패킷 필터링 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipf.conf` 파일을 배치하고 `ipf` 명령을 사용하여 수동으로 패킷 필터링을 활성화합니다.

IP 필터를 사용하여 두 개의 패킷 필터링 규칙 세트(활성 규칙 세트 및 비활성 규칙 세트)를 유지 관리할 수 있습니다. 대부분의 경우 활성 규칙 세트와 관련된 작업을 수행합니다. 하지만 `ipf -I` 명령을 사용하여 비활성 규칙 목록에 명령 작업을 적용할 수 있습니다. 비활성 규칙 목록을 선택하지 않을 경우 해당 목록은 IP 필터에 사용되지 않습니다. 비활성 규칙 목록은 활성 패킷 필터링에 영향을 끼치지 않고 규칙을 저장할 수 있는 위치를 제공합니다.

IP 필터는 패킷을 전달하거나 차단하기 전에 구성된 규칙 목록의 처음부터 규칙 목록의 끝까지 규칙 목록에 있는 규칙을 처리합니다. IP 필터는 패킷 전달 여부를 결정하는 플래그를 유지 관리합니다. 전체 규칙 세트를 확인하고 마지막 일치 규칙을 기반으로 패킷을 전달할지 아니면 차단할지 결정합니다.

이 프로세스에는 두 가지 예외가 있습니다. 첫번째 예외는 패킷이 `quick` 키워드를 포함하는 규칙과 일치하는 경우입니다. 규칙에 `quick` 키워드가 포함되면 해당 규칙에 대한 작업이 수행되고 후속 규칙이 확인되지 않습니다. 두번째 예외는 패킷이 `group` 키워드를 포함하는 규칙과 일치하는 경우입니다. 패킷이 그룹과 일치되면 그룹 태그가 지정된 규칙만 확인됩니다.

패킷 필터링 규칙 구성

다음 구문을 사용하여 패킷 필터링 규칙을 만들 수 있습니다.

action `[in|out]` *option keyword, keyword...*

1. 각 규칙은 작업으로 시작합니다. IP 필터는 패킷이 규칙과 일치하는 경우 패킷에 작업을 적용합니다. 다음은 패킷에 적용되는 가장 일반적으로 사용되는 작업을 나열한 것입니다.

   `block` 패킷이 필터를 통과하지 못하도록 합니다.

   `pass` 패킷이 필터를 통과할 수 있도록 합니다.

   `log` 패킷을 기록하되 패킷 차단 또는 통과를 결정하지 않습니다. `ipmon` 명령을 사용하여 로그를 확인할 수 있습니다.

   `count` 필터 통계에 패킷을 포함합니다. `ipfstat` 명령을 사용하여 통계를 확인할 수 있습니다.

   `skip` *number* 필터가 *number*개의 필터링 규칙을 건너 뛸 수 있도록 합니다.

   `auth` 패킷 정보를 검증하는 사용자 프로그램이 패킷 인증을 수행하도록 요청합니다. 프로그램에서 패킷 전달 또는 차단을 결정합니다.

2. 작업 뒤에 오는 단어는 `in` 또는 `out`이어야 합니다. 선택한 단어에 따라 패킷 필터링 규칙이 수신 패킷에 적용될지 아니면 송신 패킷에 적용될지 결정됩니다.
3. 그런 다음 옵션 목록에서 옵션을 선택할 수 있습니다. 옵션을 두 개 이상 사용할 경우 여기에 표시되는 순서를 따라야 합니다.

   `log` 규칙이 마지막 일치 규칙인 경우 패킷을 기록합니다. `ipmon` 명령을 사용하여 로그를 확인할 수 있습니다.

   `quick` 패킷 일치가 있을 경우 `quick` 옵션이 포함된 규칙을 실행합니다. 모든 후속 규칙 확인이 중지됩니다.

   `on` *interface-name* 패킷이 지정된 인터페이스 내부 또는 외부로 이동되고 있는 경우에만 규칙을 적용합니다.

   `dup-to` *interface-name* 패킷을 복사하고 *interface-name*의 중복 출력을 선택적으로 지정된 IP 주소로 보냅니다.

   `to` *interface-name* 패킷을 *interface-name*의 아웃바운드 대기열로 이동합니다.
4. 옵션을 지정한 후 패킷이 규칙과 일치하는지 여부를 확인하는 다양한 키워드를 선택할 수 있습니다. 다음 키워드는 여기에 표시된 순서대로 사용해야 합니다.

주 – 기본적으로 구성 파일의 규칙과 일치하지 않는 패킷은 필터를 통해 전달됩니다.

   `tos` 16진수 또는 십진수 정수로 표시되는 type-of-service 값을 기준으로 패킷을 필터링합니다.

   `ttl` time-to-live 값을 기준으로 패킷을 일치시킵니다. 패킷에 저장된 time-to-live 값은 패킷을 폐기하기 전에 네트워크에 보관할 수 있는 기간을 나타냅니다.

`proto`	특정 프로토콜을 일치시킵니다. `/etc/protocols` 파일에 지정된 프로토콜 이름을 사용할 수도 있고, 십진수를 사용하여 프로토콜을 나타낼 수도 있습니다. `tcp/udp` 키워드를 사용하여 TCP 또는 UDP 패킷을 일치시킬 수 있습니다.
`from/to/all/ any`	소스 IP 주소, 대상 IP 주소, 포트 번호 중 일부 또는 전체와 일치시킵니다. `all` 키워드는 모든 소스에서 수신되고 모든 대상으로 송신되는 패킷을 승인할 수 있습니다.
`with`	패킷과 연관되어 있는 지정된 속성을 일치시킵니다. 옵션이 없는 경우에만 패킷을 일치시키려면 키워드 앞에 `not` 또는 `no` 단어를 삽입하십시오.
`flags`	설정된 TCP 플래그를 기준으로 필터링할 TCP에 사용됩니다. TCP 플래그에 대한 자세한 내용은 `ipf(4)` 매뉴얼 페이지를 참조하십시오.
`icmp-type`	ICMP 유형에 따라 필터링합니다. 이 키워드는 `proto` 옵션이 `icmp`로 설정된 경우에만 사용되며 `flags` 옵션이 설정된 경우 사용되지 않습니다.
`keep` *keep-options*	패킷에 대해 보관되는 정보를 결정합니다. 사용 가능한 *keep-options*로는 `state` 옵션과 `frags` 옵션이 있습니다. `state` 옵션은 세션에 대한 정보를 보관하며 TCP, UDP 및 ICMP 패킷에 보관될 수 있습니다. `frags` 옵션은 패킷 단편에 정보를 보관하며 후속 단편에 정보를 적용합니다. *keep-options*를 사용하면 액세스 제어 목록을 확인하지 않고서도 일치 패킷을 전달할 수 있습니다.
`head` *number*	*number* 번호로 표시되는 필터링 규칙에 대한 새 그룹을 만듭니다.
`group` *number*	기본 그룹 대신 그룹 번호 *number*에 규칙을 추가합니다. 지정된 다른 그룹이 없을 경우 모든 필터링 규칙이 그룹 0에 배치됩니다.

다음 예에서는 규칙을 만드는 패킷 필터링 규칙 구문을 배치하는 방법을 보여 줍니다. IP 주소 `192.168.0.0/16`의 수신 트래픽을 차단하려면 규칙 목록에 다음 규칙을 포함시킵니다.

```
block in quick from 192.168.0.0/16 to any
```

패킷 필터링 규칙을 작성하는 데 사용되는 전체 문법 및 구문은 `ipf(4)` 매뉴얼 페이지를 참조하십시오. 패킷 필터링과 관련된 작업은 626 페이지 “IP 필터에 대한 패킷 필터링 규칙 세트 관리”를 참조하십시오. 예에 표시된 IP 주소 체계(`192.168.0.0/16`)에 대한 설명은 2 장, “TCP/IP 네트워크 계획(작업)”을 참조하십시오.

IP 필터의 NAT 기능 사용

NAT는 소스 및 대상 IP 주소를 다른 인터넷 또는 인트라넷 주소로 변환하는 매핑 규칙을 설정합니다. 이러한 규칙은 수신 또는 송신 IP 패킷의 소스 및 대상 주소를 수정하고 패킷을 보냅니다. NAT를 사용하여 포트 간에 트래픽을 재지정할 수도 있습니다. NAT는 패킷이 수정되거나 재지정되는 동안 패킷의 무결성을 유지합니다.

`ipnat` 명령을 사용하여 NAT 규칙 목록과 관련된 작업을 수행할 수 있습니다. `ipnat` 명령에 대한 자세한 내용은 `ipnat(1M)` 명령을 참조하십시오.

명령줄에서 `ipnat` 명령을 사용하거나 NAT 구성 파일에서 NAT 규칙을 만들 수 있습니다. NAT 구성 규칙은 `ipnat.conf` 파일에 상주합니다. 부트 시 NAT 규칙이 로드되도록 하려면 NAT 규칙을 배치할 `/etc/ipf/ipnat.conf`라는 파일을 만듭니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipnat.conf` 파일을 배치하고 `ipnat` 명령을 사용하여 수동으로 패킷 필터링을 활성화합니다.

NAT 규칙은 IPv4 및 IPv6 주소 모두에 적용할 수 있습니다. 하지만 두 유형의 주소를 단일 규칙으로 지정할 수 없습니다. 대신 각 주소 유형에 대해 별도의 규칙을 설정해야 합니다. IPv6 주소가 포함된 NAT 규칙에서는 `mapproxy` 및 `rdrproxy` NAT 명령을 동시에 사용할 수 없습니다.

NAT 규칙 구성

다음 구문을 사용하여 NAT 규칙을 만들 수 있습니다.

command interface-name parameters

1. 각 규칙은 다음 명령 중 하나로 시작합니다.

   `map` 제한되지 않은 라운드 로빈 프로세스에서 특정 IP 주소 또는 네트워크를 다른 IP 주소 또는 네트워크에 매핑합니다.

   `rdr` 특정 IP 주소와 포트 쌍의 패킷을 다른 IP 주소와 포트 쌍으로 재지정합니다.

   `bimap` 외부 IP 주소와 내부 IP 주소 간에 양방향 NAT를 설정합니다.

   `map-block` 정적 IP 주소 기반 변환을 설정합니다. 이 명령은 주소를 강제로 대상 범위로 변환하는 알고리즘을 기반으로 합니다.

2. 명령 뒤에 오는 단어는 인터페이스 이름(예: `hme0`)입니다.
3. 그런 다음 NAT 구성을 결정하는 다양한 매개변수를 선택할 수 있습니다. 몇 가지 매개변수는 다음과 같습니다.

   `ipmask` 네트워크 마스크를 지정합니다.

   `dstipmask` `ipmask`가 변환되는 주소를 지정합니다.

   `mapport` 포트 번호 범위와 함께 `tcp`, `udp` 또는 `tcp/udp` 프로토콜을 지정합니다.

다음 예에서는 NAT 규칙을 만드는 NAT 규칙 구문을 배치하는 방법을 보여 줍니다. 소스 주소가 192.168.1.0/24인 de0 장치에서 송신되는 패킷을 재작성하고 외부적으로 소스 주소를 10.1.0.0/16으로 표시하려면 NAT 규칙 세트에 다음 규칙을 포함시킵니다.

```
map de0 192.168.1.0/24 -> 10.1.0.0/16
```

IPv6 주소에는 다음 규칙이 적용됩니다.

```
map ppp0 fec0:1::/64 -> 2000:1:2::/72 portmap tcp/udp 1025:65000
map-block ppp0 fe80:0:0:209::/64 -> 209:1:2::/72 ports auto
rdr ce0 209::ffff:fe13:e43e port 80 -> fec0:1::e,fec0:1::f port 80 tcp round-robin
```

NAT 규칙을 작성하는 데 사용되는 전체 문법 및 구문은 ipnat(4) 매뉴얼 페이지를 참조하십시오.

IP 필터의 주소 풀 기능 사용

주소 풀은 주소/넷마스크 쌍 그룹의 이름을 지정하는 데 사용되는 단일 참조를 설정합니다. 주소 풀은 IP 주소를 규칙과 일치시키는 데 필요한 시간을 단축시킬 프로세스를 제공합니다. 또한 주소 풀을 사용하면 큰 주소 그룹을 간편하게 관리할 수 있습니다.

주소 풀 구성 규칙은 ippool.conf 파일에 상주합니다. 부트 시 주소 풀 규칙이 로드되도록 하려면 주소 풀 규칙을 배치할 /etc/ipf/ippool.conf라는 파일을 만듭니다. 부트 시 주소 풀 규칙이 로드되지 않도록 하려면 선택한 위치에 ippool.conf 파일을 배치하고 ippool 명령을 사용하여 수동으로 패킷 필터링을 활성화합니다.

주소 풀 구성

다음 구문을 사용하여 주소 풀을 만들 수 있습니다.

table role = *role-name* type = *storage-format* number = *reference-number*

table 여러 주소에 대한 참조를 정의합니다.

role IP 필터의 풀 역할을 지정합니다. 지금은 ipf 역할만 참조할 수 있습니다.

type 풀에 대한 저장 형식을 지정합니다.

number 필터링 규칙에 사용되는 참조 번호를 지정합니다.

예를 들어, 10.1.1.1 및 10.1.1.2 주소 그룹과 192.16.1.0 네트워크를 풀 번호 13으로 참조하려면 주소 풀 구성 파일에 다음 규칙을 포함시킵니다.

```
table role = ipf type = tree number = 13
{ 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24 };
```

그런 다음 필터링 규칙의 풀 번호 13을 참조하려면 다음 예와 유사한 규칙을 생성합니다.

```
pass in from pool/13 to any
```

풀에 대한 참조를 포함하는 규칙 파일을 로드하기 전에 풀 파일을 로드해야 합니다. 그렇지 않을 경우 다음 출력과 같이 풀이 정의되지 않습니다.

```
# ipfstat -io
empty list for ipfilter(out)
block in from pool/13(!) to any
```

나중에 풀을 추가하는 경우에도 풀 추가로 인해 커널 규칙 세트가 업데이트되지 않습니다. 또한 풀을 참조하는 규칙 파일을 다시 로드해야 합니다.

패킷 필터링 규칙을 작성하는 데 사용되는 전체 문법 및 구문은 `ippool(4)` 매뉴얼 페이지를 참조하십시오.

패킷 필터 후크

Solaris 10 7/07 릴리스부터 패킷 필터 후크가 `pfil` 모듈을 대체하여 IP 필터를 사용으로 설정합니다. 이전 Solaris 릴리스에서는 추가 IP 필터 설정 단계로 `pfil` 모듈을 구성해야 했습니다. 이 추가 구성 요구 사항으로 인해 IP 필터가 제대로 작동하지 않을 수 있는 오류 발생 위험이 많았습니다. 또한 IP와 장치 드라이버 사이에 `pfil` STREAMS 모듈을 삽입하는 것도 성능 저하의 원인이었습니다. 마지막으로 `pfil` 모듈은 영역 간에 패킷 가로채기를 수행할 수 없습니다.

패킷 필터 후크를 사용하면 IP 필터를 사용으로 설정하는 절차가 간소화됩니다. 이러한 후크를 통해 IP 필터는 사전 경로 지정(입력) 및 사후 경로 지정(출력) 필터 탭을 사용하여 Oracle Solaris 시스템에서의 패킷 플로우 입력 및 출력을 제어합니다.

패킷 필터 후크를 사용하면 `pfil` 모듈이 필요하지 않습니다. 따라서 모듈과 연관된 다음 구성 요소도 제거되었습니다.

- `pfil` 드라이버
- `pfil` 데몬
- `svc:/network/pfil` SMF 서비스

IP 필터를 사용으로 설정하는 것과 관련된 작업은 26 장, “IP 필터(작업)”를 참조하십시오.

IP 필터 및 pfil STREAMS 모듈

주 –

pfil 모듈은 다음 Solaris 릴리스에서만 IP 필터와 함께 사용됩니다.

- Solaris 10 3/05 릴리스
- Solaris 10 1/06 릴리스
- Solaris 10 6/06 릴리스
- Solaris 10 11/06 릴리스

Solaris 10 7/07 릴리스부터 pfil 모듈이 패킷 필터 후크로 대체되었으므로 더 이상 IP 필터와 함께 사용되지 않습니다.

IP 필터를 사용으로 설정하려면 pfil STREAMS 모듈이 필요합니다. 그러나 IP 필터는 모듈을 모든 인터페이스로 푸시하는 자동 방식을 제공하지 않습니다. 대신 pfil STREAMS 모듈은 SMF 서비스 svc:/network/pfil에 의해 관리됩니다. 네트워크 인터페이스에서 필터링을 활성화하려면 먼저 pfil.ap 파일을 구성합니다. 그런 다음 svc:/network/pfil 서비스를 활성화하여 네트워크 인터페이스에 pfil STREAMS 모듈을 제공합니다. STREAMS 모듈을 적용하려면 시스템을 재부트하거나 필터링할 각 네트워크 인터페이스의 연결을 해제한 다음 다시 연결해야 합니다. IPv6 패킷 필터링 기능을 활성화하려면 인터페이스의 inet6 버전을 연결해야 합니다.

네트워크 인터페이스에 대한 pfil 모듈을 찾을 수 없으면 SMF 서비스가 유지 관리 상태가 됩니다. 이러한 상황이 발생하는 가장 일반적인 원인은 /etc/ipf/pfil.ap 파일을 잘못 편집했기 때문입니다. 서비스가 유지 관리 모드가 되면 현상이 필터링 로그 파일에 기록됩니다.

IP 필터를 활성화하는 것과 관련된 작업은 613 페이지 “IP 필터 구성”을 참조하십시오.

IP 필터용 IPv6

Solaris 6/06 릴리스부터 IP 필터와 함께 IPv6에 대한 지원이 제공됩니다. IPv6 패킷 필터링은 소스/대상 IPv6 주소, IPv6 주소를 포함하는 풀 및 IPv6 확장 헤더를 기준으로 필터링을 수행할 수 있습니다.

IPv6은 여러 측면에서 IPv4와 유사합니다. 단, IP의 두 버전 간에 헤더 및 패킷 크기가 다르므로 IP 필터를 사용할 때 반드시 고려해야 합니다. IPv6 패킷(**점보그램**이라고도 함)에는 65,535바이트 이상의 데이터그램이 포함되어 있습니다. IP 필터는 IPv6 점보그램을 지원하지 않습니다. 기타 IPv6 기능에 대해 자세히 알아보려면 66 페이지 “IPv6의 주요 기능”을 참조하십시오.

주 – 점보그램에 대한 자세한 내용은 IETF(Internet Engineering Task Force)[http://www.ietf.org/rfc/rfc2675.txt]의 IPv6 Jumbograms, RFC 2675 문서를 참조하십시오.

IPv6과 관련된 IP 필터 작업은 IPv4와 유사합니다. 가장 큰 차이는 특정 명령에 -6 옵션을 사용한다는 점입니다. ipf 명령과 ipfstat 명령에는 IPv6 패킷 필터링에 사용할 -6 옵션이 포함됩니다. ipf 명령에 -6 옵션을 사용하여 IPv6 패킷 필터링 규칙을 로드하고 비울 수 있습니다. IPv6 통계를 표시하려면 ipfstat 명령에 -6 옵션을 사용하십시오. ipmon 및 ippool 명령도 IPv6을 지원하지만 IPv6 지원과 관련된 옵션이 없습니다. ipmon 명령이 IPv6 패킷 로깅을 수행하도록 개선되었습니다. ippool 명령은 IPv6 주소와 함께 풀을 지원합니다. IPv4 주소와 IPv6 주소 중 하나에 대해서만 풀을 만들 수도 있고, 동일한 풀 내에 IPv4 주소와 IPv6 주소를 모두 포함하는 풀을 만들 수도 있습니다.

ipf6.conf 파일을 사용하여 IPv6에 대한 패킷 필터링 규칙 세트를 만들 수 있습니다. 기본적으로 ipf6.conf 구성 파일은 /etc/ipf 디렉토리에 포함됩니다. 다른 필터링 구성 파일에서와 마찬가지로 ipf6.conf 파일은 부트 프로세스 동안 자동으로 로드됩니다(이 파일이 /etc/ipf 디렉토리에 저장되어 있는 경우). 다른 위치에 IPv6 구성 파일을 만들어 저장하고 수동으로 파일을 로드할 수도 있습니다.

IPv6에 대한 패킷 필터링 규칙이 설정되면 인터페이스의 inet6 버전을 연결하여 IPv6 패킷 필터링 기능을 활성화하십시오.

IPv6에 대한 자세한 내용은 3 장, “IPv6 소개(개요)”를 참조하십시오. IP 필터와 관련된 작업은 26 장, “IP 필터(작업)”를 참조하십시오.

IP 필터 매뉴얼 페이지

다음 표에서는 IP 필터와 관련된 매뉴얼 페이지에 대해 설명합니다.

매뉴얼 페이지	설명
ipf(1M)	다음 작업을 완료하려면 ipf 명령을 사용합니다. ■ 패킷 필터링 규칙 세트와 관련된 작업을 수행합니다. ■ 필터링을 사용 안함/사용으로 설정합니다. ■ 통계를 재설정하고 커널 내 인터페이스 목록을 현재 인터페이스 상태 목록과 다시 동기화합니다.
ipf(4)	IP 필터 패킷 필터링 규칙 생성 문법 및 구문을 포함합니다.
ipfilter(5)	오픈 소스 IP 필터 라이센스 정보를 제공합니다.
ipfs(1M)	재부트 시 NAT 정보 및 상태 테이블 정보를 저장하고 복원하려면 ipfs 명령을 사용합니다.
ipfstat(1M)	패킷 처리에 대한 통계를 검색하고 표시하려면 ipfstat 명령을 사용합니다.
ipmon(1M)	로그 장치를 열고 패킷 필터링과 NAT에 대해 기록된 패킷을 보려면 ipmon 명령을 사용합니다.

매뉴얼 페이지	설명
ipnat(1M)	다음 작업을 완료하려면 ipnat 명령을 사용합니다. ■ NAT 규칙과 관련된 작업을 수행합니다. ■ NAT 통계를 검색하고 표시합니다.
ipnat(4)	NAT 규칙 생성 문법 및 구문을 포함합니다.
ippool(1M)	주소 풀을 만들고 관리하려면 ippool 명령을 사용합니다.
ippool(4)	IP 필터 주소 풀 생성 문법 및 구문을 포함합니다.
ndd(1M)	pfil STREAMS 모듈의 현재 필터링 매개변수 및 조정 가능한 매개변수의 현재 값을 표시합니다.

◆ ◆ ◆ 26 장

26 IP 필터(작업)

이 장에서는 단계별 작업 지침을 제공합니다. IP 필터에 대한 개요 정보는 25 장, “Oracle Solaris의 IP 필터(개요)”를 참조하십시오.

이 장은 다음 정보를 포함합니다.

IP 필터 구성

다음 작업 맵에서는 IP 필터 구성과 관련된 절차를 식별합니다.

표 26–1 IP 필터 구성(작업 맵)

작업	설명	수행 방법
초기에 IP 필터를 사용으로 설정합니다.	기본적으로 IP 필터는 사용으로 설정되어 있지 않습니다. 수동으로 사용으로 설정하거나 /etc/ipf/ 디렉토리의 구성 파일을 사용한 후 시스템을 재부트해야 합니다. Solaris 10 7/07 릴리스부터 패킷 필터 후크가 pfil 모듈을 대체하여 IP 필터를 사용으로 설정합니다.	614 페이지 “IP 필터를 사용으로 설정하는 방법”

표 26–1 IP 필터 구성(작업 맵) *(계속)*

작업	설명	수행 방법
IP 필터를 다시 사용으로 설정합니다.	IP 필터가 비활성화되거나 사용 안함으로 설정된 경우 시스템을 재부트하거나 `ipf` 명령을 사용하여 IP 필터를 다시 사용으로 설정할 수 있습니다.	615 페이지 "IP 필터를 다시 사용으로 설정하는 방법"
루프백 필터링을 사용으로 설정합니다.	선택적으로 영역 간의 트래픽을 필터링하는 등의 용도로 루프백 필터링을 사용으로 설정할 수 있습니다.	616 페이지 "루프백 필터링을 사용으로 설정하는 방법"

▼ IP 필터를 사용으로 설정하는 방법

이 절차를 사용하여 Solaris 10 7/07 OS 이상을 실행 중인 시스템에서 IP 필터를 사용으로 설정합니다. Solaris 10 7/07보다 이전 OS인 Solaris 10을 실행 중인 시스템에서 IP 필터를 사용으로 설정하려면 619 페이지 "`pfil` 모듈 작업"을 참조하십시오.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 패킷 필터링 규칙 세트를 만듭니다.

패킷 필터링 규칙 세트에는 IP 필터에 사용되는 패킷 필터링 규칙이 포함되어 있습니다. 부트 시 패킷 필터링 규칙이 로드되도록 하려면 IPv4 패킷 필터링이 구현되도록 `/etc/ipf/ipf.conf` 파일을 편집합니다. IPv6 패킷 필터링 규칙에 `/etc/ipf/ipf6.conf` 파일을 사용합니다. 부트 시 패킷 필터링 규칙이 로드되지 않도록 하려면 선택한 파일에 규칙을 배치하고 수동으로 패킷 필터링을 활성화합니다. 패킷 필터링에 대한 자세한 내용은 603 페이지 "IP 필터의 패킷 필터링 기능 사용"을 참조하십시오. 구성 파일 사용에 대한 자세한 내용은 644 페이지 "IP 필터 구성 파일 만들기 및 편집"를 참조하십시오.

3 (선택 사항) NAT(Network Address Translation) 구성 파일을 만듭니다.

주 – NAT(Network Address Translation)는 IPv6을 지원하지 않습니다.

NAT를 사용하려면 `ipnat.conf` 파일을 만듭니다. 부트 시 NAT 규칙이 로드되도록 하려면 NAT 규칙을 배치할 `/etc/ipf/ipnat.conf`라는 파일을 만듭니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipnat.conf` 파일을 배치하고 수동으로 NAT 규칙을 활성화합니다.

NAT에 대한 자세한 내용은 606 페이지 "IP 필터의 NAT 기능 사용"을 참조하십시오.

4 **(선택 사항) 주소 풀 구성 파일을 만듭니다.**

단일 주소 풀로 사용할 주소 그룹을 나타내려면 `ipool.conf` 파일을 만듭니다. 부트 시 주소 풀 구성 파일이 로드되도록 하려면 주소 풀을 배치할 `/etc/ipf/ippool.conf`라는 파일을 만듭니다. 부트 시 주소 풀 구성 파일이 로드되지 않도록 하려면 선택한 위치에 `ippool.conf` 파일을 배치하고 수동으로 규칙을 활성화합니다.

주소 풀에는 IPv4 주소와 IPv6 주소 중 하나만 포함될 수도 있고, IPv4 주소와 IPv6 주소가 모두 포함될 수도 있습니다.

주소 풀에 대한 자세한 내용은 607 페이지 “IP 필터의 주소 풀 기능 사용”을 참조하십시오.

5 **(선택 사항) 루프백 트래픽의 필터링을 사용으로 설정합니다.**

시스템에서 구성된 영역 간의 트래픽을 필터링하려면 루프백 필터링을 사용으로 설정해야 합니다. 616 페이지 “루프백 필터링을 사용으로 설정하는 방법”을 참조하십시오. 영역에 적용할 적합한 규칙 세트도 정의해야 합니다.

6 **IP 필터를 활성화합니다.**

```
# svcadm enable network/ipfilter
```

▼ IP 필터를 다시 사용으로 설정하는 방법

패킷 필터링을 일시적으로 사용 안함으로 설정한 후 다시 사용으로 설정할 수 있습니다.

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”을 참조하십시오.

2 **다음 방법 중 하나로 IP 필터를 사용으로 설정하고 필터링을 활성화합니다.**

- 시스템을 재부트합니다.

  ```
  # reboot
  ```

  주 – IP 필터가 사용으로 설정되면 재부트 후 `/etc/ipf/ipf.conf` 파일, `/etc/ipf/ipf6.conf` 파일(IPv6을 사용하는 경우) 또는 `/etc/ipf/ipnat.conf` 파일이 있을 경우 로드됩니다.

- 다음과 같은 일련의 명령을 실행하여 IP 필터를 사용으로 설정하고 필터링을 활성화합니다.

  a. IP 필터를 사용으로 설정합니다.

  ```
  # ipf -E
  ```

b. 패킷 필터링을 활성화합니다.

```
# ipf -f filename
```

c. (선택 사항) NAT를 활성화합니다.

```
# ipnat -f filename
```

주 – NAT(Network Address Translation)는 IPv6을 지원하지 않습니다.

▼ 루프백 필터링을 사용으로 설정하는 방법

주 – 시스템에서 Solaris 10 7/07 이상의 릴리스를 실행 중인 경우에만 루프백 트래픽을 필터링할 수 있습니다. 이전 Oracle Solaris 10 릴리스에서는 루프백 필터링이 지원되지 않습니다.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 IP 필터가 실행 중인 경우 중지합니다.

```
# svcadm disable network/ipfilter
```

3 파일 시작 부분에 다음 행을 추가하여 `/etc/ipf.conf` 또는 `/etc/ipf6.conf` 파일을 편집합니다.

```
set intercept_loopback true;
```

파일에서 정의된 모든 IP 필터 규칙 앞에 이 행이 와야 합니다. 단, 다음 예와 유사하게 행 앞에 주석을 삽입할 수 있습니다.

```
#
# Enable loopback filtering to filter between zones
#
set intercept_loopback true;
#
# Define policy
#
block in all
block out all
<other rules>
...
```

4 IP 필터를 시작합니다.

```
# svcadm enable network/ipfilter
```

5 **루프백 필터링 상태를 확인하려면 다음 명령을 사용합니다.**

```
# ipf -T ipf_loopback
ipf_loopback    min 0   max 0x1 current 1
#
```

루프백 필터링이 사용 안함으로 설정된 경우 명령으로 다음 출력이 생성됩니다.

```
ipf_loopback    min 0   max 0x1 current 0
```

IP 필터 비활성화 및 사용 안함으로 설정

다음과 같은 경우 패킷 필터링 및 NAT를 비활성화하거나 사용 안함으로 설정할 수 있습니다.

- 테스트 용도로 사용하려는 경우
- 문제의 원인이 IP 필터인 것으로 간주되어 시스템 문제를 해결하려는 경우

다음 작업 맵에서는 IP 필터 기능을 비활성화하거나 사용 안함으로 설정하는 것과 관련된 절차를 식별합니다.

표 26–2 IP 필터 비활성화 및 사용 안함으로 설정(작업 맵)

작업	설명	수행 방법
패킷 필터링을 비활성화합니다.	`ipf` 명령을 사용하여 패킷 필터링을 비활성화합니다.	617 페이지 “패킷 필터링 비활성화 방법”
NAT를 비활성화합니다.	`ipnat` 명령을 사용하여 NAT를 비활성화합니다.	618 페이지 “NAT 비활성화 방법”
패킷 필터링 및 NAT를 사용 안함으로 설정합니다.	`ipf` 명령을 사용하여 패킷 필터링 및 NAT를 사용 안함으로 설정합니다.	618 페이지 “패킷 필터링을 사용 안함으로 설정하는 방법”

▼ 패킷 필터링 비활성화 방법

다음 절차에서는 활성 필터링 규칙 세트에서 패킷 필터링 규칙을 비워 IP 필터 패킷 필터링을 비활성화합니다. 이 절차에서는 IP 필터를 사용 안함으로 설정하지 않습니다. 규칙 세트에 규칙을 추가하여 IP 필터를 재활성화할 수 있습니다.

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”을 참조하십시오.

2 **다음 방법 중 하나로 IP 필터 규칙을 비활성화합니다.**

- 커널에서 활성 규칙 세트를 제거합니다.

  ```
  # ipf -Fa
  ```

  이 명령은 모든 패킷 필터링 규칙을 비활성화합니다.
- 수신 패킷 필터링 규칙을 제거합니다.

  ```
  # ipf -Fi
  ```

  이 명령은 수신 패킷에 대한 패킷 필터링 규칙을 비활성화합니다.
- 송신 패킷 필터링 규칙을 제거합니다.

  ```
  # ipf -Fo
  ```

  이 명령은 송신 패킷에 대한 패킷 필터링 규칙을 비활성화합니다.

▼ NAT 비활성화 방법

다음 절차에서는 활성 NAT 규칙 세트에서 NAT 규칙을 비워 IP 필터 NAT 규칙을 비활성화합니다. 이 절차에서는 IP 필터를 사용 안함으로 설정하지 않습니다. 규칙 세트에 규칙을 추가하여 IP 필터를 재활성화할 수 있습니다.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 커널에서 NAT를 제거합니다.

```
# ipnat -FC
```

`-C` 옵션은 현재 NAT 규칙 목록의 모든 항목을 제거합니다. `-F` 옵션은 현재 활성 NAT 매핑을 보여 주는 현재 NAT 변환 테이블의 모든 활성 항목을 제거합니다.

▼ 패킷 필터링을 사용 안함으로 설정하는 방법

이 절차를 실행하면 커널에서 패킷 필터링과 NAT가 모두 제거됩니다. 이 절차를 사용할 경우 패킷 필터링 및 NAT를 재활성화하려면 IP 필터를 다시 사용으로 설정해야 합니다. 자세한 내용은 615 페이지 "IP 필터를 다시 사용으로 설정하는 방법"을 참조하십시오.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **패킷 필터링을 사용 안함으로 설정하고 모든 패킷을 네트워크로 전달할 수 있도록 허용합니다.**

```
# ipf –D
```

주 – `ipf -D` 명령은 규칙 세트에서 규칙을 비웁니다. 필터링을 다시 사용으로 설정하는 경우 규칙 세트에 규칙을 추가해야 합니다.

pfil 모듈 작업

이 절에서는 `pfil` STREAMS 모듈을 사용하여 IP 필터를 활성화/비활성화하는 방법 및 `pfil` 통계를 보는 방법에 대해 설명합니다. 이 절차는 다음 Solaris 릴리스 중 하나를 실행하는 시스템에만 적용됩니다.

- Solaris 10 3/05 릴리스
- Solaris 10 1/06 릴리스
- Solaris 10 6/06 릴리스
- Solaris 10 11/06 릴리스

다음 작업 맵에서는 `pfil` 모듈 구성과 관련된 절차를 식별합니다.

표 26–3 pfil 모듈 작업(작업 맵)

작업	설명	수행 방법
IP 필터를 사용으로 설정	기본적으로 IP 필터는 사용으로 설정되어 있지 않습니다. 수동으로 사용으로 설정하거나 `/etc/ipf/` 디렉토리의 구성 파일을 사용한 후 시스템을 재부트해야 합니다.	620 페이지 "이전 Solaris 릴리스에서 IP 필터를 사용으로 설정하는 방법"
패킷 필터링을 위해 NIC 활성화	NIC에서 패킷 필터링을 활성화하도록 `pfil` 모듈을 구성합니다.	622 페이지 "패킷 필터링을 위해 NIC를 활성화하는 방법"
NIC에서 IP 필터 비활성화	NIC를 제거하고 모든 패킷이 NIC를 통해 전달되도록 허용합니다.	623 페이지 "NIC에서 IP 필터를 비활성화하는 방법"
`pfil` 통계 보기	`pfil` 모듈의 통계를 보고 `ndd` 명령을 사용하여 IP 필터의 문제를 해결합니다.	625 페이지 "IP 필터에 대한 pfil 통계를 보는 방법"

▼ 이전 Solaris 릴리스에서 IP 필터를 사용으로 설정하는 방법

IP 필터는 Oracle Solaris와 함께 설치됩니다. 그러나 기본적으로 패킷 필터링은 사용으로 설정되어 있지 않습니다. 다음 절차를 사용하여 IP 필터를 활성화합니다.

주 – 시스템에서 Solaris 10 7/07 이상의 릴리스를 실행하는 경우 패킷 필터 후크를 사용하는 614 페이지 "IP 필터를 사용으로 설정하는 방법" 절차를 따릅니다.

---

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 선택한 파일 편집기를 시작하여 `/etc/ipf/pfil.ap` 파일을 편집합니다.

이 파일에는 호스트의 네트워크 인터페이스 카드(NIC) 이름이 있습니다. 기본적으로 이름은 주석 처리되어 있습니다. 필터링할 네트워크 트래픽을 전달하는 장치 이름을 주석 해제합니다. 시스템의 NIC 이름이 없으면 NIC를 지정하는 라인을 추가합니다.

```
# vi /etc/ipf/pfil.ap
# IP Filter pfil autopush setup
#
# See autopush(1M) manpage for more information.
#
# Format of the entries in this file is:
#
#major  minor lastminor modules

#le     -1      0       pfil
#qe     -1      0       pfil
hme     -1      0       pfil (Device has been uncommented for filtering)
#qfe    -1      0       pfil
#eri    -1      0       pfil
#ce     -1      0       pfil
#bge    -1      0       pfil
#be     -1      0       pfil
#vge    -1      0       pfil
#ge     -1      0       pfil
#nf     -1      0       pfil
#fa     -1      0       pfil
#ci     -1      0       pfil
#el     -1      0       pfil
#ipdptp -1      0       pfil
#lane   -1      0       pfil
#dmfe   -1      0       pfil
```

3 `network/pfil` 서비스 인스턴스를 다시 시작하여 `/etc/ipf/pfil.ap` 파일에 대한 변경 사항을 활성화합니다.

```
# svcadm restart network/pfil
```

4 패킷 필터링 규칙 세트를 만듭니다.

패킷 필터링 규칙 세트에는 IP 필터에 사용되는 패킷 필터링 규칙이 포함되어 있습니다. 부트 시 패킷 필터링 규칙이 로드되도록 하려면 IPv4 패킷 필터링이 구현되도록 `/etc/ipf/ipf.conf` 파일을 편집합니다. IPv6 패킷 필터링 규칙에 `/etc/ipf/ipf6.conf` 파일을 사용합니다. 부트 시 패킷 필터링 규칙이 로드되지 않도록 하려면 선택한 파일에 규칙을 배치하고 수동으로 패킷 필터링을 활성화합니다. 패킷 필터링에 대한 자세한 내용은 603 페이지 “IP 필터의 패킷 필터링 기능 사용”을 참조하십시오. 구성 파일 사용에 대한 자세한 내용은 644 페이지 “IP 필터 구성 파일 만들기 및 편집”를 참조하십시오.

5 (선택 사항) NAT(Network Address Translation) 구성 파일을 만듭니다.

주 – NAT(Network Address Translation)는 IPv6을 지원하지 않습니다.

NAT를 사용하려면 `ipnat.conf` 파일을 만듭니다. 부트 시 NAT 규칙이 로드되도록 하려면 NAT 규칙을 배치할 `/etc/ipf/ipnat.conf`라는 파일을 만듭니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipnat.conf` 파일을 배치하고 수동으로 NAT 규칙을 활성화합니다.

NAT에 대한 자세한 내용은 606 페이지 “IP 필터의 NAT 기능 사용”을 참조하십시오.

6 (선택 사항) 주소 풀 구성 파일을 만듭니다.

단일 주소 풀로 사용할 주소 그룹을 나타내려면 `ipool.conf` 파일을 만듭니다. 부트 시 주소 풀 구성 파일이 로드되도록 하려면 주소 풀을 배치할 `/etc/ipf/ippool.conf`라는 파일을 만듭니다. 부트 시 주소 풀 구성 파일이 로드되지 않도록 하려면 선택한 위치에 `ippool.conf` 파일을 배치하고 수동으로 규칙을 활성화합니다.

주소 풀에는 IPv4 주소와 IPv6 주소 중 하나만 포함될 수도 있고, IPv4 주소와 IPv6 주소가 모두 포함될 수도 있습니다.

주소 풀에 대한 자세한 내용은 607 페이지 “IP 필터의 주소 풀 기능 사용”을 참조하십시오.

7 다음 방법 중 하나를 사용하여 IP 필터를 활성화합니다.

- IP 필터를 사용으로 설정하고 시스템을 재부트합니다.

```
# svcadm enable network/ipfilter
# reboot
```

주 – NIC에서 `ifconfig unplumb` 및 `ifconfig plumb` 명령을 안전하게 사용할 수 없는 경우 재부트가 필요합니다.

- `ifconfig unplumb` 및 `ifconfig plumb` 명령을 사용하여 NIC를 사용으로 설정합니다. 그런 다음 IP 필터를 사용으로 설정합니다. IPv6 패킷 필터링을 구현하려면 인터페이스의 `inet6` 버전을 연결해야 합니다.

```
# ifconfig hme0 unplumb
# ifconfig hme0 plumb 192.168.1.20 netmask 255.255.255.0 up
# ifconfig hme0 inte6 unplumb
# ifconfig hme0 inet6 plumb fec3:f849::1/96 up
# svcadm enable network/ipfilter
```

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

▼ 패킷 필터링을 위해 NIC를 활성화하는 방법

`/etc/ipf/ipf.conf` 파일(또는 IPv6을 사용하는 경우 `/etc/ipf/ipf6.conf` 파일)이 있는 경우에는 재부트 시 IP 필터가 사용으로 설정됩니다. IP 필터를 사용으로 설정한 후 NIC에서 필터링을 사용으로 설정하려면 다음 절차를 사용합니다.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 선택한 파일 편집기를 시작하여 `/etc/ipf/pfil.ap` 파일을 편집합니다.

이 파일에는 호스트의 NIC 이름이 있습니다. 기본적으로 이름은 주석 처리되어 있습니다. 필터링할 네트워크 트래픽을 전달하는 장치 이름을 주석 해제합니다. 시스템의 NIC 이름이 없으면 NIC를 지정하는 라인을 추가합니다.

```
# vi /etc/ipf/pfil.ap
# IP Filter pfil autopush setup
#
# See autopush(1M) manpage for more information.
#
# Format of the entries in this file is:
#
#major  minor lastminor modules

#le     -1      0       pfil
#qe     -1      0       pfil
hme     -1      0       pfil (Device has been uncommented for filtering)
#qfe    -1      0       pfil
#eri    -1      0       pfil
#ce     -1      0       pfil
#bge    -1      0       pfil
#be     -1      0       pfil
#vge    -1      0       pfil
#ge     -1      0       pfil
#nf     -1      0       pfil
#fa     -1      0       pfil
#ci     -1      0       pfil
#el     -1      0       pfil
#ipdptp -1      0       pfil
#lane   -1      0       pfil
#dmfe   -1      0       pfil
```

3 **network/pfil 서비스 인스턴스를 다시 시작하여 /etc/ipf/pfil.ap 파일에 대한 변경 사항을 활성화합니다.**

```
# svcadm restart network/pfil
```

4 **다음 방법 중 하나를 사용하여 NIC를 사용으로 설정합니다.**

- 시스템을 재부트합니다.

```
# reboot
```

주 – NIC에서 `ifconfig unplumb` 및 `ifconfig plumb` 명령을 안전하게 사용할 수 없는 경우 재부트가 필요합니다.

- `ifconfig` 명령과 `unplumb` 및 `plumb` 옵션을 사용하여 필터링할 NIC를 사용으로 설정합니다. IPv6 패킷 필터링을 구현하려면 각 인터페이스의 `inet6` 버전을 연결해야 합니다.

```
# ifconfig hme0 unplumb
# ifconfig hme0 plumb 192.168.1.20  netmask 255.255.255.0  up
# ifconfig hme0 inet6 unplumb
# ifconfig hme0 inet6 plumb fec3:f840::1/96 up
```

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

▼ NIC에서 IP 필터를 비활성화하는 방법

NIC에서 패킷 필터링을 중지하려면 다음 절차를 사용합니다.

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **선택한 파일 편집기를 시작하여 /etc/ipf/pfil.ap 파일을 편집합니다.**

이 파일에는 호스트의 NIC 이름이 있습니다. 네트워크 트래픽을 필터링하는 데 사용되는 NIC는 주석 해제되어 있습니다. 더 이상 네트워크 트래픽을 필터링하는 데 사용하지 않을 장치 이름을 주석 처리합니다.

```
# vi /etc/ipf/pfil.ap
# IP Filter pfil autopush setup
#
# See autopush(1M) manpage for more information.
#
# Format of the entries in this file is:
#
#major  minor lastminor modules
```

```
#le     -1      0       pfil
#qe     -1      0       pfil
#hme    -1      0       pfil (Commented-out device no longer filters network traffic)
#qfe    -1      0       pfil
#eri    -1      0       pfil
#ce     -1      0       pfil
#bge    -1      0       pfil
#be     -1      0       pfil
#vge    -1      0       pfil
#ge     -1      0       pfil
#nf     -1      0       pfil
#fa     -1      0       pfil
#ci     -1      0       pfil
#el     -1      0       pfil
#ipdptp -1      0       pfil
#lane   -1      0       pfil
#dmfe   -1      0       pfil
```

3 다음 방법 중 하나를 사용하여 NIC를 비활성화합니다.

- 시스템을 재부트합니다.

  ```
  # reboot
  ```

  주 – NIC에서 `ifconfig unplumb` 및 `ifconfig plumb` 명령을 안전하게 사용할 수 없는 경우 재부트가 필요합니다.

- `ifconfig` 명령과 `unplumb` 및 `plumb` 옵션을 사용하여 NIC를 비활성화합니다. IPv6 패킷 필터링을 비활성화하려면 각 인터페이스의 `inet6` 버전을 연결 해제해야 합니다. 다음 단계를 수행하십시오. 시스템의 샘플 장치는 `hme`입니다.

  a. 비활성화 중인 장치의 주 번호를 식별합니다.

  ```
  # grep hme /etc/name_to_major
  hme 7
  ```

  b. `hme0.`의 현재 `autopush` 구성을 표시합니다.

  ```
  # autopush -g -M 7 -m 0
     Major     Minor     Lastminor       Modules
         7       ALL           -           pfil
  ```

  c. `autopush` 구성을 제거합니다.

  ```
  # autopush -r -M 7 -m 0
  ```

  d. 장치를 열고 장치에 IP 주소를 지정합니다.

  ```
  # ifconfig hme0 unplumb
  # ifconfig hme0 plumb 192.168.1.20  netmask 255.255.255.0  up
  # ifconfig hme0 inet6 unplumb
  # ifconfig hme0 inet6 plumb fec3:f840::1/96 up
  ```

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

▼ IP 필터에 대한 pfil 통계를 보는 방법

IP 필터의 문제를 해결할 때 `pfil` 통계를 볼 수 있습니다.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 `pfil` 통계 보기

```
# ndd -get /dev/pfil qif_status
```

예 26–1 IP 필터에 대한 pfil 통계 보기

다음 예에서는 **`pfil`** 통계를 보는 방법을 보여 줍니다.

```
# ndd -get /dev/pfil qif_status
ifname ill q OTHERQ num sap hl nr nw bad copy copyfail drop notip nodata
   notdata
QIF6 0 300011247b8 300011248b0 6 806 0 4 9 0 0 0 0 0 0 0
dmfe1 3000200a018 30002162a50 30002162b48 5 800 14 171 13681 0 0 0 0 0 0 0
```

IP 필터 규칙 세트 작업

다음 작업 맵에서는 IP 필터 규칙 세트와 관련된 절차를 식별합니다.

표 26–4 IP 필터 규칙 세트 작업(작업 맵)

작업	설명	수행 방법
IP 필터 패킷 필터링 규칙 세트를 관리, 확인 및 수정합니다.		626 페이지 "IP 필터에 대한 패킷 필터링 규칙 세트 관리"
	활성 패킷 필터링 규칙 세트를 확인합니다.	627 페이지 "활성 패킷 필터링 규칙 세트 확인 방법"
	비활성 패킷 필터링 규칙 세트를 확인합니다.	627 페이지 "비활성 패킷 필터링 규칙 세트 확인 방법"
	다른 활성 규칙 세트를 활성화합니다.	628 페이지 "다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법"
	규칙 세트를 제거합니다.	629 페이지 "패킷 필터링 규칙 세트 제거 방법"

표 26–4 IP 필터 규칙 세트 작업(작업 맵) (계속)

작업	설명	수행 방법
	규칙 세트에 규칙을 추가합니다.	630 페이지 “활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법”
		631 페이지 “비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법”
	활성 규칙 세트와 비활성 규칙 세트 간에 전환합니다.	631 페이지 “활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법”
	커널에서 비활성 규칙 세트를 삭제합니다.	632 페이지 “커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법”
IP 필터 NAT 규칙을 관리, 확인 및 수정합니다.		633 페이지 “IP 필터에 대한 NAT 규칙 관리”
	활성 NAT 규칙을 확인합니다.	633 페이지 “활성 NAT 규칙 확인 방법”
	NAT 규칙을 제거합니다.	634 페이지 “NAT 규칙 제거 방법”
	NAT 규칙에 다른 규칙을 추가합니다.	634 페이지 “NAT 규칙에 규칙을 추가하는 방법”
IP 필터 주소 풀을 관리, 확인 및 수정합니다.		635 페이지 “IP 필터에 대한 주소 풀 관리”
	활성 주소 풀을 확인합니다.	635 페이지 “활성 주소 풀 확인 방법”
	주소 풀을 제거합니다.	636 페이지 “주소 풀 제거 방법”
	주소 풀에 다른 규칙을 추가합니다.	636 페이지 “주소 풀에 규칙을 추가하는 방법”

IP 필터에 대한 패킷 필터링 규칙 세트 관리

사용으로 설정된 경우 활성 및 비활성 패킷 필터링 규칙 세트가 모두 커널에 상주할 수 있습니다. 활성 규칙 세트에 따라 수신 패킷 및 송신 패킷에 대해 수행하려는 필터링이 결정됩니다. 비활성 규칙 세트도 규칙을 저장합니다. 비활성 규칙 세트를 활성 규칙 세트로 설정하지 않은 경우 해당 규칙이 사용되지 않습니다. 활성 및 비활성 패킷 필터링 규칙 세트를 모두 관리, 확인 및 수정할 수 있습니다.

