Oracle® Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화

부품 번호: E25835
2011년 12월

ORACLE

120410@25097

목차

머리말

Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화입니다. 이 책은 Oracle Solaris 시스템 관리 정보의 중요한 부분을 다루며 전체 14권 중 일부입니다. 이 책에서는 Oracle Solaris를 이미 설치했다고 가정합니다. 네트워크를 구성할 준비가 되었거나 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: Hardware Compatibility Lists**를 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: Hardware Compatibility Lists**를 참조하십시오.

이 책의 대상

이 책은 네트워크에 구성된 Oracle Solaris 실행 시스템의 관리 책임자를 대상으로 작성되었습니다. 이 책을 사용하려면 2년 이상의 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.

책 제목	내용
SPARC 플랫폼에서 Oracle Solaris 부트 및 종료	SPARC 플랫폼에서 시스템 부트 및 종료, 부트 서비스 관리, 부트 동작 수정, ZFS에서 부트, 부트 아카이브 관리 및 부트 문제 해결

책 제목	내용
x86 플랫폼에서 Oracle Solaris 부트 및 종료	x86 플랫폼에서 시스템 부트 및 종료, 부트 서비스 관리, 부트 동작 수정, ZFS에서 부트, 부트 아카이브 관리 및 부트 문제 해결
Oracle Solaris 관리: 일반 작업	Oracle Solaris 명령 사용, 시스템 부트 및 종료, 사용자 계정 및 그룹 관리, 서비스, 하드웨어 오류, 시스템 정보, 시스템 리소스 및 시스템 성능 관리, 소프트웨어, 인쇄, 콘솔 및 터미널 관리, 시스템 및 소프트웨어 문제 해결
Oracle Solaris 관리: 장치 및 파일 시스템	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 및 IPQoS
Oracle Solaris Administration: Naming and Directory Services	NIS에서 LDAP으로 전환을 비롯한 DNS, NIS 및 LDAP 이름 지정 및 디렉토리 서비스
Oracle Solaris 관리: 네트워크 인터페이스 및 네트워크 가상화	WiFi 무선을 포함하는 자동 및 수동 IP 인터페이스 구성, 브릿지, VLAN, 통합, LLDP 및 IPMP 관리, 가상 NIC 및 리소스 관리
Oracle Solaris 관리: 네트워크 서비스	웹 캐시 서버, 시간 관련 서비스, 네트워크 파일 시스템(NFS 및 Autofs), 메일, SLP, PPP
Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리	응용 프로그램이 사용 가능한 시스템 리소스를 사용하는 방식을 제어할 수 있는 리소스 관리 기능, 운영 체제 서비스를 가상화하여 응용 프로그램을 실행하기 위한 격리된 환경을 만드는 Oracle Solaris Zones 소프트웨어 분할 기술, Oracle Solaris 11 커널에서 실행되는 Oracle Solaris 10 환경을 호스트하는 Oracle Solaris 10 영역
Oracle Solaris 관리: 보안 서비스	감사, 장치 관리, 파일 보안, BART, Kerberos 서비스, PAM, 암호화 프레임워크, 키 관리, 권한, RBAC, SASL, 보안 셸 및 바이러스 검사
Oracle Solaris Administration: SMB and Windows Interoperability	SMB 클라이언트가 SMB 공유를 사용할 수 있도록 Oracle Solaris 시스템을 구성할 수 있는 SMB 서비스, SMB 공유에 액세스할 수 있도록 해주는 SMB 클라이언트, 사용자 및 그룹 ID를 Oracle Solaris 시스템과 Windows 시스템 간에 매핑할 수 있도록 해주는 기본 ID 매핑 서비스
Oracle Solaris 관리: ZFS 파일 시스템	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 구성 및 관리	Trusted Extensions와 관련된 시스템 설치, 구성 및 관리
Oracle Solaris 11 보안 지침	영역, ZFS 및 Trusted Extensions와 같은 보안 기능에 대한 사용 시나리오와 Oracle Solaris 시스템의 보안 설정

책 제목	내용
Oracle Solaris 10에서 Oracle Solaris 11로 전환	설치, 장치, 디스크 및 파일 시스템 관리, 소프트웨어 관리, 네트워킹, 시스템 관리, 보안, 가상화, 데스크탑 기능, 사용자 계정 관리, 사용자 환경 에뮬레이트된 볼륨, 문제 해결 및 데이터 복구 영역에서 Oracle Solaris 10에서 Oracle Solaris 11로의 전환을 위한 시스템 관리 정보 및 예 제공

타사 웹 사이트

주 – 본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제 3자로부터 제공되는 컨텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 컨텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 컨텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

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=info를 참조하거나, 청각 장애가 있는 경우 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs를 방문하십시오.

활자체 규약

다음 표는 이 책에서 사용되는 활자체 규약에 대해 설명합니다.

표 P–1 활자체 규약

활자체 또는 기호	설명	예제
`AaBbCc123`	명령 및 파일, 디렉토리 이름; 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:`

표 P–1 활자체 규약 *(계속)*

활자체 또는 기호	설명	예제
AaBbCc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	`rm` *filename* 명령을 사용하여 파일을 제거합니다.
AaBbCc123	책 제목, 장, 절	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하면 **안 됩니다**. **주:** 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

명령 예의 셸 프롬프트

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

표 P–2 셸 프롬프트

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

◆ ◆ ◆ 1 장

1 네트워킹 스택 개요

이 장에서는 Oracle Solaris의 네트워크 관리를 소개합니다. 인터페이스의 기반이 되는 상관 관계, 인터페이스가 구성된 데이터 링크 및 네트워크 장치에 대해 설명합니다. 데이터 링크에 대한 유연한 이름 지원도 자세히 설명합니다.

이 Oracle Solaris 릴리스의 네트워크 구성

이 릴리스에서 네트워크가 구성된 방식과 관련하여 이전 Oracle Solaris 릴리스와 구분하는 차이점은 다음과 같습니다.

- 네트워크 구성이 프로파일에서 관리됩니다. 시스템에서 작동하는 구성 유형은 구성 프로파일이 활성 상태인 네트워크에 따라 달라집니다. 제1부을 참조하십시오.
- 네트워킹 스택의 계층 2에 있는 데이터 링크는 `dladm` 명령을 사용하여 관리됩니다. 이 명령이 이전 `ifconfig` 명령 옵션을 대체하여 데이터 링크 등록 정보를 구성합니다. 따라서 링크 통합, VLAN 및 IP 터널의 구성도 변경됩니다. 8 장, “데이터 링크 구성 및 관리”, 12 장, “링크 통합 관리” 및 13 장, “VLAN 관리”를 참조하십시오. **Oracle Solaris 관리: IP 서비스**의 6 장, “IP 터널 구성”을 참조하십시오.
- 데이터 링크 이름이 더 이상 해당 하드웨어 드라이버에 바인딩되지 않습니다. 따라서 데이터 링크에 기본적으로 `net0`, `net1` 등의 일반 링크 이름이 지정됩니다. 24 페이지 “네트워크 장치 및 데이터 링크 이름”을 참조하십시오.
- 네트워킹 스택의 계층 3에 있는 IP 인터페이스는 `ipadm` 명령을 사용하여 관리됩니다. 이 명령이 이전 `ifconfig` 명령 옵션을 대체하여 IP 인터페이스를 구성합니다. 9 장, “IP 인터페이스 구성”을 참조하십시오.
- IPMP 그룹이 IP 인터페이스로 구현되므로 `ipadm` 명령을 사용하여 유사하게 구성됩니다. 또한 IPMP 관련 정보와 통계를 가져올 수 있는 `ipmpstat`가 도입되었습니다. 14 장, “IPMP 소개” 및 15 장, “IPMP 관리”를 참조하십시오.
- 가상화가 네트워크 장치 레벨에서 구현됩니다. 따라서 VNIC를 구성하고 효율성 향상을 위해 네트워크 리소스 사용을 관리할 수 있습니다. 제3부를 참조하십시오.

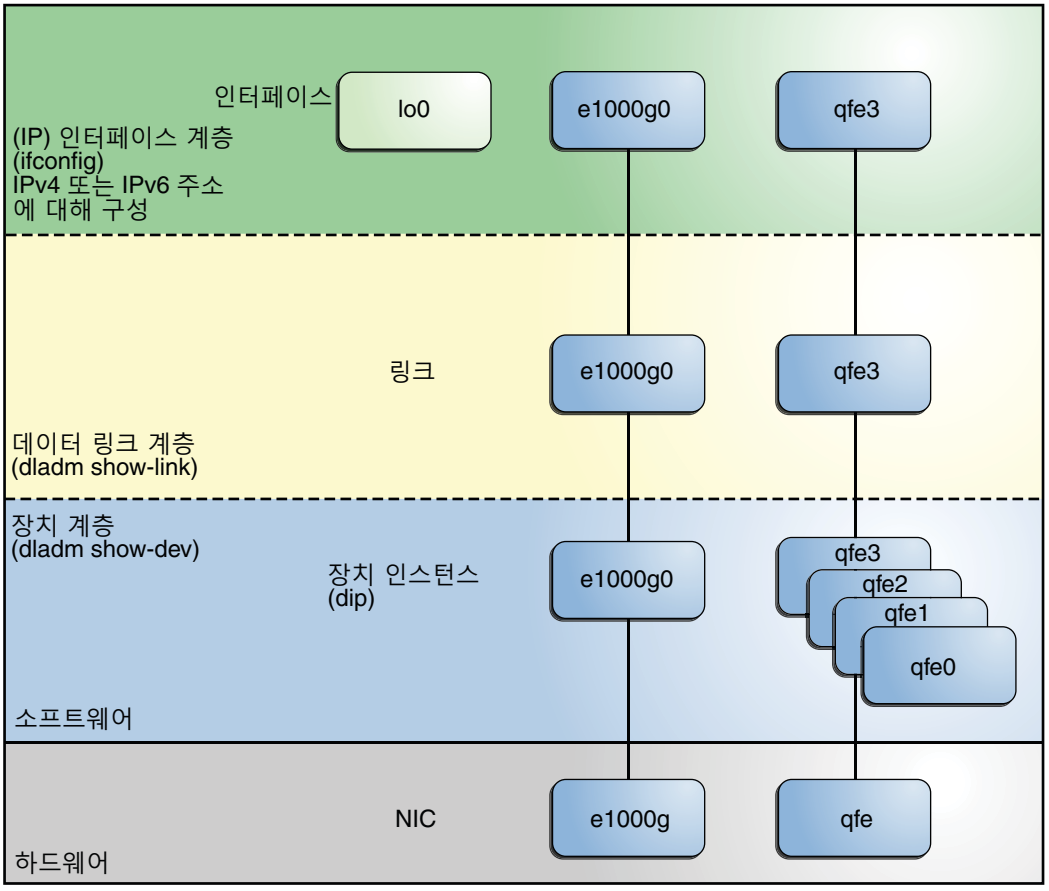
Oracle Solaris의 네트워크 스택

네트워크 인터페이스는 시스템과 네트워크 간의 연결을 제공합니다. 이러한 인터페이스는 시스템의 하드웨어 장치 인스턴스에 해당하는 데이터 링크에 구성됩니다. 네트워크 하드웨어 장치를 *NIC(***네트워크 인터페이스 카드***)* 또는 **네트워크 어댑터**라고도 합니다. NIC는 시스템 구입 시 시스템에 미리 내장되어 있을 수 있습니다. 하지만 시스템에 추가할 NIC를 개별적으로 구입할 수도 있습니다. 특정 NIC는 카드에 있는 단일 인터페이스만 사용합니다. 네트워크 작업 수행을 위해 구성할 수 있는 여러 인터페이스가 포함된 브랜드도 있습니다.

현재 모델의 네트워크 스택에서는 소프트웨어 계층의 인터페이스와 링크가 하드웨어 계층의 장치에 구축됩니다. 즉, 하드웨어 계층의 하드웨어 장치 인스턴스에 해당하는 링크가 데이터 링크 계층에 있고 구성된 인터페이스가 인터페이스 계층에 있습니다. 다음 그림에서는 네트워크 장치, 해당 데이터 링크 및 IP 인터페이스 간의 이러한 일대일 관계를 보여줍니다.

주 – TCP/IP 스택에 대한 자세한 설명은 **System Administration Guide: IP Services**의 1 장, “Oracle Solaris TCP/IP Protocol Suite (Overview)”을 참조하십시오.

그림 1–1 네트워크 장치, 링크 및 인터페이스를 보여주는 네트워크 스택 - Oracle Solaris 10 모델



이 그림에서는 하드웨어 계층의 NIC 두 개를 보여줍니다. `e1000`에는 단일 장치 인스턴스 `e1000g0`이 있고 `qfe`에는 `qfe0`에서 `qfe3`까지 여러 장치 인스턴스가 있습니다. `qfe0`에서 `qfe2`까지의 장치는 사용되지 않습니다. 장치 `e1000g`와 `qfe3`은 사용되며 데이터 링크 계층에 해당 링크 `e1000g`와 `qfe3`이 있습니다. 그림에서 IP 인터페이스의 이름은 해당 기본 하드웨어 `e1000g`와 `qfe3`을 따서 지정되었습니다. 이러한 인터페이스를 IPv4 또는 IPv6 주소로 구성하여 두 유형의 네트워크 트래픽을 모두 호스트할 수 있습니다. 인터페이스 계층에 루프백 인터페이스 `lo0`이 있는 것도 확인합니다. 예를 들어, 이 인터페이스는 IP 스택이 제대로 작동하는지 테스트하는 데 사용됩니다.

스택의 각 계층에서는 서로 다른 관리 명령이 사용됩니다. 예를 들어, 시스템에 설치된 하드웨어 장치는 `dladm show-dev` 명령을 통해 나열됩니다. 데이터 링크 계층의 링크에 대한 정보는 `dladm show-link` 명령을 통해 표시됩니다. `ifconfig` 명령은 인터페이스 계층의 IP 인터페이스 구성을 보여줍니다.

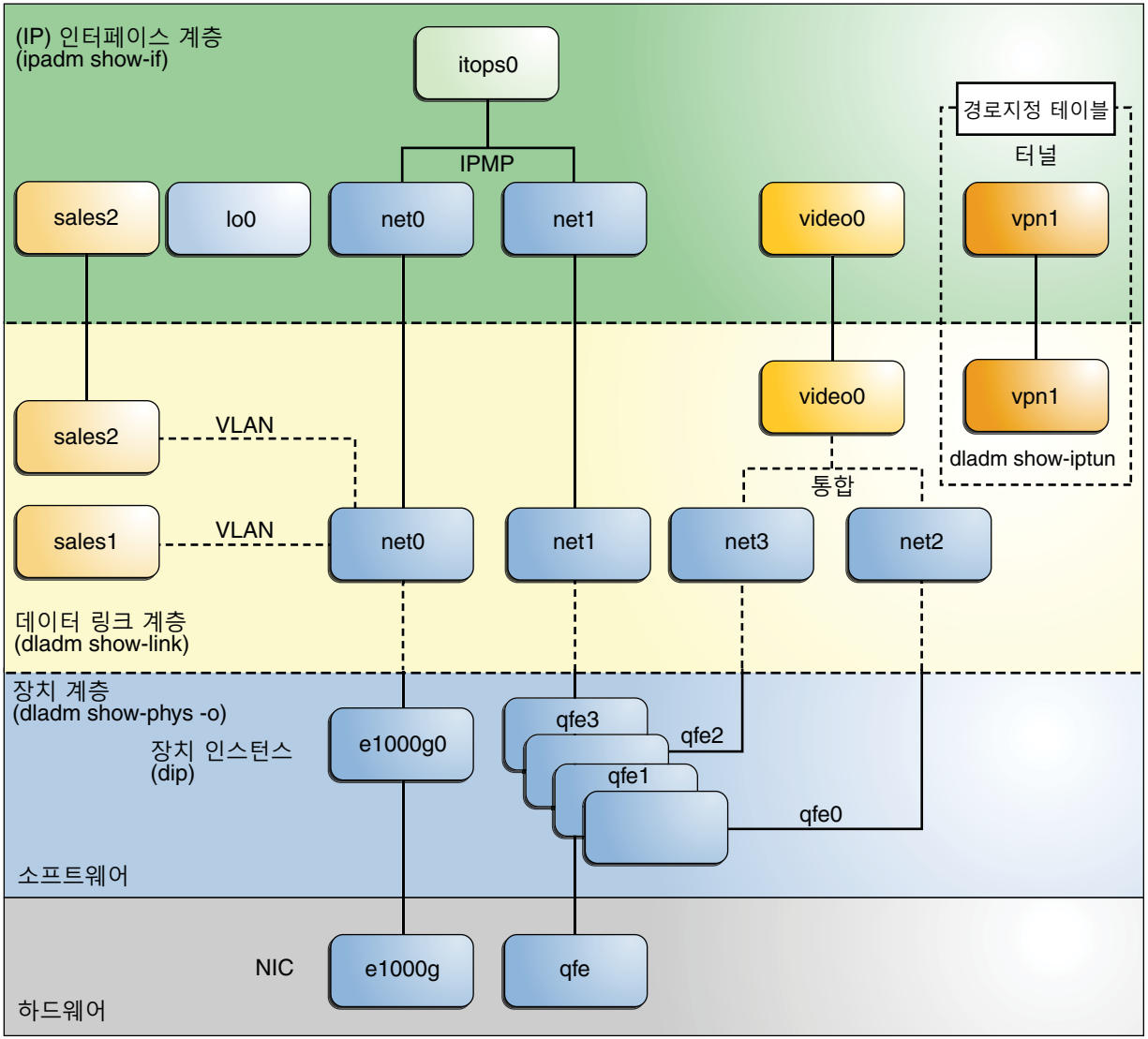
이 모델에는 장치, 데이터 링크 및 인터페이스를 바인딩하는 일대일 관계가 있습니다. 이 관계는 네트워크 구성이 하드웨어 구성과 네트워크 토폴로지에 종속됨을 의미합니다. NIC 교체 또는 네트워크 토폴로지 변경과 같은 변경 사항이 하드웨어 계층에서 구현된 경우 인터페이스를 재구성해야 합니다.

Oracle Solaris 11에는 하드웨어, 데이터 링크 및 인터페이스 계층 간의 기본 관계가 유지되는 네트워크 스택의 구현이 도입되었습니다. 하지만 소프트웨어 계층과 하드웨어 계층은 분리됩니다. 이러한 분리를 사용할 경우 소프트웨어 레벨의 네트워크 구성이 하드웨어 계층의 네트워크 토폴로지나 칩셋에 더 이상 바인딩되지 않습니다. 이 구현에서는 다음과 같은 방식으로 네트워크 관리의 유연성이 향상됩니다.

- 네트워크 구성이 하드웨어 계층에서 발생할 수 있는 변경 사항으로부터 보호됩니다. 기본 하드웨어를 제거하는 경우에도 링크 및 인터페이스 구성이 유지됩니다. 두 NIC가 동일한 유형인 경우 교체 NIC에 동일한 구성을 재적용할 수 있습니다.
- 네트워크 하드웨어 구성에서 네트워크 구성을 분리하면 데이터 링크 계층에 사용자 정의 링크 이름을 사용할 수 있습니다.
- 데이터 링크 계층의 추상화를 사용할 경우 VLAN, VNIC, 물리적 장치, 링크 통합 및 IP 터널과 같은 여러 네트워킹 추상화 또는 구성이 공통된 관리 엔티티인 데이터 링크에 통합됩니다.

다음 그림에서는 이러한 네트워크 구성이 네트워킹 스택에 생성되는 방식을 보여줍니다.

그림 1–2 네트워크 장치, 링크 및 인터페이스를 보여주는 네트워크 스택 - Oracle Solaris 11 모델



이 그림의 구성은 28 페이지 “기타 링크 유형의 관리”에서 자세히 설명합니다.

네트워크 장치 및 데이터 링크 이름

관리 관점에서 관리자는 **데이터 링크** 위에 IP 인터페이스를 만듭니다. 데이터 링크는 OSI(Open Systems Interconnection) 모델의 두번째 계층에 링크 객체를 나타냅니다. **물리적 링크**는 장치와 직접 연결되며 장치 이름을 갖습니다. 장치 이름은 근본적으로 장치 인스턴스 이름이며, 드라이버 이름과 장치 인스턴스 번호로 구성됩니다. 인스턴스 번호는 시스템에서 해당 드라이버를 사용하는 NIC 수에 따라 0에서 *n*까지의 값을 가질 수 있습니다.

예를 들어, 호스트 시스템과 서버 시스템에서 모두 주 NIC로 사용되는 기가비트 이더넷 카드를 고려해 보십시오. 이 NIC의 일반 드라이버 이름은 `bge`와 `e1000g`입니다. 주 NIC로 사용될 경우 기가비트 이더넷 인터페이스는 `bge0` 또는 `e1000g0`과 같은 장치 이름을 갖습니다. 기타 드라이버 이름은 `nge`, `nxge` 등입니다.

이 Oracle Solaris 릴리스에서는 장치 인스턴스 이름이 기본 하드웨어에 계속 종속됩니다. 하지만 이러한 장치의 위에 있는 데이터 링크는 유사하게 바인딩되지 않고 의미 있는 이름이 지정될 수 있습니다. 예를 들어, 관리자는 장치 인스턴스 `e1000g0` 위에 있는 데이터 링크에 `itops0`이라는 이름을 지정할 수 있습니다. 이 Oracle Solaris 릴리스에서는 기본적으로 데이터 링크에 일반 이름이 제공됩니다. 일반 이름을 가진 데이터 링크와 해당 장치 인스턴스 간의 매핑을 표시하려면 `dladm sho-phys` 하위 명령을 사용합니다.

기본 일반 링크 이름

이 Oracle Solaris 릴리스를 시스템에 처음 설치하면 Oracle Solaris에서 자동으로 시스템의 모든 물리적 네트워크 장치에 일반 링크 이름을 제공합니다. 이 이름 지정은 `net #` 이름 지정 규약을 사용합니다. 여기서 `#`은 인스턴스 번호입니다. 이 인스턴스 번호는 각 장치마다 증가합니다(예: `net0`, `net1`, `net2` 등).

일반 링크 이름이나 유연한 링크 이름은 다음 예와 같은 네트워크 구성에서 이점을 제공합니다.

- 단일 시스템 내에서는 동적 재구성이 더 쉽습니다. 지정된 NIC에 대해 설정된 네트워크 구성이 다른 NIC 교체에 상속될 수 있습니다.
- 네트워크 설정과 관련하여 영역 마이그레이션이 덜 복잡합니다. 대상 시스템의 링크가 마이그레이션 전에 영역에 할당된 링크와 동일한 이름을 공유하는 경우 마이그레이션된 시스템의 영역이 해당 네트워크 구성을 유지합니다. 따라서 마이그레이션 후에 영역에서 추가 네트워크 구성을 수행할 필요가 없습니다.
- 일반 이름 지정 체계는 SC(시스템 구성) 매니페스트에 지정된 네트워크 구성에 사용됩니다. 일반적으로 모든 시스템에 대해 주 네트워크 데이터 링크의 이름이 `net0`으로 지정됩니다. 따라서 `net0`의 구성을 지정하는 여러 시스템에 일반 SC 매니페스트를 사용할 수 있습니다.
- 또한 데이터 링크 관리가 유연해집니다. 예를 들어, 그림 1–2와 같이 데이터 링크가 제공하는 특정 기능을 반영하기 위해 데이터 링크의 이름을 추가로 사용자 정의할 수 있습니다.

다음 표에서는 하드웨어(NIC), 장치 인스턴스, 링크 이름 및 링크의 인터페이스 간에 새로 지정된 이름을 보여줍니다. OS에서 자동으로 데이터 링크의 이름을 제공합니다.

하드웨어(NIC)	장치 인스턴스	링크의 지정된 이름	IP 인터페이스
e1000g	e1000g0	net0	net0
qfe	qfe1	net1	net1

위의 표와 같이 장치 인스턴스 이름이 하드웨어 기반으로 유지되는 동안 OS가 설치 후에 데이터 링크의 이름을 바꾼 것입니다.

데이터 링크에 대한 일반 이름 지정

Oracle Solaris에서는 특정 기준에 따라 모든 데이터 링크에 자동으로 일반 이름이 지정됩니다. 모든 장치가 동일한 접두어 net을 공유합니다. 하지만 인스턴스 번호는 다음을 기준으로 할당됩니다.

- 물리적 네트워크 장치는 매체 유형에 따라 정렬됩니다. 이 경우 특정 유형이 다른 유형보다 높은 우선 순위를 갖습니다. 매체 유형은 다음과 같이 우선 순위의 내림차순으로 정렬됩니다.
  1. 이더넷
  2. IP over IB(Infiniband 장치)
  3. Ethernet over IB
  4. WiFi
- 장치가 매체 유형에 따라 그룹화 및 정렬된 후 물리적 위치를 기준으로 장치가 추가로 정렬됩니다. 이 경우 내장 장치가 주변 장치보다 선호됩니다.
- 매체 유형과 위치를 기준으로 우선 순위가 더 높은 장치에 더 낮은 인스턴스 번호가 할당됩니다.

기준에 따라 하위 마더보더나 ioboard의 이더넷 장치, hostbridge, PCIe rootcomplex, 버스, 장치 및 기능에 다른 장치보다 앞선 순위가 지정됩니다.

링크 이름, 장치 및 위치의 지정된 이름을 표시하려면 다음과 같이 dladm show-phys 명령을 사용합니다.

```
# dladm show-phys -L
LINK         DEVICE       LOCATION
net0         e1000g0      MB
net1         e1000g1      MB
net2         e1000g2      MB
net3         e1000g3      MB
net4         ibp0         MB/RISER0/PCIE0/PORT1
net5         ibp1         MB/RISER0/PCIE0/PORT2
net6         eoib2        MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
net7         eoib4        MB/RISER0/PCIE0/PORT2/cloud-nm2gw-2/1A-ETH-2
```

일반 링크 이름이 지정되는 방식 사용자 정의

Oracle Solaris는 링크 이름을 지정할 때 접두어 net을 사용합니다. 하지만 eth와 같은 임의의 사용자 정의 접두어를 대신 사용할 수 있습니다. 원하는 경우 중립 링크 이름의 자동 지정을 사용 안함으로 설정할 수도 있습니다.

주의 – Oracle Solaris를 설치하기 **전에** 일반 링크 이름이 자동으로 지정되는 방식을 사용자 정의해야 합니다. 설치 후에는 기존 구성을 해제하지 않고 기본 링크 이름을 사용자 정의할 수 없습니다.

자동 링크 이름 지정을 사용 안함으로 설정하거나 링크 이름의 접두어를 사용자 정의하려면 AI(자동 설치) 프로그램이 사용하는 시스템 구성 매니페스트에서 다음 등록 정보를 설정합니다.

```
<service name="network/datalink-management"
   version="1" type="service">
      <instance name="default enabled="true">
         <property_group name='linkname-policy'
            type='application'>
               <propval name='phys-prefix' type='astring'
                  value='net'/>
         </property_group>
      </instance>
</service
```

기본적으로 phys-prefix 값은 강조 표시된 것처럼 net으로 설정됩니다.

- 자동 이름 지정을 사용 안함으로 설정하려면 phys-prefix에 대해 설정된 값을 모두 제거합니다. 자동 이름 지정을 사용 안함으로 설정하는 경우 데이터 링크 이름은 bge0, e1000g0 등의 연결된 하드웨어 드라이버를 기반으로 합니다.
- net 이외의 접두어를 사용하려면 phys-prefix 값으로 eth와 같은 새 접두어를 지정합니다.

phys-prefix에 제공된 값이 잘못된 경우 해당 값은 무시됩니다. 데이터 링크 이름이 bge0, e1000g0 등의 연결된 하드웨어 드라이버에 따라 지정됩니다. 유효한 링크 이름에 대한 규칙은 28 페이지 “유효한 링크 이름 규칙”을 참조하십시오.

업그레이드된 시스템의 링크 이름

이 Oracle Solaris 릴리스가 새로 설치된 시스템에서는 데이터 링크에 자동으로 net0에서 net *N-1*까지의 이름이 지정됩니다. 여기서 *N*은 총 네트워크 장치 수를 나타냅니다.

Oracle Solaris 11 Express에서 업그레이드하는 경우에는 이 내용이 적용되지 않습니다. 업그레이드된 시스템에서는 데이터 링크의 업그레이드 전 이름을 유지합니다. 이러한 이름은 기본 하드웨어 기반 이름이거나 관리자가 업그레이드 전에 데이터 링크에

지정한 사용자 정의 이름입니다. 또한 업그레이드된 시스템에서는 이후에 추가된 새 네트워크 장치도 중립 이름을 수신하는 대신 기본 하드웨어 기반 이름을 유지합니다. 업그레이드된 시스템의 이 동작은 OS에서 지정한 중립 이름이 다른 하드웨어 기반 이름이나 업그레이드 전에 관리자가 지정한 사용자 정의 이름과 혼합되지 않도록 합니다.

이 Oracle Solaris 릴리스가 있는 시스템에서는 하드웨어 기반 이름과 OS 제공 링크 이름을 사용하려는 다른 이름으로 대체할 수 있습니다. 일반적으로 OS가 지정한 기본 링크 이름으로 시스템 네트워크 구성을 만들어도 됩니다. 하지만 링크 이름 변경을 선택한 경우 다음 절에 설명된 중요한 고려 사항을 알아야 합니다.

하드웨어 기반 링크 이름 대체

시스템 링크에 하드웨어 기반 이름이 있는 경우 가능한 일반 이름으로 이러한 링크의 이름을 바꾸십시오. 링크의 하드웨어 기반 이름을 유지하면 나중에 이러한 물리적 장치를 제거하거나 교체할 때 혼동이 생길 수 있습니다.

예를 들어, 장치 `bge0`과 연결된 링크 이름 `bge0`을 유지합니다. 이 링크 이름을 참조하여 모든 링크 구성을 수행합니다. 나중에 NIC `bge`를 NIC `e1000g`로 교체할 수 있습니다. 이전 장치의 링크 구성을 새 NIC `e1000g0`에 재적용하려면 `e1000g0`에 링크 이름 `bge0`을 재지정해야 합니다. 하드웨어 기반 링크 이름 `bge0`을 연결된 다른 NIC `e1000g0`과 조합하면 혼동이 생길 수 있습니다. 하드웨어 기반이 아닌 이름을 사용하면 연결된 장치에서 링크를 구분하는 데 도움이 됩니다.

링크 변경 변경 시의 주의 사항

하드웨어 기반 링크 이름은 대체하는 것이 좋지만 링크 이름을 바꾸기 전에 신중하게 계획해야 합니다. 장치의 링크 이름을 변경할 경우 새 이름이 기존의 모든 연결된 구성으로 자동 전파되지 않습니다. 다음 예에서는 링크 이름 변경 시의 위험을 보여줍니다.

- IP 필터 구성의 일부 규칙은 특정 링크에 적용됩니다. 링크 이름을 변경할 경우 필터 규칙은 링크의 원래 이름을 계속 참조합니다. 따라서 링크 이름을 바꾼 후에는 이러한 규칙이 예상대로 동작하지 않습니다. 새 링크 이름을 사용하여 링크에 적용할 필터 규칙을 조정해야 합니다.
- 네트워크 구성 정보의 내보내기 가능성을 고려해 보십시오. 앞에서 설명했듯이 OS가 제공한 기본 `net #` 이름을 사용하면 영역을 마이그레이션하고 네트워크 구성을 다른 시스템으로 쉽게 내보낼 수 있습니다. 대상 시스템의 네트워크 장치 이름이 `net0`, `net1` 등의 일반 이름으로 지정된 경우 데이터 링크의 네트워크 구성이 영역에 간단히 상속됩니다. 이 데이터 링크의 이름은 영역에 할당된 데이터 링크와 일치합니다.

따라서 일반적인 규칙으로, 데이터 링크의 이름을 임의로 바꾸지 마십시오. 데이터 링크의 이름을 바꾸는 경우 링크 이름이 변경된 후에도 링크의 연결된 구성이 계속해서 모두 적용되는지 확인합니다. 링크 이름 변경으로 인해 영향을 받을 수 있는 일부 구성은 다음과 같습니다.

- IP 필터 규칙

- 구성 파일에 지정된 IP 구성(예: `/etc/dhcp.*`)
- Oracle Solaris 11 영역
- `autopush` 구성

주 – 링크 이름을 바꿀 때 `autopush` 구성을 변경할 필요는 없습니다. 하지만 링크 이름을 바꾼 후 링크별 `autopush` 등록 정보를 사용하여 구성이 작동하는 방식을 알고 있어야 합니다. 자세한 내용은 158 페이지 "데이터 링크에 STREAMS 모듈을 설정하는 방법"을 참조하십시오.

유효한 링크 이름 규칙

링크 이름을 지정하는 경우 다음 규칙을 따릅니다.

- 링크 이름은 문자열과 *PPA(**물리적 연결 지점**)* 번호로 구성됩니다.
- 이름은 다음 제약 조건을 준수해야 합니다.
  - 이름은 3-8자로 구성됩니다. 하지만 이름에 최대 16자를 사용할 수 있습니다.
  - 이름에 유효한 문자는 영숫자(a-z, 0–9) 및 밑줄('_')입니다.

주의 – 링크 이름에 대문자를 사용하지 마십시오.

- 각 데이터 링크에 한 번에 한 개의 링크 이름만 있어야 합니다.
- 각 데이터 링크에 시스템 내에서 고유한 링크 이름이 있어야 합니다.

주 – 추가된 제한 사항으로, `lo0`을 유연한 링크 이름으로 사용할 수 없습니다. 이 이름은 IP 루프백 인터페이스를 식별하는 데 예약됩니다.

네트워크 설정 내의 링크 기능은 링크 이름을 지정할 때 유용한 참조가 될 수 있습니다. 예를 들어, `netmgt0`은 네트워크 관리에 전용으로 사용되는 링크일 수 있습니다. `Upstream2`는 ISP에 연결하는 링크일 수 있습니다. 혼동을 방지하기 위한 일반적인 규칙으로, 알려진 장치의 이름을 링크에 지정하지 **마십시오**.

기타 링크 유형의 관리

네트워크 구성과 네트워크 하드웨어 구성을 분리하면 다른 유형의 링크 구성에도 동일한 유연성이 적용됩니다. 예를 들어, VLAN(가상 LAN), 링크 통합 및 IP 터널에 관리상 선택한 이름을 지정한 다음 해당 이름을 참조하여 구성할 수 있습니다. 네트워크 구성이 삭제되지 않은 경우 네트워크 재구성이 필요하지 않으므로 DR(동적 재구성)을 수행하여 하드웨어 장치를 교체하는 등의 기타 관련된 작업을 수행하기 쉬워집니다.

다음 그림에서는 장치, 링크 유형 및 해당 인터페이스 간의 상관 관계를 보여줍니다.

주 – 이 그림에서 데이터 링크는 시스템에서 수행하는 특정 기능에 따라 이름이 지정됩니다(예: `video0` 또는 `sales2`). 이 그림은 데이터 링크의 이름을 지정할 수 있는 유연성을 강조하기 위한 것입니다. 하지만 `net0`과 같이 OS가 제공한 기본 중립 이름을 사용하는 것이 좋습니다.

또한 이 그림에서는 관리상 선택한 이름을 네트워크 설정에서 사용하는 방법의 샘플을 제공합니다.

- VLAN은 `net0` 링크에 구성됩니다. 또한 이러한 VLAN에 `sales1` 및 `sales2`와 같은 사용자 정의 이름이 지정됩니다. VLAN `sales2`의 IP 인터페이스가 연결되고 작동합니다.
- 장치 인스턴스 `qfe0`과 `qfe2`는 비디오 트래픽을 서비스하는 데 사용됩니다. 이에 따라 데이터 링크 계층의 해당 링크에 `subvideo0` 및 `subvideo1`이라는 이름이 지정됩니다. 이러한 두 링크가 통합되어 비디오 피드를 호스트합니다. 링크 통합도 `video0`이라는 고유한 사용자 정의 이름을 소유합니다.
- 서로 다른 기본 하드웨어(`e1000g` 및 `qfe`)가 있는 두 인터페이스(`net0` 및 `net1`)가 IPMP 그룹(`itops0`)으로 그룹화되어 전자 메일 트래픽을 호스트합니다.

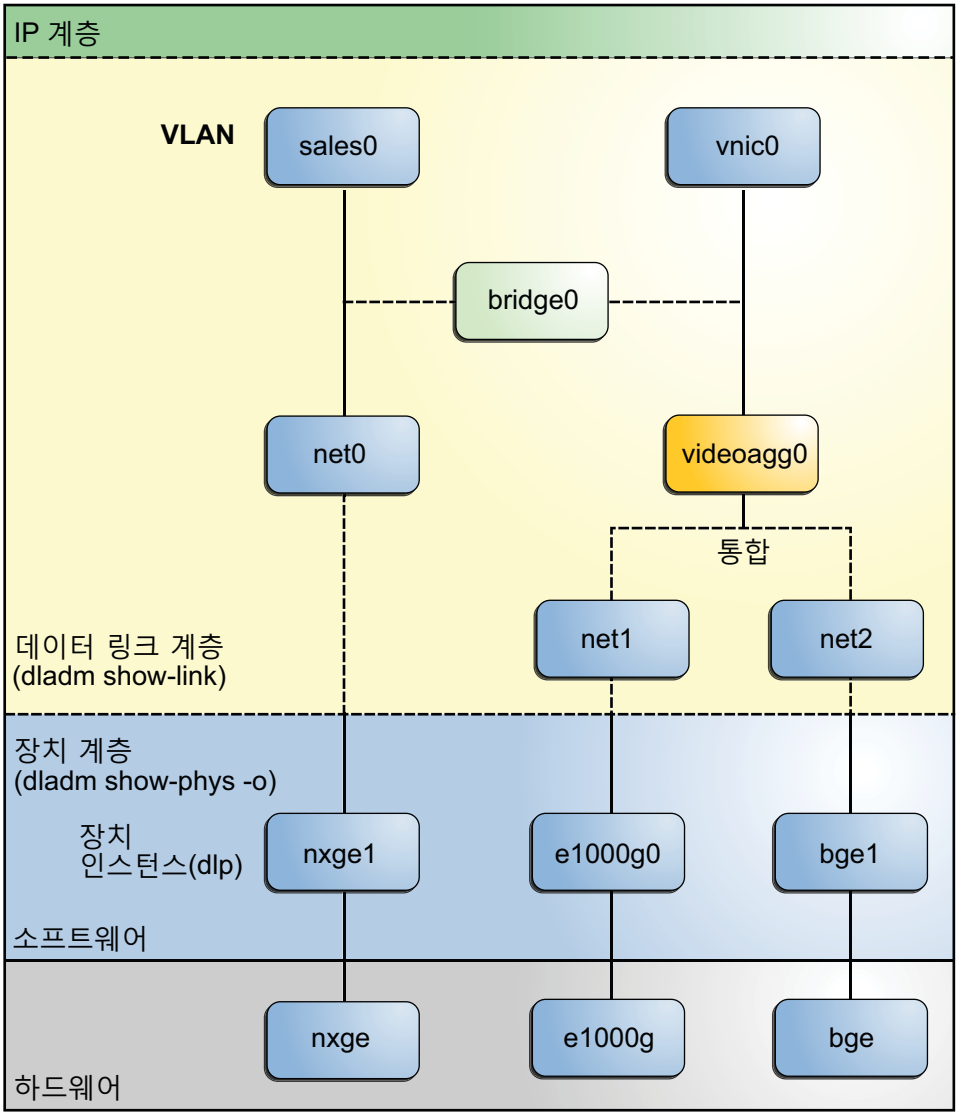
  주 – IPMP 인터페이스는 데이터 링크 계층의 링크가 아니지만 링크와 마찬가지로 해당 인터페이스에 사용자 정의 이름을 지정할 수도 있습니다. IPMP 그룹에 대한 자세한 내용은 14 장, “IPMP 소개”를 참조하십시오.

- 두 인터페이스에는 기본 장치가 없습니다. 터널 `vpn1`은 VPN 연결을 위해 구성되고 `lo0`은 IP 루프백 작업을 위해 구성됩니다.

이 그림의 모든 링크 및 인터페이스 구성은 기본 하드웨어의 구성에 독립적입니다. 예를 들어, `qfe` 카드가 교체된 경우 비디오 트래픽에 대한 `video0` 인터페이스 구성이 유지되며 나중에 교체 NIC에 적용할 수 있습니다.

다음 그림에서는 브릿지 구성을 보여줍니다. 두 인터페이스 `net0`과 `videoagg0`은 브릿지 `bridge0`으로 구성됩니다. 한 인터페이스에서 수신된 패킷이 다른 인터페이스로 전달됩니다. 브릿지 구성 후에 두 인터페이스를 사용하여 VLAN과 IP 인터페이스를 구성할 수도 있습니다.

그림 1–3 네트워크 스택의 브릿지



제 1 부

Network Auto-Magic

NWAM(Network Auto-Magic)은 시스템의 기본 네트워크 구성을 자동화하는 Oracle Solaris 기능입니다. 이 장에 포함된 항목에서는 NWAM 아키텍처의 구성 요소 및 이러한 구성 요소가 함께 작동하여 Oracle Solaris 시스템의 자동 네트워크 구성에 영향을 주는 방식에 대해 설명합니다.

이 설명서는 주로 NWAM 명령줄 유틸리티를 사용하여 네트워크 구성을 관리하는 방법에 중점을 둡니다. 또한 NWAM GUI(그래픽 사용자 인터페이스)를 사용하여 네트워크 상태를 보고 모니터하며 데스크탑에서 NWAM과 상호 작용하는 방법의 기본 정보에 대해 설명합니다. NWAM GUI를 사용하여 네트워크 구성을 모니터 및 관리하는 방법의 자세한 지침은 온라인 도움말에서 확인할 수 있습니다.

◆ ◆ ◆ 2 장

2 NWAM 소개

NWAM(Network Auto-Magic) 기능은 시작 시 유선 또는 무선 네트워크에 연결, 데스크탑에서 현재 활성 네트워크 연결의 상태에 대한 알림 표시와 같은 기본 이더넷 및 WiFi 구성을 자동으로 처리하여 네트워크 구성을 단순화합니다. 또한 NWAM은 시스템 차원 네트워크 프로파일의 만들기 및 관리와 같은 더 복잡한 네트워킹 작업을 단순화하도록 설계되었습니다. 예를 들어, 모두 Oracle Solaris의 기능인 이름 지정 서비스, IP 필터 및 IP 보안(IPsec)의 구성이 여기에 해당합니다.

이 장에서는 다음 항목을 다룹니다.

이 장의 내용은 기본 네트워킹 개념을 이해하고 일반 네트워킹 도구와 명령을 사용하여 네트워크 구성을 관리한 경험이 있는 사용자와 네트워크 관리자를 대상으로 작성되었습니다. NWAM을 사용하여 네트워크 구성을 관리할 준비가 되었으면 4 장, “NWAM 프로파일 구성(작업)”으로 건너뜁니다.

Oracle Solaris의 네트워크 인터페이스 관리에 대한 기본 정보는 제2부을 참조하십시오.

NWAM 구성

NWAM 구성은 가능한 한 자동화된 방식으로 시스템의 네트워크 구성에 영향을 주기 위해 함께 작동하는 여러 구성 요소로 이루어집니다. 이동성에 중점을 두는 NWAM은 여러 네트워크 이벤트에 대한 응답으로 또는 사용자 요청 시 시스템 구성을 동적으로 변경할 수 있습니다. NWAM에는 유선 네트워크 인터페이스가 분리되거나 새 무선 네트워크가 사용 가능한 경우와 같이 네트워크 상태의 변경을 다루는 동적 기능이 포함되어 있습니다.

NWAM을 통한 네트워크 구성은 **구성 객체**라고도 하는 여러 유형의 프로파일과 연결된 등록 정보 및 해당 값으로 이루어집니다.

이러한 프로파일과 구성 객체는 다음과 같습니다.

- **NCP(네트워크 구성 프로파일)**

  NCP는 네트워크 링크 및 인터페이스의 구성을 지정합니다. 이 프로파일은 NWAM 구성을 이루는 주요 프로파일 유형 중 하나입니다. 두번째 주요 프로파일 유형은 위치 프로파일입니다.

  시스템은 항상 자동 NCP라는 NCP를 정의합니다. 이 NCP는 사용자 입력이 없는 경우에 활성화됩니다. 자동 NCP는 시스템에서 생성되고 유지 관리되므로 수정하거나 제거할 수 없습니다.

  필요에 따라 사용자 정의 NCP를 추가로 만들 수도 있습니다. 자동 NCP 및 사용자 정의 NCP에 대한 전체 설명은 44 페이지 "자동 NCP 및 사용자 정의 NCP에 대한 설명"을 참조하십시오.