▼ 활성 패킷 필터링 규칙 세트 확인 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **커널에서 로드된 활성 패킷 필터링 규칙 세트를 확인합니다.**

```
# ipfstat -io
```

예 26–2 활성 패킷 필터링 규칙 세트 보기

다음 예에서는 커널에서 로드된 활성 패킷 필터링 규칙 세트의 출력을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
pass in quick on dmfe1 from 192.168.1.0/24 to any
pass in all
block in on dmfe1 from 192.168.1.10/32 to any
```

▼ 비활성 패킷 필터링 규칙 세트 확인 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **비활성 패킷 필터링 규칙 세트를 확인합니다.**

```
# ipfstat -I -io
```

예 26–3 비활성 패킷 필터링 규칙 세트 보기

다음 예에서는 비활성 패킷 필터링 규칙 세트의 출력을 보여 줍니다.

```
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
```

▼ 다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법

다음 작업 중 하나를 수행하려면 이 절차를 사용하십시오.

- 현재 IP 필터에 사용되고 있는 규칙 세트가 아닌 다른 패킷 필터링 규칙 세트를 활성화합니다.
- 새로 업데이트된 동일한 필터링 규칙 세트를 다시 로드합니다.

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 다음 단계 중 하나를 선택합니다.

- 완전히 다른 규칙 세트를 활성화하려면 선택한 별도의 파일에 새 규칙 세트를 만듭니다.
- 해당 규칙 세트를 포함하는 구성 파일을 편집하여 현재 규칙 세트를 업데이트합니다.

3 현재 규칙 세트를 제거하고 새 규칙 세트를 로드합니다.

```
# ipf -Fa -f filename
```

*filename*은 새 규칙 세트를 포함하는 새 파일 또는 활성 규칙 세트를 포함하는 업데이트된 파일일 수 있습니다.

커널에서 활성 규칙 세트가 제거되고, *filename* 파일의 규칙이 활성 규칙 세트가 됩니다.

주 – 현재 구성 파일을 다시 로드하는 중인 경우에도 명령을 실행해야 합니다. 그렇지 않으면 기존 규칙 세트가 계속 작동하고 업데이트된 구성 파일의 수정된 규칙 세트가 적용되지 않습니다.

업데이트된 규칙 세트를 로드하려면 `ipf -D`, `svcadm restart` 등의 명령을 사용하지 마십시오. 새 규칙 세트를 로드하기 전에 먼저 방화벽을 사용 안함으로 설정하면 해당 명령으로 인해 네트워크가 노출됩니다.

예 26–4 다른 패킷 필터링 규칙 세트 활성화

다음 예에서는 특정 패킷 필터링 규칙 세트를 별도의 구성 파일 `/etc/ipf/ipf.conf`에 있는 다른 패킷 필터링 규칙 세트로 바꾸는 방법을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
pass in quick on dmfe all
# ipf -Fa -f /etc/ipf/ipf.conf
# ipfstat -io
```

```
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
```

예 26–5 업데이트된 패킷 필터링 규칙 세트 다시 로드

다음 예에서는 현재 활성 상태이며 업데이트된 패킷 필터링 규칙 세트를 다시 로드하는 방법을 보여 줍니다. 이 예에서 사용하는 파일은 `/etc/ipf/ipf.conf`입니다.

```
# ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any

(Edit the /etc/ipf/ipf.conf configuration file.)

# ipf -Fa -f /etc/ipf/ipf.conf
# ipfstat -io (Optional)
empty list for ipfilter (out)
block in log quick from 10.0.0.0/8 to any
block in quick on elx10 from 192.168.0.0/12 to any
```

▼ 패킷 필터링 규칙 세트 제거 방법

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 규칙 세트를 제거합니다.

```
# ipf -F [a|i|o]
```

`-a` 규칙 세트에서 모든 필터링 규칙을 제거합니다.

`-i` 수신 패킷에 대한 필터링 규칙을 제거합니다.

`-o` 송신 패킷에 대한 필터링 규칙을 제거합니다.

예 26–6 패킷 필터링 규칙 세트 제거

다음 예에서는 활성 필터링 규칙 세트에서 모든 필터링 규칙을 제거하는 방법을 보여 줍니다.

```
# ipfstat -io
block out log on dmf0 all
block in log quick from 10.0.0.0/8 to any
# ipf -Fa
# ipfstat -io
empty list for ipfilter(out)
empty list for ipfilter(in)
```

▼ 활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.**

- `ipf -f -` 명령을 사용하여 명령줄에서 규칙 세트에 규칙을 추가합니다.

  ```
  # echo "block in on dmfe1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
  ```

- 다음 명령을 실행합니다.

  a. 선택한 파일에 규칙 세트를 만듭니다.

  b. 만든 규칙을 활성 규칙 세트에 추가합니다.

  ```
  # ipf -f filename
  ```

  활성 규칙 세트의 끝에 *filename*의 규칙이 추가됩니다. IP 필터는 "마지막 일치 규칙" 알고리즘을 사용하므로 `quick` 키워드를 사용하지 않는 경우 추가되는 규칙에 따라 필터링 우선 순위가 결정됩니다. 패킷이 `quick` 키워드를 포함하는 규칙과 일치하는 경우 해당 규칙에 대한 작업이 수행되고 후속 규칙이 확인되지 않습니다.

예 26–7 활성 패킷 필터링 규칙 세트에 규칙 추가

다음 예에서는 명령줄에서 활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
# echo "block in on dmfe1 proto tcp from 10.1.1.1/32 to any" | ipf -f -
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
```

▼ 비활성 패킷 필터링 규칙 세트에 규칙을 추가하는 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **선택한 파일에 규칙 세트를 만듭니다.**

3 **만든 규칙을 비활성 규칙 세트에 추가합니다.**

```
# ipf -I -f filename
```

비활성 규칙 세트의 끝에 *filename*의 규칙이 추가됩니다. IP 필터는 "마지막 일치 규칙" 알고리즘을 사용하므로 `quick` 키워드를 사용하지 않는 경우 추가되는 규칙에 따라 필터링 우선 순위가 결정됩니다. 패킷이 `quick` 키워드를 포함하는 규칙과 일치하는 경우 해당 규칙에 대한 작업이 수행되고 후속 규칙이 확인되지 않습니다.

예 26–8 비활성 규칙 세트에 규칙 추가

다음 예에서는 파일에서 비활성 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
# ipf -I -f /etc/ipf/ipf.conf
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
```

▼ 활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **활성 규칙 세트와 비활성 규칙 세트 간에 전환합니다.**

```
# ipf -s
```

이 명령을 사용하면 커널에서 활성 규칙 세트와 비활성 규칙 세트 간에 전환할 수 있습니다. 비활성 규칙 세트가 비어 있을 경우 패킷 필터링이 없는 것입니다.

예 26–9 활성 패킷 필터링 규칙 세트와 비활성 패킷 필터링 규칙 세트 간 전환

다음 예에서는 ipf -s 명령을 사용하여 비활성 규칙 세트를 활성 규칙 세트로 전환하고 활성 규칙 세트를 비활성 규칙 세트로 전환하는 방법을 보여 줍니다.

- ipf -s 명령을 실행하기 전에 ipfstat -I -io 명령의 출력은 비활성 규칙 세트의 규칙을 보여 줍니다. ipfstat -io 명령의 출력은 활성 규칙 세트의 규칙을 보여 줍니다.

```
# ipfstat -io
empty list for ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
# ipfstat -I -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
```

- ipf -s 명령을 실행한 후 ipfstat -I -io 및 ipfstat -io 명령의 출력은 두 개 규칙 세트의 내용이 전환되었음을 보여 줍니다.

```
# ipf -s
Set 1 now inactive
# ipfstat -io
pass out quick on dmfe1 all
pass in quick on dmfe1 all
block in log quick from 10.0.0.0/8 to any
# ipfstat -I -io
empty list for inactive ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
```

▼ 커널에서 비활성 패킷 필터링 규칙 세트를 제거하는 방법

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 "모두 비우기" 명령에 비활성 규칙 세트를 지정합니다.

```
# ipf -I -Fa
```

이 명령은 커널에서 비활성 규칙 세트를 비웁니다.

주 – 나중에 ipf -s를 실행할 경우 비어 있는 비활성 규칙 세트가 활성 규칙 세트로 전환됩니다. 활성 규칙 세트가 비어 있을 경우 필터링이 수행되지 **않습니다**.

예 26–10 커널에서 비활성 패킷 필터링 규칙 세트 제거

다음 예에서는 모든 규칙이 제거되도록 비활성 패킷 필터링 규칙 세트를 비우는 방법을 보여 줍니다.

```
# ipfstat -I -io
empty list for inactive ipfilter(out)
block in log quick from 10.0.0.0/8 to any
block in on dmfe1 proto tcp from 10.1.1.1/32 to any
# ipf -I -Fa
# ipfstat -I -io
empty list for inactive ipfilter(out)
empty list for inactive ipfilter(in)
```

IP 필터에 대한 NAT 규칙 관리

다음 절차에 따라 NAT 규칙을 관리, 확인 및 수정할 수 있습니다.

▼ 활성 NAT 규칙 확인 방법

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 활성 NAT 규칙을 확인합니다.

```
# ipnat -l
```

예 26–11 활성 NAT 규칙 보기

다음 예에서는 활성 NAT 규칙 세트의 출력을 보여 줍니다.

```
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
```

▼ NAT 규칙 제거 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **현재 NAT 규칙을 제거합니다.**

```
# ipnat -C
```

예 26–12 NAT 규칙 제거

다음 예에서는 현재 NAT 규칙의 항목을 제거하는 방법을 보여 줍니다.

```
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
# ipnat -C
1 entries flushed from NAT list
# ipnat -l
List of active MAP/Redirect filters:

List of active sessions:
```

▼ NAT 규칙에 규칙을 추가하는 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.**

- `ipnat -f -` 명령을 사용하여 명령줄에서 NAT 규칙 세트에 규칙을 추가합니다.

  ```
  # echo "map dmfe0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
  ```

- 다음 명령을 실행합니다.

  a. 선택한 파일에 추가 NAT 규칙을 만듭니다.

  b. 만든 규칙을 활성 NAT 규칙에 추가합니다.

  ```
  # ipnat -f filename
  ```

NAT 규칙의 끝에 *filename*의 규칙이 추가됩니다.

예 26–13 NAT 규칙 세트에 규칙 추가

다음 예에서는 명령줄에서 NAT 규칙 세트에 규칙을 추가하는 방법을 보여 줍니다.

```
# ipnat -l
List of active MAP/Redirect filters:

List of active sessions:
# echo "map dmfe0 192.168.1.0/24 -> 20.20.20.1/32" | ipnat -f -
# ipnat -l
List of active MAP/Redirect filters:
map dmfe0 192.168.1.0/24 -> 20.20.20.1/32

List of active sessions:
```

IP 필터에 대한 주소 풀 관리

다음 절차에 따라 주소 풀을 관리, 확인 및 수정할 수 있습니다.

▼ 활성 주소 풀 확인 방법

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 활성 주소 풀을 확인합니다.

```
# ippool -l
```

예 26–14 활성 주소 풀 보기

다음 예에서는 활성 주소 풀의 컨텐츠를 확인하는 방법을 보여 줍니다.

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

▼ 주소 풀 제거 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **현재 주소 풀의 항목을 제거합니다.**

```
# ippool -F
```

예 26–15 주소 풀 제거

다음 예에서는 주소 풀 제거 방법을 보여 줍니다.

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
# ippool -F
1 object flushed
# ippool -l
```

▼ 주소 풀에 규칙을 추가하는 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **다음 방법 중 하나로 활성 규칙 세트에 규칙을 추가합니다.**

- `ippool -f -` 명령을 사용하여 명령줄에서 규칙 세트에 규칙을 추가합니다.

  ```
  # echo "table role = ipf type = tree number = 13
  {10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24};" | ippool -f -
  ```

- 다음 명령을 실행합니다.

  a. 선택한 파일에 추가 주소 풀을 만듭니다.

  b. 만든 규칙을 활성 주소 풀에 추가합니다.

  ```
  # ippool -f filename
  ```

  활성 주소 풀의 끝에 *filename*의 규칙이 추가됩니다.

예 26–16 주소 풀에 규칙 추가

다음 예에서는 명령줄에서 주소 풀 규칙 세트에 주소 풀을 추가하는 방법을 보여 줍니다.

```
# ippool -l
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
# echo "table role = ipf type = tree number = 100
 {10.0.0.0/32, 172.16.1.2/32, 192.168.1.0/24};" | ippool -f -
# ippool -l
table role = ipf type = tree number = 100
        { 10.0.0.0/32, 172.16.1.2/32, 192.168.1.0/24; };
table role = ipf type = tree number = 13
        { 10.1.1.1/32, 10.1.1.2/32, 192.168.1.0/24; };
```

IP 필터에 대한 통계 및 정보 표시

표 26–5 IP 필터 통계 및 정보 표시(작업 맵)

작업	설명	수행 방법
상태 테이블을 확인합니다.	`ipfstat` 명령을 사용하여 패킷 필터링에 대한 정보를 얻을 수 있는 상태 테이블을 확인합니다.	637 페이지 "IP 필터에 대한 상태 테이블 확인 방법"
상태 통계를 확인합니다.	`ipfstat -s` 명령을 사용하여 패킷 상태 정보에 대한 통계를 확인합니다.	638 페이지 "IP 필터에 대한 상태 통계 확인 방법"
NAT 통계를 확인합니다.	`ipnat -s` 명령을 사용하여 NAT 통계를 확인합니다.	639 페이지 "IP 필터에 대한 NAT 통계 확인 방법"
주소 풀 통계를 확인합니다.	`ippool -s` 명령을 사용하여 주소 풀 통계를 확인합니다.	640 페이지 "IP 필터에 대한 주소 풀 통계 확인 방법"

▼ IP 필터에 대한 상태 테이블 확인 방법

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 상태 테이블을 확인합니다.

```
# ipfstat
```

주 – -t 옵션을 사용하여 최상위 유틸리티 형식으로 상태 테이블을 확인할 수 있습니다.

예 26–17 IP 필터에 대한 상태 테이블 보기

다음 예에서는 상태 테이블 확인 방법을 보여 줍니다.

```
# ipfstat
bad packets:            in 0    out 0
 input packets:         blocked 160 passed 11 nomatch 1 counted 0 short 0
output packets:         blocked 0 passed 13681 nomatch 6844 counted 0 short 0
 input packets logged:  blocked 0 passed 0
output packets logged:  blocked 0 passed 0
 packets logged:        input 0 output 0
 log failures:          input 0 output 0
fragment state(in):     kept 0  lost 0
fragment state(out):    kept 0  lost 0
packet state(in):       kept 0  lost 0
packet state(out):      kept 0  lost 0
ICMP replies:   0       TCP RSTs sent:  0
Invalid source(in):     0
Result cache hits(in):  152     (out):  6837
IN Pullups succeeded:   0       failed: 0
OUT Pullups succeeded:  0       failed: 0
Fastroute successes:    0       failures:       0
TCP cksum fails(in):    0       (out):  0
IPF Ticks:      14341469
Packet log flags set: (0)
        none
```

▼ IP 필터에 대한 상태 통계 확인 방법

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”을 참조하십시오.

2 상태 통계를 확인합니다.

```
# ipfstat -s
```

예 26–18 IP 필터에 대한 상태 통계 보기

다음 예에서는 상태 통계 확인 방법을 보여 줍니다.

```
# ipfstat -s
IP states added:
        0 TCP
```

```
        0 UDP
        0 ICMP
        0 hits
        0 misses
        0 maximum
        0 no memory
        0 max bucket
        0 active
        0 expired
        0 closed
State logging enabled

State table bucket statistics:
        0 in use
        0.00% bucket usage
        0 minimal length
        0 maximal length
        0.000 average length
```

▼ IP 필터에 대한 NAT 통계 확인 방법

1. **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

   만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2. **NAT 통계를 확인합니다.**

   ```
   # ipnat -s
   ```

예 26–19 IP 필터에 대한 NAT 통계 보기

다음 예에서는 NAT 통계 확인 방법을 보여 줍니다.

```
# ipnat -s
mapped  in      0       out     0
added   0       expired 0
no memory       0       bad nat 0
inuse   0
rules   1
wilds   0
```

▼ IP 필터에 대한 주소 풀 통계 확인 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **주소 풀 통계를 확인합니다.**

```
# ippool -s
```

예 26–20 IP 필터에 대한 주소 풀 통계 보기

다음 예에서는 주소 풀 통계 확인 방법을 보여 줍니다.

```
# ippool -s
Pools:  3
Hash Tables:    0
Nodes:  0
```

IP 필터 로그 파일 작업

표 26–6 IP 필터 로그 파일 작업(작업 맵)

작업	설명	수행 방법
로그 파일을 만듭니다.	별도의 IP 필터 로그 파일을 만듭니다.	640 페이지 "IP 필터 로그 파일 설정 방법"
로그 파일을 확인합니다.	`ipmon` 명령을 사용하여 상태, NAT 및 일반 로그 파일을 확인합니다.	641 페이지 "IP 필터 로그 파일 확인 방법"
패킷 로그 버퍼를 비웁니다.	`ipmon -F` 명령을 사용하여 패킷 로그 버퍼의 컨텐츠를 제거합니다.	643 페이지 "패킷 로그 파일을 비우는 방법"
기록된 패킷을 파일에 저장합니다.	나중에 참조할 수 있도록 기록된 패킷을 파일에 저장합니다.	643 페이지 "기록된 패킷을 파일에 저장하는 방법"

▼ IP 필터 로그 파일 설정 방법

기본적으로 IP 필터에 대한 모든 로그 정보는 `syslogd` 파일에 기록됩니다. 기본 로그 파일에 기록될 수 있는 다른 데이터와 별도로 IP 필터 트래픽 정보가 기록되도록 로그 파일을 설정해야 합니다. 다음 단계를 수행하십시오.

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **다음 두 행을 추가하여 `/etc/syslog.conf` 파일을 편집합니다.**

```
# Save IP Filter log output to its own file
local0.debug              /var/log/log-name
```

주 – 두번째 행에서 스페이스바가 아닌 Tab 키를 사용하여 `local0.debug`와 `/var/log/log-name`을 구분해야 합니다.

---

3 **새 로그 파일을 만듭니다.**

```
# touch /var/log/log-name
```

4 **system-log 서비스를 다시 시작합니다.**

```
# svcadm restart system-log
```

예 26–21 IP 필터 로그 만들기

다음 예에서는 IP 필터 정보를 아카이브할 `ipmon.log`를 만드는 방법을 보여 줍니다.

`/etc/syslog.conf`에서 다음을 입력합니다.

```
# Save IP Filter log output to its own file
local0.debug              /var/log/ipmon.log
```

명령줄에서 다음을 입력합니다.

```
# touch /var/log/ipmon.log
# svcadm restart system-log
```

▼ IP 필터 로그 파일 확인 방법

시작하기 전에 IP 필터 데이터를 기록할 별도의 로그 파일을 만들어야 합니다. 640 페이지 "IP 필터 로그 파일 설정 방법"을 참조하십시오.

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 상태, NAT 또는 일반 로그 파일을 확인합니다. 로그 파일을 보려면 적합한 옵션을 사용하여 다음 명령을 입력합니다.

 # ipmon -o [S|N|I] *filename*

S 상태 로그 파일을 표시합니다.

N NAT 로그 파일을 표시합니다.

I 일반 IP 로그 파일을 표시합니다.

모든 상태, NAT 및 일반 로그 파일을 보려면 옵션을 모두 사용합니다.

 # ipmon -o SNI *filename*

- **먼저 수동으로 ipmon 데몬을 중지한 경우 다음 명령을 사용하여 상태, NAT 및 IP 필터 로그 파일을 표시할 수도 있습니다.**

  # ipmon -a *filename*

주 – ipmon 데몬이 아직 실행 중인 경우 ipmon -a 구문을 사용하지 마십시오. 일반적으로 데몬은 시스템 부트 시 자동으로 시작됩니다. ipmon -a 명령을 실행하면 ipmon의 다른 복사본이 열립니다. 이 경우 두 복사본은 동일한 로그 정보를 읽고 하나의 복사본만 특정 로그 메시지를 가져옵니다.

로그 파일 확인에 대한 자세한 내용은 ipmon(1M) 매뉴얼 페이지를 참조하십시오.

예 26–22 IP 필터 로그 파일 보기

다음 예에서는 /var/ipmon.log의 출력을 보여 줍니다.

```
# ipmon -o SNI /var/ipmon.log
02/09/2004 15:27:20.606626 hme0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

또는

```
# pkill ipmon
# ipmon -aD /var/ipmon.log
02/09/2004 15:27:20.606626 hme0 @0:1 p 129.146.157.149 ->
129.146.157.145 PR icmp len 20 84 icmp echo/0 IN
```

▼ 패킷 로그 파일을 비우는 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **패킷 로그 버퍼를 비웁니다.**

```
# ipmon -F
```

예 26–23 패킷 로그 파일 비우기

다음 예에서는 로그 파일 제거 시 출력을 보여 줍니다. 시스템에서는 이 예에서와 같이 로그 파일에 저장된 항목이 없는 경우에도 보고서를 제공합니다.

```
# ipmon -F
0 bytes flushed from log buffer
0 bytes flushed from log buffer
0 bytes flushed from log buffer
```

▼ 기록된 패킷을 파일에 저장하는 방법

1 **IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.**

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 **기록된 패킷을 파일에 저장합니다.**

```
# cat /dev/ipl > filename
```

명령줄 프롬프트를 다시 가져올 `Ctrl-C`를 입력하여 프로시저를 중단할 때까지 *filename* 파일에 패킷이 계속 기록됩니다.

예 26–24 기록된 패킷을 파일에 저장

다음 예에서는 기록된 패킷을 파일에 저장한 후의 결과를 보여 줍니다.

```
# cat /dev/ipl > /tmp/logfile
^C#

# ipmon -f /tmp/logfile
```

```
02/09/2004 15:30:28.708294 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 52 -S IN
02/09/2004 15:30:28.708708 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.792611 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 70 -AP IN
02/09/2004 15:30:28.872000 hme0 @0:1 p 129.146.157.149,33923 ->
 129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.872142 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 43 -AP IN
02/09/2004 15:30:28.872808 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
02/09/2004 15:30:28.872951 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 47 -AP IN
02/09/2004 15:30:28.926792 hme0 @0:1 p 129.146.157.149,33923 ->
  129.146.157.145,23 PR tcp len 20 40 -A IN
.
.
(output truncated)
```

IP 필터 구성 파일 만들기 및 편집

규칙 세트 및 주소 풀을 만들고 수정하려면 구성 파일을 직접 편집해야 합니다. 구성 파일은 표준 UNIX 구문 규칙을 따릅니다.

- 파운드 기호(#)는 행에 주석이 포함되어 있음을 나타냅니다.
- 규칙과 주석은 동일한 행에 함께 사용될 수 있습니다.
- 규칙을 쉽게 읽을 수 있도록 임의로 공백을 사용할 수 있습니다.
- 규칙의 길이는 두 행 이상일 수 있습니다. 행 끝에 백슬래시(\)를 사용하여 규칙이 다음 행에서 계속됨을 나타낼 수 있습니다.

▼ IP 필터에 대한 구성 파일을 만드는 방법

이 절차에서는 다음을 설정하는 방법에 대해 설명합니다.

- 패킷 필터링 구성 파일
- NAT 규칙 구성 파일
- 주소 풀 구성 파일

1 IP Filter Management 권한 프로파일이 포함된 역할의 사용자 또는 수퍼 유저로 로그인합니다.

만든 역할에 IP Filter Management 권한 프로파일을 지정할 수 있습니다. 역할을 만들어 사용자에게 지정하려면 **System Administration Guide: Security Services**의 "Configuring RBAC (Task Map)"을 참조하십시오.

2 선택한 파일 편집기를 시작합니다. 구성할 기능에 대한 구성 파일을 만들거나 편집합니다.

- 패킷 필터링 규칙에 대한 구성 파일을 만들려면 `ipf.conf` 파일을 편집합니다.

  IP 필터는 `ipf.conf` 파일에 배치된 패킷 필터링 규칙을 사용합니다. 패킷 필터링에 대한 규칙 파일을 `/etc/ipf/ipf.conf` 파일에 배치할 경우 시스템 부트 시 이 파일이 로드됩니다. 부트 시 필터링 규칙이 로드되지 않도록 하려면 선택한 파일에 배치합니다. 그런 다음 628 페이지 "다른 또는 업데이트된 패킷 필터링 규칙 세트 활성화 방법"에 설명된 대로 `ipf` 명령을 사용하여 규칙을 활성화할 수 있습니다.

  패킷 필터링 규칙을 만드는 방법은 603 페이지 "IP 필터의 패킷 필터링 기능 사용"을 참조하십시오.

  주 – `ipf.conf` 파일이 비어 있을 경우 필터링이 없는 것입니다. `ipf.conf` 파일이 비어 있을 경우 다음을 읽는 규칙 세트가 있는 것과 동일합니다.

  ```
  pass in all
  pass out all
  ```

- NAT 규칙에 대한 구성 파일을 만들려면 `ipnat.conf` 파일을 편집합니다.

  IP 필터는 `ipnat.conf` 파일에 배치된 NAT 규칙을 사용합니다. NAT에 대한 규칙 파일을 `/etc/ipf/ipnat.conf` 파일에 배치할 경우 시스템 부트 시 이 파일이 로드됩니다. 부트 시 NAT 규칙이 로드되지 않도록 하려면 선택한 위치에 `ipnat.conf` 파일을 배치합니다. 그런 다음 `ipnat` 명령을 사용하여 NAT 규칙을 활성화할 수 있습니다.

  NAT에 대한 규칙을 만드는 방법은 606 페이지 "IP 필터의 NAT 기능 사용"을 참조하십시오.

- 주소 풀에 대한 구성 파일을 만들려면 `ippool.conf` 파일을 편집합니다.

  IP 필터는 `ippool.conf` 파일에 배치된 주소 풀을 사용합니다. 주소 풀에 대한 규칙 파일을 `/etc/ipf/ippool.conf` 파일에 배치할 경우 시스템 부트 시 이 파일이 로드됩니다. 부트 시 주소 풀이 로드되지 않도록 하려면 선택한 위치에 `ippool.conf` 파일을 배치합니다. 그런 다음 `ippool` 명령을 사용하여 주소 풀을 활성화할 수 있습니다.

  주소 풀을 만드는 방법은 607 페이지 "IP 필터의 주소 풀 기능 사용"을 참조하십시오.

IP 필터 구성 파일 예

다음 예에서는 필터링 구성에 사용되는 패킷 필터링 규칙의 실례를 제공합니다.

예 26–25 IP 필터 호스트 구성

이 예에서는 elxl 네트워크 인터페이스가 있는 호스트 시스템에 대한 구성을 보여 줍니다.

```
# pass and log everything by default
pass in log on bge0 all
pass out log on bge0 all

# block, but don't log, incoming packets from other reserved addresses
block in quick on bge0 from 10.0.0.0/8 to any
block in quick on bge0 from 172.16.0.0/12 to any

# block and log untrusted internal IPs. 0/32 is notation that replaces
# address of the machine running Solaris IP Filter.
block in log quick from 192.168.1.15 to <thishost>
block in log quick from 192.168.1.43 to <thishost>

# block and log X11 (port 6000) and remote procedure call
# and portmapper (port 111) attempts
block in log quick on bge0 proto tcp from any to bge0/32 port = 6000 keep state
block in log quick on bge0 proto tcp/udp from any to bge0/32 port = 111 keep state
```

이 규칙 세트는 elxl 인터페이스에서 모든 항목을 주고받을 수 있도록 허용하는 제한되지 않은 두 개의 규칙으로 시작합니다. 두번째 규칙 세트는 개인 주소 공간 10.0.0.0 및 172.16.0.0의 수신 패킷이 방화벽에 들어오지 못하도록 차단합니다. 다음 규칙 세트는 호스트 시스템의 특정 내부 주소를 차단합니다. 마지막 규칙 세트는 포트 6000 및 포트 111에서 수신되는 패킷을 차단합니다.

예 26–26 IP 필터 서버 구성

이 예에서는 웹 서버로 사용되는 호스트 시스템에 대한 구성을 보여 줍니다. 이 시스템에는 eri 네트워크 인터페이스가 있습니다.

```
# web server with an eri interface
# block and log everything by default; then allow specific services
# group 100 - inbound rules
# group 200 - outbound rules
# (0/32) resolves to our IP address)
*** FTP proxy ***


# block short packets which are packets fragmented too short to be real.
block in log quick all with short


# block and log inbound and outbound by default, group by destination
block in log on eri0 from any to any head 100
block out log on eri0 from any to any head 200


# web rules that get hit most often
pass in quick on eri0 proto tcp from any \
to eri0/32 port = http flags S keep state group 100
```

예 26–26 IP 필터 서버 구성 *(계속)*

```
pass in quick on eri0 proto tcp from any \
to eri0/32 port = https flags S keep state group 100


# inbound traffic - ssh, auth
pass in quick on eri0 proto tcp from any \
to eri0/32 port = 22 flags S keep state group 100
pass in log quick on eri0 proto tcp from any \
to eri0/32 port = 113 flags S keep state group 100
pass in log quick on eri0 proto tcp from any port = 113 \
to eri0/32 flags S keep state group 100


# outbound traffic - DNS, auth, NTP, ssh, WWW, smtp
pass out quick on eri0 proto tcp/udp from eri0/32 \
to any port = domain flags S keep state group 200
pass in quick on eri0 proto udp from any port = domain to eri0/32 group 100

pass out quick on eri0 proto tcp from eri0/32 \
to any port = 113 flags S keep state group 200
pass out quick on eri0 proto tcp from eri0/32 port = 113 \
to any flags S keep state group 200

pass out quick on eri0 proto udp from eri0/32 to any port = ntp group 200
pass in quick on eri0 proto udp from any port = ntp to eri0/32 port = ntp group 100

pass out quick on eri0 proto tcp from eri0/32 \
to any port = ssh flags S keep state group 200

pass out quick on eri0 proto tcp from eri0/32 \
to any port = http flags S keep state group 200
pass out quick on eri0 proto tcp from eri0/32 \
to any port = https flags S keep state group 200

pass out quick on eri0 proto tcp from eri0/32 \
to any port = smtp flags S keep state group 200


# pass icmp packets in and out
pass in quick on eri0 proto icmp from any to eri0/32  keep state group 100
pass out quick on eri0 proto icmp from eri0/32 to any keep state group 200


# block and ignore NETBIOS packets
block in quick on eri0 proto tcp from any \
to any port = 135 flags S keep state group 100

block in quick on eri0 proto tcp from any port = 137 \
to any flags S keep state group 100
block in quick on eri0 proto udp from any to any port = 137 group 100
block in quick on eri0 proto udp from any port = 137 to any group 100

block in quick on eri0 proto tcp from any port = 138 \
to any flags S keep state group 100
block in quick on eri0 proto udp from any port = 138 to any group 100
```

예 26–26 IP 필터 서버 구성 *(계속)*

```
block in quick on eri0 proto tcp from any port = 139 to any flags S keep state
group 100
block in quick on eri0 proto udp from any port = 139 to any group 100
```

예 26–27 IP 필터 라우터 구성

이 예에서는 내부 인터페이스 `ce0` 및 외부 인터페이스 `ce1`이 있는 라우터에 대한 구성을 보여 줍니다.

```
# internal interface is ce0 at 192.168.1.1
# external interface is ce1 IP obtained via DHCP
# block all packets and allow specific services
*** NAT ***
*** POOLS ***


# Short packets which are fragmented too short to be real.
block in log quick all with short


# By default, block and log everything.
block in log on ce0 all
block in log on ce1 all
block out log on ce0 all
block out log on ce1 all


# Packets going in/out of network interfaces that aren't on the loopback
# interface should not exist.
block in log quick on ce0 from 127.0.0.0/8 to any
block in log quick on ce0 from any to 127.0.0.0/8
block in log quick on ce1 from 127.0.0.0/8 to any
block in log quick on ce1 from any to 127.0.0.0/8


# Deny reserved addresses.
block in quick on ce1 from 10.0.0.0/8 to any
block in quick on ce1 from 172.16.0.0/12 to any
block in log quick on ce1 from 192.168.1.0/24 to any
block in quick on ce1 from 192.168.0.0/16 to any


# Allow internal traffic
pass in quick on ce0 from 192.168.1.0/24 to 192.168.1.0/24
pass out quick on ce0 from 192.168.1.0/24 to 192.168.1.0/24


# Allow outgoing DNS requests from our servers on .1, .2, and .3
pass out quick on ce1 proto tcp/udp from ce1/32 to any port = domain keep state
pass in quick on ce0 proto tcp/udp from 192.168.1.2 to any port = domain keep state
pass in quick on ce0 proto tcp/udp from 192.168.1.3 to any port = domain keep state
```

예 26–27 IP 필터 라우터 구성 *(계속)*

```
# Allow NTP from any internal hosts to any external NTP server.
pass in quick on ce0 proto udp from 192.168.1.0/24 to any port = 123 keep state
pass out quick on ce1 proto udp from any to any port = 123 keep state


# Allow incoming mail
pass in quick on ce1 proto tcp from any to ce1/32 port = smtp keep state
pass in quick on ce1 proto tcp from any to ce1/32 port = smtp keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = smtp keep state


# Allow outgoing connections: SSH, WWW, NNTP, mail, whois
pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = 22 keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = 22 keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = 80 keep state
pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = 443 keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = 443 keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = nntp keep state
block in quick on ce1 proto tcp from any to any port = nntp keep state
pass out quick on ce1 proto tcp from 192.168.1.0/24 to any port = nntp keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = smtp keep state

pass in quick on ce0 proto tcp from 192.168.1.0/24 to any port = whois keep state
pass out quick on ce1 proto tcp from any to any port = whois keep state


# Allow ssh from offsite
pass in quick on ce1 proto tcp from any to ce1/32 port = 22 keep state


# Allow ping out
pass in quick on ce0 proto icmp all keep state
pass out quick on ce1 proto icmp all keep state


# allow auth out
pass out quick on ce1 proto tcp from ce1/32 to any port = 113 keep state
pass out quick on ce1 proto tcp from ce1/32 port = 113 to any keep state


# return rst for incoming auth
block return-rst in quick on ce1 proto tcp from any to any port = 113 flags S/SA


# log and return reset for any TCP packets with S/SA
block return-rst in log on ce1 proto tcp from any to any flags S/SA


# return ICMP error packets for invalid UDP packets
block return-icmp(net-unr) in proto udp all
```

제 5 부

IPMP

이 부에서는 IPMP(IP Network Multipathing)를 소개하고 IPMP를 관리하기 위한 작업을 설명합니다. IPMP는 동일한 링크에 연결된 시스템의 인터페이스에 대해 실패 감지 및 페일오버를 제공합니다.

◆ ◆ ◆ 27 장

27 IPMP 소개(개요)

IPMP(IP Network Multipathing)는 동일한 IP 링크에 여러 인터페이스가 있는 시스템에 대해 물리적 인터페이스 장애를 감지하고 투명한 네트워크 액세스 페일오버를 제공합니다. 또한 IPMP는 여러 인터페이스를 가진 시스템에 대해 패킷 로드를 분산시킵니다.

이 장은 다음 정보를 포함합니다.

IPMP 구성 작업에 대해서는 28 장, “IPMP 관리(작업)”를 참조하십시오.

IPMP 사용 이유

IPMP는 다중 물리적 인터페이스로 구성된 시스템에 대해 향상된 안정성, 가용성 및 네트워크 성능을 제공합니다. 경우에 따라 해당 인터페이스에 연결된 네트워킹 하드웨어나 물리적 인터페이스가 실패하거나 유지 관리 작업이 필요할 수 있습니다. 일반적으로는 이 경우 실패한 인터페이스와 연결된 IP 주소를 통해 더 이상 시스템에 연결할 수 없습니다. 또한 이러한 IP 주소를 사용하는 기존의 시스템 연결이 손상됩니다.

IPMP를 사용하여 하나 이상의 물리적 인터페이스를 IP 다중 경로 그룹 또는 *IPMP* **그룹**으로 구성할 수 있습니다. IPMP를 구성한 후 시스템은 IPMP 그룹에서 인터페이스의 실패를 자동으로 모니터링합니다. 그룹의 인터페이스가 실패하거나 유지 관리를 위해 제거된 경우 IPMP는 실패한 인터페이스의 IP 주소를 자동으로 마이그레이션하거나 **페일오버**합니다. 이러한 주소의 수신자는 실패한 인터페이스의 IPMP 그룹에서 작동하는 인터페이스입니다. IPMP의 페일오버 기능은 연결을 유지하고 기존 연결의

중단을 방지합니다. 또한 IPMP는 네트워크 트래픽을 IPMP 그룹의 인터페이스 세트에 자동으로 분산하여 전체 네트워크 성능을 향상합니다. 이러한 프로세스를 **로드 확산**이라고 합니다.

Oracle Solaris IPMP 구성 요소

Oracle Solaris IPMP는 다음 소프트웨어로 구성됩니다.

- `in.mpathd` 데몬에 대해서는 `in.mpathd(1M)` 매뉴얼 페이지에서 자세히 설명합니다.
- `/etc/default/mpathd` 구성 파일에 대해서는 `in.mpathd(1M)` 매뉴얼 페이지에서 설명합니다.
- IPMP 구성의 `ifconfig` 옵션에 대해서는`ifconfig(1M)` 매뉴얼 페이지에서 설명합니다.

다중 경로 데몬, `in.mpathd`

`in.mpathd` 데몬은 인터페이스 실패를 감지하여 페일오버 및 페일백을 위한 다양한 절차를 구현합니다. `in.mpathd`에서 실패 또는 복구를 감지하면 데몬에서 `ioctl`을 전송하여 페일오버 또는 페일백을 수행합니다. `ioctl`을 구현하는 `ip` 커널 모듈은 투명하고 자동적으로 네트워크 액세스 페일오버를 수행합니다.

주 – 동일한 네트워크 인터페이스 카드 설정에서 IPMP를 사용하는 경우에는 대체 경로를 사용하지 마십시오. 마찬가지로 대체 경로를 사용하는 경우에는 IPMP를 사용하지 마십시오. 대체 경로와 IPMP를 동시에 사용하려면 인터페이스 설정이 달라야 합니다. 대체 경로에 대한 자세한 내용은 **Sun Enterprise Server Alternate Pathing 2.3.1 User Guide**를 참조하십시오.

`in.mpathd` 데몬은 IPMP 그룹에 속한 모든 인터페이스에 프로브를 전송하여 실패 및 복구를 감지합니다. 또한 `in.mpathd` 데몬은 그룹의 각 인터페이스에 있는 `RUNNING` 플래그를 모니터링하여 실패 및 복구를 감지합니다. 자세한 내용은 `in.mpathd(1M)` 매뉴얼 페이지를 참조하십시오.

주 – DHCP는 IPMP 데이터 주소 관리에 사용할 수 없습니다. 이러한 주소에 대해 DHCP를 사용하면 DHCP에서 해당 주소를 제어할 수 없게 됩니다. 데이터 주소에는 DHCP를 사용하지 마십시오.

IPMP 용어 및 개념

이 절에서는 본 설명서의 전체 IPMP 장에서 사용되는 용어와 개념을 소개합니다.

IP 링크

IPMP 용어에서 *IP* **링크**는 인터넷 프로토콜 제품군의 데이터 링크 계층에서 노드가 통신하는 데 사용되는 매체 또는 통신 기능입니다. IP 링크의 유형으로는 단순한 이더넷, 브리지된 이더넷, 허브 또는 ATM(비동기식 전송 모드) 네트워크가 있습니다. IP 링크는 하나 이상의 IPv4 서브넷 번호와 하나 이상의 IPv6 서브넷 접두어(해당하는 경우)를 가질 수 있습니다. 서브넷 번호 또는 접두어를 여러 개의 IP 링크에 지정할 수 없습니다. ATM LANE에서 IP 링크는 단일 에뮬레이트된 LAN(Local Area Network)입니다. ARP(Address Resolution Protocol)에서 ARP 프로토콜의 범위는 단일 IP 링크입니다.

주 – RFC 2460, **Internet Protocol, Version 6 (IPv6) Specification** 등의 다른 IP 관련 설명서를 보려면 *IP* **링크** 대신 **링크**라는 용어를 사용하십시오. VI부에서는 IEEE 802와의 혼동을 피하기 위해 *IP* **링크**라는 용어를 사용합니다. IEEE 802의 **링크**는 이더넷 NIC(네트워크 인터페이스 카드)에서 이더넷 스위치로의 단일 연결을 의미합니다.

물리적 인터페이스

물리적 인터페이스는 시스템을 IP 링크로 연결합니다. 이 연결은 종종 장치 드라이버와 NIC로 구현됩니다. 시스템에 동일한 링크로 연결되는 다중 인터페이스가 있으면 한 인터페이스가 실패할 경우 페일오버를 수행하도록 IPMP를 구성할 수 있습니다. 이 물리적 인터페이스에 대한 자세한 내용은 660 페이지 “IPMP 인터페이스 구성”을 참조하십시오.

네트워크 인터페이스 카드

네트워크 인터페이스 카드는 시스템에 내장할 수 있는 네트워크 어댑터입니다. 또는 NIC는 시스템을 IP 링크로 연결하는 인터페이스 역할을 하는 별도의 카드가 될 수 있습니다. 일부 NIC에는 다중 물리적 인터페이스가 있을 수 있습니다. 예를 들어, `qfe` NIC에는 `qfe0`부터 `qfe3`까지 4개의 인터페이스가 있을 수 있습니다.

IPMP 그룹

IPMP 그룹(IP 다중 경로 그룹)은 동일한 시스템에서 동일한 IPMP 그룹 이름으로 구성된 하나 이상의 물리적 인터페이스로 구성됩니다. IPMP 그룹의 모든 인터페이스는 동일한 IP 링크에 연결되어 있어야 합니다. 널이 아닌 같은 문자열로 된 IPMP 그룹 이름은 그룹의 모든 인터페이스를 식별합니다. 동일한 IPMP 그룹에 다양한 속도의 NIC 인터페이스를 배치할 수 있습니다. 단, 이 경우 NIC는 모두 동일한 유형이어야 합니다. 예를 들어, 100메가비트 이더넷 NIC와 1기가비트 NIC를 동일한 그룹으로 구성할 수 있습니다. 다른 예와 마찬가지로 2개의 100기가비트 이더넷 NIC가 있다고 가정합니다. 인터페이스 중 하나를 10메가비트로 낮추고 여전히 2개의 인터페이스를 동일한 IPMP 그룹에 배치할 수 있습니다.

다른 매체 유형의 두 인터페이스를 한 IPMP 그룹에 배치할 수는 없습니다. 예를 들어, ATM 인터페이스를 이더넷 인터페이스와 동일한 그룹에 배치할 수 없습니다.

실패 감지 및 페일오버

실패 감지는 인터페이스 또는 인터페이스에서 인터넷 계층 장치로의 경로가 더 이상 작동하지 않을 경우 이를 감지하는 프로세스입니다. IPMP는 시스템에 인터페이스 실패를 감지할 수 있는 기능을 제공합니다.

IPMP는 다음 유형의 통신 실패를 감지합니다.

- 인터페이스의 전송 또는 수신 경로 실패
- IP 링크로의 인터페이스 연결 오류
- 패킷을 전송 또는 수신하지 않는 스위치의 포트
- 시스템 부트 시 IPMP 그룹에 물리적 인터페이스가 나타나지 않는 문제

실패 감지 후 IPMP는 페일오버를 시작합니다. **페일오버**는 실패한 인터페이스에서 동일한 그룹 내에서 정상적으로 작동되는 물리적 인터페이스로 네트워크 액세스를 전환하는 자동 프로세스입니다. 네트워크 액세스에는 IPv4 유니캐스트, 멀티캐스트 및 브로드캐스트 트래픽뿐 아니라 IPv6 유니캐스트 및 멀티캐스트 트래픽도 포함됩니다. 페일오버는 IPMP 그룹에 둘 이상의 인터페이스가 구성된 경우에만 발생할 수 있습니다. 페일오버 프로세스를 사용하면 네트워크 액세스 중단을 방지할 수 있습니다.

복구 감지 및 페일백

복구 감지는 NIC 또는 NIC에서 인터넷 계층 장치로의 경로가 실패 후 올바르게 작동을 시작하는지 감지하는 프로세스입니다. NIC 복구를 감지한 후 IPMP는 네트워크 액세스를 복구된 인터페이스로 전환하는 프로세스인 **페일백**을 수행합니다. 복구 감지는 페일백 기능이 사용으로 설정되어 있다는 것을 전제로 합니다. 자세한 내용은 664 페이지 “물리적 인터페이스 복구 감지”를 참조하십시오.

대상 시스템

프로브 기반 실패 감지는 **대상 시스템**을 사용하여 인터페이스의 상태를 판단합니다. 각 대상 시스템은 IPMP 그룹의 구성원과 동일한 IP 링크에 연결되어 있어야 합니다. 로컬 시스템의 `in.mpathd` 데몬은 ICMP 프로브 메시지를 각 대상 시스템으로 전송합니다. 프로브 메시지는 IPMP 그룹의 각 인터페이스 상태를 확인하는 데 사용됩니다.

대상 시스템에서 프로브 기반 실패 감지 사용에 대한 자세한 내용은 663 페이지 “프로브 기반 실패 감지”를 참조하십시오.

아웃바운드 로드 확산

IPMP 구성에서 아웃바운드 네트워크 패킷은 패킷 순서에 영향을 주지 않고 여러 NIC로 확산됩니다. 이 프로세스를 **로드 확산**이라고 합니다. 로드 확산을 사용하면 처리량을 늘릴 수 있습니다. 로드 확산은 네트워크 트래픽이 다중 연결을 사용하는 여러 대상으로 흐르고 있을 때만 발생합니다.

동적 재구성

*DR(***동적 재구성***)*은 기존 작업에 거의 또는 전혀 영향을 주지 않고 시스템이 실행 중인 동안 시스템을 재구성하는 기능입니다. 모든 Sun 플랫폼에서 DR을 지원하는 것은 아닙니다. 일부 Sun 플랫폼은 특정 하드웨어 유형의 DR만 지원합니다. NIC의 DR을 지원하는 플랫폼에서 IPMP를 사용하여 시스템에 대한 네트워크 액세스가 중단되지 않도록 투명하게 페일오버할 수 있습니다.

IPMP가 DR을 지원하는 방식에 대한 자세한 내용은 666 페이지 “IPMP 및 동적 재구성”을 참조하십시오.

IPMP의 기본 요구 사항

IPMP는 Oracle Solaris에서 기본 제공되므로 별도의 하드웨어가 필요하지 않습니다. Oracle Solaris에서 지원하는 모든 인터페이스를 IPMP와 함께 사용할 수 있습니다. 그러나 IPMP는 네트워크 구성 및 토폴로지에 대해 다음과 같은 요구 사항이 있습니다.

- IPMP 그룹의 모든 인터페이스에 고유 MAC 주소가 있어야 합니다.

  기본적으로 SPARC 기반 시스템에 있는 네트워크 인터페이스는 모두 하나의 MAC 주소를 공유합니다. 따라서 SPARC 기반 시스템에서 IPMP를 사용하려면 기본값을 명시적으로 변경해야 합니다. 자세한 내용은 671 페이지 “IPMP 그룹을 계획하는 방법”을 참조하십시오.
- IPMP 그룹에 있는 모든 인터페이스의 매체 유형은 동일해야 합니다. 자세한 내용은 655 페이지 “IPMP 그룹”을 참조하십시오.
- IPMP 그룹에 있는 모든 인터페이스는 동일한 IP 링크에 있어야 합니다. 자세한 내용은 655 페이지 “IPMP 그룹”을 참조하십시오.

  주 – 동일한 링크 계층(L2 또는 계층 2) 브로드캐스트 도메인의 여러 IPMP 그룹은 지원되지 않습니다. L2 브로드캐스트 도메인은 일반적으로 특정 서브넷에 매핑됩니다. 따라서 서브넷당 IPMP 그룹 한 개만 구성해야 합니다.

- 실패 감지 요구 사항에 따라 특정 유형의 네트워크 인터페이스를 사용하거나 각 네트워크 인터페이스에 추가 IP 주소를 구성해야 할 수 있습니다. 662 페이지 “링크 기반 실패 감지” 및 663 페이지 “프로브 기반 실패 감지”를 참조하십시오.

IPMP 주소 지정

IPv4 네트워크와 이중 스택 IPv4 및 IPv6 네트워크에서 모두 IPMP 실패 감지를 구성할 수 있습니다. IPMP를 사용하여 구성된 인터페이스는 두 가지 유형의 주소(데이터 주소와 테스트 주소)를 지원합니다.

데이터 주소

데이터 주소는 부트 시 또는 `ifconfig` 명령을 통해 수동으로 NIC 인터페이스에 지정하는 일반적인 IPv4 및 IPv6 주소입니다. 인터페이스를 통과하는 표준 IPv4 패킷 트래픽과 IPv6 패킷 트래픽(해당하는 경우)은 **데이터 트래픽**으로 간주됩니다.

테스트 주소

테스트 주소는 `in.mpathd` 데몬에서 사용하는 IPMP 특정 주소입니다. 프로브 기반 실패 및 복구 감지를 사용하려는 인터페이스의 경우 해당 인터페이스는 하나 이상의 테스트 주소로 구성되어야 합니다.

주 – 프로브 기반 실패 감지를 사용하려는 경우에만 테스트 주소를 구성해야 합니다.

`in.mpathd` 데몬은 테스트 주소를 사용하여 IP 링크에 있는 다른 대상과 ICMP 프로브를 교환하는데 이를 **프로브 트래픽**이라고 합니다. 프로브 트래픽을 사용하면 인터페이스의 실패 여부를 비롯한 인터페이스와 해당 NIC의 상태를 확인할 수 있습니다. 프로브는 인터페이스로의 전송 및 수신 경로가 제대로 작동하는지 확인합니다.

각 인터페이스는 IP 테스트 주소로 구성할 수 있습니다. 이중 스택 네트워크에 있는 인터페이스의 경우 IPv4 테스트 주소, IPv6 테스트 주소 또는 둘 다를 구성할 수 있습니다.

한 인터페이스가 실패하면 `in.mpathd`에서 계속해서 프로브를 보내 후속 복구를 확인할 수 있도록 테스트 주소가 실패한 인터페이스에 그대로 남게 됩니다. 응용 프로그램에서 실수로 사용하지 못하도록 테스트 주소를 특별하게 구성해야 합니다. 자세한 내용은 659 페이지 “응용 프로그램의 테스트 주소 사용 방지”를 참조하십시오.