- **NCU(네트워크 구성 단위)**

  NCU는 NCP를 구성하는 모든 등록 정보가 포함된 개별 구성 객체입니다. NCP는 근본적으로 NCP를 정의하는 NCU를 저장하는 컨테이너입니다. 각 NCU는 시스템의 개별 링크 또는 인터페이스와 상호 관련됩니다. NCU에 대한 전체 설명은 43 페이지 "NCU에 대한 설명"을 참조하십시오.

- **위치**

  위치 프로파일은 NWAM 구성을 이루는 두 가지 주요 프로파일 유형 중 하나입니다. 위치는 시스템 차원의 네트워크 구성(예: 이름 지정 서비스, 도메인, IP 필터 및 IPsec 구성)을 지정합니다. 이 정보는 시스템 차원의 네트워크 구성에 적용되는 등록 정보 세트로 구성됩니다. 시스템 정의 위치와 사용자 정의 위치가 모두 있습니다. 위치 프로파일에 대한 전체 설명은 44 페이지 "위치 프로파일에 대한 설명"을 참조하십시오.

- **ENM(외부 네트워크 수정자)**

  ENM은 NWAM의 외부 응용 프로그램(예: VPN 응용 프로그램)을 관리하는 데 사용되는 프로파일입니다. 이러한 응용 프로그램은 네트워크 구성을 수정하고 만들 수 있습니다. `nwamd` 데몬은 ENM의 일부로 지정된 조건에 따라 ENM을 활성화하거나 비활성화합니다. ENM에 대한 전체 설명은 45 페이지 "ENM에 대한 설명"을 참조하십시오.

- **알려진 WLAN(무선 LAN)**

  알려진 WLAN은 NWAM이 시스템에 알려진 무선 네트워크에 대한 정보를 모니터하고 저장하는 데 사용하는 구성 객체입니다. NWAM은 이러한 모든 무선 네트워크의 목록을 유지 관리하고, 이 목록을 참조하여 사용 가능한 무선 네트워크에 대한 연결이 시도되는 순서를 확인합니다. 알려진 WLAN에 대한 전체 설명은 46 페이지 "알려진 WLAN 정보"를 참조하십시오.

NWAM 기능 구성 요소

NWAM은 다음과 같은 기능 구성 요소로 구성됩니다.

- **NWAM 프로파일 저장소** – 프로파일 저장소에 NWAM 구성 데이터가 저장됩니다. 프로파일 저장소에 대한 액세스는 저장소 데몬 `netcfgd`가 관리합니다.

  NWAM을 사용으로 설정한 경우 NWAM 프로파일 저장소에 네트워크 구성의 스냅샷이 포함됩니다. 이 데이터는 네트워크의 수동 구성으로 되돌려야 하는 경우를 위해 보존됩니다. 자세한 내용은 47 페이지 "NWAM 구성 데이터"를 참조하십시오.

- **프로파일 구성 프로그램(사용자 인터페이스)** – NWAM 아키텍처에는 CLI(명령줄 인터페이스)와 GUI(그래픽 사용자 인터페이스)가 모두 포함되어 있습니다. 이러한 인터페이스를 사용하여 프로파일 만들기 및 수정, 프로파일 활성화, 시스템에 프로파일 정보 질의와 같은 유사한 작업을 수행할 수 있습니다.

  NWAM CLI는 두 개의 관리 명령 `netcfg` 및 `netadm`으로 구성됩니다. `netcfg` 명령을 사용하면 프로파일을 만들고 수정할 수 있습니다. 이 명령은 대화식 모드, 명령줄 모드 및 명령 파일 모드로 작동합니다. `netadm` 명령을 사용하면 프로파일 사용 또는 사용 안함으로 설정, 프로파일 상태 정보 나열 등의 특정 작업을 수행할 수 있습니다. 자세한 내용은 `netcfg`(1M) 및 `netadm`(1M) 매뉴얼 페이지를 참조하십시오.

  NWAM CLI를 사용하여 프로파일을 만들고 관리하는 방법의 단계별 지침은 4 장, "NWAM 프로파일 구성(작업)" 및 5 장, "NWAM 프로파일 관리(작업)"를 참조하십시오.

  NWAM GUI를 사용하여 네트워크 프로파일을 만들고 관리할 수도 있습니다. GUI에는 데스크탑에서 네트워크 연결의 상태를 신속하게 보고 모니터할 수 있는 추가 기능이 있습니다. 또한 GUI에는 네트워크의 현재 상태 변경에 대해 알리는 알림 기능이 있습니다. 알림 기능은 GUI에서만 사용할 수 있습니다. NWAM GUI 사용에 대한 자세한 내용은 6 장, "NWAM 그래픽 사용자 인터페이스 정보" 또는 온라인 도움말을 참조하십시오. `nwammgr`(1M) 및 `nwammgr-properties`(1M) 매뉴얼 페이지를 참조하십시오.

- **정책 엔진** – `nwamd` 데몬은 NWAM의 정책 구성 요소입니다. 이 데몬은 여러 역할을 수행하며, 프로파일 저장소에 저장된 프로파일을 기준으로 네트워크 구성을 관리합니다. 데몬은 현재 네트워크 상태에 따라 활성화할 프로파일을 확인한 다음 해당 프로파일을 활성화합니다. 이 작업을 수행하기 위해 데몬은 여러 소스의 정보를 통합합니다. `nwamd` 데몬이 수행하는 여러 역할은 63 페이지 "NWAM 데몬 개요" 섹션에서 자세히 설명합니다.

- **저장소 데몬** – `netcfgd` 데몬은 프로파일 및 기타 구성 객체의 모든 구성 데이터를 저장하는 공통 프로파일 저장소를 제어합니다. `netcfg` 명령, NWAM GUI 및 `nwamd` 데몬은 프로파일 저장소에 액세스하기 위해 모두 요청을 보내 `netcfgd` 데몬과 상호 작용합니다. 저장소 데몬의 작업은 저장소 데이터에 액세스를 시도하는 다양한 프로세스에 올바른 권한을 부여했는지 확인하는 것입니다. 데몬은 인증되지 않은 프로세스의 액세스 시도를 모두 금지(실패)합니다. 자세한 내용은 63 페이지 "NWAM 저장소 데몬(netcfgd)에 대한 설명"을 참조하십시오.

- **NWAM 라이브러리 인터페이스** – `libnwam` 라이브러리는 프로파일 저장소와 상호 작용하기 위한 기능 인터페이스를 제공하여 NWAM이 프로파일 정보를 읽고 수정할 수 있게 합니다.
- **SMF(서비스 관리 기능) 네트워크 서비스** – NWAM이 사용하는 여러 네트워크 서비스는 Oracle Solaris에 이미 포함되어 있습니다. 하지만 이러한 기존 서비스 중 일부는 수정되었으며 NWAM과 관련된 새로운 서비스가 도입되었습니다. 자세한 내용은 64 페이지 "SMF 네트워크 서비스"을 참조하십시오.

NWAM 사용 시기

일반적으로 작업 환경과 연결 방법(유선 또는 무선)을 자주 변경하는 경우 NWAM의 자동 네트워크 구성 기능을 이용하는 것이 좋습니다. NWAM을 사용하여 사무실, 집 또는 이동 중과 같은 다양한 설정에서 네트워크에 연결할 수 있게 하는 사용자 정의 프로파일을 설정할 수 있습니다. NWAM은 네트워크 환경을 자주 변경해야 하는 랩탑 모델 및 시스템 사용자에게 유용한 도구입니다. 또한 NWAM GUI를 사용하면 일반 네트워크 도구와 명령보다 정적 IP 구성과 WiFi 네트워크 연결을 훨씬 쉽게 설정할 수 있습니다.

이더넷 연결 해제나 NIC(네트워크 인터페이스 카드) 추가 또는 제거와 같은 네트워크 환경의 변경 사항에 맞게 NWAM을 구성할 수 있습니다.

주 – 예를 들어, NWAM에서 현재 지원하지 않는 고급 네트워킹 기능을 사용하는 경우 수동으로 네트워크를 구성할 수 있습니다. 자세한 내용은 99 페이지 "네트워크 구성 관리"를 참조하십시오.

NWAM 구성 작동 방식

NWAM의 기본 동작은 사용자 상호 작용 없이 유선 또는 무선 네트워크의 기본 구성을 "자동으로" 수행하는 것입니다. 단, 무선 네트워크의 보안 키나 암호를 제공하는 경우와 같이 시스템에서 추가 정보를 묻는 메시지가 표시되는 경우에는 NWAM과 상호 작용해야 합니다.

자동 NWAM 구성은 다음 이벤트와 작업에 의해 트리거됩니다.

- 이더넷 케이블 연결 또는 연결 해제
- WLAN 카드 연결 또는 연결 해제
- 유선 인터페이스, 무선 인터페이스 또는 둘 다가 사용 가능한 경우 시스템 부트
- 유선 인터페이스, 무선 인터페이스 또는 둘 다가 사용 가능한 경우 일시 중지에서 다시 시작(지원되는 경우)
- DHCP 임대 획득 또는 손실

NWAM 구성 요소는 다음과 같은 방식으로 서로 상호 작용합니다.

- 항상 시스템에서 NCP 한 개와 위치 프로파일 한 개가 활성 상태여야 합니다.
- 시스템 부트 도중 정책 엔진 데몬인 nwamd는 다음 작업을 수행합니다.
  1. 현재 활성 NCP의 서비스 등록 정보 참조
  2. IP 주소가 하나 이상 구성될 때까지 계속
  3. 위치 프로파일의 조건 확인
  4. 정책 엔진에서 지정된 위치 프로파일 활성화
  5. 적절하게 네트워크 구성
- 네트워크 구성 변경을 트리거할 수 있는 이벤트가 발생하면 NWAM 데몬인 nwamd가 다양한 역할을 맡아 다음 작업을 수행합니다.
  1. 이벤트 처리기로서 nwamd는 각 이벤트가 발생할 때 이를 감지합니다.
  2. 프로파일 데몬으로서 nwamd는 활성 프로파일을 참조합니다.
  3. 변경 사항에 따라 nwamd가 네트워크를 적절하게 재구성할 수도 있습니다.

NWAM 기본 동작

사용자 정의 네트워크 프로파일이 없을 경우 nwamd는 다음 세 가지 시스템 정의 프로파일을 기준으로 네트워크 구성을 관리합니다.

- 자동 NCP
- 자동 위치
- NoNet 위치

자동 NCP는 다음 기본 정책을 구현합니다.

- DHCP를 통해 사용 가능한(연결된) 이더넷 인터페이스를 모두 구성합니다.
- 이더넷 인터페이스가 연결되어 있지 않거나 IP 주소를 얻을 수 없는 경우 무선 인터페이스를 활성화하고 **알려진** *WLAN* **목록**에서 사용 가능한 최상의 WLAN에 자동으로 연결합니다. 또는 사용자가 연결할 무선 네트워크를 선택할 때까지 기다립니다.
- 적어도 하나의 IPv4 주소를 얻을 때까지 NoNet 위치가 활성 상태로 유지됩니다. 이 위치 프로파일은 IP 주소 획득과 관련된 데이터만 전달하는 엄격한 IP 필터 규칙 세트를 제공합니다(DHCP 및 IPv6 autoconf 메시지). 활성 조건을 제외하고 NoNet 위치의 모든 등록 정보를 수정할 수 있습니다.
- 시스템 인터페이스 중 하나에 IPv4 주소가 하나 이상 할당된 경우 자동 위치가 활성화됩니다. 이 위치 프로파일에는 IP 필터 또는 IPsec 규칙이 없습니다. 위치 프로파일은 DHCP 서버로부터 얻은 DNS 구성 데이터를 적용합니다. NoNet 위치와 마찬가지로, 활성 조건을 제외하고 자동 위치의 모든 등록 정보를 수정할 수 있습니다.

- NoNet 위치는 시스템에 할당된 IPv4 주소가 없는 경우 항상 적용됩니다. 적어도 하나의 IPv4 주소가 할당된 경우 시스템이 현재 네트워크 조건과 가장 일치하는 활성화 규칙을 통해 위치 프로파일을 선택합니다. 더 나은 일치 항목이 없을 경우 시스템이 자동 위치로 대체합니다. 자세한 내용은 51 페이지 “NWAM 프로파일 활성화 방식”을 참조하십시오.

NWAM이 다른 Oracle Solaris 네트워킹 기술과 함께 작동하는 방식

NWAM은 다음과 같은 다른 Oracle Solaris 네트워킹 기술과 함께 작동합니다.

- **네트워크 가상화**

  NWAM은 다음과 같은 다양한 Oracle Solaris 네트워크 가상화 기술과 함께 작동합니다.

  - **가상 시스템: Oracle VM Server for SPARC(이전 Logical Domains) 및 Oracle VM VirtualBox**

    NWAM은 Oracle Solaris 호스트와 게스트에서 모두 지원됩니다. NWAM은 지정한 가상 시스템에 속하는 인터페이스만 관리하고 다른 가상 시스템을 방해하지 않습니다.

  - **Oracle Solaris 영역 및 스택 인스턴스**

    NWAM은 전역 영역이나 배타적 스택인 비전역 영역에서 작동합니다.

    주 – NWAM은 공유 스택 영역에서 작동하지 않습니다.

  - **VNIC**

    현재 NWAM 구현에서는 VNIC를 관리하지 않지만 수동으로 만든 VNIC가 재부트 후에도 유지되며, 가령 배타적 스택 영역에 할당하기 위해 새로 만들 수 있습니다.

  - **브릿징 기술**

    브릿징 기술은 개별 네트워크 세그먼트를 연결하여 단일 세그먼트만 사용 중인 것처럼 연결된 노드 간에 통신할 수 있게 하는 방법입니다. 현재 NWAM 구현에서는 브릿징 기술을 사용하는 네트워크 구성을 지원하지 않지만 시스템에서 이 기술을 사용하기 전에 NWAM 구성 관리를 사용 안함으로 설정할 필요는 없습니다.

  - **동적 재구성 및 네트워크 구성 프로파일**

    DR(동적 재구성) 및 핫 플러그 기능을 지원하는 시스템에서는 해당 시스템의 활성 NCP가 `DefaultFixed`인 경우에만 이러한 기능을 바로 사용할 수 있습니다.

    시스템에서 사용으로 설정된 NCP가 `Automatic`이거나 다른 사용자가 만든 NCP인 경우 DR 작업을 수행하기 전에 다음 단계 중 하나를 먼저 수행해야 합니다.

- 네트워크 서비스를 중지합니다. 이 작업은 시스템의 모든 네트워크 인터페이스 작동을 중지합니다. 따라서 서비스를 중지하려면 시스템 콘솔을 사용해야 합니다. 장치를 제거하거나 교체한 후 서비스를 다시 시작합니다.
- `netcfg` 명령을 사용하여 활성 NCP의 구성에서 IP 인터페이스를 제거합니다. 그런 다음 해당 IP 인터페이스의 기본 하드웨어 장치를 계속 물리적으로 제거 또는 교체할 수 있습니다. 해당하는 경우 DR이 완료된 후 IP 인터페이스를 재구성합니다.

- **일반 네트워킹 명령 및 유틸리티**

  항상 시스템은 NWAM 네트워크 구성 또는 일반 네트워크 구성 중 **하나**를 사용합니다. `DefaultFixed` NCP를 사용으로 설정한 경우 시스템은 일반 네트워크 구성을 사용합니다. 시스템은 이 NCP를 사용으로 설정할 때 `/etc/ipadm/ipadm.conf` 및 `/etc/dladm/datalink.conf` 파일에 저장되는 지속 구성을 적용합니다. `ipadm` 및 `dladm` 명령을 사용하여 네트워크 구성을 확인하고 변경할 수도 있습니다. NWAM NCP를 사용으로 설정한 경우 시스템이 `/etc/ipadm/ipadm.conf` 구성을 무시하며, NWAM은 활성 NCP에 지정된 정책에 따라 네트워크 구성을 관리합니다.

  NWAM이 네트워크 구성을 관리하는 경우에도 명령줄 네트워킹 유틸리티인 `dladm` 및 `ipadm`을 사용하여 현재 네트워크 구성의 구성 요소를 확인할 수 있습니다.

  주 – 명령줄 도구를 사용하여 네트워크 구성을 변경할 수는 없습니다. 이러한 변경 사항이 NWAM에서 적용된 정책과 충돌할 수 있기 때문입니다.

- **IPMP(IP Network Multipathing)**

  NWAM은 현재 IPMP 사용을 지원하지 않습니다. IPMP를 사용하도록 네트워크를 구성하기 전에 `DefaultFixed` NCP가 사용으로 설정되었는지 확인합니다.

네트워크 구성 작업을 찾을 위치

다음 표에서는 네트워크 구성 항목과 자세한 정보가 제공되는 위치를 보여줍니다.

네트워킹 작업	자세한 정보
NWAM에 대한 자세한 개요 정보를 찾습니다.	3 장, “NWAM 구성 및 관리(개요)”
NWAM CLI를 사용하여 프로파일 및 구성 객체를 만들고, 수정 및 제거합니다.	4 장, “NWAM 프로파일 구성(작업)”
NWAM CLI를 사용하여 프로파일 및 구성 객체에 대한 정보를 확인하고 관리합니다.	5 장, “NWAM 프로파일 관리(작업)”
데스크탑에서 NWAM GUI를 사용하여 네트워크 상태 정보를 확인하고, 네트워크 연결을 전환하고, 프로파일 및 구성 객체를 만들고 수정합니다.	6 장, “NWAM 그래픽 사용자 인터페이스 정보” 및 온라인 도움말

네트워킹 작업	자세한 정보
NWAM 네트워크 구성 모드와 일반 네트워크 구성 모드를 전환합니다.	99 페이지 “네트워크 구성 관리”
일반 네트워킹 도구와 명령을 사용하여 네트워크 구성을 관리합니다.	8 장, “데이터 링크 구성 및 관리” 및 9 장, “IP 인터페이스 구성”
가상 네트워크를 구성하고 관리합니다.	17 장, “네트워크 가상화 및 리소스 제어 소개(개요)”

◆ ◆ ◆ 3 장

3 NWAM 구성 및 관리(개요)

이 장에서는 NWAM 구성 및 관리 프로세스에 대한 배경 정보 및 개요 정보를 제공합니다. NWAM이 네트워크 구성을 단순화하고 자동화하는 데 사용하는 프로파일 구현에 대한 자세한 설명도 제공됩니다.

이 장에서는 다음 항목을 다룹니다.

NWAM 구성 개요

NWAM은 기본 등록 정보 값을 프로파일 형태로 시스템에 저장하여 네트워크 구성을 관리합니다. 그런 다음 NWAM은 현재 네트워크 상태에 따라 활성화할 프로파일을 확인한 다음 해당 프로파일을 활성화합니다. NWAM 프로파일 구현은 NWAM의 주요 구성 요소입니다.

네트워크 프로파일

네트워크 프로파일은 현재 네트워크 상태에 따라 네트워크 구성 방식과 작동 방식을 결정하는 등록 정보 모음입니다.

다음은 NWAM 구성을 이루는 프로파일 유형 및 구성 객체입니다.

- NCP(네트워크 구성 프로파일)
- 위치 프로파일
- ENM(외부 네트워크 수정자)
- 알려진 WLAN

두 가지 주요 네트워크 프로파일 유형은 NCP와 위치 프로파일입니다. NWAM을 통해 네트워크 자동 구성에 영향을 주려면 항상 NCP 한 개와 위치 프로파일 한 개가 시스템에서 활성 상태여야 합니다.

NCP는 물리적 링크 및 IP 인터페이스와 같은 개별 구성 요소의 구성을 포함하여 로컬 네트워크의 구성을 지정합니다. 각 NCP는 NCU(**네트워크 구성 단위**)라는 개별 구성 객체로 이루어져 있습니다. 각 NCU는 물리적 링크 또는 인터페이스를 나타내며 해당 링크나 인터페이스의 구성을 정의하는 등록 정보로 구성됩니다. 사용자 정의 NCP를 구성하는 프로세스에는 해당 NCP의 NCU를 만드는 작업이 포함됩니다. 자세한 내용은 43 페이지 “NCU에 대한 설명”을 참조하십시오.

위치 프로파일에는 다음과 같은 시스템 차원의 네트워크 구성 정보가 포함되어 있습니다.

- 위치 프로파일이 활성화된 조건
- 사용할 이름 지정 서비스
- 도메인 이름
- IP 필터 규칙 세트
- IPsec 정책

자세한 내용은 44 페이지 “위치 프로파일에 대한 설명”을 참조하십시오.

ENM은 네트워크 구성을 만들고 수정할 수 있는 외부 응용 프로그램에 사용되는 NWAM 프로파일입니다. ENM을 만들 때 지정한 조건에 따라 외부 응용 프로그램을 활성화 및 비활성화하도록 NWAM을 구성할 수 있습니다.

알려진 WLAN은 이전에 연결한 알려진 무선 네트워크 목록을 유지 관리하는 데 사용되는 NWAM 프로파일입니다. 자세한 내용은 45 페이지 “ENM에 대한 설명” 및 46 페이지 “알려진 WLAN 정보”를 참조하십시오.

NCP에 대한 설명

NCP는 시스템의 네트워크 구성을 정의합니다. 예를 들어, NCP를 구성하는 NCU는 다양한 네트워크 링크와 인터페이스를 구성하는 방법, 표시할 인터페이스, 인터페이스를 표시할 조건 및 인터페이스의 IP 주소를 가져오는 방법을 지정합니다. 자동 및 사용자 정의라는 두 가지 NCP 유형이 있습니다. 자동 NCP는 NWAM에서 자동으로 생성되는 시스템 정의 프로파일입니다. 이 프로파일은 만들고, 수정 또는 제거할 수 없습니다. 사용자 정의 NCP는 특정 네트워크 구성의 요구를 충족하기 위해 만드는 프로파일입니다. 사용자 정의 NCP는 사용자가 수정하고 제거할 수 있습니다.

자동 NCP는 현재 시스템에 있는 모든 링크와 인터페이스의 표현입니다. 네트워크 장치를 추가하거나 제거하면 자동 NCP의 컨텐츠가 변경됩니다. 하지만 자동 NCP와 연결된 구성 기본 설정은 편집할 수 없습니다. 자동 NCP는 시스템의 IP 주소를 가져올 수 있도록 하는 DHCP 및 주소 자동 구성을 활용하는 프로파일에 대한 액세스를 제공하기 위해 생성됩니다. 이 프로파일은 무선 링크보다 유선 링크를 선호하는 링크 선택 정책도 구현합니다. 대체 IP 구성 정책 또는 대체 링크 선택 정책을 지정해야 하는 경우 시스템에서 사용자 정의 NCP를 추가로 만듭니다.

NCU에 대한 설명

NCU는 NCP를 구성하는 개별 구성 객체입니다. NCU는 시스템에 있는 개별 물리적 링크와 인터페이스를 나타냅니다. 사용자 정의 NCP를 구성하는 프로세스에는 각 링크와 인터페이스를 구성하는 방법 및 조건을 지정하는 NCU를 만드는 작업이 포함됩니다.

다음 두 가지 NCU 유형이 있습니다.

- **링크 NCU**

  링크 NCU(예: 물리적 장치)는 OSI(Open Systems Interconnection) 모델의 계층 2 엔티티입니다.

- **인터페이스 NCU**

  인터페이스 NCU, 특히 IP 인터페이스는 OSI 모델의 계층 3 엔티티입니다.

링크 NCU는 데이터 링크를 나타냅니다. 다음과 같은 여러 클래스의 데이터 링크가 있습니다.

- 물리적 링크(이더넷 또는 WiFi)
- 터널
- 통합
- VLAN(가상 LAN)
- VNIC(가상 네트워크 인터페이스 카드)

주 – 현재 NWAM 구현에는 물리적 링크(이더넷 및 WiFi)의 기본 네트워크 구성에 대한 지원**만** 포함되어 있습니다. NWAM에서 지원되지는 않지만 NWAM 구성 관리를 사용 안함으로 설정할 필요없이 VNIC와 브릿징 등의 여러 고급 네트워킹 기술을 네트워크에 구성할 수 있습니다.

하지만 IPMP(IP Network Multipathing)를 사용하도록 시스템을 구성하면 NWAM 구성 관리를 사용할 수 없습니다. 따라서 일반 네트워크 구성을 사용해야 합니다. 지침은 99 페이지 “자동 네트워크 구성 모드에서 수동 네트워크 구성 모드로 전환하는 방법”을 참조하십시오.

자동 NCP 및 사용자 정의 NCP에 대한 설명

자동 NCP는 시스템에 있는 각 물리적 링크에 대해 링크 NCU 한 개와 인터페이스 NCU 한 개로 구성된 시스템 정의 프로파일입니다. 이 NCP의 NCU 활성화 정책은 무선 링크보다 연결된 유선 링크를 선호하고 각 사용 링크에서 IPv4 및 IPv6을 모두 연결하는 것입니다. DHCP는 IPv4 주소를 가져오는 데 사용됩니다. Stateless 자동 구성 및 DHCP는 IPv6 주소를 가져오는 데 사용됩니다. 자동 NCP는 새 링크를 시스템에 삽입하거나 제거할 때 동적으로 변경됩니다. 삽입 또는 제거된 링크에 해당하는 NCU도 동시에 모두 추가되거나 제거됩니다. 이 프로파일은 `nwamd` 데몬에 의해 자동으로 업데이트됩니다.

사용자 정의 NCP는 사용자가 만들고 관리합니다. 지정한 프로파일에 NCU를 명시적으로 추가하고 제거해야 합니다. 현재 시스템에 있는 링크에 대한 상관 관계가 없는 NCU를 만들 수 있습니다. 시스템에 있는 링크에 대한 상관 관계가 없는 NCU를 제거할 수도 있습니다. 또한 사용자 정의 NCP에 대한 정책을 결정할 수 있습니다. 예를 들어, 지정된 시간에 여러 링크와 인터페이스가 시스템에서 사용으로 설정될 수 있게 하고 NCU와 정적 IP 주소 간에 다른 종속성 관계를 지정할 수 있습니다.

사용자 정의 NCP를 만들고 이 NCP에 NCU를 추가하고 제거하는 방법에 대한 단계별 지침은 69 페이지 “NCP 만들기”를 참조하십시오.

위치 프로파일에 대한 설명

위치 프로파일은 기본 IP 연결이 설정된 후 추가 네트워킹 세부 정보를 제공합니다. 위치에는 시스템 차원의 네트워크 구성과 관련된 등록 정보 세트로 이루어진 네트워크 구성 정보가 포함됩니다.

위치 프로파일은 필요한 경우 함께 적용되는 특정 네트워크 구성 정보(예: 이름 지정 서비스 및 방화벽 설정)로 구성됩니다. 또한 위치가 반드시 물리적 위치와 일치하지는 않기 때문에 각 네트워킹 요구를 충족하는 위치 프로파일을 여러 개 설정할 수 있습니다. 예를 들어, 한 위치는 회사 인트라넷에 연결된 경우에 사용할 수 있습니다. 다른 위치는 사무실에 있는 무선 액세스 포인트를 통해 공용 인터넷에 연결된 경우에 사용할 수 있습니다.

기본적으로 시스템은 다음 두 개의 위치 프로파일을 미리 정의합니다.

- **NoNet**

  NoNet 위치에는 특정 활성화 조건이 있습니다. 할당된 IP 주소를 가진 로컬 인터페이스가 없는 경우 NWAM에서 이 프로파일을 독립형 시스템에 적용합니다. NoNet 위치가 시스템에서 처음 활성화된 후 이 위치를 수정할 수 있습니다. 이 위치의 기본 설정을 복원하려는 경우를 위해 원본 NoNet 위치의 읽기 전용 복사본이 시스템에 저장됩니다.

- **자동**

자동 위치는 사용 가능한 네트워크가 있지만 대체하는 다른 위치 프로파일이 없는 경우에 활성화됩니다. 자동 위치가 시스템에서 처음 활성화된 후 이 위치를 수정할 수 있습니다. 이 위치의 기본 설정을 복원하려는 경우를 위해 원본 자동 위치의 읽기 전용 복사본이 시스템에 저장됩니다.

주 – 자동 위치를 자동 NCP와 혼동해서는 안됩니다. 자동 위치는 시스템의 초기 네트워크 구성 후에 시스템 차원의 네트워크 등록 정보를 정의하는 위치 프로파일 유형입니다. 자동 NCP는 시스템의 링크 및 인터페이스 네트워크 구성을 지정합니다.

사용자 정의 위치는 시스템 차원의 네트워크 구성에 대해 지정하는 값으로 만드는 프로파일입니다. 사용자 정의 위치는 사용자가 설정한 값으로 구성되지만 시스템 정의 위치에는 미리 설정된 값이 있다는 점만 제외하고 사용자 정의 위치와 시스템 정의 위치는 동일합니다.

사용자 정의 위치 만들기에 대한 자세한 내용은 77 페이지 “위치 프로파일 만들기”를 참조하십시오.

ENM에 대한 설명

ENM은 NWAM의 외부 응용 프로그램과 관련된 프로파일입니다. 이러한 응용 프로그램은 네트워크 구성을 만들고 수정할 수 있습니다. ENM은 NCP 또는 위치 프로파일이 아닌 사용자 정의 네트워크 구성을 만들고 제거하는 수단으로 NWAM 설계에 포함됩니다. ENM은 사용 또는 사용 안함으로 설정 시 네트워크 구성을 직접 수정하는 서비스나 응용 프로그램으로 정의될 수도 있습니다. 지정한 조건에서 ENM을 활성화 및 비활성화하도록 NWAM을 구성할 수 있습니다. 지정된 한 시점에 각 프로파일 유형 중 하나만 시스템에서 활성 상태일 수 있는 NCP 또는 위치 프로파일과 달리, ENM은 동시에 여러 개가 시스템에서 활성 상태일 수 있습니다. 지정된 한 시점에 활성 상태인 ENM이 동시에 시스템에서 사용으로 설정되는 NCP 또는 위치 프로파일에 반드시 종속되는 것은 아닙니다.

ENM을 만들 수 있는 여러 외부 응용 프로그램과 서비스가 있지만 그 중 가장 많이 사용되는 것은 VPN 응용 프로그램입니다. 시스템에 VPN을 설치 및 구성한 후 지정한 조건에서 응용 프로그램을 자동으로 활성화 및 비활성화하는 ENM을 만들 수 있습니다.

주 – 시스템의 네트워크 구성을 직접 수정할 수 있는 외부 응용 프로그램에 대해 자동으로 학습하는 기능이 NWAM에는 없습니다. VPN 응용 프로그램이나 외부 응용 프로그램 또는 서비스의 활성화 또는 비활성화를 관리하려면 CLI 또는 NWAM GUI를 사용하여 응용 프로그램을 먼저 설치한 다음 ENM을 만들어야 합니다.

ENM에서 수행되는 네트워크 구성에 대한 지속 정보는 NCP 또는 위치 프로파일 정보의 저장 방식과 동일하게 NWAM에서 저장되거나 추적되지 않습니다. 하지만 NWAM은 외부적으로 시작된 네트워크 구성을 확인한 다음 ENM의 시스템 구성 변경 사항을

기준으로 활성화할 위치 프로파일을 재평가한 다음 해당 위치를 활성화할 수 있습니다. 예를 들어, 특정 IP 주소가 사용 중일 때 조건부로 활성화되는 위치로 전환합니다. 언제든지 `svc:/network/physical:default` 서비스를 다시 시작하면 활성 NCP에서 지정된 네트워크 구성이 복구됩니다. ENM도 다시 시작되어 프로세스의 네트워크 구성을 해제하고 다시 만들 수 있습니다.

ENM 등록 정보 만들기 및 수정에 대한 자세한 내용은 82 페이지 "ENM 프로파일 만들기"를 참조하십시오.

알려진 WLAN 정보

알려진 WLAN은 NWAM이 시스템에 알려진 무선 네트워크를 관리하는 데 사용하는 구성 객체입니다. NWAM은 이러한 알려진 무선 네트워크의 전역 목록을 유지 관리합니다. 이 정보는 NWAM이 사용 가능한 무선 네트워크에 연결을 시도하는 순서를 결정하는 데 사용됩니다. **알려진** *WLAN* **목록**에 있는 무선 네트워크를 사용할 수 있는 경우 NWAM이 자동으로 해당 네트워크에 연결합니다. 알려진 무선 네트워크를 두 개 이상 사용할 수 있는 경우 NWAM은 우선 순위가 가장 높은(가장 낮은 번호) 무선 네트워크에 연결을 시도합니다. NWAM이 연결하는 새 무선 네트워크가 알려진 WLAN 목록의 맨 위에 자동으로 추가되고 우선 순위가 가장 높은 현재 무선 네트워크가 됩니다.

알려진 WLAN은 우선 순위순으로 선택되며, 부호 없는 정수로 우선 순위가 할당됩니다. 알려진 WLAN 목록에서는 숫자가 작을수록 더 높은 우선 순위를 나타냅니다. 무선 네트워크에 처음 연결하면 NWAM이 자동으로 해당 WLAN을 목록에 추가합니다. 새 WLAN을 추가하면 이 목록에서 가장 높은 우선 순위를 갖게 됩니다. NWAM 기본 동작은 이전에 연결한 WLAN보다 최근에 연결한 WLAN이 선호됩니다. 알려진 WLAN은 동일한 우선 순위를 공유할 수 없습니다. 기존 WLAN과 동일한 우선 순위 값을 가진 새 WLAN을 목록에 추가하면 기존 항목이 더 낮은 우선 순위 값으로 이동됩니다. 그 뒤에 목록에서 다른 모든 WLAN의 우선 순위 값이 동적으로 더 낮은 우선 순위 값으로 이동됩니다.

하나 이상의 키 이름을 알려진 WLAN에 연결할 수도 있습니다. **키 이름**을 사용하면 `dladm create-secobj` 명령을 사용하여 고유한 키를 만들 수 있습니다. 알려진 WLAN `keyname` 등록 정보에 보안 객체 이름을 추가하여 이러한 키를 WLAN과 연결할 수 있습니다. 자세한 내용은 `dladm`(1M) 매뉴얼 페이지를 참조하십시오.

NWAM 명령줄 유틸리티를 사용한 WLAN 관리에 대한 자세한 내용은 107 페이지 "무선 검색 수행 및 사용 가능한 무선 네트워크에 연결"을 참조하십시오.

NWAM 구성 데이터

실제로 시스템에는 두 개의 구성 저장소가 있습니다. NWAM 프로파일 저장소는 `/etc/nwam` 디렉토리에 저장되고 일반 구성 저장소는 `/etc/ipadm/ipadm.conf` 및 `/etc/dladm/datalink.conf` 파일과 네트워크 서비스에 연결된 다른 구성 파일을 포함합니다.

NWAM이 네트워크 구성을 관리하는 경우 주로 고유한 저장소에서 작업합니다. `/etc/ipadm/ipadm.conf` 파일에 저장된 인터페이스 구성은 무시됩니다. NWAM은 NCP 데이터를 기준으로 물리적 링크와 인터페이스를 바로 구성합니다.

위치 프로파일 데이터는 NWAM 프로파일 저장소에서 읽어옵니다. 위치를 활성화하면 대부분의 경우 구성 변경 사항을 적용하기 위해 적절한 SMF 서비스 등록 정보를 설정하고 해당 서비스를 다시 시작하여 이 구성이 실행 중인 시스템에 적용됩니다. 이 작업은 해당 서비스 등록 정보의 기존 값을 덮어씁니다.

NWAM은 시작 시 위치 프로파일을 적용하는 동안 레거시 구성 데이터를 덮어쓰기 때문에 덮어쓸 수 있는 구성이 저장됩니다. 그런 다음 종료 시 NWAM이 해당 구성을 복원합니다. NWAM 작업의 일부로 적용될 수 있는 위치는 아니지만 이 데이터를 **레거시 위치** 데이터라고 합니다.

다음 시스템 정의 및 사용자 정의 네트워크 프로파일의 등록 정보 값은 NWAM 저장소에 저장됩니다.

- NCP - 자동 NCP 및 사용자 정의 NCP의 값이 포함됩니다.
- NCU - 링크 및 인터페이스 NCU의 값이 모두 포함됩니다.
- 위치 - 세 가지 시스템 정의 위치 유형의 값과 모든 사용자 정의 위치의 값이 포함됩니다.
- ENM - 응용 프로그램에 대한 정보가 포함됩니다.
- 알려진 WLAN - 자동으로 연결될 수 있는 무선 네트워크에 대한 정보가 포함됩니다.

각 NCP의 구성 데이터는 `ncp-` *name* 형식을 사용하여 `/etc/nwam` 디렉토리의 파일로 지속 저장됩니다. NCP당 파일 한 개가 있으며 항목은 각 NCU를 나타냅니다. 예를 들어, 자동 NCP의 파일 이름은 `ncp-Automatic.conf`로 지정됩니다. 모든 NCP 파일은 `/etc/nwam` 디렉토리에 저장됩니다.

위치 등록 정보는 `/etc/nwam/loc.conf` 파일에 저장됩니다.

ENM 등록 정보는 `/etc/nwam/enm.conf` 파일에 저장됩니다. 알려진 WLAN은 `/etc/nwam/known-wlan.conf` 파일에 저장됩니다. 이 파일은 `/etc/dladm/datalink.conf` 파일의 파일 형식과 유사합니다.

주 – NWAM 프로파일 저장소의 파일을 직접 편집하여 네트워크 프로파일을 수정할 수도 있지만 프로파일을 수정하는 적절한 방법은 `netcfg` 명령 또는 NWAM GUI 구성 패널을 사용하는 것입니다. 파일 형식과 파일 사용은 이후 릴리스에서 변경될 수도 있습니다. 88 페이지 “프로파일의 등록 정보 값 설정 및 변경”을 참조하십시오.

NCU 등록 정보 값

NCP의 개별 구성 객체인 NCU는 시스템의 개별 링크와 인터페이스를 나타냅니다. NCU 유형(링크 및 인터페이스)의 일반 등록 정보 및 각 NCU 유형과 관련된 등록 정보는 NWAM 프로파일 저장소에 저장됩니다. `type`, `class` 및 `parent` 등록 정보는 NCU를 만들 때 설정되며 나중에 변경할 수 없습니다. 또한 `enabled` 등록 정보는 직접 변경할 수 없습니다. 이 등록 정보는 `netadm` 명령을 통해 NCU를 사용 또는 사용 안함으로 설정하여 간접적으로 변경합니다.

자동 NCP는 시스템에서 검색된 각 물리적 링크에 대한 링크 NCU 한 개와 각 링크에 연결된 인터페이스 NCU 한 개로 구성됩니다. 추가 물리적 링크를 삽입하면 자동 NCP가 동적으로 변경됩니다. 새 링크를 삽입하면 새로운 각 링크에 대해 링크 NCU와 해당 인터페이스 NCU가 생성됩니다. 다음 표에서는 자동 NCP를 구성하는 각 NCU에 할당되는 값을 정의합니다.

주 – 이 표의 등록 정보는 자동 NCP의 NCU 등록 정보를 볼 때 표시되는 순서대로 나열되어 있습니다. 각 NCU 유형에 특정 값이 적용됩니다.

표 3–1 자동 NCP의 링크 NCU 등록 정보

등록 정보	링크 NCU 값
`type`	`link`
`class`	`phys`
`parent`	`Automatic`
`enabled`	`true`
`activation-mode`	`prioritized`
`priority-group`	`0`(802.3 링크) 또는 `1`(802.11 링크)
`priority-group-mode`	`shared`(802.3 링크) 또는 `exclusive`(802.11 링크)
`mac-address`	하드웨어 할당
`autopush`	해당 없음
`MTU`	해당 없음

표 3–2 자동 NCP의 인터페이스 NCU 등록 정보

등록 정보	인터페이스 NCU 값
type	interface
class	IP
parent	Automatic
enabled	true
ip-version	ipv4, ipv6
ipv4-addrsrc	dhcp
ipv4-static-addr	해당 없음
ipv6-addrsrc	dhcp, autoconf
ipv6-static-addr	해당 없음

시스템 정의 위치의 등록 정보 값

다음 표에서는 시스템 정의 프로파일인 자동 위치의 기본 등록 정보 값을 제공합니다. activation-mode 및 enabled 등록 정보를 제외하고 이러한 값을 수정할 수 있습니다. 인터페이스가 한 개 이상 활성 상태이고 대체하는 다른 위치 프로파일이 없는 경우 시스템은 항상 자동 위치를 활성화합니다.

표 3–3 시스템 정의 위치의 등록 정보

등록 정보	값
name	Automatic
activation-mode	system
enabled	필요에 따라 수정되는 system
conditions	해당 없음
default-domain	해당 없음
nameservices	dns
nameservices-config-file	/etc/nsswitch.dns
dns-nameservice-configsrc	dhcp
dns-nameservice-domain	해당 없음
dns-nameservice-servers	해당 없음

표 3–3 시스템 정의 위치의 등록 정보 *(계속)*

등록 정보	값
dns-nameservice-search	해당 없음
nis-nameservice-configsrc	해당 없음
nis-nameservice-servers	해당 없음
ldap-nameservice-configsrc	해당 없음
ldap-nameservice-servers	해당 없음
nfsv4-domain	해당 없음
ipfilter-config-file	해당 없음
ipfilter-v6–config-file	해당 없음
ipnat-config-file	해당 없음
ippool-config-file	해당 없음
ike-config-file	해당 없음
ipsecpolicy-config-file	해당 없음

다음 표에서는 NoNet 위치에 대해 미리 정의된 등록 정보를 제공합니다. `activation-mode` 및 `enabled` 등록 정보를 제외하고 이러한 값을 수정할 수 있습니다. 활성 인터페이스가 없는 경우 시스템은 항상 `NoNet` 위치를 사용으로 설정합니다.

표 3–4 NoNet 위치의 등록 정보

등록 정보	값
name	NoNet
activation-mode	system
enabled	필요에 따라 수정되는 system
conditions	해당 없음
default-domain	해당 없음
nameservices	files
nameservices-config-file	/etc/nsswitch.files
dns-nameservice-configsrc	해당 없음
dns-nameservice-domain	해당 없음
dns-nameservice-servers	해당 없음

표 3–4 NoNet 위치의 등록 정보 *(계속)*

등록 정보	값
dns-nameservice-search	해당 없음
nis-nameservice-configsrc	해당 없음
nis-nameservice-servers	해당 없음
ldap-nameservice-configsrc	해당 없음
ldap-nameservice-servers	해당 없음
nfsv4-domain	해당 없음
ipfilter-config-file	/etc/nwam/loc/NoNet/ipf.conf. 이 파일은 NWAM이 DHCP 주소 할당과 같은 네트워크 구성을 수행하는 데 필요한 최소 네트워크 트래픽 양을 제외하고 비루프백 트래픽을 모두 차단하는 IP 필터 규칙으로 구성됩니다.
ipfilter-v6–config-file	/etc/nwam/loc/NoNet/ipf6.conf. 이 파일은 ipfilter-config-file에 대해 설명된 대로 IP 필터 규칙으로 구성됩니다.
ipnat-config-file	해당 없음
ippool-config-file	해당 없음
ike-config-file	해당 없음
ipsecpolicy-config-file	해당 없음

사용자 정의 위치를 구성하는 등록 정보를 비롯한 위치 등록 정보에 대한 자세한 내용은 netcfg(1M) 매뉴얼 페이지를 참조하십시오.

NWAM 프로파일 활성화 방식

NCP, 위치 프로파일 및 ENM에는 activation-mode 등록 정보가 있습니다. 각 프로파일 유형에 허용되는 값은 서로 다릅니다. 각 프로파일이 활성화되는 조건과 마찬가지로 activation-mode 등록 정보의 검증 방식도 각 프로파일 유형에 따라 달라집니다.

시스템 정의 위치(자동 및 NoNet)의 경우 activation-mode 등록 정보 값이 system으로 설정되며, 이 경우 지정된 위치에 적합한 것으로 미리 결정한 조건에서만 시스템이 위치를 활성화할 수 있습니다.

사용자 정의 위치의 경우 activation-mode 및 conditions 등록 정보를 manual, conditional-any 또는 conditional-all로 설정할 수 있습니다. 자세한 내용은 54 페이지 “위치 활성화 선택 기준”을 참조하십시오.

netadm 명령을 사용하거나 NWAM GUI를 사용하여 위치 프로파일을 사용으로 수동 설정할 수 있습니다. 명시적으로 위치를 사용으로 설정하지 않으면 NWAM 데몬인 nwamd가 조건부 활성화된 위치 프로파일 및 시스템 활성화된 위치 프로파일에 대한 활성화 규칙을 모두 확인한 다음 현재 네트워크 환경에 가장 일치하는 위치를 선택합니다.

NWAM은 알고리즘을 사용하여 위치 선택 사항에 대한 "최상의 일치 항목"을 확인합니다. 위치에 적합한 일치 항목이 없는 경우 자동 위치가 활성화됩니다. 네트워크 환경이 변경되면 nwamd 데몬이 위치 선택 사항을 계속 재평가하여 최상의 일치 항목을 확인합니다. 하지만 netadm 명령을 사용하여 명시적으로 위치 프로파일을 사용으로 설정하는 경우(수동 활성화된 위치 또는 조건부 활성화된 위치), 명시적으로 해당 위치를 사용 안함으로 설정하거나 다른 위치를 사용으로 설정할 때까지 이 위치가 활성 상태로 유지됩니다. 이 경우 더 나은 일치 항목을 사용할 수 있는지 여부에 관계없이 네트워크 환경을 변경해도 위치 프로파일이 변경되지 않습니다. 명시적으로 현재 위치를 지정했다는 점 때문에 해당 위치가 가능한 최상의 일치 항목이 됩니다. 프로파일 활성화 및 비활성화 지침은 104 페이지 “프로파일 활성화 및 비활성화”를 참조하십시오.

NCP 활성화 정책

NWAM을 사용하면 NCU 활성화 시기와 관련하여 NCP 정책을 지정할 수 있습니다. NCP 정책은 각 NCU에 대해 지정할 수 있는 등록 정보 및 조건을 사용하여 적용됩니다. 지정할 수 있는 정책의 예로 "무선 연결 대신 유선 연결 선호" 또는 "한 번에 하나씩 인터페이스 활성화"가 있습니다. NCP 활성화 방식 및 시기는 각 NCU 유형에 대해 설정된 등록 정보에서 정의됩니다.

주 – 인터페이스 NCU는 항상 기본 링크 NCU에 연결되어야 합니다. 연결된 링크 NCU를 활성화하면 각 인터페이스 NCU도 활성화됩니다. netadm 명령을 사용하여 NCU의 기본 동작을 대체할 수 있습니다. 하지만 기본 링크 NCU에 대한 종속성은 제거할 수 없습니다. 예를 들어, 연결된 링크 NCU를 사용으로 설정하지 않고 인터페이스 NCU를 사용으로 설정하면 해당 인터페이스의 기본 NCU를 활성화할 때까지 인터페이스가 실제로 온라인 상태로 전환되지 않습니다.

NCP 정책의 예

다음 예에서는 NCP 정책이 사용 가능한 유선 링크를 모두 활성화하고 유선 연결을 사용할 수 없는 경우에만 무선 연결을 사용하도록 지정해야 하는 경우에 대해 NCU 등록 정보가 설정됩니다.

모든 물리적 링크의 경우:

- NCU 유형: `link`
- NCU 클래스: `phys`
- `activation-mode: prioritized`
- `priority-group: 0`(유선), `1`(무선)

- priority-mode: shared(유선), exclusive(무선)

다음 예에서는 항상 활성 링크 한 개만 시스템에 있고 유선 연결이 무선 연결보다 선호되도록 지정하는 NCP 정책에 따라 NCU 등록 정보가 설정됩니다.

모든 물리적 링크의 경우:

- NCU 유형: link
- NCU 클래스: phys
- activation-mode: prioritized
- priority-group: 0(유선), 1(무선)
- priority-mode: exclusive

NCU 활성화 등록 정보

네트워크 연결 활성화 방식은 링크 NCU 등록 정보에서 설정됩니다. 다음 등록 정보는 NCP 활성화 정책을 정의하는 데 사용됩니다.

- activation-mode 등록 정보

  이 등록 정보는 manual 또는 prioritized로 설정할 수 있습니다.

  - manual – 관리자가 NCU 활성화를 관리합니다. NWAM CLI 또는 GUI를 사용하여 NCU를 활성화하거나 비활성화할 수 있습니다. NCU의 activation-mode를 manual로 설정하면 priority-group 및 priority-mode NCU 등록 정보에 대해 설정된 값이 모두 무시됩니다.
  - prioritized – 지정한 NCU의 priority-group 및 priority-mode 등록 정보에 설정된 값에 따라 NCU가 활성화됩니다. prioritized NCU의 경우 enabled 등록 정보가 항상 true입니다.

    prioritized 활성화를 사용하면 동시에 링크 그룹을 활성화할 수 있습니다. 이 활성화 모드에서는 하나 이상의 링크가 다른 링크보다 선호될 수도 있습니다. priority-group 등록 정보는 지정된 링크에 숫자 우선 순위 레벨을 할당합니다. 동일한 우선 순위 레벨의 모든 링크가 그룹으로 검사됩니다. priority-mode 등록 정보는 활성화할 그룹에서 사용할 수 있거나 사용 가능해야 하는 그룹 구성원 수를 정의합니다.

- enabled 등록 정보(activation-mode가 manual로 설정됨)

  이 등록 정보의 값은 true 또는 false일 수 있습니다. 이 등록 정보의 값은 설정할 수 없습니다. 이 값은 사용으로 수동 설정된 NCU의 현재 상태를 반영하며, netadm 명령을 사용하거나 NWAM GUI를 사용하여 이 상태를 변경할 수 있습니다.

- priority-group 등록 정보(activation-mode가 prioritized로 설정됨)

  기본값은 숫자입니다. 0이 가장 높은 우선 순위를 나타냅니다. 음수 값은 잘못된 값입니다.

  사용 가능한 모든 priority-groups 중에서 사용 가능성이 가장 높은 priority-group의 NCU만 활성화됩니다. 우선 순위가 같은 NCU를 두 개 이상 사용할 수 있는 경우 priority-mode 등록 정보에 의해 활성화 동작이 정의됩니다. 우선 순위 번호는 절대값이 아닙니다. NCP 저장소를 업데이트하면 변경될 수 있습니다.

주 – 우선 순위 순서는 엄격하게 적용됩니다.

- `priority-mode` 등록 정보(`activation-mode`가 `prioritized`로 설정됨)

  이 등록 정보는 `priority-group` 등록 정보의 값을 지정한 경우에 설정됩니다.

  이 등록 정보의 값은 다음과 같습니다.

  - `exclusive` – 항상 `priority-group`의 NCU가 한 개만 활성 상태일 수 있도록 지정합니다. NWAM은 우선 순위 그룹 내에서 사용 가능한 첫번째 NCU를 활성화하고 다른 NCU는 무시합니다.
  - `shared` – 우선 순위 그룹의 여러 NCU가 동시에 활성 상태일 수 있도록 지정합니다. 우선 순위 그룹에서 사용 가능한 모든 NCU가 활성화됩니다.
  - `all` – 우선 순위 그룹이 사용 가능으로 간주되고 활성 상태가 되려면 우선 순위 그룹의 모든 NCU가 사용 가능해야 하도록 지정합니다.

위치 활성화 선택 기준

각 위치 프로파일에는 활성화 기준을 정의하는 등록 정보가 포함되어 있습니다. 이러한 등록 정보는 위치가 활성화되는 조건에 대한 정보를 지정합니다. NWAM은 구성된 모든 위치에 대해 선택 기준을 계속 재평가하고, 매번 현재 네트워크 환경에 대한 최상의 일치 항목인 기준이 있는 위치를 확인합니다. 현재 네트워크 환경이 변경되어 기준과 더 일치하는 항목이 발생하는 경우 NWAM은 현재 위치 프로파일을 비활성화하고 새 환경에 더 일치하는 위치 프로파일을 활성화합니다.

위치 활성화 시기와 방식에 대한 선택 기준은 다음 등록 정보에서 지정됩니다.

- `activation-mode`
- `conditions`

`activation-mode` 등록 정보는 다음 가능한 값 중 하나로 설정됩니다.

- `manual`
- `conditional-any`
- `conditional-all`
- `system`

주 – `activation-mode` 등록 정보의 `system` 값은 시스템 제공 위치(자동 및 NoNet 위치)에만 할당할 수 있습니다. `system` 값은 시스템이 이러한 위치의 활성화 시기를 결정함을 나타냅니다.

`activation-mode` 등록 정보를 `conditional-any` 또는 `conditional-all`로 설정하면 `conditions` 등록 정보에 사용자 정의된 조건부 표현식이 포함됩니다. 각 표현식에는 부울 값이 할당될 수 있는 조건이 포함됩니다(예: "ncu ip:net0 is-not active").

activation-mode 등록 정보를 conditional-any로 설정하면 다음 조건 중 하나가 true인 경우 조건이 충족됩니다.

activation-mode 등록 정보를 conditional-all로 설정하면 조건이 **모두** true인 경우 조건이 충족됩니다. 조건 문자열을 생성하는 데 사용할 수 있는 기준과 연산은 다음 표에 정의되어 있습니다.

표 3–5 조건 문자열 생성을 위한 기준 및 연산

객체 유형/속성	조건	객체
ncu, enm, loc	is/is-not active	이름
essid	is/is-not contains/does-not-contain	이름 문자열
bssid	is/is-not	bssid 문자열
ip-address	is/is-not	IPv4 또는 IPv6 주소
ip-address	is-in-range/is-not-in-range	IPv4 또는 IPv6 주소 및 넷마스크/prefixlen
advertised-domain	is/is-not contains/does-not-contain	이름 문자열
system-domain	is/is-not contains/does-not-contain	이름 문자열

주 – essid 등록 정보는 WLAN(무선 LAN)의 네트워크 이름인 ESSID(Extended Server Set Identifier)를 나타냅니다. bssid 등록 정보는 특정 WAP(무선 액세스 포인트) 또는 AP(액세스 포인트)의 MAC 주소인 BSSID(Basic Service Set Identifier)를 나타냅니다.

advertised-domain 및 system-domain 속성의 차이점을 확인합니다. 알려진 도메인은 DHCP 서버가 알리는 외부 통신(예: DNSdmain 또는 NISdmain 도메인 이름)을 통해 검색됩니다. 이 속성은 조건부 위치 활성화에 유용합니다. 예를 들어, 알려진 도메인이 mycompany.com이면 work 위치를 활성화합니다. system-domain 속성은 현재 시스템에 할당된 도메인입니다. 이 값은 domainname 명령에서 반환됩니다. 이 속성은 위치가 활성화되고 시스템이 해당 특정 도메인에 대해 구성된 후에만 true가 되기 때문에 ENM의 조건부 활성화에 유용합니다. 자세한 내용은 domainname(1M) 매뉴얼 페이지를 참조하십시오.

위치 등록 정보에 대한 자세한 내용은 44 페이지 “위치 프로파일에 대한 설명”을 참조하십시오.

netcfg 명령을 사용하여 프로파일 구성

netcfg(1M) 매뉴얼 페이지에 설명된 netcfg 명령은 네트워크 프로파일의 등록 정보 및 값을 구성하는 데 사용됩니다.

netcfg 명령을 사용하면 다음 작업을 수행할 수 있습니다.

- 사용자 정의 프로파일을 만들거나 삭제합니다.

  주 – 시스템 정의 프로파일은 만들거나 삭제할 수 없습니다.

- 시스템에 있는 모든 프로파일과 해당 등록 정보 값을 나열합니다.
- 지정한 프로파일의 모든 등록 정보 값과 리소스를 나열합니다.
- 프로파일과 연결된 각 등록 정보를 표시합니다.
- 지정한 프로파일의 등록 정보 중 하나 또는 모두를 설정하거나 수정합니다.
- 사용자 정의 프로파일의 현재 구성을 표준 출력이나 파일로 내보냅니다.

  주 – 시스템 정의 프로파일은 내보낼 수 없습니다.

- 프로파일에 대한 변경 사항을 삭제하고 해당 프로파일의 이전 구성으로 되돌립니다.
- 프로파일에 유효한 구성이 있는지 확인합니다.

대화식 모드, 명령줄 모드 또는 명령 파일 모드에서 netcfg 사용자 인터페이스를 사용할 수 있습니다. netcfg 명령은 계층적이기 때문에 대화식 모드에서 사용할 때 이해하기가 더 쉽습니다.

netcfg 명령에는 **범위**의 개념이 사용됩니다. 대화식으로 명령을 사용하는 경우 지정된 한 시점의 해당 범위는 프로파일 유형과 수행 중인 작업에 따라 달라집니다. 터미널 창에서 netcfg 명령을 입력하면 **전역 범위**에서 프롬프트가 표시됩니다.

여기서 select 또는 create 하위 명령을 사용하여 다음 최상위 프로파일을 확인, 수정 또는 생성할 수 있습니다.

- NCP
- 위치
- ENM
- 알려진 WLAN

프로파일을 만들거나 선택하기 전에는 netcfg 대화식 프롬프트가 다음 형태로 표시됩니다.

```
netcfg>
```

프로파일을 만들거나 선택한 후에는 netcfg 대화식 프롬프트가 다음과 같이 표시됩니다.

netcfg:*profile-type*:*profile-name*>

주 – 명령줄 모드에서는 전체 명령을 한 줄에 입력해야 합니다. 명령줄 모드에서 netcfg 명령을 사용하여 선택한 프로파일을 변경하는 경우 명령 입력을 완료하는 즉시 변경 사항이 지속 저장소에 커밋됩니다.

---

netcfg 명령 사용의 단계별 지침은 4 장, "NWAM 프로파일 구성(작업)"을 참조하십시오. netcfg 명령 사용에 대한 자세한 내용은 netcfg(1M) 매뉴얼 페이지를 참조하십시오.

netcfg 대화식 모드

netcfg 대화식 모드에서 작업하는 동안 최상위 프로파일을 선택하거나 만들면 위치 프로파일과 ENM의 **프로파일 범위**로 표시되는 명령 프롬프트가 생성됩니다. 예를 들면 다음과 같습니다.

```
netcfg> select loc foo
netcfg:loc:foo>
```

NCP를 선택하면 명령 프롬프트가 *NCP* **범위**로 표시됩니다. NCP 범위에서 NCU를 선택하거나 만들 수 있습니다. NCU를 선택하거나 만들면 선택한 NCU에 대한 프로파일 범위 프롬프트가 생성됩니다. 이 범위에서 다음 예와 같이 현재 선택한 프로파일과 연결된 모든 등록 정보를 확인하고 설정할 수 있습니다. 이 경우 User NCP가 먼저 선택된 후 NCP 범위에서 NCU가 생성되었습니다. 이 작업은 새로 만든 NCU의 프로파일 범위를 생성했습니다. 이 범위에서 NCU의 등록 정보를 확인하거나 설정할 수 있습니다.

```
netcfg> select ncp User
netcfg:ncp:User> create ncu phys net2
Created ncu 'net2'.  Walking properties ...
activation-mode (manual) [manual|prioritized]>
```

지정된 임의 범위에서 명령 프롬프트는 현재 선택한 프로파일을 나타냅니다. 이 범위에서 프로파일을 변경하는 경우 변경 사항을 **커밋**할 수 있으므로 변경 사항이 지속 저장소에 저장됩니다. 범위를 종료할 때 암시적으로 변경 사항이 커밋됩니다. 선택한 프로파일에 대한 변경 사항을 커밋하지 않으려면 해당 프로파일을 마지막으로 커밋된 상태로 되돌릴 수 있습니다. 이 작업을 수행하면 해당 레벨에서 프로파일의 변경 사항이 취소됩니다. revert 및 cancel 하위 명령이 유사하게 동작합니다.

netcfg 명령줄 모드

명령줄 모드에서는 선택한 프로파일이나 등록 정보에 영향을 주는 하위 명령을 선택한 프로파일이나 등록 정보가 있는 특정 범위에서 수행해야 합니다. 예를 들어, NCU의 등록

정보 값을 가져오려면 해당 특정 NCU의 범위에서 get 하위 명령을 사용합니다. netcfg 대화식 모드에서는 이 명령에 사용할 구문을 비교적 쉽게 이해할 수 있습니다. 하지만 명령줄 모드에서는 구문이 덜 명확할 수 있습니다.

예를 들어, User NCP에서 myncu라는 NCU 속성인 "foo" 등록 정보의 값을 가져오려면 다음 구문을 사용합니다.

```
$ netcfg "select ncp User; select ncu ip myncu; get foo"
```

이 예에서는 다음 정보를 확인합니다.

- 각 범위가 세미콜론으로 구분됩니다.
- 각 범위, 전역 범위에서 한 번, 프로파일 범위에서 한 번 select 하위 명령을 실행합니다.
- "foo" 등록 정보가 있는 범위 내에서 get 하위 명령을 사용합니다.
- 셸이 세미콜론을 해석하지 않게 하려면 곧은따옴표가 필요합니다.

netcfg 명령 파일 모드

명령 파일 모드에서는 파일에서 구성 정보를 가져옵니다. export 하위 명령을 사용하여 이 파일을 생성합니다. 그런 다음 구성을 표준 출력으로 인쇄하거나 -f 옵션을 사용하여 출력 파일을 지정할 수 있습니다. 대화식으로 export 하위 명령을 사용할 수도 있습니다. 자세한 내용은 58 페이지 “지원되는 netcfg 하위 명령”을 참조하십시오.

지원되는 netcfg 하위 명령

다음 netcfg 하위 명령은 대화식 모드와 명령줄 모드에서 지원됩니다. 특정 하위 명령은 각 범위 내에서 다른 의미 체계를 갖습니다. 특정 모드에서 하위 명령을 사용할 수 없는 경우 하위 명령의 설명에서 확인할 수 있습니다.

- cancel

  현재 변경 사항을 지속 저장소에 커밋하지 않고 현재 프로파일 사양을 끝낸 다음 한 레벨 위인 이전 범위에서 계속합니다.

- clear *prop-name*

  지정한 등록 정보의 값을 지웁니다.

- commit

  현재 프로파일을 지속 저장소에 커밋합니다. 커밋하려면 구성이 정확해야 합니다. 따라서 이 작업은 프로파일 또는 객체에 대해 자동으로 verify도 수행합니다. end 또는 exit 하위 명령을 사용하여 현재 범위를 종료하면 자동으로 commit 작업이 시도됩니다.

- create [-t *template*] *object-type* [*class*] *object-name*

지정한 유형과 이름의 메모리 내 프로파일을 만듭니다. `-t` *template* 옵션은 새 프로파일이 *template*과 일치하도록 지정합니다. 여기서 *template*은 동일한 유형의 기존 프로파일 이름입니다. `-t` 옵션을 사용하지 않으면 새 프로파일이 기본값으로 생성됩니다.

- `destroy -a`

  메모리와 지속 저장소에서 사용자 정의 프로파일을 모두 제거합니다.

- `destroy` *object-type* [*class*] *object-name*

  메모리와 지속 저장소에서 지정한 사용자 정의 프로파일을 제거합니다.

주의 – 이 작업은 즉시 반영이며 커밋할 필요가 없습니다. 삭제된 프로파일은 되돌릴 수 없습니다.

- `end`

  현재 프로파일 사양을 끝내고 한 레벨 위인 이전 범위에서 계속합니다. 편집 작업을 끝내기 전에 현재 프로파일이 확인되고 커밋됩니다. `verify` 또는 `commit` 작업이 실패하면 오류 메시지가 표시됩니다. 현재 변경 사항을 커밋하지 않고 작업을 끝내는 옵션이 제공됩니다. 또는 현재 범위를 유지하고 프로파일 편집을 계속할 수 있습니다.

- `exit`

  `netcfg` 대화식 세션을 종료합니다. 현재 세션이 끝나기 전에 현재 프로파일이 확인되고 커밋됩니다. `verify` 또는 `commit` 작업이 실패하면 오류 메시지가 표시됩니다. 현재 변경 사항을 커밋하지 않고 세션을 끝내는 옵션이 제공됩니다. 또는 현재 범위를 유지하고 프로파일 편집을 계속할 수 있습니다.

- `export` [`-d`] [`-f` *output-file*] [*object-type* [*class*] *object-name*]

  현재 또는 지정한 범위의 현재 구성을 표준 출력이나 `-f` 옵션으로 지정된 파일에 인쇄합니다. `-d` 옵션은 출력의 첫째 줄로 `destroy -a` 하위 명령을 생성합니다. 이 하위 명령은 명령 파일에서 사용하기에 적합한 형태로 출력을 생성합니다.

주 – 자동 NCP와 자동, NoNet 및 레거시 위치를 비롯한 시스템 정의 프로파일은 내보낼 수 없습니다.

- `get` [`-V`] *prop-name*

  지정한 등록 정보의 현재 메모리 내 값을 가져옵니다. 기본적으로 등록 정보 이름과 값이 모두 인쇄됩니다. `-V` 옵션을 지정하면 등록 정보 값만 인쇄됩니다.

- `help` [*subcommand*]

  일반 도움말이나 특정 주제에 대한 도움말을 표시합니다.

- `list` [`-a`] [*object-type* [*class*] *object-name*]

현재 범위나 지정한 범위에서 사용될 프로파일, 등록 정보-값 쌍 및 리소스를 모두 나열합니다. -a 옵션을 지정하면 현재 설정을 기준으로 무시될 등록 정보를 포함하여 모든 등록 정보가 나열됩니다.

- `revert`

  프로파일에 대한 현재 변경 사항을 삭제하고 지속 저장소의 값으로 되돌립니다.

- `select` *object-type* [*class*] *object-name*

  지정한 객체를 선택합니다.

- `set` *prop-name*= *value*

  지정한 등록 정보의 현재 메모리 내 값을 설정합니다.

  명령줄 모드에서 수행하면 변경 사항이 지속 저장소에도 즉시 커밋됩니다.

  다중 값 등록 정보의 분리자는 쉼표(,)입니다. 지정된 등록 정보의 개별 값에 쉼표가 포함된 경우 앞에 백슬래시(\)를 추가해야 합니다. 단일 값만 포함된 등록 정보 내의 쉼표는 분리자로 해석되지 않으므로 앞에 백슬래시를 추가할 필요가 없습니다.

- `verify`

  현재 메모리 내 프로파일이나 객체에 유효한 구성이 있는지 확인합니다.

- `walkprop` [`-a`]

  현재 프로파일과 연결된 각 등록 정보를 "검토"합니다. 각 등록 정보에 대해 이름과 현재 값이 표시됩니다. 현재 값을 변경할 수 있는 프롬프트가 제공됩니다. 등록 정보가 사용되지 않는 경우 이전에 지정한 값을 기준으로 등록 정보가 표시되지 않습니다. 예를 들어, `ipv4-addrsrc` 등록 정보를 `static`으로 설정하면 `ipv4-addr` 등록 정보가 사용되지 않으며, `-a` 옵션을 지정하지 않을 경우 검토 또는 나열되지 않습니다.

  사용되는 경우 `-a` 옵션은 지정한 프로파일이나 객체에 사용 가능한 등록 정보를 모두 반복합니다.

  다중 값 등록 정보의 분리자는 쉼표(,)입니다. 지정된 등록 정보의 개별 값에 쉼표가 포함된 경우 앞에 백슬래시(\)를 추가해야 합니다. 단일 값만 포함된 등록 정보 내의 쉼표는 분리자로 해석되지 않으므로 앞에 백슬래시를 추가할 필요가 없습니다.

주 – 이 하위 명령은 대화식 모드에서 사용되는 경우에만 의미가 있습니다.

작업 관련 정보는 4 장, “NWAM 프로파일 구성(작업)”을 참조하십시오.

netadm 명령을 사용하여 프로파일 관리

netadm 명령은 프로파일(NCP, 위치, ENM 및 WLAN)과 NCP를 구성하는 개별 구성 객체인 NCU의 상태를 관리하고 가져오는 데 사용됩니다. 또한 GUI가 없을 경우 netadm 명령을 사용하여 NWAM 데몬(nwamd)과 상호 작용할 수 있습니다. netadm에 대한 자세한 내용은 netadm(1M) 매뉴얼 페이지를 참조하십시오.

다음 netadm 하위 명령이 지원됩니다.

- enable [-p *profile-type*] [-c *ncu-class*] *profile-name*

  지정한 프로파일을 사용으로 설정합니다. 프로파일 이름이 고유하지 않은 경우 프로파일 유형을 지정해야 합니다. 프로파일 유형이 ncu이고 이름이 고유하지 않은 경우(예: 동일한 이름을 가진 링크 및 인터페이스 ncu가 모두 있는 경우) -c 옵션을 사용하여 NCU 클래스를 지정하지 않으면 두 NCU가 모두 사용으로 설정됩니다.

  프로파일 유형은 다음 중 하나여야 합니다.

  - ncp
  - ncu
  - loc
  - enm
  - wlan

  NCU 클래스는 phys 또는 ip로 지정해야 합니다.

- disable [- p *profile-type*] [-c *ncu-class*] *profile-name*

  지정한 프로파일을 사용 안함으로 설정합니다. 프로파일 이름이 고유하지 않은 경우 프로파일 유형을 지정하여 사용 안함으로 설정할 프로파일을 식별해야 합니다. 프로파일 유형이 ncu이고 이름이 고유하지 않은 경우(예: 동일한 이름을 가진 링크 및 인터페이스 ncu가 모두 있는 경우) -c 옵션을 사용하여 NCU 클래스를 지정하지 않으면 두 NCU가 모두 사용 안함으로 설정됩니다.

  프로파일 유형은 다음 중 하나여야 합니다.

  - ncp
  - ncu
  - loc
  - enm
  - wlan

  NCU 클래스는 phys 또는 ip로 지정해야 합니다.

- list [-x] [- p *profile-type*] [-c *ncu-class*] [*profile-name*]

  사용 가능한 모든 프로파일과 프로파일의 현재 상태를 나열합니다. 가능한 상태 값은 다음 절에 나와 있습니다. 프로파일을 이름으로 지정하면 해당 프로파일의 현재 상태만 나열됩니다. 프로파일 이름이 고유하지 않은 경우 해당 이름을 가진 모든

프로파일이 나열됩니다. 또는 프로파일 유형, NCU 클래스 또는 둘 다를 포함하여 특정 프로파일을 식별할 수 있습니다. 프로파일 유형만 지정하면 해당 유형의 모든 프로파일이 나열됩니다.

사용으로 설정된 NCP를 나열하면 해당 NCP를 구성하는 모든 NCU가 포함됩니다.

`-x` 옵션을 지정하면 나열된 각 프로파일의 상태에 대한 확장 설명도 출력 결과에 포함됩니다.

가능한 프로파일 상태 값은 다음과 같습니다.

- `disabled`

  사용으로 설정되지 않은, 수동으로 활성화된 프로파일을 나타냅니다.

- `offline`

  사용으로 설정되지 않은 조건부 활성화 프로파일 또는 시스템 활성화 프로파일을 나타냅니다. 해당 조건이 충족되지 않아 프로파일이 활성 상태가 아닐 수 있습니다. 또는 보다 구체적인 조건이 충족된 다른 프로파일이 대신 활성화되어 프로파일이 활성 상태가 아닐 수 있습니다. 이 조건은 한 번에 하나씩 사용으로 설정해야 하는 프로파일 유형(예: 위치 프로파일)에 적용됩니다.

- `online`

  조건이 충족되었으며 성공적으로 사용으로 설정된 조건부 활성화 프로파일 또는 시스템 활성화 프로파일을 나타냅니다. 또는 사용자 요청 시 성공적으로 사용으로 설정된, 수동으로 활성화된 프로파일을 나타낼 수 있습니다.

- `maintenance`

  프로파일 활성화를 시도했지만 실패했음을 나타냅니다.

- `initialized`

  프로파일이 아무 작업도 수행되지 않은 유효한 구성 객체를 표시함을 나타냅니다.

- `uninitialized`

  프로파일이 시스템에 없는 구성 객체를 표시함을 나타냅니다. 예를 들어, 이 상태는 물리적 링크에 해당하며 시스템에서 제거된 NCU를 나타낼 수 있습니다.

- `show-events`

  NWAM 데몬에서 이벤트 스트림을 수신 대기하고 표시합니다.

- `scan-wifi` *link-name*

  *link-name*으로 지정된 링크의 무선 검색을 나타냅니다.

- `select-wifi` *link-name*

  *link-name*으로 지정된 링크의 검색 결과에서 연결할 무선 네트워크를 선택합니다.

- `help`

  각 하위 명령에 대한 간단한 설명과 함께 사용 메시지를 표시합니다.

작업 관련 정보는 5 장, “NWAM 프로파일 관리(작업)”를 참조하십시오.

NWAM 데몬 개요

NWAM에서 사용되는 두 가지 데몬(nwamd 데몬 및 netcfgd 데몬)이 있습니다. 정책 엔진 데몬인 nwamd는 여러 역할을 수행하여 네트워크 자동 구성을 제어합니다. 저장소 데몬인 netcfgd는 네트워크 구성 저장소에 대한 액세스를 제어합니다.

NWAM 정책 엔진 데몬(nwamd)에 대한 설명

nwamd 데몬은 다음 역할을 맡아 네트워크 자동 구성을 제어합니다.

- **이벤트 수집기**

  이 역할은 경로 지정 소켓과 sysevent 등록을 통해 감지해야 하는 링크 관련 이벤트를 수집합니다. nwamd에서 이 작업을 수행하는 방식의 예로, 이 데몬은 NIC가 시스템에 핫 플러그되었음을 나타내는 EC_DEV_ADD sysevent를 가져옵니다. 이러한 이벤트는 모두 nwamd 이벤트 구조에 패키징된 다음 해당 작업을 수행하는 이벤트 처리 스레드로 전송됩니다.

- **이벤트 처리기**

  이 역할은 이벤트 루프 스레드를 실행하여 관련 이벤트에 응답합니다. 이벤트 처리기는 NWAM 서비스에서 관리되는 여러 객체와 연결된 상태 시스템에서 작동합니다. 이벤트를 처리하는 동안 nwamd 데몬이 네트워크 환경의 변경 사항을 감지하며, 그 결과 프로파일 변경이 트리거될 수 있습니다.

- **이벤트 전달기**

  이 역할은 이러한 이벤트에 대한 관심을 등록한 외부 소비자에게 이벤트를 보냅니다. 이벤트 전달의 예로 사용 가능한 WLAN에 대한 정보가 포함된 무선 검색 이벤트가 있습니다. 이 정보는 NWAM GUI에 유용합니다. 그런 다음 GUI가 사용 가능한 옵션을 사용자에게 표시할 수 있습니다.

- **프로파일 관리자**

  nwamd 데몬을 통해 이러한 프로파일을 관리하려면 다음 정보에 따라 네트워크 구성을 적용해야 합니다.

  - 활성화되는 링크 및 인터페이스
  - 연결된 네트워크의 특성
  - 사용으로 설정된 프로파일에 내장된 대체성 및 종속성
  - 수신된 외부 이벤트

NWAM 저장소 데몬(netcfgd)에 대한 설명

프로파일 데몬인 netcfgd는 네트워크 구성 저장소에 대한 액세스를 제어하고 관리합니다. 이 데몬은 svc:/network/netcfg:default SMF 서비스에 의해 자동으로 시작됩니다. 이 데몬을 사용하면 저장소에서 정보를 읽거나 쓰려는 응용 프로그램에 다음 권한이 부여됩니다.

- solaris.network.autoconf.read
- solaris.network.autoconf.write

권한 부여에 대한 자세한 내용은 auth_attr(4) 매뉴얼 페이지를 참조하십시오. 보안 프로파일에 대한 자세한 내용은 prof_attr(4) 매뉴얼 페이지를 참조하십시오.

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

SMF 네트워크 서비스

Oracle Solaris에서는 네트워크 구성이 여러 SMF 서비스에서 구현됩니다.

- svc:/network/loopback:default – IPv4 및 IPv6 루프백 인터페이스를 만듭니다.
- svc:/network/netcfg:default – 이 서비스는 svc:/network/physical:default 서비스의 필수 조건입니다. 이 서비스는 주요 기능이 netcfgd 데몬을 시작하는 네트워크 구성 저장소를 관리합니다.
- svc:/network/physical:default – 링크를 작동하고 IP 인터페이스를 연결합니다. 이 서비스는 현재 활성 NCP를 기준으로 NWAM 또는 일반 네트워크 구성이 사용되고 있는지 확인합니다. NWAM이 사용 중인 경우 서비스에서 정책 데몬 nwamd를 시작합니다. DefaultFixed NCP가 활성 상태인 경우 서비스에서 nwamd를 중지하고 지속 ipadm 구성을 적용합니다.
- svc:/network/location:default – 이 서비스는 svc:/network/physical:default 서비스에 종속되며 nwamd 데몬이 선택한 Location 프로파일을 활성화합니다.

주 – svc:/network/location:default 서비스에는 현재 위치 프로파일을 저장하는 등록 정보가 있습니다. 이 등록 정보는 직접 조작하지 마십시오. 대신 NWAM GUI 또는 CLI를 사용하여 이러한 유형의 변경을 수행합니다.

NWAM 보안 개요

NWAM의 보안은 다음 구성 요소를 포함하도록 설계되었습니다.

- CLI(netcfg 및 netadm 명령)
- NWAM GUI
- NWAM 프로파일 저장소 데몬(netcfgd)
- 정책 엔진 데몬(nwamd)
- NWAM 라이브러리(libnwam)

netcfgd 데몬은 모든 네트워크 구성 정보가 저장되는 저장소를 제어합니다. netcfg 명령, NWAM GUI 및 nwamd 데몬은 저장소에 액세스하기 위해 모두 netcfgd 데몬에 요청을 보냅니다. 이러한 기능 구성 요소는 NWAM 라이브러리 libnwam을 통해 요청합니다.

nwamd 데몬은 시스템 이벤트를 받고, 네트워크를 구성하고, 네트워크 구성 정보를 읽는 정책 엔진입니다. NWAM GUI 및 netcfg 명령은 네트워크 구성을 확인하고 수정하는 데 사용할 수 있는 구성 도구입니다. 이러한 구성 요소는 새 구성을 시스템에 적용해야 하는 경우 NWAM 서비스를 새로 고치는 데도 사용됩니다.

NWAM과 관련된 권한 및 프로파일

현재 NWAM 구현에서는 다음 권한을 사용하여 특정 작업을 수행합니다.

- solaris.network.autoconf.read – netcfgd 데몬에 의해 확인된 NWAM 구성 데이터를 읽도록 설정합니다.
- solaris.network.autoconf.write – netcfgd 데몬에 의해 확인된 NWAM 구성 데이터를 쓰도록 설정합니다.
- solaris.network.autoconf.select – nwamd 데몬에 의해 확인된 새 구성 데이터를 적용하도록 설정합니다.
- solaris.network.autconf.wlan – 알려진 WLAN 구성 데이터를 쓰도록 설정합니다.

이러한 권한은 auth_attr 데이터베이스에 등록됩니다. 자세한 내용은 auth_attr(4) 매뉴얼 페이지를 참조하십시오.

Network Autoconf User 및 Network Autoconf Admin이라는 두 개의 보안 프로파일이 제공됩니다. User 프로파일에는 read, select 및 wlan 권한이 있습니다. Admin 프로파일은 write 권한을 추가합니다. Network Autoconf User 프로파일이 Console User 프로파일에 할당됩니다. 따라서 콘솔에 로그인한 누구든지 기본적으로 프로파일을 확인하고 사용 및 사용 안함으로 설정할 수 있습니다. Console User는 solaris.network.autoconf.write 권한이 할당되지 않으므로 NCP, NCU, 위치 또는 ENM을 만들거나 수정할 수 없습니다. 하지만 Console User는 WLAN을 확인, 만들기 및 수정할 수 있습니다.

NWAM 사용자 인터페이스를 사용하는 데 필요한 권한

Console User 권한을 가진 누구든지 NWAM 명령 netcfg 및 netadm을 사용하여 NWAM 프로파일을 확인 및 사용으로 설정할 수 있습니다. 이러한 권한은 /dev/console에서 시스템에 로그인한 모든 사용자에게 자동으로 할당됩니다.

netcfg 명령을 사용하여 NWAM 프로파일을 수정하려면 solaris.network.autoconf.write 권한 또는 Network Autoconf Admin 프로파일이 필요합니다.

profiles 명령에 프로파일 이름을 사용하면 권한 프로파일과 연결된 권한을 확인할 수 있습니다. 자세한 내용은 profiles(1) 매뉴얼 페이지를 참조하십시오.

예를 들어, Console User 권한 프로파일과 연결된 권한을 확인하려면 다음 명령을 사용합니다.

```
$ profiles -p "Console User" info
Found profile in files repository.
    name=Console User
    desc=Manage System as the Console User
    auths=solaris.system.shutdown,solaris.device.cdrw,solaris.smf.manage.vbiosd,
    solaris.smf.value.vbiosd
    profiles=Suspend To RAM,Suspend To Disk,Brightness,CPU Power Management,
    Network Autoconf User,Desktop Removable Media User
    help=RtConsUser.html
```

NWAM GUI에는 권한이 없는 다음 세 가지 구성 요소가 포함됩니다. 시작 방식 및 수행해야 하는 작업에 따라 이러한 구성 요소에 권한이 부여됩니다.

- **NWAM 관련 패널 존재**

  이 구성 요소는 사용자가 NWAM과 상호 작용할 수 있게 하는 데스크탑의 패널 애플릿입니다. 이 패널은 모든 사용자가 실행할 수 있으며, 시스템의 자동 구성을 모니터하고 이벤트 알림을 처리하는 데 사용됩니다. 이 패널을 사용하여 기본 네트워크 구성 작업(예: WiFi 네트워크 선택 또는 수동으로 위치 전환)을 수행할 수도 있습니다. 이러한 유형의 작업을 수행하려면 `Network Autoconf User` 권한 프로파일이 필요합니다. 패널이 `/dev/console`에서 로그인한 사용자의 권한으로 실행되고 있어 `Console User` 프로파일이 있으므로 이 권한 프로파일은 기본 구성에서 사용할 수 있습니다.

- **NWAM GUI**

  NWAM GUI는 데스크탑에서 NWAM과 상호 작용하기 위한 주요 수단입니다. GUI를 사용하여 네트워크 상태를 확인하고, NCP 및 위치 프로파일을 만들고 수정하며, 구성된 ENM을 시작 및 중지합니다. GUI와 상호 작용하려면 `solaris.network.autoconf` 권한 네 개와 `Network Autoconf Admin` 프로파일이 필요합니다. 기본적으로 `Console User` 프로파일에는 GUI를 사용하여 네트워크 상태와 프로파일을 확인하기 위한 권한이 있습니다. 또한 GUI를 사용하여 프로파일을 수정하려면 `solaris.network.autoconf.write` 권한 또는 `Network Autoconf Admin` 프로파일이 필요합니다.

다음 방식 중 하나로 추가 권한을 부여 받을 수 있습니다.

- 특정 사용자에게 `Network Autoconf Admin` 프로파일을 할당합니다.

  사용자의 `/etc/user_attr` 파일을 편집하여 지정된 사용자에게 직접 적절한 권한 또는 권한 프로파일을 할당할 수 있습니다.

- `Console User`에게 `Network Autoconf Admin` 프로파일을 할당합니다.

  기본적으로 할당되는 `Network Autoconf User` 프로파일 대신 이 프로파일을 `Console User`에게 할당할 수 있습니다. 이 프로파일을 할당하려면 `/etc/security/prof_attr` 파일의 항목을 편집합니다.

◆ ◆ ◆ 4 장

4 NWAM 프로파일 구성(작업)

이 장에서는 netcgf 명령을 사용하여 수행할 수 있는 NWAM 프로파일 구성 작업에 대해 설명합니다. 이러한 구성 작업에는 프로파일 만들기, 수정 및 삭제와 NWAM 구성을 제어하는 다양한 SMF 서비스 관리가 포함됩니다. 이 장에서는 대화식 모드와 명령줄 모드 둘 다에서 netcfg 명령을 사용하는 방법에 대해 설명합니다.

이 장에서는 다음 항목을 다룹니다.

프로파일 상태 표시, 프로파일 활성화 및 비활성화, netadm 명령을 사용한 알려진 무선 네트워크 관리에 대한 자세한 내용은 5 장, "NWAM 프로파일 관리(작업)"를 참조하십시오.

NWAM과 상호 작용하는 방법 및 데스크탑에서 네트워크 구성을 관리하는 방법에 대한 자세한 내용은 6 장, "NWAM 그래픽 사용자 인터페이스 정보"를 참조하십시오.

NWAM에 대한 소개는 2 장, "NWAM 소개"를 참조하십시오.

netcfg 사용자 인터페이스 모드에 대한 설명을 비롯한 NWAM에 대한 자세한 개요 정보는 3 장, "NWAM 구성 및 관리(개요)"를 참조하십시오.

프로파일 만들기

netcfg(1M) 매뉴얼 페이지에 설명된 netcfg 명령은 NWAM 명령줄 인터페이스의 두 가지 관리 명령 중 하나입니다.

Console User 권한을 가진 사용자는 netcfg 명령을 사용하여 프로파일 구성 데이터를 표시하고 알려진 WLAN 객체를 표시, 만들기 및 수정할 수 있습니다. 이러한 권한은 /dev/console에서 시스템에 로그인한 모든 사용자에게 자동으로 할당됩니다. Network Autoconf Admin 프로파일을 가진 사용자는 모든 유형의 NWAM 프로파일과 구성 객체를 만들고 수정할 수도 있습니다. 자세한 내용은 64 페이지 "NWAM 보안 개요"를 참조하십시오.

netcfg 명령을 사용하여 사용자 정의 프로파일을 선택, 만들기, 수정 및 삭제할 수 있습니다. 이 명령은 대화식 모드 또는 명령줄 모드에서 사용할 수 있습니다. netcfg 명령은 프로파일 구성 정보를 명령 파일로 내보내는 기능도 지원합니다.

다음 프로파일 및 구성 객체를 만들고, 수정 및 제거할 수 있습니다.

- NCP(네트워크 구성 프로파일)
- 위치 프로파일
- ENM(외부 네트워크 수정자)
- 알려진 WLAN(무선 LAN)
- NCU(네트워크 구성 단위)

명령줄 모드에서 프로파일 만들기

명령줄에서 프로파일을 만드는 데 사용할 기본 명령 구문은 다음과 같습니다.

netcfg create [-t *template* **]** *object-type* **[** *class* **]** *object-name*

create — 지정한 유형과 이름의 메모리 내 프로파일(또는 구성 객체)을 만듭니다.

-t *template* — 새 프로파일이 *template*과 일치하도록 지정합니다. 여기서 *template*은 동일한 유형의 기존 프로파일 이름입니다. -t 옵션을 사용하지 않으면 새 프로파일이 기본값으로 생성됩니다.

object-type — 만들 프로파일의 유형을 지정합니다.

object-type 옵션에 대해 다음 값 중 하나를 지정할 수 있습니다.

- ncp
- ncu
- loc
- enm
- wlan

netcfg select 명령을 사용하여 특정 객체를 선택하려면 ncu를 제외하고 *object-type* 옵션으로 지정된 모든 프로파일을 먼저 전역 범위에서 만들어야 합니다.

class *object-type*으로 지정된 프로파일의 클래스를 지정합니다. 이 매개변수는 ncu 객체 유형에만 사용되며, 두 개의 가능한 값 phys 또는 ip가 있습니다.

object-name 사용자 정의 프로파일의 이름을 지정합니다. NCU의 경우 *object-name*은 해당 링크 또는 인터페이스의 이름입니다. 다른 모든 프로파일 유형의 경우 *object-name*은 사용자 정의 이름입니다.

예를 들어, User라는 NCP를 만들려면 다음 명령을 입력합니다.

```
$ netcfg create ncp User
```

여기서 ncp는 *object-type*이고 User는 *object-name*입니다.

주 – NCP를 만드는 경우 class 옵션이 필요하지 않습니다.

필요에 따라 자동 NCP의 복사본을 템플리트로 사용한 다음 아래와 같이 해당 프로파일을 변경할 수 있습니다.

```
$ netcfg create -t Automatic ncp
```

office라는 이름으로 위치 프로파일을 만들려면 다음 명령을 입력합니다.

```
$ netcfg create loc office
```

대화식으로 프로파일 만들기

대화식 모드로 netcfg 명령을 사용하면 다음 작업을 수행할 수 있습니다.

- 프로필을 만듭니다.
- 프로파일을 선택하고 수정합니다.
- 프로파일에 대한 모든 필수 정보가 설정되고 유효한지 확인합니다.
- 새 프로파일에 대한 변경 사항을 커밋합니다.
- 변경 사항을 지속 저장소에 커밋하지 않고 현재 프로파일 구성을 취소합니다.
- 프로파일에 대한 변경 사항을 되돌립니다.

NCP 만들기

대화식 모드에서 프로파일을 만들면 다음 범위 중 하나에 있는 명령 프롬프트가 생성됩니다.

- NCP가 생성된 경우 NCP 범위
- 위치 프로파일, ENM 또는 WLAN 객체가 생성된 경우 프로파일 범위

NCP 또는 NCU를 만들면 포커스가 객체 범위로 이동하고 지정한 프로파일의 기본 등록 정보가 검토됩니다.