프로브 기반 실패 감지에 대한 자세한 내용은 663 페이지 “프로브 기반 실패 감지”를 참조하십시오.

IPv4 테스트 주소

일반적으로 서브넷에 있는 모든 IPv4 주소를 테스트 주소로 사용할 수 있습니다. IPv4 테스트 주소는 경로 지정 가능하지 않아도 됩니다. IPv4 주소는 대부분의 사이트에서 제한된 리소스이므로 경로 지정 불가능한 RFC 1918 개인 주소를 테스트 주소로 사용하는 것이 좋습니다. `in.mpathd` 데몬은 테스트 주소와 동일한 서브넷에 있는 다른

호스트하고만 ICMP 프로브를 교환합니다. RFC 1918 스타일 테스트 주소를 사용하는 경우 해당 RFC 1918 서브넷의 주소를 가진 IP 링크에서 다른 시스템(특히 라우터)을 구성해야 합니다. 그러면 `in.mpathd` 데몬이 대상 시스템과 프로브를 성공적으로 교환할 수 있습니다.

IPMP 예에서는 `192.168.0/24` 네트워크의 RFC 1918 주소를 IPv4 테스트 주소로 사용합니다. RFC 1918 개인 주소에 대한 자세한 내용은 RFC 1918, Address Allocation for Private Internets. (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`)을 참조하십시오.

IPv4 테스트 주소를 구성하려면 673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”을 참조하십시오.

IPv6 테스트 주소

유효한 IPv6 테스트 주소는 물리적 인터페이스의 링크 로컬 주소뿐입니다. IPMP 테스트 주소로 사용할 별도의 IPv6 주소가 필요 없습니다. IPv6 링크 로컬 주소는 인터페이스의 MAC(Media Access Control) 주소를 기반으로 합니다. 인터페이스가 부트 시 IPv6 사용으로 설정되거나 `ifconfig`를 통해 인터페이스를 수동으로 구성하면 링크 로컬 주소가 자동으로 구성됩니다.

인터페이스의 링크 로컬 주소를 확인하려면 IPv6 사용 노드에서 `ifconfig` *interface* 명령을 실행합니다. 링크 로컬 접두어인 `fe80`으로 시작하는 주소의 출력을 확인합니다. 다음 `ifconfig` 출력의 `NOFAILOVER` 플래그는 `hme0` 인터페이스의 링크 로컬 주소 `fe80::a00:20ff:feb9:17fa/10`이 테스트 주소로 사용된다는 것을 나타냅니다.

```
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
          inet6 fe80::a00:20ff:feb9:17fa/10
```

링크 로컬 주소에 대한 자세한 내용은 74 페이지 “링크 로컬 유니캐스트 주소”를 참조하십시오.

IPMP 그룹의 모든 인터페이스에 IPv4와 IPv6이 모두 연결되어 있는 경우 별도의 IPv4 테스트 주소를 구성할 필요가 없습니다. `in.mpathd` 데몬은 IPv6 링크 로컬 주소를 테스트 주소로 사용할 수 있습니다.

IPv6 테스트 주소를 만들려면 673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”을 참조하십시오.

응용 프로그램의 테스트 주소 사용 방지

테스트 주소를 구성한 후 이 주소가 응용 프로그램에서 사용되지 않도록 설정해야 합니다. 그렇지 않으면 인터페이스가 실패할 경우 페일오버 작업 중 테스트 주소가 페일오버되지 않아 응용 프로그램에 연결할 수 없게 됩니다. IP에서 일반 응용 프로그램의 테스트 주소를 선택하지 않도록 하려면 테스트 주소를 `deprecated`로 표시합니다.

응용 프로그램에서 주소로 명시적으로 바인딩하지 않는 한, IPv4는 어떠한 통신에서도 제거된 주소를 소스 주소로 사용하지 않습니다. `in.mpathd` 데몬은 프로브 트래픽을 송수신하기 위해 이러한 주소를 명시적으로 바인딩합니다. 그러나 응용 프로그램에서 주소로 명시적으로 바인딩하지 않고 인터페이스에서 `UP`으로 표시된 주소만 deprecated로 표시된 경우 최후의 수단으로 해당 주소는 소스 주소로 사용됩니다.

주 – 페일오버 및 페일백에서 중복 주소 감지가 실행 중인 경우 응용 프로그램에서 제거된 주소를 소스 주소로 사용하여 패킷을 수신할 수 있습니다. 이 동작은 예상 가능합니다. 일반적으로 DAD가 완료되면 제거된 주소는 더 이상 응용 프로그램에서 처리되지 않습니다. 그러나 간혹 TCP 패킷의 경우에는 예외 사항이 발생하기도 합니다. TCP 연결에서 특정 소스 주소를 선택한 후 해당 연결 동안 주소 사용을 변경할 수 없습니다. 이 시간은 매우 길어질 수 있습니다. 이러한 경우에는 DAD 완료 후에도 응용 프로그램에서 제거된 주소를 사용할 가능성이 있습니다.

IPv6 링크 로컬 주소는 주로 이름 서비스에 나타나지 않으므로 DNS 및 NIS 응용 프로그램에서는 통신에 링크 로컬 주소를 사용하지 않습니다. 따라서 IPv6 링크 로컬 주소를 `deprecated`로 표시해서는 안됩니다.

IPv4 테스트 주소를 DNS 및 NIS 이름 서비스 테이블에 배치해서는 안됩니다. IPv6에서 링크 로컬 주소는 일반적으로 이름 서비스 테이블에 배치되지 않습니다.

IPMP 인터페이스 구성

IPMP 구성은 일반적으로 동일한 시스템에서 동일한 IP 링크에 연결된 둘 이상의 물리적 인터페이스로 구성됩니다. 이러한 물리적 인터페이스는 동일한 NIC에 있을 수도 있고 그렇지 않을 수도 있습니다. 인터페이스는 동일한 IPMP 그룹의 구성원으로 구성됩니다. 시스템의 두번째 IP 링크에 추가 인터페이스가 있으면 이러한 인터페이스를 다른 IPMP 그룹으로 구성해야 합니다.

단일 인터페이스를 고유한 IPMP 그룹으로 구성할 수 있습니다. 단일 인터페이스 IPMP 그룹은 여러 인터페이스가 포함된 IPMP 그룹과 동일한 방식으로 동작합니다. 그러나 인터페이스가 하나뿐인 IPMP 그룹에서는 페일오버 및 페일백이 발생하지 않습니다.

또한 동일한 단계를 사용하여 IP 인터페이스의 그룹을 구성하여 VLAN을 IPMP 그룹으로 구성할 수 있습니다. 절차는 672 페이지 “IPMP 그룹 구성”을 참조하십시오. 657 페이지 “IPMP의 기본 요구 사항”에 나열된 동일한 요구 사항은 VLAN을 IPMP 그룹으로 구성하는 데 적용됩니다.

주의 – VLAN의 이름 지정에 사용되는 규칙은 VLAN을 IPMP 그룹으로 구성할 때 오류를 발생시킬 수 있습니다. VLAN 이름에 대한 자세한 내용은 **System Administration Guide: IP Services**의 145 페이지 “VLAN 태그 및 물리적 연결 지점”를 참조하십시오. 예를 들어, 4개의 VLAN인 `bge1000`, `bge1001`, `bge2000`, `bge2001`이 있다고 가정합니다. IPMP 구현을 위해서는 이러한 VLAN을 그룹화해야 합니다. 즉, `bge1000` 및 `bge1001`을 동일한 VLAN 1의 한 그룹으로 `bge2000` 및 `bge2001`을 동일한 VLAN 2의 한 그룹으로 그룹화합니다. VLAN 이름 때문에 `bge1000` 및 `bge2000`과 같이 다른 링크에 속한 VLAN을 하나의 IPMP 그룹으로 혼합할 때 오류가 쉽게 발생할 수 있습니다.

IPMP 그룹의 대기 인터페이스

IPMP 그룹의 **대기 인터페이스**는 그룹의 다른 인터페이스가 실패하지 않는 한 데이터 트래픽에 사용되지 않습니다. 실패가 발생하면 실패한 인터페이스의 데이터 주소가 대기 인터페이스로 마이그레이션됩니다. 그러면 실패한 인터페이스가 복구될 때까지 대기 인터페이스가 다른 활성 인터페이스와 동일하게 처리됩니다. 일부 페일오버 작업에서는 대기 인터페이스를 선택하지 않을 수 있습니다. 대신 이러한 페일오버 작업에서는 UP로 구성된 데이터 주소가 대기 인터페이스보다 더 적은 활성 인터페이스를 선택합니다.

대기 인터페이스에서는 테스트 주소만 구성해야 합니다. IPMP를 사용할 경우 `ifconfig` 명령을 통해 `standby`로 구성된 인터페이스에 데이터 주소를 추가할 수 없습니다. 이러한 유형으로 구성하려고 하면 실패합니다. 마찬가지로 이미 데이터 주소가 있는 인터페이스를 `standby`로 구성하면 이러한 주소는 자동으로 IPMP 그룹의 다른 인터페이스로 페일오버됩니다. 이러한 제한으로 인해 인터페이스를 `standby`로 설정하려면 먼저 `ifconfig` 명령을 사용하여 테스트 주소를 `deprecated` 및 `-failover`로 표시해야 합니다. 대기 인터페이스를 구성하려면 679 페이지 “IPMP 그룹의 대기 인터페이스를 구성하는 방법”을 참조하십시오.

공통 IPMP 인터페이스 구성

658 페이지 “IPMP 주소 지정”에서 설명했듯이 IPMP 그룹의 인터페이스는 인터페이스 구성에 따라 정규 데이터 트래픽 및 프로브 트래픽을 처리합니다. `ifconfig` 명령의 IPMP 옵션을 사용하여 구성을 만듭니다.

활성 인터페이스는 데이터 트래픽 및 프로브 트래픽을 전송하는 물리적 인터페이스입니다. 673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법” 또는 681 페이지 “단일 인터페이스 IPMP 그룹을 구성하는 방법”을 수행하여 인터페이스를 “활성”으로 구성합니다.

IPMP 구성의 공통 유형은 다음과 같습니다.

활성-활성 구성 모두 “활성” 상태인 두 개의 인터페이스로 구성된 IPMP 그룹이며 프로브 및 데이터 트래픽을 언제든지 전송할 수 있습니다.

활성-대기 구성 두 개의 인터페이스로 구성된 IPMP 그룹이며 한 인터페이스는 "대기"로 구성됩니다.

인터페이스 상태 확인

`ifconfig` *interface* 명령을 실행하여 인터페이스의 현재 상태를 확인할 수 있습니다. `ifconfig` 상태 보고에 대한 일반적인 내용은 191 페이지 "특정 인터페이스에 대한 정보를 얻는 방법"을 참조하십시오.

예를 들어, `ifconfig` 명령을 사용하여 대기 인터페이스의 상태를 확인할 수 있습니다. 대기 인터페이스가 데이터 주소를 호스팅하지 않는 경우 인터페이스의 상태에 `INACTIVE` 플래그가 설정됩니다. 이 플래그는 `ifconfig` 출력에서 인터페이스 상태 라인에 나타납니다.

IPMP 실패 감지 및 복구 기능

`in.mpathd` 데몬은 다음 유형의 실패 감지를 처리합니다.

- 링크 기반 실패 감지. NIC 드라이버에서 지원되는 경우에 사용합니다.
- 프로브 기반 실패 감지(테스트 주소가 구성된 경우)
- 부트 시 누락된 인터페이스 감지

`in.mpathd` 데몬이 인터페이스 실패 감지를 처리하는 방법에 대한 자세한 내용은 `in.mpathd(1M)` 매뉴얼 페이지에서 자세히 설명합니다.

링크 기반 실패 감지

인터페이스에서 이러한 유형의 실패 감지를 지원하는 경우 링크 기반 실패 감지는 항상 사용으로 설정됩니다. 현재 Oracle Solaris 릴리스에서는 다음 Sun 네트워크 드라이버를 지원합니다.

- `hme`
- `eri`
- `ce`
- `ge`
- `bge`
- `qfe`
- `dmfe`
- `e1000g`
- `igb`
- `ixgb`
- `nge`
- `nxge`

- rge
- xge

타사 인터페이스가 링크 기반 실패 감지를 지원하는지 확인하려면 해당 제조업체의 설명서를 참조하십시오.

이러한 네트워크 인터페이스 드라이버는 인터페이스의 링크 상태를 모니터링하여 링크 상태가 변경될 경우 네트워킹 부속 시스템에 이를 알립니다. 변경 알림을 받으면 네트워킹 부속 시스템이 해당 인터페이스에 대해 RUNNING 플래그를 적절하게 설정하거나 지웁니다. 데몬이 인터페이스의 RUNNING 플래그가 지워진 것을 감지하면 데몬이 즉시 인터페이스 실패를 발생시킵니다.

프로브 기반 실패 감지

in.mpathd 데몬은 테스트 주소가 있는 IPMP 그룹의 각 인터페이스에 대해 프로브 기반 실패 감지를 수행합니다. 프로브 기반 실패 감지는 테스트 주소를 사용하는 ICMP 프로브 메시지를 송수신합니다. 이러한 메시지는 인터페이스를 통해 동일한 IP 링크에 있는 하나 이상의 대상 시스템으로 전달됩니다. 테스트 주소에 대한 소개는 658 페이지 “테스트 주소”를 참조하십시오. 테스트 주소 구성 방법에 대한 자세한 내용은 673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”을 참조하십시오.

in.mpathd 데몬은 동적으로 프로빙할 대상 시스템을 결정합니다. IP 링크에 연결된 라우터는 프로브 대상으로 자동 선택됩니다. 링크에 라우터가 없으면 in.mpathd가 링크의 인접 호스트로 프로브를 전송합니다. 모든 호스트 멀티캐스트 주소인 224.0.0.1(IPv4) 및 ff02::1(IPv6)로 멀티캐스트 패킷이 전송되어 대상 시스템으로 사용할 호스트를 결정합니다. 에코 패킷에 응답하는 처음 몇 개 호스트가 프로브 대상으로 선택됩니다. in.mpathd가 ICMP 에코 패킷에 응답하는 라우터나 호스트를 찾을 수 없는 경우 in.mpathd는 프로브 기반 실패를 감지할 수 없습니다.

호스트 경로를 사용하여 in.mpathd에서 사용할 대상 시스템 목록을 명시적으로 구성할 수 있습니다. 자세한 내용은 677 페이지 “대상 시스템 구성”을 참조하십시오.

IPMP 그룹의 각 인터페이스가 제대로 작동하는지 확인하기 위해 in.mpathd는 IPMP 그룹의 모든 인터페이스를 통해 모든 대상을 별도로 프로빙합니다. 5회 연속 프로브에 대해 응답이 없을 경우 in.mpathd는 해당 인터페이스가 실패했다고 간주합니다. 프로브 속도는 FDT(**실패 감지 시간**)에 따라 달라집니다. 실패 감지 시간의 기본값은 10초입니다. 그러나 /etc/default/mpathd 파일에서 실패 감지 시간을 조정할 수 있습니다. 자세한 내용은 689 페이지 “/etc/default/mpathd 파일을 구성하는 방법”을 참조하십시오.

복구 감지 시간이 10초인 경우 프로브 속도는 약 2초마다 프로브 1회입니다. 10회 연속 프로브에 대한 응답이 수신되어야 하므로 최소 복구 감지 시간은 실패 감지 시간의 2배(기본값: 20초)입니다. 실패 및 복구 감지 시간은 프로브 기반 실패 감지에만 적용됩니다.

주 – VLAN으로 구성된 IPMP 그룹에서 링크 기반 실패 감지는 물리적 링크 단위로 구현되므로 해당 링크의 모든 VLAN에 영향을 줍니다. 프로브 기반 실패 감지는 VLAN 링크 단위로 수행됩니다. 예를 들어, `bge0/bge1` 및 `bge1000/bge1001`은 한 그룹으로 구성됩니다. `bge0`의 케이블 연결이 끊어지면 링크 기반 실패 감지에서 `bge0` 및 `bge1000`이 모두 즉시 실패한 것으로 보고합니다. 그러나 `bge0`의 모든 프로브 대상을 연결할 수 없게 된 경우 `bge1000`은 해당 VLAN에 자체 프로브 대상이 있으므로 `bge0`만 실패한 것으로 보고됩니다.

그룹 실패

그룹 실패는 IPMP 그룹의 모든 인터페이스가 동시에 실패한 것으로 나타날 때 발생합니다. `in.mpathd` 데몬은 그룹 실패에 대해 페일오버를 수행하지 않습니다. 또한 모든 대상 시스템이 동시에 실패할 경우 어떠한 페일오버도 발생하지 않습니다. 이 경우 `in.mpathd`는 현재 대상 시스템을 모두 비우고 새 대상 시스템을 찾습니다.

물리적 인터페이스 복구 감지

`in.mpathd` 데몬에서 인터페이스 복구를 감지하려면 해당 인터페이스에 `RUNNING` 플래그가 설정되어 있어야 합니다. 프로브 기반 실패 감지가 사용된 경우 `in.mpathd` 데몬에서 인터페이스로부터 10회 연속 프로브 패킷에 대한 응답을 받아야 해당 인터페이스가 복구된 것으로 간주합니다. 인터페이스가 복구된 것으로 간주되면 다른 인터페이스로 페일오버되었던 모든 주소가 복구된 인터페이스로 페일백됩니다. 인터페이스가 실패하기 전 "활성"으로 구성된 경우에는 복구 후 트래픽 전송 및 수신을 계속할 수 있습니다.

인터페이스 페일오버 중 발생하는 작업

다음 두 예에서는 일반 구성 및 인터페이스가 실패할 경우 해당 구성이 자동으로 변경되는 방법을 보여 줍니다. `hme0` 인터페이스가 실패하면 모든 데이터 주소가 `hme0`에서 `hme1`로 이동합니다.

예 27–1 인터페이스 실패 전 인터페이스 구성

```
hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
     mtu 1500 index 2
     inet 192.168.85.19 netmask ffffff00 broadcast 192.168.85.255
     groupname test
hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
     mtu 1500
     index 2 inet 192.168.85.21 netmask ffffff00 broadcast 192.168.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
8     inet 192.168.85.20 netmask ffffff00 broadcast 192.168.85.255
```

예 27–1 인터페이스 실패 전 인터페이스 구성 *(계속)*

```
     groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
     mtu 1500
     index 2 inet 192.168.85.22 netmask ffffff00 broadcast 192.168.85.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
     inet6 fe80::a00:20ff:feb9:19fa/10
     groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
     inet6 fe80::a00:20ff:feb9:1bfc/10
     groupname test
```

예 27–2 인터페이스 실패 후 인터페이스 구성

```
hme0: flags=19000842<BROADCAST,RUNNING,MULTICAST,IPv4,
      NOFAILOVER,FAILED> mtu 0 index 2
      inet 0.0.0.0 netmask 0
      groupname test
hme0:1: flags=19040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
      NOFAILOVER,FAILED> mtu 1500 index 2
      inet 192.168.85.21 netmask ffffff00 broadcast 10.0.0.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 192.168.85.20 netmask ffffff00 broadcast 192.168.85.255
      groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
      NOFAILOVER> mtu 1500
      index 2 inet 192.168.85.22 netmask ffffff00 broadcast 10.0.0.255
hme1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 6
      inet 192.168.85.19 netmask ffffff00 broadcast 192.168.18.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER,FAILED> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:19fa/10
      groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:1bfc/10
      groupname test
```

hme0에 FAILED 플래그가 설정되어 인터페이스가 실패했음을 알 수 있습니다. 또한 hme1:2가 만들어진 것을 알 수 있습니다. hme1:2는 원래 hme0이었습니다. 그런 다음 주소 192.168.85.19는 hme1을 통해 액세스할 수 있게 됩니다.

192.168.85.19와 연결된 멀티캐스트 구성원은 여전히 패킷을 수신할 수는 있지만 이제 hme1을 통해 패킷을 수신합니다. 주소 192.168.85.19가 hme0에서 hme1로 페일오버되면 hme0에 가상 주소 0.0.0.0이 만들어집니다. 가상 주소는 hme0을 계속 액세스하기 위해 만들어진 것입니다. hme0:1은 hme0 없이 존재할 수 없습니다. 후속 페일백이 수행되면 가상 주소가 제거됩니다.

마찬가지로 IPv6 주소가 hme0에서 hme1로 페일오버됩니다. IPv6에서 멀티캐스트 구성원은 인터페이스 색인과 연결되어 있습니다. 멀티캐스트 구성원도 hme0에서 hme1로 페일오버됩니다. in.ndpd가 구성한 모든 주소도 함께 이동됩니다. 이 작업은 예에 표시되어 있지 않습니다.

in.mpathd 데몬은 계속해서 실패한 인터페이스 hme0을 통해 프로브를 수행합니다. 데몬이 기본 복구 감지 시간인 20초 동안 10회 연속 응답을 받게 되면 해당 인터페이스가 복구된 것으로 판단합니다. hme0에는 RUNNING 플래그도 설정되어 있기 때문에 데몬은 페일백을 호출합니다. 페일백 후 원래 구성이 복원됩니다.

실패 및 복구 중 콘솔에 기록되는 모든 오류 메시지에 대한 설명은 in.mpathd(1M) 매뉴얼 페이지를 참조하십시오.

IPMP 및 동적 재구성

DR(동적 재구성) 기능을 사용으로 설정하면 시스템이 실행되는 동안 인터페이스 같은 시스템 하드웨어를 재구성할 수 있습니다. 이 절에서는 DR이 IPMP와 상호 운용되는 방법을 설명합니다.

NIC의 DR을 지원하는 시스템에서는 IPMP를 사용하여 연결을 유지하고 기존 연결의 중단을 방지할 수 있습니다. DR을 지원하고 IPMP를 사용하는 시스템에서는 안전하게 NIC를 연결, 분리 또는 재연결할 수 있습니다. 이는 IPMP가 RCM(Reconfiguration Coordination Manager) 프레임워크에 통합되어 있기 때문입니다. *RCM*은 시스템 구성 요소의 동적 재구성을 관리합니다.

일반적으로 cfgadm 명령을 사용하여 DR 작업을 수행합니다. 하지만 일부 플랫폼은 다른 방법을 제공합니다. 자세한 내용은 플랫폼의 설명서를 참조하십시오. DR에 대한 설명서는 다음 리소스에서 찾을 수 있습니다.

표 27–1 동적 재구성에 대한 설명서 리소스

설명	정보
cfgadm 명령에 대한 자세한 정보	cfgadm(1M) 매뉴얼 페이지
Sun Cluster 환경의 DR에 대한 특정 정보	**Sun Cluster 3.1 System Administration Guide**
Sun Fire 환경의 DR에 대한 특정 정보	**Sun Fire 880 Dynamic Reconfiguration Guide**
DR 및 cfgadm 명령에 대한 소개 정보	**System Administration Guide: Devices and File Systems**의 4 장, “Dynamically Configuring Devices (Tasks)”
DR을 지원하는 시스템에서 IPMP 그룹을 관리하는 작업	685 페이지 “동적 재구성을 지원하는 시스템에서 실패한 물리적 인터페이스 바꾸기”

NIC 연결

673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”에 설명된 대로 언제든지 ifconfig 명령을 사용하여 IPMP 그룹에 인터페이스를 추가할 수 있습니다.

따라서 시스템 부트 후 연결한 시스템 구성 요소의 인터페이스는 기존 IPMP 그룹에 연결 및 추가할 수 있습니다. 또는 해당하는 경우 새로 추가한 인터페이스를 고유한 IPMP 그룹으로 구성할 수 있습니다.

이러한 인터페이스와 여기에 구성된 데이터 주소는 IPMP 그룹에서 즉시 사용할 수 있습니다. 그러나 재부트 시 시스템에서 자동으로 인터페이스를 구성 및 사용하게 하려면 각 새 인터페이스에 대해 `/etc/hostname.`*interface* 파일을 만들어야 합니다. 자세한 내용은 138 페이지 “시스템 설치 후 물리적 인터페이스 구성 방법”을 참조하십시오.

인터페이스를 연결할 때 `/etc/hostname.`*interface* 파일이 이미 있으면 RCM이 자동으로 이 파일의 내용에 따라 인터페이스를 구성합니다. 그러면 인터페이스는 시스템 부트 후 수신한 것과 동일한 구성을 수신하게 됩니다.

NIC 분리

NIC가 포함된 시스템 구성 요소를 분리하는 모든 요청이 먼저 검사되어 연결을 유지할 수 있는지 확인합니다. 예를 들어, IPMP 그룹에 없는 NIC는 기본적으로 분리할 수 없습니다. IPMP 그룹에서 작동하는 유일한 인터페이스를 포함하는 NIC도 분리할 수 없습니다. 하지만 시스템 구성 요소를 제거해야 하는 경우 `cfgadm`(1M) 매뉴얼 페이지에 설명된 대로 `cfgadm`의 `-f` 옵션을 사용하여 이 동작을 대체할 수 있습니다.

확인이 성공하면 분리 중인 NIC가 실패한 것처럼 분리된 NIC와 연결된 데이터 주소가 동일한 그룹에서 제대로 작동하는 NIC로 페일오버됩니다. NIC가 분리되면 NIC의 인터페이스에 있는 모든 테스트 주소가 구성 해제됩니다. 그런 다음 NIC가 시스템에서 연결 취소됩니다. 이러한 단계 중 하나라도 실패하거나 동일한 시스템 구성 요소에 있는 다른 하드웨어의 DR이 실패하면 이전 구성이 원래 상태로 복원됩니다. 이 이벤트와 관련된 상태 메시지가 수신됩니다. 그렇지 않으면 분리 요청이 성공적으로 완료됩니다. 시스템에서 구성 요소를 제거할 수 있습니다. 기존 연결은 중단되지 않습니다.

NIC 재연결

RCM은 실행 중인 시스템에서 분리된 NIC와 연결된 구성 정보를 기록합니다. 그 결과 RCM은 이전에 분리된 NIC의 재연결을 새 NIC의 연결과 동일하게 처리합니다. 즉, RCM은 연결만 수행합니다.

그러나 재연결된 NIC에는 일반적으로 기존 `/etc/hostname.`*interface* 파일이 있습니다. 이 경우 RCM은 기존 `/etc/hostname.`*interface* 파일의 내용에 따라 인터페이스를 자동으로 구성합니다. 또한 RCM은 재연결된 인터페이스에서 원래 호스팅되었던 각 데이터 주소를 `in.mpathd` 데몬에 알립니다. 따라서 재연결된 인터페이스가 제대로 작동하게 되면 인터페이스가 복구된 것처럼 해당 데이터 주소가 모두 재연결된 인터페이스로 페일백됩니다.

재연결 중인 NIC에 /etc/hostname.*interface* 파일이 없으면 구성 정보를 사용할 수 없게 됩니다. RCM에는 인터페이스 구성 방법에 대한 정보가 없습니다. 따라서 이전에 다른 인터페이스로 페일오버되었던 주소가 페일백되지 않을 수 있습니다.

시스템 부트 시 누락된 NIC

시스템 부트 시 나타나지 않는 NIC는 실패 감지 특수 인터페이스를 나타냅니다. 부트 시 시작 스크립트는 연결할 수 없는 /etc/hostname.*interface* 파일이 있는 인터페이스를 추적합니다. 이러한 인터페이스의 /etc/hostname. *interface* 파일에 있는 모든 데이터 주소는 IPMP 그룹에 있는 대체 인터페이스에서 자동으로 호스팅됩니다.

이러한 경우 다음과 같은 오류 메시지가 나타납니다.

```
moving addresses from failed IPv4 interfaces: hme0 (moved to hme1)
moving addresses from failed IPv6 interfaces: hme0 (moved to hme1)
```

대체 인터페이스가 없는 경우 다음과 같은 오류 메시지가 나타납니다.

```
moving addresses from failed IPv4 interfaces: hme0 (couldn't move;
   no alternative interface)
 moving addresses from failed IPv6 interfaces: hme0 (couldn't move;
   no alternative interface)
```

주 – 이러한 실패 감지 상황의 경우 누락된 인터페이스의 /etc/hostname. *interface* 파일에서 명시적으로 지정한 데이터 주소만 대체 인터페이스로 이동합니다. 일반적으로 RARP 또는 DHCP 등 다른 방법으로 얻는 주소는 얻거나 이동되지 않습니다.

시스템 부트 시 누락된 다른 인터페이스와 같은 이름의 인터페이스가 DR을 통해 재연결되면 RCM이 자동으로 인터페이스를 연결합니다. 그런 다음 RCM은 인터페이스의 /etc/hostname.*interface* 파일 내용에 따라 인터페이스를 구성합니다. 결국 RCM은 인터페이스가 복구된 것처럼 모든 데이터 주소를 페일백합니다. 따라서 최종 네트워크는 시스템이 인터페이스와 함께 부트된 경우와 동일한 구성을 갖게 됩니다.

◆ ◆ ◆ 28 장

28 IPMP 관리(작업)

이 장에서는 IPMP(IP Network Multipathing)를 사용하여 인터페이스 그룹을 관리하는 작업을 제공합니다. 다음 주요 내용으로 구성되어 있습니다.

IPMP 개념에 대한 개요는 27 장, “IPMP 소개(개요)”를 참조하십시오.

IPMP 구성(작업 맵)

이 절에서는 이 장에 설명된 작업에 대한 링크를 설명합니다.

IPMP 그룹 구성 및 관리(작업 맵)

작업	설명	수행 방법
IPMP 그룹을 계획합니다.	IPMP 그룹을 구성하기 전에 필요한 작업과 보조 정보를 모두 나열합니다.	671 페이지 “IPMP 그룹을 계획하는 방법”
다중 인터페이스가 있는 IPMP 인터페이스 그룹을 구성합니다.	다중 인터페이스를 IPMP 그룹의 구성원으로 구성합니다.	673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”

작업	설명	수행 방법
인터페이스 중 하나가 대기 인터페이스인 IPMP 그룹을 구성합니다.	다중 인터페이스 IPMP 그룹의 인터페이스 중 하나를 대기 인터페이스로 구성합니다.	679 페이지 “IPMP 그룹의 대기 인터페이스를 구성하는 방법”
단일 인터페이스로 구성된 IPMP 그룹을 구성합니다.	단일 인터페이스 IPMP 그룹을 만듭니다.	681 페이지 “단일 인터페이스 IPMP 그룹을 구성하는 방법”
물리적 인터페이스가 속한 IPMP 그룹을 표시합니다.	`ifconfig` 명령 출력에서 인터페이스의 IPMP 그룹 이름을 가져오는 방법을 설명합니다.	682 페이지 “인터페이스의 IPMP 그룹 구성원을 표시하는 방법”
IPMP 그룹에 인터페이스를 추가합니다.	기존 IPMP 그룹의 구성원으로 새 인터페이스를 구성합니다.	683 페이지 “IPMP 그룹에 인터페이스를 추가하는 방법”
IPMP 그룹에서 인터페이스를 제거합니다.	IPMP 그룹에서 인터페이스를 제거하는 방법을 설명합니다.	683 페이지 “IPMP 그룹에서 인터페이스를 제거하는 방법”
기존 IPMP 그룹의 인터페이스를 다른 그룹으로 이동합니다.	IPMP 그룹 간에 인터페이스를 이동합니다.	684 페이지 “한 IPMP 그룹에서 다른 그룹으로 인터페이스를 이동하는 방법”
`in.mpathd` 데몬의 3가지 기본 설정을 변경합니다.	`in.mpathd` 데몬의 실패 감지 시간 및 다른 매개변수를 사용자 정의합니다.	689 페이지 “`/etc/default/mpathd` 파일을 구성하는 방법”

동적 재구성을 지원하는 인터페이스에서 IPMP 관리(작업 맵)

작업	설명	수행 방법
실패한 인터페이스를 제거합니다.	시스템에서 실패한 인터페이스를 제거합니다.	685 페이지 “실패한 물리적 인터페이스를 제거하는 방법(DR 분리)”
실패한 인터페이스를 바꿉니다.	실패한 인터페이스를 바꿉니다.	686 페이지 “실패한 물리적 인터페이스를 바꾸는 방법(DR 연결)”
부트 시 구성되지 않은 인터페이스를 복구합니다.	실패한 인터페이스를 복구합니다.	687 페이지 “시스템 부트 시 표시되지 않는 물리적 인터페이스를 복구하는 방법”

고가용성을 위해 IPMP 그룹 사용

이 절에서는 IPMP 그룹 구성을 위한 절차를 제공합니다. 또한 대기 인터페이스를 구성하는 방법도 설명합니다.

IPMP 그룹 계획

시스템의 인터페이스를 IPMP 그룹의 일부로 구성하려면 먼저 일부 사전 구성 계획을 세워야 합니다.

▼ IPMP 그룹을 계획하는 방법

다음 절차에는 IPMP 그룹을 구성하기 전에 필요한 계획 작업 및 수집할 정보가 포함되어 있습니다. 작업을 순서대로 수행할 필요는 없습니다.

1 시스템에서 IPMP 그룹에 포함할 인터페이스를 결정합니다.

IPMP 그룹은 일반적으로 동일한 IP 링크에 연결된 2개 이상의 물리적 인터페이스로 구성됩니다. 그러나 필요한 경우 단일 인터페이스 IPMP 그룹을 구성할 수 있습니다. IPMP 그룹에 대한 소개는 660 페이지 “IPMP 인터페이스 구성”을 참조하십시오. 예를 들어, 동일한 IPMP 그룹에 동일한 이더넷 스위치 또는 동일한 IP 서브넷을 구성할 수 있습니다. 모든 개수의 인터페이스를 동일한 IPMP 그룹으로 구성할 수 있습니다.

논리적 인터페이스에는 `ifconfig` 명령의 `group` 매개변수를 사용할 수 없습니다. 예를 들어, `hme0`에 `group` 매개변수는 사용할 수 있지만 `hme0:1`에는 사용할 수 없습니다.

2 그룹의 각 인터페이스에 고유한 MAC 주소가 있는지 확인합니다.

절차는 141 페이지 “SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법”을 참조하십시오.

3 IPMP 그룹의 이름을 선택합니다.

그룹에는 널이 아닌 모든 이름을 사용할 수 있습니다. 인터페이스가 연결될 IP 링크를 식별하는 이름을 사용할 수 있습니다.

4 IPMP 그룹의 모든 인터페이스에서 동일한 STREAMS 모듈 세트가 푸시되고 구성되었는지 확인합니다.

동일한 그룹의 모든 인터페이스에 동일한 STREAMS 모듈이 동일한 순서로 구성되어 있어야 합니다.

a. 잠재 IPMP 그룹의 모든 인터페이스에서 STREAMS 모듈의 순서를 확인합니다.

`ifconfig` *interface* `modlist` 명령을 사용하여 STREAMS 모듈 목록을 인쇄할 수 있습니다. 예를 들어, `hme0` 인터페이스의 `ifconfig` 출력은 다음과 같습니다.

```
# ifconfig hme0 modlist
    0 arp
```

```
1 ip
2 hme
```

`ifconfig hme0 modlist`의 출력과 같이 인터페이스는 대체로 IP 모듈 바로 아래에 네트워크 드라이버로 존재합니다. 이러한 인터페이스에는 추가 구성이 필요하지 않습니다.

하지만 NCA 또는 IP 필터와 같은 특정 기술은 IP 모듈과 네트워크 드라이버 간에 STREAMS 모듈로 삽입됩니다. 그 결과 동일한 IPMP 그룹의 인터페이스가 동작하는 방식에 문제가 발생할 수 있습니다.

STREAMS 모듈이 Stateful인 경우 그룹의 모든 인터페이스에 동일한 모듈을 푸시해도 페일오버 시 예기치 않은 동작이 발생할 수 있습니다. 하지만 IPMP 그룹의 모든 인터페이스에 모듈을 동일한 순서로 푸시하는 경우 Stateless STREAMS 모듈을 사용할 수 있습니다.

b. IPMP 그룹에 대한 표준 순서로 인터페이스의 모듈을 푸시합니다.

`ifconfig` *interface* **`modinsert`** *module-name*

```
ifconfig hme0 modinsert ip
```

5 IPMP 그룹의 모든 인터페이스에서 동일한 IP 주소 형식을 사용합니다.

IPv4에 대해 한 인터페이스가 구성된 경우 그룹의 모든 인터페이스를 IPv4에 대해 구성해야 합니다. 여러 NIC의 인터페이스로 구성된 IPMP 그룹이 있다고 가정합니다. 한 NIC의 인터페이스에 IPv6 주소 지정을 추가하는 경우 IPMP 그룹의 모든 인터페이스에서 IPv6 지원을 구성해야 합니다.

6 IPMP 그룹의 모든 인터페이스가 동일한 IP 링크에 연결되어 있는지 확인합니다.

7 IPMP 그룹에 서로 다른 네트워크 매체 유형의 인터페이스가 포함되어 있지 않은지 확인합니다.

그룹화되는 인터페이스는 `/usr/include/net/if_types.h`에 정의된 대로 동일한 인터페이스 유형이어야 합니다. 예를 들어, 이더넷 및 토큰 링 인터페이스를 IPMP 그룹에 결합할 수 없습니다. 또 다른 예로 토큰 버스 인터페이스와 ATM(비동기식 전송 모드) 인터페이스를 동일한 IPMP 그룹에 결합할 수 없습니다.

8 ATM 인터페이스가 있는 IPMP의 경우 LAN 에뮬레이션 모드로 ATM 인터페이스를 구성합니다.

Classical IP over ATM을 사용하는 인터페이스에서는 IPMP가 지원되지 않습니다.

IPMP 그룹 구성

이 절에서는 둘 이상의 물리적 인터페이스가 있는 일반적인 IPMP 그룹에 대한 구성 작업을 설명합니다.

- 다중 인터페이스로 구성된 IPMP 그룹에 대한 소개는 655 페이지 “IPMP 그룹”을 참조하십시오.
- 계획 작업에 대해서는 671 페이지 “IPMP 그룹 계획”을 참조하십시오.
- 물리적 인터페이스가 하나뿐인 IPMP 그룹을 구성하려면 681 페이지 “단일 물리적 인터페이스가 있는 IPMP 그룹 구성”을 참조하십시오.

▼ 다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법

IPMP 그룹을 구성하기 위한 다음 단계는 VLAN을 IPMP 그룹으로 구성할 때도 적용됩니다.

시작하기 전에 IPv4 주소가 이미 구성되어 있어야 하며, 해당하는 경우 잠재 IPMP 그룹에 있는 모든 인터페이스의 IPv6 주소도 구성되어 있어야 합니다.

주의 – 각 서브넷 또는 L2 브로드캐스트 도메인에 대해 IPMP 그룹 한 개만 구성해야 합니다. 자세한 내용은 657 페이지 “IPMP의 기본 요구 사항”을 참조하십시오.

1 구성할 인터페이스가 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 각 물리적 인터페이스를 IPMP 그룹에 배치합니다.

```
# ifconfig interface group group-name
```

예를 들어, `hme0` 및 `hme1`을 그룹 `testgroup1`에 배치하려면 다음 명령을 입력합니다.

```
# ifconfig hme0 group testgroup1
# ifconfig hme1 group testgroup1
```

그룹 이름에 공백을 사용하지 마십시오. `ifconfig` 상태 표시에는 공백이 나타나지 않습니다. 따라서 공백만 다른 비슷한 그룹 이름을 두 개 만들지 마십시오. 그룹 이름 중 하나에 공백이 있어도 두 그룹 이름은 상태 표시에서 똑같이 나타납니다.

이중 스택 환경에서 인터페이스의 IPv4 인스턴스를 특정 그룹 아래에 배치하면 IPv6 인스턴스도 동일한 그룹 아래에 자동으로 배치됩니다.

3 (선택 사항) 하나 이상의 물리적 인터페이스에서 IPv4 테스트 주소를 구성합니다.

특정 인터페이스에서 프로브 기반 실패 감지를 사용하려는 경우에만 테스트 주소를 구성해야 합니다. 테스트 주소는 `ifconfig` 명령에 지정한 물리적 인터페이스의 논리적 인터페이스로 구성됩니다.

그룹의 한 인터페이스를 대기 인터페이스로 만들려면 해당 인터페이스의 테스트 주소는 지금 구성하지 마십시오. 대기 인터페이스의 테스트 주소는 679 페이지 “IPMP 그룹의 대기 인터페이스를 구성하는 방법”의 작업 중에 구성합니다.

테스트 주소를 구성하려면 ifconfig 명령의 다음 구문을 사용합니다.

```
# ifconfig interface addif ip-address parameters -failover deprecated up
```

예를 들어, 주 네트워크 인터페이스 hme0에 대해 다음 테스트 주소를 만듭니다.

```
# ifconfig hme0 addif 192.168.85.21 netmask + broadcast + -failover deprecated up
```

이 명령은 주 네트워크 인터페이스 hme0에 대해 다음 매개변수를 설정합니다.

- 주소를 192.168.85.21로 설정
- 넷마스크 및 브로드캐스트 주소를 기본값으로 설정
- -failover 및 deprecated 옵션 설정

주 – 응용 프로그램에서 테스트 주소를 사용하지 않도록 하기 위해 IPv4 테스트 주소를 deprecated로 표시해야 합니다.

---

4 특정 인터페이스의 IPv4 구성을 확인합니다.

언제든지 ifconfig *interface*를 입력하여 인터페이스의 현재 상태를 볼 수 있습니다. 인터페이스의 상태 보기에 대한 자세한 내용은 191 페이지 “특정 인터페이스에 대한 정보를 얻는 방법”을 참조하십시오.

테스트 주소에 지정된 논리적 인터페이스를 지정하여 물리적 인터페이스의 테스트 주소 구성에 대한 정보를 얻을 수 있습니다.

```
# ifconfig hme0:1
    hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
    mtu 1500 index 2
    inet 192.168.85.21 netmask ffffff00 broadcast 192.168.85.255
```

5 (선택 사항) 해당하는 경우 IPv6 테스트 주소를 구성합니다.

```
# ifconfig interface inet6 -failover
```

IPv6 주소가 있는 물리적 인터페이스는 인터페이스의 IPv4 주소와 동일한 IPMP 그룹에 배치됩니다. 이는 IPv4 주소가 있는 물리적 인터페이스를 IPMP 그룹으로 구성할 때 수행됩니다. 먼저 IPv6 주소가 있는 물리적 인터페이스를 IPMP 그룹으로 배치한 경우 IPv4 주소가 있는 물리적 인터페이스도 동일한 IPMP 그룹에 암시적으로 배치됩니다.

예를 들어, IPv6 테스트 주소가 있는 hme0을 구성하려면 다음을 입력합니다.

```
# ifconfig hme0 inet6 -failover
```

응용 프로그램에서 테스트 주소를 사용하지 않도록 하기 위해 IPv6 테스트 주소를 deprecated로 표시할 필요가 없습니다.

6 IPv6 구성을 확인합니다.

```
# ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:17fa/10
            groupname test
```

IPv6 테스트 주소는 인터페이스의 링크 로컬 주소입니다.

7 (선택 사항) 재부트 시 IPMP 그룹 구성을 유지합니다.

- IPv4의 경우 다음 라인을 `/etc/hostname.` *interface* 파일에 추가합니다.

  ```
  interface-address <parameters> group group-name up \
      addif logical-interface -failover deprecated <parameters> up
  ```

  이 인스턴스의 테스트 IPv4 주소는 다음 재부트 시에만 구성됩니다. 구성을 현재 세션에서 호출하려면 단계 1과 2(선택적으로 3)를 수행합니다.
- IPv6의 경우 다음 라인을 `/etc/hostname6.` *interface* 파일에 추가합니다.

  ```
  -failover group group-name up
  ```

  이 테스트 IPv6 주소는 다음 재부트 시에만 구성됩니다. 구성을 현재 세션에서 호출하려면 단계 1과 2(선택적으로 5)를 수행합니다.

8 (선택 사항) 단계 1-6을 반복하여 다른 인터페이스를 IPMP 그룹에 추가합니다.

라이브 시스템의 기존 그룹에 새 인터페이스를 추가할 수 있습니다. 그러나 이 변경 사항은 재부트 시 손실됩니다.

예 28–1 두 개의 인터페이스가 있는 IPMP 그룹 구성

다음 작업을 수행한다고 가정합니다.

- 넷마스크 및 브로드캐스트 주소를 기본값으로 설정합니다.
- 인터페이스를 테스트 주소 `192.168.85.21`로 구성합니다.

이 경우 다음 명령을 입력합니다.

```
# ifconfig hme0 addif 192.168.85.21 netmask + broadcast + -failover deprecated up
```

응용 프로그램에서 테스트 주소를 사용하지 않도록 하기 위해 IPv4 테스트 주소를 `deprecated`로 표시해야 합니다. 673 페이지 “다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법”을 참조하십시오.

주소의 페일오버 속성을 설정하려면 대시 없이 `failover` 옵션을 사용합니다.

IPMP 그룹의 모든 테스트 IP 주소는 동일한 네트워크 접두어를 사용해야 합니다. 테스트 IP 주소는 단일 IP 서브넷에 속해야 합니다.

예 28–2 재부트 시 IPv4 IPMP 그룹 구성 유지

다음과 같은 구성의 IPMP 그룹(`testgroup1`)을 만든다고 가정합니다.

- 데이터 주소가 `192.168.85.19`인 물리적 인터페이스 `hme0`
- 테스트 주소가 `192.168.85.21`인 논리적 인터페이스

주 – 이 예에서 물리적 인터페이스와 데이터 주소는 함께 쌍을 이룹니다. 논리적 인터페이스와 테스트 주소도 마찬가지로 쌍을 이룹니다. 그러나 인터페이스 "유형"과 주소 유형 간에 본질적인 관계가 존재하는 것은 아닙니다.

- deprecated 및 -failover 옵션 설정
- 넷마스크 및 브로드캐스트 주소를 기본값으로 설정

/etc/hostname.hme0 파일에 다음 라인을 추가합니다.

```
192.168.85.19 netmask + broadcast + group testgroup1 up \
    addif 192.168.85.21 deprecated -failover netmask + broadcast + up
```

마찬가지로 동일한 그룹 testgroup1에 두번째 인터페이스 hme1을 배치하고 테스트 주소를 구성하려면 다음 라인을 추가합니다.

```
192.168.85.20 netmask + broadcast + group testgroup1 up \
    addif 192.168.85.22 deprecated -failover netmask + broadcast + up
```

예 28–3 재부트 시 IPv6 IPMP 그룹 구성 유지

IPv6 주소가 있는 인터페이스 hme0의 테스트 그룹을 만들려면 /etc/hostname6.hme0 파일에 다음 라인을 추가합니다.

```
-failover group testgroup1 up
```

마찬가지로 그룹 testgroup1에 두번째 인터페이스 hme1을 배치하고 테스트 주소를 구성하려면 /etc/hostname6.hme1 파일에 다음 라인을 추가합니다.

```
-failover group testgroup1 up
```

일반 오류 IPMP 그룹 구성 중 in.mpathd 명령을 실행하면 시스템 콘솔이나 syslog 파일에 여러 개의 메시지가 출력됩니다. 이러한 메시지는 원래 정보 전달을 위한 것이며 IPMP 구성이 정상적으로 작동됨을 나타냅니다.

- 이 메시지는 인터페이스 hme0이 IPMP 그룹 testgroup1에 추가되었음을 나타냅니다. 그러나 hme0에는 테스트 주소가 구성되어 있지 않습니다. 프로브 기반 실패 감지를 사용으로 설정하려면 테스트 주소를 인터페이스에 지정해야 합니다.

  ```
  May 24 14:09:57 host1 in.mpathd[101180]:
  No test address configured on interface hme0;
  disabling probe-based failure detection on it.
  testgroup1
  ```

- 이 메시지는 IPMP 그룹에 추가된 IPv4 주소만 있는 모든 인터페이스에 나타납니다.

  ```
  May 24 14:10:42 host4 in.mpathd[101180]:
  NIC qfe0 of group testgroup1 is not
  plumbed for IPv6 and may affect failover capability
  ```

- 이 메시지는 인터페이스의 테스트 주소를 구성할 때 표시되어야 합니다.

```
Created new logical interface hme0:1
May 24 14:16:53 host1 in.mpathd[101180]:
Test address now configured on interface hme0;
 enabling probe-based failure detection on it
```

참조 IPMP 그룹을 활성-대기로 구성하려면 679 페이지 "IPMP 그룹의 대기 인터페이스를 구성하는 방법"을 참조하십시오.

대상 시스템 구성

프로브 기반 실패 감지의 경우 663 페이지 "프로브 기반 실패 감지"에 설명된 대로 대상 시스템을 사용해야 합니다. 일부 IPMP 그룹의 경우 `in.mpathd`에 사용되는 기본 대상이면 충분합니다. 하지만 프로브 기반 실패 감지에 대해 특정 대상을 구성해야 하는 IPMP 그룹도 있습니다. 경로 지정 테이블의 호스트 경로를 프로브 대상으로 설정하여 프로브 기반 실패 감지를 수행합니다. 경로 지정 테이블에 구성된 호스트 경로는 기본 라우터 앞에 나열됩니다. 따라서 IPMP는 명시적으로 정의된 호스트 경로를 대상 선택으로 사용합니다. 두 가지 방법을 사용하여 대상을 직접 지정할 수 있습니다. 호스트 경로를 수동으로 설정하거나 시작 스크립트로 사용할 셸 스크립트를 만듭니다.

적합한 대상이 될 수 있는 네트워크의 호스트를 평가하는 경우 다음 기준을 고려해 보십시오.

- 잠재 대상이 사용 가능하고 실행되고 있는지 확인합니다. 해당 IP 주소 목록을 만듭니다.
- 대상 인터페이스가 구성 중인 IPMP 그룹과 동일한 네트워크에 있는지 확인합니다.
- 대상 시스템의 넷마스크 및 브로드캐스트 주소가 IPMP 그룹의 주소와 같아야 합니다.
- 대상 호스트가 프로브 기반 실패 감지를 사용하는 인터페이스의 ICMP 요청에 대답할 수 있어야 합니다.

▼ 프로브 기반 실패 감지의 대상 시스템을 수동으로 지정하는 방법

1 프로브 기반 실패 감지를 구성 중인 시스템에 사용자 계정으로 로그인합니다.

2 프로브 기반 실패 감지에서 대상으로 사용할 특정 호스트에 경로를 추가합니다.

```
$ route add -host destination-IP gateway-IP -static
```

destination-IP 및 *gateway-IP* 값을 대상으로 사용할 호스트의 IPv4 주소로 바꿉니다. 예를 들어, IPMP 그룹 `testgroup1`의 인터페이스와 동일한 서브넷에 있는 대상 시스템 `192.168.85.137`을 지정하려면 다음을 입력합니다.

```
$ route add -host 192.168.85.137 192.168.85.137 -static
```

3 네트워크에서 대상 시스템으로 사용할 추가 호스트에 경로를 추가합니다.

▼ 셸 스크립트에 대상 시스템을 지정하는 방법

1 **IPMP 그룹을 구성한 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **정적 경로를 제안된 대상으로 설정하는 셸 스크립트를 만듭니다.**

예를 들어, 다음 내용이 있는 셸 스크립트(`ipmp.targets`)를 만들 수 있습니다.

```
TARGETS="192.168.85.117 192.168.85.127 192.168.85.137"

case "$1" in
        'start')
            /usr/bin/echo "Adding static routes for use as IPMP targets"
        for target in $TARGETS; do
      /usr/sbin/route add -host $target $target
        done
                ;;
        'stop')
              /usr/bin/echo "Removing static routes for use as IPMP targets"
         for target in $TARGETS; do
        /usr/sbin/route delete -host $target $target
         done
                ;;
  esac
```

3 **셸 스크립트를 시작 스크립트 디렉토리에 복사합니다.**

```
# cp ipmp.targets /etc/init.d
```

4 **새 시작 스크립트의 권한을 변경합니다.**

```
# chmod 744 /etc/init.d/ipmp.targets
```

5 **새 시작 스크립트의 소유권을 변경합니다.**

```
# chown root:sys /etc/init.d/ipmp.targets
```

6 **`/etc/init.d` 디렉토리에서 시작 스크립트의 링크를 만듭니다.**

```
# ln /etc/init.d/ipmp.targets /etc/rc2.d/S70ipmp.targets
```

파일 이름 `S70ipmp.targets`의 접두어 S70은 다른 시작 스크립트를 기준으로 새 시작 스크립트의 순서를 지정합니다.

대기 인터페이스 구성

IPMP 그룹을 활성-대기로 구성하려면 이 절차를 사용합니다. 이 구성 유형에 대한 자세한 내용은 660 페이지 “IPMP 인터페이스 구성”을 참조하십시오.

▼ IPMP 그룹의 대기 인터페이스를 구성하는 방법

시작하기 전에
- 모든 인터페이스가 IPMP 그룹의 구성원으로 구성되어 있어야 합니다.
- 인터페이스의 테스트 주소가 대기 인터페이스로 구성되어 있으면 안됩니다.

IPMP 그룹 구성 및 테스트 주소 지정에 대한 자세한 내용은 673 페이지 "다중 인터페이스가 포함된 IPMP 그룹을 구성하는 방법"을 참조하십시오.

1 구성할 대기 인터페이스가 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 인터페이스를 대기로 구성하고 테스트 주소를 지정합니다.

```
# ifconfig interface plumb \
ip-address other-parameters deprecated -failover standby up
```

대기 인터페이스는 하나의 IP 주소(테스트 주소)만 가질 수 있습니다. `standby up` 옵션을 설정하기 전에 `-failover` 옵션을 설정해야 합니다. `<other-parameters>`의 경우 `ifconfig(1M)` 매뉴얼 페이지에 설명된 대로 구성에 필요한 매개변수를 사용합니다.

- 예를 들어, IPv4 테스트 주소를 만들려면 다음 명령을 입력합니다.

```
# ifconfig hme1 plumb 192.168.85.22 netmask + broadcast + deprecated -failover standby up
```

`hme1`	`hme1`을 대기 인터페이스로 구성할 물리적 인터페이스로 정의합니다.
`192.168.85.22`	이 테스트 주소를 대기 인터페이스에 지정합니다.
`deprecated`	테스트 주소가 아웃바운드 패킷에 사용되지 않음을 나타냅니다.
`-failover`	인터페이스가 실패할 경우 테스트 주소가 페일오버되지 않음을 나타냅니다.
`standby`	인터페이스를 대기 인터페이스로 표시합니다.

- 예를 들어, IPv6 테스트 주소를 만들려면 다음 명령을 입력합니다.

```
# ifconfig hme1 plumb -failover standby up
```

3 대기 인터페이스 구성의 결과를 확인합니다.

```
# ifconfig hme1
hme1: flags=69040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,
      STANDBY,INACTIVE mtu 1500
         index 4 inet 192.168.85.22 netmask ffffff00 broadcast 19.16.85.255
         groupname test
```

`INACTIVE` 플래그는 이 인터페이스가 아웃바운드 패킷에 사용되지 않음을 나타냅니다. 이 대기 인터페이스에서 페일오버가 발생하면 `INACTIVE` 플래그가 지워집니다.

주 – 언제든지 `ifconfig` *interface* 명령을 입력하여 인터페이스의 현재 상태를 볼 수 있습니다. 인터페이스의 상태를 보는 방법에 대한 자세한 내용은 191 페이지 "특정 인터페이스에 대한 정보를 얻는 방법"을 참조하십시오.

4 (선택 사항) 재부트 시 IPv4 대기 인터페이스를 유지합니다.

대기 인터페이스를 동일한 IPMP 그룹에 지정하고 대기 인터페이스의 테스트 주소를 구성합니다.

예를 들어, `hme1`을 대기 인터페이스로 구성하려면 `/etc/hostname.hme1` 파일에 다음 라인을 추가합니다.

```
192.168.85.22 netmask + broadcast + deprecated group test -failover standby up
```

5 (선택 사항) 재부트 시 IPv6 대기 인터페이스를 유지합니다.

대기 인터페이스를 동일한 IPMP 그룹에 지정하고 대기 인터페이스의 테스트 주소를 구성합니다.

예를 들어, `hme1`을 대기 인터페이스로 구성하려면 `/etc/hostname6.hme1` 파일에 다음 라인을 추가합니다.

```
-failover group test standby up
```

예 28–4 IPMP 그룹의 대기 인터페이스 구성

다음과 같이 구성된 테스트 주소를 만든다고 가정합니다.

- 물리적 인터페이스 `hme2`를 대기 인터페이스로 설정
- 테스트 주소 `192.168.85.22`
- `deprecated` 및 `-failover` 옵션 설정
- 넷마스크 및 브로드캐스트 주소를 기본값으로 설정

다음을 입력합니다.

```
# ifconfig hme2 plumb 192.168.85.22 netmask + broadcast + \
deprecated -failover standby up
```

인터페이스는 주소를 `NOFAILOVER` 주소로 표시한 후에만 대기 인터페이스로 표시됩니다.

다음을 입력하여 인터페이스의 대기 상태를 제거합니다.

```
# ifconfig interface -standby
```

단일 물리적 인터페이스가 있는 IPMP 그룹 구성

IPMP 그룹의 인터페이스가 하나인 경우에는 페일오버를 수행할 수 없습니다. 그러나 인터페이스를 IPMP 그룹에 지정하여 해당 인터페이스에서 실패 감지 기능을 사용으로 설정할 수 있습니다. 전용 테스트 IP 주소를 구성하여 단일 인터페이스 IPMP 그룹의 실패 감지를 설정할 필요가 없습니다. 데이터를 전송하고 실패를 감지할 때는 단일 IP 주소를 사용할 수 있습니다.

▼ 단일 인터페이스 IPMP 그룹을 구성하는 방법

1 잠재 단일 인터페이스 IPMP 그룹이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 IPv4의 경우 단일 인터페이스 IPMP 그룹을 만듭니다.

다음 구문을 사용하여 IPMP 그룹에 단일 인터페이스를 지정합니다.

 # ifconfig *interface* **group** *group-name*

다음 예에서는 인터페이스 `hme0`을 IPMP 그룹 `v4test`에 지정합니다.

```
# ifconfig hme0 group v4test
```

이 단계를 수행한 다음 IPMP에서 인터페이스에 대해 링크 기반 실패 감지를 사용으로 설정합니다.

또한 `ifconfig` 명령의 `-failover` 하위 명령을 사용하여 프로브 기반 실패 감지를 사용으로 설정할 수 있습니다. 다음 예에서는 현재 `hme0`에 지정된 IP 주소를 사용하여 `hme0`에서 프로브 기반 실패 감지를 사용으로 설정합니다.

```
# ifconfig hme0 -failover
```

다중 인터페이스 그룹과 달리 동일한 IP 주소가 데이터 주소와 테스트 주소로 모두 사용될 수 있습니다. 응용 프로그램에서 테스트 주소를 데이터 주소로 사용으로 설정하려면 단일 인터페이스 IPMP 그룹에서 테스트 주소를 `deprecated`로 표시해서는 안됩니다.

3 IPv6의 경우 단일 인터페이스 IPMP 그룹을 만듭니다.

다음 구문을 사용하여 IPMP 그룹에 단일 인터페이스를 지정합니다.

 # ifconfig *interface* **inet6 group** *group-name*

예를 들어, 단일 인터페이스 `hme0`을 IPMP 그룹 `v6test`에 추가하려면 다음을 입력합니다.

```
# ifconfig hme0 inet6 group v6test
```

이 단계를 수행한 다음 IPMP에서 인터페이스에 대해 링크 기반 실패 감지를 사용으로 설정합니다.

또한 `ifconfig` 명령의 `-failover` 하위 명령을 사용하여 프로브 기반 실패 감지를 사용으로 설정할 수 있습니다. 다음 예에서는 현재 `hme0`에 지정된 IP 주소를 사용하여 `hme0`에서 프로브 기반 실패 감지를 사용으로 설정합니다.

```
# ifconfig hme0 inet6 -failover
```

다중 인터페이스 그룹과 달리 동일한 IP 주소가 데이터 주소와 테스트 주소로 모두 사용될 수 있습니다. 응용 프로그램에서 테스트 주소를 데이터 주소로 사용으로 설정하려면 단일 인터페이스 IPMP 그룹에서 테스트 주소를 `deprecated`로 표시해서는 안됩니다.

단일 물리적 인터페이스 구성에서는 프로빙 중인 대상 시스템의 실패인지 아니면 인터페이스의 실패인지 여부를 확인할 수 없습니다. 대상 시스템은 하나의 물리적 인터페이스를 통해서만 프로빙할 수 있습니다. 서브넷에 기본 라우터가 하나뿐인데 단일 물리적 인터페이스가 그룹에 있는 경우 IPMP를 해제합니다. 별도의 IPv4 및 IPv6 기본 라우터가 있거나 여러 기본 라우터가 있는 경우 둘 이상의 대상 시스템을 프로빙해야 합니다. 따라서 IPMP를 안전하게 설정할 수 있습니다.

IPMP 그룹 유지 관리

이 절에서는 기존 IPMP 그룹과 이러한 그룹을 구성하는 인터페이스를 유지 관리하는 작업을 설명합니다. 이 작업에서는 671 페이지 “고가용성을 위해 IPMP 그룹 사용”에 설명된 대로 IPMP 그룹을 이미 구성한 것으로 가정합니다.

▼ 인터페이스의 IPMP 그룹 구성원을 표시하는 방법

1. **IPMP 그룹 구성이 있는 시스템에서 수퍼 유저 또는 동등한 역할의 사용자로 로그인합니다.**

   역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. 역할에 대한 자세한 내용은 **System Administration Guide: Security Services**의 “Configuring RBAC (Task Map)”를 참조하십시오.

2. **인터페이스가 속할 그룹을 포함하여 인터페이스에 대한 정보를 표시합니다.**

   ```
   # ifconfig interface
   ```

3. **해당하는 경우 인터페이스의 IPv6 정보를 표시합니다.**

   ```
   # ifconfig interface inet6
   ```

예 28–5 물리적 인터페이스 그룹 표시

hme0의 그룹 이름을 표시하려면 다음을 입력합니다.

```
# ifconfig hme0
    hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
      index 2 inet 192.168.85.19 netmask ffffff00 broadcast 192.168.85.255
      groupname testgroup1
```

IPv6 정보의 그룹 이름만 표시하려면 다음을 입력합니다.

```
# ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
            inet6 fe80::a00:20ff:feb9:19fa/10
            groupname testgroup1
```

▼ IPMP 그룹에 인터페이스를 추가하는 방법

1 **IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **IPMP 그룹에 인터페이스를 추가합니다.**

```
# ifconfig interface group group-name
```

*interface*에 지정된 인터페이스가 IPMP 그룹 *group-name*의 구성원이 됩니다.

예 28–6 IPMP 그룹에 인터페이스 추가

IPMP 그룹 testgroup2에 hme0을 추가하려면 다음 명령을 입력합니다.

```
# ifconfig hme0 group testgroup2
  hme0: flags=9000843<UP ,BROADCAST,RUNNING,MULTICAST,IPv4,NOFAILOVER> mtu 1500 index 2
  inet 192.168.85.19 netmask ff000000 broadcast 10.255.255.255
  groupname testgroup2
  ether 8:0:20:c1:8b:c3
```

▼ IPMP 그룹에서 인터페이스를 제거하는 방법

ifconfig 명령의 group 매개변수를 널 문자열과 함께 실행하면 인터페이스가 현재 IPMP 그룹에서 제거됩니다. 그룹에서 인터페이스를 제거할 때는 주의하십시오. IPMP 그룹에서 다른 인터페이스가 실패한 경우 페일오버는 더 일찍 발생해야 합니다. 예를 들어, hme0이 이전에 실패했고 hme1이 동일한 그룹의 구성원인 경우 모든 주소가 hme1로 페일오버됩니다. 그룹에서 hme1을 제거하면 in.mpathd 데몬이 모든 페일오버 주소를

그룹의 다른 인터페이스로 반환합니다. 그룹에서 제대로 작동하는 다른 인터페이스가 없으면 페일오버가 모든 네트워크 액세스를 복원하지 못할 수 있습니다.

마찬가지로 그룹에 있는 한 인터페이스를 연결 해제할 때는 먼저 그룹에서 해당 인터페이스를 제거해야 합니다. 그런 다음 인터페이스에 원래의 IP 주소가 모두 구성되어 있는지 확인합니다. `in.mpathd` 데몬은 그룹에서 제거된 인터페이스의 원래 구성을 복원하려고 합니다. 인터페이스를 연결 해제하기 전에 구성이 복원되었는지 확인해야 합니다. 페일오버 전후의 인터페이스 상태는 664 페이지 “인터페이스 페일오버 중 발생하는 작업”을 참조하십시오.

1 **IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **IPMP 그룹에서 인터페이스를 제거합니다.**

```
# ifconfig interface group ""
```

인용 부호는 널 문자열을 의미합니다.

예 28–7 그룹에서 인터페이스 제거

IPMP 그룹 `test`에서 `hme0`을 제거하려면 다음 명령을 입력합니다.

```
# ifconfig hme0 group ""
    # ifconfig hme0
    hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
    index 2 inet 192.168.85.19 netmask ffffff00 broadcast 192.168.85.255
    # ifconfig hme0 inet6
    hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
    inet6 fe80::a00:20ff:feb9:19fa/10
```

▼ 한 IPMP 그룹에서 다른 그룹으로 인터페이스를 이동하는 방법

인터페이스가 기존 IPMP 그룹에 속하는 경우 새 IPMP 그룹에 인터페이스를 배치할 수 있습니다. 현재 IPMP 그룹에서 인터페이스를 제거할 필요는 없습니다. 새 그룹에 인터페이스를 배치하면 기존 IPMP 그룹에서 해당 인터페이스가 자동으로 제거됩니다.

1 **IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 인터페이스를 새 IPMP 그룹으로 이동합니다.

```
# ifconfig interface group group-name
```

새 그룹에 인터페이스를 배치하면 기존 그룹에서 해당 인터페이스가 자동으로 제거됩니다.

예 28–8 다른 IPMP 그룹으로 인터페이스 이동

인터페이스 hme0의 IPMP 그룹을 변경하려면 다음을 입력합니다.

```
# ifconfig hme0 group cs-link
```

이 명령은 IPMP 그룹 test에서 hme0 인터페이스를 제거하고 cs-link 그룹에 인터페이스를 추가합니다.

동적 재구성을 지원하는 시스템에서 실패한 물리적 인터페이스 바꾸기

이 절에는 DR(동적 재구성)을 지원하는 시스템 관리와 관련된 절차가 포함되어 있습니다.

주 – 이 작업은 ifconfig 명령을 사용하여 구성된 IP 계층에만 적용됩니다. 계층이 자동화되지 않은 경우 ATM이나 다른 서비스와 같은 IP 계층 전후의 계층에는 특정 수동 단계가 필요합니다. 다음 절차의 단계는 분리 전 인터페이스의 구성을 해제하고 연결 후 인터페이스를 구성하는 데 사용됩니다.

▼ 실패한 물리적 인터페이스를 제거하는 방법(DR 분리)

이 절차에서는 DR을 지원하는 시스템에서 물리적 인터페이스를 제거하는 방법을 설명합니다. 이 절차는 다음 조건이 존재한다고 가정합니다.

- 예제 인터페이스는 물리적 인터페이스 hme0 및 hme1입니다.
- 두 인터페이스 모두 동일한 IPMP 그룹에 속합니다.
- hme0이 실패했습니다.
- 논리적 인터페이스 hme0:1에는 테스트 주소가 있습니다.
- 실패한 인터페이스를 동일한 물리적 인터페이스 이름으로 바꿉니다(예: hme0을 hme0로).

주 – 테스트 주소가 `/etc/hostname.hme0` 파일을 사용하여 연결된 경우 단계 2를 건너뛸 수 있습니다.

1 **IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **테스트 주소 구성을 표시합니다.**

```
# ifconfig hme0:1

hme0:1:
flags=9040842<BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
inet 192.168.233.250 netmask ffffff00 broadcast 192.168.233.255
```

물리적 인터페이스를 바꿀 때 이 정보를 사용하여 테스트 주소를 다시 연결해야 합니다.

3 **물리적 인터페이스를 제거합니다.**

물리적 인터페이스를 제거하는 방법에 대한 자세한 내용은 다음 자료를 참조하십시오.

- `cfgadm`(1M) 매뉴얼 페이지
- **Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide**
- **Sun Enterprise 10000 DR 구성 설명서**

▼ 실패한 물리적 인터페이스를 바꾸는 방법(DR 연결)

이 절차에서는 DR을 지원하는 시스템에서 물리적 인터페이스를 바꾸는 방법을 설명합니다.

1 **IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **물리적 인터페이스를 바꿉니다.**

자세한 지침은 다음 자료를 참조하십시오.

- `cfgadm`(1M) 매뉴얼 페이지
- **Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide**

- **Sun Enterprise 10000 DR 구성 설명서** 또는 **Sun Fire 880 Dynamic Reconfiguration User's Guide**

시스템 부트 시 표시되지 않는 물리적 인터페이스 복구

주 – 다음 절차는 `ifconfig` 명령을 사용하여 구성된 IP 계층에만 적용됩니다. 계층이 자동화되지 않은 경우 ATM이나 다른 서비스와 같은 IP 계층 전후의 계층에는 특정 수동 단계가 필요합니다. 다음 절차의 특정 단계는 분리 전 인터페이스의 구성을 해제하고 연결 후 인터페이스를 구성하는 데 사용됩니다.

Sun Fire™ 플랫폼에 있는 I/O 보드의 일부인 인터페이스의 경우 동적 재구성 후 복구는 자동으로 수행됩니다. NIC가 Sun Crypto Accelerator I - cPCI 보드인 경우에도 복구는 자동으로 수행됩니다. 따라서 DR 작업의 일부로 복구되는 인터페이스의 경우에는 다음 단계가 필요하지 않습니다. Sun Fire x800 및 Sun Fire 15000 시스템에 대한 자세한 내용은 `cfgadm_sbd(1M)` 매뉴얼 페이지를 참조하십시오. 물리적 인터페이스는 `/etc/hostname.`*interface* 파일에 지정된 구성으로 페일백됩니다. 재부트 시 구성을 유지하도록 인터페이스를 구성하는 방법은 671 페이지 “고가용성을 위해 IPMP 그룹 사용”을 참조하십시오.

주 – Sun Fire 레거시(Exx00) 시스템에서는 DR 분리를 수동 절차로 수행해야 합니다. 그러나 DR 연결은 자동입니다.

▼ 시스템 부트 시 표시되지 않는 물리적 인터페이스를 복구하는 방법

시스템 부트 시 표시되지 않는 물리적 인터페이스를 복구하려면 먼저 다음 절차를 완료해야 합니다. 이 절차의 예는 다음과 같이 구성되어 있습니다.

- 인터페이스는 물리적 인터페이스 `hme0` 및 `hme1`입니다.
- 두 인터페이스 모두 동일한 IPMP 그룹에 속합니다.
- 시스템 부트 시 `hme0`이 설치되지 않았습니다.

주 – 실패한 물리적 인터페이스의 복구 중 IP 주소의 페일백은 최대 3분이 소요됩니다. 이 시간은 네트워크 트래픽에 따라 달라질 수 있습니다. 또한 `in.mpathd` 데몬에서 페일오버 인터페이스를 페일백하기 위한 수신 인터페이스의 안정성에 따라서도 달라집니다.

1 **IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 **콘솔 로그의 실패 오류 메시지에서 실패한 네트워크 정보를 검색합니다.**

syslog(3C) 매뉴얼 페이지를 참조하십시오. 다음과 유사한 오류 메시지가 표시될 수 있습니다.

```
moving addresses from failed IPv4 interfaces:
hme1 (moved to hme0)
```

이 메시지는 실패한 인터페이스 hme1의 IPv4 주소가 hme0 인터페이스로 페일오버되었음을 나타냅니다.

또는 다음과 같은 오류 메시지가 나타날 수 있습니다.

```
moving addresses from failed IPv4 interfaces:
hme1 (couldn’t move, no alternative interface)
```

이 메시지는 실패한 인터페이스 hme1과 동일한 그룹에 활성 인터페이스가 없음을 나타냅니다. 따라서 hme1의 IPv4 주소는 페일오버될 수 없습니다.

3 **물리적 인터페이스를 시스템에 연결합니다.**

물리적 인터페이스를 바꾸는 방법은 다음을 참조하십시오.

- cfgadm(1M) 매뉴얼 페이지
- **Sun Enterprise 10000 DR 구성 설명서**
- **Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide**

4 **단계 2의 메시지 내용을 참조하십시오. 주소를 이동할 수 없는 경우 단계 6으로 이동합니다. 주소가 이동된 경우 단계 5를 진행합니다.**

5 **페일오버 프로세스 중 구성된 논리적 인터페이스를 연결 해제합니다.**

a. **/etc/hostname.** *moved-from-interface* **파일의 내용을 검토하여 페일오버 프로세스 중 구성된 논리적 인터페이스를 확인합니다.**

b. **각 페일오버 IP 주소를 연결 해제합니다.**

```
# ifconfig moved-to-interface removeif moved-ip-address
```

주 – 페일오버 주소는 failover 매개변수로 표시되거나 -failover 매개변수로 표시되어 있지 않습니다. -failover로 표시된 IP 주소는 연결 해제할 필요가 없습니다.

예를 들어, /etc/hostname.hme0 파일의 내용에 다음 라인이 있다고 가정합니다.

```
inet 10.0.0.4 -failover up group one
addif 10.0.0.5 failover up
addif 10.0.0.6 failover up
```

각 페일오버 IP 주소를 연결 해제하려면 다음 명령을 입력합니다.

```
# ifconfig hme0 removeif 10.0.0.5
# ifconfig hme0 removeif 10.0.0.6
```

6 제거된 각 인터페이스에 대해 다음 명령을 입력하여 물리적 인터페이스를 바꾸도록 IPv4 정보를 재구성합니다.

```
# ifconfig removed-from-NIC <parameters>
```

예를 들어, 다음 명령을 입력할 수 있습니다.

```
# ifconfig hme1 inet plumb
# ifconfig hme1 inet 10.0.0.4 -failover up group one
# ifconfig hme1 addif 10.0.0.5 failover up
# ifconfig hme1 addif 10.0.0.6 failover up
```

IPMP 구성 수정

IPMP 구성 파일 /etc/default/mpathd를 사용하여 IPMP 그룹에 대해 다음과 같은 시스템 차원의 매개변수를 구성합니다.

- FAILURE_DETECTION_TIME
- TRACK_INTERFACES_ONLY_WITH_GROUPS
- FAILBACK

▼ /etc/default/mpathd 파일을 구성하는 방법

1 IPMP 그룹 구성이 있는 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 /etc/default/mpathd 파일을 편집합니다.

세 매개변수 중 하나 이상의 기본값을 변경합니다.

a. FAILURE_DETECTION_TIME 매개변수의 새 값을 입력합니다.

FAILURE_DETECTION_TIME=*n*

여기서 *n*은 ICMP 프로브에서 인터페이스 실패가 발생했는지 여부를 감지하는 데 걸리는 시간(초)입니다. 기본값은 10초입니다.

b. FAILBACK 매개변수의 새 값을 입력합니다.

```
FAILBACK=[yes | no]
```

- *yes* - *yes* 값은 IPMP의 기본 페일백 동작입니다. 실패한 인터페이스의 복구가 감지되면 662 페이지 “IPMP 실패 감지 및 복구 기능”에 설명된 대로 네트워크 액세스가 복구된 인터페이스로 페일백됩니다.
- *no* - *no* 값은 데이터 트래픽이 복구된 인터페이스로 돌아가지 않음을 나타냅니다. 실패한 인터페이스가 복구된 것으로 감지되면 *INACTIVE* 플래그가 해당 인터페이스에 설정됩니다. 이 플래그는 인터페이스가 현재 데이터 트래픽에 사용되지 않음을 나타냅니다. 프로브 트래픽에는 계속 인터페이스를 사용할 수 있습니다.

  예를 들어, IPMP 그룹이 두 개의 인터페이스인 ce0과 ce1로 구성되어 있다고 가정합니다. 또한 `/etc/default/mpathd`에 *FAILBACK*=*no* 값이 설정되어 있다고 간주합니다. ce0이 실패하면 IPMP의 예상 동작에 따라 해당 트래픽이 ce1로 페일오버됩니다. 그러나 IPMP에서 ce0이 복구된 것을 감지하면 `/etc/default/mpathd`의 *FAILBACK*=*no* 매개변수로 인해 트래픽이 ce1에서 페일백되지 않습니다. ce0 인터페이스는 해당 `INACTIVE` 상태를 유지하고 ce1 인터페이스가 실패하지 않는 한 트래픽에 사용되지 않습니다. ce1 인터페이스가 실패하면 ce1의 주소가 ce0으로 다시 마이그레이션되고 INACTIVE 플래그는 지워집니다. 이 마이그레이션은 ce0이 그룹에서 유일한 INACTIVE 인터페이스인 경우에만 발생합니다. 그룹에 다른 INACTIVE 인터페이스가 있는 경우 ce0이 아닌 다른 INACTIVE 인터페이스로 주소가 마이그레이션될 수 있습니다.

c. TRACK_INTERFACES_ONLY_WITH_GROUPS 매개변수의 새 값을 입력합니다.

```
TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
```

- *yes* - *yes* 값은 IPMP의 기본 동작입니다. 이 매개변수를 사용하면 IPMP가 IPMP 그룹으로 구성되지 않은 네트워크 인터페이스를 무시합니다.
- *no* - *no* 값은 IPMP 그룹으로 구성되었는지 여부에 관계없이 **모든** 네트워크 인터페이스에 대해 실패 및 복구 감지를 설정합니다. 그러나 IPMP 그룹으로 구성되지 않은 인터페이스에서 실패 또는 복구가 감지되면 페일오버나 페일백이 발생하지 않습니다. 따라서 *no* 값은 실패 보고에만 유용하며 네트워크 가용성을 직접 향상시키지는 않습니다.

3 in.mpathd 데몬을 다시 시작합니다.

```
# pkill -HUP in.mpathd
```

제 6 부

IPQoS(IP Quality of Service)

이 파트에서는 Oracle Solaris의 차별화 서비스 구현인 IPQoS(IP Quality of Service) 관련 작업과 정보를 다룹니다.

◆ ◆ ◆ 29장

29 IPQoS 소개(개요)

IPQoS(IP Quality of Service)를 통해 계산 통계 우선 순위 지정, 제어 및 수집이 가능합니다. IPQoS를 사용하면 네트워크 사용자에게 일관된 레벨의 서비스를 제공할 수 있습니다. 또한 트래픽 관리를 통해 네트워크 정체를 피할 수 있습니다.

다음은 이 장에 포함된 항목 목록입니다.

IPQoS 기본

IPQoS는 IETF(Internet Engineering Task Force)의 Differentiated Services Working Group에서 정의한 Diffserv(차별화 서비스) 아키텍처를 사용으로 설정합니다. Oracle Solaris에서 IPQoS는 TCP/IP 프로토콜 스택의 IP 레벨에서 구현됩니다.

차별화 서비스란?

IPQoS를 사용으로 설정하면 선택한 고객 및 선택한 응용 프로그램에 대해 서로 다른 레벨의 네트워크 서비스를 제공할 수 있습니다. 서로 다른 레벨의 서비스를 총칭하여 **차별화 서비스**라고 합니다. 고객에게 제공하는 차별화 서비스는 회사에서 고객에게 제공하는 서비스 레벨의 구조를 기준으로 할 수 있습니다. 또한 네트워크의 응용 프로그램이나 사용자에 대해 설정된 우선 순위를 기준으로 차별화 서비스를 제공할 수 있습니다.

QoS 제공에는 다음 작업이 포함됩니다.

- 서로 다른 그룹(예: 고객 또는 엔터프라이즈의 부서)에 서비스 레벨 지정
- 특정 그룹이나 응용 프로그램에 제공되는 네트워크 서비스 우선 순위 지정
- 네트워크 병목 영역 및 기타 형태의 정체 발견 및 제거
- 네트워크 성능 모니터링 및 성능 통계 제공
- 네트워크 리소스 간의 대역폭 규제

IPQoS 기능

IPQoS에는 다음과 같은 기능이 있습니다.

- QoS 정책 구성을 위한 `ipqosconf` 명령줄 도구
- 조직의 QoS 정책을 구성하는 필터를 기준으로 작업을 선택하는 분류기
- Diffserv 모델과 호환되어 네트워크 트래픽을 측정하는 측정 모듈
- 패킷의 IP 헤더를 전달 정보로 표시하는 기능을 기반으로 하는 서비스 차별화
- 트래픽 흐름에 대한 통계를 수집하는 흐름 계산 모듈
- UNIX® `kstat` 명령을 통해 트래픽 클래스에 대한 통계 수집
- SPARC® 및 x86 아키텍처 지원
- IPv4 및 IPv6 주소 지원
- IP 보안 아키텍처(IPsec)와 상호 운용성
- VLAN(virtual local area networks)에 대한 802.1D 사용자 우선 순위 표시 지원

QoS(Quality-of-Service) 이론 및 실제에 대한 추가 정보를 얻을 수 있는 위치

차별화 서비스 및 QoS에 대한 정보는 서적 및 온라인 소스에서 찾을 수 있습니다.

QoS 관련 서적

QoS 이론 및 실제에 대한 자세한 내용은 다음 서적을 참조하십시오.

- Ferguson, Paul 및 Geoff Huston. *Quality of Service*. John Wiley & Sons, Inc., 1998.
- Kilkki, Kalevi. *Differentiated Services for the Internet*. Macmillan Technical Publishing, 1999.

QoS에 대한 RFC(Requests for Comments)

IPQoS는 다음 RFC 및 다음 인터넷 초안에 기술된 사양을 준수합니다.

- RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (`http://www.ietf.org/rfc/rfc2474.txt?number=2474`) – 차별화 서비스 지원을 위한 IPv4 및 IPv6 패킷 헤더의 ToS(type of service) 필드 또는 DS 필드에 대한 향상된 기능을 설명합니다.
- RFC 2475, An Architecture for Differentiated Services (`http://www.ietf.org/rfc/rfc2475.txt?number=2475`) – Diffserv 구조의 구성 및 모듈에 대한 자세한 설명을 제공합니다.
- RFC 2597, Assured Forwarding PHB Group (`http://www.ietf.org/rfc/rfc2597.txt?number=2597`) – 홉별 AF(assured forwarding) 동작이 어떻게 작동하는지 설명합니다.
- RFC 2598, An Expedited Forwarding PHB (`http://www.ietf.org/rfc/rfc2598.txt?number=2598`) – 홉별 EF(expedited forwarding) 동작이 어떻게 작동하는지 설명합니다.
- 인터넷 초안, *An Informal Management Model for Diffserv Routers* – 라우터에서 Diffserv 아키텍처 구현을 위한 모델을 제공합니다.

QoS 정보 웹 사이트

IETF의 Differentiated Services Working Group은 Diffserv 인터넷 초안에 대한 링크를 제공하는 웹 사이트(`http://www.ietf.org/html.charters/diffserv-charter.html`)를 유지 관리합니다.

라우터 제조업체(예: Cisco Systems 및 Juniper Networks)는 차별화 서비스가 자사 제품에서 어떻게 구현되는지 설명하는 자사 웹 사이트에서 정보를 제공합니다.

IPQoS 매뉴얼 페이지

IPQoS 설명서에는 다음 매뉴얼 페이지가 포함되어 있습니다.

- `ipqosconf(1M)` - IPQoS 구성 파일 설정을 위한 명령을 설명합니다.
- `ipqos(7ipp)` – Diffserv 아키텍처 모델의 IPQoS 구현을 설명합니다.
- `ipgpc(7ipp)` – Diffserv 분류기의 IPQoS 구현을 설명합니다.
- `tokenmt(7ipp)` – IPQoS `tokenmt` 측정기를 설명합니다.
- `tswtclmt(7ipp)` – IPQoS `tswtclmt` 측정기를 설명합니다.
- `dscpmk(7ipp)` – DSCP 표시기 모듈을 설명합니다.
- `dlcosmk(7ipp)` – IPQoS 802.1D 사용자 우선 순위 표시기 모듈을 설명합니다.
- `flowacct(7ipp)`– IPQoS 흐름 계산 모듈을 설명합니다.
- `acctadm(1M)` – Oracle Solaris 확장 계산 기능을 구성하는 명령을 설명합니다. `acctadm` 명령에는 IPQoS 확장이 포함됩니다.

IPQoS에서 QoS 제공

IPQoS 기능을 통해 인터넷 서비스 제공자(ISP) 및 응용 프로그램 서비스 제공자(ASP)는 고객에게 서로 다른 레벨의 네트워크 서비스를 제공할 수 있습니다. 이러한 기능을 통해 개별 회사 및 교육 기관은 내부 조직이나 주요 응용 프로그램에 대한 서비스 우선 순위를 지정할 수 있습니다.

서비스 단계 계약 구현

조직이 ISP 또는 ASP인 경우 IPQoS 구성을 회사에서 고객에게 제공하는 *SLA(***서비스 단계 계약***)*를 기준으로 할 수 있습니다. SLA에서 서비스 제공자는 고객에게 가격 구조를 기준으로 특정 레벨의 네트워크 서비스를 보장합니다. 예를 들어, 높은 가격의 SLA는 고객이 모든 유형의 네트워크 트래픽에 대해 24시간 내내 가장 높은 우선 순위를 가지도록 보장할 수 있습니다. 반면, 중간 가격의 SLA는 고객이 업무 시간 중 전자 메일에 대해서만 높은 우선 순위를 가지도록 보장할 수 있습니다. 기타 모든 트래픽은 24시간 내내 중간 우선 순위를 가집니다.

개별 조직에 대해 QoS 보장

조직이 엔터프라이즈이거나 기관인 경우 해당 네트워크에 대한 QoS를 제공할 수도 있습니다. 특정 그룹이나 특정 응용 프로그램의 트래픽이 더 높거나 낮은 서비스 레벨을 가지도록 보장할 수 있습니다.

QoS 정책 소개

*QoS(***서비스 품질***)* **정책**을 정의하여 서비스 품질을 구현합니다. QoS 정책은 고객 또는 응용 프로그램의 우선 순위 및 서로 다른 범주의 트래픽 처리를 위한 작업과 같이 다양한 네트워크 속성을 정의합니다. IPQoS 구성 파일에서 조직의 QoS 정책을 구현합니다. 이 파일은 Oracle Solaris 커널에 상주하는 IPQoS 모듈을 구성합니다. IPQoS 정책이 적용된 호스트는 *IPQoS* **사용 시스템**으로 간주됩니다.

QoS 정책에서는 일반적으로 다음을 정의합니다.

- **서비스 클래스**라고 하는 고유의 네트워크 트래픽 그룹.
- 각 클래스에 대한 네트워크 트래픽의 양을 규제하기 위한 측정 단위. 이러한 측정 단위는 **측정**이라고 하는 트래픽 측정 프로세스를 제어합니다.
- IPQoS 시스템 및 Diffserv 라우터가 패킷 흐름에 적용해야 하는 작업. 이 유형의 작업을 **홉별 동작**(PHB)이라고 합니다.
- 조직이 서비스 클래스에 대해 필요로 하는 통계 수집. 예를 들면 고객이나 특정 응용 프로그램이 생성하는 트래픽입니다.

패킷이 네트워크에 전달될 때 IPQoS 사용 시스템은 패킷 헤더를 검사합니다. IPQoS 시스템이 수행하는 작업은 QoS 정책으로 결정됩니다.

QoS 정책 설계를 위한 작업은 711 페이지 “서비스 품질 정책 계획”에 설명되어 있습니다.

IPQoS를 사용하여 네트워크 효율성 향상

IPQoS에는 QoS를 구현할 때 네트워크 성능을 더욱 효율화할 수 있는 기능이 포함되어 있습니다. 컴퓨터 네트워크가 확장되면 사용자 수 및 더욱 강력한 프로세서 증가로 생성되는 네트워크 트래픽 관리의 필요성도 높아집니다. 과다 사용되는 네트워크의 증상으로는 데이터 손실 및 트래픽 정체를 들 수 있습니다. 두 증상은 모두 느린 응답 시간이라는 결과를 초래합니다.

과거에는 시스템 관리자가 더 많은 대역폭을 추가하여 네트워크 트래픽 문제를 처리했습니다. 링크에서 트래픽 레벨은 광범위하게 변동하는 경우가 많습니다. IPQoS를 사용하면 기존 네트워크의 트래픽을 관리하고 확장이 필요한지 여부 및 필요한 위치를 평가할 수 있습니다.

예를 들어, 엔터프라이즈나 기관의 경우 트래픽 병목 현상을 피하려면 효율적인 네트워크를 유지 관리해야 합니다. 또한 그룹이나 응용 프로그램에서 할당된 대역폭보다 많이 소비하지 않도록 해야 합니다. ISP 또는 ASP의 경우, 고객이 지불한 레벨의 네트워크 서비스를 받도록 네트워크 성능을 관리해야 합니다.

대역폭이 네트워크 트래픽에 미치는 영향

IPQoS를 사용하여 네트워크 **대역폭**(완전히 사용된 네트워크 링크나 장치에서 전송할 수 있는 최대 데이터 양)을 규제할 수 있습니다. QoS 정책에서 대역폭 사용 우선 순위를 지정하여 고객이나 사용자에게 QoS를 제공해야 합니다. IPQoS 측정 모듈을 통해 IPQoS 사용 호스트에서 다양한 트래픽 클래스 간에 대역폭 할당을 측정하고 제어할 수 있습니다.

네트워크의 트래픽을 효과적으로 관리할 수 있으려면 먼저 대역폭 사용에 대한 다음 질문에 답해야 합니다.

- 귀사의 로컬 네트워크에서 트래픽 문제가 있는 영역은 어디입니까?
- 사용 가능한 대역폭을 최대한 사용하기 위해 무엇을 해야 합니까?
- 우선 순위가 가장 높은 사이트의 중요 응용 프로그램은 무엇입니까?
- 정체에 민감한 응용 프로그램은 무엇입니까?
- 낮은 우선 순위로 지정해도 되는 덜 중요한 응용 프로그램은 무엇입니까?

서비스 클래스를 사용하여 트래픽 우선 순위 지정

서비스 품질을 구현하려면 네트워크 트래픽을 분석하여 트래픽을 분할할 수 있는 모든 포괄적인 그룹 지정을 식별합니다. 그런 다음 여러 그룹을 개별 특성 및 개별 우선 순위를 가지는 서비스 클래스로 조직합니다. 이러한 클래스는 조직에 대한 QoS 정책의 기준이 되는 기본 범주를 형성합니다. 서비스 클래스는 제어할 트래픽 그룹을 나타냅니다.

예를 들어, 제공자는 계단식 가격 구조로 프리미엄, 골드, 실버 및 브론즈 레벨의 서비스를 제공할 수 있습니다. 프리미엄 SLA는 ISP가 고객을 위해 호스트하는 웹 사이트를 대상으로 한 수신 트래픽에 대해 가장 높은 우선 순위를 보장합니다. 따라서 고객의 웹 사이트에 대한 수신 트래픽이 하나의 트래픽 클래스가 될 수 있습니다.

엔터프라이즈의 경우, 부서 요구 사항을 기준으로 서비스 클래스를 만들 수 있습니다. 또는 네트워크 트래픽에서 특정 응용 프로그램의 수를 기준으로 클래스를 만들 수 있습니다.

다음은 엔터프라이즈에 대한 트래픽 클래스의 몇 가지 예입니다.

- 특정 서버에 대한 전자 메일 및 나가는 FTP와 같이 자주 사용되는 응용 프로그램(둘 다 하나의 클래스가 될 수 있음). 직원들은 이러한 응용 프로그램을 지속적으로 사용하므로 QoS 정책에서 전자 메일 및 나가는 FTP에 대해 적은 양의 대역폭과 낮은 우선 순위를 보장할 수 있습니다.
- 하루 24시간 실행되어야 하는 주문 입력 데이터베이스. 엔터프라이즈에 대한 데이터베이스 응용 프로그램의 중요도에 따라 데이터베이스에 많은 양의 대역폭과 높은 우선 순위를 지정할 수 있습니다.
- 중요한 업무 또는 민감한 업무를 수행하는 부서(예: 급여 부서). 조직에 대한 부서의 중요도에 따라 해당 부서에 지정하는 우선 순위 및 대역폭의 양이 결정됩니다.
- 회사의 외부 웹 사이트로 들어오는 호출. 이 클래스에는 낮은 우선 순위로 실행되는 적당한 양의 대역폭을 지정할 수 있습니다.

차별화 서비스 모델

IPQoS에는 RFC 2475에 정의된 **차별화 서비스***(Diffserv)* 아키텍처에 속하는 다음 모듈이 포함됩니다.

- 분류기
- 측정기
- 표시기

IPQoS는 Diffserv 모델에 다음 향상된 기능을 추가합니다.

- 플로우 계산 모듈
- 802.1D 데이터그램 표시기

이 절에서는 IPQoS에서 사용되는 Diffserv 모듈을 소개합니다. QoS 정책을 설정하려면 이러한 모듈, 이름 및 용도에 대해 알고 있어야 합니다. 각 모듈에 대한 자세한 내용은 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”을 참조하십시오.

분류기(ipgpc) 개요

Diffserv 모델에서 **분류기**는 네트워크 트래픽 플로우에서 패킷을 선택합니다. **트래픽 플로우**은 다음 IP 헤더 필드에서 동일한 정보를 가지는 패킷 그룹을 구성합니다.

- 소스 주소
- 대상 주소
- 소스 포트
- 대상 포트
- 프로토콜 번호

IPQoS에서 이러한 필드를 5-**튜플**이라고 합니다.

IPQoS 분류기 모듈의 이름은 `ipgpc`로 지정됩니다. `ipgpc` 분류기는 트래픽 흐름을 IPQoS 구성 파일에서 구성하는 특성을 기준으로 클래스로 분류합니다.

`ipgpc`에 대한 자세한 내용은 767 페이지 “분류기 모듈”을 참조하십시오.

IPQoS 클래스

클래스는 유사한 특성을 공유하는 네트워크 흐름의 그룹입니다. 예를 들어, ISP는 고객에게 제공하는 서로 다른 서비스 레벨을 나타내기 위해 클래스를 정의할 수 있습니다. ASP는 다양한 응용 프로그램에 서로 다른 서비스 레벨을 제공하는 SLA를 정의할 수 있습니다. ASP QoS 정책의 경우, 클래스에는 특정 대상 IP 주소로 향하는 FTP 트래픽이 포함될 수 있습니다. 회사의 외부 웹 사이트의 송신 트래픽도 클래스로 정의될 수 있습니다.

트래픽을 클래스로 그룹화하는 것은 QoS 정책 계획에서 큰 부분을 차지합니다. `ipqosconf` 유틸리티를 사용하여 클래스를 만드는 경우 실제로는 `ipgpc` 분류기를 구성하는 것입니다.

클래스를 정의하는 방법에 대한 자세한 내용은 713 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.

IPQoS 필터

필터는 **선택기**라고 부르는 매개변수가 포함된 일련의 규칙입니다. 각 필터는 클래스를 가리켜야 합니다. IPQoS는 패킷을 각 필터의 선택기에 대해 일치시켜 패킷이 해당 필터의 클래스에 속하는지 여부를 결정합니다. IPQoS 5-튜플 및 기타 공통 매개변수 등의 다양한 선택기를 사용하여 패킷을 필터링할 수 있습니다.

- 소스 주소 및 대상 주소
- 소스 포트 및 대상 포트
- 프로토콜 번호
- 사용자 ID
- 프로젝트 ID
- 차별화 서비스 코드 포인트(DSCP)

- 인터페이스 인덱스

예를 들어, 단순 필터에는 값이 80인 대상 포트가 포함될 수 있습니다. 그런 다음 ipgpc 분류기는 대상 포트 80(HTTP)으로 향하는 모든 패킷을 선택하고 QoS 정책에서 정의된 대로 패킷을 처리합니다.

필터 만들기에 대한 자세한 내용은 716 페이지 "QoS 정책에서 필터를 정의하는 방법"을 참조하십시오.

측정기(tokenmt 및 tswtclmt) 개요

Diffserv 모델에서 **측정기**는 클래스별 기준에서 트래픽 플로우의 전송 속도를 추적합니다. 측정기는 플로우의 실제 속도가 구성된 속도를 얼마나 준수하는지 평가하여 해당하는 결과를 결정합니다. 트래픽 플로우의 결과를 기준으로 측정기는 후속 작업을 선택합니다. 후속 작업에는 다른 작업으로 패킷 보내기 또는 추가 처리 없이 네트워크로 패킷 돌려보내기가 포함될 수 있습니다.

IPQoS 측정기는 네트워크 플로우가 QoS 정책에서 해당 클래스에 대해 정의된 전송 속도를 준수하는지 여부를 결정합니다. IPQoS에는 두 측정 모듈이 포함됩니다.

- `tokenmt` – 두 토큰 버킷 측정 체계를 사용합니다.
- `tswtclmt` – 시간별 창 측정 체계를 사용합니다.

두 측정 모듈은 모두 빨간색, 노란색 및 녹색의 세 가지 결과를 인식합니다. `red_action_name`, `yellow_action_name` 및 `green_action_name` 매개변수에서 각 결과에 대해 수행할 작업을 정의합니다.

또한 `tokenmt`가 색상을 인식하도록 구성할 수 있습니다. 색상 인식 측정 인스턴스에서는 패킷의 크기, DSCP, 트래픽 속도 및 구성된 매개변수를 사용하여 결과를 결정합니다. 측정기는 DSCP를 사용하여 패킷의 결과를 녹색, 노란색 또는 빨간색으로 매핑합니다.

IPQoS 측정기의 매개변수 정의에 대한 자세한 내용은 717 페이지 "플로우 제어 계획 방법"을 참조하십시오.

표시기(dscpmk 및 dlcosmk) 개요

Diffserv 모델에서 **표시기**는 패킷에 전달 동작을 반영하는 값을 표시합니다. **표시**는 패킷의 헤더에 값을 두어 패킷을 네트워크에 어떻게 전달할지 나타내는 프로세스입니다.

IPQoS에는 두 표시기 모듈이 포함됩니다.

- `dscpmk` – IP 패킷 헤더의 DS 필드를 **차별화 서비스 코드 포인트** 또는 *DSCP*라는 숫자 값으로 표시합니다. 그러면 Diffserv 인식 라우터에서 DS 코드 포인트를 사용하여 알맞은 전달 동작을 패킷에 적용할 수 있습니다.
- `dlcosmk` – 이더넷 프레임 헤더의 VLAN(virtual local area network) 태그를 **사용자 우선 순위**라는 숫자 값으로 표시합니다. 사용자 우선 순위는 데이터그램에 적용할 알맞은 전달 동작을 정의하는 *CoS(***서비스 클래스***)*를 나타냅니다.

  `dlcosmk`는 IETF에서 설계한 Diffserv 모델의 일부가 아닌 IPQoS 추가 기능입니다.

QoS 정책의 표시기 전략 구현에 대한 자세한 내용은 720 페이지 “전달 동작 계획 방법”을 참조하십시오.

플로우 계산(flowacct) 개요

IPQoS는 `flowacct` 정산 모듈을 Diffserv 모델에 추가합니다. `flowacct`를 사용하여 트래픽 플로우에 대한 통계를 수집하고 해당 SLA에 따라 고객에게 청구할 수 있습니다. 플로우 계산은 용량 계획 및 시스템 모니터링에도 유용합니다.

`flowacct` 모듈은 `acctadm` 명령과 함께 작동하여 계산 로그 파일을 만듭니다. 기본 로그에는 다음 목록에 나온 대로 IPQoS 5-튜플 및 두 가지 추가 속성이 포함됩니다.