대화식으로 NCP를 만들려면 먼저 `netcfg` 대화식 세션을 시작합니다. 그런 다음 `create` 하위 명령을 사용하여 다음과 같이 새 NCP `User`를 만듭니다.

```
$ netcfg
netcfg> create ncp User
netcfg:ncp:User>
```

NCP의 NCU 만들기

NCP는 근본적으로 NCU 세트로 구성된 컨테이너입니다. 모든 NCP에 링크와 인터페이스 NCU가 모두 포함됩니다. 링크 NCU는 링크 구성과 링크 선택 정책을 모두 지정합니다. 인터페이스 NCU는 인터페이스 구성 정책을 지정합니다. IP 연결이 필요한 경우 링크 및 인터페이스 NCU가 모두 필요합니다. `netcfg` 명령을 사용하거나 GUI를 사용하여 명시적으로 NCU를 추가 또는 제거해야 합니다.

주 – 현재 시스템에 설치된 링크에 대한 상관 관계가 없는 NCU를 추가할 수 있습니다. 또한 현재 시스템에 설치된 링크에 매핑되는 NCU를 제거할 수 있습니다.

대화식 모드 또는 명령줄 모드에서 `netcfg` 명령을 사용하여 NCU를 만들 수 있습니다. NCU를 만드는 과정에는 여러 작업이 포함되므로 NCU 및 모든 등록 정보는 만드는 한 줄 명령을 생성하는 대신 대화식 모드에서 NCU를 만드는 것이 더 쉽고 효율적입니다. NCU는 처음에 NCP를 만들 때나 그 이후에 생성될 수 있습니다. NCU를 만들거나 수정하는 프로세스에는 일반 NCU 등록 정보를 설정하는 작업과 구체적으로 각 NCU 유형에 적용되는 등록 정보를 설정하는 작업이 포함됩니다.

NCP의 NCU를 만드는 동안 제공되는 등록 정보는 해당 특정 NCP를 만드는 동안 선택한 사항을 기준으로 가장 관련된 항목입니다.

대화식으로 NCU를 만드는 경우 `netcfg`는 각 관련 등록 정보를 검토하고 기본값이 있는 경우 기본값과 가능한 값을 모두 표시합니다. 값을 지정하지 않고 Return 키를 눌러 기본값을 적용하거나(또는 기본값이 없는 경우 등록 정보를 비워 둠), 대체 값을 지정할 수 있습니다. NCP의 NCU를 만드는 동안 표시되는 등록 정보는 이미 선택한 사항을 기준으로 관련된 항목입니다. 예를 들어, 인터페이스 NCU의 `ipv4-addrsrc` 등록 정보에 대해 `dhcp`를 선택하면 `ipv4-addr` 등록 정보의 값을 지정하라는 메시지가 표시되지 않습니다.

다음 표에서는 NCU를 만들거나 수정할 때 지정할 수 있는 모든 NCU 등록 정보에 대해 설명합니다. 일부 등록 정보는 두 NCU 유형에 모두 적용됩니다. 다른 등록 정보는 링크 NCU 또는 인터페이스 NCU에 적용됩니다. 이러한 등록 정보를 지정할 때 적용되는 규칙 및 조건을 비롯한 모든 NCU 등록 정보에 대한 전체 설명은 netcfg(1M) 매뉴얼 페이지를 참조하십시오.

표 4–1 NCU를 만들거나 수정하는 NCU 등록 정보

등록 정보	설명	가능한 값	NCU 유형
`type`	NCU 유형(링크 또는 인터페이스)을 지정합니다.	`link` 또는 interface	링크 및 인터페이스
`class`	NCU 클래스를 지정합니다.	`phys`(링크 NCU) 또는 `ip`(인터페이스 NCU)	링크 및 인터페이스
`parent`	이 NCU가 속하는 NCP를 지정합니다.	*parent-NCP*	링크 및 인터페이스
`enabled`	NCU를 사용 또는 사용 안함으로 설정할 것인지 지정합니다. 이 등록 정보는 읽기 전용입니다. `netadm` 명령이나 NWAM GUI를 사용하여 NCU를 사용 또는 사용 안함으로 설정할 때 간접적으로만 변경됩니다.	`true` 또는 `false`	링크 및 인터페이스
`activation-mode`	NCU의 자동 활성화에 대한 트리거 유형을 지정합니다.	`manual` 또는 `prioritized` 기본값은 `manual`입니다.	링크
`priority-group`	그룹 우선 순위 번호를 지정합니다.	`0`(유선 링크) 또는 `1`(무선 링크) 사용자 정의 NCP의 경우 다른 정책을 지정할 수 있습니다. 예를 들어, 무선 링크 1이 우선 순위 1이고 유선 링크 1이 우선 순위 2이고 유선 링크 2가 우선 순위 3입니다. **주** – 숫자가 작을수록 더 높은 우선 순위를 나타냅니다.	링크

표 4–1 NCU를 만들거나 수정하는 NCU 등록 정보 *(계속)*

등록 정보	설명	가능한 값	NCU 유형
priority-mode	activation-mode 등록 정보를 prioritized로 설정한 경우 우선 순위 그룹의 활성화 동작을 결정하는 데 사용되는 모드를 지정합니다.	exclusive, shared 또는 all 이러한 값을 지정할 때 적용되는 규칙은 netcfg(1M) 매뉴얼 페이지를 참조하십시오.	링크
link-mac-addr	이 링크에 할당되는 MAC 주소를 지정합니다. 기본적으로 NWAM은 출하 시 할당된 MAC 주소나 다른 기본 MAC 주소를 사용합니다. 여기서 다른 값을 설정하여 해당 선택을 대체할 수 있습니다.	48비트 MAC 주소가 포함된 문자열	
link-autopush	링크를 열 때 자동으로 링크에 푸시되는 모듈을 식별합니다.	문자열 목록(링크에 푸시되는 모듈) autopush(1M)를 참조하십시오.	링크
link-mtu	물리적 링크의 기본 MTU로 자동 설정됩니다. 등록 정보를 다른 값으로 설정하여 이 값을 대체할 수 있습니다.	링크의 MTU 크기	링크
ip-version	사용할 IP 버전을 지정합니다. 여러 값을 할당할 수 있습니다.	ipv4 및 ipv6 기본값은 ipv4, ipv6입니다.	인터페이스
ipv4-addrsrc	이 NCU에 할당된 IPv4 주소의 소스를 식별합니다. 여러 값을 할당할 수 있습니다.	dhcp 및 static 기본값은 dhcp입니다.	인터페이스
ipv6-addrsrc	이 NCU에 할당된 IPv6 주소의 소스를 식별합니다. 여러 값을 할당할 수 있습니다.	dhcp, autoconf 또는 static 기본값은 dhcp, autoconf입니다.	인터페이스
ipv4-addr	이 NCU에 할당할 IPv4 주소를 하나 이상 지정합니다.	할당할 하나 이상의 IPv4 주소	인터페이스

표 4–1 NCU를 만들거나 수정하는 NCU 등록 정보 (계속)

등록 정보	설명	가능한 값	NCU 유형
ipv6-addr	이 NCU에 할당할 IPv6 주소를 하나 이상 지정합니다.	할당할 하나 이상의 IPv6 주소	인터페이스
ipv4-default-route	IPv4 주소의 기본 경로를 지정합니다.	IPv4 주소	인터페이스
ipv6-default-route	IPv6 주소의 기본 경로를 지정합니다.	IPv6 주소	인터페이스

▼ 대화식으로 NCP를 만드는 방법

다음 절차에서는 대화식 모드에서 NCP를 만드는 방법에 대해 설명합니다.

참고 – NWAM에서 초기 프로파일 생성 도중 수행하는 검토 프로세스는 이전 선택 사항을 고려하여 적합한 등록 정보에 대해서만 확인 메시지가 표시되게 합니다. 또한 이 절차에서 설명하는 `verify` 하위 명령은 구성을 확인합니다. 필요한 값이 없는 경우 알림이 표시됩니다. 프로파일을 만들거나 수정할 때 명시적으로 `verify` 하위 명령을 사용하거나, `commit` 하위 명령을 사용하여 암시적으로 변경 사항을 저장할 수 있습니다.

1 netcfg 대화식 세션을 시작합니다.

```
$ netcfg
netcfg>
```

2 NCP를 만듭니다.

```
netcfg> create ncp User
netcfg:ncp:User>
```

여기서 `ncp`는 프로파일 유형이고 `User`는 프로파일 이름입니다.

NCP를 만들면 자동으로 NCP 범위로 이동됩니다. 위치, ENM 또는 WLAN 객체를 만드는 경우 명령 프롬프트에서 프로파일 범위로 이동됩니다.

3 NCP의 링크 및 인터페이스 NCU를 만듭니다.

a. 링크 NCU를 만들려면 다음 명령을 입력합니다.

```
netcfg:ncp:User> create ncu phys net0
Created ncu 'net0', Walking properties ...
```

여기서 `ncu`는 객체 유형이고, `phys`는 클래스이고, `net0`(예로**만** 사용됨)은 객체 이름입니다.

NCU를 만들면 객체 범위로 이동되고 객체의 기본 등록 정보가 검토됩니다.

b. 인터페이스 NCU를 만들려면 다음 명령을 입력합니다.

```
netcfg:ncp:User> create ncu ip net0
Created ncu 'net0'. walking properties ...
```

여기서 ncu는 객체 유형이고, ip는 클래스이고, net0(예로**만** 사용됨)은 객체 이름입니다.

NCU를 만들면 객체 범위로 이동되고 객체의 기본 등록 정보가 검토됩니다.

NCU를 만드는 동안 class 옵션을 사용하여 두 가지 NCU 유형을 구별합니다. 이 옵션은 특히 여러 NCU 유형이 동일한 이름을 공유하는 경우에 유용합니다. class 옵션을 생략하면 동일한 이름을 공유하는 NCU를 구별하기가 훨씬 어렵습니다.

4 만든 NCU에 대한 적절한 등록 정보를 추가합니다.

주 – NCP의 필수 NCU가 모두 생성될 때까지 3단계와 4단계를 반복합니다.

---

5 NCU를 만드는 동안 또는 지정한 NCU의 등록 정보 값을 설정할 때 verify 하위 명령을 사용하여 변경한 사항이 올바른지 확인합니다.

```
netcfg:ncp:User:ncu:net0> verify
All properties verified
```

6 NCU에 대해 설정한 등록 정보를 커밋합니다.

```
netcfg:ncp:User:ncu:net0> commit
committed changes.
```

또는 end 하위 명령을 사용하여 암시적 커밋을 수행할 수 있습니다. 이 경우 대화식 세션이 한 레벨 위의 다음 상위 범위로 이동합니다. 이 인스턴스에서 NCP를 만들고 NCU를 추가한 경우 NCP 범위에서 직접 대화식 세션을 종료할 수 있습니다.

주 –

- 대화식 모드에서는 커밋할 때까지 변경 사항이 지속 저장소에 저장되지 않습니다. commit 하위 명령을 사용하면 전체 프로파일이 커밋됩니다. 지속 저장소의 일관성을 유지하기 위해 커밋 작업에는 확인 단계도 포함됩니다. 확인이 실패하면 커밋도 실패합니다. 암시적 커밋이 실패할 경우 현재 변경 사항을 커밋하지 않고 대화식 세션을 끝내거나 종료하는 옵션이 제공됩니다. 또는 현재 범위를 유지하고 프로파일 변경을 계속할 수 있습니다.
- 변경 사항을 취소하려면 cancel 또는 revert 하위 명령을 사용합니다.

  cancel 하위 명령은 현재 변경 사항을 지속 저장소에 커밋하지 않고 현재 프로파일 구성을 끝낸 다음 대화식 세션을 한 레벨 위의 다음 상위 범위로 이동합니다. revert 하위 명령은 변경 사항을 실행 취소하고 이전 구성을 다시 읽습니다. revert 하위 명령을 사용하는 경우 대화식 세션이 동일한 범위로 유지됩니다.

---

7 list 하위 명령을 사용하여 NCP 구성을 표시합니다.

8 NCP 구성을 완료했으면 대화식 세션을 종료합니다.

```
netcfg:ncp:User> exit
```

exit 하위 명령을 사용하여 netcfg 대화식 세션을 끝낼 때마다 현재 프로파일이 확인되고 커밋됩니다. 확인 또는 커밋 작업이 실패하면 해당 오류 메시지가 실행되며 현재 변경 사항을 커밋하지 않고 종료하는 옵션이 제공됩니다. 또는 현재 범위를 유지하고 프로파일 변경을 계속할 수 있습니다.

주 – netcfg 대화식 세션을 종료하지 않고 범위를 종료하려면 end 명령을 입력합니다.

```
netcfg:ncp:User> end
netcfg>
```

예 4–1 대화식으로 NCP 만들기

다음 예에서는 NCP 한 개와 NCU 두 개(링크 한 개 및 인터페이스 한 개)가 생성됩니다.

```
$ netcfg
netcfg> create ncp User
netcfg:ncp:User> create ncu phys net0
Created ncu 'net0', Walking properties ...
activation-mode (manual) [manual|prioritized]>
link-mac-addr>
link-autopush>
link-mtu>
netcfg:ncp:User:ncu:net0> end
Committed changes
netcfg:ncp:User> create ncu ip net0
Created ncu 'net0'. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]>
ipv4-default-route>
netcfg:ncp:User:ncu:net0> verify
All properties verified
netcfg:ncp:User:ncu:net0> end
Committed changes
netcfg:ncp:User> list
NCUs:
        phys    net0
        ip      net0
netcfg:ncp:User> list ncu phys net0
ncu:net0
        type                    link
        class                   phys
        parent                  "User"
        activation-mode         manual
        enabled                 true
netcfg:ncp:User> list ncu ip net0
ncu:net0
        type                    interface
        class                   ip
        parent                  "User"
        enabled                 true
```

```
        ip-version              ipv4
        ipv4-addrsrc            dhcp
        ipv6-addrsrc            dhcp,autoconf
netcfg:ncp:User> exit
$
```

이 예에서는 값 ipv4가 선택되었으므로 사용되지 않는 ipv6-addrsrc 등록 정보에 대해서는 프롬프트가 표시되지 않습니다. 이와 마찬가지로, phys NCU의 경우 priority-group 등록 정보의 기본값(수동 활성화)이 수락되었으므로 다른 조건부 관련 등록 정보가 적용되지 않습니다.

예 4–2 기존 NCP의 NCU 만들기

기존 NCP의 NCU를 만들거나 기존 프로파일의 등록 정보를 수정하려면 netcfg 명령에 select 하위 명령을 사용합니다.

다음 예에서는 기존 NCP에 대해 IP NCU 한 개가 생성됩니다. 대화식 모드에서 기존 프로파일을 수정하는 프로세스는 프로파일을 만드는 것과 유사합니다. 다음 예와 예 4–1에는 차이점이 있는데, 다음 예에서는 NCP가 이미 있어서 create 하위 명령 대신 select 명령이 사용된다는 것입니다.

```
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> list
NCUs:
        phys   net0
netcfg:ncp:User> create ncu ip net0
Created ncu 'net0'. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]>
ipv4-default-route>
netcfg:ncp:User:ncu:net0> end
Committed changes
netcfg:ncp:User> list
NCUs:
        phys    net0
        ip      net0
netcfg:ncp:User> list ncu phys net0
ncu:net0
        type                    link
        class                   phys
        parent                  "User"
        activation-mode         manual
        enabled                 true
netcfg:ncp:User> list ncu ip net0
NCU:net0
        type                    interface
        class                   ip
        parent                  "User"
        enabled                 true
        ip-version              ipv4
        ipv4-addrsrc            dhcp
        ipv6-addrsrc            dhcp,autoconf
netcfg:ncp:User> exit
$
```

위치 프로파일 만들기

위치 프로파일에는 기본 링크 및 IP 연결과 직접 관련이 없는 네트워크 구성 설정을 정의하는 등록 정보가 포함됩니다. 일부 예에는 필요한 경우 함께 적용되는 IP 필터 설정과 이름 지정 서비스가 포함되어 있습니다. 항상 시스템에서는 위치 프로파일 한 개와 NCP 한 개가 활성 상태여야 합니다. 시스템 정의 위치와 사용자 정의 위치가 있습니다. 시스템 위치는 위치를 지정하지 않았거나 수동으로 활성화된 위치가 사용으로 설정되지 않았으며 조건부 활성화된 위치의 조건이 하나도 충족되지 않은 경우와 같이 특정 조건에서 NWAM이 선택하는 기본값입니다. 시스템 정의 위치에는 `system` 활성화 모드가 있습니다. 사용자 정의 위치는 네트워크 연결에서 얻은 IP 주소와 같은 네트워크 조건에 따라 수동으로 또는 조건부로 활성화되도록 구성된 위치입니다.

위치 프로파일 수동 활성화(사용으로 설정)에 대한 자세한 내용은 104 페이지 “프로파일 활성화 및 비활성화”를 참조하십시오.

대화식 모드 또는 명령줄 모드에서 `netcfg` 명령을 사용하여 위치를 만들 수 있습니다. 위치 프로파일을 만드는 경우 해당 위치의 특정 구성 매개변수를 정의하는 값을 지정하여 위치의 등록 정보를 설정해야 합니다. 위치 등록 정보는 그룹별로 분류되며, 여기서 그룹은 구성 기본 설정의 특정 클래스를 나타냅니다.

또한 위치 등록 정보는 NWAM에 의해 저장소에 저장됩니다. 특정 위치 프로파일을 활성화하면 NWAM이 해당 위치에 대해 설정된 등록 정보를 기준으로 네트워크를 자동 구성합니다. 위치를 만들거나 수정하는 경우 프로파일 구성 방식을 정의하는 다양한 등록 정보를 설정해야 합니다. 프로파일 구성 방식에 따라 NWAM이 네트워크를 자동 구성하는 방식이 결정됩니다. 구성 프로세스에서 제공되는 등록 정보는 이전에 선택한 사항을 기준으로 가장 관련된 항목입니다.

다음 표에서는 지정할 수 있는 모든 위치 등록 정보에 대해 설명합니다. 위치 등록 정보는 그룹별로 분류됩니다. 이러한 등록 정보를 지정할 때 적용되는 규칙, 조건 및 종속성을 비롯한 모든 위치 등록 정보에 대한 전체 설명은 `netcfg(1M)` 매뉴얼 페이지를 참조하십시오.

표 4–2 위치 등록 정보 및 해당 설명

등록 정보 그룹 및 설명	등록 정보 값 및 설명
선택 기준 위치의 활성화 또는 비활성화 방식과 시기에 대한 기준을 지정합니다.	■ `activation-mode` `activation-mode` 등록 정보의 가능한 값은 `manual`, `conditional-any` 및 `conditional-all`입니다. ■ `conditions`
시스템 도메인 NIS 이름 지정 서비스에서 직접 사용할 호스트의 도메인 이름을 결정합니다.	`system-domain` 등록 정보는 `default-domain` 등록 정보로 구성됩니다. 이 등록 정보는 원격 프로시저 호출(RPC) 교환에 사용되는 시스템 차원의 도메인을 지정합니다.

표 4–2 위치 등록 정보 및 해당 설명 (계속)

등록 정보 그룹 및 설명	등록 정보 값 및 설명
이름 서비스 정보 사용할 이름 지정 서비스 및 이름 지정 서비스 스위치 구성을 지정합니다.	다음은 지정한 이름 지정 서비스의 등록 정보 목록입니다. ■ domain-name ■ nameservices ■ nameservices-config-file ■ dns-nameservice-configsrc ■ dns-nameservice-domain ■ dns-namservice-servers ■ dns-nameservice-search ■ dns-nameservice-sortlist ■ dns-nameservice-options ■ nis-nameservice-configsrc ■ nis-namservice-servers ■ ldap-nameservice-configsrc ■ ldap-namservice-servers 이러한 등록 정보에 대한 자세한 내용은 netcfg(1M) 매뉴얼 페이지의 "위치 등록 정보" 절을 참조하십시오.
NFSv4 도메인 NFSv4 도메인을 지정합니다.	시스템의 nfsmapid_domain 등록 정보에 사용되는 값입니다. 이 값은 위치가 활성 상태인 동안 nfsmapid 매뉴얼 페이지에 설명된 대로 nfsmapid_domain SMF 등록 정보를 설정하는 데 사용됩니다. 이 등록 정보를 설정하지 않으면 위치가 활성 상태일 때 시스템의 nfsmapid_property가 지워집니다. 자세한 내용은 nfsmapid(1M) 매뉴얼 페이지를 참조하십시오.
IP 필터 구성 IP 필터 구성에 사용되는 매개변수를 지정합니다. 이러한 등록 정보의 경우 IP 필터와 NAT 규칙이 포함된 해당 ipf 및 ipnat 파일의 경로가 지정됩니다.	■ ipfilter-config-file ■ ipfilter-v6-config-file ■ ipnat-config-file ■ ippool-config-file 구성 파일을 지정하면 식별된 파일에 포함된 규칙이 해당 ipfilter 부속 시스템에 적용됩니다.
IPsec의 구성 파일 IPsec 구성에 사용할 파일을 지정합니다.	■ ike-config-file ■ ipsecpolicy-config-file

▼ 대화식으로 위치 프로파일을 만드는 방법

다음 절차에서는 위치 프로파일을 만드는 방법에 대해 설명합니다.

참고 – NWAM에서 초기 프로파일 생성 도중 수행하는 검토 프로세스는 이전에 입력한 값을 고려하여 적합한 등록 정보에 대해서만 확인 메시지가 표시되게 합니다. 또한 `verify` 하위 명령은 구성이 올바른지 확인합니다. 필요한 값이 없는 경우 알림이 표시됩니다. 프로파일 구성을 만들거나 수정할 때 명시적으로 `verify` 하위 명령을 사용하거나, `commit` 하위 명령을 사용하여 암시적으로 변경 사항을 저장할 수 있습니다.

1 `netcfg` 대화식 세션을 시작합니다.

```
$ netcfg
netcfg>
```

2 위치를 만들거나 선택합니다.

```
netcfg> create loc office
netcfg:loc:office>
```

이 예에서는 `office` 위치가 생성됩니다.

위치를 만들면 자동으로 이 위치의 프로파일 범위로 이동됩니다.

3 위치에 대한 적절한 등록 정보를 설정합니다.

4 프로파일 구성을 표시합니다.

예를 들어, 다음 출력 결과에는 `office` 위치의 등록 정보가 표시됩니다.

```
netcfg:loc:office> list
LOC:office
    activation-mode             conditional-any
    conditions                  "ncu ip:wpi0 is active"
    enabled                     false
    nameservices                dns
    nameservices-config-file    "/etc/nsswitch.dns"
    dns-nameservice-configsrc   dhcp
    ipfilter-config-file        "/export/home/test/wifi.ipf.conf"
```

5 프로파일 구성이 올바른지 확인합니다.

다음 예에서는 `office` 위치의 구성이 확인됩니다.

```
netcfg:loc:office> verify
All properties verified
```

6 확인을 완료했으면 위치 프로파일을 지속 저장소으로 커밋합니다.

```
netcfg:loc:office> commit
Committed changes
```

또는 `end` 하위 명령을 사용하여 세션을 끝낼 수 있습니다. 이 경우 프로파일 구성도 저장됩니다.

```
netcfg:loc:office> end
Committed changes
```

주 –

- 대화식 모드에서는 커밋할 때까지 변경 사항이 지속 저장소에 저장되지 않습니다. commit 하위 명령을 사용하면 전체 프로파일이 커밋됩니다. 지속 저장소의 일관성을 유지하기 위해 커밋 작업에는 확인 단계도 포함됩니다. 확인이 실패하면 커밋도 실패합니다. 암시적 커밋이 실패할 경우 현재 변경 사항을 커밋하지 않고 대화식 세션을 끝내거나 종료하는 옵션이 제공됩니다. 또는 현재 범위를 유지하고 프로파일 변경을 계속할 수 있습니다.
- 변경 사항을 취소하려면 cancel 하위 명령을 사용합니다.

  cancel 하위 명령은 현재 변경 사항을 지속 저장소에 커밋하지 않고 현재 프로파일 구성을 끝낸 다음 대화식 세션을 한 레벨 위의 다음 상위 범위로 이동합니다.

7 대화식 세션을 종료합니다.

```
netcfg> exit
Nothing to commit
$
```

예 4–3 대화식으로 위치 프로파일 만들기

다음 예에서는 office라는 위치가 생성됩니다.

```
$ netcfg
netcfg> create loc office
Created loc 'office'.  Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-any
conditions> ncu ip:wpi0 is active
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain>
ipfilter-config-file> /export/home/test/wifi.ipf.conf
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsecpolicy-config-file>
netcfg:loc:office> list
LOC:office
    activation-mode             conditional-any
    conditions                  "ncu ip:wpi0 is active"
    enabled                     false
    nameservices                dns
    nameservices-config-file    "/etc/nsswitch.dns"
    dns-nameservice-configsrc   dhcp
    ipfilter-config-file        "/export/home/test/wifi.ipf.conf"
netcfg:loc:office> verify
All properties verified
netcfg:loc:office> commit
Committed changes
netcfg> list
```

```
NCPs:
    User
    Automatic
Locations:
    Automatic
    NoNet
    test-loc
WLANs:
    sunwifi
    ibahn
    gogoinflight
    admiralsclub
    hhonors
    sjcfreewifi
netcfg> exit
Nothing to commit
$
```

이 예에서는 `office` 위치에 대해 다음 등록 정보가 지정되었습니다.

- `activation-mode` 등록 정보가 `conditional-any`로 설정되었습니다. 그 결과 활성화 조건을 지정할 수 있는 명령 프롬프트가 표시됩니다.
- 활성화 조건은 `ncu ip:wpi0 is active`로 지정되었습니다.

주 – `conditions` 등록 정보는 이전 단계에서 `conditional-any` 등록 정보가 지정되었기 때문에 필요했습니다. 예를 들어, `manual` 등록 정보가 지정된 경우에는 `conditions` 등록 정보가 필요하지 않습니다.

- Return 키를 눌러 다음 기본값이 수락되었습니다.
  - `nameservices`
  - `nameservices-config-file`
  - `dns-nameservice-configsrc`
  - `nfsv4-domain`
- `ipfilter-config-file` 등록 정보에 대해 `/export/home/test/wifi.ipf.conf` 파일이 지정되었습니다.
- Return 키를 눌러 다음 기본값이 수락되었습니다.
  - `ipfilter-v6-config-file`
  - `ipnat-config-file`
  - `ippool-config-file`
  - `ike-config-file`
  - `ipsecpolicy-config-file`
- `list` 하위 명령을 사용하여 위치 프로파일의 등록 정보를 표시했습니다.
- `verify` 하위 명령을 사용하여 구성 확인을 수행했습니다.
- `commit` 하위 명령을 사용하여 변경 사항을 지속 저장소에 커밋했습니다.

- list 하위 명령을 다시 사용하여 새 위치가 올바르게 생성되었으며 올바른 정보가 포함되었는지 확인했습니다.
- exit 하위 명령을 사용하여 netcfg 대화식 세션을 종료했습니다.

이러한 등록 정보에 설정할 수 있는 값에 대한 지침은 netcfg(1M) 매뉴얼 페이지를 참조하십시오.

ENM 프로파일 만들기

ENM은 NWAM 외부 응용 프로그램(예: VPN 응용 프로그램)의 구성과 관련이 있습니다. 이러한 응용 프로그램은 네트워크 구성을 만들고 수정할 수 있습니다. ENM은 활성화 또는 비활성화 시 네트워크 구성을 직접 수정하는 서비스나 응용 프로그램으로 정의될 수도 있습니다. 지정한 조건에서 ENM을 활성화 및 비활성화하도록 NWAM을 구성할 수 있습니다. 지정된 한 시점에 각 프로파일 유형 중 하나만 시스템에서 활성 상태일 수 있는 NCP 또는 위치 프로파일과 달리, ENM은 동시에 여러 개가 시스템에서 활성 상태일 수 있습니다. 지정된 한 시점에 시스템에서 활성 상태인 ENM이 동시에 시스템에서 활성 상태인 NCP 또는 위치 프로파일에 반드시 종속되는 것은 아닙니다.

주 – NWAM은 ENM을 만들 수 있는 응용 프로그램을 자동으로 인식하지 않습니다. netcfg 명령을 사용하여 ENM을 만들려면 먼저 이러한 응용 프로그램을 시스템에 설치하고 구성해야 합니다.

ENM을 만들려면 다음 명령을 입력합니다.

```
$ netcfg
netcfg> create enm my_enm
Created enm 'my_enm'.  Walking properties ...
```

여기서 enm은 ENM 프로파일이고 my_enm은 객체 이름입니다.

ENM을 만드는 동안 새로 만든 ENM의 프로파일 범위로 이동되며 새로 만든 ENM의 등록 정보가 자동으로 검토되기 시작합니다. 여기서 ENM 활성화 시기와 방식을 나타내는 ENM의 등록 정보와 ENM의 시작 및 중지 방법을 포함하는 기타 조건을 설정할 수 있습니다.

ENM 등록 정보 지정에 대한 지침은 netcfg(1M) 매뉴얼 페이지를 참조하십시오.

다음 표에서는 ENM을 만들거나 수정할 때 지정할 수 있는 등록 정보에 대해 설명합니다.

등록 정보 이름	설명	가능한 값
activation-mode	ENM의 활성화를 결정하는 데 사용되는 모드	conditional-any, conditional-all, manual

등록 정보 이름	설명	가능한 값
conditions	activation-mode가 conditional-any 또는 conditional-all인 경우 ENM의 활성화 여부를 결정하는 테스트를 지정합니다.	등록 정보가 사용되는 경우 netcfg(1M) 매뉴얼 페이지의 "코드 표현식" 절에 지정된 대로 형식이 지정된 문자열
start	(옵션) 활성화 시 실행할 스크립트의 절대 경로	이 등록 정보가 사용되는 경우 스크립트의 경로
stop	(옵션) 비활성화 시 실행할 스크립트의 절대 경로	이 등록 정보가 사용되는 경우 스크립트의 경로
fmri	(옵션) ENM 활성화 시 사용으로 설정할 FMRI(오류 관리 자원 식별자) **주** – FMRI 또는 시작 스크립트를 지정해야 합니다. FMRI를 지정한 경우 start 및 stop 등록 정보가 모두 무시됩니다.	스크립트 경로

예 4–4 대화식으로 ENM 프로파일 만들기

다음 예에서는 대화식 모드에서 test-enm이라는 ENM이 생성됩니다.

```
$ netcfg
netcfg> create enm test-enm
Created enm 'testenm'.  Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]>
fmri> svc:/application/test-app:default
start>
stop>
netcfg:enm:test-enm> list
ENM:test-enm
    activation-mode     manual
    enabled             false
    fmri                "svc:/application/test-enm:default"
netcfg:enm:test-enm> verify
All properties verified
netcfg:enm:test-enm> end
Committed changes
netcfg> list
NCPs:
    User
    Automatic
Locations:
    Automatic
    NoNet
    test-loc
ENMs:
    test-enm
WLANs:
    sunwifi
    ibahn
    gogoinflight
```

예 4–4 대화식으로 ENM 프로파일 만들기 *(계속)*

```
    admiralsclub
    hhonors
    sjcfreewifi
netcfg> end
$
```

이 예에서는 다음 등록 정보 값을 사용하여 test-enm이라는 ENM이 생성되었습니다.

- Return 키를 눌러 activation-mode 등록 정보의 기본값(manual)을 수락했습니다.
- SMF FMRI 등록 정보 svc:/application/test-enm:default가 응용 프로그램을 활성화 및 비활성화하는 데 사용할 방법으로 지정되었습니다.

  FMRI를 지정했으므로 start 및 stop 메소드 등록 정보가 무시되었습니다.
- list 하위 명령을 사용하여 ENM의 등록 정보를 표시했습니다.
- verify 하위 명령을 사용하여 프로파일 구성이 올바른지 확인했습니다.
- end 하위 명령을 사용하여 구성을 암시적으로 저장했습니다.
- end 하위 명령을 다시 사용하여 대화식 세션을 끝냈습니다.

WLAN 만들기

NWAM은 알려진 WLAN의 시스템 차원 목록을 유지 관리합니다. WLAN은 시스템에서 연결하는 무선 네트워크의 기록 및 구성 정보가 포함된 구성 객체입니다. 이 목록은 NWAM이 사용 가능한 무선 네트워크에 연결을 시도하는 순서를 결정하는 데 사용됩니다. 알려진 WLAN 목록에 있는 무선 네트워크를 사용할 수 있는 경우 NWAM이 자동으로 해당 네트워크에 연결합니다. 알려진 네트워크를 두 개 이상 사용할 수 있는 경우 NWAM은 우선 순위가 가장 높은(가장 낮은 번호) 무선 네트워크에 연결합니다. NWAM이 연결하는 새 무선 네트워크가 알려진 WLAN 목록의 맨 위에 추가되고 우선 순위가 가장 높은 새 무선 네트워크가 됩니다.

WLAN 객체를 만들려면 다음 명령을 입력합니다.

```
$ netcfg
netcfg> create wlan mywifi
Created wlan 'mywifi'.  Walking properties ...
```

여기서 wlan은 WLAN 객체이고 mywifi는 객체 이름입니다.

WLAN 객체를 만드는 동안 새로 만든 WLAN의 프로파일 범위로 이동되며 새로 만든 WLAN의 등록 정보가 자동으로 검토되기 시작합니다. 여기서 구성을 정의하는 WLAN의 등록 정보를 설정할 수 있습니다.

다음 표에서는 WLAN을 만들거나 수정할 때 지정할 수 있는 등록 정보에 대해 설명합니다.

알려진 WLAN 등록 정보	등록 정보의 데이터 유형
name	ESSID(무선 네트워크 이름)
bssids	지정한 WLAN에 연결되어 있는 동안 시스템이 연결한 WLAN의 기본 스테이션 ID
priority	WLAN 연결 기본 설정(낮은 값이 선호됨)
keyslot	WEP 키가 포함된 슬롯 번호(1–4)
keyname	dladm create-secobj 명령을 사용하여 만든 WLAN 키의 이름
security-mode	사용 중인 암호화 키의 유형. 유형은 none, wep 또는 wpa여야 합니다.

예 4–5 WLAN 만들기

다음 예에서는 mywifi라는 WLAN 객체가 생성됩니다.

이 예에서는 WLAN을 추가하기 **전에** WLAN mywifi의 keyname 등록 정보로 지정된 키가 포함된 mywifi-key라는 보안 객체가 생성되었다고 가정합니다.

다른 WLAN을 추가하거나 제거하면 우선 순위 번호가 변경될 수 있습니다. 두 WLAN에 동일한 우선 순위 번호를 할당할 수 없습니다. 선호되는 WLAN에서는 숫자가 작을수록 더 높은 우선 순위를 나타냅니다. 이 예에서는 WLAN에 우선 순위 번호 100을 할당하여 알려진 다른 WLAN보다 낮은 우선 순위를 갖도록 합니다.

절차의 끝에서 list 하위 명령을 사용하면 새 WLAN이 목록의 맨 아래에 추가되어 기존의 알려진 모든 WLAN 중에서 가장 낮은 우선 순위가 할당됨을 나타냅니다. WLAN에 우선 순위 번호 0(기본값)이 할당된 경우 목록의 맨 위에 표시되어 가장 높은 우선 순위를 나타냅니다. 이후에는 기존의 다른 모든 WLAN의 우선 순위가 아래로 이동하고 목록에서 새로 추가된 WLAN 뒤에 표시됩니다.

```
$ netcfg
netcfg> create wlan mywifi
Created wlan 'mywifi'.  Walking properties ...
priority (0)> 100
bssids>
keyname> mywifi-key
keyslot>
security-mode [none|wep|wpa]> wpa
netcfg:wlan:mywifi> list
WLAN:mywifi
    priority            100
    keyname             "mywifi-key"
    security-mode       wpa
netcfg:wlan:mywifi> verify
All properties verified
netcfg:wlan:mywifi> end
Committed changes
netcfg> list
```

예 4–5 WLAN 만들기 (계속)

```
NCPs:
    User
    Automatic
Locations:
    Automatic
    NoNet
    test-loc
ENMs:
    test-enm
WLANs:
    sunwifi
    ibahn
    gogoinflight
    admiralsclub
    hhonors
    sjcfreewifi
    mywifi
netcfg> exit
Nothing to commit
$
```

프로파일 제거

netcfg destroy -a 명령을 사용하여 메모리 및 지속 저장소에서 모든 사용자 정의 프로파일이나 지정된 사용자 정의 프로파일을 제거할 수 있습니다.

주 – 자동 NCP 프로파일과 NoNet 및 자동 위치 프로파일이 포함된 시스템 정의 프로파일은 제거할 수 없습니다.

---

destroy 명령의 구문은 다음과 같습니다.

netcfg destroy *object-type* **[** *class* **]** *object-name*

또는 다음 명령을 사용하여 시스템에서 사용자 정의 프로파일을 모두 제거할 수 있습니다.

netcfg destroy -a

예 4–6 netcfg 명령줄 모드를 사용하여 사용자 정의 프로파일 모두 제거

시스템에서 사용자 정의 프로파일을 모두 제거하려면 다음 명령을 입력합니다.

```
$ netcfg destroy -a
```

항상 시스템에서는 하나 이상의 프로파일이 활성 상태여야 하므로 사용자 정의 프로파일을 제거할 때 사용 중 오류 발생을 방지하려면 destroy -a 명령을 사용하기 전에 자동 NCP를 사용으로 설정해야 합니다.

예 4–7 netcfg 명령줄 모드를 사용하여 특정 사용자 정의 프로파일 제거

시스템에서 특정 사용자 정의 프로파일(예: `User`라는 NCP)을 제거하려면 다음 명령을 입력합니다.

```
$ netcfg destroy ncp User
```

`destroy` 명령을 사용하여 기존 NCP에서 NCU를 제거할 수도 있습니다. 다음 예에서는 이름이 `net1`인 인터페이스 NCU가 사용자 정의 NCP에서 제거됩니다.

```
$ netcfg "select ncp User; destroy ncu ip net1"
```

프로파일이 제거되었는지 확인하려면 다음과 같이 `list` 하위 명령을 사용합니다.

```
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> list
NCUs:
        phys    net1
netcfg> exit
Nothing to commit
$
```

예 4–8 대화식으로 프로파일 제거

다음 예에서는 `net2`라는 IP NCU가 제거됩니다.

```
$ netcfg list
NCPs:
        Automatic
        User
Locations:
        Automatic
        NoNet
        test
        foo
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> list
NCUs:
        phys    net2
        ip      net2
netcfg:ncp:User> destroy ncu ip net2
Destroyed ncu 'net2'
netcfg:ncp:User> list
NCUs:
        phys    net2
netcfg:ncp:User> end
netcfg> exit
Nothing to commit
$
```

프로파일의 등록 정보 값 설정 및 변경

새 사용자 정의 프로파일과 기존 사용자 정의 프로파일의 등록 정보 값은 netcfg 명령에 set 하위 명령을 사용하여 설정됩니다. 이 하위 명령은 대화식 모드 또는 명령줄 모드에서 사용할 수 있습니다. 명령줄 모드에서 등록 정보 값을 설정하거나 변경하면 변경 사항이 지속 저장소에 즉시 커밋됩니다.

set 하위 명령의 구문은 다음과 같습니다.

netcfg set *prop-name*=*value1*[,*value2*...]

특정 등록 정보 값을 검색해야 하는 경우 netcfg get 명령을 사용합니다. 자세한 내용은 92 페이지 "특정 등록 정보 값 가져오기"를 참조하십시오.

예 4–9 netcfg 명령줄 모드에서 등록 정보 값 설정

명령줄 모드에서 netcfg 명령을 사용하여 등록 정보 값을 설정하는 경우 명령줄에서 하위 명령을 여러 개 입력해야 합니다.

예를 들어, net1이라는 링크 NCU의 mtu 등록 정보를 설정하려면 다음 명령을 입력합니다.

```
$ netcfg "select ncp User; select ncu phys net1; set mtu=1492"
```

이 예에서는 select 하위 명령을 사용하여 최상위 프로파일을 선택한 다음 수정된 mtu 등록 정보 값이 포함된 NCU를 다시 선택합니다.

명령줄에서 지정된 등록 정보에 대해 동시에 여러 값을 설정할 수 있습니다. 여러 값을 설정하는 경우 각 값을 쉼표(,)로 구분해야 합니다. 지정한 등록 정보의 개별 값에도 쉼표가 포함되어 있는 경우 등록 정보 값의 일부인 쉼표 앞에 백슬래시를 추가해야 합니다(\,). 단일 값만 포함된 등록 정보 내의 쉼표는 분리자로 해석되지 않으므로 앞에 백슬래시를 추가할 필요가 없습니다.

다음 예에서는 NCP User의 NCU myncu에 대한 ip-version 등록 정보 값이 설정됩니다.

```
$ netcfg "select ncp User; select ncu ip myncu; set ip-version=ipv4,ipv6"
```

예 4–10 대화식으로 프로파일의 등록 정보 값 설정

대화식으로 등록 정보 값을 설정하는 경우 먼저 현재 범위에서 프로파일을 선택해야 합니다. 이 경우 대화식 세션이 해당 프로파일의 범위로 이동됩니다. 이 범위에서 등록 정보를 수정하려는 객체를 선택할 수 있습니다. 선택한 프로파일이 지속 저장소에서 메모리로 로드됩니다. 다음 예와 같이 이 범위에서 프로파일 또는 해당 등록 정보를 수정할 수 있습니다.

```
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> select ncu ip iwk0
netcfg:ncp:User:ncu:iwk0> set ipv4-default-route = 129.174.7.366
```

예 4–10 대화식으로 프로파일의 등록 정보 값 설정 *(계속)*

다음 예에서는 foo 위치의 ipfilter-config-file 등록 정보가 설정됩니다.

```
$ netcfg
netcfg> list
NCPs:
   Automatic
   User
Locations:
   Automatic
   NoNet
   foo

netcfg> select loc foo
netcfg:loc:foo> list
LOC:foo
   activation-mode              manual
   enabled                      false
   nameservices                 dns
   dns-nameservice-configsrc    dhcp
   nameservices-config-file     "/etc/nsswitch.dns"
netcfg:loc:foo> set ipfilter-config-file=/path/to/ipf-file
netcfg:loc:foo> list
LOC:foo
   activation-mode              manual
   enabled                      false
   nameservices                 dns
   dns-nameservice-configsrc    dhcp
   nameservices-config-file     "/etc/nsswitch.dns"
   ipfilter-config-file         "/path/to/ipf-file"
netcfg:loc:foo> end
Committed changes
netcfg> exit
Nothing to commit
$
```

다음 예에서는 NCP User의 NCU net0에 대한 link-mtu 등록 정보가 대화식으로 수정됩니다.

```
$ netcfg
netcfg> select ncp User
netcfg:ncp:User> select ncu phys net0
netcfg:ncp:User:ncu:net0> list
NCU:net0
   type                 link
   class                phys
   parent               "User"
   enabled              true
   activation-mode      prioritized
   priority-mode        exclusive
   priority-group       1
netcfg:ncp:User:ncu:net0> set link-mtu=5000
netcfg:ncp:User:ncu:net0> list
NCU:net0
   type                 link
```

예 4–10 대화식으로 프로파일의 등록 정보 값 설정 (계속)

```
   class               phys
   parent              "User"
   enabled             true
   activation-mode     prioritized
   priority-mode       exclusive
   priority-group      1
   link-mtu            5000
netcfg:ncp:User:ncu:net0> commit
Committed changes
netcfg:ncp:User:ncu:net0> exit
Nothing to commit
$
```

시스템에 프로파일 정보 질의

netcfg 명령에 list 하위 명령을 사용하여 현재 범위나 지정한 범위에 있는 프로파일, 등록 정보-값 쌍 및 리소스를 나열할 수 있습니다. list 하위 명령을 사용하여 시스템에 모든 프로파일에 대한 일반 정보를 질의하거나 특정 프로파일에 대한 특정 정보를 검색할 수 있습니다. list 하위 명령은 대화식 모드 또는 명령줄 모드에서 사용할 수 있습니다.

프로파일 정보와 현재 상태를 가져와야 하는 경우 netadm 명령에 list 하위 명령을 사용합니다. 자세한 내용은 102 페이지 “프로파일의 현재 상태 표시”를 참조하십시오.

시스템의 모든 프로파일 나열

netcfg list 명령은 시스템의 시스템 정의 프로파일과 사용자 정의 프로파일을 모두 나열합니다. 옵션 없이 list 하위 명령을 사용하면 시스템에 있는 최상위 프로파일이 모두 표시됩니다. 이 명령은 각 프로파일의 상태를 나열하지 않습니다. 프로파일 및 해당 상태(온라인 또는 오프라인) 목록을 표시하려면 netadm list 명령을 사용합니다.