- 소스 주소
- 소스 포트
- 대상 주소
- 대상 포트
- 프로토콜 번호
- 패킷 수
- 바이트 수

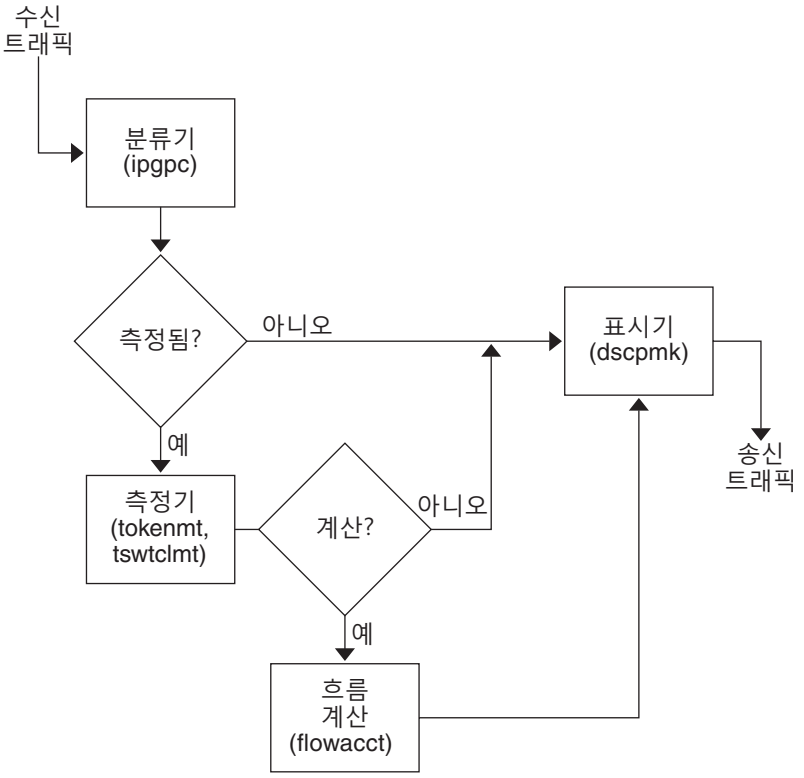
762 페이지 “트래픽 플로우에 대한 정보 기록”, `flowacct(7ipp)` 및 `acctadm(1M)` 매뉴얼 페이지에 설명된 대로 다른 속성에 대한 통계도 수집할 수 있습니다.

플로우 계산 전략 계획에 대한 자세한 내용은 722 페이지 “플로우 계산 계획 방법”을 참조하십시오.

IPQoS 모듈을 통한 트래픽 플로우 방식

다음 그림은 수신 트래픽이 몇 가지 IPQoS 모듈을 통과할 수 있는 경로를 보여줍니다.

그림 29–1 Diffserv 모델의 IPQoS 구현을 통한 트래픽 흐름



이 그림은 IPQoS 사용 시스템에서 일반적인 트래픽 흐름 시퀀스를 보여줍니다.

1. 분류기는 패킷 스트림에서 시스템 QoS 정책의 필터링 조건과 일치하는 모든 패킷을 선택합니다.
2. 그런 다음 선택된 패킷은 수행할 다음 작업에 대해 평가됩니다.
3. 분류기는 플로우 제어가 필요하지 않은 모든 트래픽을 표시기로 보냅니다.
4. 플로우 제어가 필요한 트래픽은 측정기로 보내집니다.
5. 측정기는 구성된 속도를 적용합니다. 그런 다음 측정기는 트래픽 준수 값을 플로우 제어 패킷에 지정합니다.
6. 그런 다음 플로우 제어 패킷은 평가되어 패킷에 계산이 필요한지 여부를 결정합니다.
7. 측정기는 플로우 계산이 필요하지 않은 모든 트래픽을 표시기로 보냅니다.
8. 플로우 계산 모듈은 수신된 패킷에 대한 통계를 수집합니다. 그런 다음 모듈은 패킷을 표시기로 보냅니다.
9. 표시기는 DS 코드 포인트를 패킷 헤더에 지정합니다. 이 DSCP는 Diffserv 인식 시스템에서 패킷에 적용해야 하는 홉별 동작을 나타냅니다.

IPQoS 사용 네트워크에서 트래픽 전달

이 절에서는 IPQoS 사용 네트워크에서 패킷 전달과 관련된 요소를 소개합니다. IPQoS 사용 시스템은 네트워크 스트림에서 시스템의 IP 주소를 대상으로 가지는 모든 패킷을 처리합니다. 그런 다음 IPQoS 시스템은 QoS 정책을 패킷에 적용하여 차별화 서비스를 설정합니다.

DS 코드 포인트

DSCP(DS 코드 포인트)는 패킷 헤더에서 Diffserv 인식 시스템이 표시된 패킷에 대해 수행해야 하는 작업을 정의합니다. Diffserv 아키텍처는 사용할 IPQoS 사용 시스템 및 Diffserv 라우터에 대한 DS 코드 포인트 집합을 정의합니다. 또한 Diffserv 아키텍처는 DSCP와 일치하는 **전달 동작**이라는 작업 집합을 정의합니다. IPQoS 사용 시스템은 패킷 헤더에서 DS 필드의 우선권 비트를 DSCP로 표시합니다. 라우터가 DSCP 값이 있는 패킷을 수신하면 라우터는 해당 DSCP와 연결된 전달 동작을 적용합니다. 그런 다음 패킷은 네트워크로 보내집니다.

주 – `dlcosmk` 표시기는 DSCP를 사용하지 않습니다. 대신 `dlcosmk`가 이더넷 프레임 헤더를 CoS 값으로 표시합니다. VLAN 장치를 사용하는 네트워크에서 IPQoS를 구성하려는 경우 772 페이지 “표시기 모듈”을 참조하십시오.

홉별 동작

Diffserv 용어에서 DSCP에 지정된 전달 동작을 *PHB(**홉별 동작**)*이라고 합니다. PHB는 Diffserv 인식 시스템에서 다른 트래픽과 관련하여 표시된 패킷이 수신하는 전달 우선권을 정의합니다. 이 우선권은 최종적으로 IPQoS 사용 시스템이나 Diffserv 라우터가 표시된 패킷을 전달할지 또는 삭제할지 결정합니다. 전달되는 패킷의 경우, 대상으로 향하는 경로에서 패킷이 만나는 각 Diffserv 라우터는 동일한 PHB를 적용합니다. 다른 Diffserv 시스템이 DSCP를 변경할 경우는 예외입니다. PHB에 대한 자세한 내용은 772 페이지 “패킷 전달을 위해 dscpmk 표시기 사용”을 참조하십시오.

PHB의 목적은 지정된 양의 네트워크 리소스를 인접 네트워크의 트래픽 클래스에 제공하는 것입니다. QoS 정책에서 이 목적을 달성할 수 있습니다. 트래픽 흐름이 IPQoS 사용 시스템을 떠날 때 트래픽 클래스에 대한 우선권 레벨을 나타내는 DSCP를 정의합니다. 우선권은 높은 우선권/낮은 삭제 가능성에서 낮은 우선권/높은 삭제 가능성의 범위에 있을 수 있습니다.

예를 들어, QoS 정책은 한 트래픽 클래스에 낮은 삭제 가능성의 PHB를 보장하는 DSCP를 지정할 수 있습니다. 그러면 이 트래픽 클래스는 Diffserv 인식 라우터에서 이 클래스의 패킷에 대역폭을 보장하는 낮은 삭제 우선권의 PHB를 수신합니다. 다양한 레벨의 우선권을 다른 트래픽 클래스에 지정하는 다른 DSCP를 QoS 정책에 추가할 수 있습니다. 낮은 우선권의 패킷에는 패킷의 DSCP에 표시된 우선 순위에 따라 Diffserv 시스템에서 대역폭을 제공합니다.

IPQoS는 Diffserv 아키텍처에서 정의된 빠른 전달 및 보장 전달의 두 가지 전달 동작 유형을 지원합니다.

빠른 전달

*EF(***빠른 전달***)* 홉별 동작은 EF 관련 DSCP를 가진 트래픽 클래스에 가장 높은 우선 순위가 부여되도록 합니다. EF DSCP를 가진 트래픽은 대기열에 두지 않습니다. EF는 낮은 손실, 대기 시간 및 지터를 제공합니다. EF에 권장되는 DSCP는 101110입니다. 101110으로 표시된 패킷은 대상으로 향하는 경로에서 Diffserv 인식 네트워크를 통과할 때 보장된 낮은 삭제 우선권을 받습니다. 프리미엄 SLA의 고객이나 응용 프로그램에 우선 순위를 지정할 때 EF DSCP를 사용합니다.

보장 전달

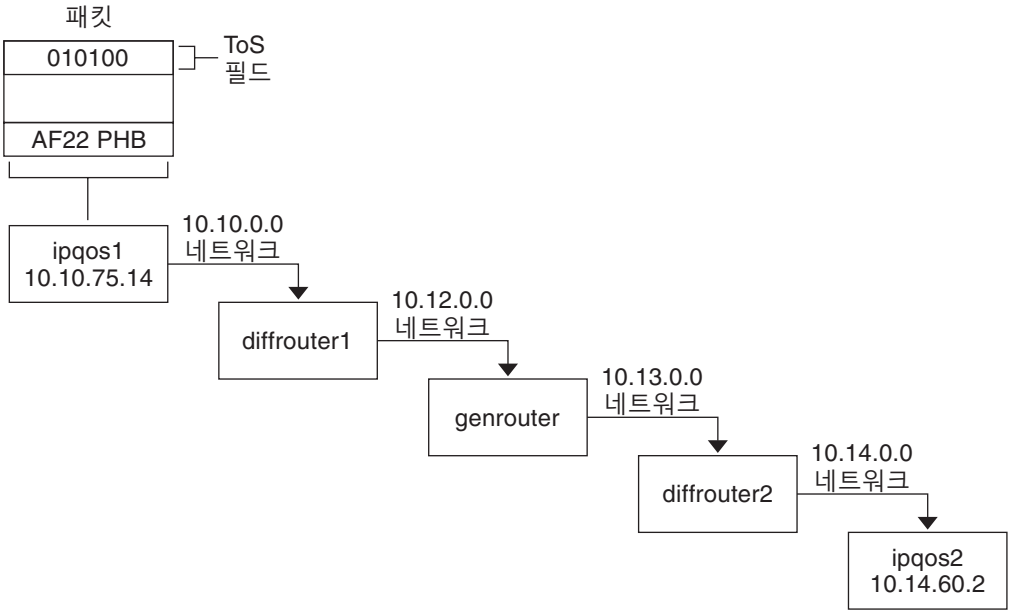
*AF(***보장 전달***)* 홉별 동작은 패킷에 지정할 수 있는 4가지 서로 다른 전달 클래스를 제공합니다. 모든 전달 클래스는 표 34–2에 나온 대로 3가지 삭제 우선권을 제공합니다.

다양한 AF 코드 포인트는 고객 및 응용 프로그램에 서로 다른 레벨의 서비스를 지정할 수 있는 기능을 제공합니다. QoS 정책에서는 QoS 정책을 계획할 때 네트워크에서 트래픽 및 서비스의 우선 순위를 지정할 수 있습니다. 그런 다음 서로 다른 AF 레벨을 우선 순위가 지정된 트래픽에 지정할 수 있습니다.

Diffserv 환경에서 패킷 전달

다음 그림은 부분적으로 Diffserv 사용 환경을 갖춘 회사 인트라넷의 일부를 보여줍니다. 이 시나리오에서 `10.10.0.0` 및 `10.14.0.0` 네트워크의 모든 호스트는 IPQoS가 사용으로 설정되어 있고, 두 네트워크의 로컬 라우터는 Diffserv를 인식합니다. 하지만 임시 네트워크는 Diffserv에 대해 구성되지 않았습니다.

그림 29–2 Diffserv 인식 네트워크 홉에서 패킷 전달



다음 단계에서는 이 그림에 표시된 패킷의 흐름을 추적합니다. 단계는 ipqos1 호스트에서 발생하는 패킷의 진행부터 시작됩니다. 그런 다음 단계는 여러 홉을 거쳐 ipqos2 호스트로 계속됩니다.

1. ipqos1의 사용자는 ftp 명령을 실행하여 세 홉 떨어진 ipqos2 호스트에 액세스합니다.
2. ipqos1은 QoS 정책을 결과 패킷 흐름에 적용합니다. 그런 다음 ipqos1은 ftp 트래픽을 성공적으로 분류합니다.

   시스템 관리자는 로컬 네트워크 10.10.0.0에서 발생되는 모든 나가는 ftp 트래픽에 대한 클래스를 만들었습니다. ftp 클래스에 대한 트래픽에는 클래스 2인 중간 삭제 우선권의 AF22 홉별 동작이 지정되었습니다. ftp 클래스에 대해서는 2Mb/초의 트래픽 흐름 속도가 구성되었습니다.
3. ipqos-1은 ftp 흐름을 측정하여 흐름이 2Mb/초의 약정된 속도를 초과하는지 여부를 결정합니다.
4. ipqos1의 표시기는 나가는 ftp 패킷의 DS 필드를 AF22 PHB와 일치하는 010100 DSCP로 표시합니다.
5. diffrouter1 라우터는 ftp 패킷을 수신합니다. 그런 다음 diffrouter1은 DSCP를 검사합니다. diffrouter1가 정체된 경우 AF22로 표시된 패킷은 삭제됩니다.
6. ftp 트래픽은 diffrouter1의 파일에서 AF22에 대해 구성된 홉별 동작에 따라 다음 홉으로 전달됩니다.
7. ftp 트래픽은 10.12.0.0 네트워크를 통과하여 Diffserv를 인식하지 못하는 genrouter로 이동합니다. 결과적으로 트래픽은 "최선 조건" 전달 동작을 수신합니다.

8. genrouter는 ftp 트래픽을 10.13.0.0 네트워크에 전달합니다. 여기에서 트래픽은 diffrouter2로 수신됩니다.
9. diffrouter2는 Diffserv를 인식합니다. 따라서 라우터는 AF22 패킷에 대한 라우터 정책에서 정의된 PHB에 따라 ftp 패킷을 네트워크에 전달합니다.
10. ipqos2는 ftp 트래픽을 수신합니다. 그런 다음 ipqos2는 ipqos1의 사용자에게 사용자 이름과 암호를 물어봅니다.

◆ ◆ ◆ 30장

30 IPQoS 사용 네트워크 계획(작업)

Oracle Solaris를 실행하는 모든 시스템에서 IPQoS를 구성할 수 있습니다. 그러면 IPQoS 시스템은 Diffserv 인식 라우터와 함께 작동하여 인트라넷에서 차별화된 서비스와 트래픽 관리를 제공합니다.

이 장에는 Diffserv 인식 네트워크에 IPQoS 사용 시스템을 추가하는 계획 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

일반 IPQoS 구성 계획(작업 맵)

네트워크에서 IPQoS를 비롯하여 차별화된 서비스를 구현하려면 광범위한 계획이 필요합니다. 각 IPQoS 사용 시스템의 위치 및 기능뿐 아니라 각 시스템과 로컬 네트워크에 있는 라우터의 관계도 고려해야 합니다. 다음 작업 맵에서는 네트워크에서 IPQoS를 구현하는 주요 계획 작업을 나열하고 작업을 완료하는 데 필요한 절차와 관련된 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS 사용 시스템을 통합하는 Diffserv 네트워크 토폴로지를 계획합니다.	다양한 Diffserv 네트워크 토폴로지를 조사하여 사이트에 가장 적합한 솔루션을 결정합니다.	708 페이지 "Diffserv 네트워크 토폴로지 계획"
2. IPQoS 시스템이 제공할 다양한 유형의 서비스를 계획합니다.	네트워크가 제공하는 서비스의 유형을 SLA(서비스 단계 계약)별로 구성합니다.	711 페이지 "서비스 품질 정책 계획"

작업	설명	수행 방법
3. 각 IPQoS 시스템에 대한 QoS 정책을 계획합니다.	각 SLA 구현에 필요한 클래스, 측정 및 계산 기능을 결정합니다.	711 페이지 "서비스 품질 정책 계획"
4. 해당하는 경우 Diffserv 라우터에 대한 정책을 계획합니다.	IPQoS 시스템에서 사용되는 Diffserv 라우터에 대한 일정 잡기 및 대기열 지정 정책을 결정합니다.	대기열 지정 및 일정 잡기 정책은 라우터 설명서를 참조하십시오.

Diffserv 네트워크 토폴로지 계획

네트워크에 대해 차별화된 서비스를 제공하려면 하나 이상의 IPQoS 사용 시스템 및 Diffserv 인식 라우터가 필요합니다. 이 절에 설명된 다양한 방법으로 이와 같은 기본 시나리오를 확장할 수 있습니다.

Diffserv 네트워크에 대한 하드웨어 전략

일반적으로 고객은 서버 및 서버 통합(예: Oracle의 Sun Enterprise(tm) 서버)에서 IPQoS를 실행합니다. 반대로 네트워크 요구 사항에 따라 데스크탑 시스템(예: UltraSPARC® 시스템)에서도 IPQoS를 실행할 수 있습니다.

다음 목록에서는 IPQoS 구성이 가능한 시스템에 대해 설명합니다.

- 웹 서버, 데이터베이스 서버 등의 다양한 서비스를 제공하는 Oracle Solaris 시스템
- 전자 메일, FTP 또는 기타 많이 사용되는 네트워크 응용 프로그램을 제공하는 애플리케이션 서버
- 웹 캐시 서버 또는 프록시 서버
- Diffserv 인식 로드 밸런서가 관리하는 IPQoS 사용 서버 팜의 네트워크
- 단일 이기종 네트워크에 대한 트래픽을 관리하는 방화벽
- 가상 근거리 통신망(LAN)의 일부인 IPQoS 시스템

Diffserv 인식 라우터가 이미 작동되고 있는 네트워크 토폴로지에 IPQoS 시스템을 도입할 수 있습니다. 라우터가 현재 Diffserv를 제공하지 않을 경우 Cisco Systems, Juniper Networks 및 기타 라우터 제조업체에서 제공하는 Diffserv 솔루션을 고려해 보십시오. 로컬 라우터가 Diffserv를 구현하지 않은 경우 라우터는 표시를 평가하지 않은 상태로 표시된 패킷을 다음 홉으로 전달합니다.

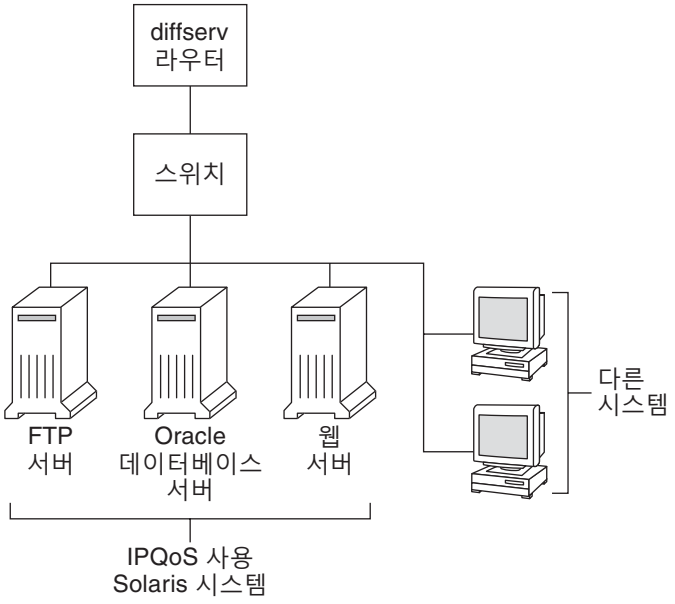
IPQoS 네트워크 토폴로지

이 절에서는 다양한 네트워크 요구 사항에 대한 IPQoS 전략에 대해 설명합니다.

개별 호스트의 IPQoS

다음 그림에서는 IPQoS 사용 시스템의 단일 네트워크를 보여 줍니다.

그림 30–1 네트워크 세그먼트의 IPQoS 시스템



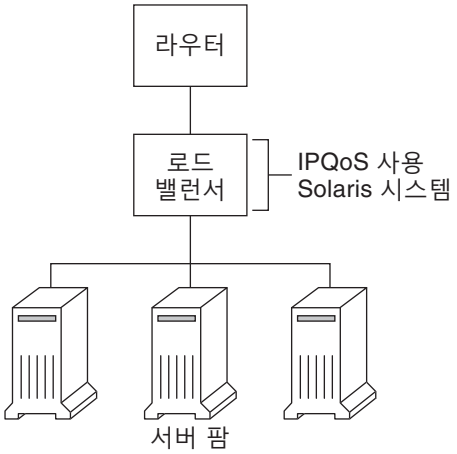
이 네트워크는 회사 인트라넷의 유일한 세그먼트입니다. 애플리케이션 서버 및 웹 서버에서 IPQoS를 사용으로 설정하면 각 IPQoS 시스템이 송신 트래픽을 릴리스하는 속도를 제어할 수 있습니다. 라우터가 Diffserv를 인식하도록 설정할 경우 추가로 수신 및 송신 트래픽을 제어할 수 있습니다.

본 설명서의 예에는 "개별 호스트의 IPQoS" 시나리오가 사용됩니다. 설명서 전체에서 사용되는 토폴로지 예는 그림 30–4를 참조하십시오.

서버 팜 네트워크의 IPQoS

다음 그림에서는 이기종 서버 팜이 여러 개인 네트워크를 보여 줍니다.

그림 30–2 IPQoS 사용 서버 팜의 네트워크



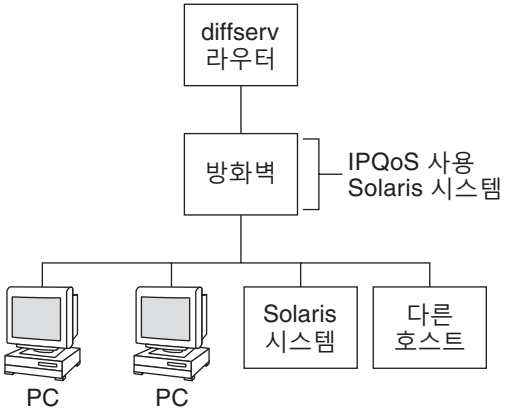
해당 토폴로지에서는 라우터가 Diffserv를 인식하므로 수신 트래픽과 송신 트래픽을 모두 대기열에 지정하고 속도를 제어할 수 있습니다. 로드 밸런서도 Diffserv를 인식하며 서버 팜에서 IPQoS가 사용됩니다. 로드 밸런서가 사용자 ID, 프로젝트 ID 등의 선택기를 사용하여 라우터 이외의 추가 필터링을 제공할 수 있습니다. 이러한 선택기는 응용 프로그램 데이터에 포함되어 있습니다.

이 시나리오는 로컬 네트워크의 혼잡을 관리할 수 있도록 흐름 제어 및 트래픽 전달을 제공합니다. 또한 이 시나리오는 서버 팜의 송신 트래픽으로 인해 인트라넷의 다른 부분이 과부화되지 않도록 합니다.

방화벽의 IPQoS

다음 그림에서는 방화벽을 통해 다른 세그먼트로부터 보호되는 회사 네트워크 세그먼트를 보여 줍니다.

그림 30–3 IPQoS 사용 방화벽으로 보호되는 네트워크



이 시나리오에서 트래픽은 패킷이 필터링되고 대기열에 지정되는 Diffserv 인식 라우터로 들어옵니다. 라우터가 전달한 모든 수신 트래픽은 IPQoS 사용 방화벽을 통과합니다. IPQoS를 사용하려면 방화벽이 IP 전달 스택을 무시하지 않아야 합니다.

방화벽의 보안 정책에 따라 수신 트래픽이 내부 네트워크로 들어오거나 나갈 수 있는지 여부가 결정됩니다. QoS 정책은 방화벽을 통과한 수신 트래픽의 서비스 레벨을 제어합니다. QoS 정책에 따라 송신 트래픽에 전달 동작을 표시할 수도 있습니다.

서비스 품질 정책 계획

서비스 품질(QoS) 정책을 계획할 때는 네트워크가 제공하는 서비스를 검토 및 분류하고 우선 순위를 설정해야 합니다. 또한 사용 가능한 대역폭을 평가하여 각 트래픽 클래스가 네트워크로 릴리스되는 속도를 결정해야 합니다.

QoS 정책 계획 지원

IPQoS 구성 파일에 필요한 정보를 포함하는 형식으로 QoS 정책을 계획하기 위한 정보를 수집합니다. 예를 들어, 다음 템플리트를 사용하여 IPQoS 구성 파일에서 사용될 주요 정보 범주를 나열할 수 있습니다.

표 30–1 QoS 계획 템플리트

클래스	우선 순위	필터	선택기	속도	전달 여부	계산 여부
클래스 1	1	필터 1 필터 3	선택기 1 선택기 2	측정기 유형에 따른 측정기 속도	표시자 삭제 우선 순위	플로우 계산 통계 필요

표 30–1 QoS 계획 템플리트 (계속)

클래스	우선 순위	필터	선택기	속도	전달 여부	계산 여부
클래스 1	1	필터 2	선택기 1 선택기 2	해당 없음	해당 없음	해당 없음
클래스 2	2	필터 1	선택기 1 선택기 2	측정기 유형에 따른 측정기 속도	표시자 삭제 우선 순위	플로우 계산 통계 필요
클래스 2	2	필터 2	선택기 1 선택기 2	해당 없음	해당 없음	해당 없음

각 주요 범주를 구분하여 추가로 QoS 정책을 정의할 수 있습니다. 후속 절에서는 템플리트에 표시되는 범주에 대한 정보를 얻는 방법을 설명합니다.

QoS 정책 계획(작업 맵)

이 작업 맵에서는 QoS 정책 계획에 대한 주요 작업을 나열하고 각 작업에 대한 수행 지침과 관련된 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS를 지원하도록 네트워크 토폴로지를 설계합니다.	네트워크에서 차별화된 서비스를 제공할 호스트 및 라우터를 식별합니다.	713 페이지 “네트워크에서 IPQoS를 준비하는 방법”
2. 네트워크의 서비스를 구분해야 할 클래스를 정의합니다.	사이트에서 제공하는 서비스 유형 및 SLA를 확인하고 해당 서비스가 속한 고유한 트래픽 클래스를 결정합니다.	713 페이지 “QoS 정책에 대한 클래스 정의 방법”
3. 클래스에 대한 필터를 정의합니다.	특정 클래스의 트래픽과 네트워크 트래픽 플로우를 구분할 가장 적합한 방법을 결정합니다.	716 페이지 “QoS 정책에서 필터를 정의하는 방법”
4. 패킷이 IPQoS 시스템에서 나갈 때 트래픽을 측정할 플로우 제어 속도를 정의합니다.	각 트래픽 클래스에 대해 허용 가능한 플로우 속도를 결정합니다.	717 페이지 “플로우 제어 계획 방법”
5. QoS 정책에서 사용할 DSCP 또는 사용자 우선 순위 값을 정의합니다.	라우터 또는 스위치가 흐름을 처리할 때 트래픽 흐름에 지정되는 전달 동작을 결정할 체계를 계획합니다.	720 페이지 “전달 동작 계획 방법”
6. 해당하는 경우 네트워크의 트래픽 흐름에 대한 통계 모니터링 계획을 설정합니다.	트래픽 클래스를 평가하여 계산 또는 통계 용도로 모니터링해야 할 트래픽 흐름을 결정합니다.	722 페이지 “플로우 계산 계획 방법”

주 – 이 절의 나머지 부분에서는 IPQoS 사용 시스템의 QoS 정책을 계획하는 방법에 대해 설명합니다. Diffserv 라우터에 대한 QoS 정책을 계획하려면 라우터 설명서 및 라우터 제조업체 웹 사이트를 참조하십시오.

▼ 네트워크에서 IPQoS를 준비하는 방법

다음 절차에서는 QoS 정책을 만들기 전에 수행할 일반적인 계획 작업을 나열합니다.

1 **네트워크 토폴로지를 검토합니다. 그런 다음 IPQoS 시스템 및 Diffserv 라우터를 사용하는 전략을 계획합니다.**

토폴로지 예는 708 페이지 "Diffserv 네트워크 토폴로지 계획"을 참조하십시오.

2 **토폴로지에서 IPQoS를 필요로 하거나 IPQoS 서비스로 사용 가능한 적합한 후보가 될 수 있는 호스트를 식별합니다.**

3 **동일한 QoS 정책을 사용할 수 있는 IPQoS 사용 시스템을 결정합니다.**

예를 들어, 네트워크의 모든 호스트에서 IPQoS를 사용으로 설정하려면 동일한 QoS 정책을 사용할 수 있는 호스트를 식별합니다. 각 IPQoS 사용 시스템에는 해당 IPQoS 구성 파일에서 구현되는 로컬 QoS 정책이 있어야 합니다. 하지만 특정 범위의 시스템에서 사용할 하나의 IPQoS 구성 파일을 만들 수 있습니다. 그런 다음 QoS 정책 요구 사항이 동일한 모든 시스템에 구성 파일을 복사할 수 있습니다.

4 **네트워크의 Diffserv 라우터에 필요한 계획 작업을 검토하고 수행합니다.**

자세한 내용은 라우터 설명서 및 라우터 제조업체 웹 사이트를 참조하십시오.

▼ QoS 정책에 대한 클래스 정의 방법

첫번째 QoS 정책 정의 단계는 트래픽 플로우를 클래스로 구성하는 것입니다. Diffserv 네트워크에서 모든 유형의 트래픽에 대해 클래스를 만들 필요는 없습니다. 네트워크 토폴로지에 따라 각 IPQoS 사용 시스템에 대해 다른 QoS 정책을 만들어야 할 수도 있습니다.

주 – 클래스 개요는 699 페이지 "IPQoS 클래스"를 참조하십시오.

다음 절차에서는 713 페이지 "네트워크에서 IPQoS를 준비하는 방법"에 설명된 대로 IPQoS를 사용할 네트워크의 시스템을 결정했다고 가정합니다.

1 **QoS 정책 정보를 구성하는 데 필요한 QoS 계획 테이블을 만듭니다.**

제안 사항은 표 30–1을 참조하십시오.

2 네트워크에 있는 모든 QoS 정책에 대해 나머지 단계를 수행합니다.

3 QoS 정책에서 사용할 클래스를 정의합니다.

다음 질문은 가능한 클래스 정의를 위한 네트워크 트래픽 분석 지침입니다.

- **회사에서 고객에게 서비스 단계 계약을 제공합니까?**

  그럴 경우 회사에서 고객에게 제공하는 SLA의 상대적인 우선 순위 레벨을 평가합니다. 다른 우선 순위 레벨이 보장된 고객에게 동일한 응용 프로그램을 제공할 수 있습니다.

  예를 들어, 회사에서 각 고객에게 웹 사이트 호스팅을 제공할 수 있습니다. 이 경우 각 고객 웹 사이트에 대한 클래스를 정의해야 합니다. 고급 웹 사이트를 하나의 서비스 레벨로 제공하는 SLA도 있을 수 있고, 할인 고객에게 "최상의" 개인 웹 사이트를 제공하는 SLA도 있을 수 있습니다. 이 요소는 다양한 웹 사이트를 클래스뿐만 아니라 웹 사이트 클래스에 지정되는 잠재적으로 다른 홉별 동작도 나타냅니다.

- **IPQoS 시스템이 흐름 제어가 필요할 수 있는 많이 사용되는 응용 프로그램을 제공합니까?**

  과도한 트래픽을 생성하는 많이 사용되는 응용 프로그램을 제공하는 서버에서 IPQoS를 사용으로 설정하여 네트워크 성능을 향상시킬 수 있습니다. 일반적인 예로 전자 메일, 네트워크 뉴스 및 FTP를 들 수 있습니다. 가능한 경우 서비스 유형별로 수신 및 송신 트래픽에 대해 별도의 클래스를 만드는 것이 좋습니다. 예를 들어, 메일 서버용 QoS 정책에 대해 `mail-in` 클래스와 `mail-out` 클래스를 만들 수 있습니다.

- **네트워크에서 우선 순위가 가장 높은 전달 동작을 필요로 하는 특정 응용 프로그램이 실행됩니까?**

  우선 순위가 가장 높은 전달 동작을 필요로 하는 중요한 응용 프로그램은 라우터 대기열에서 가장 높은 우선 순위를 받아야 합니다. 일반적인 예로 스트리밍 비디오 및 스트리밍 오디오를 들 수 있습니다.

  이와 같이 우선 순위가 높은 응용 프로그램에 대해 수신 클래스와 송신 클래스를 정의합니다. 그런 다음 응용 프로그램을 제공하는 IPQoS 사용 시스템과 Diffserv 라우터의 QoS 정책에 클래스를 추가합니다.

- **흐름에 대역폭이 많이 사용되어 네트워크에서 트래픽 흐름이 제어되어야 합니까?**

  `netstat`, `snoop` 및 기타 네트워크 모니터링 유틸리티를 사용하여 네트워크에 문제를 일으키고 있는 트래픽의 유형을 검색할 수 있습니다. 지금까지 만든 클래스를 검토한 다음 정의되지 않은 문제 트래픽 범주에 대해 새 클래스를 만듭니다. 문제 트래픽 범주에 대한 클래스를 이미 정의한 경우 문제 트래픽을 제어할 측정기의 속도를 정의합니다.

  네트워크에 있는 모든 IPQoS 사용 시스템의 문제 트래픽에 대한 클래스를 만듭니다. 그러면 각 IPQoS 시스템이 트래픽 흐름을 네트워크로 릴리스하는 속도를 제한하여 문제 트래픽을 처리할 수 있습니다. 또한 Diffserv 라우터에서 QoS 정책에 해당 문제 클래스를 정의해야 합니다. 그러면 라우터가 QoS 정책에 구성된 대로 문제 흐름을 대기열에 지정하고 일정을 잡을 수 있습니다.

- **특정 유형의 트래픽에 대한 통계를 얻어야 합니까?**

간단한 SLA 검토를 통해 계산해야 할 고객 트래픽의 유형을 확인할 수 있습니다. 사이트에서 SLA를 제공하는 경우 계산해야 할 트래픽에 대한 클래스가 이미 만들어진 상태일 것입니다. 모니터링하고 있는 트래픽 흐름에 대한 통계 수집을 사용으로 설정할 클래스를 정의할 수도 있습니다. 또한 보안상 액세스를 제한할 트래픽에 대한 클래스를 만들 수 있습니다.

4 **1단계에서 만든 QoS 계획 테이블에서 정의한 클래스를 나열합니다.**

5 **각 클래스에 우선 순위 레벨을 지정합니다.**

예를 들어, 우선 순위 레벨 1이 가장 높은 우선 순위의 클래스를 나타내도록 지정하고 나머지 클래스에 우선 순위를 내림차순으로 지정합니다. 지정한 우선 순위 레벨은 구조적인 용도로만 사용됩니다. QoS 정책 템플리트에서 설정한 우선 순위 레벨은 IPQoS에 실제로 사용되지 않습니다. QoS 정책에 적합한 경우 두 개 이상의 클래스에 동일한 우선 순위를 지정할 수도 있습니다.

6 **클래스 정의가 완료되면 716 페이지 "QoS 정책에서 필터를 정의하는 방법"에 설명된 대로 각 클래스에 대한 필터를 정의합니다.**

자세한 정보

클래스 우선 순위 설정

클래스를 만들면 우선 순위가 가장 높은 클래스, 우선 순위가 중간인 클래스, 우선 순위가 최상인 클래스를 빠르게 파악할 수 있습니다. 적합한 클래스 우선 순위 설정 체계는 720 페이지 "전달 동작 계획 방법"에 설명된 대로 송신 트래픽에 홉별 동작을 지정할 때 특히 중요합니다.

클래스에 PHB를 지정하는 것 외에 클래스에 대한 필터에 우선 순위 선택기를 정의할 수도 있습니다. 우선 순위 선택기는 IPQoS 사용 호스트에서만 활성화됩니다. 속도와 DSCP가 동일한 여러 클래스가 IPQoS 시스템에서 나갈 때 대역폭 경합이 발생하는 경우가 있다고 가정합니다. 이 경우 각 클래스의 우선 순위 선택기가 동일한 값의 클래스에 지정된 서비스 레벨의 순서를 추가로 지정할 수 있습니다.

필터 정의

패킷 흐름을 특정 클래스의 구성원으로 식별할 필터를 만듭니다. 각 필터에는 패킷 플로우 평가 기준을 정의하는 선택기가 포함되어 있습니다. 그러면 IPQoS 사용 시스템이 선택기의 기준을 사용하여 트래픽 플로우에서 패킷을 추출합니다. 그런 다음 IPQoS 시스템이 패킷을 클래스와 연관시킵니다. 필터 소개는 699 페이지 "IPQoS 필터"를 참조하십시오.

다음 표에서는 가장 일반적으로 사용되는 선택기를 나열합니다. 처음 다섯 개의 선택기는 IPQoS 시스템이 패킷을 플로우 구성원으로 식별하는 데 사용하는 IPQoS 5 튜플을 나타냅니다. 전체 선택기 목록은 표 34–1을 참조하십시오.

표 30–2 일반적인 IPQoS 선택기

이름	정의
saddr	소스 주소입니다.
daddr	대상 주소입니다.
sport	소스 포트 번호입니다. /etc/services에 정의된 잘 알려진 포트 번호 또는 사용자 정의 포트 번호를 사용할 수 있습니다.
dport	대상 포트 번호입니다.
protocol	/etc/protocols의 트래픽 플로우 유형에 지정된 IP 프로토콜 번호 또는 프로토콜 이름입니다.
ip_version	사용할 주소 지정 스타일입니다. IPv4 또는 IPv6을 사용하십시오. IPv4가 기본값입니다.
dsfield	DS 필드 내용, 즉 DSCP입니다. 이미 특정 DSCP가 표시된 수신 패킷을 추출하려면 이 선택기를 사용하십시오.
priority	클래스에 지정된 우선 순위 레벨입니다. 자세한 내용은 713 페이지 "QoS 정책에 대한 클래스 정의 방법"을 참조하십시오.
user	상위 레벨 응용 프로그램이 실행될 때 사용되는 UNIX 사용자 ID 또는 사용자 이름입니다.
projid	상위 레벨 응용 프로그램이 실행될 때 사용되는 프로젝트 ID입니다.
direction	트래픽 플로우 방향입니다. 값은 LOCAL_IN, LOCAL_OUT, FWD_IN 또는 FWD_OUT입니다.

주 – 선택기를 선택할 때는 신중하십시오. 클래스에 대한 패킷을 추출하는 데 필요한 만큼만 선택기를 사용하십시오. 선택기를 많이 정의할수록 IPQoS 성능에 끼치는 영향이 커집니다.

▼ QoS 정책에서 필터를 정의하는 방법

시작하기 전에 다음 단계를 수행하려면 713 페이지 "QoS 정책에 대한 클래스 정의 방법" 절차를 완료해야 합니다.

1. **713 페이지 "QoS 정책에 대한 클래스 정의 방법"에서 만든 QoS 계획 테이블에 각 클래스에 대한 필터를 하나 이상 만듭니다.**

   가능한 경우 클래스별로 수신 및 송신 트래픽에 대해 별도의 필터를 만드는 것이 좋습니다. 예를 들어, IPQoS 사용 FTP 서버의 QoS 정책에 ftp-in 필터 및 ftp-out 필터를 추가합니다. 그런 다음 기본 선택기 외에 적합한 direction 선택기도 정의할 수 있습니다.

2. **클래스의 각 필터에 대한 선택기를 하나 이상 정의합니다.**

   표 30–1에서 소개된 QoS 계획 테이블을 사용하여 정의한 클래스에 대한 필터를 채웁니다.

예 30–1 FTP 트래픽에 대한 필터 정의

다음 표는 송신 FTP 트래픽에 대한 필터를 정의하는 방법을 보여 주는 예입니다.

클래스	우선 순위	필터	선택기
ftp-traffic	4	ftp-out	saddr 10.190.17.44
			daddr 10.100.10.53
			sport 21
			direction LOCAL_OUT

참조

- 플로우 제어 체계를 정의하려면 717 페이지 "플로우 제어 계획 방법"을 참조하십시오.
- 플로우가 네트워크 스트림으로 반환될 때의 플로우에 대한 전달 동작을 정의하려면 720 페이지 "전달 동작 계획 방법"을 참조하십시오.
- 특정 유형의 트래픽에 대한 플로우 계산을 계획하려면 722 페이지 "플로우 계산 계획 방법"을 참조하십시오.
- QoS 정책에 다른 클래스를 추가하려면 713 페이지 "QoS 정책에 대한 클래스 정의 방법"을 참조하십시오.
- QoS 정책에 다른 필터를 추가하려면 716 페이지 "QoS 정책에서 필터를 정의하는 방법"을 참조하십시오.

▼ 플로우 제어 계획 방법

플로우 제어 과정에서는 클래스에 대한 트래픽 플로우가 측정되고 정의된 속도로 패킷이 네트워크로 릴리스됩니다. 플로우 제어를 계획할 때 IPQoS 측정 모듈에 사용할 매개변수를 정의합니다. 측정기는 트래픽이 네트워크로 릴리스되는 속도를 결정합니다. 측정 모듈 소개는 700 페이지 "측정기(tokenmt 및 tswtclmt) 개요"를 참조하십시오.

다음 절차에서는 716 페이지 "QoS 정책에서 필터를 정의하는 방법"에 설명된 대로 필터 및 선택기를 정의했다고 가정합니다.

1. **네트워크에 대한 최대 대역폭을 확인합니다.**

2. **네트워크에서 지원되는 SLA를 검토합니다. 고객 및 각 고객에게 보장되는 서비스의 유형을 식별합니다.**

   특정 레벨의 서비스를 보장하려면 고객이 생성한 특정 트래픽 클래스를 측정해야 할 수도 있습니다.

3. **713 페이지 "QoS 정책에 대한 클래스 정의 방법"에서 만든 클래스 목록을 검토합니다.**

   SLA와 연관된 클래스 이외의 다른 클래스를 측정해야 할지 여부를 결정합니다.

IPQoS 시스템이 높은 레벨의 트래픽을 생성하는 응용 프로그램을 실행한다고 가정합니다. 응용 프로그램의 트래픽을 분류한 후 플로우를 측정하여 플로우의 패킷이 네트워크로 반환되는 속도를 제어합니다.

주 – 모든 클래스를 측정해야 하는 것은 아닙니다. 클래스 목록을 검토할 때 이 지침을 염두에 두십시오.

---

4 플로우 제어가 필요한 트래픽을 선택하는 각 클래스의 필터를 결정합니다. 그런 다음 측정이 필요한 클래스 목록을 세분화합니다.

필터가 두 개 이상인 클래스의 경우 하나의 필터에 대해서만 측정해야 합니다. 특정 클래스의 수신 및 송신 트래픽에 대한 필터를 정의한 것으로 가정합니다. 한 방향의 트래픽만 플로우 제어가 필요한 것으로 결론지을 수 있습니다.

5 플로우를 제어할 각 클래스에 대한 측정기 모듈을 선택합니다.

QoS 계획 테이블의 측정기 열에 모듈 이름을 추가합니다.

6 구조적 테이블에 측정할 각 클래스에 대한 속도를 추가합니다.

`tokenmt` 모듈을 사용할 경우 다음 속도(비트/초)를 정의해야 합니다.

- 커밋 속도
- 최고 속도

이러한 속도가 특정 클래스를 측정하기에 충분할 경우 `tokenmt`에 대한 커밋 속도 및 커밋 버스트만 정의할 수 있습니다.

필요한 경우 다음 속도도 정의할 수 있습니다.

- 커밋 버스트
- 최고 버스트

`tokenmt` 속도에 대한 전체 정의는 771 페이지 “두 속도 측정기로 tokenmt 구성”을 참조하십시오. `tokenmt(7ipp)` 매뉴얼 페이지에서도 자세한 내용을 확인할 수 있습니다.

`tswtclmt` 모듈을 사용할 경우 다음 속도(비트/초)를 정의해야 합니다.

- 커밋 속도
- 최고 속도

창 크기(밀리초)도 정의할 수 있습니다. 이러한 속도는 772 페이지 “tswtclmt 측정 모듈” 및 `tswtclmt(7ipp)` 매뉴얼 페이지에 정의되어 있습니다.

7 측정된 트래픽에 대한 트래픽 준수 결과를 추가합니다.

두 측정 모듈의 결과는 녹색, 빨간색 및 노란색입니다. 정의한 속도에 적용되는 트래픽 준수 결과를 QoS 구조적 테이블에 추가합니다. 측정기 결과는 769 페이지 “측정기 모듈”에서 자세히 설명됩니다.

커밋 속도를 준수하는 트래픽 또는 준수하지 않는 트래픽에 대해 수행해야 할 작업을 결정해야 합니다. 항상은 아니지만 이 작업은 패킷 헤더에 홉당 동작을 표시하는 경우가 많습니다. 트래픽 플로우가 커밋 속도를 초과하지 않는 상태에서 녹색 레벨의 트래픽에 대해 허용 가능한 작업 중 하나는 처리를 계속하는 것일 수 있습니다. 플로우가 최고 속도를 초과할 경우 클래스의 패킷을 삭제하는 작업을 수행할 수도 있습니다.

예 30–2 측정기 정의

다음 표는 전자 메일 트래픽의 클래스에 대한 측정기 항목을 보여 주는 예입니다. IPQoS 시스템이 있는 네트워크의 총 대역폭은 100메가비트/초 또는 10000000비트/초입니다. QoS 정책은 전자 메일 클래스에 낮은 우선 순위를 지정합니다. 또한 이 클래스는 최상의 전달 동작을 수신합니다.

클래스	우선 순위	필터	선택기	속도
email	8	mail_in	daddr10.50.50.5 dport imap direction LOCAL_IN	
email	8	mail_out	saddr10.50.50.5 sport imap direction LOCAL_OUT	meter=tokenmt 커밋 속도=5000000 커밋 버스트=5000000 최고 속도=10000000 최고 버스트=1000000 녹색 우선 순위=처리 계속 노란색 우선 순위=노란색 PHB 표시 빨간색 우선 순위=삭제

참조

- 패킷이 네트워크 스트림으로 반환될 때의 플로우에 대한 전달 동작을 정의하려면 720 페이지 “전달 동작 계획 방법”을 참조하십시오.
- 특정 유형의 트래픽에 대한 플로우 계산을 계획하려면 722 페이지 “플로우 계산 계획 방법”을 참조하십시오.
- QoS 정책에 다른 클래스를 추가하려면 713 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
- QoS 정책에 다른 필터를 추가하려면 716 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.

- 다른 플로우 제어 체계를 정의하려면 717 페이지 “플로우 제어 계획 방법”을 참조하십시오.
- IPQoS 구성 파일을 만들려면 731 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

▼ 전달 동작 계획 방법

전달 동작에 따라 네트워크로 전달될 트래픽 흐름의 우선 순위 및 삭제 우선 순위가 결정됩니다. 두 가지 주요 전달 동작(다른 트래픽 클래스와 관계가 있는 클래스의 흐름 우선 순위 설정 또는 전체 흐름 삭제)을 선택할 수 있습니다.

Diffserv 모델은 표시자를 사용하여 선택된 전달 동작을 트래픽 흐름에 지정합니다. IPQoS는 다음 표시자 모듈을 제공합니다.

- `dscpmk` – IP 패킷의 DS 필드에 DSCP를 표시하는 데 사용됩니다.
- `dlcosmk` – 데이터그램의 VLAN 태그에 CoS(class-of-service) 값을 표시하는 데 사용됩니다.

주 – 이 절의 제안 사항은 IP 패킷에 해당하는 것입니다. IPQoS 시스템에 VLAN 장치가 포함된 경우 `dlcosmk` 표시자를 사용하여 데이터그램에 대한 전달 동작을 표시할 수 있습니다. 자세한 내용은 775 페이지 “VLAN 장치에서 dlcosmk 표시기 사용”을 참조하십시오.

IP 트래픽의 우선 순위를 설정하려면 각 패킷에 DSCP를 지정해야 합니다. `dscpmk` 표시자는 패킷의 DS 필드에 DSCP를 표시합니다. 전달 동작 유형과 연관된 잘 알려진 코드점 그룹에서 클래스에 대한 DSCP를 선택합니다. 이러한 잘 알려진 코드점은 EF PHB의 경우 46(101110)이며 AF PHB의 경우 코드점 범위입니다. DSCP 및 전달에 대한 개요 정보는 703 페이지 “IPQoS 사용 네트워크에서 트래픽 전달”을 참조하십시오.

시작하기 전에 다음 단계에서는 QoS 정책에 대한 클래스 및 필터를 정의했다고 가정합니다. 측정기와 표시자를 함께 사용하여 트래픽을 제어하는 경우가 많기는 하지만 표시자만으로도 전달 동작을 정의할 수 있습니다.

1. **지금까지 만든 클래스 및 각 클래스에 지정한 우선 순위를 검토합니다.**

   모든 트래픽 클래스를 표시해야 하는 것은 아닙니다.

2. **우선 순위가 가장 높은 클래스에 EF 홉당 동작을 지정합니다.**

   EF PHB는 EF DSCP 46(101110)이 지정된 패킷이 AF PHB가 지정된 패킷보다 먼저 네트워크에 릴리스되도록 합니다. 우선 순위가 가장 높은 트래픽에 EF PHB를 사용합니다. EF에 대한 자세한 내용은 773 페이지 “EF(빠른 전달) PHB”를 참조하십시오.

3. **측정할 트래픽이 있는 클래스에 전달 동작을 지정합니다.**

4 **클래스에 지정한 우선 순위에 따라 나머지 클래스에 DS 코드점을 지정합니다.**

예 30–3 게임 응용 프로그램에 대한 QoS 정책

일반적으로 트래픽은 다음 이유로 측정됩니다.

- 네트워크 사용량이 많을 때 SLA가 이 클래스의 패킷에 대한 서비스 레벨을 보장합니다.
- 우선 순위가 보다 낮은 클래스가 네트워크의 혼잡을 야기할 수 있습니다.

표시자와 측정기를 함께 사용하여 이러한 클래스에 차별화된 서비스 및 대역폭 관리를 제공합니다. 예를 들어, 다음 표에서는 QoS 정책의 일부를 보여 줍니다. 이 정책은 높은 레벨의 트래픽을 생성하는 많이 사용되는 게임 응용 프로그램에 대한 클래스를 정의합니다.

클래스	우선 순위	필터	선택기	속도	전달 여부
games_app	9	games_in	sport 6080	해당 없음	해당 없음
games_app	9	games_out	dport 6081	meter=tokenmt	녹색=AF31
				커밋 속도=5000000	노란색=AF42
				커밋 버스트=5000000	빨간색=삭제
				최고 속도=10000000	
				최고 버스트=15000000	
				녹색 우선 순위=처리 계속	
				노란색 우선 순위=노란색 PHB 표시	
				빨간색 우선 순위=삭제	

전달 동작은 커밋 속도를 준수하거나 최고 속도에 미치지 않는 games_app 트래픽에 우선 순위가 낮은 DSCP를 지정합니다. games_app 트래픽이 최고 속도를 초과하면 QoS 정책은 games_app의 패킷이 삭제되도록 합니다. 모든 AF 코드점은 표 34–2에 나와 있습니다.