시스템의 최상위 프로파일을 모두 나열하려면 다음 명령을 입력합니다.

```
$ netcfg list
NCPs:
        Automatic
        User
Locations:
        Automatic
        NoNet
        home
        office
ENMs:
        myvpn
        testenm
WLANs:
```

```
        workwifi
        coffeeshop
        homewifi
```

이 예에서는 다음 프로파일이 나열됩니다.

- NCP

  NCP 두 개가 나열됩니다. 하나는 시스템 정의 프로파일인 자동 NCP이고 다른 하나는 `User`라는 사용자 정의 NCP입니다.

- 위치

  위치 프로파일 4개가 나열됩니다. 두 위치는 시스템 정의(`Automatic` 및 `NoNet`)이고, 다른 두 위치는 사용자 정의(`home` 및 `office`)입니다.

- ENM

  ENM 두 개가 나열됩니다. 한 ENM은 설치 및 구성된 VPN 응용 프로그램에 사용되고 다른 ENM은 테스트 ENM입니다.

- WLAN

  WLAN 세 개가 나열됩니다. 첫번째 WLAN은 직장에 사용되고, 두번째 WLAN은 현지 커피숍에 사용되고, 세번째 WLAN은 사용자의 홈 무선 네트워크에 사용됩니다.

주 – 사용자 정의 프로파일만 만들고, 수정 또는 제거할 수 있습니다.

특정 프로파일의 모든 등록 정보 값 나열

`netcfg` 명령에 `list` 하위 명령을 사용하여 지정한 프로파일의 등록 정보 값을 모두 나열할 수 있습니다.

`list` 하위 명령의 구문은 다음과 같습니다.

```
$ netcfg list [ object-type [ class ] object-name ]
```

예 4–11 NCU의 모든 등록 정보 값 나열

예를 들어, `User` NCP의 IP NCU에 대한 등록 정보 값을 모두 나열하려면 다음 명령을 입력합니다.

```
$ netcfg "select ncp User; list ncu ip net0"
NCU:net0
        type                    interface
        class                   ip
        parent                  "User"
        enabled                 true
        ip-version              ipv4
        ipv4-addrsrc            dhcp
        ipv6-addrsrc            dhcp,autoconf
```

예 4–12 ENM의 모든 등록 정보 값 나열

다음 예에서는 myenm이라는 ENM의 모든 등록 정보가 나열됩니다.

```
$ list enm myenm
ENM:myenm
activation-mode manual
enabled         true
start           "/usr/local/bin/myenm start"
stop            "/bin/alt_stop"
```

이 예에서 list 하위 명령의 출력 결과에는 다음 정보가 표시됩니다.

- 이 ENM의 activation-mode 등록 정보는 manual로 설정됩니다.
- ENM이 사용으로 설정됩니다.
- FMRI를 사용하는 대신 start 및 stop 메소드 등록 정보가 지정되었습니다.

특정 등록 정보 값 가져오기

netcfg 명령에 get 하위 명령을 사용하여 지정한 등록 정보의 특정 값을 가져올 수 있습니다. 이 하위 명령은 대화식 모드 또는 명령줄 모드에서 사용할 수 있습니다.

get 하위 명령의 구문은 다음과 같습니다.

netcfg get [-V] *prop-name*

User NCP에 속하는 myncu라는 NCU의 ip-version 등록 정보 값을 가져오려면 다음 명령을 입력합니다. 예를 들면 다음과 같습니다.

```
$ netcfg "select ncp User; select ncu ip myncu; get -V ip-version"
ipv4
```

get 하위 명령에 -V 옵션을 사용하면 다음과 같이 등록 정보 값만 표시됩니다.

```
netcfg:ncp:User:ncu:net0> get -V activation-mode
manual
```

그렇지 않으면 등록 정보와 해당 값이 모두 표시됩니다. 예를 들면 다음과 같습니다.

```
netcfg:ncp:User:ncu:net0> get activation-mode
activation-mode         manual
```

▼ 대화식으로 단일 등록 정보 값을 가져오는 방법

이 절차에서는 netcfg 대화식 모드에 있는 동안 netcfg get 명령을 사용하여 단일 등록 정보 값을 가져오는 방법에 대해 설명합니다. 이 특정 절차의 일부 예에서는 User NCP의 NCU에 대한 단일 등록 정보 값을 가져오는 방법을 보여줍니다. 해당 예는 데모용으로**만** 사용됩니다. 이 명령을 사용할 때 제공하는 정보는 검색하려는 프로파일 및 등록 정보 값에 따라 달라집니다.

프로파일의 등록 정보 값을 모두 보려는 경우 walkprop 하위 명령을 교대로 사용할 수 있습니다. 이 하위 명령은 지정된 프로파일의 모든 등록 정보를 한 번에 하나씩 검토하여 프로파일 등록 정보 중 하나 또는 모두를 수정할 수 있게 합니다. 자세한 내용은 94 페이지 "walkprop 하위 명령을 사용하여 대화식으로 등록 정보 값 확인 및 변경"을 참조하십시오.

1 netcfg 대화식 세션을 시작합니다.

```
$ netcfg
netcfg>
```

2 가져오려는 등록 정보 값이 포함된 프로파일 또는 구성 객체를 선택합니다.

```
netcfg> select object-type [ class ] object-name
```

주 – *class* 매개변수는 NCU를 선택하는 경우에**만** 해당됩니다. 또한 phys 및 ip 클래스 NCU가 동일한 이름을 공유하는 경우 class 매개변수를 지정해야 합니다. 하지만 NCU 이름이 고유한 경우에는 class 매개변수가 필요 없습니다.

예를 들어, User NCP를 선택하려면 다음을 입력합니다.

```
netcfg> select User NCP
```

이 예에서 User NCP를 선택하면 대화식 세션이 선택한 객체의 범위로 이동합니다.

3 (옵션) 프로파일의 구성 요소를 표시합니다.

```
netcfg:ncp:User> list
NCUs:
        phys    net0
        ip      net0
```

4 가져오려는 등록 정보 값이 포함된 객체를 선택합니다.

다음 예에서는 User NCP의 링크(phys) NCU net0이 선택됩니다.

```
netcfg:ncp:User> select ncu phys net0
```

NCU net0을 선택하면 대화식 세션이 해당 객체의 범위로 이동하고 메모리에서 NCU의 현재 등록 정보가 로드됩니다.

5 지정한 등록 정보 값을 가져옵니다.

```
netcfg:ncp:User:ncu:net0> get property-value
```

예를 들어, activation-mode 등록 정보의 값을 가져오려면 다음을 입력합니다.

```
netcfg:ncp:User:ncu:net0> get activation-mode
        activation-mode         manual
```

다음 순서 이때 set 하위 명령을 사용하여 등록 정보에 새 값을 설정하거나, 변경하지 않고 대화식 세션을 종료할 수 있습니다. 대화식 모드에 있는 동안 등록 정보 값을 수정하는 경우

commit 또는 exit 하위 명령을 사용하여 변경 사항을 저장해야 합니다. netcfg 대화식 모드에서 등록 정보 값 설정에 대한 자세한 내용은 88 페이지 "프로파일의 등록 정보 값 설정 및 변경"을 참조하십시오.

walkprop 하위 명령을 사용하여 대화식으로 등록 정보 값 확인 및 변경

대화식으로 walkprop 하위 명령을 사용하여 프로파일의 등록 정보를 확인할 수 있습니다. 이 하위 명령은 한 번에 한 등록 정보의 프로파일을 "검토"하여 각 등록 정보의 이름과 현재 값을 표시합니다. 지정한 등록 정보의 현재 값을 변경하는 데 사용할 수 있는 대화식 명령 프롬프트도 표시됩니다. 다중 값 등록 정보의 분리자는 쉼표(,)입니다. 지정된 등록 정보의 개별 값에 쉼표가 포함된 경우 앞에 백슬래시(\)를 추가해야 합니다. 단일 값만 포함된 등록 정보 내의 쉼표는 분리자로 해석되지 않으므로 앞에 백슬래시를 추가할 필요가 없습니다.

주 – walkprop 하위 명령은 대화식 모드에서 사용되는 경우에만 의미가 있습니다.

예 4–13 특정 프로파일의 등록 정보 값 확인 및 변경

다음 예에서는 walkprop 하위 명령을 사용하여 위치 foo의 activation-mode 등록 정보를 확인하고 변경합니다. walkprop 하위 명령을 사용할 때는 set 하위 명령을 사용하여 등록 정보 값을 설정할 필요가 없습니다.

```
$ netcfg
netcfg> select loc foo
netcfg:loc:foo> list
loc:foo
        activation-mode                 manual
        enabled                         false
        nameservices                    dns
        nameservices-config-file        "/etc/nsswitch.dns"
        dns-nameservice-configsrc       dhcp
        nfsv4-domain                    "Central.oracle.com"
netcfg:loc:foo> walkprop
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
conditions> advertised-domain is oracle.com
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain ("Central.oracle.com")>
ipfilter-config-file>
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsecpolicy-config-file>
netcfg:loc:foo> list
loc:foo
        activation-mode                 conditional-all
```

예 4–13 특정 프로파일의 등록 정보 값 확인 및 변경 *(**계속**)*

```
        conditions                      "advertised-domain is oracle.com"
        enabled                         false
        nameservices                    dns
        nameservices-config-file        "/etc/nsswitch.dns"
        dns-nameservice-configsrc       dhcp
        nfsv4-domain                    "Central.oracle.com"
netcfg:loc:foo> commit
Committed changes
netcfg:loc:foo> end
netcfg> exit
$
```

주 – 관련 등록 정보만 검토됩니다. 예를 들어, `ipv4-addrsrc` 등록 정보를 `static`으로 설정하면 `ipv4-addr` 등록 정보가 검토에 포함됩니다. 하지만 `ipv4-addrsrc`를 `dhcp`으로 설정하면 `ipv4-addr` 등록 정보가 검토되지 않습니다.

프로파일 구성 내보내기 및 복원

`export` 하위 명령을 사용하여 프로파일 구성을 저장하고 복원할 수 있습니다. 프로파일 내보내기는 동일한 네트워크 구성이 필요한 여러 서버를 유지 관리해야 하는 시스템 관리자에게 유용할 수 있습니다. `export` 하위 명령은 대화식 모드 또는 명령줄 모드에서 사용할 수 있습니다. 또는 명령 파일 모드에서 이 명령을 사용하여 파일을 명령 출력으로 지정할 수 있습니다.

`export` 하위 명령의 명령 구문은 다음과 같습니다.

```
$ netcfg export [ -d ] [ -f output-file ] [ object-type  [ class ] object-name ]
```

주 – `export` 하위 명령의 `-d` 및 `-f` 옵션은 서로 독립적으로 사용할 수 있습니다.

예 4–14 프로파일 구성 내보내기

다음 예에서는 `export` 하위 명령을 사용하여 시스템의 프로파일 구성을 화면에 표시합니다.

```
$ netcfg
netcfg> export
create ncp "User"
create ncu ip "net2"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net2"
```

예 4–14 프로파일 구성 내보내기 *(계속)*

```
set activation-mode=manual
set link-mtu=5000
end
create ncu phys "wpi2"
set activation-mode=prioritized
set priority-group=1
set priority-mode=exclusive
set link-mac-addr="13:10:73:4e:2"
set link-mtu=1500
end
end
create loc "test"
set activation-mode=manual
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
set nfsv4-domain="domainl.oracle.com"
end
create loc "foo"
set activation-mode=conditional-all
set conditions="system-domain is oracle.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
set nfsv4-domain="domain.oracle.com"
end
create enm "myenm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 1.2.3.4"
set start="/my/start/script"
set stop="/my/stop/script"
end
create wlan "mywlan"
set priority=0
set bssids="0:13:10:73:4e:2"
end
netcfg> end
$
```

예 4–15 netcfg 대화식 모드에서 프로파일 구성 내보내기

다음 예에서는 `export` 하위 명령에 `-d` 옵션을 사용합니다. `-d` 옵션은 `netcfg export` 출력의 첫째 줄로 `destroy -a` 명령을 추가합니다.

```
$ netcfg
netcfg> export -d
destroy -a
create ncp "User"
create ncu ip "net2"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net2"
set activation-mode=manual
```

예 4–15 netcfg 대화식 모드에서 프로파일 구성 내보내기 *(계속)*

```
set link-mtu=5000
end
create ncu phys "wpi2"
set activation-mode=prioritized
set priority-group=1
set priority-mode=exclusive
set link-mac-addr="13:10:73:4e:2"
set link-mtu=1500
end
end
create loc "test"
set activation-mode=manual
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
set nfsv4-domain="domain.oracle.com"
end
create loc "foo"
set activation-mode=conditional-all
set conditions="system-domain is oracle.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
set nfsv4-domain="domain.oracle.com"
end
create enm "myenm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 1.2.3.4"
set start="/my/start/script"
set stop="/my/stop/script"
end
create wlan "mywlan"
set priority=0
set bssids="0:13:10:73:4e:2"
end
netcfg> end
$
```

예 4–16 netcfg 명령 파일 모드에서 프로파일 구성 내보내기

다음 예에서는 `netcfg export` 명령에 `-f` 옵션을 사용하여 `User` NCP의 구성 정보를 파일에 씁니다. `-f` 옵션은 `user2`라는 새 파일에 출력을 씁니다. `-d` 옵션은 `netcfg export` 출력의 첫째 줄로 `destroy -a` 명령을 추가합니다.

```
$ netcfg export -d -f user2 ncp User

$ ls -al
drwx------   3 root     root           4 Oct 14 10:53 .
drwxr-xr-x  37 root     root          40 Oct 14 10:06 ..

-rw-r--r--   1 root     root         352 Oct 14 10:53 user2
$

$ cat user2
destroy -a
```

예 4–16 netcfg 명령 파일 모드에서 프로파일 구성 내보내기 *(계속)*

```
create ncp "User"
create ncu ip "net2"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net2"
set activation-mode=manual
set link-mtu=5000
end
create ncu phys "wpi2"
set activation-mode=prioritized
set priority-group=1
set priority-mode=exclusive
set link-mac-addr="13:10:73:4e:2"
set link-mtu=1500
end
end
create loc "test"
set activation-mode=manual
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
set nfsv4-domain="domain.oracle.com"
end
create loc "foo"
set activation-mode=conditional-all
set conditions="system-domain is oracle.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
set nfsv4-domain="domain.oracle.com"
end
create enm "myenm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 1.2.3.4"
set start="/my/start/script"
set stop="/my/stop/script"
end
create wlan "mywlan"
set priority=0
set bssids="0:13:10:73:4e:2"
end
$
```

사용자 정의 프로파일 복원

다음과 같이 `netcfg` 명령에 `-f` 옵션을 사용하여 사용자 정의 프로파일을 복원할 수 있습니다.

```
$ netcfg [ -f ] profile-name
```

예를 들면 다음과 같습니다.

```
$ netcfg -f user2
```

이 명령은 내보낸 구성이 포함된 명령 파일을 실행합니다.

네트워크 구성 관리

네트워크 구성 관리는 프로파일 기반 기능으로, 수동 및 자동의 두 네트워크 구성 모드를 전환하여 관리할 수 있습니다. 모드 사이를 전환하려면 적절한 NCP를 사용으로 설정합니다. 수동 네트워크 구성의 경우 `DefaultFixed` NCP를 사용으로 설정합니다. 자동(NWAM) 네트워크 구성의 경우 `Automatic` 또는 사용자 정의 NCP를 사용으로 설정합니다.

▼ 자동 네트워크 구성 모드에서 수동 네트워크 구성 모드로 전환하는 방법

현재 NWAM 구성 관리에서 지원되지 않는 고급 네트워킹 기능을 사용하는 경우 또는 수동 네트워크 구성 관리를 선호하는 경우 다음 절차와 같이 `DefaultFixed` NCP를 사용으로 설정할 수 있습니다.

1 root 사용자로 전환합니다.

2 DefaultFixed NCP를 사용으로 설정합니다.

```
# netadm enable -p ncp DefaultFixed
```

3 network/physical:default 서비스가 다시 시작되고 온라인 상태인지 확인합니다.

```
# svcs -xv network/physical:default
svc:/network/physical:default (physical network interface configuration)
 State: online since Fri Aug 26 16:19:18 2011
   See: man -M /usr/share/man -s 1M ipadm
   See: man -M /usr/share/man -s 5 nwam
   See: /var/svc/log/network-physical:default.log
Impact: None.
#
```

4 DefaultFixed NCP가 활성 상태인지 확인합니다.

```
# netadm list
netadm: DefaultFixed NCP is enabled;
automatic network management is not available.
'netadm list' is only supported when automatic network management is active.
```

주 – netadm 명령은 네트워크 구성이 자동 모드일 때만 지원됩니다. 따라서 수동 모드에서는 DefaultFixed 프로파일이 사용으로 설정됨을 나타내는 것으로만 명령 출력이 제한됩니다. 시스템의 다른 NCP에 대한 정보는 제공되지 않습니다.

▼ 수동 네트워크 구성 모드에서 자동 네트워크 구성 모드로 전환하는 방법

수동 네트워크 구성 모드에서 자동 네트워크 구성 모드로 다시 전환하려면 사용할 네트워크 구성 프로파일을 사용으로 설정합니다.

1 root 사용자로 전환합니다.

2 Automatic과 같은 NCP를 사용으로 설정합니다.

```
# netadm enable -p ncp Automatic
```

3 network/physical:default 서비스가 다시 시작되고 온라인 상태인지 확인합니다.

```
# svcs -xv network/physical:default
svc:/network/physical:default (physical network interface configuration)
 State: online since Fri Aug 26 16:19:18 2011
   See: man -M /usr/share/man -s 1M ipadm
   See: man -M /usr/share/man -s 5 nwam
   See: /var/svc/log/network-physical:default.log
Impact: None.
#
```

4 NCP 및 기타 NWAM 프로파일의 상태를 확인합니다.

```
# netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         Automatic      online         active
ncu:phys    net0           online         interface/link is up
ncu:ip      net0           online         interface/link is up
ncu:phys    net1           offline        interface/link is down
ncu:ip      net1           offline        conditions for activation are unmet
ncp         User           disabled       disabled by administrator
loc         Automatic      online         active
loc         NoNet          offline        conditions for activation are unmet
#
```

◆ ◆ ◆ 5 장

5 NWAM 프로파일 관리(작업)

이 장에서는 netadm 명령을 사용하여 NCP, 위치, ENM 및 WLAN 프로파일을 관리하는 방법에 대해 설명합니다. netadm 명령을 사용하여 NCU를 관리할 수도 있습니다. NCU는 NCP를 구성하고 NWAM GUI가 없을 경우 NWAM 데몬(nwamd)과 상호 작용하는 개별 구성 객체입니다. netadm 명령 사용에 대한 자세한 내용은 netadm(1M) 매뉴얼 페이지를 참조하십시오.

이 장에서는 다음 항목을 다룹니다.

프로파일을 만들고 netcfg 명령을 사용하여 등록 정보를 구성하는 방법에 대한 자세한 내용은 4 장, "NWAM 프로파일 구성(작업)"을 참조하십시오.

NWAM 구성과 상호 작용하는 방법 및 NWAM GUI를 사용하여 데스크탑에서 네트워크 구성을 관리하는 방법에 대한 자세한 내용은 6 장, "NWAM 그래픽 사용자 인터페이스 정보"를 참조하십시오.

NWAM에 대한 소개는 2 장, "NWAM 소개"를 참조하십시오.

모든 NWAM 구성 요소 및 NWAM 구성 세부 정보에 대한 자세한 내용은 3 장, "NWAM 구성 및 관리(개요)"를 참조하십시오.

프로파일 상태에 대한 정보 가져오기

netadm 명령에 list 하위 명령을 사용하여 시스템에서 사용 가능한 모든 프로파일과 현재 상태를 표시하거나 특정 프로파일과 해당 상태를 표시할 수 있습니다.

list 하위 명령의 구문은 다음과 같습니다.

netadm list [**-p** *profile-type*] [**-c** *ncu-class*] [*profile-name*]

예를 들어, 시스템의 모든 프로파일과 해당 상태를 표시하려면 다음 명령을 입력합니다.

```
$ netadm list
TYPE         PROFILE        STATE
ncp          User           disabled
ncp          Automatic      online
ncu:ip       net1           offline
ncu:phys     net1           offline
ncu:ip       net0           online
ncu:phys     net0           online
loc          foo            disabled
loc          test           disabled
loc          NoNet          offline
loc          Automatic      online
$
```

이 예에서는 시스템의 모든 시스템 정의 프로파일 및 사용자 정의 프로파일과 현재 상태가 표시됩니다. list 하위 명령은 사용으로 설정된 NCP 및 해당 특정 NCP를 구성하는 모든 NCU를 표시합니다.

프로파일의 현재 상태 표시

명령 구문에 프로파일 유형과 NCU 클래스를 포함하여 특정 프로파일을 식별할 수 있습니다. 프로파일 유형만 제공하면 해당 유형의 모든 프로파일이 표시됩니다. 프로파일을 이름으로 지정하면 해당 프로파일의 현재 상태가 표시됩니다. 프로파일 이름이 고유하지 않은 경우 해당 이름을 가진 모든 프로파일이 나열됩니다.

각 프로파일에 가능한 상태 값은 다음과 같습니다.

disabled — 사용으로 설정되지 않은, 수동으로 활성화된 프로파일을 나타냅니다.

offline — 활성화되지 않은 조건부 활성화 프로파일 또는 시스템 활성화 프로파일을 나타냅니다. 조건이 충족되지 않았거나 더 구체적인 조건을 가진, 충족된 다른 프로파일이 활성 상태이므로 프로파일이 활성 상태가 아닐 수 있습니다.

주 – offline 상태는 위치 프로파일과 같이 한 번에 하나씩 활성화되어야 하는 프로파일 유형에서 더 자주 발생합니다.

---

online	충족된 조건이 있으며 성공적으로 활성화된 조건부 활성화 프로파일 또는 시스템 활성화 프로파일을 나타냅니다. 또는 사용자 요청 시 성공적으로 사용으로 설정된, 수동으로 활성화된 프로파일입니다.
maintenance	프로파일 활성화를 시도했지만 활성화에 실패했음을 나타냅니다.
initialized	프로파일이 유효하지만 프로파일에 대해 아무 작업도 수행되지 않았음을 나타냅니다.
uninitialized	프로파일이 시스템에 없음을 나타냅니다. 예를 들어, 이 상태는 물리적 링크에 해당하는 NCU가 시스템에서 제거될 때 발생할 수 있습니다.

예 5–1 지정한 프로파일의 현재 상태 표시

다음 예에서는 이름으로 지정된 자동 NCP의 현재 상태를 나열합니다.

```
$ netadm list Automatic
TYPE         PROFILE        STATE
ncp          Automatic      online
ncu:ip       net1           offline
ncu:phys     net1           offline
ncu:ip       net0           online
ncu:phys     net0           online
loc          Automatic      online
```

다음 예에서는 `list` 하위 명령에 `-p` 옵션을 사용하여 현재 시스템에 있는 모든 위치를 표시합니다.

```
$ netadm list -p loc
TYPE         PROFILE        STATE
loc          foo            disabled
loc          test           disabled
loc          NoNet          offline
loc          Automatic      online
$
```

다음 예에서는 `list` 하위 명령에 `-c` 옵션을 사용하여 현재 활성 NCP의 인터페이스 NCU를 모두 표시합니다.

```
$ netadm list -c ip
TYPE         PROFILE         STATE
ncu:ip       net0            online
ncu:ip       net1            disabled
$
```

보조 상태 값

프로파일의 보조 상태는 지정된 프로파일이 online 또는 offline(사용 또는 사용 안함)인 이유에 대해 설명합니다. 보조 상태 값을 나열하려면 다음 예와 같이 `list` 하위 명령에 `-x` 옵션을 사용합니다.

```
$ netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         Automatic      disabled       disabled by administrator
ncp         User           online         active
ncu:phys    nge0           online         interface/link is up
ncu:ip      nge0           online         interface/link is up
ncu:phys    nge1           offline        interface/link is down
ncu:ip      nge1           offline        conditions for activation are unmet
loc         Automatic      offline        conditions for activation are unmet
loc         NoNet          offline        conditions for activation are unmet
loc         office         online         active
```

프로파일 유형에 따라 보조 상태 값이 달라집니다. 보조 상태에 대한 자세한 내용은 `nwamd`(1M) 매뉴얼 페이지를 참조하십시오.

프로파일 활성화 및 비활성화

사용자 정의 NCP, 위치 프로파일 및 ENM에는 모두 `activation-mode` 등록 정보가 있습니다. 각 프로파일에 허용되는 값은 해당 유형에 의해 결정됩니다.

프로파일 또는 구성 객체를 수동으로 사용 또는 사용 안함으로 설정(활성화 또는 비활성화)하려면 `netadm enable` 명령이나 `netadm disable` 명령을 사용합니다. 지정한 프로파일의 `activation-mode` 등록 정보를 `manual`로 설정한 경우 시스템 정의 프로파일과 사용자 정의 프로파일을 모두 사용 및 사용 안함으로 설정할 수 있습니다. `activation-mode` 등록 정보는 프로파일을 만들거나 수정할 때 `netcfg` 명령을 사용하여 설정됩니다. 자세한 내용은 51 페이지 “NWAM 프로파일 활성화 방식”을 참조하십시오.

항상 시스템에는 활성 NCP 한 개와 활성 위치 프로파일 한 개가 있어야 합니다. `activation-mode`가 `manual`인 다른 NCP 또는 위치를 사용으로 설정하면 현재 활성 NCP 또는 위치 프로파일이 암시적으로 비활성화됩니다. `activation-mode` 등록 정보를 `manual`로 설정한 경우 현재 위치를 비활성화할 수도 있습니다. 다른 위치를 사용할 수 없는 경우 NWAM은 시스템 정의 위치 중 하나(IP 구성에 성공한 경우 자동 위치 또는 NoNet 위치)로 변경됩니다. 조건부 위치와 시스템 위치를 수동으로 활성화할 수 있으며, 이 경우 명시적으로 사용 안함으로 설정할 때까지 위치가 활성 상태로 유지됩니다. 이 동작을 사용하면 조건부 위치 프로파일을 "항상 설정"으로 쉽게 전환할 수 있습니다. 조건부 위치를 사용 안함으로 설정하면 시스템이 다시 정상적인 조건부 동작으로 전환됩니다. 수동으로 임의 위치를 사용으로 설정하면 조건부 사용 위치의 조건이 충족된 경우에도 시스템이 위치를 변경하지 않습니다.

주 – 시스템에서 현재 활성 상태인 NCP를 명시적으로 사용 안함으로 설정할 수는 없습니다. 이 경우 시스템의 기본 네트워크 연결이 사실상 종료되기 때문입니다. 수동으로 다른 NCP를 사용으로 설정하면 NCP가 암시적으로 사용 안함으로 설정됩니다. 하지만 ENM 활성화에 대한 제약 조건은 없습니다. 항상 0개 이상의 ENM이 시스템에서 활성 상태일 수 있습니다. 따라서 ENM을 사용 또는 사용 안함으로 설정해도 현재 활성 상태인 다른 ENM에는 영향을 주지 않습니다.

개별 NCU를 수동으로 사용 또는 사용 안함으로 설정할 수도 있습니다. 지정한 NCU는 현재 활성 NCP의 일부여야 하며 `activation-mode` 등록 정보인 `manual`이 있어야 합니다. NCU 클래스를 지정하지 않으면 모든 NCU(링크 NCU 한 개 및 해당 이름을 가진 인터페이스 NCU 한 개)가 활성화되거나 비활성화됩니다.

객체 활성화와 비활성화는 비동기적으로 수행됩니다. 따라서 작업(활성화 또는 비활성화)은 실패해도 사용 또는 사용 안함으로 설정하는 요청이 성공할 수도 있습니다. 이 종류의 실패는 프로파일의 상태에 반영되어 상태가 `maintenance`로 변경됩니다. 이 상태는 프로파일에 대해 수행된 마지막 작업이 실패했음을 나타냅니다. 프로파일 상태 표시에 대한 자세한 내용은 102 페이지 “프로파일 상태에 대한 정보 가져오기”를 참조하십시오.

예 5–2 프로파일 사용

수동으로 프로파일을 사용으로 설정하는 구문은 다음과 같습니다.

`netadm enable` [**`-p`** *profile-type*][**`-c`** *ncu-class*] *profile-name*

예를 들어, 프로파일 이름이 고유하지 않은 경우 이름은 같지만 유형이 다른 여러 프로파일이 시스템에 있으면 프로파일 유형도 지정해야 합니다.

`-p` 옵션을 사용하여 다음 프로파일 유형 중 하나를 지정할 수 있습니다.

- `ncp`
- `ncu`
- `loc`
- `enm`

구성 객체의 유형이 `ncu`이면 `-c` 옵션을 사용하여 NCU 클래스를 구별할 수 있습니다. `-c` 옵션은 동일한 이름을 가진 NCU 두 개가 시스템에 있는 경우에 유용합니다.

`-c` 옵션을 사용하는 경우 `phys` 또는 `ip` 클래스 유형을 지정해야 합니다.

다음 예에서는 `office`라는 위치가 사용으로 설정됩니다.

```
$ netadm enable -p loc office
```

예 5–2 프로파일 사용 *(계속)*

여기서 *profile-type*은 `loc`이고 *profile-name*은 `office`입니다. 프로파일 유형이 위치이고 NCP가 아니기 때문에 이 예에서는 `-c` *ncu-class* 옵션이 사용되지 않습니다.

```
$ netadm enable -p ncp user
Enabling ncp 'User'
.
.
.
```

프로파일 이름을 지정할 때 `netadm` 명령은 대소문자를 구분하지 않습니다.

예 5–3 프로파일 사용 안함

수동으로 프로파일을 사용 안함으로 설정하는 구문은 다음과 같습니다.

netadm disable [**-p** *profile-type*][**-c** *ncu-class*] *profile-name*

프로파일 이름이 고유하지 않은 경우 프로파일 유형도 지정해야 합니다.

`-p` 옵션을 사용하여 다음 프로파일 또는 객체 유형 중 하나를 지정할 수 있습니다.

- `ncp`
- `ncu`
- `loc`
- `enm`

구성 객체의 유형이 `ncu`이면 NCU 클래스를 구별하기 위해 `-c` 옵션도 사용해야 합니다.

NCU 클래스는 `phys` 또는 `ip`로 지정해야 합니다.

예를 들어, `net1`이라는 링크 NCU를 사용 안함으로 수동 설정하려면 다음 명령을 입력합니다.

```
$ netadm disable -p ncu -c phys net1
```

여기서 *profile-type*은 `ncu`이고 *ncu-class*는 `phys`이고 *profile-name*은 `net1`입니다. 구성 객체가 NCU이기 때문에 이 예에서는 `-c` *ncu-class* 옵션이 사용됩니다.

예 5–4 프로파일 전환

활성 NCP를 변경하고 수동 구성을 사용으로 설정하려면 다음 명령을 입력합니다.

```
$ netadm enable -p ncp DefaultFixed
```

이와 유사하게, 자동 NCP가 포함된 자동(NWAM) 구성을 사용으로 설정하려면 다음 명령을 입력합니다.

예 5–4 프로파일 전환 *(계속)*

```
$ netadm enable -p ncp Automatic
```

netadm에 대한 자세한 내용은 netadm(1M) 매뉴얼 페이지를 참조하십시오.

무선 검색 수행 및 사용 가능한 무선 네트워크에 연결

netadm 명령을 사용하여 사용 가능한 무선 네트워크를 검색하고 연결할 수 있습니다.

무선 링크를 검색하여 사용 가능한 무선 네트워크 목록을 가져오려면 netadm scan-wifi *link-name* 명령을 사용합니다.

*link-name*으로 지정된 링크의 검색 결과에서 무선 네트워크를 선택하고 연결하려면 netadm select-wifi *link-name* 명령을 사용합니다. select-wifi *link-name* 하위 명령은 필요한 경우 WiFi 선택, 키 및 키 슬롯을 묻는 메시지를 표시합니다.

주 – netadm select-wifi 명령을 사용하기 전에 이미 키를 만든 상태여야 합니다.

---

netadm scan-wifi *link-name* 명령으로 후속 네트워크 검색을 트리거하여 사용 가능한 무선 네트워크를 검색할 수도 있습니다. 새 검색 결과가 기존 검색 결과와 동일한 경우 후속 검색에서 검색 이벤트를 트리거하지 않을 수도 있습니다. nwamd 데몬은 마지막 검색 이후 데이터가 변경되었는지 여부에 관계없이 검색합니다.

다음 예에서는 netadm scan-wifi 명령을 사용하여 무선 링크 net1을 검색합니다. 그런 다음 netadm select-wifi 명령을 사용하여 선택할 무선 네트워크 목록을 표시합니다. 표시되는 목록은 이전에 net1에서 수행된 검색 결과를 기반으로 합니다.

```
$ netadm select-wifi net1
1: ESSID home BSSID 0:b:e:85:26:c0
2: ESSID neighbor1 BSSID 0:b:e:49:2f:80
3: ESSID testing BSSID 0:40:96:29:e9:d8
4: Other
Choose WLAN to connect to [1-4]: 1
$
```

이 예에서는 숫자 1이 나타내는 무선 네트워크가 home 네트워크를 선택합니다.

WLAN에 키가 필요한 경우 키와 키 슬롯을 입력하라는 메시지가 표시됩니다(WEP가 지정된 경우). 예를 들면 다음과 같습니다.

```
Enter WLAN key for ESSID home: mywlankey
Enter key slot [1-4]: 1
```

NWAM 네트워크 구성 문제 해결

이 절의 정보는 NWAM 네트워크 구성 문제를 해결하는 방법에 대해 설명합니다.

모든 네트워크 연결의 현재 상태 모니터링

netadm 명령을 show-events 하위 명령과 함께 사용하여 NWAM 데몬인 nwamd가 모니터하는 이벤트를 수신 대기하고 표시할 수 있습니다. 이 하위 명령은 NWAM에서 구성될 때 프로파일 및 구성 객체의 구성 프로세스와 관련된 이벤트에 대한 유용한 정보를 제공합니다.

netadm show-events 명령의 구문은 다음과 같습니다.

```
netadm show-events [-v]
```

다음 예에서는 nwam show-events 명령에 -v 옵션을 사용하여 Verbose 모드로 이벤트를 표시합니다.

```
$ netadm show-events -v
EVENT                   DESCRIPTION
LINK_STATE              net0 -> state down
OBJECT_STATE            ncu link:net0 -> state online*, interface/link is down
OBJECT_STATE            ncu link:net0 -> state offline, interface/link is down
OBJECT_STATE            ncu interface:net0 -> state online*, conditions for act
OBJECT_STATE            ncu interface:net0 -> state offline, conditions for act
IF_STATE                net0 -> state (0) flags 2004801
IF_STATE                net0 -> state (0) flags 2004800
IF_STATE                net0 -> state (0) flags 1004803
IF_STATE                net0 -> state index 4 flags 0x0 address fe80::214:4fff:
IF_STATE                net0 -> state (0) flags 1004802
IF_STATE                net0 -> state index 4 flags 0x0 address 129.156.235.229
IF_STATE                net0 -> state (0) flags 1004803
IF_STATE                net0 -> state (0) flags 1004802
IF_STATE                net0 -> state (0) flags 1004803
IF_STATE                net0 -> state (0) flags 1004802
```

네트워크 인터페이스 구성 문제 해결

netadm list -x 명령은 네트워크 인터페이스가 올바르게 구성되지 않을 수 있는 이유를 확인하는 데 유용합니다. 이 명령은 NWAM에서 구성된 다양한 엔티티, 현재 상태 및 이러한 엔티티가 해당 상태인 이유를 표시합니다.

예를 들어, 케이블이 연결되지 않은 경우 netadm list -x 명령을 사용하여 링크 상태가 offline인지 여부 및 이유(예: "link is down")를 확인할 수 있습니다. 이와 유사하게, 중복

주소 감지의 경우 `netadm list -x` 명령의 출력 결과에 물리적 링크가 online(작동) 상태이지만 IP 인터페이스가 유지 관리 상태라고 표시됩니다. 이 인스턴스에서 제공된 이유는 "Duplicate address detected"입니다.

다음은 `netadm list -x` 명령 출력 결과의 예입니다.

```
$ netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         Automatic      online         active
ncu:phys    net0           offline        interface/link is down
ncu:ip      net0           offline        conditions for activation are unmet
ncu:phys    net1           offline*       need WiFi network selection
ncu:ip      net1           offline        conditions for activation are unmet
ncp         User           disabled       disabled by administrator
loc         Automatic      offline        conditions for activation are unmet
loc         NoNet          online         active
loc         office         offline        conditions for activation are unmet
$
```

링크 또는 인터페이스가 offline 상태인 이유를 확인한 후 문제 해결을 계속할 수 있습니다. 중복 IP 주소의 경우 `netcfg` 명령을 사용하여 지정한 인터페이스에 할당된 정적 IP 주소를 수정해야 합니다. 지침은 88 페이지 “프로파일의 등록 정보 값 설정 및 변경”을 참조하십시오. 변경 사항을 커밋한 후 `netadm list -x` 명령을 다시 실행하여 이제 인터페이스가 올바르게 구성되었으며 해당 상태가 `online`으로 표시되는지 확인합니다.

인터페이스가 올바르게 구성되지 않을 수 있는 이유의 또 다른 예는 사용 가능한 알려진 WLAN이 없는 경우입니다. 이 경우 WiFi 링크의 상태가 `offline`으로 표시되며 이유는 "need wifi selection"입니다. 또는 WiFi 선택이 수행되었지만 키가 필요한 경우 이유는 "need wifi key"가 됩니다.

◆ ◆ ◆ 6 장

6 NWAM 그래픽 사용자 인터페이스 정보

이 장에서는 NWAM GUI를 구성하는 구성 요소에 대한 설명을 포함하여 NWAM GUI(그래픽 사용자 인터페이스)에 대한 소개를 제공합니다. 이 장에는 데스크탑에서 NWAM과 상호 작용, 네트워크 연결 제어, 무선 네트워크 추가, 네트워크 프로파일 만들기 및 관리에 대한 기본 지침도 들어 있습니다.

GUI를 사용하여 네트워크를 배타적으로 관리하는 방법의 단계별 지침은 제공되지 않습니다. 자세한 지침은 항상 데스크탑의 패널 알림 영역에 표시되는 Network Status(네트워크 상태) 아이콘을 마우스 오른쪽 버튼으로 눌러 액세스할 수 있는 온라인 도움말을 참조하십시오. GUI 내의 링크는 각 항목에 대한 자세한 정보를 제공하는 온라인 도움말 페이지로 이동합니다. 텍스트에 표시되는 링크를 누르거나 측면 창에서 다양한 항목을 눌러 온라인 도움말을 탐색할 수도 있습니다.

이 장에서는 다음 항목을 다룹니다.

NWAM 그래픽 사용자 인터페이스 소개

NWAM GUI(그래픽 사용자 인터페이스)는 NWAM 명령줄 사용자 인터페이스와 동등한 그래픽 항목입니다. NWAM GUI를 사용하면 데스크탑에서 네트워크 상태를 보고 모니터하는 것은 물론 NWAM과 상호 작용하여 이더넷 및 무선 구성을 관리할 수 있습니다. 또한 시작 시 유선 또는 무선 네트워크에 연결, 새 유선 또는 무선 네트워크 구성과 같은 다양한 네트워킹 작업을 데스크탑에서 수행할 수 있습니다. NWAM GUI를 사용하여 시스템 차원 네트워크 구성의 복잡한 작업을 단순화하는 프로파일인 위치를

만들고 관리할 수도 있습니다. GUI 구성 요소에는 네트워크 연결의 현재 상태에 대한 알림과 네트워크 환경의 전체 상태에 대한 정보를 표시하는 기능이 포함됩니다.

NWAM GUI의 기본 기능은 다음과 같습니다.

- 네트워크 상태 알림
- 핫 플러그형 이벤트 감지
- 네트워크 프로파일 만들기 및 관리
- 무선 네트워크 관리

NWAM GUI는 원하는 등록 정보 값을 프로파일 형태로 시스템에 저장하여 NWAM CLI와 동일한 방식으로 네트워크 구성을 관리합니다. NWAM 서비스는 현재 네트워크 상태에 따라 지정된 시간에 활성화할 프로파일을 확인하고 가장 적절한 프로파일을 활성화합니다.

데스크탑에서 NWAM GUI 액세스

NWAM GUI를 구성하는 두 가지 구성 요소가 있습니다. Network Status(네트워크 상태) 알림 아이콘은 데스크탑 패널에 계속 표시되고, 네트워크 구성 대화 상자는 System(시스템) → Administration(관리) 메뉴에서 또는 알림 아이콘을 마우스 오른쪽 버튼으로 눌러 액세스할 수 있습니다. NWAM GUI는 전원 관리 아이콘이나 프린터 아이콘과 같이 지속적인 상태 알림 아이콘이 있는 다른 응용 프로그램과 동일한 방식으로 동작합니다. 이러한 응용 프로그램에서 마우스 오른쪽 버튼을 누르면 나타나는(컨텍스트) 메뉴에 액세스하거나 아이콘 또는 다양한 기본 설정 메뉴에서 액세스되는 구성 대화 상자를 사용하여 특정 작업을 수행할 수 있습니다.

패널 아이콘은 NWAM에서 가장 자주 사용되는 항목입니다. 이 아이콘은 현재 유선 또는 무선 네트워크에 연결되었는지를 보여줍니다. 아이콘 위로 마우스를 이동하면 현재 활성 NCP 및 위치 프로파일과 같은 추가 정보가 도구 설명에 표시됩니다. 아이콘을 마우스 오른쪽 버튼으로 누르면 다른 무선 네트워크에 연결하는 등 시스템의 기본 네트워크 구성을 변경할 수 있습니다.

패널 아이콘을 누르면(마우스 왼쪽 버튼으로 누르면) Network Preferences(네트워크 기본 설정) 대화 상자가 열립니다. System(시스템) →Administration(관리) 메뉴에서 이 대화 상자를 열 수도 있습니다. 여기서 정적 IPv4 및 IPv6 주소 정의, 연결 우선 순위 설정, ENM(외부 네트워크 수정자) 관리, 다른 위치에서 사용할 네트워크 설정 그룹 만들기 등의 자세한 네트워크 구성을 수행할 수 있습니다.

NWAM CLI와 NWAM GUI 간의 차이점

CLI 또는 GUI를 사용하여 NWAM을 통해 네트워크 구성을 관리할 수 있습니다. 두 사용자 인터페이스는 모두 네트워크 구성을 관리하고 NWAM 구성과 상호 작용하는 데 사용할 수 있습니다. 특정 작업을 수행할 때 CLI 또는 GUI를 사용할지 여부는 해당 작업과 주어진 상황에 따라 달라집니다. 일부 작업의 경우 NWAM GUI를 사용하는 것이

논리적으로 가장 적합한 선택입니다. 예를 들어, 현재 활성 네트워크 연결의 상태를 확인하거나 시작 시 연결할 무선 네트워크를 선택하는 경우가 있습니다. 이러한 작업은 GUI를 통해 데스크탑에서 NWAM과 직접 상호 작용함으로써 더 쉽고 빠르게 수행할 수 있습니다. 새 ENM을 시작하고 중지하는 방법으로 스크립트를 지정하는 경우와 같은 더 복잡한 작업의 경우 명령줄 모드에서 작업할 수 있습니다.

CLI와 GUI는 근본적으로 동일하지만 다음과 같은 차이점에 주의해야 합니다.

- **기능 차이점**

  GUI에는 NWAM과 상호 작용하고 데스크탑에서 네트워크 연결을 확인할 수 있는 기능이 있습니다. GUI 및 CLI 유틸리티 간에는 네트워크 상태에 관한 정보를 가져오는 방법이 약간 다릅니다. GUI 구성 요소를 사용하는 경우 알림이 발생할 때 데스크탑에 표시됩니다. 명령줄 유틸리티를 사용하는 경우 NWAM 이벤트가 발생할 때 `netadm show-events` 명령을 사용하여 이벤트를 모니터할 수 있습니다. 자세한 내용은 108 페이지 “모든 네트워크 연결의 현재 상태 모니터링”을 참조하십시오.

  또한 GUI를 사용하여 네트워크 상태에 대한 정보를 가져오려면 데스크탑에 표시되는 Network Status(네트워크 상태) 알림 아이콘을 시각적으로 확인하거나, 아이콘 위로 마우스를 이동하거나, 아이콘을 누릅니다. 명령줄에서 네트워크 상태에 대한 정보를 가져오려면 `netadm` 명령에 `list` 하위 명령을 사용합니다. 이 명령의 출력 결과에는 시스템에 구성된 각 네트워크 객체의 기본 상태에 대한 정보가 포함됩니다. 하지만 GUI는 연결된 무선 네트워크 및 네트워크 연결의 IP 주소 등 네트워크 상태에 대한 더 자세한 정보를 제공합니다.

  CLI를 사용하여 수행할 수 있는 일부 명령을 GUI에서는 수행할 수 없습니다. 예를 들어, GUI 구성 요소를 사용하여 프로파일 구성을 내보낼 수 없습니다. 프로파일 구성을 내보내려면 `netcfg export` 명령을 사용합니다. 자세한 내용은 95 페이지 “프로파일 구성 내보내기 및 복원”을 참조하십시오.

- **구성 요소 이름 및 용어 사용 차이점**

  GUI에서는 NCP(네트워크 구성 프로파일)가 **네트워크 프로파일**과 같습니다. CLI에서 NCU(네트워크 구성 단위)라고 불리는 항목이 GUI에서는 **네트워크 연결**을 나타냅니다.

  명령줄 인터페이스를 사용하여 NCP를 사용 및 사용 안함으로 설정하는 작업은 GUI를 사용 중인 경우의 **네트워크 프로파일 또는 연결 전환** 작업과 동일합니다.

NWAM GUI의 기능 구성 요소

NWAM GUI에는 CLI를 사용하여 수행할 수 있는 것과 거의 동일한 작업을 수행하는 데 사용되는 여러 기능 구성 요소가 있습니다. 표 6–1에서는 이러한 각 구성 요소에 대해 설명합니다. 일부 대화 상자는 여러 가지 방법으로 액세스하거나 열 수 있습니다. 또한 일부 대화 상자는 액세스 방식에 따라 다른 정보를 표시합니다. 이러한 차이점에 대한 특정 정보는 이 장 전체의 관련 절에 나와 있으며 온라인 도움말에서 자세히 설명합니다.

표 6–1 NWAM GUI 주요 구성 요소

구성 요소	기능	액세스 방법
Network Status(네트워크 상태) 알림 아이콘	데스크탑에서 네트워크의 상태를 확인하고 NWAM과 상호 작용하는 방법입니다. 이 아이콘에는 GUI를 사용하여 네트워크 구성을 만들고 관리하기 위해 액세스할 수 있는 컨텍스트 메뉴도 포함되어 있습니다.	■ 항상 데스크탑 패널의 알림 영역에 표시되는 아이콘을 확인합니다. ■ 현재 네트워크 상태에 대한 정보를 제공하는 도구 설명이 표시되도록 마우스를 아이콘 위로 이동합니다. ■ Network Preferences(네트워크 기본 설정) 대화 상자가 표시되는 아이콘을 누릅니다. ■ 컨텍스트 메뉴가 열리는 아이콘을 마우스 오른쪽 버튼으로 누릅니다.
Network Preferences(네트워크 기본 설정) 대화 상자	두 가지 주 네트워크 프로파일 유형인 시스템 정의 자동 프로파일과 여러 사용자 정의 네트워크 프로파일을 활성화하고 관리하는 방법입니다. 자동 및 사용자 정의 네트워크 프로파일은 개별 네트워크 인터페이스의 네트워크 구성을 관리합니다. 이 대화 상자는 개별 네트워크 인터페이스의 IPv4 및 IPv6 주소를 구성하고 즐겨 찾는 무선 네트워크를 관리하는 데도 사용됩니다.	■ 데스크탑에서 Network Status(네트워크 상태) 알림 아이콘을 누릅니다. ■ 데스크탑 패널의 기본 메뉴 모음에서 System(시스템) → Administration(관리) → Network(네트워크)를 선택합니다. ■ Network Status(네트워크 상태) 알림 아이콘 메뉴에서 Network Preferences(네트워크 기본 설정)를 선택합니다.

표 6–1 NWAM GUI 주요 구성 요소 (계속)

구성 요소	기능	액세스 방법
Network Locations(네트워크 위치) 대화 상자	시스템 정의 및 사용자 정의 위치 프로파일의 등록 정보를 만들고, 활성화 및 관리하는 방법입니다. 위치는 필요한 경우 함께 적용되는 특정 네트워크 구성 요소(예: 이름 지정 서비스 및 방화벽 설정)를 지정합니다.	■ Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르면 나타나는 메뉴에서 Network Locations(네트워크 위치)를 선택합니다. ■ 또는 Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Locations(위치) 버튼을 누릅니다.
Join Wireless Network(무선 네트워크 연결) 대화 상자	무선 네트워크를 연결하고 즐겨찾는 네트워크 목록을 관리하는 방법입니다. **주 –** 이 대화 상자는 사용자가 무선 네트워크를 추가하려고 시도할 때 네트워크에 대한 추가 정보가 필요한 경우에 자동으로 열립니다.	■ 알림 아이콘을 마우스 오른쪽 버튼으로 누르면 나타나는 메뉴에서 Join Unlisted Wireless Network(나열되지 않은 무선 네트워크 연결) 옵션을 선택합니다. ■ Wireless Chooser(무선 선택기) 대화 상자에서 Join Unlisted(나열되지 않은 무선 네트워크 연결) 버튼을 누릅니다. ■ "No wireless networks found. Click this message to join an unlisted wireless network.(무선 네트워크를 찾을 수 없습니다. 나열되지 않은 네트워크를 연결하려면 이 메시지를 누르십시오.)"라는 알림 메시지를 누릅니다.

표 6–1 NWAM GUI 주요 구성 요소 *(계속)*

구성 요소	기능	액세스 방법
Wireless Chooser(무선 선택기) 대화 상자	무선 네트워크를 선택하고 연결하는 방법입니다.	"*interface* disconnected from *ESSID*. Click this message to view other available networks.(interface이(가) ESSID에서 연결 해제됨. 사용 가능한 다른 네트워크를 보려면 이 메시지를 누르십시오.)"라는 알림 메시지를 누릅니다. **주 –** 이 대화 상자는 사용 가능한 무선 네트워크 중 연결할 네트워크를 선택할 수 있을 때마다 자동으로 열립니다.
Network Modifiers(네트워크 수정자) 대화 상자	네트워크 구성을 만들거나 수정할 수 있는 외부 네트워크 수정자 응용 프로그램을 추가하는 방법입니다.	■ Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Modifiers(수정자) 버튼을 누릅니다. ■ Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르고 Network Modifier Preferences(네트워크 수정자 기본 설정) 메뉴 항목을 선택합니다.

데스크탑에서 NWAM과 상호 작용

항상 데스크탑 패널의 알림 영역에 표시되는 Network Status(네트워크 상태) 알림 아이콘은 네트워크의 상태를 확인하고 자동 네트워크 구성 프로세스와 상호 작용하는 주요 방법입니다. Network Status(네트워크 상태) 알림 아이콘은 네트워크에 대한 정보 메시지가 표시되는 위치이기도 합니다. 이 아이콘의 컨텍스트(마우스 오른쪽 버튼을 누르면 나타나는) 메뉴를 사용하면 중요한 네트워크 기능에 빠르게 액세스할 수 있습니다. 아이콘 모양은 네트워크의 전반적인 상태를 나타냅니다.

네트워크 연결 상태 확인

네트워크에 대한 중요한 정보를 얻는 가장 빠른 방법은 데스크탑의 패널 알림 영역에 표시되는 Network Status(네트워크 상태) 알림 아이콘을 확인하는 것입니다. Network Status(네트워크 상태) 알림 아이콘은 현재 사용으로 설정된 네트워크 연결의 상태를

확인하고 NWAM과 상호 작용하는 주요 방법입니다. 아이콘의 모양은 현재 사용으로 설정된 네트워크 연결의 상태에 따라 변경됩니다. 현재 사용으로 설정된 네트워크 연결에 대한 정보를 표시할 수 있는 또 다른 방법은 마우스를 Network Status(네트워크 상태) 알림 아이콘 위로 이동하는 것입니다. 알림 아이콘의 컨텍스트 메뉴에 액세스하려면 아이콘을 마우스 오른쪽 버튼으로 누릅니다. 여기서 현재 사용으로 설정된 네트워크 인터페이스를 변경하고 연결된 무선 네트워크(있는 경우)에 대한 자세한 정보를 볼 수 있습니다.

주 – Network Status(네트워크 상태) 알림 아이콘은 NWAM을 사용하여 네트워크를 자동으로 구성하는 경우에만 데스크탑에 표시됩니다.

다음 표에서는 시스템에서 사용으로 설정된 네트워크 연결의 상태에 따라 변경되는 Network Status(네트워크 상태) 알림 아이콘의 모양을 보여줍니다.

아이콘	상태	설명
	모두 온라인(유선)	사용으로 설정된 네트워크 프로파일에서 사용으로 수동 설정된 모든 연결이 온라인 상태이고, 사용으로 설정된 프로파일 그룹(해당 그룹이 있는 경우)에서 필요한 수의 연결이 온라인 상태임을 나타냅니다. \"필요한 수\"는 다음과 같습니다. ■ 배타적 우선 순위 유형의 그룹인 경우 하나의 연결 ■ 공유 우선 순위 유형의 그룹인 경우 하나 이상의 연결 ■ 모두 우선 순위 유형의 그룹인 경우 그룹의 모든 연결
	모두 온라인(무선)	사용으로 설정된 네트워크 프로파일에서 사용으로 수동 설정된 모든 연결이 온라인 상태이고, 사용으로 설정된 프로파일 그룹(해당 그룹이 있는 경우)에서 필요한 수의 연결이 온라인 상태임을 나타냅니다. 필요한 수는 **모두 온라인(유선)** 상태에 대해 설명된 개수와 같습니다. 적어도 하나의 온라인 연결이 무선입니다.

아이콘	상태	설명
	부분적으로 온라인(유선)	사용으로 수동 설정된 연결이나 우선 순위 그룹 연결이 하나 이상 오프라인 상태이므로 상태가 더 이상 **모두 온라인**이 아님을 나타냅니다. 이 예에서는 적어도 하나의 유선 연결이 온라인 상태입니다. 무선 연결이 사용 가능한 무선 네트워크 선택 또는 무선 네트워크 암호 제공과 같은 사용자 입력을 대기 중인 경우에도 Network Status(네트워크 상태) 알림 아이콘이 **부분적으로 온라인**으로 표시됩니다.
	오프라인(유선)	NWAM 서비스가 사용 안함으로 설정되거나 유지 관리 모드에 있음을 나타냅니다.

▼ 사용으로 설정된 네트워크 연결에 대한 세부 정보를 표시하는 방법

1. **필요한 경우 Network Preferences(네트워크 기본 설정) 대화 상자를 열고 드롭다운 목록에서 Connection Status(연결 상태)를 선택합니다.**

   다음 방식 중 하나로 Network Preferences(네트워크 기본 설정) 대화 상자를 열 수 있습니다.

   - **데스크탑 패널에서 Network Status(네트워크 상태) 알림 아이콘을 누릅니다.**
   - **데스크탑 패널의 기본 메뉴 모음에서 System(시스템) → Administration(관리) → Network(네트워크)를 선택합니다.**
   - **Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 눌러 메뉴를 열고 Network Preferences(네트워크 기본 설정)를 선택합니다.**

     무선 네트워크 연결의 경우 IP 주소, 신호 강도, 연결 속도, 연결 상태 및 보안 유형이 표시됩니다.

2. **특정 네트워크 연결의 추가 등록 정보를 보거나 편집하려면 목록에서 연결을 두 번 누르거나 대화 상자의 맨 위에 있는 Show(표시) 드롭다운 메뉴에서 연결을 선택합니다.**

데스크탑에서 네트워크 연결 제어

기본적으로 NWAM은 항상 네트워크 연결을 유지하려고 합니다. 유선 네트워크 연결이 실패하면 즐겨 찾는 무선 네트워크 중 하나에 연결을 시도합니다. 이 시도가 실패하면 사용자의 승인에 따라 다른 사용 가능한 무선 네트워크가 시도됩니다.

필요에 따라 유선 및 무선 네트워크 사이를 수동으로 전환할 수 있습니다.

주 – 모든 연결 유형에서 연결 동작은 현재 세션에 대해서**만** 설정됩니다. 시스템을 재부트하거나 연결을 해제하면 사용으로 설정된 네트워크 프로파일에 정의된 우선 순위에 따라 네트워크 연결 설정이 시도됩니다.

다음과 같은 방식으로 NWAM을 사용하여 데스크탑에서 네트워크 연결을 제어할 수 있습니다.

- **기본 연결 우선 순위 수정**

  기본적으로 모든 유선 네트워크 연결이 모든 무선 네트워크 연결보다 우선합니다. 즉, 무선 네트워크 연결은 유선 연결을 설정할 수 없는 경우에만 시도됩니다. 현재 위치에서 사용 가능한 무선 네트워크가 두 개 이상 있는 경우 연결할 네트워크를 선택하라는 메시지가 표시됩니다. 이 동작은 기본적으로 활성화되는 자동 네트워크 프로파일에서 정의됩니다. 다른 동작을 적용하려면 다른 네트워크 프로파일을 만들고 활성화해야 합니다.

- **유선 네트워크에서 무선 네트워크로 전환**

  자동 네트워크 프로파일을 사용으로 설정한 경우 사용으로 설정된 모든 유선 인터페이스에서 네트워크 케이블을 뽑습니다.

  기본적으로 즐겨 찾는 무선 네트워크를 사용할 수 있는 경우 즐겨찾기 목록에 표시되는 순서대로 네트워크 연결이 시도됩니다. 그렇지 않으면 Wireless Chooser(무선 선택기) 대화 상자가 표시됩니다. 이 대화 상자에서 연결할 네트워크를 선택할 수 있습니다.

  주 – Connection Properties(연결 등록 정보) 뷰의 Wireless(무선) 탭에서 무선 네트워크 연결 방식을 변경할 수 있습니다.

  자동 네트워크 프로파일이 아닌 다른 네트워크 프로파일을 사용으로 설정한 경우 무선 네트워크로 전환하는 데 사용되는 방법은 해당 네트워크 프로파일의 정의에 따라 달라집니다.

  다음 방법 중 하나를 선택합니다.

  - Network Status(네트워크 상태) 알림 아이콘의 Connections(연결) 하위 메뉴를 사용하여 유선 연결을 사용 안함으로 설정하고 무선 연결을 활성화합니다. 이 방법은 두 연결이 모두 수동 활성화 유형인 경우에만 사용할 수 있습니다.
  - 필요한 경우 사용으로 설정된 네트워크 프로파일을 편집하여 유선 연결을 활성화하고 다른 연결을 사용 안함으로 설정합니다.

  무선 연결을 설정할 때 알림 메시지가 표시됩니다.

- **무선 네트워크에서 유선 네트워크로 전환**

  자동 네트워크 프로파일을 사용으로 설정한 경우 사용 가능한 유선 인터페이스에 네트워크 케이블을 꽂습니다.

자동 네트워크 프로파일이 아닌 다른 네트워크 프로파일을 사용으로 설정한 경우 유선 네트워크로 전환하는 데 사용되는 방법은 해당 네트워크 프로파일의 정의에 따라 달라집니다.

다음 방법 중 하나를 선택합니다.

- Network Status(네트워크 상태) 알림 아이콘의 Connections(연결) 하위 메뉴를 사용하여 무선 연결을 사용 안함으로 설정하고 유선 연결을 사용으로 설정합니다. 이 방법은 두 연결이 모두 수동 활성화 유형인 경우에만 사용할 수 있습니다.
- 사용으로 설정된 네트워크 프로파일을 편집하여 유선 연결을 사용으로 설정하고 무선 연결을 사용 안함으로 설정합니다.

  유선 연결을 설정할 때 알림 메시지가 표시됩니다.

NWAM GUI를 사용하여 수행할 수 있는 기타 작업은 온라인 도움말을 참조하십시오.

즐겨 찾는 무선 네트워크 연결 및 관리

기본적으로 무선 네트워크 연결을 사용으로 설정한 경우 NWAM은 확인 메시지를 표시하지 않고 연결이 나열된 우선 순위 순서에 따라 즐겨찾기 목록에서 사용 가능한 네트워크에 연결을 시도합니다. 즐겨 찾는 네트워크를 사용할 수 없는 경우 Wireless Chooser(무선 선택기) 대화 상자가 열립니다. 이 대화 상자에서 연결할 무선 네트워크를 선택할 수 있습니다.

Network Preferences(네트워크 기본 설정) 대화 상자에 있는 Connection Properties(연결 등록 정보) 뷰의 Wireless(무선) 탭에서 무선 네트워크 연결 방식을 수정할 수도 있습니다. 필요한 경우 Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르면 나타나는 메뉴에 액세스하여 수동으로 다른 무선 네트워크에 연결할 수 있습니다.

참고 – Network Preferences(네트워크 기본 설정) 대화 상자를 통해 선택한 네트워크에 대한 Connection Properties(연결 등록 정보) 뷰에 액세스할 수 있습니다. 이 대화 상자에는 Show(표시)라는 드롭다운 목록이 있습니다. 이 목록을 사용하여 지정된 네트워크의 뷰를 전환할 수 있습니다. 각 뷰에는 수행할 수 있는 다른 작업 및 해당 뷰와 관련된 선택한 네트워크에 대한 정보가 있습니다.

다음 뷰는 시스템에 있는 각 네트워크 프로파일의 모든 네트워크 연결에 대해 존재합니다.

- Connection Status(연결 상태)
- Network Profile(네트워크 프로파일)
- Connection Properties(연결 등록 정보)

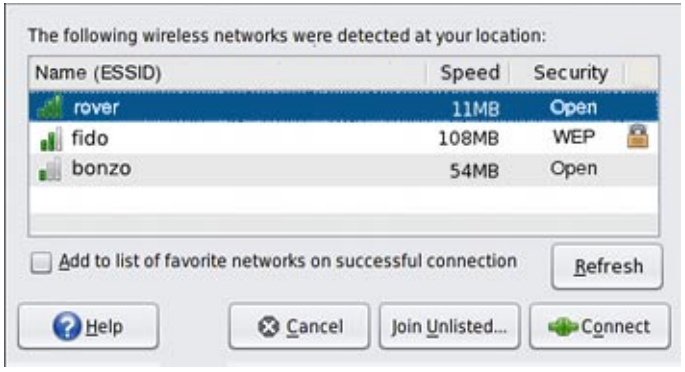
Network Preferences(네트워크 기본 설정) 대화 상자 설명을 비롯한 네트워크 프로파일 작업에 대한 자세한 내용은 122 페이지 “네트워크 프로필 관리”를 참조하십시오.

▼ 무선 네트워크를 연결하는 방법

Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르면 사용할 수 있는 Join Wireless Network(무선 네트워크 연결) 옵션을 선택하여 무선 네트워크를 연결합니다. Wireless Chooser(무선 선택기) 대화 상자에 표시되는 사용 가능한 네트워크 목록에서 연결할 무선 네트워크를 선택합니다.

1 **다른 무선 네트워크에 수동으로 연결하려면 다음 중 하나를 수행합니다.**

- **Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르면 나타나는 메뉴에서 사용 가능한 무선 네트워크를 선택합니다.**
- **Network Status(네트워크 상태) 알림 아이콘 메뉴에서 Join unlisted wireless network(나열되지 않은 무선 네트워크 연결) 옵션을 선택합니다.**
  나열되지 않은 무선 네트워크는 네트워크 이름을 브로드캐스트하지 않지만 아직 연결에 사용할 수 있도록 구성된 것입니다.
- **Wireless Chooser(무선 선택기) 대화 상자에서 사용 가능한 무선 네트워크를 선택합니다. 이 대화 상자는 사용 가능한 무선 네트워크 중 연결할 네트워크를 선택할 수 있을 때 자동으로 표시됩니다.**



2 **Join Wireless Network(무선 네트워크 연결) 대화 상자가 열리면 선택한 무선 네트워크에 대해 필요한 정보를 모두 제공합니다.**
제공해야 하는 정보에 대한 자세한 내용은 NWAM GUI 온라인 도움말을 참조하십시오.

즐겨 찾는 네트워크 관리

기본적으로 무선 네트워크에 처음 연결하면 Join Wireless Network(무선 네트워크 연결) 대화 상자에 Add to List of Favorite Networks on Successful Connection(연결되면 즐겨 찾는 네트워크 목록에 추가)라는 확인란이 표시됩니다.

- 연결에 성공할 경우 즐겨찾기 목록에 무선 네트워크를 추가하려면 이 확인란을 선택합니다. 즐겨찾기 목록에 네트워크를 추가하지 않으려면 확인란의 선택을 해제합니다. 이 확인란은 기본적으로 선택됩니다.
- 현재 사용할 수 없거나 현재 네트워크 이름을 즐겨찾기 목록에 브로드캐스트하지 않는 무선 네트워크를 추가하려면 Connection Properties(연결 등록 정보) 뷰의 Wireless(무선) 탭으로 이동한 다음 Add(추가) 버튼을 누릅니다. 네트워크를 추가하려면 해당 네트워크 이름, 보안 유형 및 보안 키를 알고 있어야 합니다.

네트워크 프로필 관리

NWAM GUI를 사용하는 경우 네트워크 프로파일이 42 페이지 “NCP에 대한 설명”에 설명된 NCP와 같습니다.

네트워크 프로파일은 지정된 한 시점에 사용 또는 사용 안함으로 설정할 수 있는 네트워크 인터페이스를 지정합니다. 네트워크 프로파일을 사용하면 사용 가능한 네트워크 인터페이스가 두 개 이상 있는 경우에 유용합니다. 예를 들어, 대부분의 현대식

랩탑 브랜드에는 유선 및 무선 인터페이스가 모두 있습니다. 물리적 위치 및 작업 환경에 따라 이러한 인터페이스 중 하나만 사용하고 보안이나 기타 이유로 다른 인터페이스를 사용 안함으로 설정할 수 있습니다.

NWAM GUI에서 사용할 수 있는 두 가지 네트워크 프로파일 유형에는 기본값인 자동 네트워크 프로파일과 사용자 정의 네트워크 프로파일이 있습니다. 두 유형의 프로파일을 모두 사용 및 사용 안함으로 설정할 수 있습니다. 사용자 정의 프로파일은 수정할 수 있지만 자동 프로파일은 수정할 수 없습니다. NWAM GUI 또는 CLI를 사용하여 자동 프로파일을 만들거나 삭제할 수 없습니다. 하지만 사용자 정의 네트워크 프로파일은 GUI 또는 CLI를 사용하여 만들고, 수정 및 삭제할 수 있습니다.

기본적으로 자동 네트워크 프로파일은 먼저 유선 연결을 사용으로 설정하려고 시도합니다. 이 시도가 실패하면 무선 연결을 사용으로 설정하려고 시도합니다.

네트워크 기본 설정 대화 상자 정보

Network Preferences(네트워크 기본 설정) 대화 상자에서는 개별 네트워크 연결을 구성하고 각 네트워크 연결의 현재 상태를 확인할 수 있습니다. 이 대화 상자의 맨 위에 있는 드롭다운 목록을 사용하여 전환할 수 있는 다양한 뷰에 액세스할 수 있습니다.

이 대화 상자는 다음과 같은 방식으로 열 수 있습니다.

- 데스크탑에서 Network Status(네트워크 상태) 알림 아이콘을 누릅니다.
- 데스크탑 패널의 기본 메뉴 모음에서 System(시스템) → Administration(관리) → Network(네트워크)를 선택합니다.
- Network Status(네트워크 상태) 알림 아이콘 메뉴에서 Network Preferences(네트워크 기본 설정)를 선택합니다.

Network Preferences(네트워크 기본 설정) 대화 상자의 맨 위에는 Show(표시)라는 드롭다운 목록이 있습니다. 이 목록을 사용하여 각 네트워크 프로파일의 모든 네트워크 연결에 대한 Connection Status(연결 상태) 뷰, Network Profile(네트워크 프로파일) 뷰 및 Connection Properties(연결 등록 정보) 뷰를 전환할 수 있습니다.

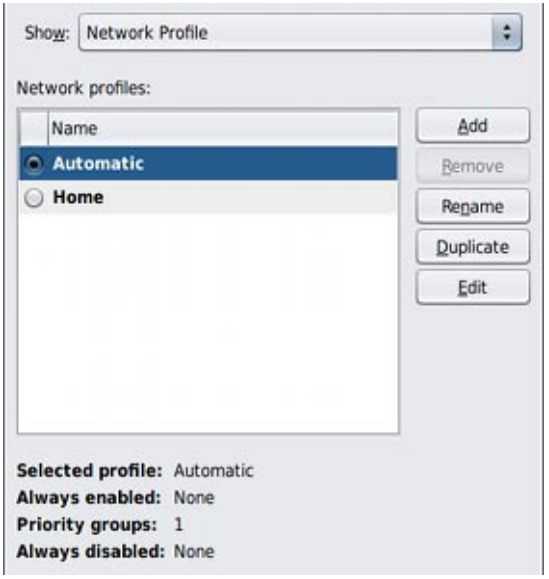
Connection Status(연결 상태) 뷰

- Connection Status(연결 상태) 뷰는 수동 활성화 유형인 사용으로 설정된 네트워크 프로파일에서 사용으로 설정된 각 네트워크 연결과 활성 우선 순위 그룹의 각 연결(사용 또는 사용 안함)에 대한 정보를 표시합니다. Enabled Connections:(사용으로 설정된 연결:) 섹션에는 사용으로 설정된 모든 연결이 Network Profile(네트워크 프로파일) 뷰에 나열된 것과 동일한 순서로 표시됩니다. 118 페이지 "사용으로 설정된 네트워크 연결에 대한 세부 정보를 표시하는 방법"을 참조하십시오.

Network Profile(네트워크 프로파일) 뷰

- 네트워크 프로파일 정보는 Network Preferences(네트워크 기본 설정) 대화 상자의 Network Profile(네트워크 프로파일) 뷰에서 확인할 수 있습니다.

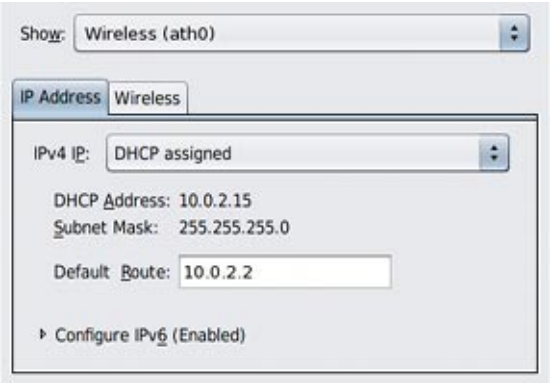
  이 뷰를 표시하려면 Network Preferences(네트워크 기본 설정) 대화 상자의 맨 위에 있는 드롭다운 목록에서 Network Profile(네트워크 프로파일)을 선택합니다.



Connection Properties(연결 등록 정보) 뷰

- Connection Properties(연결 등록 정보) 뷰에서는 지정한 네트워크 연결의 등록 정보를 확인하고 변경할 수 있습니다. 이 뷰로 전환하려면 Show(표시) 드롭다운 목록에서 연결 이름을 선택하거나 Connection Status(연결 상태) 또는 Network Profile(네트워크 프로파일) 뷰에 있는 동안 연결 이름을 두 번 누릅니다. 탭 뷰가 표시되며, 여기서 연결의 등록 정보를 확인하거나 편집할 수 있습니다.

  Connection Properties(연결 등록 정보) 뷰에는 두 개의 탭인 IP 주소 탭과 무선 탭이 있습니다. 무선 탭은 연결 유형이 무선인 경우에만 표시됩니다. 이 IP 주소 탭에서 IPv4 및 IPv6 주소를 구성할 수 있습니다. 무선 탭에서는 즐겨 찾는 네트워크 목록을 구성하고 무선 인터페이스가 사용 가능한 네트워크에 연결하는 방식을 선택할 수 있습니다.



네트워크 프로파일에 대한 정보 확인

네트워크 프로파일 정보는 Network Preferences(네트워크 기본 설정) 대화 상자의 Network Profile(네트워크 프로파일) 뷰에서 확인할 수 있습니다.

이 뷰를 표시하려면 Network Preferences(네트워크 기본 설정) 대화 상자의 맨 위에 있는 드롭다운 목록에서 Network Profile(네트워크 프로파일)을 선택합니다.

Network Profiles(네트워크 프로파일) 목록에는 사용 가능한 각 네트워크 프로파일의 이름이 표시됩니다. 현재 사용으로 설정된 프로파일은 라디오 버튼 표시기와 함께 표시됩니다. 기본적으로 활성화할 수 있지만 편집하거나 삭제할 수 없는 자동 프로파일 한 개가 있습니다. 하지만 여러 네트워크 프로파일을 추가로 만들 수 있습니다. 수동으로 만든 네트워크 프로파일은 필요에 따라 활성화, 편집 또는 삭제할 수 있습니다.

Network Profile(네트워크 프로파일) 목록 아래에는 선택한 프로파일의 요약이 표시됩니다. 선택한 프로파일의 전체를 표시하거나 프로파일을 편집하려면 Edit(편집) 버튼을 누릅니다.

주 – 선택한 프로파일이 **사용으로 설정된** 프로파일과 다를 수도 있습니다.

한 네트워크 프로파일에서 다른 네트워크 프로파일로 전환

1. Network Preferences(네트워크 기본 설정) 대화 상자의 Network Profile(네트워크 프로파일) 뷰를 엽니다.
2. 활성화할 네트워크 프로필 옆에 있는 라디오 버튼을 선택합니다.
3. 네트워크 프로파일을 전환하려면 OK(확인)를 누르고, 프로파일 전환 없이 대화 상자를 닫으려면 Cancel(취소)을 누릅니다.

네트워크 프로파일 추가 또는 제거

네트워크 프로파일을 만들거나 편집하려면 Network Preferences(네트워크 기본 설정) 대화 상자의 맨 위에 있는 드롭다운 목록에서 Network Profile(네트워크 프로파일)을 선택합니다.

- 새 네트워크 프로파일을 만들려면 Add(추가) 버튼을 누르고 새 프로파일의 이름을 입력합니다.
- 기존 네트워크 프로파일을 복제하려면 목록에서 프로파일을 선택하고 Duplicate(복제) 버튼을 누른 다음 새 프로파일의 이름을 입력합니다.
- 네트워크 프로파일을 제거하려면 목록에서 프로파일을 선택하고 Remove(제거) 버튼을 누릅니다.

주 – 자동 네트워크 프로필은 제거할 수 없습니다.

추가 또는 복제된 프로파일의 편집에 대한 자세한 내용은 126 페이지 “네트워크 프로필 편집”을 참조하십시오.

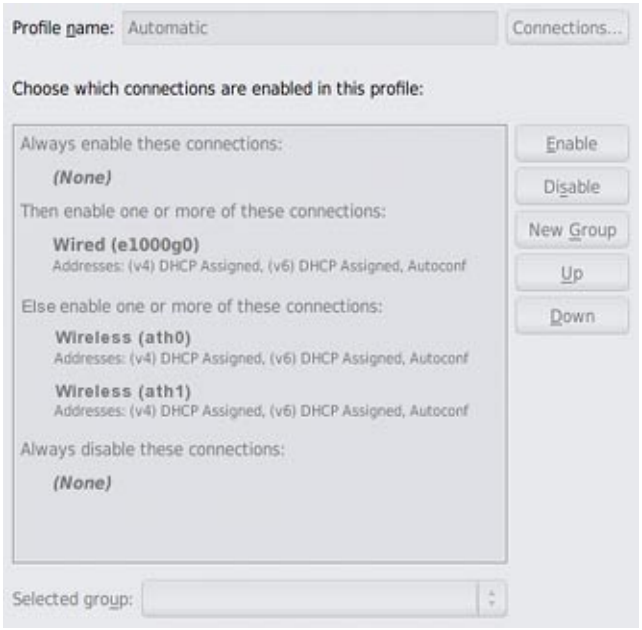
네트워크 프로필 편집

수동으로 새 네트워크 프로필을 추가하거나 기존 네트워크 프로필을 복제할 때 해당 네트워크 연결이 새 프로필에 의해 사용/사용 안함으로 설정되도록 새 프로필을 편집해야 합니다.

주 – 수동으로 만든 네트워크 프로파일은 편집 및 제거할 수 있습니다. 하지만 자동 네트워크 프로파일은 편집하거나 제거할 수 없습니다.

▼ Network Profile(네트워크 프로파일) 대화 상자를 여는 방법

- **네트워크 프로파일을 편집하려면 Network Preferences(네트워크 기본 설정) 대화 상자의 Network Profile(네트워크 프로파일) 뷰에서 프로파일을 선택하고 Edit(편집) 버튼을 누릅니다.**



네트워크 프로파일 목록은 두 개 이상의 최상위 그룹 설명으로 구성됩니다. 예를 들어, 앞의 그림에 표시된 자동 프로파일에는 다음 절에서 자세히 설명하는 네 개의 그룹 설명이 포함되어 있습니다.

주 – 자동 네트워크 프로파일은 변경하거나 삭제할 수 없습니다. Edit Network Profile(네트워크 프로파일 편집) 대화 상자에서 Automatic(자동) 네트워크 프로파일을 선택할 때마다 프로파일 편집 버튼과 드롭다운 목록이 모두 사용 안함으로 설정됩니다.

자세한 내용은 온라인 도움말을 참조하십시오.

우선 순위 그룹 작업

"always enabled(항상 사용)" 그룹의 네트워크 연결은 선택한 네트워크 프로파일이 활성일 때 항상 사용으로 설정됩니다.

네트워크 연결을 "always enabled(항상 사용)" 그룹으로 이동하려면 먼저 연결을 선택하고 다음 중 하나를 수행합니다.

- Enable(사용) 버튼을 누릅니다.
- 연결이 "always enabled(항상 사용)" 그룹으로 이동될 때까지 Up(위로) 버튼을 누릅니다.

"always disabled(항상 사용 안함)" 그룹의 네트워크 연결은 선택한 네트워크 프로파일이 활성일 때 항상 사용 안함으로 설정됩니다.

네트워크 연결을 "always disabled(항상 사용 안함)" 그룹으로 이동하려면 먼저 연결을 선택하고 다음 중 하나를 수행합니다.

- Disable(사용 안함) 버튼을 누릅니다.
- 연결이 "always disabled(항상 사용 안함)" 그룹으로 이동될 때까지 Down(아래로) 버튼을 누릅니다.

하나 이상의 네트워크 인터페이스를 그룹으로 처리하는 네트워크 프로파일을 만들 수 있습니다. 그룹의 우선 순위 유형에 따라 우선 순위가 가장 높은 그룹의 인터페이스를 하나 이상 사용으로 설정할 수 없는 경우 우선 순위가 다음으로 높은 그룹이 고려됩니다.

다음 표는 세 가지 사용 가능한 우선 순위 그룹을 설명합니다.

우선 순위 유형	설명
`Exclusive`	그룹의 한 연결은 사용으로 설정되고 다른 연결은 모두 사용 안함으로 설정됩니다. 그룹에서 적어도 하나의 연결이 사용으로 설정되어 있으면(항상 동일한 연결일 필요는 없음) 하위 우선 순위 그룹의 연결을 사용으로 설정하려고 시도되지 않습니다.
`Shared`	그룹에서 사용으로 설정할 수 있는 모든 연결이 사용으로 설정됩니다. 그룹에서 적어도 하나의 연결이 사용으로 설정되어 있으면 하위 우선 순위 그룹의 연결을 사용으로 설정하려고 시도되지 않습니다.
`All`	그룹의 모든 연결이 사용으로 설정됩니다. 연결이 하나라도 해제되면 그룹의 모든 연결이 사용 안함으로 설정됩니다. 모든 연결이 사용으로 설정되어 있으면 하위 우선 순위 그룹의 연결을 사용으로 설정하려고 시도되지 않습니다.

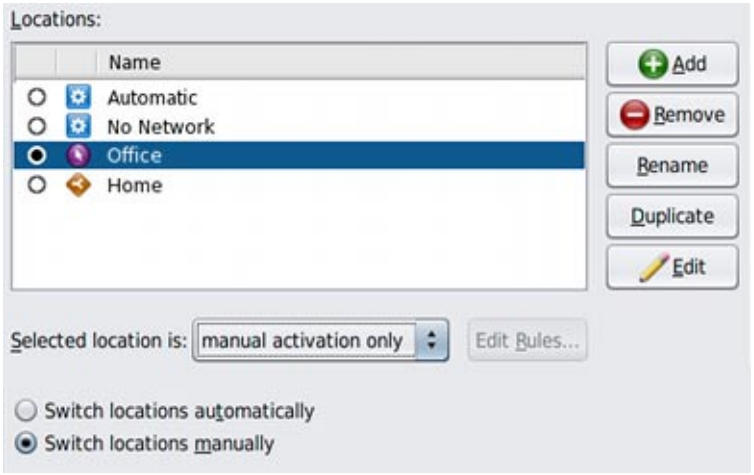
예를 들어, 기본값인 자동 네트워크 프로파일에는 두 개의 배타적 우선 순위 그룹이 있습니다. 상위 우선 순위 그룹에는 **유선** 네트워크 연결이 모두 포함됩니다. 하위 우선 순위 그룹에는 **무선** 네트워크 연결이 모두 포함됩니다.

이러한 작업과 기타 작업 수행에 대한 자세한 지침은 온라인 도움말을 참조하십시오.

위치 만들기 및 관리

위치는 필요한 경우 함께 적용되는 특정 네트워크 구성 요소(예: 이름 지정 서비스 및 방화벽 설정)로 구성됩니다. 다양한 용도로 여러 위치를 만들 수 있습니다. 예를 들어, 한 위치는 사무실에서 회사 인트라넷을 사용하여 연결된 경우에 사용할 수 있습니다. 다른 위치는 집에서 무선 액세스 포인트를 사용하여 공용 인터넷에 연결된 경우에 사용할 수 있습니다. 위치는 네트워크 연결에서 얻은 IP 주소와 같은 환경 조건에 따라 수동으로 또는 자동으로 활성화할 수 있습니다.

Network Locations(네트워크 위치) 대화 상자에서 위치를 전환하고, 위치 등록 정보를 편집하고, 새 위치를 만들고, 위치를 제거할 수 있습니다. 사용자 정의 위치만 만들고 제거할 수 있습니다. Location(위치) 대화 상자는 Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 열 수 있습니다.



Locations(위치) 목록은 Network Status(네트워크 상태) 알림 아이콘 메뉴의 목록과 유사합니다. 사용 가능한 각 위치가 활성화 유형을 나타내는 아이콘과 함께 나열됩니다.

위치 유형은 다음과 같습니다.

- System(시스템) – 이 유형의 위치는 시스템 정의 위치(Automatic(자동) 및 No Network(네트워크 없음))이므로 시스템이 현재 네트워크 상태에 따라 위치 활성화 시기를 결정합니다.
- Manual(수동) – 이 유형의 위치는 Network Locations(네트워크 위치) 대화 상자를 사용하거나 Network Status(네트워크 상태) 알림 아이콘과 상호 작용하여 수동으로 사용 또는 사용 안함으로 설정할 수 있습니다.
- Conditional(조건부) – 이 유형의 위치는 위치를 만드는 동안 지정한 규칙에 따라 자동으로 사용 또는 사용 안함으로 설정됩니다.

Selected location(선택된 위치) 드롭다운 목록에 선택한 위치의 활성화 유형도 표시됩니다. 사용으로 설정된 위치는 목록의 첫번째 열에 표시되는 선택된 라디오 버튼으로 식별됩니다.

▼ 위치의 활성화 모드를 변경하는 방법

다음 작업은 NWAM GUI를 사용하여 위치의 활성화 모드를 변경하는 방법에 대해 설명합니다. `netcfg` 명령을 사용하는 경우 지정한 위치의 등록 정보를 수정하여 활성화 모드를 변경합니다. 자세한 내용은 88 페이지 "프로파일의 등록 정보 값 설정 및 변경"을 참조하십시오.

1 **Network Status(네트워크 상태) 알림 아이콘의 Location(위치) 하위 메뉴에서 Network Locations(네트워크 위치)를 선택합니다. 또는 Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Locations(위치) 버튼을 누릅니다.**

2 **위치의 활성화 모드를 변경하려면 목록에서 위치를 선택하고 Selected location(선택된 위치) 드롭다운 목록에서 새 활성화 모드를 선택합니다.**

주 – 시스템 위치를 선택하면 드롭다운 목록에 Activated by system(시스템에 의해 활성화됨)이 표시되고 드롭다운 목록과 Edit Rules(규칙 편집) 버튼이 모두 사용 안함으로 설정됩니다.

수동 또는 조건부 위치를 선택한 경우 드롭다운 목록 옵션은 다음과 같습니다.

- Manual activation only(수동 활성화만): 이 위치가 수동으로 선택된 경우에만 사용으로 설정됩니다. 이 옵션을 선택하면 Edit Rules(규칙 편집) 버튼이 **사용 안함**으로 설정됩니다.
- Activated by rules(규칙에 의해 활성화됨): 이 위치가 특정 네트워크 조건에서 자동으로 선택됩니다. 이 옵션을 선택하면 Edit Rules(규칙 편집) 버튼이 **사용**으로 설정됩니다.

3 **(옵션) 위치 활성화 방식과 시기에 대한 규칙을 설정하려면 Edit Rules(규칙 편집) 버튼을 누릅니다.**

자세한 지침은 온라인 도움말에서 "규칙 대화 상자 작업"을 참조하십시오.

▼ 한 위치에서 다른 위치로 전환하는 방법

다음 작업에서는 NWAM GUI를 사용하여 한 위치에서 다른 위치로 전환하는 방법에 대해 설명합니다. CLI를 사용하여 위치를 전환하려면 `netadm` 명령을 사용하여 새 위치를 활성화합니다. 시스템에서는 항상 위치 한 개만 활성화되어야 하므로 새 위치를 활성화하면 현재 사용으로 설정된 위치가 암시적으로 사용 안함으로 설정됩니다. 네트워크 프로파일을 활성화하는 경우에도 동일한 규칙이 적용됩니다. 위치 활성화 및 비활성화에 대한 자세한 내용은 104 페이지 "프로파일 활성화 및 비활성화"를 참조하십시오.

- **Network Status(네트워크 상태) 알림 아이콘의 Location(위치) 하위 메뉴에서 활성화할 위치를 선택합니다.**

  Location(위치) 하위 메뉴에서 Switch Locations Automatically(자동으로 위치 전환) 옵션을 선택하면 수동으로 활성화할 위치를 선택할 수 없습니다. 네트워크 환경의 변경 사항에 따라 지정된 한 시점에 가장 적절한 시스템 또는 조건부 위치가 자동으로 활성화됩니다.

  Location(위치) 하위 메뉴에서 Switch Locations Manually(수동으로 위치 전환) 옵션을 선택하면 활성화 유형에 관계없이 사용 가능한 위치를 활성화할 수 있습니다. 선택한 위치는 무기한 활성 상태로 유지됩니다.

  - **또는 Network Locations(네트워크 위치) 대화 상자에서 위치를 전환할 수 있습니다. 이렇게 하려면 다음 단계를 수행합니다.**

    a. **Network Status(네트워크 상태) 알림 아이콘의 Location(위치) 하위 메뉴에서 Network Locations(네트워크 위치)를 선택합니다. 또는 Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Locations(위치) 버튼을 누릅니다.**

    b. **전환하려는 위치의 라디오 버튼을 선택하고 OK(확인)를 누릅니다.**

       - **Network Location(네트워크 위치) 대화 상자에서 Switch Locations Automatically(자동으로 위치 전환) 라디오 버튼을 선택하면 수동으로 활성화할 위치를 선택할 수 없습니다. 네트워크 환경의 변경 사항에 따라 지정된 한 시점에 가장 적절한 시스템 또는 조건부 위치가 자동으로 활성화됩니다.**

       - **Network Location(네트워크 위치) 대화 상자에서 Switch Locations Manually(수동으로 위치 전환) 라디오 버튼을 선택하면 활성화 유형에 관계없이 사용 가능한 위치를 활성화할 수 있습니다. 위치는 무기한 활성 상태로 유지됩니다.**

위치 편집

NWAM GUI를 사용하여 위치를 편집하는 것은 NWAM CLI를 사용하는 경우 위치의 등록 정보를 수정하는 것과 같습니다.

위치를 편집하려면 Network Status(네트워크 상태) 알림 아이콘의 Location(위치) 하위 메뉴에서 Network Locations(네트워크 위치)를 선택합니다. 또는 Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Locations(위치) 버튼을 누릅니다.

지정한 위치의 등록 정보를 편집하려면 목록에서 위치를 선택하고 Edit(편집)를 누릅니다.

다른 방법으로, 목록에서 위치를 두 번 누를 수 있습니다.

Edit Location(위치 편집) 대화 상자가 열리고 다음 두 탭을 사용할 수 있습니다.

Name Services(이름 서비스)	지정한 위치에 이름 지정 서비스를 구성할 수 있습니다.
Security(보안)	지정한 위치를 사용으로 설정한 경우 IP 필터 및 IPsec 기능에서 사용할 구성 파일을 선택할 수 있습니다.