참조
- 특정 유형의 트래픽에 대한 플로우 계산을 계획하려면 722 페이지 “플로우 계산 계획 방법”을 참조하십시오.
- QoS 정책에 다른 클래스를 추가하려면 713 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.

- QoS 정책에 다른 필터를 추가하려면 716 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.
- 플로우 제어 체계를 정의하려면 717 페이지 “플로우 제어 계획 방법”을 참조하십시오.
- 패킷이 네트워크 스트림으로 반환될 때의 플로우에 대한 추가 전달 동작을 정의하려면 720 페이지 “전달 동작 계획 방법”을 참조하십시오.
- IPQoS 구성 파일을 만들려면 731 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

▼ 플로우 계산 계획 방법

IPQoS `flowacct` 모듈을 사용하여 청구 또는 네트워크 관리 용도로 트래픽 플로우를 추적할 수 있습니다. 다음 절차에 따라 QoS 정책에 플로우 계산이 포함되어야 할지 여부를 결정하십시오.

1 회사에서 고객에게 SLA를 제공합니까?

그럴 경우 플로우 계산을 사용해야 합니다. SLA를 검토하여 회사에서 고객에게 청구할 네트워크 트래픽의 유형을 결정합니다. 그런 다음 QoS 정책을 검토하여 청구할 트래픽을 선택하는 클래스를 결정합니다.

2 네트워크 문제가 발생하지 않도록 모니터링하거나 테스트해야 할 응용 프로그램이 있습니까?

있을 경우 플로우 계산을 사용하여 이러한 응용 프로그램의 동작을 관찰하는 것이 좋습니다. QoS 정책을 검토하여 모니터링해야 할 트래픽에 지정한 클래스를 확인합니다.

3 QoS 계획 테이블에서 플로우 계산이 필요한 각 클래스에 대해 플로우 계산 열에 Y를 표시합니다.

참조

- QoS 정책에 다른 클래스를 추가하려면 713 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
- QoS 정책에 다른 필터를 추가하려면 716 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.
- 플로우 제어 체계를 정의하려면 717 페이지 “플로우 제어 계획 방법”을 참조하십시오.
- 패킷이 네트워크 스트림으로 반환될 때의 플로우에 대한 전달 동작을 정의하려면 720 페이지 “전달 동작 계획 방법”을 참조하십시오.
- 특정 유형의 트래픽에 대한 추가 플로우 계산을 계획하려면 722 페이지 “플로우 계산 계획 방법”을 참조하십시오.
- IPQoS 구성 파일을 만들려면 731 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

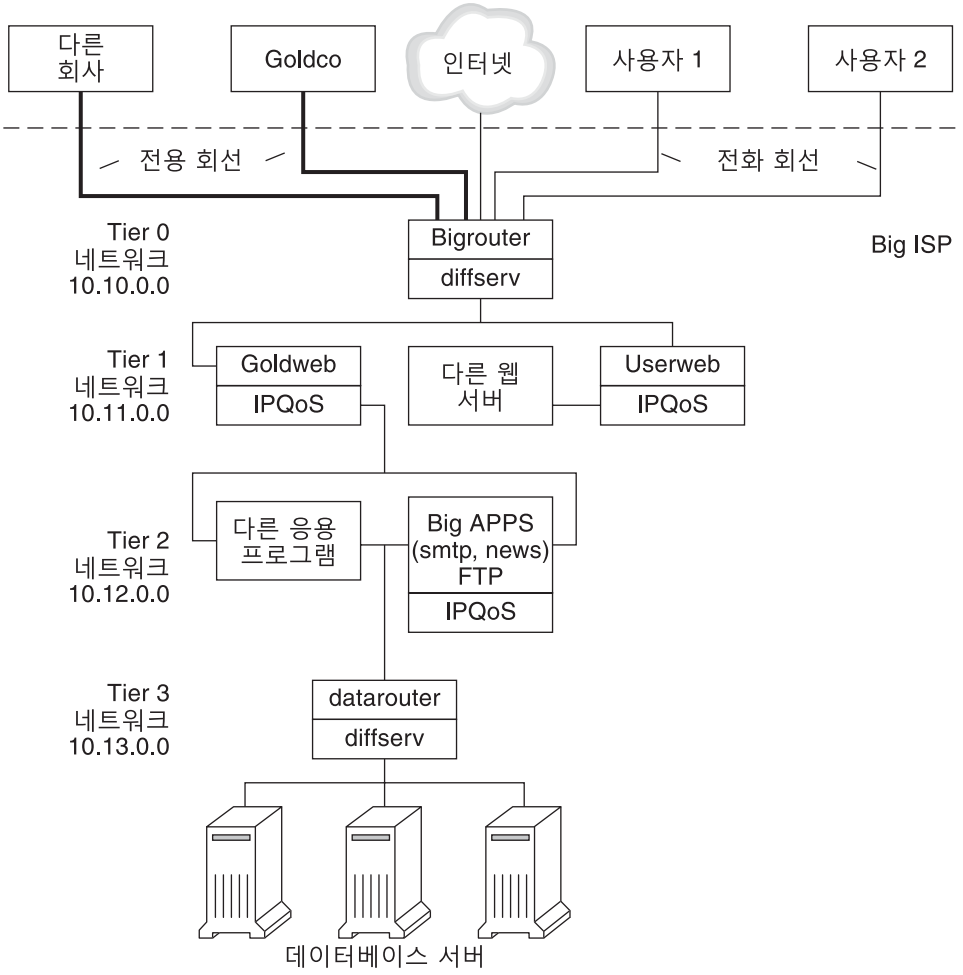
IPQoS 구성 예 소개

본 설명서의 나머지 장에 나오는 작업에서는 이 절에 소개된 IPQoS 구성 예를 사용합니다. 예에서는 가상 서비스 제공업체인 BigISP의 공용 인트라넷에 있는 차별화된 서비스 솔루션을 보여 줍니다. BigISP는 전용 회선을 통해 BigISP에 연결하는 대규모 회사에 서비스를 제공합니다. 모뎀을 통한 전화 접속을 사용하는 개인도 BigISP에서 서비스를 구입할 수 있습니다.

IPQoS 토폴로지

다음 그림에서는 BigISP의 공용 인트라넷에 사용되는 네트워크 토폴로지를 보여 줍니다.

그림 30–4 IPQoS 예 토폴로지



BigISP는 공용 인트라넷에서 다음 네 계층을 구현했습니다.

- **Tier 0** – `10.10.0.0` 네트워크에는 외부 인터페이스와 내부 인터페이스가 모두 있는 `Bigrouter`라는 큰 Diffserv 라우터가 있습니다. Goldco라는 대규모 조직을 비롯하여 여러 회사가 `Bigrouter`에서 종료되는 전용 회선 서비스를 임대했습니다. Tier 0은 전화 회선 또는 ISDN을 통해 연결하는 개인 고객도 처리합니다.
- **Tier 1** – `10.11.0.0` 네트워크는 웹 서비스를 제공합니다. `Goldweb` 서버는 Goldco가 BigISP로부터 구매한 고급 서비스에 포함된 웹 사이트를 호스팅합니다. `Userweb` 서버는 개인 고객이 구매한 작은 웹 사이트를 호스팅합니다. `Goldweb`과 `Userweb`에는 모두 IPQoS가 사용됩니다.

- **Tier 2** – `10.12.0.0` 네트워크는 모든 고객에게 사용할 응용 프로그램을 제공합니다. 애플리케이션 서버 중 하나인 `BigAPPS`에는 IPQoS가 사용됩니다. `BigAPPS`는 SMTP, 뉴스 및 FTP 서비스를 제공합니다.
- **Tier 3** – `10.13.0.0` 네트워크는 큰 데이터베이스 서버를 다룹니다. Tier 3에 대한 액세스는 Diffserv 라우터인 `datarouter`를 통해 제어됩니다.

◆ ◆ ◆ 31장

31 IPQoS 구성 파일 만들기(작업)

이 장에서는 IPQoS 구성 파일을 만드는 방법을 설명합니다. 이 장에서는 다음 항목을 다룹니다.

이 장에서는 완전한 QoS 정책이 정의되어 있고, 이 정책을 IPQoS 구성 파일에 대한 기준으로 사용할 준비가 되어 있다고 가정합니다. QoS 정책 계획에 대한 자세한 내용은 711 페이지 "서비스 품질 정책 계획"을 참조하십시오.

IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)

이 작업 맵에서는 IPQoS 구성 파일을 만들기 위한 일반적인 작업을 나열하고 작업 수행 단계를 설명하는 각 절에 대한 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS 사용 네트워크 구성을 계획합니다.	로컬 시스템에서 IPQoS 사용 시스템이 되어야 하는 시스템을 결정합니다.	713 페이지 "네트워크에서 IPQoS를 준비하는 방법"
2. 네트워크에서 IPQoS 시스템에 대한 QoS 정책을 계획합니다.	트래픽 흐름을 고유의 서비스 클래스로 식별합니다. 그런 다음 트래픽 관리가 필요한 흐름을 결정합니다.	711 페이지 "서비스 품질 정책 계획"
3. IPQoS 구성 파일을 만들고 첫번째 작업을 정의합니다.	IPQoS 파일을 만들고 IP 분류기를 호출한 다음 처리할 클래스를 정의합니다.	731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"

작업	설명	수행 방법
4. 클래스에 대한 필터를 만듭니다.	어떤 클래스가 선택되고, 클래스로 구성되는지 제어하는 필터를 추가합니다.	733 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”
5. 더 많은 클래스와 필터를 IPQoS 구성 파일에 추가합니다.	IP 분류기로 처리할 더 많은 클래스와 필터를 만듭니다.	739 페이지 “최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법”
6. 측정 모듈을 구성하는 매개변수와 함께 `action` 명령문을 추가합니다.	QoS 정책에서 흐름 제어를 요구하는 경우 흐름 제어 속도 및 준수 레벨을 측정기에 지정합니다.	748 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”
7. 표시기를 구성하는 매개변수와 함께 `action` 명령문을 추가합니다.	QoS 정책에서 차별화된 전달 동작을 요구하는 경우 트래픽 클래스가 전달되는 방식을 정의합니다.	735 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”
8. 흐름 계산 모듈을 구성하는 매개변수와 함께 `action` 명령문을 추가합니다.	QoS 정책에서 트래픽 흐름에 대한 통계 수집을 요구하는 경우 계산 통계가 수집되는 방식을 정의합니다.	738 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”
9. IPQoS 구성 파일을 적용합니다.	지정된 IPQoS 구성 파일의 내용을 해당하는 커널 모듈에 추가합니다.	754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”
10. 라우터 파일에서 전달 동작을 구성합니다.	네트워크의 IPQoS 구성 파일에서 전달 동작을 정의하는 경우 결과 DSCP를 라우터의 해당하는 일정 파일에 추가합니다.	751 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”

QoS 정책을 만들기 위한 도구

네트워크에 대한 QoS 정책은 IPQoS 구성 파일에 상주합니다. 이 구성 파일은 텍스트 편집기를 사용하여 만듭니다. 그런 다음 파일을 `ipqosconf`(IPQoS 구성 유틸리티)에 인수로 제공합니다. `ipqosconf`가 구성 파일에서 정의된 정책을 적용하도록 지시하면 정책이 커널 IPQoS 시스템에 쓰여집니다. `ipqosconf` 명령에 대한 자세한 내용은 `ipqosconf`(1M) 매뉴얼 페이지를 참조하십시오. `ipqosconf` 사용에 대한 자세한 내용은 754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.

기본 IPQoS 구성 파일

IPQoS 구성 파일은 711 페이지 "서비스 품질 정책 계획"에서 정의한 QoS 정책을 구현하는 `action` 명령문 트리로 구성됩니다. IPQoS 구성 파일은 IPQoS 모듈을 구성합니다. 각 작업 명령문에는 작업 명령문에서 호출된 모듈로 처리될 **클래스**, **필터** 또는 **매개변수** 세트가 포함됩니다.

IPQoS 구성 파일의 전체 구문은 예 34–3 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

IPQoS 예제 토폴로지 구성

이 장의 작업에서는 세 IPQoS 사용 시스템에 대한 IPQoS 구성 파일을 만드는 방법을 설명합니다. 이러한 시스템은 그림 30–4에 소개된 BigISP 회사의 네트워크 토폴로지에 속합니다.

- `Goldweb` – 프리미엄 레벨 SLA를 구매한 고객을 위한 웹 사이트를 호스트하는 웹 서버입니다.
- `Userweb` – "최선 조건" SLA를 구매한 가정 사용자를 위한 개인용 웹 사이트를 호스트하는 덜 강력한 웹 서버입니다.
- `BigAPPS` – 골드 레벨 및 최선 조건 고객을 위한 메일, 네트워크 뉴스 및 FTP를 서비스하는 애플리케이션 서버입니다.

이러한 세 구성 파일은 가장 일반적인 IPQoS 구성을 보여줍니다. 다음 절에 나오는 샘플 파일을 고유의 IPQoS 구현을 위한 템플리트로 사용할 수 있습니다.

웹 서버에 대한 IPQoS 구성 파일 만들기

이 절에서는 프리미엄 웹 서버에 대한 구성을 만드는 방법을 통해 IPQoS 구성 파일을 소개합니다. 그런 다음 개인용 웹 사이트를 호스트하는 서버에 대한 다른 구성 파일에서 완전히 다른 레벨의 서비스를 구성하는 방법을 보여줍니다. 두 서버는 그림 30–4에 나온 네트워크 예의 일부입니다.

다음 구성 파일은 `Goldweb` 서버에 대한 IPQoS 작업을 정의합니다. 이 서버는 프리미엄 SLA를 구매한 회사인 Goldco에 대한 웹 사이트를 호스트합니다.

예 31–1 프리미엄 웹 서버에 대한 샘플 IPQoS 구성 파일

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
```

예 31–1 프리미엄 웹 서버에 대한 샘플 IPQoS 구성 파일 *(계속)*

```
    class {
        name goldweb
        next_action markAF11
        enable_stats FALSE
    }
    class {
        name video
        next_action markEF
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class goldweb
    }
    filter {
        name videoout
        sport videosrv
        direction LOCAL_OUT
        class video
    }
}
action {
    module dscpmk
    name markAF11
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}
action {
    module dscpmk
    name markEF
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}
action {
    module flowacct
    name acct
    params {
        enable_stats TRUE
        timer 10000
        timeout 10000
        max_limit 2048
    }
}
```

다음 구성 파일은 Userweb에 대한 IPQoS 작업을 정의합니다. 이 서버는 낮은 가격 또는 **최선 조건** SLA의 개인을 위한 웹 사이트를 호스트합니다. 이 레벨의 서비스는 IPQoS 시스템에서 더 높은 가격 SLA의 고객 트래픽을 처리한 후 최선 조건 고객에게 제공할 수 있는 최상의 서비스를 보장합니다.

예 31–2 최선 조건 웹 서버에 대한 샘플 구성

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name Userweb
        next_action markAF12
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class Userweb
   }
}
action {
    module dscpmk
    name markAF12
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}
```

▼ IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법

유지 관리하기 가장 쉬운 디렉토리에서 첫번째 IPQoS 구성 파일을 만들 수 있습니다. 이 장의 작업에서는 IPQoS 구성 파일에 대한 위치로 /var/ipqos 디렉토리를 사용합니다. 다음 절차에서는 예 31–1에 소개된 IPQoS 구성 파일의 초기 세그먼트를 만듭니다.

주 – IPQoS 구성 파일을 만들 때 각 action 문과 절을 중괄호({})로 묶는 경우 주의하십시오. 중괄호 사용 예는 예 31–1을 참조하십시오.

---

1 프리미엄 웹 서버에 로그인하고 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

모든 IPQoS 구성 파일은 버전 번호 fmt_version 1.0이 첫번째 주석 처리되지 않은 행으로 시작되어야 합니다.

2 일반 IP 분류기 ipgpc를 구성하는 초기 action 명령문에서 여는 매개변수를 따릅니다.

이 초기 작업은 IPQoS 구성 파일을 구성하는 action 명령문 트리를 시작합니다. 예를 들어, /var/ipqos/Goldweb.qos 파일은 ipgpc 분류기를 호출하는 초기 action 명령문으로 시작됩니다.

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
```

fmt_version 1.0	IPQoS 구성 파일을 시작합니다.
action {	작업 명령문을 시작합니다.
module ipgpc	ipgpc 분류기를 구성 파일의 첫번째 작업으로 구성합니다.
name ipgpc.classify	항상 ipgpc.classify가 되어야 하는 분류기 action 명령문의 이름을 정의합니다.

action 명령문에 대한 자세한 구문 정보는 780 페이지 “action 명령문” 및 ipqosconf(1M) 매뉴얼 페이지를 참조하십시오.

3 통계 매개변수 global_stats와 함께 params 절을 추가합니다.

```
params {
        global_stats TRUE
   }
```

ipgpc.classify 명령문의 global_stats TRUE 매개변수는 해당 작업에 대한 통계 수집을 사용으로 설정합니다. 또한 global_stats TRUE는 클래스 절 정의에서 enable_stats TRUE를 지정할 때마다 클래스별 통계 수집을 사용으로 설정합니다.

통계를 설정하면 성능이 영향을 받습니다. 새 IPQoS 구성 파일에 대한 통계를 수집하여 IPQoS가 제대로 작동하는지 확인할 수 있습니다. 나중에 global_stats 인수를 FALSE로 변경하여 통계 수집을 해제할 수 있습니다.

전역 통계는 params 절에서 정의할 수 있는 유일한 매개변수 유형입니다. params 절에 대한 구문 및 기타 자세한 내용은 782 페이지 “params 절” 및 ipqosconf(1M) 매뉴얼 페이지를 참조하십시오.

4 프리미엄 서버로 향하는 트래픽을 식별하는 클래스를 정의합니다.

```
class {
        name goldweb
        next_action markAF11
        enable_stats FALSE
    }
```

이 명령문을 **클래스 절**이라고 합니다. class 절에는 다음과 같은 내용이 있습니다.

name goldweb	Goldweb 서버로 향하는 트래픽을 식별하는 goldweb 클래스를 만듭니다.

next_action markAF11	ipgpc 모듈이 goldweb 클래스의 패킷을 markAF11 작업 명령문에 전달하도록 지시합니다. markAF11 작업 명령문은 dscpmk 표시기를 호출합니다.
enable_stats FALSE	goldweb 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계는 설정되지 않습니다.

class 절의 구문에 대한 자세한 내용은 781 페이지 "class 절" 및 ipqosconf(1M)를 참조하십시오.

5 가장 높은 우선 순위 전달을 가져야 하는 응용 프로그램을 식별하는 클래스를 정의합니다.

```
class {
        name video
        next_action markEF
        enable_stats FALSE
    }
```

name video	Goldweb 서버에서 나가는 스트리밍 비디오 트래픽을 식별하는 video 클래스를 만듭니다.
next_action markEF	ipgpc가 처리를 완료한 후 ipgpc 모듈이 video 클래스의 패킷을 markEF 명령문에 전달하도록 지시합니다. markEF 명령문은 dscpmk 표시기를 호출합니다.
enable_stats FALSE	video 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계 수집은 설정되지 않습니다.

참조
- 방금 만든 클래스에 대한 필터를 정의하려면 733 페이지 "IPQoS 구성 파일에서 필터를 정의하는 방법"을 참조하십시오.
- 구성 파일에 대한 다른 클래스 절을 만들려면 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"을 참조하십시오.

▼ IPQoS 구성 파일에서 필터를 정의하는 방법

다음 절차에서는 IPQoS 구성 파일에서 클래스에 대한 필터를 정의하는 방법을 보여줍니다.

시작하기 전에 이 절차에서는 이미 파일 만들기를 시작하고 클래스를 정의했다고 가정합니다. 단계는 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"에서 만든 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

주 – IPQoS 구성 파일을 만들 때 각 `class` 절 및 각 `filter` 절을 중괄호({})로 묶는 경우 주의하십시오. 중괄호 사용 예는 예 31–1을 참조하십시오.

1 IPQoS 구성 파일을 열고 정의한 마지막 클래스의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 `Goldweb`에서 `/var/ipqos/Goldweb.qos`의 다음 `class` 절 이후 시작할 수 있습니다.

```
class {
        name video
        next_action markEF
        enable_stats FALSE
    }
```

2 IPQoS 시스템의 송신 트래픽을 선택하는 `filter` 절을 정의합니다.

```
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class goldweb
    }
```

`name webout`	`webout` 이름을 필터에 제공합니다.
`sport 80`	HTTP(웹) 트래픽에 대해 잘 알려진 포트인 소스 포트 80의 트래픽을 선택합니다.
`direction LOCAL_OUT`	로컬 시스템의 송신 트래픽을 추가로 선택합니다.
`class goldweb`	필터가 속한 클래스(이 경우 `goldweb` 클래스)를 식별합니다.

IPQoS 구성 파일의 `filter` 절에 대한 구문 및 자세한 내용은 782 페이지 “`filter` 절”을 참조하십시오.

3 IPQoS 시스템에서 스트리밍 비디오 트래픽을 선택하는 `filter` 절을 정의합니다.

```
    filter {
        name videoout
        sport videosrv
        direction LOCAL_OUT
        class video
    }
```

name videoout	`videoout` 이름을 필터에 제공합니다.
`sport videosrv`	이 시스템의 스트리밍 비디오 응용 프로그램에 대해 이전에 정의한 포트인 소스 포트 `videosrv`의 트래픽을 선택합니다.
`direction LOCAL_OUT`	로컬 시스템의 송신 트래픽을 추가로 선택합니다.
`class video`	필터가 속한 클래스(이 경우 `video` 클래스)를 식별합니다.

참조
- 표시기 모듈에 대한 전달 동작을 정의하려면 735 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”을 참조하십시오.
- 측정 모듈에 대한 흐름 제어 매개변수를 정의하려면 748 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.
- IPQoS 구성 파일을 활성화하려면 754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
- 추가 필터를 정의하려면 733 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
- 응용 프로그램의 트래픽 플로우에 대한 클래스를 만들려면 744 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법

다음 절차에서는 IPQoS 구성 파일에 클래스에 대한 홉별 동작을 추가하여 트래픽 전달을 정의하는 방법을 보여줍니다.

시작하기 전에 이 절차에서는 이미 정의된 클래스와 필터가 있는 기존 IPQoS 구성 파일이 있다고 가정합니다. 단계는 예 31–1에서 `/var/ipqos/Goldweb.qos` 파일 만들기를 계속합니다.

주 – 이 절차에서는 `dscpmk` 표시기 모듈을 사용하여 트래픽 전달을 구성하는 방법을 보여줍니다. `dlclosmk` 표시기를 사용하여 VLAN 시스템에서 트래픽 전달에 대한 자세한 내용은 775 페이지 “VLAN 장치에서 `dlcosmk` 표시기 사용”을 참조하십시오.

---

1 IPQoS 구성 파일을 열고 정의한 마지막 필터의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 `Goldweb`에서 `/var/ipqos/Goldweb.qos`의 다음 `filter` 절 이후 시작할 수 있습니다.

```
filter {
        name videoout
        sport videosrv
        direction LOCAL_OUT
        class video
    }
}
```

이 `filter` 절은 `ipgpc` 분류기 `action` 명령문의 끝에 있습니다. 그러므로 필터를 종료하는 닫는 중괄호와 `action` 명령문을 종료하는 두번째 닫는 중괄호가 필요합니다.

2 다음 `action` 명령문으로 표시기를 호출합니다.

```
action {
    module dscpmk
    name markAF11
```

module dscpmk 표시기 모듈 `dscpmk`를 호출합니다.

name markAF11 `markAF11` 이름을 `action` 명령문에 제공합니다.

이전에 정의한 클래스 `goldweb`에는 `next_action` `markAF11` 명령문이 포함되어 있습니다. 이 명령문은 분류기가 처리를 완료한 후 트래픽 흐름을 `markAF11` 작업 명령문에 보냅니다.

3 표시기가 트래픽 흐름에 대해 수행할 작업을 정의합니다.

```
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}
```

global_stats FALSE `markAF11` 표시기 `action` 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 통계는 수집되지 않습니다.

dscp_map{0–63:10} 표시기에서 현재 처리 중인 트래픽 클래스 `goldweb`의 패킷 헤더에 DSCP `10`을 지정합니다.

next_action continue 트래픽 클래스 `goldweb`의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

DSCP `10`은 표시기가 `dscp` 맵의 모든 항목을 십진수 값 10(이진수 001010)으로 설정하도록 지시합니다. 이 코드 포인트는 `goldweb` 트래픽 클래스의 패킷이 AF11 홉별 동작에 종속된다는 것을 나타냅니다. AF11은 DSCP `10`의 모든 패킷이 낮은 삭제, 높은 우선 순위의 서비스를 받도록 보장합니다. 따라서 `Goldweb`의 프리미엄 고객에 대한 송신 트래픽에는 AF(보장 전달) PHB에 대해 사용 가능한 가장 높은 우선 순위가 제공됩니다. AF에 대해 가능한 DSCP 표는 표 34–2를 참조하십시오.

4 다른 표시기 `action` 명령문을 시작합니다.

```
action {
    module dscpmk
    name markEF
```

module dscpmk 표시기 모듈 `dscpmk`를 호출합니다.

`name markEF` `markEF` 이름을 `action` 명령문에 제공합니다.

5 표시기가 트래픽 흐름에 대해 수행할 작업을 정의합니다.

```
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}
```

`global_stats TRUE` 스트리밍 비디오 패킷을 선택하는 `video` 클래스에 대한 통계 수집을 사용으로 설정합니다.

`dscp_map{0–63:46}` 표시기에서 현재 처리 중인 트래픽 클래스 `video`의 패킷 헤더에 DSCP 46을 지정합니다.

`next_action acct` `dscpmk`가 처리를 완료한 후 `dscpmk` 모듈이 `video` 클래스의 패킷을 `acct action` 명령문에 전달하도록 지시합니다. `acct action` 명령문은 `flowacct` 모듈을 호출합니다.

DSCP 46은 `dscpmk` 모듈이 `dscp` 맵의 모든 항목을 DS 필드에서 십진수 값 46(이진수 101110)으로 설정하도록 지시합니다. 이 코드 포인트는 `video` 트래픽 클래스의 패킷이 EF(빠른 전달) 홉별 동작에 종속된다는 것을 나타냅니다.

주 – EF에 대해 권장되는 코드 포인트는 46(이진수 101110)입니다. 기타 DSCP는 AF PHB를 패킷에 지정합니다.

EF PHB는 DSCP 46의 패킷이 IPQoS 및 Diffserv 인식 시스템에서 가장 높은 우선권을 받도록 보장합니다. 스트리밍 응용 프로그램에는 가장 높은 우선 순위의 서비스가 필요하므로 QoS 정책에서 스트리밍 응용 프로그램에 EF PHB를 지정하게 됩니다. 빠른 전달 PHB에 대한 자세한 내용은 773 페이지 “EF(빠른 전달) PHB”를 참조하십시오.

6 방금 만든 DSCP를 Diffserv 라우터의 해당하는 파일에 추가합니다.

자세한 내용은 751 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.

참조

- 트래픽 흐름에 대한 흐름 계산 통계 수집을 시작하려면 738 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
- 표시기 모듈에 대한 전달 동작을 정의하려면 735 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”을 참조하십시오.
- 측정 모듈에 대한 흐름 제어 매개변수를 정의하려면 748 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.
- IPQoS 구성 파일을 활성화하려면 754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
- 추가 필터를 정의하려면 733 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
- 응용 프로그램의 트래픽 플로우에 대한 클래스를 만들려면 744 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법

다음 절차에서는 IPQoS 구성 파일에서 트래픽 클래스에 대한 계산을 사용으로 설정하는 방법을 보여줍니다. 절차는 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"에 소개된 video 클래스에 대한 플로우 계산을 정의하는 방법을 보여줍니다. 이 클래스는 프리미엄 고객의 SLA의 일부로 청구되어야 하는 스트리밍 비디오 트래픽을 선택합니다.

시작하기 전에 이 절차에서는 이미 정의된 클래스, 필터, 측정 작업(해당하는 경우) 및 표시 작업(해당하는 경우)이 있는 기존 IPQoS 구성 파일이 있다고 가정합니다. 단계는 예 31–1에서 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

1 IPQoS 구성 파일을 열고 정의한 마지막 action 명령문의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 markEF action 명령문 이후 시작할 수 있습니다.

```
action {
    module dscpmk
    name markEF
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}
```

2 흐름 계산을 호출하는 action 명령문을 시작합니다.

```
action {
    module flowacct
    name acct
```

module flowacct — 흐름 계산 모듈 flowacct를 호출합니다.

name acct — acct 이름을 action 명령문에 제공합니다.

3 트래픽 클래스에 대한 계산을 제어하는 params 절을 정의합니다.

```
params {
        global_stats TRUE
        timer 10000
        timeout 10000
        max_limit 2048
        next_action continue
    }
}
```

global_stats TRUE — 스트리밍 비디오 패킷을 선택하는 video 클래스에 대한 통계 수집을 사용으로 설정합니다.

`timer 10000`	시간 초과된 흐름에 대해 흐름 테이블이 검사되는 간격(밀리초)를 지정합니다. 이 매개변수에서 간격은 10000밀리초입니다.
`timeout 10000`	최소 간격 시간 초과 값을 지정합니다. 흐름 패킷이 시간 초과 간격 동안 보이지 않으면 흐름이 "시간 초과"됩니다. 이 매개변수에서 패킷은 10000밀리초 후 시간 초과됩니다.
`max_limit 2048`	이 작업 인스턴스에 대한 흐름 테이블에서 최대 활성 흐름 레코드 수를 설정합니다.
`next_action continue`	트래픽 클래스 `video`의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

`flowacct` 모듈은 지정된 `timeout` 값에 도달할 때까지 특정 클래스의 패킷 흐름에 대한 통계 정보를 수집합니다.

참조
- 라우터에 대한 홉별 동작을 구성하려면 751 페이지 "IPQoS 사용 네트워크에서 라우터를 구성하는 방법"을 참조하십시오.
- IPQoS 구성 파일을 활성화하려면 754 페이지 "IPQoS 커널 모듈에 새 구성을 적용하는 방법"을 참조하십시오.
- 응용 프로그램의 트래픽 플로우에 대한 클래스를 만들려면 744 페이지 "애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법"을 참조하십시오.

▼ 최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법

최선 조건 웹 서버에 대한 IPQoS 구성 파일은 프리미엄 웹 서버에 대한 IPQoS 구성 파일과 약간 다릅니다. 예로 절차에서는 예 31–2의 구성 파일을 사용합니다.

1 최선 조건 웹 서버에 로그인합니다.

2 `.qos` 확장자로 새 IPQoS 구성 파일을 만듭니다.

```
fmt_vesion 1.0
action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
   }
```

/var/ipqos/userweb.qos 파일은 ipgpc 분류기를 호출하는 부분 action 명령문으로 시작되어야 합니다. 또한 action 명령문에는 통계 수집을 설정하는 params 절도 있어야 합니다. 이 action 명령문에 대한 설명은 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"을 참조하십시오.

3 최선 조건 웹 서버로 향하는 트래픽을 식별하는 클래스를 정의합니다.

```
class {
        name userweb
        next_action markAF12
        enable_stats FALSE
    }
```

name userweb	사용자의 웹 트래픽 전달을 위한 userweb이라는 클래스를 만듭니다.
next_action markAF1	ipgpc가 처리를 완료한 후 ipgpc 모듈이 userweb 클래스의 패킷을 markAF12 action 명령문에 전달하도록 지시합니다. markAF12 action 명령문은 dscpmk 표시기를 호출합니다.
enable_stats FALSE	userweb 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계 수집은 발생하지 않습니다.

class 절 작업에 대한 설명은 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"을 참조하십시오.

4 userweb 클래스에 대한 트래픽 흐름을 선택하는 filter 절을 정의합니다.

```
    filter {
       name webout
       sport 80
       direction LOCAL_OUT
       class userweb
   }
}
```

name webout	webout 이름을 필터에 제공합니다.
sport 80	HTTP(웹) 트래픽에 대해 잘 알려진 포트인 소스 포트 80의 트래픽을 선택합니다.
direction LOCAL_OUT	로컬 시스템의 송신 트래픽을 추가로 선택합니다.
class userweb	필터가 속한 클래스(이 경우 userweb 클래스)를 식별합니다.

filter 절 작업에 대한 설명은 733 페이지 "IPQoS 구성 파일에서 필터를 정의하는 방법"을 참조하십시오.

5 dscpmk 표시기를 호출하는 action 명령문을 시작합니다.

```
action {
    module dscpmk
    name markAF12
```

`module dscpmk` 표시기 모듈 `dscpmk`를 호출합니다.

`name markAF12` `markAF12` 이름을 `action` 명령문에 제공합니다.

이전에 정의한 클래스 `userweb`에는 `next_action markAF12` 명령문이 포함되어 있습니다. 이 명령문은 분류기가 처리를 완료한 후 트래픽 흐름을 `markAF12 action` 명령문에 보냅니다.

6 표시기가 트래픽 흐름 처리를 위해 사용할 매개변수를 정의합니다.

```
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}
```

`global_stats FALSE` `markAF12` 표시기 `action` 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 통계 수집은 발생하지 않습니다.

`dscp_map{0–63:12}` 표시기에서 현재 처리 중인 트래픽 클래스 `userweb`의 패킷 헤더에 DSCP 12를 지정합니다.

next_action continue 트래픽 클래스 `userweb`의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

DSCP 12는 표시기가 `dscp` 맵의 모든 항목을 십진수 값 12(이진수 001100)로 설정하도록 지시합니다. 이 코드 포인트는 `userweb` 트래픽 클래스의 패킷이 AF12 홉별 동작에 종속된다는 것을 나타냅니다. AF12는 DS 필드에서 DSCP 12의 모든 패킷이 중간 삭제, 높은 우선 순위의 서비스를 받도록 보장합니다.

7 IPQoS 구성 파일을 완료한 경우 구성을 적용합니다.

참조

- 응용 프로그램의 트래픽 흐름에 대한 클래스 및 기타 구성을 추가하려면 744 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.
- 라우터에 대한 홉별 동작을 구성하려면 751 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.
- IPQoS 구성 파일을 활성화하려면 754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.

애플리케이션 서버에 대한 IPQoS 구성 파일 만들기

이 절에서는 고객에게 주요 응용 프로그램을 제공하는 애플리케이션 서버에 대한 구성 파일을 만드는 방법을 설명합니다. 이 절차에서는 예로 그림 30–4의 BigAPPS 서버를 사용합니다.

다음 구성 파일은 BigAPPS 서버에 대한 IPQoS 작업을 정의합니다. 이 서버는 고객을 위한 FTP, 전자 메일(SMTP) 및 네트워크 뉴스(NNTP)를 호스트합니다.

예 31–3 애플리케이션 서버에 대한 샘플 IPQoS 구성 파일

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name smtp
        enable_stats FALSE
        next_action markAF13
    }
    class {
        name news
        next_action markAF21
    }
    class {
        name ftp
        next_action meterftp
    }
    filter {
        name smtpout
        sport smtp
        class smtp
    }
    filter {
        name newsout
        sport nntp
        class news
    }
    filter {
        name ftpout
        sport ftp
        class ftp
    }
   filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
action {
    module dscpmk
```

예 31–3 애플리케이션 서버에 대한 샘플 IPQoS 구성 파일 *(계속)*

```
    name markAF13
    params {
        global_stats FALSE
        dscp_map{0-63:14}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
action {
    module tokenmt
    name meterftp
    params {
        committed_rate 50000000
        committed_burst 50000000
        red_action_name AF31
        green_action_name markAF22
        global_stats TRUE
    }
}
action {
    module dscpmk
    name markAF31
    params {
        global_stats TRUE
        dscp_map{0-63:26}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF22
    params {
        global_stats TRUE
        dscp_map{0-63:20}
        next_action continue
    }
}
```

▼ 애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법

1 IPQoS 사용 애플리케이션 서버에 로그인하고 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

예를 들어, 애플리케이션 서버에 대해 /var/ipqos/BigAPPS.qos 파일을 만듭니다. 다음 필수 문구로 시작하여 ipgpc 분류기를 호출하는 action 명령문을 시작합니다.

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
```

여는 action 명령문에 대한 설명은 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"을 참조하십시오.

2 BigAPPS 서버에서 세 응용 프로그램의 트래픽을 선택하는 클래스를 만듭니다.

action 명령문을 연 후 클래스 정의를 추가합니다.

```
    class {
        name smtp
        enable_stats FALSE
        next_action markAF13
    }
    class {
        name news
        next_action markAF21
    }
    class {
        name ftp
        enable_stats TRUE
        next_action meterftp
    }
```

name smtp	SMTP 응용 프로그램에서 처리할 전자 메일 트래픽 흐름을 포함하는 smtp라는 클래스를 만듭니다.
enable_stats FALSE	smtp 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계는 수집되지 않습니다.
next_action markAF13	ipgpc가 처리를 완료한 후 ipgpc 모듈이 smtp 클래스의 패킷을 markAF13 action 명령문에 전달하도록 지시합니다.
name news	NNTP 응용 프로그램에서 처리할 네트워크 뉴스 트래픽 흐름을 포함하는 news라는 클래스를 만듭니다.

`next_action markAF21`	`ipgpc`가 처리를 완료한 후 `ipgpc` 모듈이 `news` 클래스의 패킷을 `markAF21` 작업 명령문에 전달하도록 지시합니다.
`name ftp`	FTP 응용 프로그램에서 처리할 송신 트래픽을 처리하는 `ftp`라는 클래스를 만듭니다.
`enable_stats TRUE`	`ftp` 클래스에 대한 통계 수집을 사용으로 설정합니다.
`next_action meterftp`	`ipgpc`가 처리를 완료한 후 `ipgpc` 모듈이 `ftp` 클래스의 패킷을 `meterftp action` 명령문에 전달하도록 지시합니다.

클래스 정의에 대한 자세한 내용은 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"을 참조하십시오.

3 2단계에서 정의한 클래스의 트래픽을 선택하는 `filter` 절을 정의합니다.

```
    filter {
        name smtpout
        sport smtp
        class smtp
    }
    filter {
        name newsout
        sport nntp
        class news
    }
        filter {
        name ftpout
        sport ftp
        class ftp
    }
        filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
```

`name smtpout`	`smtpout` 이름을 필터에 제공합니다.
`sport smtp`	`sendmail`(SMTP) 응용 프로그램에 대해 잘 알려진 포트인 소스 포트 25의 트래픽을 선택합니다.
`class smtp`	필터가 속한 클래스(이 경우 `smtp` 클래스)를 식별합니다.
`name newsout`	`newsout` 이름을 필터에 제공합니다.
`sport nntp`	네트워크 뉴스(NNTP) 응용 프로그램에 대해 잘 알려진 포트 이름인 소스 포트 이름 `nntp`의 트래픽을 선택합니다.
`class news`	필터가 속한 클래스(이 경우 `news` 클래스)를 식별합니다.
`name ftpout`	`ftpout` 이름을 필터에 제공합니다.

`sport ftp`	FTP 트래픽에 대해 잘 알려진 포트 번호인 소스 포트 21의 제어 데이터를 선택합니다.
`name ftpdata`	`ftpdata` 이름을 필터에 제공합니다.
`sport ftp-data`	FTP 데이터 트래픽에 대해 잘 알려진 포트 번호인 소스 포트 20의 트래픽을 선택합니다.
`class ftp`	`ftpout` 및 `ftpdata` 필터가 속한 클래스(이 경우 `ftp` 클래스)를 식별합니다.

참조
- 필터를 정의하려면 733 페이지 "IPQoS 구성 파일에서 필터를 정의하는 방법"을 참조하십시오.
- 응용 프로그램 트래픽에 대한 전달 동작을 정의하려면 746 페이지 "IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법"을 참조하십시오.
- 측정 모듈을 사용하여 플로우 제어를 구성하려면 748 페이지 "IPQoS 구성 파일에서 플로우 제어를 구성하는 방법"을 참조하십시오.
- 플로우 계산을 구성하려면 738 페이지 "IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법"을 참조하십시오.

▼ IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법

다음 절차에서는 응용 프로그램 트래픽에 대한 전달을 구성하는 방법을 보여줍니다. 이 절차에서는 네트워크의 다른 트래픽보다 낮은 우선권을 가질 수 있는 응용 프로그램 트래픽 클래스에 대한 홉별 동작을 정의합니다. 단계는 예 31–3의 `/var/ipqos/BigAPPS.qos` 파일 만들기를 계속합니다.

시작하기 전에 이 절차에서는 표시할 응용 프로그램에 대해 이미 정의된 클래스와 필터가 있는 기존 IPQoS 구성 파일이 있다고 가정합니다.

1 애플리케이션 서버에 대해 만든 IPQoS 구성 파일을 열고 마지막 `filter` 절의 끝을 찾습니다.

`/var/ipqos/BigAPPS.qos` 파일에서 마지막 필터는 다음과 같습니다.

```
 filter {
        name ftpdata
        sport ftp-data
        class ftp
    }
}
```

2 표시기를 다음과 같이 호출합니다.

```
action {
    module dscpmk
    name markAF13
```

`module dscpmk` 표시기 모듈 `dscpmk`를 호출합니다.

`name markAF13` `markAF13` 이름을 `action` 명령문에 제공합니다.

3 전자 메일 트래픽 흐름에 대해 표시할 홉별 동작을 정의합니다.

```
    params {
        global_stats FALSE
        dscp_map{0-63:14}
        next_action continue
    }
}
```

`global_stats FALSE` `markAF13` 표시기 `action` 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 통계는 수집되지 않습니다.

`dscp_map{0–63:14}` 표시기에서 현재 처리 중인 트래픽 클래스 `smtp`의 패킷 헤더에 DSCP 14를 지정합니다.

`next_action continue` 트래픽 클래스 `smtp`의 패킷에 추가 처리가 필요하지 않음을 나타냅니다. 그러면 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있습니다.

DSCP 14는 표시기가 `dscp` 맵의 모든 항목을 십진수 값 14(이진수 001110)로 설정하도록 지시합니다. DSCP 14는 AF13 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 14의 `smtp` 트래픽 클래스 패킷을 표시합니다.

AF13은 DSCP 14의 모든 패킷을 높은 삭제 우선권으로 지정합니다. 하지만 AF13은 클래스 1 우선 순위도 보장하므로 라우터는 대기열에서 나가는 전자 메일 트래픽을 높은 우선 순위로 보장합니다. 가능한 AF 코드 포인트 표는 표 34–2를 참조하십시오.

4 네트워크 뉴스 트래픽에 대한 홉별 동작을 정의하는 표시기 `action` 명령문을 추가합니다.

```
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
```

`name markAF21` `markAF21` 이름을 `action` 명령문에 제공합니다.

`dscp_map{0–63:18}` 표시기에서 현재 처리 중인 트래픽 클래스 `nntp`의 패킷 헤더에 DSCP 18을 지정합니다.

DSCP 18은 표시기가 dscp 맵의 모든 항목을 십진수 값 18(이진수 010010)로 설정하도록 지시합니다. DSCP 18은 AF21 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 18의 news 트래픽 클래스 패킷을 표시합니다.

AF21은 DSCP 18의 모든 패킷이 낮은 삭제 우선권을 받도록 보장하지만 클래스 2 우선순위를 가집니다. 따라서 네트워크 뉴스 트래픽이 삭제될 가능성은 낮습니다.

참조

- 웹 서버에 대한 구성 정보를 추가하려면 731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"을 참조하십시오.
- 측정 모듈을 사용하여 흐름 제어를 구성하려면 748 페이지 "IPQoS 구성 파일에서 플로우 제어를 구성하는 방법"을 참조하십시오.
- 흐름 계산을 구성하려면 738 페이지 "IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법"을 참조하십시오.
- 라우터에 대한 전달 동작을 구성하려면 751 페이지 "IPQoS 사용 네트워크에서 라우터를 구성하는 방법"을 참조하십시오.
- IPQoS 구성 파일을 활성화하려면 754 페이지 "IPQoS 커널 모듈에 새 구성을 적용하는 방법"을 참조하십시오.

▼ IPQoS 구성 파일에서 플로우 제어를 구성하는 방법

특정 트래픽 플로우가 네트워크로 전송되는 속도를 제어하려면 측정기에 대한 매개변수를 정의해야 합니다. IPQoS 구성 파일에서 두 가지 측정기 모듈 tokenmt 또는 tswtclmt를 사용할 수 있습니다.

다음 절차에서는 예 31–3의 애플리케이션 서버에 대한 IPQoS 구성 파일 만들기를 계속합니다. 이 절차에서는 측정기뿐 아니라 측정기 action 명령문 내에서 호출되는 두 표시기 작업도 구성합니다.

시작하기 전에 단계에서는 플로우를 제어할 응용 프로그램에 대한 클래스 및 필터를 이미 정의했다고 가정합니다.

1 애플리케이션 서버에 대해 만든 IPQoS 구성 파일을 엽니다.

/var/ipqos/BigAPPS.qos 파일에서 다음 표시기 작업 이후 시작합니다.

```
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
```

2 ftp 클래스의 트래픽을 흐름 제어하는 측정기 action 명령문을 만듭니다.

```
action {
    module tokenmt
    name meterftp
```

`module tokenmt` tokenmt 측정기를 호출합니다.

`name meterftp` meterftp 이름을 action 명령문에 제공합니다.

3 측정기의 속도를 구성하는 매개변수를 추가합니다.

```
params {
       committed_rate 50000000
       committed_burst 50000000
```

`committed_rate 50000000` ftp 클래스의 트래픽에 전송 속도 50,000,000bps를 지정합니다.

`committed_burst 50000000` ftp 클래스의 트래픽에 버스트 크기 50,000,000비트를 커밋합니다.

tokenmt 매개변수에 대한 설명은 771 페이지 "두 속도 측정기로 tokenmt 구성"을 참조하십시오.

4 트래픽 준수 우선권을 구성하는 매개변수를 추가합니다.

```
    red_action markAF31
    green_action_name markAF22
    global_stats TRUE
    }
}
```

`red_action_name markAF31` ftp 클래스의 트래픽 흐름이 약정된 속도를 초과할 경우 패킷이 markAF31 표시기 action 명령문으로 보내짐을 나타냅니다.

`green_action_name markAF22` ftp의 트래픽 흐름이 약정된 속도를 준수할 경우 패킷이 markAF22 작업 명령문으로 보내짐을 나타냅니다.

`global_stats TRUE` ftp 클래스에 대한 측정 통계를 사용으로 설정합니다.

트래픽 준수에 대한 자세한 내용은 769 페이지 "측정기 모듈"을 참조하십시오.

5 홉별 동작을 ftp 클래스의 비준수 트래픽 흐름에 지정하는 표시기 action 명령문을 추가합니다.

```
action {
    module dscpmk
    name markAF31
    params {
        global_stats TRUE
        dscp_map{0-63:26}
        next_action continue
```

```
    }
}
```

`module dscpmk`	표시기 모듈 `dscpmk`를 호출합니다.
`name markAF31`	`markAF31` 이름을 `action` 명령문에 제공합니다.
`global_stats TRUE`	`ftp` 클래스에 대한 통계를 사용으로 설정합니다.
`dscp_map{0–63:26}`	이 트래픽이 약정된 속도를 초과할 때마다 트래픽 클래스 `ftp`의 패킷 헤더에 DSCP 26을 지정합니다.
`next_action continue`	트래픽 클래스 `ftp`의 패킷에 추가 처리가 필요하지 않음을 나타냅니다. 그러면 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있습니다.

DSCP 26은 표시기가 `dscp` 맵의 모든 항목을 십진수 값 26(이진수 011010)으로 설정하도록 지시합니다. DSCP 26은 AF31 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 26의 `ftp` 트래픽 클래스 패킷을 표시합니다.

AF31은 DSCP 26의 모든 패킷이 낮은 삭제 우선권을 받도록 보장하지만 클래스 3 우선 순위를 가집니다. 따라서 비준수 FTP 트래픽이 삭제될 가능성은 낮습니다. 가능한 AF 코드 포인트 표는 표 34–2를 참조하십시오.

6 약정된 속도를 준수하는 `ftp` 트래픽 흐름에 홉별 동작을 지정하는 표시기 `action` 명령문을 추가합니다.