편집할 정보를 표시하려면 적절한 탭을 선택합니다.

외부 네트워크 수정자 정보

ENM(외부 네트워크 수정자)은 NWAM의 외부 응용 프로그램에 대해 생성된 프로파일입니다. 하지만 이러한 응용 프로그램은 네트워크 구성을 만들고 수정할 수 있습니다. 예를 들어, VPN 응용 프로그램을 사용하면 네트워크 연결이 가상 개인 네트워크와 통신할 수 있습니다. ENM은 NWAM GUI에서 *Network Modifiers(***네트워크 수정자***)* 대화 상자를 사용하여 구성 및 모니터됩니다.

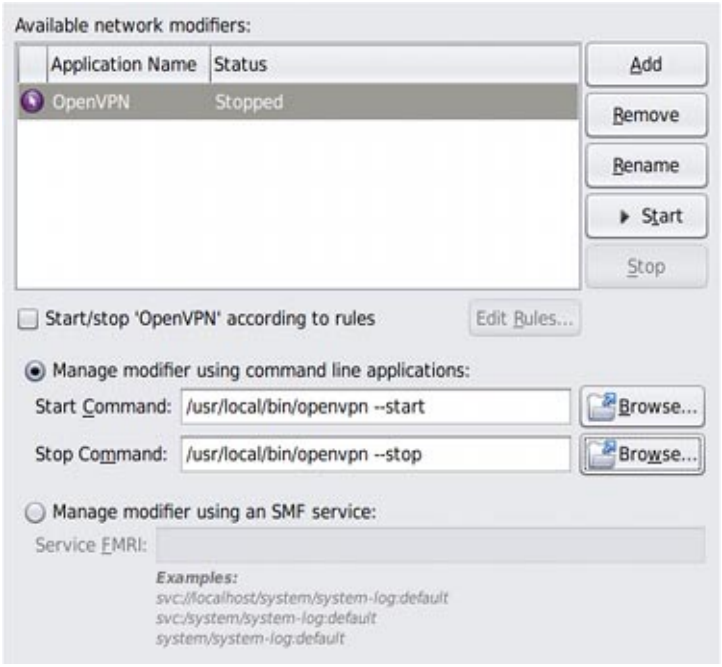
주 – NWAM GUI를 사용하여 네트워크 수정자 응용 프로그램이나 서비스를 관리하려면 먼저 수동으로 설치한 다음 인증서 또는 공유 암호 설치와 같은 초기 설정을 완료해야 합니다.

필요한 경우 ENM을 수동으로 시작하고 중지할 수 있습니다. 사용자 정의 규칙에 따라 ENM을 자동으로 시작할 수도 있습니다. 이 대화 상자를 사용하여 관리하려면 네트워크 수정자 응용 프로그램을 명령줄 도구나 SMF 서비스로 구현해야 합니다.

NWAM CLI를 사용하여 ENM을 만들고 관리하는 방법에 대한 자세한 내용은 82 페이지 “ENM 프로파일 만들기”를 참조하십시오.

네트워크 수정자 대화 상자 정보

이 대화 상자에서는 네트워크 구성을 만들고 수정할 수 있는 응용 프로그램인 ENM(외부 네트워크 수정자)을 추가 또는 제거하고, 시작 및 중지하고, 편집할 수 있습니다.



다음 방법 중 하나를 사용하여 대화 상자를 엽니다.

- Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Modifiers(수정자) 버튼을 누릅니다.
- Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르고 Network Modifier Preferences(네트워크 수정자 기본 설정) 메뉴 항목을 선택합니다.

각 ENM에 대해 다음 정보를 표시하는 3열 목록이 대화 상자의 주요 부분입니다.

- 활성화 상태(수동 또는 조건부)
- 사용자 정의 이름(예: \"Cisco VPN\")
- 현재 상태 \"실행 중\" 또는 \"중지됨\"

선택한 네트워크 수정자 응용 프로그램에 Conditional(조건부) 활성화 유형이 있는 경우 Start/Stop according to rules(규칙에 따라 시작/중지) 확인란이 선택되고, 활성화 유형이 Manual(수동)인 경우 선택이 해제됩니다. 활성화 유형을 변경하려면 확인란을 설정/해제합니다.

▼ 명령줄 ENM을 추가하는 방법

다음 절차에서는 명령줄 ENM을 추가하는 방법에 대해 설명합니다. 네트워크 수정자 응용 프로그램 서비스 추가에 대한 자세한 내용은 온라인 도움말을 참조하십시오.

1. **다음 방법 중 하나를 사용하여 Network Modifiers(네트워크 수정자) 대화 상자를 엽니다.**
   - Network Preferences(네트워크 기본 설정) 대화 상자의 Connection Status(연결 상태) 뷰에서 Modifiers(수정자) 버튼을 누릅니다.
   - Network Status(네트워크 상태) 알림 아이콘을 마우스 오른쪽 버튼으로 누르고 Network Modifier Preferences(네트워크 수정자 기본 설정) 메뉴 항목을 선택합니다.
2. **추가 버튼을 누릅니다.**
3. **새 네트워크 수정자 응용 프로그램의 이름을 입력합니다.**
4. **다음 중 하나를 수행합니다.**
   - **Manual(수동) 활성화 유형이 지정될 새 항목을 추가하려면 Enter 또는 Tab 키를 누릅니다.**

     Manage modifiers(수정자 관리) 라디오 버튼 두 개가 사용으로 설정됩니다. 첫번째 버튼인 Command Line Applications(명령줄 응용 프로그램)가 기본적으로 선택됩니다. Start Command(시작 명령) 및 Stop Command(중지 명령) 필드와 Browse(찾아보기) 버튼 두 개도 사용으로 설정됩니다.
   - **변경 사항을 취소하려면 Esc 키를 누릅니다.**
5. **Start Command(시작 명령) 필드에 네트워크 수정자 응용 프로그램을 시작하는 명령을 입력합니다.**

   또는 Browse(찾아보기) 버튼을 사용하여 파일 선택기 대화 상자를 열고, 여기서 사용할 명령을 선택할 수 있습니다.

   이 필드에 유효한 명령을 입력할 때까지 네트워크 수정자 응용 프로그램에 대해 Start(시작) 버튼이 사용 안함으로 설정됩니다.
6. **Stop Command(중지 명령) 필드에 네트워크 수정자 응용 프로그램을 중지하는 명령을 입력합니다.**

   또는 Browse(찾아보기) 버튼을 사용하여 파일 선택기 대화 상자를 열고, 여기서 사용할 명령을 선택할 수 있습니다.

   이 필드에 유효한 명령을 입력할 때까지 네트워크 수정자 응용 프로그램에 대해 Stop(중지) 버튼이 사용 안함으로 설정됩니다.
7. **이 응용 프로그램을 추가하려면 OK(확인)를 누릅니다.**

   외부 네트워크 수정자가 추가됩니다.

제 2 부

데이터 링크 및 인터페이스 구성

이 부에서는 제1부에 소개된 네트워크 구성 프로파일의 컨텍스트에서 데이터 링크 및 인터페이스 구성 절차에 대해 설명합니다. 이 절차는 사용으로 설정되었거나 활성화된 모든 수정된 프로파일에 적용됩니다.

◆ ◆ ◆ 7 장

7 프로파일에 데이터 링크 및 인터페이스 구성 명령 사용

이 장에서는 프로파일 기반 네트워크 구성과 관련될 경우 `dladm` 및 `ipadm`과 같은 일반적인 구성 명령 사용에 대해 설명합니다.

프로파일 기반 네트워크 구성의 주요 내용

이 Oracle Solaris 릴리스에서는 네트워크 구성이 프로파일을 기반으로 합니다. 시스템의 네트워크 구성 설정은 특정 NCP(네트워크 구성 프로파일) 및 해당 위치 프로파일에서 관리됩니다. NCP, 위치 프로파일과 기타 프로파일 유형, 해당 등록 정보 및 프로파일 관리와 모니터에 사용되는 명령에 대한 자세한 내용은 제1부을 참조하십시오.

주 – 네트워크 구성의 경우 주요 프로파일 유형은 NCP, 위치 프로파일, ENM(외부 네트워크 수정자) 및 WLAN(무선 LAN)입니다. 이러한 유형 중 기본 프로파일은 NCP입니다. 달리 지정되지 않은 경우 이 설명서 전체에서 **프로파일**이란 용어는 NCP를 가리킵니다.

프로파일 기반 네트워크 구성의 주요 내용은 다음과 같습니다.

- 한 번에 한 쌍의 NCP 및 위치 프로파일만 활성화되어 시스템의 네트워크 구성을 관리할 수 있습니다. 시스템의 다른 기존 NCP는 모두 작동하지 않습니다.
- 활성 NCP는 **반응적** 또는 **수정된** 프로파일일 수 있습니다. 반응적 프로파일을 사용할 경우 네트워크 구성을 모니터하여 시스템 네트워크 환경의 변경 사항에 맞게 조정합니다. 수정된 프로파일을 사용할 경우 네트워크 구성이 인스턴스화되지만 모니터되지는 않습니다.
- NCP의 여러 등록 정보 값으로 구성된 정책이 프로파일에서 네트워크 구성을 관리하는 방식을 제어합니다.
- NCP 등록 정보의 변경 사항은 네트워크 구성을 관리하는 프로파일 정책의 일부가 되는 새 등록 정보 값으로 즉시 구현됩니다.

주 – Oracle Solaris 11 11 Express 릴리스에서 업그레이드된 시스템의 경우 업그레이드 전에 작동한 네트워크 구성이 업그레이드 후의 활성 프로파일이 됩니다. 이전 구성이 `dladm` 및 `ipadm` 명령으로 생성된 경우 해당 구성이 시스템에서 활성화되는 `DefaultFixed` 프로파일을 생성합니다. 그렇지 않으면 구성이 시스템의 네트워크 구성을 관리하는 `Automatic` 프로파일이 됩니다.

프로파일 및 구성 도구

프로파일 사용자 정의에 사용할 도구는 활성 프로파일에 따라 달라집니다. 활성 프로파일이 `Automatic` 등의 반응적 프로파일인 경우 `netcfg` 및 `netadm` 명령을 사용하여 프로파일을 구성하고 모니터합니다. 활성 프로파일이 `DefaultFixed` 등의 수정된 프로파일인 경우 `dladm` 및 `ipadm` 명령을 사용합니다.

`dladm` 및 `ipadm` 명령은 활성 프로파일에서만 적용됩니다. 따라서 두 명령을 사용하기 전에 다음을 확인해야 합니다.

- 적절한 명령을 사용하여 올바른 대상 프로파일을 변경하도록 활성화된 프로파일을 확인합니다.
- 명령을 사용한 후 예기치 않은 구성 동작이 발생하지 않도록 대상 프로파일이 반응적 프로파일인지 또는 수정된 프로파일인지 확인합니다. 반응적 프로파일은 수정된 프로파일과 다른 방식으로 네트워크 구성을 관리합니다. 따라서 변경 사항을 구현할 때도 두 프로파일의 동작이 서로 다릅니다.

주 – `dladm` 및 `ipadm` 명령의 `-t` 옵션을 사용하여 임시 설정을 만드는 작업은 수정된 프로파일에서만 적용됩니다. 반응적 프로파일에서는 이 옵션이 지원되지 않습니다.

다음 두 절차에 따라 프로파일에서 `dladm` 및 `ipadm` 명령을 올바르게 사용합니다.

▼ 네트워크 관리 모드를 결정하는 방법

시스템의 네트워크 관리 모드는 `Automatic`과 같은 반응적 NCP가 시스템에서 활성 NCP일 경우 자동입니다. 이 절차를 사용하여 네트워크 구성을 수행하기 전에 네트워크 관리 모드에 대해 알 수 있습니다. 이 절차를 통해 올바른 명령을 사용하여 적절한 프로파일에서 구성을 구현하고 있는지 확인할 수 있습니다.

1 시스템의 프로파일을 나열합니다.

```
# netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         Automatic      online         active
ncu:phys    net0           online         interface/link is up
```

```
ncu:ip      net0          online         interface/link is up
ncu:phys    net1          online         interface/link is up
ncu:ip      net1          offline*       waiting for IP address to be set
ncp         testcfg       disabled       disabled by administrator
loc         Automatic     offline        conditions for activation are unmet
loc         NoNet         offline        conditions for activation are unmet
loc         Lab           online         active
loc         User          disabled       disabled by administrator
```

출력은 다음과 같은 두 가지 정보를 제공합니다.

- `netadm list` 명령은 네트워크 관리 모드가 자동인 경우에만 지원됩니다. 그러므로 프로파일 목록의 생성은 네트워크 관리가 자동 모드임을 나타냅니다. 그렇지 않으면 `netadm list` 명령은 시스템에서 DefaultFixed 프로파일이 활성임을 나타내는 다음과 같은 메시지를 대신 생성했을 것입니다.

  ```
  netadm: DefaultFixed NCP is enabled; automatic network management is not available.
  'netadm list' is only supported when automatic network management is active.
  ```

- 생성된 경우, 프로파일 목록은 또한 NCP의 `online` 상태를 통해 어떤 특정 반응적 NCP가 사용으로 설정되었는지도 식별합니다. 샘플 출력에서 `Automatic` NCP가 유일한 기존 반응적 NCP로 나열되어 있습니다. 다른 사용자가 만든 NCP가 시스템에 존재하는 경우 이러한 NCP도 목록에 포함되었을 것입니다.

2 사용하려는 구성 도구에 대해 적절한 프로파일이 활성 상태가 되도록 합니다.

예를 들어 `dladm` 및 `ipadm` 명령은 `DefaulfFixed` 프로파일에서만 사용할 수 있습니다. 그러나 `netcfg` 명령은 네트워크 관리가 자동 모드인 `Automatic`과 같은 반응적 프로파일에서만 사용할 수 있습니다.

선택한 구성 도구로 수정하려고 하는 등록 정보가 활성이 아닌 프로파일의 경우 다음 단계를 계속하여 적절한 프로파일을 사용으로 설정하십시오. 그렇지 않으면 도구를 사용하여 네트워크를 구성하기 시작할 수도 있습니다.

예를 들어 네트워크 관리를 자동 모드로 설정하지 않고 `dladm` 및 `ipadm` 명령줄을 사용하여 수동으로 데이터 링크와 인터페이스를 구성하는 것을 선호할 수도 있습니다. 1단계의 출력은 `Automatic` 프로파일이 사용으로 설정되었음을 보여줍니다. 네트워크 구성에 명령줄을 사용하려면 `DefaultFixed` 프로파일을 사용으로 설정해야 합니다.

3 다른 프로파일을 구성하려면 다음을 입력하여 해당 프로파일을 사용으로 설정합니다.

`# netadm enable -p ncp` *profile-name*

예를 들면 다음과 같습니다.

```
# netadm enable -p ncp defaultfixed
```

또한 네트워크 관리가 자동 모드인 경우 동일한 명령 구문을 사용하고 다른 반응적 NCP를 사용할 수도 있습니다. 1단계의 샘플 출력에서 `Automatic` 대신 사용자가 만든 NCP `testcfg`를 활성화하려고 한다고 가정합니다. 그러면 다음을 입력합니다.

```
# netadm enable -p ncp testcfg
```

주의 – 이 명령은 활성 프로파일을 전환합니다. 활성 프로파일을 전환하면 기존 네트워크 구성이 제거되고 새 구성이 생성됩니다. 이전 활성 NCP에서 구현된 지속 변경 사항은 새 활성 NCP에서 제외됩니다.

다음 단계

다음 장에서는 다양한 유형의 데이터 링크 및 인터페이스 구성을 수행하는 데 사용할 수 있는 절차에 대해 설명합니다.

- 데이터 링크를 구성하려면 8 장, “데이터 링크 구성 및 관리”를 참조하십시오.
- IP 인터페이스를 구성하려면 9 장, “IP 인터페이스 구성”을 참조하십시오.
- 무선 인터페이스를 구성하려면 10 장, “Oracle Solaris에서 무선 인터페이스 통신 구성”을 참조하십시오.
- 브릿지를 구성하려면 11 장, “브릿지 관리”를 참조하십시오.
- 링크 통합을 구성하려면 12 장, “링크 통합 관리”를 참조하십시오.
- VLAN을 구성하려면 13 장, “VLAN 관리”를 참조하십시오.
- IPMP 그룹을 구성하려면 14 장, “IPMP 소개” 및 15 장, “IPMP 관리”를 참조하십시오.
- LLDP(Link Layer Discovery Protocol)를 구성하려면 16 장, “LLDP를 사용하여 네트워크 연결 정보 교환”을 참조하십시오.

◆ ◆ ◆ 8 장

8 데이터 링크 구성 및 관리

이 장에서는 dladm 명령과 이 명령을 사용하여 데이터 링크를 구성하는 방법에 대해 설명합니다.

데이터 링크 구성(작업)

다음 표에서는 dladm 명령을 사용하여 수행할 수 있는 다양한 데이터 링크 구성 작업을 보여줍니다. 작업을 완료하는 단계별 절차에 대한 링크도 제공됩니다.

표 8–1 기본 데이터 링크 구성 수행(작업 맵)

작업	설명	수행 방법
데이터 링크의 이름을 바꿉니다.	하드웨어 기반 이름을 사용하는 대신 데이터 링크 이름을 사용자 정의합니다.	143 페이지 "데이터 링크의 이름을 바꾸는 방법"
데이터 링크의 물리적 속성을 표시합니다.	매체 유형, 연결된 장치 인스턴스 및 기타 정보를 비롯하여 데이터 링크의 기반이 되는 물리적 정보를 나열합니다.	144 페이지 "데이터 링크의 물리적 속성에 대한 정보를 표시하는 방법"
데이터 링크의 상태를 표시합니다.	데이터 링크의 상태 정보를 나열합니다.	145 페이지 "데이터 링크 정보를 표시하는 방법"
데이터 링크를 제거합니다.	더 이상 사용하지 않는 NIC와 연결된 링크 구성을 제거합니다.	146 페이지 "데이터 링크를 삭제하는 방법"

표 8–2 데이터 링크 등록 정보 설정(작업 맵)

작업	설명	수행 방법
MTU 크기를 수정합니다.	점보 프레임을 처리할 패킷 전송의 MTU 크기를 늘립니다.	148 페이지 "점보 프레임 지원을 사용으로 설정하는 방법"

표 8–2 데이터 링크 등록 정보 설정(작업 맵) (계속)

작업	설명	수행 방법
링크 속도를 수정합니다.	상위 링크 속도를 해제하고 이전 시스템과의 통신을 허용하도록 하위 링크 속도만 알립니다.	150 페이지 "링크 속도 매개변수를 변경하는 방법"
링크 등록 정보에 대한 정보를 표시합니다.	링크 등록 정보와 현재 구성, 그리고 이더넷 매개변수 설정을 나열합니다.	150 페이지 "데이터 링크 등록 정보에 대한 상태 정보를 가져오는 방법"
DMA 바인딩을 사용하도록 드라이버를 구성합니다.	전송 도중 드라이버가 DMA 바인딩과 bcopy 함수를 전환하게 하는 임계값을 설정합니다.	152 페이지 "직접 메모리 액세스 바인딩을 사용하도록 e1000g 드라이버를 설정하는 방법"
인터럽트 속도를 설정합니다.	자동으로 정의되는 속도 대신 드라이버가 인터럽트를 전달하는 속도를 수동으로 정의합니다.	153 페이지 "인터럽트 속도를 수동으로 설정하는 방법"
NIC(네트워크 인터페이스 카드)를 교체합니다.	DR(동적 재구성) 도중 시스템의 NIC를 변경합니다.	155 페이지 "동적 재구성을 사용하여 네트워크 인터페이스 카드를 교체하는 방법"
링크별 autopush 등록 정보를 설정합니다.	데이터 링크 위에 푸시되도록 STREAMS 모듈을 구성합니다.	158 페이지 "데이터 링크에 STREAMS 모듈을 설정하는 방법"

dladm 명령

GLDv3 드라이버 구성 프레임워크의 전체 구현 후 dladm 명령이 나중에 확장 기능을 획득했습니다. 이 프레임워크는 다음과 같이 NIC 드라이버의 구성을 향상시킵니다.

- 네트워크 드라이버 등록 정보를 구성하려면 단일 명령 인터페이스인 dladm 명령만 있으면 됩니다.
- 등록 정보에 관계없이 동일한 구문이 사용됩니다. dladm *subcommand properties datalink* .
- dladm 명령 사용은 드라이버의 공용 및 개인 등록 정보에 모두 적용됩니다.
- 특정 드라이버에서 dladm 명령을 사용하는 경우 비슷한 유형의 다른 NIC 네트워크 연결은 중단되지 않습니다. 따라서 데이터 링크 등록 정보를 동적으로 구성할 수 있습니다.
- 데이터 링크 구성 설정은 dladm 저장소에 저장되며 시스템을 재부트한 후에도 지속됩니다.

데이터 링크를 구성할 때 위에 나열된 장점을 활용하려면 ndd 명령과 같은 이전 릴리스의 사용자 정의 도구 대신 dladm을 구성 도구로 사용해야 합니다.

데이터 링크를 관리하려면 다음 dladm 하위 명령을 사용합니다.

- `dladm rename-link`는 데이터 링크의 이름을 변경합니다.
- `dladm show-link`는 시스템의 기존 데이터 링크를 표시합니다.
- `dladm show-phys`는 데이터 링크의 물리적 속성을 표시합니다.
- `dladm delete-phys`는 데이터 링크를 삭제합니다.
- `dladm show-linkprop`은 데이터 링크와 연결된 등록 정보를 표시합니다.
- `dladm set-linkprop`은 지정한 데이터 링크 등록 정보를 설정합니다.
- `dladm reset-linkprop`은 등록 정보를 기본 설정으로 복원합니다.
- `dladm show-ether`는 데이터 링크의 이더넷 매개변수 설정을 표시합니다.

`dladm` 명령을 사용하여 다음과 같은 기타 유형의 링크 관리도 수행합니다.

- 브릿지 구성. 11 장, “브릿지 관리”를 참조하십시오.
- 링크 통합 구성. 12 장, “링크 통합 관리”를 참조하십시오.
- VLAN 구성. 13 장, “VLAN 관리”를 참조하십시오.
- 터널 구성. **Oracle Solaris 관리: IP 서비스**의 6 장, “IP 터널 구성”을 참조하십시오.

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

다음 절차에서는 `dladm` 명령을 사용하여 데이터 링크를 구성하는 방법을 보여줍니다. 대부분의 경우 데이터 링크 구성은 해당 링크의 IP 인터페이스 구성에 포함됩니다. 따라서 해당하는 경우 이 절차에 `ipadm` 명령을 사용한 IP 인터페이스 구성 단계가 포함됩니다. 하지만 IP 인터페이스 구성과 `ipadm` 명령은 9 장, “IP 인터페이스 구성”에서 자세히 설명합니다.

▼ 데이터 링크의 이름을 바꾸는 방법

데이터 링크 이름을 사용자 정의 이름으로 변경하려는 경우 이 절차를 사용합니다. 예를 들어, 업그레이드된 시스템의 일부 데이터 링크에서 레거시 하드웨어 기반 이름이 유지되었을 수 있으며 해당 이름을 일반 이름으로 변경하려고 합니다.

시작하기 전에 링크 이름 변경 시 영향을 받을 수 있는 관련 구성에서 수행해야 하는 기타 단계를 조사하고 준비했는지 확인합니다. 자세한 내용은 26 페이지 “업그레이드된 시스템의 링크 이름”을 참조하십시오.

1 관리자로 전환합니다.

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

2 IP 인터페이스가 데이터 링크에 구성된 경우 IP 인터페이스를 제거합니다.

```
# ipadm delete-ip interface
```

3 링크의 현재 링크 이름을 변경합니다.

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

old-linkname 데이터 링크의 현재 이름을 나타냅니다. 기본적으로 링크 이름은 하드웨어 기반입니다(예: `bge0`).

new-linkname 데이터 링크에 할당하려는 이름을 나타냅니다. 링크 이름 지정 규칙은 28 페이지 "유효한 링크 이름 규칙"을 참조하십시오. 데이터 링크 이름 바꾸기에 대한 자세한 내용은 26 페이지 "업그레이드된 시스템의 링크 이름"을 참조하십시오.

시스템 재부트 후 새 링크 이름을 유지하지 않으려는 경우 하위 명령 바로 뒤에 `-t` 옵션을 사용합니다. 이 옵션은 일시적으로 링크의 이름을 바꿉니다. 시스템을 재부트하면 원래 링크 이름으로 돌아갑니다.

주 – `dladm rename-link`를 사용하여 한 데이터 링크에서 다른 데이터 링크로 링크 구성을 전송할 수 있습니다. 예를 들어, 155 페이지 "동적 재구성을 사용하여 네트워크 인터페이스 카드를 교체하는 방법"을 참조하십시오. 이런 용도로 링크의 이름을 바꾸는 경우 구성을 상속하는 링크에 기존 구성이 없는지 확인합니다. 기존 구성이 있으면 전송에 실패합니다.

예 8–1 시스템의 주 네트워크 인터페이스 변경

다음 예에서는 데이터 링크의 이름을 바꿔서 시스템의 주 네트워크 인터페이스를 두번째 NIC로 전환하는 방법을 보여줍니다. 시스템의 주 네트워크 인터페이스는 `net0`이고 데이터 링크의 일반 이름은 `e1000g0`입니다. 이 주 네트워크 인터페이스는 기본 인터페이스로 `e1000g0` 사용에서 `nge0`으로 전환됩니다. 새 부트 환경을 만드는 절차의 일부로 이 예를 사용할 수 있습니다.

```
# dladm show-phys
LINK    MEDIA       STATE    SPEED    DUPLEX    DEVICE
net0    Ethernet    up       1000     full      e1000g0
net1    Ethernet    up       1000     full      nge0

# dladm rename-link net0 oldnet0
# dladm rename-link net1 net0

# dladm show-phys
LINK       MEDIA       STATE    SPEED    DUPLEX    DEVICE
oldnet0    Ethernet    up       1000     full      e1000g0
net0       Ethernet    up       1000     full      nge0
```

▼ 데이터 링크의 물리적 속성에 대한 정보를 표시하는 방법

이 절차에서는 시스템 데이터 링크의 물리적 속성에 대한 정보를 표시하는 단계를 보여줍니다.

1 **관리자로 전환합니다.**

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

2 **현재 시스템에 있는 데이터 링크의 물리적 속성에 대한 정보를 표시합니다.**

```
# dladm show-phys
```

이 명령과 함께 -P를 사용하여 각 링크의 플래그 상태를 표시할 수도 있습니다. 연결된 하드웨어가 제거된 경우 데이터 링크를 사용할 수 없게 됩니다. -P 옵션을 사용하지 않으면 이 명령은 사용 가능한 데이터 링크만 표시합니다.

데이터 링크의 /devices 경로를 보려면 -v 옵션을 사용합니다.

예 8–2 사용 가능한 데이터 링크 표시

다음 예에서 -P 옵션에는 사용할 수 없는 링크가 표시되는 FLAGS 열이 포함됩니다. 데이터 링크 net0의 r 플래그는 링크(nge)와 연결된 하드웨어가 제거되었음을 나타냅니다.

```
# dladm show-phys
LINK        MEDIA          STATE     SPEED     DUPLEX     DEVICE
net0        Ethernet       up        100Mb     full       e1000g0
net1        Infiniband     down      0Mb       --         ibd0
net3        Ethernet       up        100Mb     full       bge0
net4        Ethernet       --        0Mb       --         nge0
```

다음 예에서는 -L 옵션을 사용할 때 표시되는 링크 및 해당 물리적 위치를 보여줍니다.

```
# dladm show-phys -L
LINK     DEVICE     LOCATION
net0     bge0       MB
net2     ibp0       MB/RISER0/PCIE0/PORT1
net3     ibp1       MB/RISER0/PCIE0/PORT2
net4     eoib2      MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

▼ 데이터 링크 정보를 표시하는 방법

이 절차에서는 사용 가능한 링크의 상태를 표시합니다.

1 **관리자로 전환합니다.**

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

2 **링크 정보를 표시합니다.**

```
# dladm show-link
```

예 8–3 사용 가능한 링크 표시

다음 예에서는 시스템의 지속적이고 사용 가능한 링크를 보여줍니다.

```
# dladm show-link -P
LINK        CLASS     BRIDGE     OVER
net0        phys      --         --
net1        phys      --         --
net2        phys      --         --
```

-P 옵션은 지속적이지만 사용할 수 없는 기존 링크도 표시합니다. 링크가 일시적으로 삭제된 경우 지속 링크를 사용할 수 없게 됩니다. 연결된 하드웨어가 제거된 경우 링크도 사용할 수 없게 됩니다.

▼ 데이터 링크를 삭제하는 방법

이 절차에서는 NIC와 연결된 링크 구성을 삭제합니다. 교체하려는 의도 없이 NIC를 분리하는 경우 해당 NIC와 연결된 링크 구성을 삭제할 수 있습니다. 이 절차를 완료한 후 링크 이름을 재사용할 수 있습니다.

1 관리자로 전환합니다.

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

2 하드웨어가 제거된 링크를 포함하여 시스템의 데이터 링크를 표시합니다.

제거된 하드웨어에 대한 정보를 포함하려면 -P 옵션을 사용합니다.

```
# dladm show-phys
```

3 교체하지 않으려는 제거된 하드웨어의 링크 구성을 제거합니다.

```
# dladm delete-phys link
```

예 8–4 데이터 링크 삭제

다음 예에서 net2의 r 플래그는 링크에 연결된 하드웨어(e1000g0)가 제거되었음을 나타냅니다. 따라서 net2 링크를 제거한 후 새 데이터 링크에 이름을 재지정할 수도 있습니다.

```
# dladm show-phys -P
LINK        DEVICE     MEDIA        FLAGS
net0        nge0       Ethernet     -----
net1        bge0       Ethernet     -----
net2        e1000g0    Ethernet     r----

# dladm delete-phys net2
```

데이터 링크 등록 정보 설정

기본 데이터 링크 구성을 수행하는 것은 물론 dladm 명령을 사용하여 데이터 링크 등록 정보를 설정하고 네트워크 요구에 따라 사용자 정의할 수도 있습니다.

주 – 링크의 네트워크 드라이버가 GLDv3 프레임워크(예: e1000g)로 변환된 경우 dladm 명령을 사용하여 데이터 링크 등록 정보를 사용자 정의할 수 있습니다. 특정 드라이버가 이 기능을 지원하는지 여부를 확인하려면 드라이버의 매뉴얼 페이지를 참조하십시오.

데이터 링크 등록 정보 개요

사용자 정의할 수 있는 데이터 링크 등록 정보는 특정 NIC 드라이버가 지원하는 등록 정보에 따라 달라집니다. dladm 명령을 사용하여 구성할 수 있는 데이터 링크 등록 정보는 다음 두 가지 범주 중 하나에 속합니다.

- **공용 등록 정보**: 링크 속도, 이더넷의 자동 협상 또는 모든 데이터 링크 드라이버에 적용할 수 있는 MTU 크기 등 지정된 매체 유형의 드라이버에 적용할 수 있습니다.
- **개인 등록 정보**: 지정된 매체 유형에 대한 NIC 드라이버의 특정 부분에 따라 고유합니다. 이러한 등록 정보는 드라이버와 연결된 하드웨어 또는 드라이버 구현 자체의 세부 정보(예: 디버깅 관련 조정 가능 항목)와 긴밀한 관계가 있기 때문에 이 부분에만 관련된 것일 수 있습니다.

일반적으로 링크 등록 정보에는 기본 설정이 있습니다. 하지만 특정 네트워킹 시나리오에서는 데이터 링크의 특정 등록 정보 설정을 변경해야 할 수도 있습니다. 이러한 등록 정보 설정은 공용 또는 개인 등록 정보일 수 있습니다. 예를 들어, NIC가 자동 협상을 제대로 수행하지 않는 이전 스위치와 통신 중일 수 있습니다. 또는 스위치가 점보 프레임을 지원하도록 구성되었을 수도 있습니다. 또는 패킷 전송 또는 패킷 수신을 규제하는 드라이버별 등록 정보를 지정된 드라이버에 맞게 수정해야 할 수도 있습니다. Oracle Solaris에서는 이러한 모든 설정을 단일 관리 도구 dladm으로 재설정할 수 있습니다.

dladm 명령을 사용하여 데이터 링크 등록 정보 설정

다음 절에서는 특정 데이터 링크 등록 정보를 설정하는 절차를 예와 함께 제공합니다. 선택한 등록 정보는 공용이며 모든 NIC 드라이버에 적용됩니다. 드라이버별 데이터 링크 등록 정보는 별도의 절에서 설명합니다. 이 절 뒤에는 e1000g 드라이버의 선택한 개인 등록 정보를 구성하는 절차가 나와 있습니다.

▼ 점보 프레임 지원을 사용으로 설정하는 방법

네트워크 설정에서 점보 프레임 지원을 사용으로 설정하는 것은 대부분의 네트워크 시나리오에서 공통된 작업입니다. 점프 프레임을 지원하려면 데이터 링크의 MTU(최대 전송 단위) 크기를 늘려야 합니다. 다음 절차에는 사용자 정의 이름을 사용한 데이터 링크 식별이 포함됩니다. 네트워크 구성의 사용자 정의 이름 및 해당 사용에 대한 개요는 20 페이지 "Oracle Solaris의 네트워크 스택"을 참조하십시오.

1 관리자로 전환합니다.

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

2 MTU 크기를 재설정해야 하는 특정 이더넷 장치를 식별하려면 시스템의 링크를 표시합니다.

```
# dladm show-phys
```

특히 네트워크 구성에서 데이터 링크에 사용자 정의 이름을 사용하는 경우 이 단계를 수행합니다. 사용자 정의 이름을 사용하는 경우 데이터 링크가 더 이상 하드웨어 기반 이름으로 식별되지 않습니다. 예를 들어, 이더넷 장치는 `bge0`입니다. 하지만 장치를 통한 데이터 링크의 이름이 `net0`으로 바뀝니다. 따라서 `net0`의 MTU 크기를 구성해야 합니다. 사용자 정의 이름을 사용하는 데이터 링크의 구성 작업 예는 162 페이지 "IP 인터페이스 구성(작업)"을 참조하십시오.

3 (옵션) 데이터 링크의 현재 MTU 크기 및 기타 등록 정보를 표시합니다.

- **데이터 링크의 특정 등록 정보를 표시하려면 다음 구문을 사용합니다.**

  dladm show-linkprop -p *property datalink*

  이 명령은 지정한 등록 정보의 설정을 표시합니다.

- **데이터 링크의 선택한 등록 정보를 여러 개 표시하려면 다음 구문을 사용합니다.**

  # dladm show-link *datalink*

  이 명령은 MTU 크기를 비롯한 데이터 링크 정보를 표시합니다.

4 IP 인터페이스가 데이터 링크에 구성된 경우 IP 인터페이스를 제거합니다.

 # ipadm delete-ip *interface*

5 링크의 MTU 크기를 점보 프레임 설정인 9000으로 변경합니다.

 # dladm set-linkprop -p mtu=9000 *datalink*

6 IP 인터페이스를 만듭니다.

 # ipadm create-ip *interface*

7 IP 인터페이스를 구성합니다.

```
# ipadm create-addr -T addr-type [-a address] addrobj
```

ipadm 명령에 대한 자세한 내용은 ipadm(1M)을 참조하십시오.

8 (옵션) 3단계의 명령 구문 중 하나를 사용하여 인터페이스가 새 MTU를 사용하는지 확인합니다.

```
# dladm show-linkprop -p mtu datalink
```

9 (옵션) 링크의 현재 이더넷 설정을 표시합니다.

```
# dladm show-ether datalink
```

예 8–5 점보 프레임 지원 사용

점보 프레임 지원을 사용으로 설정하는 다음 예는 아래의 시나리오를 기반으로 합니다.

- 시스템에 두 개의 bge NIC(bge0 및 bge1)가 있습니다.
- bge0 장치는 주 인터페이스로 사용되고 bge1 장치는 테스트 용도로 사용됩니다.
- bge1에서 점보 프레임 지원을 사용으로 설정하지만 주 인터페이스의 기본 MTU 크기를 유지하려고 합니다.
- 네트워크 구성에서 데이터 링크에 사용자 정의 이름을 사용합니다. bge0의 링크 이름은 net0입니다. bge1의 링크 이름은 net1입니다.

```
# dladm show-phys
LINK        MEDIA        STATE     SPEED     DUPLEX     DEVICE
net0        ether        up        100Mb     full       bge0
net1        ether        up        100Mb     full       bge1
net2        ether        up        100Mb     full       nge3

# dladm show-linkprop -p mtu net1
LINK     PROPERTY     VALUE     DEFAULT     POSSIBLE
net1     mtu          1500      1500        --

# ipadm delete-ip net1
# dladm set-linkprop -p mtu=9000 net1
# ipadm create-ip net1
# ipadm create-addr -T static -a 10.10.1.2/35 net1/v4

# dladm show-link web1
LINK     CLASS     MTU      STATE     BRIDGE     OVER
web1     phys      9000     up        --         --
```

이제 MTU 설정이 9000입니다. 이 예에서 dladm 명령을 사용하여 net1의 MTU 크기를 직접 변경할 수 있었습니다. ndd 명령을 사용하는 이전 방법에서는 net0도 삭제해야 했으므로 주 인터페이스의 작업을 불필요하게 중단했었습니다.

▼ 링크 속도 매개변수를 변경하는 방법

대부분의 네트워크 설정은 다양한 속도 기능의 시스템 조합으로 구성됩니다. 예를 들어, 통신을 허용하기 위해 이전 시스템과 최신 시스템 간에 알려진 속도를 하위 설정으로 변경해야 할 수 있습니다. 기본적으로 NIC 카드의 모든 속도와 이중 기능이 알려집니다. 이 절차에서는 기가비트 기능을 해제하고 메가비트 기능만 알리는 방법을 보여줍니다.

1. **관리자로 전환합니다.**

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

2. **(옵션) 수정하려는 등록 정보의 현재 상태를 표시합니다.**

   # dladm show-linkprop -p *property datalink*

3. **하위 속도 기능을 알리려면 상위 속도 기능을 해제하여 알려지지 않도록 합니다.**

   # dladm set-linkprop -p *property=value1 datalink*

예 8–6 NIC 기가비트 기능 알림 사용 안함

이 예에서는 net1 링크가 기가비트 기능을 알리지 않도록 하는 방법을 보여줍니다.

```
# dladm show-linkprop -p adv_1000fdx_cap net1
LINK     PROPERTY             VALUE     DEFAULT     POSSIBLE
net1     adv_1000fdx_cap      1         --          1,0

# dladm show-linkprop -p adv_1000hdx_cap web1
LINK     PROPERTY             VALUE     DEFAULT     POSSIBLE
net1     adv_1000hdx_cap      1         --          1,0
```

링크의 기가비트 기능을 알리는 등록 정보는 adv_1000fdx_cap와 adv_1000hdx_cap입니다. 이러한 등록 정보가 알려지지 않게 하려면 다음 명령을 입력합니다.

```
# dladm set-linkprop -p adv_1000fdx_cap=0 net1
# dladm set-linkprop -p adv_1000hdx_cap=0 net1
```

이더넷 매개변수 설정을 나열하면 다음 출력 결과가 표시됩니다.

```
# dladm show-ether net1
LINK     PTYPE       STATE    AUTO  SPEED-DUPLEX            PAUSE
net1     current     up       yes   1G-f                    both
```

▼ 데이터 링크 등록 정보에 대한 상태 정보를 가져오는 방법

이더넷 매개변수 설정 또는 링크 등록 정보를 표시하여 데이터 링크 등록 정보에 대한 정보를 가져올 수 있습니다.

1 **관리자로 전환합니다.**

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

2 **이더넷 매개변수 설정에 대한 정보를 가져오려면 다음 명령을 사용합니다.**

 # dladm show-ether [-x] *datalink*

여기서 -x 옵션에는 링크에 대한 추가 매개변수 설정이 포함됩니다. -x 옵션을 사용하지 않으면 현재 매개변수 설정만 표시됩니다.

3 **링크의 모든 등록 정보에 대한 정보를 가져오려면 다음 명령을 사용합니다.**

 # dladm show-linkprop *datalink*

예 8–7 이더넷 매개변수 설정 표시

이 예에서는 지정한 링크에 대한 매개변수 정보의 확장 목록을 표시합니다.

```
# dladm show-ether -x net1
LINK     PTYPE       STATE    AUTO  SPEED-DUPLEX              PAUSE
net1     current     up       yes   1G-f                      both
--       capable     --       yes   1G-fh,100M-fh,10M-fh      both
--       adv         --       yes   100M-fh,10M-fh           both
--       peeradv     --       yes   100M-f,10M-f             both
```

-x 옵션을 사용하면 이 명령은 지정한 링크의 내장 기능뿐 아니라 호스트와 링크 파트너 간에 현재 알려진 기능도 표시합니다. 다음 정보가 표시됩니다.

- 현재 이더넷 장치 상태의 경우 링크가 전이중에서 초당 1기가비트 속도로 작동하고 있습니다. 자동 협상 기능이 사용으로 설정되었으며 호스트와 링크 파트너가 일시 중지 프레임을 보내고 받을 수 있는 양방향 흐름 제어가 있습니다.
- 현재 설정에 관계없이 이더넷 장치의 기능이 나열됩니다. 협상 유형을 자동으로 설정할 수 있고, 장치가 전이중 및 반이중에서 초당 1기가비트, 초당 100메가비트 및 초당 10메가비트 속도를 지원할 수 있습니다. 마찬가지로, 호스트와 링크 파트너 간에 일시 중지 프레임을 양방향으로 보내거나 받을 수 있습니다.
- net1의 기능은 자동 협상, 속도-이중 및 일시 중지 프레임의 흐름 제어로 알려집니다.
- 이와 유사하게, net1의 링크 또는 피어 파트너도 자동 협상, 속도-이중 및 일시 중지 프레임의 흐름 제어를 알립니다.

예 8–8 링크 등록 정보 표시

이 예에서는 링크의 모든 등록 정보를 나열하는 방법을 보여줍니다. 특정 등록 정보만 표시하려는 경우 -p 옵션을 모니터하려는 특정 등록 정보와 함께 사용합니다.