```
action {
    module dscpmk
    name markAF22
    params {
        global_stats TRUE
        dscp_map{0-63:20}
        next_action continue
    }
}
```

`name markAF22`	`markAF22` 이름을 `marker` 작업에 제공합니다.
`dscp_map{0–63:20}`	`ftp` 트래픽이 구성된 속도를 준수할 때마다 트래픽 클래스 `ftp`의 패킷 헤더에 DSCP `20`을 지정합니다.

DSCP 20은 표시기가 `dscp` 맵의 모든 항목을 십진수 값 20(이진수 010100)으로 설정하도록 지시합니다. DSCP `20`은 AF22 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP `20`의 `ftp` 트래픽 클래스 패킷을 표시합니다.

AF22는 DSCP `20`의 모든 패킷이 클래스 2 우선 순위로 중간 삭제 우선권을 받도록 보장합니다. 따라서 준수하는 FTP 트래픽은 IPQoS 시스템에서 동시에 전송되는 플로우 중에서 중간 삭제 우선권이 보장됩니다. 하지만 라우터는 클래스 1 중간 삭제 우선권 표시 이상의 트래픽 클래스에 더 높은 전달 우선 순위를 제공합니다. 가능한 AF 코드 포인트 표는 표 34–2를 참조하십시오.

7 애플리케이션 서버에 대해 만든 DSCP를 Diffserv 라우터의 해당하는 파일에 추가합니다.

참조
- IPQoS 구성 파일을 활성화하려면 754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
- 웹 서버에 대한 구성 정보를 추가하려면 731 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.
- 흐름 계산을 구성하려면 738 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
- 라우터에 대한 전달 동작을 구성하려면 751 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.

라우터에서 차별화 서비스 제공

진정한 차별화 서비스를 제공하려면 708 페이지 “Diffserv 네트워크에 대한 하드웨어 전략”에 설명된 대로 네트워크 토폴로지에 Diffserv 인식 라우터를 포함시켜야 합니다. 라우터에서 Diffserv를 구성하고 해당 라우터의 파일을 업데이트하기 위한 실제 단계는 이 설명서의 범위를 벗어납니다.

이 절에서는 네트워크의 다양한 IPQoS 사용 시스템 및 Diffserv 라우터 사이에서 전달 정보를 조정하기 위한 일반적인 단계를 설명합니다.

▼ IPQoS 사용 네트워크에서 라우터를 구성하는 방법

다음 절차에서는 그림 30–4의 토폴로지를 예로 사용합니다.

시작하기 전에 다음 절차에서는 이 장의 이전 작업을 수행하여 네트워크에서 IPQoS 시스템을 이미 구성했다고 가정합니다.

1 네트워크의 모든 IPQoS 사용 시스템에 대한 구성 파일을 검토합니다.

2 QoS 다양한 정책에서 사용되는 각 코드 포인트를 식별합니다.

코드 포인트 및 코드 포인트가 적용되는 시스템과 클래스를 나열합니다. 다음 표는 동일한 코드 포인트를 사용했을 수 있는 영역을 나타냅니다. 이 연습은 수용할 수 있습니다. 하지만 동일하게 표시된 클래스의 우선권을 결정하려면 IPQoS 구성 파일에서 다른 조건(예: `precedence` 선택기)을 제공해야 합니다.

예를 들어, 이 장의 절차에서 사용된 샘플 네트워크의 경우 다음 코드 포인트 표를 만들 수 있습니다.

시스템	클래스	PHB	DS 코드 포인트
Goldweb	video	EF	46 (101110)
Goldweb	goldweb	AF11	10 (001010)
Userweb	webout	AF12	12 (001100)
BigAPPS	smtp	AF13	14 (001110)
BigAPPS	news	AF18	18 (010010)
BigAPPS	ftp 준수 트래픽	AF22	20 (010100)
BigAPPS	ftp 비준수 트래픽	AF31	26 (011010)

3 네트워크의 IPQoS 구성 파일에서 코드 포인트를 Diffserv 라우터의 해당하는 파일에 추가합니다.

제공하는 코드 포인트는 라우터의 Diffserv 일정 예약 방식을 구성하는 데 도움이 되어야 합니다. 자세한 내용은 라우터 제조업체의 설명서 및 웹 사이트를 참조하십시오.

◆ ◆ ◆ 32장

32 IPQoS 시작 및 유지 관리(작업)

이 장에는 IPQoS 구성 파일을 활성화하고 IPQoS 관련 이벤트를 기록하기 위한 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

IPQoS 관리(작업 맵)

이 절에서는 Oracle Solaris 시스템에서 IPQoS를 시작하고 유지 관리하기 위한 작업을 나열합니다. 이 작업을 사용하기 전에 727 페이지 “IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)”에 설명된 대로 완성된 IPQoS 구성 파일을 가지고 있어야 합니다.

다음 표에서는 이러한 작업을 나열하고 설명하며 이러한 작업을 완료하는 방법을 자세히 설명하는 링크를 제공합니다.

작업	설명	수행 방법
1. 시스템에서 IPQoS를 구성합니다.	ipqosconf 명령을 사용하여 시스템에서 IPQoS 구성 파일을 활성화합니다.	754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”
2. 각 시스템 부트 후 Oracle Solaris 시작 스크립트가 디버깅된 IPQoS 구성 파일을 적용하도록 합니다.	시스템이 재부트할 때마다 IPQoS 구성이 적용되도록 합니다.	755 페이지 “재부트 때마다 IPQoS 구성이 적용되도록 하는 방법”
3. IPQoS에 대해 syslog 로깅을 사용으로 설정합니다.	항목을 추가하여 IPQoS 메시지의 syslog 로깅을 사용으로 설정합니다.	755 페이지 “부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법”.

작업	설명	수행 방법
4. 발생하는 IPQoS 문제를 해결합니다.	오류 메시지를 사용하여 IPQoS 문제를 해결합니다.	표 32–1의 오류 메시지를 참조하십시오.

IPQoS 구성 적용

`ipqosconf` 명령을 사용하여 IPQoS 구성을 활성화하고 조작합니다.

▼ IPQoS 커널 모듈에 새 구성을 적용하는 방법

`ipqosconf` 명령을 사용하여 IPQoS 구성 파일을 읽고 UNIX 커널에서 IPQoS 모듈을 구성합니다. 다음 절차에서는 729 페이지 "웹 서버에 대한 IPQoS 구성 파일 만들기"에서 만든 `/var/ipqos/Goldweb.qos` 파일을 예로 사용합니다. 자세한 내용은 `ipqosconf`(1M)를 참조하십시오.

1 IPQoS 사용 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, "Solaris Management Console 작업(작업)"을 참조하십시오.

2 새 구성을 적용합니다.

```
# /usr/sbin/ipqosconf -a/var/ipqos/Goldweb.qos
```

`ipqosconf`는 지정된 IPQoS 구성 파일의 정보를 Oracle Solaris 커널의 IPQoS 모듈에 씁니다. 이 예에서는 `/var/ipqos/Goldweb.qos`의 내용이 현재 Oracle Solaris 커널에 적용됩니다.

주 – `-a` 옵션을 사용하여 IPQoS 구성 파일을 적용할 경우 파일의 작업이 현재 세션에 대해서만 활성화됩니다.

---

3 새 IPQoS 구성을 테스트하고 디버깅합니다.

UNIX 유틸리티를 사용하여 IPQoS 동작을 추적하고 IPQoS 구현에 대한 통계를 수집합니다. 이 정보는 구성이 예상한 대로 작동하는지 여부를 결정하는 데 도움이 됩니다.

참조

- IPQoS 모듈이 어떻게 작동하는지에 대한 통계를 보려면 764 페이지 "통계 정보 수집"을 참조하십시오.
- `ipqosconf` 메시지를 기록하려면 755 페이지 "IPQoS 메시지에 대한 `syslog` 로깅 사용"을 참조하십시오.
- 각 부트 후 현재 IPQoS 구성이 적용되도록 하려면 755 페이지 "재부트 때마다 IPQoS 구성이 적용되도록 하는 방법"을 참조하십시오.

▼ 재부트 때마다 IPQoS 구성이 적용되도록 하는 방법

재부트되더라도 IPQoS 구성을 명시적으로 유지해야 합니다. 그렇지 않으면 시스템이 재부트할 때까지만 현재 구성이 적용됩니다. IPQoS가 시스템에서 올바르게 작동하는 경우 다음을 수행하여 재부트되더라도 구성이 유지되도록 하십시오.

1. **IPQoS 사용 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2. **커널 모듈에 IPQoS 구성이 존재하는지 테스트합니다.**

   ```
   # ipqosconf -l
   ```

   구성이 존재하는 경우 `ipqosconf`가 화면에 구성을 표시합니다. 출력을 받지 못할 경우 754 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”에 설명된 대로 구성을 적용합니다.

3. **IPQoS 시스템이 재부트될 때마다 기존 IPQoS 구성이 적용되도록 합니다.**

   ```
   # /usr/sbin/ipqosconf -c
   ```

   `-c` 옵션은 현재 IPQoS 구성이 부트 시 구성 파일 `/etc/inet/ipqosinit.conf`에 다시 나타나도록 합니다.

IPQoS 메시지에 대한 syslog 로깅 사용

IPQoS 부트 시 메시지를 기록하려면 다음 절차에 나온 대로 `/etc/syslog.conf` 파일을 수정해야 합니다.

▼ 부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법

1. **IPQoS 사용 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.**

   기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2. **`/etc/syslog.conf` 파일을 엽니다.**

3. **다음 텍스트를 파일에 최종 항목으로 추가합니다.**

   ```
   user.info                       /var/adm/messages
   ```

열 사이에는 공백 대신 탭을 사용합니다.

이 항목은 IPQoS로 생성되는 모든 부트 시 메시지를 /var/adm/messages 파일에 기록합니다.

4 시스템을 재부트하여 메시지를 적용합니다.

예 32–1 /var/adm/messages의 IPQoS 출력

시스템이 재부트된 후 /var/adm/messages를 보면 출력에 다음과 유사한 IPQoS 로깅 메시지가 포함되어 있습니다.

```
May 14 10:44:33 ipqos-14 ipqosconf: [ID 815575 user.info]
 New configuration applied.
May 14 10:44:46 ipqos-14 ipqosconf: [ID 469457 user.info]
Current configuration saved to init file.
May 14 10:44:55 ipqos-14 ipqosconf: [ID 435810 user.info]
Configuration flushed.
```

또한 IPQoS 시스템의 /var/adm/messages 파일에서 다음과 유사한 IPQoS 오류 메시지도 볼 수 있습니다.

```
May 14 10:56:47 ipqos-14 ipqosconf: [ID 123217 user.error]
 Missing/Invalid config file fmt_version.
May 14 10:58:19 ipqos-14 ipqosconf: [ID 671991 user.error]
No ipgpc action defined.
```

이러한 오류 메시지에 대한 설명은 표 32–1을 참조하십시오.

IPQoS 오류 메시지를 사용하여 문제 해결

이 절에서는 IPQoS로 생성되는 오류 메시지 및 가능한 해결 방법에 대한 표가 포함되어 있습니다.

표 32–1 IPQoS 오류 메시지

오류 메시지	설명	해결 방법
Undefined action in parameter *parameter-name's* action *action-name*	IPQoS 구성 파일에서 *parameter-name*에 지정한 작업 이름이 구성 파일에 존재하지 않습니다.	작업을 만듭니다. 또는 매개변수의 다른 기존 작업을 참조합니다.
action *action-name* involved in cycle	IPQoS 구성 파일에서 *action-name*이 작업 순환의 일부이며, 이는 IPQoS에서 허용되지 않습니다.	작업 순환을 확인합니다. 그런 다음 IPQoS 구성 파일에서 순환 참조 중 하나를 제거합니다.
Action *action-name* isn't referenced by any other actions	비ipgpc 작업 정의가 IPQoS에서 정의된 다른 작업에 의해 참조되지 않으며, 이는 IPQoS에서 허용되지 않습니다.	참조되지 않는 작업을 제거합니다. 또는 다른 작업이 현재 참조되지 않는 작업을 참조하도록 만듭니다.

표 32–1 IPQoS 오류 메시지 *(계속)*

오류 메시지	설명	해결 방법
`Missing/Invalid config file fmt_version`	구성 파일의 형식이 파일의 첫번째 항목으로 지정되지 않았으며, 이는 IPQoS에서 필수입니다.	731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"에 설명된 대로 형식 버전을 추가합니다.
`Unsupported config file format version`	구성 파일에 지정된 형식 버전이 IPQoS에서 지원되지 않습니다.	IPQoS의 Solaris 9 9/02 릴리스로 시작하는 데 필요한 `fmt_version 1.0`으로 형식 버전을 변경합니다.
`No ipgpc action defined.`	구성 파일에서 `ipgpc` 분류기에 대한 작업을 정의하지 않았으며, 이는 IPQoS 필수 사항입니다.	731 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"에 나온 대로 `ipgpc`에 대한 작업을 정의합니다.
`Can't commit a null configuration`	`ipqosconf -c`를 실행하여 구성을 커밋할 때 해당 구성이 비어 있었으며, 이는 IPQoS에서 허용되지 않습니다.	구성 커밋을 시도하기 전에 구성 파일을 적용합니다. 자세한 내용은 754 페이지 "IPQoS 커널 모듈에 새 구성을 적용하는 방법"을 참조하십시오.
`Invalid CIDR mask on line` *line-number*	구성 파일에서 CIDR 마스크를 IP 주소에 대한 유효한 범위를 벗어난 IP 주소의 일부로 사용했습니다.	마스크 값이 IPv4의 경우 1–32 및 IPv6의 경우 1–128 범위에 있도록 변경합니다.
`Address masks aren't allowed for host names line` *line-number*	구성 파일에서 호스트 이름에 대한 CIDR 마스크를 정의했으며, 이는 IPQoS에서 허용되지 않습니다.	마스크를 제거하거나 호스트 이름을 IP 주소로 변경합니다.
`Invalid module name line` *line-number*	구성 파일에서 작업 명령문에 지정한 모듈 이름이 잘못되었습니다.	모듈 이름의 철자를 확인합니다. IPQoS 모듈 목록은 표 34–5를 참조하십시오.
`ipgpc action has incorrect name line` *line-number*	구성 파일에서 `ipgpc` 작업에 제공한 이름이 필요한 `ipgpc.classify`가 아닙니다.	`ipgpc.classify` 작업 이름을 바꿉니다.
`Second parameter clause not supported line` *line-number*	구성 파일에서 단일 작업에 대해 두 매개변수 절을 지정했으며, 이는 IPQoS에서 허용되지 않습니다.	작업에 대한 모든 매개변수를 단일 매개변수 절로 합칩니다.
`Duplicate named action`	구성 파일에서 두 작업에 동일한 이름을 제공했습니다.	작업 중 하나의 이름을 바꾸거나 제거합니다.
`Duplicate named filter/class in action` *action-name*	동일 작업에서 두 필터 또는 두 클래스에 동일한 이름을 제공했으며, 이는 IPQoS 구성 파일에서 허용되지 않습니다.	필터 또는 클래스 중 하나의 이름을 바꾸거나 제거합니다.
`Undefined class in filter` *filter-name* `in action` *action-name*	구성 파일에서 필터가 작업에 정의되지 않은 클래스를 참조합니다.	클래스를 만들거나 기존 클래스에 대한 필터 참조를 변경합니다.
`Undefined action in class` *class-name* `action` *action-name*	클래스가 구성 파일에서 정의되지 않은 작업을 참조합니다.	작업을 만들거나 기존 작업에 대한 참조를 변경합니다.

표 32–1 IPQoS 오류 메시지 (계속)

오류 메시지	설명	해결 방법
`Invalid parameters for action` *action-name*	구성 파일에서 매개변수 중 하나가 잘못되었습니다.	이름이 지정된 작업으로 호출되는 모듈의 경우 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 항목을 참조합니다. 또는 `ipqosconf`(1M)을 참조할 수 있습니다.
`Mandatory parameter missing for action` *action-name*	구성 파일에서 작업에 대한 필수 매개변수를 정의하지 않았습니다.	이름이 지정된 작업으로 호출되는 모듈의 경우 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 항목을 참조합니다. 또는 `ipqosconf`(1M) 매뉴얼 페이지를 참조할 수 있습니다.
`Max number of classes reached in ipgpc`	IPQoS 구성 파일의 `ipgpc` 작업에서 허용되는 것보다 많은 클래스를 지정했습니다. 최대 수는 10007입니다.	구성 파일을 검토하고 불필요한 클래스를 제거합니다. 또는 `/etc/system` 파일에 `ipgpc_max_classes`*class-number* 항목을 추가하여 최대 클래스 수를 늘릴 수 있습니다.
`Max number of filters reached in action ipgpc`	IPQoS 구성 파일의 `ipgpc` 작업에서 허용되는 것보다 많은 필터를 지정했습니다. 최대 수는 10007입니다.	구성 파일을 검토하고 불필요한 필터를 제거합니다. 또는 `/etc/system` 파일에 `ipgpc_max_filters`*filter-number* 항목을 추가하여 최대 필터 수를 늘릴 수 있습니다.
`Invalid/missing parameters for filter` *filter-name* `in action ipgpc`	구성 파일에 *filter-name* 필터에 잘못되거나 누락된 매개변수가 있습니다.	유효한 매개변수 목록은 `ipqosconf`(1M) 매뉴얼 페이지를 참조하십시오.
`Name not allowed to start with '!', line` *line-number*	작업, 필터 또는 클래스 이름을 느낌표(!)로 시작했으며, 이는 IPQoS 파일에서 허용되지 않습니다.	느낌표를 제거하거나 작업, 클래스 또는 필터 이름을 바꿉니다.
`Name exceeds the maximum name length line` *line-number*	최대 길이 23자를 초과하는 작업, 클래스 또는 필터 이름을 구성 파일에 정의했습니다.	작업, 클래스 또는 필터에 더 짧은 이름을 제공합니다.
`Array declaration line` *line-number* `is invalid`	구성 파일에서 *line-number* 행의 매개변수에 대한 배열 선언이 잘못되었습니다.	잘못된 배열의 `action` 명령문으로 호출되는 배열 선언의 올바른 구문은 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”을 참조하십시오. 또는 `ipqosconf`(1M) 매뉴얼 페이지를 참조하십시오.
`Quoted string exceeds line,` *line-number*	동일 행에서 문자열에 닫는 인용 부호가 없으며, 이는 구성 파일에서 필수입니다.	구성 파일에서 인용 문자열은 동일 행에서 시작되고 끝나야 합니다.
`Invalid value, line` *line-number*	구성 파일의 *line-number*에 제공된 값이 매개변수에 대해 지원되지 않습니다.	`action` 명령문으로 호출되는 모듈에 대해 허용되는 값은 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 설명을 참조하십시오. 또는 `ipqosconf`(1M) 매뉴얼 페이지를 참조할 수 있습니다.

표 32–1 IPQoS 오류 메시지 *(계속)*

오류 메시지	설명	해결 방법
`Unrecognized value, line` *line-number*	구성 파일의 *line-number*에 대한 값이 해당 매개변수에 대해 지원되는 열거 값이 아닙니다.	열거 값이 매개변수에 대해 올바른지 확인합니다. 인식할 수 없는 행 번호의 `action` 명령문으로 호출되는 모듈에 대한 설명은 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”을 참조하십시오. 또는 `ipqosconf`(1M) 매뉴얼 페이지를 참조할 수 있습니다.
`Malformed value list line` *line-number*	구성 파일의 *line-number*에 지정된 열거가 사양 구문을 준수하지 않습니다.	값 목록 형식이 잘못된 `action` 명령문으로 호출되는 모듈에 대한 올바른 구문은 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 설명을 참조하십시오. 또는 `ipqosconf`(1M) 매뉴얼 페이지를 참조할 수 있습니다.
`Duplicate parameter line` *line-number*	*line-number*에 지정된 매개변수가 중복되었으며, 이는 구성 파일에서 허용되지 않습니다.	중복된 매개변수 중 하나를 제거합니다.
`Invalid action name line` *line-number*	구성 파일의 *line-number*에 대한 작업에 사전 정의된 이름 “continue” 또는 “drop”을 사용하는 이름을 제공했습니다.	작업에서 사전 정의된 이름을 사용하지 않도록 작업 이름을 바꿉니다.
`Failed to resolve src/dst host name for filter at line` *line-number*`, ignoring filter`	`ipqosconf`가 구성 파일에 제공된 필터에 대해 정의된 소스 또는 대상 주소를 확인할 수 없습니다. 따라서 필터가 무시되었습니다.	필터가 중요한 경우 나중에 구성 적용을 시도합니다.
`Incompatible address version line` *line-number*	*line-number*에서 주소의 IP 버전이 이전에 지정된 IP 주소 또는 `ip_version` 매개변수의 버전과 호환되지 않습니다.	두 충돌 항목이 호환되도록 변경합니다.
`Action at line` *line-number* `has the same name as currently installed action, but is for a different module`	시스템의 IPQoS 구성에 존재하는 작업의 모듈을 변경하려고 시도했으며, 이는 허용되지 않습니다.	새 구성을 적용하기 전에 현재 구성을 지웁니다.

◆ ◆ ◆ 33장

33 플로우 계산 및 통계 수집 사용(작업)

이 장에서는 IPQoS 시스템이 처리하는 트래픽에 대한 계산 및 통계 정보를 얻는 방법을 설명합니다. 다음 항목을 다룹니다.

흐름 계산 설정(작업 맵)

다음 작업 맵에서는 `flowacct` 모듈을 사용하여 트래픽 흐름에 대한 정보를 얻는 일반적인 작업을 나열합니다. 이 맵에서는 해당 작업을 수행하는 절차와 관련된 링크도 제공합니다.

작업	설명	수행 방법
1. 트래픽 플로우에 대한 계산 정보를 포함할 파일을 만듭니다.	`acctadm` 명령을 사용하여 `flowacct`를 통한 처리 결과를 보관할 파일을 만듭니다.	762 페이지 "플로우 계산 데이터에 대한 파일을 만드는 방법"
2. IPQoS 구성 파일에서 `flowacct` 매개변수를 정의합니다.	`timer`, `timeout` 및 `max_limit` 매개변수에 대한 값을 정의합니다.	738 페이지 "IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법"

트래픽 플로우에 대한 정보 기록

IPQoS flowacct 모듈을 사용하여 트래픽 플로우에 대한 정보를 수집합니다. 예를 들어, 소스 및 대상 주소, 플로우의 패킷 수 및 유사한 데이터를 수집할 수 있습니다. 플로우에 대한 정보를 수집하고 기록하는 프로세스를 **플로우 계산**이라고 합니다.

특정 클래스의 트래픽에 대한 플로우 계산 결과는 **플로우 레코드** 테이블에 기록됩니다. 각 플로우 레코드는 일련의 속성으로 구성됩니다. 이러한 속성에는 일정한 시간 간격에 해당하는 특정 클래스의 트래픽 플로우에 대한 데이터가 포함되어 있습니다. flowacct 속성 목록은 표 34–4를 참조하십시오.

흐름 계산은 SLA(서비스 단계 계약)에 정의된 청구 클라이언트에 특히 유용합니다. 플로우 계산을 사용하여 중요한 응용 프로그램에 대한 플로우 통계를 얻을 수도 있습니다. 이 절에는 Oracle Solaris 확장 계산 기능에 flowacct를 사용하여 트래픽 흐름에 대한 데이터를 얻는 작업이 포함되어 있습니다.

다음 정보는 이 장에서 다루지 않는 소스에서 제공됩니다.

- IPQoS 구성 파일에서 flowacct에 대한 작업문을 만드는 지침은 748 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.
- flowacct 작동 방식에 대해 알아보려면 767 페이지 “분류기 모듈”을 참조하십시오.
- 기술적인 정보는 flowacct(7ipp) 매뉴얼 페이지를 참조하십시오.

▼ 플로우 계산 데이터에 대한 파일을 만드는 방법

IPQoS 구성 파일에 flowacct 작업을 추가하기 전에 flowacct 모듈에서 흐름 레코드에 대한 파일을 만들어야 합니다. 이 용도로 acctadm 명령을 사용합니다. acctadm은 파일에 기본 속성 또는 확장 속성을 기록할 수 있습니다. 모든 flowacct 속성은 표 34–4에 나열됩니다. acctadm에 대한 자세한 내용은 acctadm(1M) 매뉴얼 페이지를 참조하십시오.

1 IPQoS 사용 시스템에서 기본 관리자 역할 또는 수퍼 유저로 로그인합니다.

기본 관리자 역할에는 기본 관리자 프로파일이 포함됩니다. 역할을 만들어 사용자에게 지정하려면 **Oracle Solaris 관리: 기본 관리**의 2 장, “Solaris Management Console 작업(작업)”을 참조하십시오.

2 기본 플로우 계산 파일을 만듭니다.

다음 예에서는 예 31–1에서 구성된 고급 웹 서버에 대한 기본 플로우 계산 파일을 만드는 방법을 보여 줍니다.

```
# /usr/sbin/acctadm -e basic -f /var/ipqos/goldweb/account.info flow
```

acctadm -e	-e 옵션과 함께 acctadm을 호출합니다. -e 옵션 뒤에 인수가 올 수 있습니다.
basic	여덟 개의 기본 flowacct 속성에 대한 데이터만 파일에 기록되도록 지정합니다.

`/var/ipqos/goldweb/account.info`	`flowacct`의 흐름 레코드를 보관할 파일의 정규화된 경로 이름을 지정합니다.
`flow`	플로우 계산을 사용으로 설정하도록 `acctadm`에 지시합니다.

3 인수 없이 `acctadm`을 입력하여 IPQoS 시스템에서 플로우 계산에 대한 정보를 확인합니다.

`acctadm`은 다음 출력을 생성합니다.

```
Task accounting: inactive
       Task accounting file: none
     Tracked task resources: none
   Untracked task resources: extended
         Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
Untracked process resources: extended,host,mstate
           Flow accounting: active
       Flow accounting file: /var/ipqos/goldweb/account.info
     Tracked flow resources: basic
   Untracked flow resources: dsfield,ctime,lseen,projid,uid
```

마지막 네 개를 제외한 모든 항목은 Oracle Solaris 리소스 관리자 기능에 사용됩니다. 다음 표에서는 IPQoS와 관련된 항목에 대해 설명합니다.

항목	설명
`Flow accounting: active`	플로우 계산이 켜져 있음을 나타냅니다.
`Flow accounting file: /var/ipqos/goldweb/account.info`	현재 플로우 계산 파일의 이름을 지정합니다.
`Tracked flow resources: basic`	기본 흐름 속성만 추적됨을 나타냅니다.
`Untracked flow resources: dsfield,ctime,lseen,projid,uid`	파일에서 추적되지 않는 `flowacct` 속성을 나열합니다.

4 (선택 사항) 계산 파일에 확장 속성을 추가합니다.

```
# acctadm -e extended -f /var/ipqos/goldweb/account.info flow
```

5 (선택 사항) 계산 파일에 기본 속성만 기록하도록 되돌립니다.

```
# acctadm -d extended -e basic -f /var/ipqos/goldweb/account.info
```

`-d` 옵션은 확장 계산을 사용 안함으로 설정합니다.

6 플로우 계산 파일의 내용을 확인합니다.

플로우 계산 파일 내용 확인 지침은 **System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones**의 "Perl Interface to libexacct"에서 확인할 수 있습니다.

참조
- 확장 계산 기능에 대한 자세한 내용은 **System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones**의 4 장, “Extended Accounting (Overview)”을 참조하십시오.
- IPQoS 구성 파일에서 `flowacct` 매개변수를 정의하려면 738 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
- `acctadm`으로 만든 파일의 데이터를 인쇄하려면 **System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones**의 “Perl Interface to libexacct”를 참조하십시오.

통계 정보 수집

`kstat` 명령을 사용하여 IPQoS 모듈에서 통계 정보를 생성할 수 있습니다. 다음 구문을 사용하십시오.

`/bin/kstat -m` *ipqos-module-name*

표 34–5와 같이 유효한 IPQoS 모듈 이름을 지정할 수 있습니다. 예를 들어, `dscpmk` 표시자가 생성한 통계를 보려면 다음 형식의 `kstat`를 사용합니다.

```
/bin/kstat -m dscpmk
```

자세한 기술 정보는 `kstat(1M)` 매뉴얼 페이지를 참조하십시오.

예 33–1 IPQoS에 대한 `kstat` 통계

다음은 `flowacct` 모듈에 대한 통계를 얻기 위해 `kstat`를 실행하여 발생할 수 있는 결과의 예입니다.

```
# kstat -m flowacct
module: flowacct                        instance: 3
name:   Flowacct statistics             class:    flacct
        bytes_in_tbl                    84
        crtime                          345728.504106363
        epackets                        0
        flows_in_tbl                    1
        nbytes                          84
        npackets                        1
        snaptime                        345774.031843301
        usedmem                         256
```

`class: flacct` 트래픽 흐름이 속한 클래스의 이름(이 예의 경우 `flacct`)을 지정합니다.

`bytes_in_tbl` 흐름 테이블의 총 바이트 수입니다. 총 바이트 수는 현재 흐름 테이블에 상주하는 모든 흐름 레코드의 합계(바이트)입니다. 이 흐름 테이블의 총 바이트 수는 84입니다. 테이블에 흐름이 없을 경우 `bytes_in_tbl`에 대한 값은 0입니다.

`crtime` 마지막으로 이 `kstat` 출력이 만들어진 시간입니다.

예 33–1 IPQoS에 대한 kstat 통계 *(계속)*

epackets	처리 중 오류가 발생한 패킷 수(이 예의 경우 0)입니다.
flows_in_tbl	흐름 테이블의 흐름 레코드 수(이 예의 경우 1)입니다. 테이블에 레코드가 없을 경우 flows_in_tbl에 대한 값은 0입니다.
nbytes	이 flowacct 작업 인스턴스가 확인한 총 바이트 수(이 예의 경우 84)입니다. 값에는 현재 흐름 테이블에 있는 바이트가 포함됩니다. 또한 값에는 시간이 초과되었으며 흐름 테이블에 더 이상 존재하지 않는 바이트가 포함됩니다.
npackets	이 flowacct 작업 인스턴스가 확인한 총 패킷 수(이 예의 경우 1)입니다. npackets에는 현재 흐름 테이블에 있는 패킷이 포함됩니다. 또한 npackets에는 시간이 초과되었으며 흐름 테이블에 더 이상 존재하지 않는 패킷이 포함됩니다.
usedmem	이 flowacct 인스턴스가 유지 관리하는 흐름 테이블에서 사용 중인 메모리(바이트)입니다. 이 예의 경우 usedmem 값은 256입니다. 흐름 테이블에 흐름 레코드가 없을 경우 usedmem에 대한 값은 0입니다.

◆ ◆ ◆ 34장

34 IPQoS 세부 정보(참조)

이 장에는 다음 IPQoS 항목에 대한 세부 정보를 제공하는 참조 자료가 포함되어 있습니다.

개요는 29 장, "IPQoS 소개(개요)"를 참조하십시오. 계획 정보는 30 장, "IPQoS 사용 네트워크 계획(작업)"을 참조하십시오. IPQoS 구성 절차는 31 장, "IPQoS 구성 파일 만들기(작업)"를 참조하십시오.

IPQoS 아키텍처 및 Diffserv 모델

이 절에서는 IPQoS 아키텍처 및 IPQoS가 RFC 2475, An Architecture for Differentiated Services (`http://www.ietf.org/rfc/rfc2475.txt?number=2475`)에 정의된 차별화 서비스(Diffserv) 모델을 어떻게 구현하는지 설명합니다. IPQoS에는 Diffserv 모델의 다음 요소가 포함됩니다.

- 분류기
- 측정기
- 표시기

또한 IPQoS에는 흐름 계산 모듈 및 VLAN(virtual local area network) 장치와 함께 사용하기 위한 `dlcosmk` 표시기가 포함됩니다.

분류기 모듈

Diffserv 모델에서 **분류기**는 선택된 트래픽 플로우를 서로 다른 서비스 레벨이 적용되는 그룹으로 구성하기 위한 작업을 수행합니다. RFC 2475에서 정의된 분류기는 원래 경계 라우터를 위해 고안되었습니다. 반면, IPQoS 분류기 `ipgpc`은 로컬 네트워크의 내부에

있는 호스트의 트래픽 플로우를 처리하기 위해 고안되었습니다. 그러므로 IPQoS 시스템과 Diffserv 라우터가 모두 있는 네트워크는 더욱 뛰어난 차별화 서비스를 제공할 수 있습니다. ipgpc의 기술적인 설명은 ipgpc(7ipp) 매뉴얼 페이지를 참조하십시오.

ipgpc 분류기는 다음을 수행합니다.

1. IPQoS 사용 시스템의 IPQoS 구성 파일에 지정된 조건을 충족하는 트래픽 플로우를 선택합니다.

   QoS 정책은 패킷 헤더에 있어야 하는 다양한 조건을 정의합니다. 이러한 조건을 **선택기**라고 합니다. ipgpc 분류기는 이러한 선택기를 IPQoS 시스템에서 수신한 패킷의 헤더와 비교한 다음 ipgpc는 모든 일치하는 패킷을 선택합니다.
2. IPQoS 구성 파일에 정의된 대로 패킷 흐름을 동일 특성을 가진 네트워크 트래픽인 **클래스**로 구분합니다.
3. 패킷의 DS(차별화 서비스) 필드 값에 DSCP(차별화 서비스 코드 포인트)가 있는지 검사합니다.

   DSCP가 있으면 수신 트래픽이 전달 동작으로 발신자에 의해 표시되었는지 여부를 나타냅니다.
4. 특정 클래스의 패킷에 대해 IPQoS 구성 파일에서 지정된 추가 작업을 확인합니다.
5. 패킷을 IPQoS 구성 파일에서 지정된 다음 IPQoS 모듈에 전달하거나 패킷을 네트워크 스트림으로 돌려 보냅니다.

분류기의 개요는 699 페이지 “분류기(ipgpc) 개요”를 참조하십시오. IPQoS 구성 파일에서 분류기 호출에 대한 자세한 내용은 779 페이지 “IPQoS 구성 파일”을 참조하십시오.

IPQoS 선택기

ipgpc 분류기는 IPQoS 구성 파일의 filter 절에서 사용할 수 있는 다양한 선택기를 지원합니다. 필터를 정의할 경우 항상 특정 클래스의 트래픽을 성공적으로 검색하는 데 필요한 최소 수의 선택기를 사용하십시오. 정의하는 필터 수에 따라 IPQoS 성능이 영향을 받을 수 있습니다.

다음 표는 ipgpc에 대해 사용 가능한 선택기를 나열합니다.

표 34–1 IPQoS 분류기에 대한 필터 선택기

선택기	인수	선택되는 정보
saddr	IP 주소 번호.	소스 주소입니다.
daddr	IP 주소 번호.	대상 주소입니다.
sport	/etc/services에서 정의된 포트 번호 또는 서비스 이름.	트래픽 클래스가 발생한 소스 포트.

표 34–1 IPQoS 분류기에 대한 필터 선택기 *(계속)*

선택기	인수	선택되는 정보
dport	/etc/services에서 정의된 포트 번호 또는 서비스 이름.	트래픽 클래스가 향하는 대상 포트.
protocol	/etc/protocols에서 정의된 프로토콜 번호 또는 프로토콜 이름.	이 트래픽 클래스에서 사용할 프로토콜.
dsfield	0–63 값의 DSCP(DS 코드 포인트).	패킷에 적용할 전달 동작을 정의하는 DSCP. 이 매개변수가 지정되면 dsfield_mask 매개변수도 지정되어야 합니다.
dsfield_mask	0–255 값의 비트 마스크.	dsfield 선택기와 함께 사용됨. dsfield_mask는 dsfield 선택기에 적용되어 일치시킬 해당 비트를 결정합니다.
if_name	인터페이스 이름.	특정 클래스의 수신 또는 송신 트래픽에 사용될 인터페이스.
user	선택할 UNIX 사용자 ID 또는 사용자 이름 수. 패킷에 사용자 ID 또는 사용자 이름이 없으면 기본값 –1이 사용됩니다.	응용 프로그램에 제공된 사용자 ID.
projid	선택할 프로젝트 ID 수.	응용 프로그램에 제공된 프로젝트 ID.
priority	우선 순위 번호. 가장 낮은 우선 순위는 0입니다.	이 클래스의 패킷에 제공된 우선 순위. 우선 순위는 동일 클래스에 대해 필터의 중요도 순서를 정렬하는 데 사용됩니다.
direction	인수는 다음 중 하나가 될 수 있습니다.	IPQoS 시스템에서 패킷 흐름의 방향.
	LOCAL_IN	IPQoS 시스템에 로컬 입력 트래픽.
	LOCAL_OUT	IPQoS 시스템에 로컬 출력 트래픽.
	FWD_IN	전달할 입력 트래픽.
	FWD_OUT	전달할 출력 트래픽.
precedence	우선권 값. 가장 높은 우선권은 0입니다.	우선권은 동일 우선 순위를 가진 필터 순서를 정렬하는 데 사용됩니다.
ip_version	V4 또는 V6	패킷에서 사용되는 주소 지정 체계(IPv4 또는 IPv6)

측정기 모듈

측정기는 패킷별 기준에서 플로우의 전송 속도를 추적합니다. 그런 다음 측정기는 패킷이 구성된 매개변수를 준수하는지 여부를 확인합니다. 측정기 모듈은 패킷 크기, 구성된 매개변수 및 플로우 속도에 의존하는 작업 세트에서 패킷에 대한 다음 작업을 결정합니다.

측정기는 tokenmt 및 tswtclmt의 두 측정 모듈로 구성되며, IPQoS 구성 파일에서 구성합니다. 한 클래스에 대해 둘 중 하나의 모듈 또는 둘 다 구성할 수 있습니다.

측정 모듈을 구성할 때 속도에 대해 두 매개변수를 정의할 수 있습니다.

- committed-rate – 특정 클래스의 패킷에 대해 수용할 만한 전송 속도(bps, 초당 비트)를 정의합니다.
- peak-rate – 특정 클래스의 패킷에 대해 허용되는 최대 전송 속도(bps, 초당 비트)를 정의합니다.

패킷에 대한 측정 작업 결과는 세 가지 결과 중 하나가 될 수 있습니다.

- green – 패킷으로 인해 흐름이 약정된 속도 내에서 유지됩니다.
- yellow – 패킷으로 인해 흐름이 약정된 속도를 초과하지만 최대 속도를 초과하지는 않습니다.
- red – 패킷으로 인해 플로우가 최대 속도를 초과합니다.

IPQoS 구성 파일에서 서로 다른 작업으로 각 결과를 구성할 수 있습니다. 약정된 속도 및 최대 속도는 다음 절에서 설명합니다.

tokenmt 측정 모듈

tokenmt 모듈은 **토큰 버킷**을 사용하여 플로우의 전송 속도를 측정합니다. tokenmt가 단일 속도 또는 두 가지 속도 측정기로 작동하도록 구성할 수 있습니다. tokenmt 작업 인스턴스는 트래픽 흐름이 구성된 매개변수를 준수하는지 여부를 결정하는 두 토큰 버킷을 유지 관리합니다.

tokenmt(7ipp) 매뉴얼 페이지에서 IPQoS가 토큰 측정기 패러다임을 구현하는 방식을 설명합니다. 토큰 버킷에 대한 일반적인 정보는 Kalevi Kilkki의 *Differentiated Services for the Internet* 및 여러 웹 사이트에서 찾을 수 있습니다.

tokenmt에 대한 구성 매개변수는 다음과 같습니다.

- committed_rate – 플로우의 약정된 속도 bps(초당 비트)로 지정합니다.
- committed_burst – 약정된 버스트 크기를 비트로 지정합니다. committed_burst 매개변수는 특정 클래스의 나가는 패킷이 약정된 속도로 네트워크에 전달될 수 있는 크기를 정의합니다.
- peak_rate – 최대 속도를 bps(초당 비트)로 지정합니다.
- peak_burst – 최대 또는 초과 버스트 크기를 비트로 지정합니다. peak_burst 매개변수는 약정된 속도를 초과하는 최대 버스트 크기를 트래픽 클래스에 부여합니다.
- color_aware – tokenmt에 대한 인식 모드를 설정합니다.
- color_map – DSCP 값을 녹색, 노란색 또는 빨간색으로 매핑하는 정수 배열을 정의합니다.

단일 속도 측정기로 tokenmt 구성

tokenmt를 단일 속도 측정기로 구성하려면 IPQoS 구성 파일에서 tokenmt에 대해 peak_rate 매개변수를 지정하지 마십시오. 단일 속도 tokenmt 인스턴스가 빨간색, 녹색 또는 노란색 결과를 가지도록 구성하려면 peak_burst 매개변수를 지정해야 합니다. peak_burst 매개변수를 사용하지 않을 경우 tokenmt가 빨간색 결과 또는 녹색 결과만 가지도록 구성할 수 있습니다. 두 결과를 가지는 단일 속도 tokenmt의 예는 예 31–3을 참조하십시오.

tokenmt가 단일 속도 측정기로 작동하는 경우 peak_burst 매개변수는 실제로 초과 버스트 크기입니다. committed_rate 및 committed_burst 또는 peak_burst는 0이 아닌 양의 정수여야 합니다.

두 속도 측정기로 tokenmt 구성

tokenmt를 두 속도 측정기로 구성하려면 IPQoS 구성 파일에서 tokenmt 작업에 대해 peak_rate 매개변수를 지정합니다. 두 속도 tokenmt는 항상 빨간색, 노란색 및 녹색의 세 가지 결과를 가집니다. committed_rate , committed_burst 및 peak_burst 매개변수는 0이 아닌 양의 정수여야 합니다.

색상을 인식하도록 tokenmt 구성

두 속도 tokenmt가 색상을 인식하도록 구성하려면 "색상 인식"을 구체적으로 추가하는 매개변수를 추가해야 합니다. 다음은 색상을 인식하도록 tokenmt를 구성하는 예제 작업 명령문입니다.

예 34–1 IPQoS 구성 파일에 대한 색상 인식 tokenmt 작업

```
action {
    module tokenmt
    name meter1
    params {
          committed_rate 4000000
          peak_rate 8000000
          committed_burst 4000000
          peak_burst 8000000
          global_stats true
          red_action_name continue
          yellow_action_name continue
          green_action_name continue
          color_aware true
          color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
    }
}
```

color_aware 매개변수를 true로 설정하여 색상 인식을 사용으로 설정해야 합니다. 색상 인식 측정기로서 tokenmt는 패킷이 이전 tokenmt 작업에 의해 이미 빨간색, 노란색 또는 녹색으로 표시되었다고 간주합니다. 색상 인식 tokenmt는 두 속도 측정기에 대한 매개변수와 함께 패킷 헤더의 DSCP를 사용하여 패킷을 평가합니다.

color_map 매개변수에는 패킷 헤더의 DSCP가 매핑되는 배열이 포함됩니다. 다음 color_map 배열을 고려하십시오.

```
color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
```

DSCP 0–20 및 22의 패킷은 녹색으로 매핑됩니다. DSCP 21 및 23–42의 패킷은 빨간색으로 매핑됩니다. DSCP 43–63의 패킷은 노란색으로 매핑됩니다. tokenmt는 기본 색상 맵을 유지 관리합니다. 하지만 color_map 매개변수를 사용하여 필요에 따라 기본값을 변경할 수 있습니다.

*color*_action_name 매개변수에서 continue를 지정하여 패킷 처리를 완료할 수 있습니다. 또는 패킷을 표시기 작업에 보내는 인수를 추가할 수 있습니다(예: yellow_action_name mark22).

tswtclmt 측정 모듈

tswtclmt 측정 모듈은 시간 기반 **속도 추정기**를 사용하여 트래픽 클래스에 대한 평균 대역폭을 추정합니다. tswtclmt는 항상 세 가지 결과 측정기로 작동합니다. 속도 추정기는 플로우의 도착 추정 속도를 제공합니다. 이 속도는 지정된 기간(**시간 창**) 동안 트래픽 스트림의 실행 평균 대역폭에 근접해야 합니다. 속도 추정 알고리즘은 RFC 2859, *A Time Sliding Window Three Colour Marker*에서 가져옵니다.

다음 매개변수를 사용하여 tswtclmt를 구성합니다.

- committed_rate – 약정된 속도를 bps(초당 비트)로 지정합니다.
- peak_rate – 최대 속도를 bps(초당 비트)로 지정합니다.
- window – 평균 대역폭 내역이 보관되는 시간 창을 밀리초로 정의합니다.

tswtclmt에 대한 자세한 기술 정보는 tswtclmt(7ipp) 매뉴얼 페이지를 참조하십시오. tswtclmt와 유사한 속도 셔퍼(Shaper)에 대한 일반적인 정보는 RFC 2963, A Rate Adaptive Shaper for Differentiated Services (http://www.ietf.org/rfc/rfc2963.txt?number=2963)를 참조하십시오.

표시기 모듈

IPQoS에는 dscpmk 및 dlcosmk의 두 표시기 모듈이 포함됩니다. 이 절에서는 두 표시기 사용에 대한 정보가 포함되어 있습니다. dlcosmk만 VLAN 장치가 있는 IPQoS 시스템에 대해 사용할 수 있으므로 일반적으로는 dscpmk를 사용해야 합니다.

dscpmk에 대한 기술적인 정보는 dscpmk(7ipp) 매뉴얼 페이지를 참조하십시오. dlcosmk에 대한 기술적인 정보는 dlcosmk(7ipp) 매뉴얼 페이지를 참조하십시오.

패킷 전달을 위해 dscpmk 표시기 사용

표시기는 분류기 또는 측정 모듈이 플로우를 처리한 후 트래픽 플로우를 수신합니다. 표시기는 트래픽을 전달 동작으로 표시합니다. 이 전달 동작은 플로우가 IPQoS 시스템을 떠난 후 수행할 작업입니다. 트래픽 클래스에 대해 수행할 전달 동작은 *PHB(홉별*

동작)에서 정의됩니다. PHB는 다른 트래픽 클래스와 관련하여 해당 클래스의 우선권 플로우를 나타내는 우선 순위를 트래픽 클래스에 지정합니다. PHB는 IPQoS 시스템의 인접 네트워크에 대한 전달 동작만 제어합니다. PHB에 대한 자세한 내용은 703 페이지 "홉별 동작"을 참조하십시오.

패킷 전달은 특정 클래스의 트래픽을 네트워크의 다음 대상으로 보내는 프로세스입니다. IPQoS 시스템과 같은 호스트의 경우, 패킷은 호스트에서 로컬 네트워크 스트림으로 전달됩니다. Diffserv 라우터의 경우, 패킷은 로컬 네트워크에서 라우터의 다음 홉으로 전달됩니다.

표시기는 패킷 헤더의 DS 필드를 IPQoS 구성 파일에서 정의된 잘 알려진 전달 동작으로 표시합니다. 그러면 IPQoS 시스템 및 후속 Diffserv 인식 시스템은 표시가 바뀔 때까지 DS 필드에 나타난 대로 트래픽을 전달합니다. PHB를 지정하기 위해 IPQoS 시스템은 패킷 헤더의 DS 필드에 값을 표시합니다. 이 값을 DSCP(차별화 서비스 코드 포인트)라고 합니다. Diffserv 아키텍처는 서로 다른 DSCP를 사용하는 두 가지 유형의 전달 동작인 EF 및 AF를 정의합니다. DSCP에 대한 개요는 703 페이지 "DS 코드 포인트"를 참조하십시오.

IPQoS 시스템은 트래픽 플로우에 대해 DSCP를 읽고 다른 송신 트래픽 플로우와 관련하여 플로우의 우선권을 평가합니다. 그런 다음 IPQoS 시스템은 모든 동시 트래픽 플로우에 우선 순위를 지정하고 각 플로우를 우선 순위에 따라 네트워크로 보냅니다.

Diffserv 라우터는 송신 트래픽 플로우를 수신하고 패킷 헤더의 DS 필드를 읽습니다. DSCP는 라우터가 동시 트래픽 플로우에 우선 순위를 지정하고 일정을 예약하도록 합니다. 라우터는 PHB로 지정된 우선 순위에 따라 각 플로우를 전달합니다. 후속 홉의 Diffserv 인식 시스템도 동일한 PHB를 인식하지 못하면 PHB는 네트워크의 경계 라우터를 벗어나서 적용할 수 없습니다.

EF(빠른 전달) PHB

빠른 전달(EF)은 권장 EF 코드 포인트 46(101110)의 패킷이 네트워크로 전송 시 사용 가능한 가장 좋은 취급을 받도록 보장합니다. 빠른 전달은 임대 회선과 비교되기도 합니다. 46(101110) 코드 포인트의 패킷은 패킷의 대상으로 향하는 모든 Diffserv 경로에서 선호 취급이 보장됩니다. EF에 대한 기술적인 정보는 An Expedited Forwarding PHB (`http://www.ietf.org/rfc/rfc2598.txt`)를 참조하십시오.

AF(보장 전달) PHB

보장 전달(AF)은 표시기에 지정할 수 있는 네 가지 서로 다른 클래스의 전달 동작을 제공합니다. 다음 표는 클래스, 각 클래스에 제공되는 세 가지 삭제 우선권 및 각 우선권과 연결된 권장 DSCP를 보여줍니다. 각 DSCP는 해당 AF 값, 십진수 값 및 이진수 값으로 표시됩니다.

표 34–2 보장 전달 코드점

	클래스 1	클래스 2	클래스 3	클래스 4
낮은 삭제 우선권	AF11 = 10 (001010)	AF21 = 18 (010010)	AF31 = 26 (011010)	AF41 = 34 (100010)
중간 삭제 우선권	AF12 = 12 (001100)	AF22 = 20 (010100)	AF32 = 28 (011100)	AF42 = 36 (100100)
높은 삭제 우선권	AF13 = 14 (001110)	AF23 = 22 (010110)	AF33 = 30 (011110)	AF43 = 38 (100110)

모든 Diffserv 인식 시스템에서는 AF 코드 포인트를 기준으로 사용하여 서로 다른 클래스의 트래픽에 차별화된 전달 동작을 제공할 수 있습니다.

이러한 패킷이 Diffserv 라우터에 도달하면 라우터는 대기열에 있는 다른 트래픽의 DSCP와 함께 패킷의 코드 포인트를 평가합니다. 그런 다음 라우터는 사용 가능한 대역폭 및 패킷의 DSCP로 지정된 우선 순위에 따라 패킷을 전달하거나 삭제합니다. EF PHB로 표시된 패킷은 다양한 AF PHB로 표시된 패킷에 비해 대역폭이 보장됩니다.

패킷이 예상한 대로 전달되도록 하려면 네트워크의 IPQoS 시스템과 Diffserv 라우터 사이에 패킷 표시를 조정하십시오. 예를 들어, 네트워크의 IPQoS 시스템이 AF21(010010), AF13(001110), AF43(100110) 및 EF(101110) 코드 포인트로 패킷을 표시한다고 가정해 보겠습니다. 그러면 AF21, AF13, AF43 및 EF DSCP를 Diffserv 라우터의 해당 파일에 추가해야 합니다.

AF 코드점 표의 기술적인 설명은 Assured Forwarding PHB Group (`http://tools.ietf.org/html/rfc2597`)을 참조하십시오. 라우터 제조업체 Cisco Systems 및 Juniper Networks의 회사 웹 사이트에 가면 자세한 AF PHB 설정 정보가 제공됩니다. 이러한 정보를 활용하여 IPQoS 시스템 및 라우터에 대한 AF PHB를 정의할 수 있습니다. 또한 라우터 제조업체의 설명서에는 자사 장비에서 DS 코드 포인트 설정에 대한 지침이 포함되어 있습니다.

표시기에 DSCP 제공

DSCP는 6비트 길이입니다. DS 필드는 1바이트 길이입니다. DSCP를 정의할 때 표시기는 패킷 헤더의 처음 중요 6비트를 DS 코드 포인트로 표시합니다. 나머지 덜 중요한 2비트는 사용되지 않습니다.

DSCP를 정의하려면 표시기 작업 매개변수 내에서 다음 매개변수를 사용합니다.

`dscp_map{0-63:`*DS_codepoint*`}`

`dscp_map` 매개변수는 (DSCP) 값으로 채우는 64 요소 배열입니다. `dscp_map`은 들어오는 DSCP를 `dscpmk` 표시기에 의해 적용된 나가는 DSCP로 매핑하는 데 사용됩니다.

DSCP 값은 십진수 형식의 dscp_map으로 지정해야 합니다. 예를 들어, EF 코드 포인트 101110은 십진수 값 46으로 변환해야 하며, 결과적으로 dscp_map{0-63:46}이 됩니다. AF 코드점의 경우 표 34–2에 나온 다양한 코드점을 dscp_map에서 사용할 십진수 표기법으로 변환해야 합니다.

VLAN 장치에서 dlcosmk 표시기 사용

dlcosmk 표시기 모듈은 데이터 그램의 MAC 헤더에서 전달 동작을 표시합니다. VLAN 인터페이스가 있는 IPQoS 시스템에서만 dlcosmk를 사용할 수 있습니다.

dlcosmk는 *VLAN* **태그**로 알려진 4바이트를 MAC 헤더에 추가합니다. VLAN 태그에는 IEEE 801.D 표준에서 정의된 3비트 사용자 우선 순위 값이 포함됩니다. VLAN을 이해하는 Diffserv 인식 스위치는 데이터그램의 사용자 우선 순위 필드를 읽을 수 있습니다. 801.D 사용자 우선 순위 값은 잘 알려지고 상용 스위치에서 이해할 수 있는 CoS(서비스 클래스) 표시를 구현합니다.

다음 표에 나열된 서비스 클래스 표시를 정의하여 dlcosmk 표시기 작업에서 사용자 우선 순위 값을 사용할 수 있습니다.

표 34–3 801.D 사용자 우선 순위 값

서비스 클래스	정의
0	최선 조건
1	백그라운드
2	여분
3	최우선 조건
4	제어 로드
5	100ms 대기 시간 미만의 비디오
6	10ms 대기 시간 미만의 비디오
7	네트워크 제어

dlcosmk에 대한 자세한 내용은 dlcosmk(7ipp) 매뉴얼 페이지를 참조하십시오.

VLAN 장치가 있는 시스템에 대한 IPQoS 구성

이 절에서는 VLAN 장치가 있는 시스템에서 IPQoS를 구현하는 방법을 보여주는 단순한 네트워크 시나리오를 소개합니다. 시나리오에는 스위치로 연결된 machine1 및 machine2의 두 IPQoS 시스템이 포함됩니다. machine1의 VLAN 장치는 IP 주소 10.10.8.1을 가집니다. machine2의 VLAN 장치는 IP 주소 10.10.8.3을 가집니다.

machine1에 대한 다음 IPQoS 구성 파일은 스위치를 거쳐 machine2로 이동하는 트래픽을 표시하기 위한 간단한 솔루션을 보여줍니다.

예 34–2 VLAN 장치가 있는 시스템에 대한 IPQoS 구성 파일

```
fmt_version 1.0
action {
        module ipgpc
          name ipgpc.classify

        filter {
                name myfilter2
                daddr 10.10.8.3
                class myclass
        }

        class {
                name myclass
                next_action mark4
        }
}

action {
        name mark4
        module dlcosmk
        params {
                cos 4
                next_action continue
        global_stats true
        }
}
```

이 구성에서 machine2의 VLAN 장치를 대상으로 하는 machine1의 모든 트래픽은 dlcosmk 표시기로 전달됩니다. mark4 표시기 작업은 dlcosmk가 VLAN 표시를 CoS가 4인 myclass 클래스의 데이터그램에 추가하도록 지시합니다. 사용자 우선 순위 값 4는 두 시스템 사이에 있는 스위치가 machine1의 myclass 트래픽 흐름에 제어 로드 전달을 제공해야 한다는 것을 나타냅니다.

flowacct 모듈

IPQoS flowacct 모듈은 트래픽 흐름에 대한 정보를 기록하며, 이 프로세스를 **흐름 계산**이라고 합니다. 플로우 계산은 고객 청구 또는 특정 클래스에 대한 트래픽의 양 평가 목적으로 사용될 수 있는 데이터를 생성합니다.

플로우 계산은 선택 사항입니다. flowacct는 일반적으로 측정되거나 표시된 트래픽 플로우가 네트워크 스트림으로 보내지기 전에 만날 수 있는 마지막 모듈입니다. Diffserv 모델에서 flowacct의 위치에 대한 그림은 그림 29–1을 참조하십시오. flowacct에 대한 자세한 기술 정보는 flowacct(7ipp) 매뉴얼 페이지를 참조하십시오.

흐름 계산을 사용으로 설정하려면 flowacct와 함께 Oracle Solaris exacct 계산 기능 및 acctadm 명령을 사용해야 합니다. 플로우 계산 설정에 대한 전체 단계는 761 페이지 “흐름 계산 설정(작업 맵)”을 참조하십시오.

flowacct 매개변수

flowacct 모듈은 **플로우 레코드**로 구성된 **플로우 테이블**에서 플로우에 대한 정보를 수집합니다. 테이블의 각 항목은 하나의 플로우 레코드를 포함합니다. 플로우 테이블을 표시할 수는 없습니다.

IPQoS 구성 파일에서 다음 flowacct 매개변수를 정의하여 흐름 레코드를 측정하고 레코드를 흐름 테이블에 기록합니다.

- timer – 시간 초과된 플로우가 플로우 테이블에서 제거되고 acctadm으로 만들어진 파일에 기록되는 간격을 밀리초로 정의합니다.
- timeout – 플로우가 시간 초과되기 전에 패킷 플로우가 비활성화되어야 하는 기간을 밀리초로 정의합니다.

주 – timer 및 timeout이 서로 다른 값을 가지도록 구성할 수 있습니다.

- max_limit – 플로우 테이블에 저장할 수 있는 플로우 레코드 수에 대한 상한 제한을 둡니다.

flowacct 매개변수가 IPQoS 구성 파일에서 사용되는 예는 748 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.

플로우 테이블

flowacct 모듈은 flowacct 인스턴스에서 확인되는 모든 패킷 플로우를 기록하는 플로우 테이블을 유지 관리합니다.

플로우는 flowacct 8-튜플을 포함하는 다음 매개변수로 식별됩니다.

- 소스 주소
- 대상 주소
- 소스 포트
- 대상 포트
- DSCP
- 사용자 ID
- 프로젝트 ID
- 프로토콜 번호

흐름에 대한 8–튜플의 모든 매개변수가 동일하게 유지될 경우 흐름 테이블은 하나의 항목만 포함합니다. max_limit 매개변수는 흐름 테이블에 포함될 수 있는 항목 수를 결정합니다.

흐름 테이블은 timer 매개변수에 대해 IPQoS 구성 파일에서 정의된 간격으로 검사됩니다. 기본값은 15초입니다. 흐름은 IPQoS 구성 파일의 timeout 간격 이상 동안 IPQoS 시스템에서 해당 패킷을 볼 수 없을 때 “시간 초과”됩니다. 기본 시간 초과 간격은 60초입니다. 그런 다음 시간 초과된 항목은 acctadm 명령으로 만들어진 계산 파일에 기록됩니다.

flowacct 레코드

flowacct 레코드에는 다음 표에 설명된 속성이 포함됩니다.

표 34–4 flowacct 레코드의 속성

속성 이름	속성 내용	유형
src-addr-*address-type*	발신자의 소스 주소. *address-type*은 IPQoS 구성 파일에 지정된 대로 IPv4의 경우 v4 또는 IPv6의 경우 v6입니다.	기본
dest-addr-*address-type*	패킷에 대한 대상 주소. *address-type*은 IPQoS 구성 파일에 지정된 대로 IPv4의 경우 v4 또는 IPv6의 경우 v6입니다.	기본
src-port	흐름이 발생한 소스 포트.	기본
dest-port	이 흐름이 향하는 대상 포트 번호.	기본
protocol	흐름에 대한 프로토콜 번호.	기본
total-packets	흐름의 패킷 수.	기본
total-bytes	흐름의 바이트 수.	기본
action-name	이 흐름을 기록한 flowacct 작업의 이름.	기본
creation-time	흐름에 대한 패킷이 flowacct에 의해 처음으로 목격된 시간.	확장 전용
last-seen	흐름의 패킷이 마지막으로 목격된 시간.	확장 전용
diffserv-field	흐름의 나가는 패킷 헤더에 있는 DSCP.	확장 전용
user	응용 프로그램에서 가져온 UNIX 사용자 ID 또는 사용자 이름.	확장 전용
projid	응용 프로그램에서 가져온 프로젝트 ID.	확장 전용

flowacct 모듈에서 acctadm 사용

acctadm 명령을 사용하여 flowacct로 생성된 다양한 흐름 레코드를 저장할 파일을 만듭니다. acctadm은 확장 계산 기능과 함께 작동합니다. acctadm에 대한 기술적인 정보는 acctadm(1M) 매뉴얼 페이지를 참조하십시오.

flowacct 모듈은 흐름을 관찰하고 흐름 테이블을 레코드로 채웁니다. 그런 다음 flowacct는 timer에서 지정된 간격으로 해당 매개변수 및 속성을 평가합니다. 패킷이 last_seen + timeout 값 이상 동안 보이지 않으면 패킷은 시간 초과됩니다. 모든 시간 초과된 항목은 흐름 테이블에서 삭제됩니다. 그런 다음 이러한 항목은 timer 매개변수에 지정된 간격이 경과할 때마다 계산 파일에 기록됩니다.

flowacct 모듈에서 사용할 acctadm을 호출하려면 다음 구문을 사용합니다.

```
acctadm -e file-type -f filename flow
```

acctadm -e -e 옵션과 함께 acctadm을 호출합니다. -e는 리소스 목록이 있음을 나타냅니다.

file-type	수집할 속성을 지정합니다. *file-type*은 `basic` 또는 `extended`로 바뀌어야 합니다. 각 파일 유형의 속성 목록은 표 34–4를 참조하십시오.
`-f`*file-name*	플로우 레코드를 보관할 *file-name* 파일을 만듭니다.
`flow`	`acctadm`이 IPQoS에서 실행됨을 나타냅니다.

IPQoS 구성 파일

이 절에서는 IPQoS 구성 파일의 부분에 대한 전체 세부 정보가 포함되어 있습니다. IPQoS 부트 시 활성화되는 정책은 `/etc/inet/ipqosinit.conf` 파일에 저장됩니다. 이 파일을 편집할 수 있지만 새 IPQoS 시스템의 경우 가장 좋은 방법은 다른 이름으로 구성 파일을 만드는 것입니다. IPQoS 구성을 적용하고 디버깅하는 작업은 31 장, "IPQoS 구성 파일 만들기(작업)"를 참조하십시오.

IPQoS 구성 파일의 구문은 예 34–3을 참조하십시오.

예에서는 다음 규약을 사용합니다.

- **컴퓨터 스타일 유형** – 구성 파일의 부분을 설명하기 위해 제공되는 구문 정보입니다. 컴퓨터 스타일 유형으로 나타나는 텍스트는 입력하지 않습니다.
- **굵은체 유형** – IPQoS 구성 파일에 입력해야 하는 리터럴 텍스트입니다. 예를 들어, IPQoS 구성 파일은 항상 **`fmt_version`**으로 시작해야 합니다.
- **기울임꼴 유형** – 구성에 대한 설명 정보로 바꾸는 변수 텍스트입니다. 예를 들어, *action-name* 또는 *module-name*은 항상 조직에 해당하는 정보로 바꾸어야 합니다.

예 34–3 IPQoS 구성 파일의 구문

```
file_format_version ::= fmt_version version

action_clause ::= action {
      name action-name
      module module-name
      params-clause |  ""
      cf-clauses
}
action_name ::= string
module_name ::= ipgpc | dlcosmk | dscpmk | tswtclmt | tokenmt | flowacct

params_clause ::= params {
      parameters
      params-stats |   ""
      }
parameters ::=    prm-name-value parameters |  ""
prm_name_value ::= param-name param-value

params_stats ::= global-stats boolean

cf_clauses ::= class-clause cf-clauses |
```

예 34–3 IPQoS 구성 파일의 구문 (계속)

```
                    filter-clause cf-clauses | ""

class_clause ::= class {
      name class-name
      next_action next-action-name
      class-stats | ""
                   }
class_name  ::= string
next_action_name  ::= string
class_stats ::= enable_stats boolean
boolean ::= TRUE | FALSE

filter_clause ::= filter {
                  name filter-name
                  class class–name
                  parameters
                  }
filter_name ::= string
```

IPQoS 구성 파일의 각 주요 부분을 설명하는 나머지 텍스트입니다.

action 명령문

action 명령문을 사용하여 767 페이지 “IPQoS 아키텍처 및 Diffserv 모델”에 설명된 다양한 IPQoS 모듈을 호출합니다.

IPQoS 구성 파일을 만들 때는 항상 버전 번호로 시작해야 합니다. 그리고 다음 action 명령문을 추가하여 분류기를 호출합니다.

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
}
```

분류기 action 명령문 다음에는 params 절 또는 class 절이 옵니다.

기타 모든 action 명령문에 대해 다음 구문을 사용합니다.

```
action {
name action-name
module module-name
params-clause | ""
cf-clauses
}
```

name *action_name* 작업에 이름을 지정합니다.

`module` *module_name*	호출할 IPQoS 모듈을 식별합니다. 표 34–5의 모듈 중 하나이어야 합니다.
params_clause	분류기가 처리할 매개변수가 될 수 있습니다(예: 전역 통계 또는 처리할 다음 작업).
cf_clauses	0개 이상의 `class` 절 또는 `filter` 절 집합입니다.

모듈 정의

모듈 정의는 `action` 명령문에서 매개변수를 처리할 모듈을 나타냅니다. IPQoS 구성 파일에는 다음 모듈이 포함될 수 있습니다.

표 34–5 IPQoS 모듈

모듈 이름	정의
`ipgpc`	IP 분류기
`dscpmk`	IP 패킷에서 DSCP를 만드는 데 사용할 표시기
`dlcosmk`	VLAN 장치에서 사용할 표시기
`tokenmt`	토큰 버킷 측정기
`tswtclmt`	시간별 창 측정기
`flowacct`	흐름 계산 모듈

`class` 절

각 트래픽 클래스에 대해 `class` 절을 정의합니다.

이 구문을 사용하여 IPQoS 구성의 나머지 절을 정의합니다.

```
class {

      name class-name
      next_action next-action-name
}
```

특정 클래스에 대한 통계 수집을 사용으로 설정하려면 먼저 `ipgpc.classify action` 명령문에서 전역 통계를 사용으로 설정해야 합니다. 자세한 내용은 780 페이지 “action 명령문”을 참조하십시오.

클래스에 대한 통계 수집을 설정할 때는 항상 `enable_stats TRUE` 명령문을 사용합니다. 클래스에 대한 통계를 수집할 필요가 없는 경우 `enable_stats FALSE`를 지정할 수 있습니다. 또는 `enable_stats` 명령문을 제거할 수 있습니다.

명시적으로 정의하지 않은 IPQoS 사용 네트워크에 대한 트래픽은 **기본 클래스**로 들어갑니다.

filter 절

필터는 트래픽 플로우를 클래스로 그룹화하는 선택기로 구성됩니다. 이러한 선택기는 class 절에서 만들어진 클래스의 트래픽에 적용될 조건을 구체적으로 정의합니다. 패킷이 가장 높은 우선 순위 필터의 모든 선택기와 일치할 경우 해당 패킷은 필터 클래스의 멤버로 간주됩니다. ipgpc 분류기에서 사용할 수 있는 전체 선택기 목록은 표 34–1을 참조하십시오.

다음 구문을 가지는 *filter* **절**을 사용하여 IPQoS 구성 파일에서 필터를 정의합니다.

```
filter {
        name filter-name
        class class-name
        parameters (selectors)
        }
```

params 절

params 절에는 작업 명령문에서 정의된 모듈에 대한 처리 지침이 포함됩니다. params 절에 대해 다음 구문을 사용합니다.

```
params {
            parameters
            params-stats | ""
        }
```

params 절에서 모듈에 적용 가능한 매개변수를 사용합니다.

params 절의 *params-stats* 값은 global_stats TRUE 또는 global_stats FALSE입니다. global_stats TRUE 지침은 전역 통계가 호출되는 action 명령문에 대해 UNIX 스타일 통계를 설정합니다. 통계는 kstat 명령을 사용하여 볼 수 있습니다. 클래스별 통계를 사용으로 설정하려면 먼저 action 명령문 통계를 사용으로 설정해야 합니다.

ipqosconf 구성 유틸리티

ipqosconf 유틸리티를 사용하여 IPQoS 구성 파일을 읽고 UNIX 커널에서 IPQoS 모듈을 구성합니다. ipqosconf는 다음 작업을 수행합니다.

- 구성 파일을 IPQoS 커널 모듈에 적용합니다(ipqosconf -a *filename*).
- 커널에 현재 상주하는 IPQoS 구성 파일을 나열합니다(ipqosconf -l).
- 시스템이 재부트될 때마다 현재 IPQoS 구성을 읽고 적용되도록 합니다(ipqosconf -c).

- 현재 IPQoS 커널 모듈을 비웁니다(ipqosconf -f).

기술적인 정보는 ipqosconf(1M) 매뉴얼 페이지를 참조하십시오.

용어집

3DES Triple-DES를 참조하십시오.

AES Advanced Encryption Standard입니다. 대칭 128비트 블록 데이터 암호화 기술입니다. 미국 정부는 2000년 10월 알고리즘의 Rijndael 변형을 암호화 표준으로 채택했습니다. AES가 정부 표준으로 DES 암호화를 대체합니다.

Blowfish 32-448비트의 가변 길이 키를 사용하는 대칭 블록 암호화 알고리즘입니다. 저작자인 Bruce Schneier에 따르면, Blowfish는 키를 자주 바꾸지 않는 응용 프로그램에 최적화되어 있습니다.

CA CA(인증 기관)을 참조하십시오.

CA (인증 기관) 디지털 서명 및 공개-개인 키 쌍을 만드는 데 사용된 디지털 인증서를 발행하는 신뢰된 타사 조직 또는 회사입니다. CA는 고유한 인증서를 부여받은 개인의 신원을 보증합니다.

care-of 주소 모바일 노드가 외래 네트워크에 연결되었을 때 터널 출구점으로 사용되는 모바일 노드의 임시 주소입니다.

CIDR (classless inter-domain routing) 주소 네트워크 클래스(클래스 A, B, C)를 기반으로 하지 않는 IPv4 주소 형식입니다. CIDR 주소는 길이가 32비트입니다. 네트워크 접두어를 추가하여 표준 IPv4 점으로 구분된 십진수 표기법 형식을 사용합니다. 이 접두어는 네트워크 번호와 네트워크 마스크를 정의합니다.

class IPQoS에서 유사한 특성을 공유하는 네트워크 플로우 그룹입니다. IPQoS 구성 파일에서 클래스를 정의합니다.

CRL (인증서 해지 목록) CA에 의해 해지된 공개 키 인증서 목록입니다. CRL은 IKE를 통해 유지 관리하는 CRL 데이터베이스에 저장됩니다.

DEPRECATED 주소 IPMP 그룹에서 데이터의 소스 주소로 사용할 수 없는 IP 주소입니다. 일반적으로 IPMP 테스트 주소는 DEPRECATED입니다. 하지만 아무 주소나 DEPRECATED로 표시하여 해당 주소가 소스 주소로 사용되지 않도록 할 수 있습니다.

DES Data Encryption Standard입니다. 1975년에 개발되고 1981년에 ANSI에 의해 ANSI X.3.92로 표준화된 대칭 키 암호화 방법입니다. DES에서는 56비트 키를 사용합니다.

Diffie-Hellman 알고리즘 공개 키 암호화라고도 합니다. 1976년에 Diffie와 Hellman이 공동 개발한 비대칭 암호화 키 협정 규약입니다. 이 프로토콜을 사용하면 어떤 예비 보안 없이도 두 사용자가 비보안 매체를 통해 보안 키를 교환할 수 있습니다. Diffie-Hellman은 IKE 프로토콜에서 사용됩니다.

diffserv 모델 IP 네트워크에서 차등화 서비스를 구현하기 위한 IETF(Internet Engineering Task Force) 구조 표준입니다. 주 모듈에는 분류자, 측정자, 표시자, 스케줄러, 삭제자가 있습니다. IPQoS는 분류자, 측정자, 표시자 모듈을 구현합니다. diffserv 모델은 RFC 2475 *An Architecture for Differentiated Services*에 설명됩니다.

DOI (Domain of Interpretation) DOI는 데이터 형식, 네트워크 트래픽 교환 유형 및 보안 관련 정보의 이름 지정 규약을 정의합니다. 보안 관련 정보의 예로 보안 정책, 암호화 알고리즘, 암호화 모드 등이 있습니다.

DS 코드점 (DSCP) IP 헤더의 DS 필드에 포함될 때 패킷의 전달 방법을 나타내는 6비트 값입니다.

DSA 디지털 서명 알고리즘(Digital Signature Algorithm). 512-4096비트의 가변 키 크기를 사용하는 공개 키 알고리즘입니다. 미국 정부 표준인 DSS는 1024비트까지 지원합니다. DSA는 입력에 SHA-1을 사용합니다.

ESP (보안 페이로드 캡슐화) 데이터그램에 무결성 및 기밀성을 제공하는 확장 헤더입니다. ESP는 IP 보안 구조(IPsec)의 5개 구성 요소 중 하나입니다.

GRE (Generic Routing Encapsulatio) 홈 에이전트, 외래 에이전트, 모바일 노드에서 지원할 수 있는 선택적 형태의 터널링입니다. GRE를 사용으로 설정하면 네트워크 계층 프로토콜의 패킷을 다른 또는 동일한 네트워크 계층 프로토콜의 배달 패킷 내에서 캡슐화할 수 있습니다.

HMAC 메시지 인증을 위해 입력한 해싱 방법입니다. HMAC는 보안 키 인증 알고리즘입니다. HMAC는 비밀 공유 키와 조합하여 MD5 또는 SHA-1과 같은 반복 암호화 해시 기능과 함께 사용합니다. 기본 해시 기능의 등록 정보에 따라 HMAC의 암호화 강도가 달라집니다.

ICMP 인터넷 제어 메시지 프로토콜(Internet Control Message Protocol). 오류를 처리하고 제어 메시지를 교환하는 데 사용됩니다.

ICMP 에코 요청 패킷 인터넷에서 응답을 간청하기 위해 시스템으로 보낸 패킷입니다. 이러한 패킷을 흔히 "ping" 패킷이라고 합니다.

IKE 인터넷 키 교환(Internet Key Exchange). IKE는 IPsec 보안 연관(SA)에 대한 인증된 키 입력 자료의 프로비전을 자동화합니다.

IP IP(인터넷 프로토콜), IPv4, IPv6을 참조하십시오.

IP-in-IP 캡슐화 IP 패킷 내의 IP 패킷을 터널링하기 위한 방식입니다.

IP 데이터그램 IP를 통해 전달된 정보의 패킷입니다. IP 데이터그램은 헤더 및 데이터를 포함합니다. 헤더는 데이터그램의 소스 및 대상 주소를 포함합니다. 헤더의 다른 필드를 통해 대상에서 데이터와 동반 데이터그램을 식별하고 재컴파일할 수 있습니다.

IP 링크 링크 계층에서 노드가 통신할 수 있는 통신 기능 또는 매체입니다. 링크 계층은 IPv4/IPv6 바로 아래에 있는 계층입니다. 예를 들어, 이더넷(단순 또는 브리지됨) 또는 ATM 네트워크가 있습니다. IP 링크 한 개에 IPv4 서브넷 번호 또는 접두어가 한 개 이상 지정됩니다. 서브넷 번호 또는 접두어 한 개를 여러 IP 링크에 지정할 수는 없습니다. ATM LANE에서 IP 링크는 에뮬레이트된 단일 LAN입니다. ARP를 사용하는 경우 ARP 프로토콜의 범위는 단일 IP 링크입니다.

IP 스택 TCP/IP를 종종 "스택"이라고도 합니다. 이것은 데이터 교환의 클라이언트측과 서버측 양쪽에서 모든 데이터가 전달되는 계층(TCP, IP 및 기타)을 가리킵니다.

IP (인터넷 프로토콜) 인터넷을 통해 한 컴퓨터에서 다른 컴퓨터로 데이터를 보내는 방법 또는 규약입니다.

IP 헤더 인터넷 패킷을 고유하게 식별하는 20바이트의 데이터입니다. 헤더는 패킷의 소스 및 대상 주소를 포함합니다. 헤더 내에는 바이트를 더 추가할 수 있는 옵션이 존재합니다.

IPMP 그룹 네트워크 가용성과 사용률 향상을 위해 시스템에서 교환 가능한 것으로 처리되는 데이터 주소 세트를 가진 네트워크 인터페이스 세트로 구성된 IP 다중 경로 그룹입니다. 모든 기본 IP 인터페이스와 데이터 주소를 비롯한 IPMP 그룹은 IPMP 인터페이스로 나타냅니다.

IPQoS diffserv 모델 표준의 구현과 가상 LAN에 대한 플로우 계정 및 802.1 D 표시를 제공하는 소프트웨어 기능입니다. IPQoS를 사용하면 IPQoS 구성 파일에 정의된 대로 여러 레벨의 네트워크 서비스를 고객과 응용 프로그램에 제공할 수 있습니다.

IPsec IP 보안. IP 데이터그램에 대한 보호를 제공하는 보안 구조입니다.

IPv4 인터넷 프로토콜, 버전 4. IPv4를 종종 IP라고도 합니다. 이 버전은 32비트 주소 공간을 지원합니다.

IPv6 인터넷 프로토콜, 버전 6. IPv6은 128비트 주소 공간을 지원합니다.

local-use 주소 (서브넷 내에 또는 가입자 네트워크 내에) 로컬 경로 지정 가능성 범위만 갖는 유니캐스트 주소입니다. 이 주소는 로컬 또는 전역 고유성 범위를 가질 수도 있습니다.

MAC (메시지 인증 코드) MAC는 데이터 무결성을 보증하고 데이터 발신을 인증합니다. MAC는 도청에 대해 보호되지 않습니다.

MD5 디지털 서명을 포함하여 메시지 인증용으로 사용되는 반복적인 암호화 해시 함수입니다. 이 기능은 1991년 Rivest가 개발했습니다.

MTU Maximum Transmission Unit입니다. 링크를 통해 전송할 수 있는 옥테트 단위의 크기입니다. 예를 들어, 이더넷의 MTU는 1500옥테트입니다.

NAI (Network Access Identifier) 모바일 노드를 고유하게 식별하는 user@domain 형식의 지정입니다.

NAT NAT(Network Address Translation)을 참조하십시오.

NAT (Network Address Translation) NAT의 전체 이름입니다. 한 네트워크 내에 사용된 IP 주소를 다른 네트워크 내에 알려진 다른 IP 주소로 변환합니다. 필요한 전역 IP 주소 수를 제한하는 데 사용됩니다.

NIC (네트워크 인터페이스 카드) 네트워크에 인터페이스로 연결된 네트워크 어댑터 카드입니다. 일부 NIC는 `igb` 카드와 같은 여러 물리적 인터페이스를 가질 수 있습니다.

PFS (완전 순방향 비밀성) PFS에서 데이터 전송을 보호하는 키는 추가 키를 파생하는 데 사용되지 않습니다. 또한 데이터 전송을 보호하는 키의 소스도 추가 키를 파생하는 데 사용되지 않습니다.

PFS는 인증된 키 교환에만 적용됩니다. Diffie-Hellman 알고리즘도 참조하십시오.

PHB (홉별 동작) 트래픽 클래스에 지정된 우선 순위입니다. PHB는 다른 트래픽 클래스와 비교하여 해당 클래스의 어떤 플로우가 우선권을 갖는지 나타냅니다.

PKI 공개 키 기반구조입니다. 인터넷 트랜잭션에 관여한 해당자의 유효성을 확인 및 인증하는 디지털 인증서, 인증 기관 및 기타 등록 기관의 시스템입니다.

RSA 디지털 서명 및 공개 키 암호화 체계를 얻기 위한 방법입니다. 1978년에 개발자 Rivest, Shamir, Adleman이 처음 기술했습니다.

SA SA(보안 연관)를 참조하십시오.

SA (보안 연관) 한 호스트에서 두번째 호스트로 보안 등록 정보를 지정하는 연관입니다.

SADB 보안 연관 데이터베이스(Security Associations Database). 암호화 키 및 암호화 알고리즘을 지정하는 테이블입니다. 키 및 알고리즘은 보안 데이터 전송에 사용됩니다.

SCTP 흐름 제어 전송 프로토콜을 참조하십시오.

SHA-1 보안 해시 알고리즘(Secure Hashing Algorithm). 알고리즘은 2^{64} 미만의 입력 길이에서 작동하여 메시지 다이제스트를 생성합니다. SHA-1 알고리즘은 DSA로 입력됩니다.

site-local-use address 단일 링크에 주소 배정을 위해 사용되는 지정입니다.

SPD SPD(보안 정책 데이터베이스)를 참조하십시오.

SPD (보안 정책 데이터베이스) 패킷에 적용할 보호 레벨을 지정하는 데이터베이스입니다. SPD는 IP 트래픽을 필터링하여 패킷을 폐기할지, 일반 텍스트로 전달할지, IPsec로 보호할지 결정합니다.

SPI SPI(보안 매개변수 색인)를 참조하십시오.

SPI (보안 매개변수 색인) 수신자가 받은 패킷을 해독하기 위해 사용할 보안 연관 데이터베이스(SADB)의 행을 지정하는 정수입니다.

stateful 패킷 필터 활성 연결의 상태를 모니터하여 얻은 정보를 바탕으로 네트워크 패킷이 방화벽을 통과할지 여부를 확인할 수 있는 패킷 필터입니다. 요청 및 회신을 추적하고 일치시키면 stateful 패킷 필터가 요청과 일치하지 않는 회신을 차단할 수 있습니다.

stateless 자동 구성 호스트가 로컬 IPv6 라우터에서 보급한 MAC 주소와 IPv6 접두어를 결합하여 고유의 IPv6 주소를 생성하는 프로세스입니다.

TCP/IP TCP/IP(Transmission Control Protocol/Internet Protocol)는 인터넷의 기본 통신 언어 또는 프로토콜입니다. 개인 네트워크(인트라넷 또는 엑스트라넷)에서 TCP/IP를 통신 프로토콜로 사용할 수도 있습니다.

Triple-DES Triple-Data Encryption Standard입니다. 대칭 키 암호화 방법입니다. Triple-DES에는 키 길이 168비트가 필요합니다. 또한 Triple-DES는 3DES로 작성됩니다.

VPN (가상 사설망) 인터넷과 같은 공중망에서 터널을 사용하는 단일의 안전한 논리적 네트워크입니다.

가상 LAN (VLAN) 장치 이더넷(datalink) 레벨의 IP 프로토콜 스택에서 트래픽 전달을 제공하는 네트워크 인터페이스입니다.

가상 네트워크 소프트웨어 및 하드웨어 네트워크 리소스 및 기능의 조합으로, 단일 소프트웨어 엔티티로 함께 관리됩니다. **내부** 가상 네트워크는 네트워크 리소스를 단일 시스템으로 통합하며, 이를 때때로 "일체형 네트워크"라고도 합니다.

가상 네트워크 인터페이스 (VNIC) 물리적 네트워크 인터페이스에 구성되었는지 여부에 관계없이 가상 네트워크 연결을 제공하는 의사 인터페이스입니다. 배타적 IP 영역과 같은 컨테이너에서 위의 VNIC이 가상 네트워크를 형성하도록 구성됩니다.

개인 주소 인터넷을 통해 경로를 지정할 수 없는 IP 주소입니다. 개인 주소는 인터넷 연결이 필요하지 않은 호스트의 내부 네트워크에서 사용할 수 있습니다. 이러한 주소는 Address Allocation for Private Internets (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`)에 정의되며 종종 "1918" 주소라고도 합니다.

결과 트래픽 측정 결과로 취할 조치입니다. IPQoS 측정자에는 IPQoS 구성 파일에서 정의한 빨강, 노랑, 녹색의 세 가지 결과가 있습니다.

공개 키 암호화 두 개의 다른 키를 사용하는 암호화 시스템입니다. 공개 키는 모든 사람이 알 수 있습니다. 개인 키는 메시지의 수신자만 알 수 있습니다. IKE는 IPsec에 공개 키를 제공합니다.

노드 IPv6에서 호스트든 라우터든 관계없이 IPv6이 사용으로 설정된 시스템입니다.

대기 다른 물리적 인터페이스가 실패하지 않는 한, 데이터 트래픽 전달에 사용되지 않는 물리적 인터페이스입니다.

대칭 키 암호화 메시지를 보낸 사람과 받는 사람이 단일 공통 키를 공유하는 암호화 시스템입니다. 이 공통 키는 메시지를 암호화하고 암호를 해독하는 데 사용됩니다. 대칭 키는 IPsec에서 대량 데이터 전송을 암호화하는 데 사용됩니다. DES는 대칭 키 시스템의 한 예입니다.

데이터 주소 데이터의 소스 또는 대상 주소로 사용할 수 있는 IP 주소입니다. 데이터 주소는 IPMP 그룹의 일부이며 그룹의 모든 인터페이스에서 트래픽을 보내고 받는 데 사용될 수 있습니다. 또한 그룹의 한 인터페이스가 작동하는 경우 IPMP 그룹의 데이터 주소 세트를 계속해서 사용할 수 있습니다.

데이터그램 IP 데이터그램을 참조하십시오.

동적 재구성 (DR) 진행 중인 작업에 거의 또는 전혀 영향을 주지 않고 시스템이 실행 중인 동안 시스템을 재구성할 수 있는 기능입니다. Oracle의 모든 Sun 플랫폼이 DR을 지원하지는 않습니다. Oracle의 일부 Sun 플랫폼은 NIC와 같은 특정 유형의 하드웨어에만 DR을 지원할 수도 있습니다.

동적 패킷 필터 stateful 패킷 필터를 참조하십시오.

등록 이동 중인 경우 모바일 노드가 해당 care-of 주소를 홈 에이전트 및 외래 에이전트에 등록하는 프로세스입니다.

디지털 서명 발신자를 고유하게 식별하는, 전자적으로 전송된 메시지에 첨부된 디지털 코드입니다.

라우터 대개 여러 개의 인터페이스가 있고 경로 지정 프로토콜을 실행하며 패킷을 전달하는 시스템입니다. 시스템이 PPP 링크의 끝점인 경우 하나의 인터페이스만 있는 시스템을 라우터로 구성할 수 있습니다.

라우터 간청 호스트가 다음 일정이 잡힌 시간이 아닌, 즉시 라우터 알림을 생성하도록 라우터에 요청하는 프로세스입니다.

라우터 검색 호스트가 연결된 링크에 상주하는 라우터를 찾는 프로세스입니다.

라우터 알림 정기적으로 또는 라우터 간청 메시지의 응답으로, 라우터가 다양한 링크 및 인터넷 매개변수를 함께 사용하여 자신의 존재를 알리는 프로세스입니다.

로드 확산 인터페이스를 통해 인바운드 또는 아웃바운드 트래픽을 분배하는 프로세스입니다. 로드 확산을 사용하면 더 높은 처리량을 달성할 수 있습니다. 로드 확산은 네트워크 트래픽이 다중 연결을 사용하는 여러 대상으로 흐르고 있을 때만 발생합니다. 두 가지 유형의 로드 확산이 존재합니다. 인바운드 트래픽에는 인바운드 로드 확산을 사용하고 아웃바운드 트래픽에는 아웃바운드 로드 확산을 사용합니다.

링크 계층 IPv4/IPv6 바로 아래의 계층입니다.

링크 로컬 주소 IPv6에서 자동 주소 구성과 같은 목적으로 단일 링크에 주소 배정을 위해 사용되는 지정입니다. 기본적으로 링크 로컬 주소는 시스템의 MAC 주소에서 생성됩니다.

멀티캐스트 주소 특수한 방법으로 인터페이스 그룹을 식별하는 IPv6 주소입니다. 멀티캐스트 주소로 보낸 패킷은 그룹의 모든 인터페이스로 전달됩니다. IPv6 멀티캐스트 주소는 IPv4 브로드캐스트 주소와 기능상 비슷합니다.

멀티홈 호스트 패킷 전달을 수행하지 않는 여러 개의 물리적 인터페이스가 있는 시스템입니다. 멀티홈 호스트는 경로 지정 프로토콜을 실행할 수 있습니다.

모바일 노드 IP 홈 주소를 사용하여 모든 기존 통신을 유지하면서 네트워크 간 연결 지점을 변경할 수 있는 호스트 또는 라우터입니다.

물리적 인터페이스 시스템의 링크 연결입니다. 이 연결은 종종 장치 드라이버와 NIC(네트워크 인터페이스 카드)로 구현됩니다. 일부 NIC는 여러 연결 지점(예: `igb`)을 가질 수 있습니다.

바인딩 테이블 모바일 IP에서 남은 수명과 허가된 시간을 포함하여 홈 주소를 care-of 주소와 연결하는 홈 에이전트 테이블입니다.

방화벽 조직의 사설망이나 인트라넷을 인터넷에서 격리시켜서 외부 침입으로부터 보호할 수 있는 장치 또는 소프트웨어입니다. 방화벽은 패킷 필터링, 프록시 서버 및 NAT(Network Address Translation)를 포함할 수 있습니다.

복구 감지 NIC 또는 NIC에서 어떤 layer-3 장치로의 경로가 실패 후에 올바르게 작동을 시작하는지 감지하는 프로세스입니다.

브로드캐스트 주소 주소의 호스트 부분이 모두 제로(10.50.0.0) 또는 모두 한 비트(10.50.255.255)인 IPv4 네트워크 주소입니다. 로컬 네트워크의 시스템에서 브로드캐스트 주소로 보낸 패킷은 해당 네트워크의 모든 시스템에 전달됩니다.

비대칭 키 암호화 메시지를 암호화 및 해독하기 위해 메시지의 발신자 및 수신자가 서로 다른 키를 사용하는 암호화 시스템입니다. 비대칭 키는 대칭 키 암호화에 대한 보안 채널을 설정하는 데 사용됩니다. Diffie-Hellman 알고리즘은 비대칭 키 프로토콜의 예입니다. 대칭 키 암호화와 대조됩니다.

사용자 우선 순위 class-of-service 표시를 구현하는 3비트 값으로, VLAN 장치의 네트워크에서 이더넷 데이터그램의 전달 방법을 정의합니다.

선택기 네트워크 시스템에서 트래픽을 선택하기 위해 특정 클래스의 패킷에 적용할 기준을 특별히 정의하는 요소입니다. IPQoS 구성 파일의 filter 절에 선택기를 정의합니다.

속임수 메시지가 신뢰된 호스트에서 들어오고 있음을 나타내는 메시지를 IP 주소와 함께 보내어 컴퓨터에 허용되지 않은 액세스를 얻는 것입니다. IP 속임수에 관여하려면 먼저 해커가 다양한 기법을 사용하여 신뢰된 호스트의 IP 주소를 찾은 다음, 패킷이 해당 호스트에서 들어오고 있다고 나타나도록 패킷 헤더를 수정해야 합니다.

스니프 컴퓨터 네트워크에서 도청하는 것입니다. 일반 텍스트 암호, 유선 끄기와 같은 정보를 조사하기 위해 자동화된 프로그램의 일부로 자주 사용됩니다.

스머프 공격 원격 위치에서 IP 브로드캐스트 주소 또는 다중 브로드캐스트 주소로 지정된 ICMP 에코 요청 패킷을 사용하여 심각한 네트워크 혼잡 또는 정전을 일으킵니다.

스택 IP 스택을 참조하십시오.

실패 감지 인터페이스 또는 인터페이스에서 인터넷 계층 장치로의 경로가 더 이상 작동하지 않을 경우 이를 감지하는 프로세스입니다. IPMP(IP Network Multipathing)에는 링크 기반(기본값) 및 프로브 기반(선택 사항)의 두 가지 실패 감지 유형이 포함됩니다.

애니캐스트 그룹 동일한 애니캐스트 IPv6 주소를 가진 인터페이스 그룹입니다. Oracle Solaris IPv6 구현은 애니캐스트 주소 및 그룹의 생성을 지원하지 않습니다. 그러나 Oracle Solaris IPv6 노드가 애니캐스트 그룹으로 트래픽을 보낼 수 있습니다.

애니캐스트 주소 (일반적으로 서로 다른 노드에 속하는) 인터페이스 그룹에 지정된 IPv6 주소입니다. 애니캐스트 주소로 보낸 패킷은 해당 주소를 가진 **가장 가까운** 인터페이스로 경로가 지정됩니다. 패킷의 경로는 경로 지정 프로토콜의 거리 측정을 준수합니다.

양방향 터널 데이터그램을 양방향으로 전송할 수 있는 터널입니다.

에이전트 검색 모바일 IP에서 모바일 노드가 에이전트의 이동 여부, 현재 위치 및 외래 네트워크의 care-of 주소를 판단하는 프로세스입니다.

에이전트 알림 모바일 IP에서 연결된 링크에 대한 존재를 알리기 위해 홈 에이전트와 외래 에이전트가 주기적으로 전송하는 메시지입니다.

역방향 터널 모바일 노드의 care-of 주소에서 시작해서 홈 에이전트에서 끝나는 터널입니다.

연결 물리적 인터페이스 이름에 연결된 장치를 여는 작업입니다. 인터페이스가 연결되면 IP 프로토콜에서 장치를 사용할 수 있도록 스트림이 설정됩니다. 시스템의 현재 세션 동안 인터페이스를 연결하려면 `ifconfig` 명령을 사용합니다.

외래 네트워크 모바일 노드의 홈 네트워크가 아닌 모든 네트워크입니다.

외래 에이전트 모바일 노드가 방문하는 외래 네트워크의 라우터 또는 서버입니다.

유니캐스트 주소 IPv6 사용 노드의 단일 인터페이스를 식별하는 IPv6 주소입니다. 유니캐스트 주소의 부분은 사이트 접두어, 서브넷 ID, 인터페이스 ID입니다.

이동성 바인딩 해당 연결의 남은 수명을 포함한 홈 주소와 care-of 주소의 연결입니다.

이동성 보안 연관 노드 한 쌍 간의 인증 알고리즘과 같은 보안 조치 모음으로, 두 노드 간에 교환되는 모바일 IP 프로토콜 메시지에 적용됩니다.

이동성 에이전트 홈 에이전트 또는 외래 에이전트입니다.

이웃 검색 호스트가 연결된 링크에 상주하는 다른 호스트를 찾을 수 있는 IP 방식입니다.

이웃 알림 이웃 간청 메시지에 대한 응답 또는 link-layer 주소 변경을 공지하기 위해 노드가 청하지 않은 이웃 알림을 보내는 프로세스입니다.

이중 스택 네트워크 계층에 IPv4 및 IPv6이 모두 있는 TCP/IP 프로토콜 스택입니다(스택의 나머지는 동일함). Oracle Solaris 설치 중 IPv6을 사용으로 설정하면 호스트가 TCP/IP의 이중 스택 버전을 수신합니다.

인접 라우터 요청 이웃의 link-layer 주소를 결정하기 위해 노드에서 보낸 간청입니다. 또한 이웃 간청은 캐시된 link-layer 주소에서 이웃에 아직 연결할 수 있는지 확인합니다.

인증 헤더 IP 데이터그램에 (기밀성 없이) 인증 및 무결성을 제공하는 확장 헤더입니다.

자동 구성 호스트가 사이트 접두어 및 로컬 MAC 주소로부터 해당 IPv6 주소를 자동으로 구성하는 프로세스입니다.

재전송 공격 IPsec에서 침입자가 패킷을 캡처하는 공격입니다. 그런 다음 저장된 패킷이 나중에 원본을 대체하거나 반복합니다. 이러한 공격으로부터 보호하려면 패킷을 보호 중인 보안 키의 수명 주기 동안 증분하는 필드를 포함할 수 있습니다.

재지정 라우터에서 특정 대상에 연결하기 위해 더 좋은 첫번째 홉 노드를 호스트에 알려주는 것입니다.

정방향 터널 홈 에이전트에서 시작되고 모바일 노드의 care-of 주소에서 종료되는 터널입니다.

주소 마이그레이션 네트워크 인터페이스 간에 주소를 이동하는 프로세스입니다. 주소 마이그레이션은 인터페이스가 실패한 경우 페일오버 또는 인터페이스가 복구된 경우 페일백의 일환으로 발생합니다.

주소 풀 모바일 IP에서 홈 주소가 필요한 모바일 노드에서 사용하도록 홈 네트워크 관리자가 지정한 일련의 주소입니다.

최소 캡슐화 홈 에이전트, 외래 에이전트, 모바일 노드에서 지원할 수 있는 선택적 형태의 IPv4-in-IPv4 터널링입니다. 최소 캡슐화는 IP-in-IP 캡슐화보다 8 또는 12바이트 정도 오버헤드가 적습니다.

측정자 특정 클래스에 대한 트래픽 플로우의 비율을 측정하는 diffserv 구조의 모듈입니다. IPQoS 구현에는 `tokenmt` 및 `tswtclmt`의 두 측정자가 포함됩니다.

캡슐화 헤더 및 페이로드를 첫번째 패킷에 넣고, 이어서 두번째 패킷의 페이로드에 넣는 프로세스입니다.

키 관리 보안 연관(SA)을 관리하는 방법입니다.

키 저장소 이름 NIC(네트워크 인터페이스 카드)의 저장소 영역 또는 키 저장소에 관리자가 부여하는 이름입니다. 키 저장소 이름을 토큰 또는 토큰 ID라고도 합니다.

터널 캡슐화된 동안 데이터그램에 이어지는 경로입니다. 캡슐화를 참조하십시오.

테스트 주소 프로브의 소스 또는 대상 주소로 사용해야 하고 데이터 트래픽의 소스 또는 대상 주소로 사용하면 안되는 IPMP 그룹의 IP 주소입니다.

패킷 통신 회선을 통해 한 단위로 전송되는 정보 그룹입니다. IP 헤더와 페이로드를 포함합니다.

패킷 필터 방화벽을 통해 지정된 패킷을 허용하도록 구성하거나 허용하지 않도록 구성할 수 있는 방화벽 기능입니다.

패킷 헤더 IP 헤더를 참조하십시오.

페이로드 패킷에 전달된 데이터입니다. 페이로드는 패킷을 대상으로 가져오는 데 필요한 헤더 정보를 포함하지 않습니다.

페일백 복구가 감지된 인터페이스로 네트워크 액세스를 다시 전환하는 프로세스입니다.

페일오버 실패한 인터페이스에서 정상적인 물리적 인터페이스로 네트워크 액세스를 전환하는 프로세스입니다. 네트워크 액세스에는 IPv4 유니캐스트, 멀티캐스트 및 브로드캐스트 트래픽뿐 아니라 IPv6 유니캐스트 및 멀티캐스트 트래픽도 포함됩니다.

표시자 1. 패킷의 전달 방법을 나타내는 값으로 IP 패킷의 DS 필드를 표시하는 diffserv 구조 및 IPQoS의 모듈입니다. IPQoS 구현에서 표시자 모듈은 `dscpmk`입니다.

2. 이더넷 데이터그램의 가상 LAN 태그를 사용자 우선 순위 값으로 표시하는 IPQoS 구현의 모듈입니다. 사용자 우선 순위 값은 VLAN 장치가 포함된 네트워크에서 데이터그램의 전달 방법을 나타냅니다. 이 모듈을 `dlcosmk`라고 합니다.

프로토콜 스택 IP 스택을 참조하십시오.

프록시 서버 클라이언트 응용 프로그램(예: 웹 브라우저)과 다른 서버 사이에 앉은 서버입니다. 요청을 필터링하는 데 사용됩니다. 예를 들어, 특정 웹 사이트에 액세스를 금지할 수 있습니다.

플로우 계산 IPQoS에서 트래픽 플로우에 대한 정보를 누적하고 기록하는 프로세스입니다. IPQoS 구성 파일에 `flowacct` 모듈의 매개변수를 정의하여 플로우 계산을 설정합니다.

필터 IPQoS 구성 파일에 클래스의 특성의 정의하는 규칙 세트입니다. IPQoS 시스템이 IPQoS 구성 파일에서 필터를 준수하는 트래픽 플로우를 처리하기 위해 선택합니다. 패킷 필터를 참조하십시오.

해시 값 텍스트의 문자열에서 생성된 숫자입니다. 해시 함수를 사용하여 전송된 메시지가 변조되지 않았는지 확인할 수 있습니다. MD5 및 SHA-1은 단방향 해시 함수의 예입니다.

헤더 IP 헤더를 참조하십시오.

호스트 패킷 전달을 수행하지 않는 시스템입니다. Oracle Solaris 설치 시 시스템은 기본적으로 호스트가 됩니다. 즉 시스템이 패킷을 전달할 수 없습니다. 호스트는 다중 인터페이스를 가질 수 있지만 일반적으로 하나의 물리적 인터페이스를 가집니다.

홈 네트워크 모바일 노드 홈 주소의 네트워크 접두어와 일치하는 네트워크 접두어를 가진 네트워크입니다.

홈 에이전트 모바일 노드의 홈 네트워크에 있는 라우터 또는 서버입니다.

홈 주소 연장된 동안 모바일 노드에 지정되는 IP 주소입니다. 노드를 인터넷 또는 조직 네트워크의 다른 곳에 연결하는 경우에도 주소는 변경되지 않습니다.

홉 두 호스트를 구분하는 라우터 수를 식별하는 데 사용되는 측정값입니다. 3개의 라우터가 소스 및 대상을 구분하는 경우 호스트가 서로 4홉씩 떨어져 있습니다.

흐름 제어 전송 프로토콜 TCP와 비슷한 방법으로 연결 지향적 통신을 제공하는 전송 계층 프로토콜입니다. 추가적으로, SCTP는 멀티홈 기능을 지원하므로 연결 끝점 중 하나가 여러 개의 IP 주소를 가질 수 있습니다.

색인

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

가

개

게

경

계

공

관

구

권

규

그

기

나

난

네

노

루

릴

링

만

매

멀

메

명

모

목

문

물

미

바

익

인

터

테

토

통

트

파

패

페

포

표

프

플

필