```
# dladm show-linkprop net1
LINK     PROPERTY             VALUE       DEFAULT     POSSIBLE
```

```
net1      speed                1000        --          --
net1      autopush             --          --          --
net1      zone                 --          --          --
net1      duplex               half        --          half,full
net1      state                unknown     up          up,down
net1      adv_autoneg_cap      1           1           1,0
net1      mtu                  1500        1500        --
net1      flowctrl             no          bi          no,tx,rx,bi
net1      adv_1000fdx_cap      1           1           1,0
net1      en_1000fdx_cap       1           1           1,0
net1      adv_1000hdx_cap      1           1           1,0
net1      en_1000hdx_cap       1           1           1,0
net1      adv_100fdx_cap       0           0           1,0
net1      en_100fdx_cap        0           0           1,0
net1      adv_100hdx_cap       0           0           1,0
net1      en_100hdx_cap        0           0           1,0
net1      adv_10fdx_cap        0           0           1,0
net1      en_10fdx_cap         0           0           1,0
net1      adv_10hdx_cap        0           0           1,0
net1      en_10hdx_cap         0           0           1,0
```

링크의 속도 및 이중 기능에 대한 설정은 `en_*_cap` 레이블이 있는 사용 속도 등록 정보에서 수동으로 구성됩니다. 예를 들어, `en_1000fdx_cap`는 기가비트 전이중 기능의 등록 정보이고 `en_100hdx_cap`는 100메가비트 반이중 기능의 등록 정보입니다. 이러한 속도 사용 등록 정보의 설정은 `adv_*_cap` 레이블이 지정된 알려진 해당 속도 등록 정보(예: `adv_1000fdx_cap` 및 `adv_100hdx_cap`)를 통해 호스트와 링크 파트너 간에 알려집니다.

일반적으로 지정된 사용 속도 등록 정보와 알려진 해당 등록 정보의 설정은 같습니다. 하지만 NIC가 전원 관리 등의 고급 기능을 지원하는 경우 해당 기능이 호스트와 링크 파트너 간에 실제로 알려지는 비트에 제한을 설정할 수도 있습니다. 예를 들어, 전원 관리를 사용할 경우 `adv_*_cap` 등록 정보의 설정이 `en_*_cap` 등록 정보 설정의 일부일 뿐입니다. 사용 및 알려진 속도 등록 정보에 대한 자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 직접 메모리 액세스 바인딩을 사용하도록 e1000g 드라이버를 설정하는 방법

이 절차와 다음 절차에서는 개인 등록 정보를 구성하는 방법을 보여줍니다. 두 절차는 모두 `e1000g` 드라이버와 관련된 등록 정보에 적용됩니다. 하지만 일반 단계를 사용하여 다른 NIC 드라이버의 개인 등록 정보도 구성할 수 있습니다.

파일 전송 등의 대량 트래픽이 있을 경우 대체로 네트워크에서 큰 패킷을 협상해야 합니다. 이 경우 패킷 조각 크기에 대해 임계값이 정의되는 DMA 바인딩을 자동으로 사용하도록 구성하면 `e1000g` 드라이버의 성능을 향상시킬 수 있습니다. 조각 크기가 임계값을 초과할 경우 DMA 바인딩이 전송에 사용됩니다. 조각 크기가 임계값 이내인 경우 조각 데이터가 미리 할당된 전송 버퍼에 복사되는 `bcopy` 모드가 사용됩니다.

임계값을 설정하려면 다음 단계를 수행합니다.

1 **관리자로 전환합니다.**

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

2 **_tx_bcopy_threshold 등록 정보에 적절한 설정을 지정합니다.**

```
# dladm set-linkprop -p _tx_bcopy_threshold=value e1000g-datalink
```

이 등록 정보에 유효한 임계값 설정 범위는 60에서 2048 사이입니다.

주 – 공용 등록 정보 구성과 마찬가지로 개인 등록 정보 설정을 수정하려면 먼저 인터페이스를 연결 취소해야 합니다.

---

3 **(옵션) 새 임계값 설정을 확인합니다.**

```
# dladm show-linkprop -p _tx_bcopy_threshold e1000g-datalink
```

▼ 인터럽트 속도를 수동으로 설정하는 방법

인터럽트가 e1000g 드라이버에 의해 전달되는 속도를 규제하는 매개변수는 네트워크 및 시스템 성능에도 영향을 줍니다. 일반적으로 네트워크 패킷은 각 패킷에 대해 인터럽트를 생성하여 스택의 상위 계층으로 전달됩니다. 기본적으로 인터럽트 속도는 커널의 GLD 계층에서 자동으로 조정됩니다. 하지만 이 모드가 모든 네트워크 트래픽 상태에서 바람직한 것은 아닙니다. 이 문제에 대한 자세한 내용은 1996년 USENIX 기술 회의에서 발표된 문서(http://www.stanford.edu/class/cs240/readings/mogul.pdf)를 참조하십시오. 따라서 성능 향상을 위해 수동으로 인터럽트 속도를 설정해야 하는 경우도 있습니다.

인터럽트 속도를 정의하려면 다음 매개변수를 설정합니다.

- _intr_throttling_rate는 네트워크 트래픽 상태에 관계없이 인터럽트 검증 간의 지연을 결정합니다.
- _intr_adaptive는 인터럽트 제한 속도의 자동 조정을 사용으로 설정할지 여부를 결정합니다. 기본적으로 이 매개변수는 사용으로 설정됩니다.

1 **관리자로 전환합니다.**

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

2 **필요한 경우 드라이버 등록 정보를 수정하려는 장치를 식별합니다.**

```
# dladm show-phys
```

3 **인터럽트 제한 속도의 자동 조정을 사용 안함으로 설정합니다.**

```
# dladm set-linkprop -p _intr_adaptive=0 e1000g-datalink
```

주 – 인터럽트 제한 속도의 자동 조정을 사용으로 설정할 경우 `_intr_throttling_rate` 매개변수의 기존 설정이 모두 무시됩니다.

4 **데이터 링크에 구성된 IP 인터페이스를 제거합니다.**

5 **최소 인터럽트 간 레벨의 설정을 지정합니다.**

```
# dladm set-linkprop -p _intr_throttling_rate=value e1000g-datalink
```

주 – `_intr_throttling_rate` 매개변수의 기본 설정은 SPARC 기반 시스템에서는 550이고 x86 기반 시스템에서는 260입니다. 최소 인터럽트 간 레벨을 0으로 설정하면 인터럽트 제한 논리가 사용 안함으로 설정됩니다.

6 **IP 인터페이스를 구성합니다.**

7 **(옵션) 임계값의 새 설정을 표시합니다.**

예 8–9 DMA 바인딩 구성 및 인터럽트 제한 속도 설정

이 예에서는 x86 기반 시스템을 `e1000g` NIC와 함께 사용합니다. 패킷 전송에 대한 임계값 설정을 DMA 바인딩 또는 `bcopy` 모드 사용 간에 토글하여 드라이버를 구성합니다. 인터럽트 제한 속도 설정도 수정합니다. 또한 `e1000g` 데이터 링크는 OS에서 할당된 기본 일반 이름을 사용합니다. 따라서 사용자 정의 이름 `net0`을 참조하여 데이터 링크에서 구성을 수행합니다.

```
# dladm show-phys
LINK        MEDIA        STATE     SPEED     DUPLEX     DEVICE
net0        ether        up        100Mb     full       e1000g0

# dladm show-linkprop -p _tx_bcopy_threshold net0
LINK        PROPERTY                VALUE     DEFAULT     POSSIBLE
net0        _tx_bcopy_threshold     512       512         --

# dladm show-linkprop -p _intr-throttling_rate
LINK        PROPERTY                  VALUE     DEFAULT     POSSIBLE
net0        _intr-throttling_rate     260       260         --

# ipadm delete-ip net0
# dladm set-linkprop -p _tx_bcopy_threshold=1024 net0
# dladm set-linkprop -p _intr_adaptive=0 net0
# dladm set-linkprop -p _intr-throttling_rate=1024 net0

# ipadm create-ip net0
# ipadm create-addr -T static -a 10.10.1.2/24 net0/v4addr
# dladm show-linkprop -p _tx_bocopy_threshold=1024 net0
LINK        PROPERTY                VALUE     DEFAULT     POSSIBLE
net0        _tx_bcopy_threshold     1024      512         --
```

```
# dladm show-linkprop -p _intr_adaptive net0
LINK        PROPERTY           VALUE     DEFAULT     POSSIBLE
net0        _intr-adaptive     0         1           --

# dladm show-linkprop -p _intr-throttling_rate
LINK        PROPERTY                  VALUE     DEFAULT     POSSIBLE
net0        _intr-throttling_rate     1024      260         --
```

데이터 링크의 추가 구성 작업

이 절에서는 DR(동적 재구성) 수행 및 STREAMS 모듈 작업 등 `dladm` 명령을 사용하여 간소화된 기타 일반적인 구성 절차에 대해 설명합니다.

▼ 동적 재구성을 사용하여 네트워크 인터페이스 카드를 교체하는 방법

이 절차는 DR(동적 재구성)을 지원하는 시스템에만 적용됩니다. 네트워크 하드웨어 구성에서 네트워크 링크 구성을 분리하여 DR을 용이하게 하는 방법을 보여줍니다. 이제 DR을 완료한 후 네트워크 링크를 재구성할 필요가 없습니다. 대신 제거된 NIC의 링크 구성만 교체 NIC에 상속되도록 전송합니다.

시작하기 전에 DR 수행 절차는 시스템 유형에 따라 달라집니다. 먼저 다음을 완료해야 합니다.

- 시스템이 DR을 지원하는지 확인합니다.
- 활성 네트워크 구성 프로파일이 `DefaultFixed`인지 확인합니다. 시스템의 활성 NCP가 `DefaultFixed`가 아닌 경우 DR 사용에 대한 자세한 내용은 38 페이지 “NWAM이 다른 Oracle Solaris 네트워킹 기술과 함께 작동하는 방식”의 **동적 재구성 및 네트워크 구성 프로파일**을 참조하십시오.
- 시스템의 DR에 대해 설명하는 해당 매뉴얼을 참조하십시오.

  Oracle Sun 서버의 DR에 대한 현재 설명서를 찾으려면 `http://www.oracle.com/technetwork/indexes/documentation/index.html`에서 `dynamic reconfiguration`을 검색합니다.

주 – 다음 절차에서는 특히 데이터 링크에 대한 유연한 이름 사용과 관련된 DR 측면만 참조합니다. DR을 수행하는 전체 단계는 이 절차에 포함되지 않습니다. 시스템에 해당하는 DR 설명서를 참조해야 합니다.

---

1 관리자로 전환합니다.

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

2 (옵션) 시스템에서 데이터 링크의 물리적 속성 및 해당 위치에 대한 정보를 표시합니다.

```
# dladm show-phys -L
```

`dladm show-phys -L`로 표시되는 정보 유형에 대한 자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

3 시스템 설명서에 설명된 대로 DR 절차를 수행하여 NIC를 제거한 다음 교체 NIC를 삽입합니다.

이 단계를 수행하려면 시스템의 DR 설명서를 참조하십시오.

교체 NIC를 설치한 후 다음 단계에서 계속합니다.

4 이전 NIC와 동일한 슬롯에 교체 NIC를 삽입한 경우 6단계로 건너뜁니다. 그렇지 않으면 다음 단계에서 계속합니다.

새 NIC가 이전 NIC에서 사용한 위치를 사용하는 경우 이전 NIC의 링크 이름과 구성을 상속합니다.

5 적용되는 상황에 따라 다음 단계 중 하나를 수행합니다.

- 교체할 이전 NIC가 시스템 슬롯에 사용되지 않은 NIC로 남아 있는 경우 다음 단계를 수행합니다.

  a. 교체할 NIC에 다른 이름을 지정합니다.

  ```
  # dladm rename-link oldNIC new-name
  ```

  oldNIC 교체되지만 시스템에 유지되는 NIC를 나타냅니다.

  new-name *removedNIC*에 지정되는 새 이름을 나타냅니다. 시스템의 다른 링크에서 이름을 공유하면 안됩니다.

  b. 이전 NIC의 이름을 교체 NIC에 지정합니다.

  ```
  # dladm rename-link replacementNIC oldNIC
  ```

  replacementNIC 방금 설치한 새 NIC를 나타냅니다. 이 NIC는 시스템에서 사용하는 슬롯에 따라 자동으로 기본 링크 이름을 받습니다.

  oldNIC 교체되지만 시스템에 유지되는 NIC를 나타냅니다.

- 이전 NIC를 제거했으며 다른 슬롯에 교체 NIC를 설치하지만 NIC에서 이전 NIC의 구성을 상속하려는 경우 이전 NIC의 이름을 새 NIC에 지정합니다.

  ```
  # dladm rename-link replacementNIC oldNIC
  ```

6 새 NIC의 리소스를 Oracle Solaris에서 사용할 수 있게 하여 DR 프로세스를 완료합니다.

예를 들어, `cfgadm` 명령을 사용하여 NIC를 구성합니다. 자세한 내용은 `cfgadm(1M)` 매뉴얼 페이지를 참조하십시오.

7 **(옵션) 링크 정보를 표시합니다.**

예를 들어, `dladm show-phys` 또는 `dladm show-link`를 사용하여 데이터 링크 정보를 표시할 수 있습니다.

예 8–10 새 네트워크 카드를 설치하여 동적 재구성 수행

이 예에서는 링크 이름이 `net0`인 `bge` 카드가 `e1000g` 카드로 교체되는 방식을 보여줍니다. `e1000g`가 시스템에 연결된 후 `net0`의 링크 구성이 `bge`에서 `e1000g`로 전송됩니다.

```
# dladm show-phys -L
LINK     DEVICE     LOCATION
net0     bge0       MB
net1     ibp0       MB/RISER0/PCIE0/PORT1
net2     ibp1       MB/RISER0/PCIE0/PORT2
net3     eoib2      MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

`cfgadm`을 사용하여 `bge` 제거, 해당 위치에 `e1000g` 설치 등의 DR 관련 단계를 수행합니다. 카드가 설치된 후 `e1000g0`의 데이터 링크는 자동으로 이름 `net0`을 갖게 되고 링크 구성을 상속합니다.

```
# dladm show-phys -L
LINK     DEVICE     LOCATION
net0     e1000g0    MB
net1     ibp0       MB/RISER0/PCIE0/PORT1
net2     ibp1       MB/RISER0/PCIE0/PORT2
net3     eoib2      MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2

# dladm show-link
LINK     CLASS     MTU    STATE    OVER
net0     phys      9600   up       ---
net1     phys      1500   down     ---
net2     phys      1500   down     --
net3     phys      1500   down     ---
```

데이터 링크에 STREAMS 모듈 구성

필요한 경우 최대 8개의 STREAMS 모듈이 데이터 링크 위에 푸시되도록 설정할 수 있습니다. 일반적으로 이러한 모듈은 VPN(가상 사설망) 및 방화벽과 같은 타사 네트워킹 소프트웨어에 사용됩니다. 네트워킹 소프트웨어에 대한 설명서는 소프트웨어 공급업체가 제공합니다.

특정 데이터 링크에 푸시할 STREAMS 모듈 목록은 `autopush` 링크 등록 정보로 제어됩니다. `autopush` 링크 등록 정보의 값은 `dladm set-linkprop` 하위 명령을 사용하여 설정됩니다.

별도의 autopush 명령을 사용하여 드라이버별로 STREAMS autopush 모듈을 설정할 수도 있습니다. 하지만 드라이버는 항상 NIC에 바인딩됩니다. 데이터 링크의 기본 NIC를 제거하면 링크의 autopush 등록 정보에 대한 정보도 손실됩니다.

STREAMS 모듈이 데이터 링크 위에 푸시되도록 구성하려면 autopush 명령 대신 dladm set-linkprop 명령을 사용합니다. 특정 데이터 링크에 대해 드라이버별 및 링크별 autoputsh 구성 유형이 있는 경우 dladm set-linkprop으로 설정된 링크별 정보가 사용되며 드라이버별 정보는 무시됩니다.

▼ 데이터 링크에 STREAMS 모듈을 설정하는 방법

다음 절차에서는 dladm set-linkprop 명령을 사용하여 STREAMS 모듈을 구성하는 방법에 대해 설명합니다.

1 관리자로 전환합니다.

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

2 링크를 열 때 모듈을 스트림에 푸시합니다.

```
# dladm set-linkprop -p autopush=modulelist  link
```

modulelist 자동으로 스트림에 푸시할 모듈 목록을 지정합니다. 링크 하나에 최대 8개 모듈을 푸시할 수 있습니다. 이러한 모듈은 *modulelist* 에 나열된 순서대로 푸시됩니다. 점을 구분자로 사용하여 목록에서 모듈을 구분합니다.

link 모듈이 푸시되는 링크를 지정합니다.

예 8–11 autopush 링크 등록 정보 설정

이 예에서는 vpnmod 및 bufmod 모듈을 net0 링크 위에 푸시합니다. 링크의 기본 장치는 bge0입니다.

```
# dladm set-linkprop -p autopush=vpnmod.bufmod net0
```

나중에 bge 카드를 e1000g로 교체하는 경우 autopush 설정을 재구성할 필요 없이 새 데이터 링크로 전환할 수 있습니다. e1000g 카드는 bge의 링크 이름과 구성을 자동으로 상속합니다.

▼ autopush 링크 등록 정보 설정을 가져오는 방법

1 관리자로 전환합니다.

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

2 **autopush 링크 등록 정보 설정을 표시합니다.**

```
# dladm show-linkprop -p autopush [link]
```

*link*를 지정하지 않으면 구성된 모든 링크의 정보가 표시됩니다.

▼ autopush 링크 등록 정보 설정을 제거하는 방법

1 **관리자로 전환합니다.**

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

2 **특정 데이터 링크의 autopush 링크 등록 정보 설정을 제거합니다.**

```
# dladm reset-linkprop [-t] -p autopush link
```

`-t` 옵션을 사용하여 등록 정보 설정을 일시적으로 제거합니다. 시스템을 재부트하면 설정이 복원됩니다.

◆ ◆ ◆ 9 장

9 IP 인터페이스 구성

이 장에서는 데이터 링크에 IP 인터페이스를 구성하는 데 사용되는 절차를 제공합니다.

IP 인터페이스 구성 정보

Oracle Solaris를 설치한 후 다음 작업을 수행할 수 있습니다.

- 기본 인터페이스 구성을 위해 데이터 링크에 IP 인터페이스를 구성합니다. 이 장에서는 절차에 대해 설명합니다.
- 무선 인터페이스를 구성합니다. 이 절차는 10 장, “Oracle Solaris에서 무선 인터페이스 통신 구성”에서 설명합니다.
- IP 인터페이스를 IPMP 그룹의 구성원으로 구성합니다. 이 절차는 15 장, “IPMP 관리”에서 설명합니다.

ipadm 명령

Oracle Solaris의 향상 기능이 기존 도구의 기능보다 훨씬 효율적으로 다양한 네트워크 측면을 관리합니다. 예를 들어, `ifconfig` 명령은 네트워크 인터페이스를 구성하는 사용자 정의 도구였습니다. 하지만 이 명령은 지속 구성 설정을 구현하지 않습니다. 여러 기간에 걸쳐 네트워크 관리의 기능 추가를 위해 `ifconfig`가 향상되었습니다. 하지만 그 결과, 명령이 복잡하고 사용 시 혼동을 가져왔습니다.

인터페이스 구성 및 관리의 다른 문제는 TCP/IP 인터넷 프로토콜 등록 정보나 조정 가능 항목을 관리하는 간단한 도구가 없다는 것입니다. `ndd` 명령은 이런 용도로 사전 설정된 사용자 정의 도구였습니다. 하지만 `ifconfig` 명령과 마찬가지로 `ndd`는 지속 구성 설정을 구현하지 않습니다. 이전에는 부트 스크립트를 편집하여 네트워크 시나리오에 대해 지속 설정을 시뮬레이션할 수 있었습니다. Oracle Solaris의 SMF 기능이 도입되면서 특히 Oracle Solaris 설치로 업그레이드할 경우 SMF 종속성 관리가 복잡하기 때문에 이러한 해결 방법의 사용이 위험할 수 있습니다.

ipadm 명령은 궁극적으로 인터페이스 구성에 대해 ifconfig 명령을 대체하도록 도입되었습니다. 또한 이 명령은 프로토콜 등록 정보를 구성하는 ndd 명령을 대체합니다.

인터페이스 구성 도구로서 ipadm 명령은 다음과 같은 이점을 제공합니다.

- 인터페이스 구성 이외의 용도로 사용되는 ifconfig 명령과 달리 IP 인터페이스 관리에만 사용되는 도구이므로 IP 인터페이스와 IP 주소를 보다 효율적으로 관리합니다.
- 지속 인터페이스 및 주소 구성 설정을 구현하는 옵션을 제공합니다.

ifconfig 옵션 및 동등한 ipadm 하위 명령의 목록은 184 페이지 “ifconfig 명령 옵션 및 ipadm 명령 옵션”을 참조하십시오.

프로토콜 등록 정보를 설정하는 도구로서 ipadm 명령은 다음과 같은 이점을 제공합니다.

- IP, ARP(Address Resolution Protocol), SCTP(Stream Control Transmission Protocol) 및 ICMP(Internet Control Messaging Protocol)와 TCP, UDP(User Datagram Protocol) 등의 상위 계층 프로토콜에 대한 임시 또는 지속 프로토콜 등록 정보를 설정할 수 있습니다.
- 등록 정보의 현재 및 기본 설정, 가능한 설정 범위 등 각 TCP/IP 매개변수에 대한 정보를 제공합니다. 따라서 디버깅 정보를 보다 쉽게 얻을 수 있습니다.
- 또한 ipadm 명령은 일관된 명령 구문을 따르므로 사용하기 더 쉽습니다.

ndd 옵션 및 동등한 ipadm 하위 명령의 목록은 186 페이지 “ndd 명령 옵션 및 ipadm 명령 옵션”을 참조하십시오.

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

IP 인터페이스 구성(작업)

이 절에서는 IP 인터페이스의 기본 구성 절차에 대해 설명합니다. 다음 표에서는 구성 작업을 설명하고 이러한 작업을 해당 절차에서 매핑할 수 있습니다.

표 9–1 IP 인터페이스 구성(작업 맵)

작업	설명	수행 방법
시스템에서 고유한 MAC 주소를 지원하도록 설정합니다.	SPARC 기반 시스템에서 인터페이스에 고유한 MAC 주소를 허용하도록 구성합니다.	163 페이지 “SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법”
ipadm 명령을 사용하여 기본 IP 인터페이스 구성을 수행합니다.	IP 인터페이스를 만들고 정적 또는 DHCP인 유효한 IP 주소를 할당합니다.	164 페이지 “IP 인터페이스를 구성하는 방법”
ipadm 명령을 사용하여 IP 주소를 사용자 정의합니다.	지정된 IP 주소의 네트워크 ID를 설정합니다.	169 페이지 “IP 주소의 등록 정보를 설정하는 방법”

표 9–1 IP 인터페이스 구성(작업 맵) (계속)

작업	설명	수행 방법
ipadm 명령을 사용하여 인터페이스 정보를 가져옵니다.	인터페이스, 주소 및 프로토콜의 여러 등록 정보와 해당 설정을 나열합니다.	179 페이지 "네트워크 인터페이스 정보를 가져오는 방법"

▼ SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법

일부 응용 프로그램에서는 호스트의 각 인터페이스가 고유한 MAC 주소를 가져야 합니다. 하지만 모든 SPARC 기반 시스템에는 시스템 차원의 MAC 주소가 있으며, 기본적으로 모든 인터페이스가 이 주소를 사용합니다. 다음은 SPARC 시스템의 인터페이스에 대해 출하 시 설치된 MAC 주소를 구성하려는 두 가지 경우입니다.

- 링크 통합의 경우 인터페이스의 출하 시 설정된 MAC 주소를 통합 구성에 사용해야 합니다.
- IPMP 그룹의 경우 그룹의 각 인터페이스에 고유한 MAC 주소가 있어야 합니다. 이러한 인터페이스는 출하 시 설치된 MAC 주소를 사용해야 합니다.

EEPROM 매개변수 `local-mac-address?`는 SPARC 시스템의 모든 인터페이스가 시스템 차원의 MAC 주소를 사용하는지 또는 고유한 MAC 주소를 사용하는지를 확인합니다. 다음 절차에서는 `eeprom` 명령을 사용하여 `local-mac-address?`의 현재 값을 확인하고 필요한 경우 변경하는 방법을 보여줍니다.

1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 "관리 권한을 얻는 방법"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. 또한 통합에 대해 `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 주소가 있는 인터페이스가 이제 시스템 차원의 MAC 주소 대신 이러한 출하 시 설정을 사용합니다. 출하 시 설정된 MAC 주소가 없는 인터페이스는 계속해서 시스템 차원의 MAC 주소를 사용합니다.

4 **시스템에 있는 모든 인터페이스의 MAC 주소를 확인합니다.**

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

```
# dladm show-linkprop -p mac-address
LINK    PROPERTY        PERM VALUE              DEFAULT           POSSIBLE
net0    mac-address     rw   8:0:20:0:0:1       8:0:20:0:0:1      --
net1    mac-address     rw   8:0:20:0:0:1       8:0:20:0:0:1      --
net3    mac-address     rw   0:14:4f:45:c:2d    0:14:4f:45:c:2d   --
```

주 – 둘 이상의 네트워크 인터페이스가 동일한 MAC 주소를 가진 경우에만 다음 단계에서 계속합니다. 그렇지 않으면 최종 단계로 이동합니다.

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

```
# dladm set-linkprop -p mac-address=mac-address  interface
```

이전 단계의 예에서는 로컬에서 관리되는 MAC 주소를 사용하여 `net0` 및 `net1`을 구성해야 합니다. 예를 들어, 로컬에서 관리되는 MAC 주소 `06:05:04:03:02`를 사용하여 `net0`을 재구성하려면 다음 명령을 입력합니다.

```
# dladm set-linkprop -p mac-address=06:05:04:03:02 net0
```

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

6 **시스템을 다시 부트합니다.**

IP 인터페이스 구성

뒤에 나오는 절차는 `ipadm` 명령을 여러 IP 구성 요구에 사용하는 방법을 보여줍니다. `ifconfig` 명령도 인터페이스 구성에 계속 작동하지만 `ipadm` 명령이 기본 도구여야 합니다. `ipadm` 명령 및 해당 이점의 개요는 161 페이지 “ipadm 명령”을 참조하십시오.

주 – 일반적으로 IP 인터페이스 구성과 데이터 링크 구성은 함께 수행됩니다. 따라서 해당하는 경우 뒤에 나오는 절차에 `dladm` 명령을 사용한 데이터 링크 구성 단계가 포함됩니다. `dladm` 명령을 사용한 데이터 링크 구성 및 관리에 대한 자세한 내용은 8 장, “데이터 링크 구성 및 관리”를 참조하십시오.

▼ IP 인터페이스를 구성하는 방법

다음 절차에서는 IP 인터페이스의 기본 구성을 수행하는 예를 제공합니다.

시작하기 전에 시스템에서 데이터 링크의 이름을 바꿀 것인지 결정합니다. 일반적으로 데이터 링크에 기본적으로 지정된 일반 이름을 사용합니다. 링크 이름을 변경하려면 143 페이지 "데이터 링크의 이름을 바꾸는 방법"을 참조하십시오.

1 관리자로 전환합니다.

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

2 (옵션) 현재 시스템에 있는 데이터 링크의 물리적 속성에 대한 정보를 표시합니다.

```
# dladm show-phys
```

이 명령은 시스템에 설치된 물리적 네트워크 카드와 일부 등록 정보를 보여줍니다. 이 명령에 대한 자세한 내용은 데이터 링크의 물리적 속성에 대한 정보를 표시하는 방법을 참조하십시오.

3 현재 시스템에 있는 데이터 링크에 대한 정보를 표시합니다.

```
# dladm show-link
```

이 명령은 데이터 링크 및 링크가 생성된 물리적 카드를 비롯하여 설정된 특정 등록 정보를 보여줍니다.

4 IP 인터페이스를 만듭니다.

```
# ipadm create-interface-class interface
```

interface-class 만들 수 있는 세 가지 인터페이스 클래스 중 하나를 나타냅니다.

- IP 인터페이스. 이 인터페이스 클래스는 네트워크 구성을 수행할 때 만드는 가장 일반적인 인터페이스입니다. 이 인터페이스 클래스를 만들려면 `create-ip` 하위 명령을 사용합니다.
- STREAMS 가상 네트워크 인터페이스 드라이버(VNI 인터페이스). 이 인터페이스 클래스를 만들려면 `create-vni` 하위 명령을 사용합니다. VNI 장치 또는 인터페이스에 대한 자세한 내용은 `vni(7d)` 매뉴얼 페이지를 참조하십시오.
- IPMP 인터페이스. 이 인터페이스는 IPMP 그룹을 구성할 때 사용됩니다. 이 인터페이스 클래스를 만들려면 `create-ipmp` 하위 명령을 사용합니다. IPMP 그룹에 대한 자세한 내용은 14 장, "IPMP 소개" 및 15 장, "IPMP 관리"를 참조하십시오.

interface 인터페이스의 이름을 나타냅니다. 이 이름은 인터페이스를 만들 링크의 이름과 같습니다.

주 – IP 주소를 할당하려면 먼저 IP 인터페이스를 만들어야 합니다.

---

5 유효한 IP 주소로 IP 인터페이스를 구성합니다.

다음 구문은 인터페이스에 정적 주소를 할당합니다. IP 주소를 할당하는 다른 옵션은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

```
# ipadm create-addr -T address-type -a address/prefixlen addrobj
```

-T *address-type* 인터페이스에 할당되는 IP 주소의 유형을 지정하며 static, dhcp 또는 addrconf 중 하나입니다. Addrconf는 자동으로 생성된 IPv6 주소를 나타냅니다.

-a 인터페이스에 구성할 IP 주소를 지정합니다. 로컬 주소만 지정하거나, 터널 구성의 경우 로컬 주소와 원격 주소를 모두 지정할 수 있습니다. 일반적으로 로컬 주소만 할당합니다. 이 경우 -a 옵션을 사용하여 주소를 직접 지정합니다(예: -a *address*). 주소가 자동으로 로컬 주소로 간주됩니다.

터널을 구성하는 경우 시스템의 로컬 주소와 대상 시스템의 원격 주소를 모두 제공해야 할 수도 있습니다. 이 경우 두 주소를 구분하려면 local 및 remote를 지정해야 합니다(예: -a local=*local-addr*, remote= *remote-addr*). 터널 구성에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 6 장, "IP 터널 구성"을 참조하십시오.

숫자 IP 주소를 사용하는 경우 CIDR 표기법의 주소인 *address*/*prefixlen* 형식을 사용합니다(예: 1.2.3.4/24). */prefixlen* 옵션에 대한 설명을 참조하십시오.

필요에 따라 숫자 IP 주소 대신 호스트 이름을 *address*에 지정할 수 있습니다. /etc/hosts 파일에서 해당 호스트 이름에 대해 숫자 IP 주소가 정의된 경우 호스트 이름을 사용해도 유효합니다. 파일에 숫자 IP 주소가 정의되지 않은 경우 name-service/switch 서비스에서 host에 대해 지정된 분석기 순서를 사용하여 숫자 값을 고유하게 가져옵니다. 지정된 호스트 이름에 대한 항목이 여러 개 있으면 오류가 생성됩니다.

주 – 부트 프로세스 도중 IP 주소가 먼저 생성된 후 이름 지정 서비스가 온라인 상태로 전환됩니다. 따라서 네트워크 구성에서 사용된 호스트 이름이 /etc/hosts 파일에 정의되어 있는지 확인해야 합니다.

/prefixlen CIDR 표기법을 사용할 때 IPv4 주소에 포함되는 네트워크 ID의 길이를 지정합니다. 주소 12.34.56.78/24에서 24는 *prefixlen*입니다. *prefixlen*을 포함하지 않으면 name-service/switch 서비스에서 netmask에 대해 나열된 시퀀스에 따라 또는 클래스 주소 의미 체계를 사용하여 넷마스크가 계산됩니다.

addrobj 시스템에서 사용된 고유 IP 주소 또는 주소 세트에 대한 식별자를 지정합니다. 주소는 IPv4 또는 IPv6 유형일 수 있습니다. 이 식별자는 *interface*/*user_specified_string* 형식을 사용합니다.

*interface*는 주소가 할당된 IP 인터페이스를 나타냅니다. *interface* 변수는 IP 인터페이스가 구성된 데이터 링크의 이름을 반영해야 합니다.

*user-specified-string*은 알파벳 문자로 시작하고 최대 길이가 32자인 영숫자의 문자열을 나타냅니다. 나중에 시스템의 주소를 관리하는 `ipadm` 하위 명령을 사용할 때 숫자 IP 주소 대신 `addrobj`를 참조할 수 있습니다(예: `ipadm show-addr` 또는 `ipadm delete-addr`).

6 (옵션) 새로 구성된 IP 인터페이스에 대한 정보를 표시합니다.

확인하려는 정보에 따라 다음 명령을 사용할 수 있습니다.

- 인터페이스의 일반 상태를 표시합니다.

  ```
  # ipadm show-if [interface]
  ```

  인터페이스를 지정하지 않으면 시스템의 모든 인터페이스에 대한 정보가 표시됩니다.

- 인터페이스의 주소 정보를 표시합니다.

  ```
  # ipadm show-addr [addrobj]
  ```

  *addrobj*를 지정하지 않으면 시스템의 모든 주소 객체에 대한 정보가 표시됩니다.

`ipadm show-*` 하위 명령의 출력 결과에 대한 자세한 내용은 178 페이지 “IP 인터페이스 및 주소 모니터링”을 참조하십시오.

7 (옵션) `/etc/hosts` 파일에 IP 주소에 대한 항목을 추가합니다.

이 파일의 항목은 IP 주소와 해당 호스트 이름으로 구성됩니다.

주 – 이 단계는 호스트 이름을 사용하는 정적 IP 주소를 구성하는 경우에만 적용됩니다. DHCP 주소를 구성하는 경우 `/etc/hosts` 파일을 업데이트할 필요가 없습니다.

예 9–1 정적 주소를 사용하여 네트워크 인터페이스 구성

```
# dladm show-phys
LINK     MEDIA        STATE     SPEED     DUPLEX     DEVICE
net3     Ethernet     up        100Mb     full       bge3

# dladm show-link
LINK     CLASS     MTU     STATE     BRIDGE   OVER
net3     phys      1500    up        --       --
```

```
# ipadm create-ip net3
# ipadm create-addr -T static -a 192.168.84.3/24 net3/v4static

# ipadm show-if
IFNAME   CLASS        STATE     ACTIVE     OVER
lo0      loopback     ok        yes        --
net3     ip           ok        yes        --

# ipadm show-addr
ADDROBJ     TYPE       STATE     ADDR
lo0/?       static     ok        127.0.0.1/8
net3/v4     static     ok        192.168.84.3/24

# vi /etc/hosts
# Internet host table
# 127.0.0.1       localhost
10.0.0.14       myhost
192.168.84.3    campus01
```

campus01이 /etc/hosts 파일에서 이미 정의된 경우 다음 주소를 할당할 때 해당 호스트 이름을 사용할 수 있습니다.

```
# ipadm create-addr -T static -a campus01 net3/v4static
```

예 9–2 IP 주소를 사용하여 자동으로 네트워크 인터페이스 구성

이 예에서는 위의 예와 동일한 네트워크 장치를 사용하지만 DHCP 서버에서 해당 주소를 받도록 IP 인터페이스를 구성합니다.

```
# dladm show-phys
LINK     MEDIA        STATE     SPEED     DUPLEX     DEVICE
net3     Ethernet     up        100Mb     full       bge3

# dladm show-link
LINK     CLASS     MTU     STATE     BRIDGE   OVER
net3     phys      1500    up        --       --

# ipadm create-ip net3

# ipadm create-addr -T dhcp net3/dhcp

# ipadm show-if
IFNAME   CLASS        STATE     ACTIVE     OVER
lo0      loopback     ok        yes        --
net3     ip           ok        yes        --

# ipadm show-addr net3/dhcp
ADDROBJ     TYPE       STATE     ADDR
net3/dhcp   dhcp       ok        10.8.48.242/24

# ipadm show-addr
ADDROBJ     TYPE       STATE     ADDR
lo0/?       static     ok        127.0.0.1/8
net3/dhcp   dhcp       ok        10.8.48.242/24
```

IP 주소 등록 정보 설정

ipadm 명령을 사용하면 인터페이스에 주소를 할당한 후 주소 관련 등록 정보를 설정할 수 있습니다. 이러한 등록 정보를 설정하여 다음을 결정할 수 있습니다.

- 주소의 prefixlen
- IP 주소를 아웃바운드 패킷의 소스 주소로 사용할 수 있는지 여부
- 주소가 전역 또는 비전역 영역에 속하는지 여부
- 주소가 개인 주소인지 여부

IP 주소의 등록 정보를 나열하려면 다음 구문을 사용합니다.

```
# ipadm show-addrprop [-p property] [addrobj]
```

표시되는 정보는 사용하는 옵션에 따라 달라집니다.

- 등록 정보와 주소 객체를 모두 지정하지 않으면 기존의 모든 주소 등록 정보가 모두 표시됩니다.
- 등록 정보만 지정하면 모든 주소의 해당 등록 정보가 표시됩니다.
- 주소 객체만 지정하면 해당 주소 객체의 등록 정보가 모두 표시됩니다.

주 – 한 번에 한 개의 주소 등록 정보만 설정할 수 있습니다.

▼ IP 주소의 등록 정보를 설정하는 방법

이 절차에서는 IP 주소의 등록 정보를 구성하는 일반적인 단계를 보여줍니다.

1 관리자로 전환합니다.

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

2 현재 시스템에서 사용 중인 IP 주소를 나열합니다.

```
# ipadm show-addr
```

3 (옵션) 변경하려는 IP 주소의 특정 등록 정보에 대한 현재 설정을 확인합니다.

```
# ipadm show-addrprop -p property addrobj
```

등록 정보를 모르는 경우 일반적인 ipadm show-addrprop 명령을 실행할 수 있습니다. 이 명령을 사용하여 IP 주소를 표시하면 모든 등록 정보의 현재 설정과 함께 주소가 표시됩니다.

4 선택한 등록 정보를 원하는 값으로 설정합니다.

```
# ipadm set-addrprop -p property=value addrobj
```

5 **등록 정보의 새 설정을 확인합니다.**

```
# ipadm show-addrprop -p property addrobj
```

예 9–3 주소의 prefixlen 등록 정보 설정

prefixlen 등록 정보는 IP 주소의 넷마스크를 나타냅니다. 다음 예에서는 net3의 IP 주소에 대한 prefixlen 등록 정보의 길이를 변경합니다. 이 예에서는 -t 옵션을 사용하여 등록 정보의 일시적인 변경만 만듭니다. 시스템을 재부트하면 등록 정보의 값이 기본 설정으로 돌아갑니다.

```
# ipadm show-addr
ADDROBJ     TYPE       STATE     ADDR
lo0/?       static     ok        127.0.0.1/8
net3/v4     static     ok        192.168.84.3/24

# ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ  PROPERTY   PERM  CURRENT  PERSISTENT  DEFAULT   POSSIBLE
net3/v4  prefixlen  rw    24       24          24        1-30,32

# ipadm set-addrprop -t -p prefixlen=8 net3/v4
# ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ  PROPERTY   PERM  CURRENT  PERSISTENT  DEFAULT   POSSIBLE
net3/v4  prefixlen  rw    8        24          24        1-30,32
```

IP 인터페이스 등록 정보 설정

데이터 링크처럼 IP 인터페이스에도 특정 네트워크 설정에 맞게 사용자 정의할 수 있는 등록 정보가 있습니다. 각 인터페이스에 대해 IPv4 및 IPv6 프로토콜에 각각 적용되는 두 개의 등록 정보 세트가 있습니다. MTU와 같은 일부 등록 정보는 데이터 링크와 IP 인터페이스에 공통됩니다. 따라서 데이터 링크의 MTU 설정과 해당 링크에 구성된 인터페이스의 MTU 설정을 다르게 지정할 수 있습니다. 해당 IP 인터페이스를 순회하는 IPv4 및 IPv6 패킷에 각각 적용되는 다른 MTU 설정을 지정할 수도 있습니다.

IP 전달은 일반적으로 네트워킹 시나리오에서 구성되는 IP 인터페이스 등록 정보입니다. 다음 절차에서 단계를 보여줍니다.

패킷 전달 사용

네트워크에서 호스트는 다른 호스트 시스템으로 전송된 데이터 패킷을 받을 수 있습니다. 수신 로컬 시스템에서 패킷 전달을 사용으로 설정하면 해당 시스템이 데이터 패킷을 대상 호스트로 전달할 수 있습니다. 기본적으로 IP 전달은 사용 안함으로 설정됩니다. 다음 두 절차에서는 이 기능을 사용으로 설정하는 방법에 대해 설명합니다. 이전 Oracle Solaris 릴리스에서는 routeadm 명령을 사용하여 패킷 전달을 사용으로 설정했습니다. 이 절차에서는 ipadm 구문이 routeadm 명령 대신 사용됩니다.

다음을 고려하여 인터페이스 기반 또는 프로토콜 기반 절차를 사용할 것인지 결정합니다.

- 패킷 전달 방식을 선택하려는 경우 인터페이스에서 패킷 전달을 사용으로 설정합니다. 예를 들어, 여러 NIC가 포함된 시스템이 있을 수 있습니다. 일부 NIC는 외부 네트워크에 연결되고 다른 NIC는 개인 네트워크에 연결됩니다. 따라서 모든 인터페이스가 아니라 일부 인터페이스에서만 패킷 전달을 사용으로 설정합니다. 171 페이지 "인터페이스 등록 정보를 설정하여 IP 패킷 전달을 사용으로 설정하는 방법"을 참조하십시오.
- 시스템 내에서 전역적으로 패킷 전달을 구현하려는 경우 프로토콜의 `forwarding` 등록 정보를 사용으로 설정합니다. 이 두번째 방법은 173 페이지 "프로토콜 등록 정보를 설정하여 패킷 전달을 사용으로 설정하는 방법"을 참조하십시오.

주 – 두 가지 패킷 전달 방법을 함께 사용할 수 있습니다. 예를 들어, 전역적으로 패킷 전달을 사용으로 설정한 다음 각 인터페이스에 대한 `forwarding` 등록 정보를 사용자 정의할 수 있습니다. 따라서 특정 시스템에 대해 패킷 전달을 선택할 수 있습니다.

▼ 인터페이스 등록 정보를 설정하여 IP 패킷 전달을 사용으로 설정하는 방법

이 절차에서는 선택적으로 특정 인터페이스의 IP 전달 등록 정보를 구성하여 패킷 전달을 사용으로 설정하는 방법을 보여줍니다.

주 – 패킷 전달에는 IP 프로토콜이 사용됩니다. 따라서 IP **프로토콜 버전** 구분도 단계에 포함되어 있습니다.

---

1 관리자로 전환합니다.

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

2 인터페이스의 IP 전달 등록 정보에 대한 현재 설정을 표시합니다.

```
# ipadm show-ifprop -p forwarding [-m protocol-version] interface
```

여기서 *protocol-version*은 `ipv4` 또는 `ipv6`일 수 있습니다. 버전을 지정하지 않으면 IPv4 및 IPv6 프로토콜에 대한 설정이 모두 표시됩니다.

주 – 지정된 인터페이스의 유효한 프로토콜 등록 정보를 모두 표시하려면 다음과 같이 등록 정보를 지정하지 마십시오.

```
# ipadm show-ifprop interface
```

이 구문은 예 9–4에도 나와 있습니다.

3. **패킷 전달을 사용으로 설정하려는 각 인터페이스에 대해 다음 명령을 입력합니다.**

```
# ipadm set-ifprop forwarding=on -m protocol-version interface
```

4. **(옵션) 인터페이스의 `forwarding` 등록 정보에 대한 설정을 표시합니다.**

```
# ipadm show-ifprop -p forwarding interface
```

5. **인터페이스의 `forwarding` 등록 정보를 기본 설정으로 복원하려면 다음 명령을 입력합니다.**

```
# ipadm reset-ifprop -p forwarding -m protocol-version interface
```

예 9–4 인터페이스에서 IPv4 패킷만 전달할 수 있도록 허용

다음 예에서는 선택적 패킷 전달을 구현하는 방법을 보여줍니다. 이 경우 `net0` 인터페이스에서 IPv4 패킷의 전달만 사용으로 설정됩니다. 시스템의 나머지 인터페이스에서는 패킷 전달이 사용 안함으로 설정되며, 이것이 기본 설정입니다.

```
# ipadm show-ifprop -p forwarding net0
IFNAME   PROPERTY     PROTO   PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
net0     forwarding   ipv4    rw     off       off          off       on,off
net0     forwarding   ipv6    rw     off       --           off       on,off
```

`-p property` 옵션을 사용하는 `ipadm show-ifprop` 명령 구문은 특정 등록 정보에 대한 정보만 제공합니다.

```
# ipadm set-ifprop -p forwarding=on -m ipv4 net0
# ipadm show-ifprop net0
IFNAME   PROPERTY         PROTO  PERM  CURRENT  PERSISTENT  DEFAULT   POSSIBLE
...
net0     forwarding       ipv4   rw    on       on          off       on,off
...
```

`-p property` 옵션이 없는 `ipadm show-ifprop` 명령 구문은 인터페이스의 모든 등록 정보와 해당 설정을 표시합니다.

```
# ipadm reset-ifprop -p forwarding -m ipv4 net0
# ipadm show-ifprop -p forwarding -m ipv4 net0
IFNAME   PROPERTY     PROTO   PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
net0     forwarding   ipv4    rw     off       off          off       on,off
```

`ipadm reset-ifprop` 명령 구문은 지정된 등록 정보를 기본 설정으로 재설정합니다.

▼ 프로토콜 등록 정보를 설정하여 패킷 전달을 사용으로 설정하는 방법

이 절차에서는 시스템에서 전역적으로 패킷 전달을 사용으로 설정하는 방법을 보여줍니다.

1 관리자로 전환합니다.

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

2 IP 전달 등록 정보의 현재 설정을 표시합니다.

```
# ipadm show-prop -p forwarding protocol-version
```

여기서 *protocol-version*은 `ipv4` 또는 `ipv6`일 수 있습니다.

주 – 지정된 프로토콜에 유효한 모든 조정 가능 등록 정보와 현재 설정을 표시하려면 다음 명령을 입력합니다.

```
# ipadm show-prop protocol
```

여기서 *protocol*은 `ip`, `ipv4`, `ipv6`, `udp`, `tcp`, `icmp` 및 `sctp`일 수 있습니다.

이 구문은 예 9–5에 나와 있습니다.

---

3 전달을 사용으로 설정하려는 각 프로토콜 버전에 대해 다음 명령을 입력합니다.

```
# ipadm set-prop forwarding=on protocol-version
```

4 (옵션) 다음 중 하나를 수행하여 IP 전달 등록 정보의 설정을 표시합니다.

- 프로토콜의 모든 등록 정보와 현재 설정을 표시하려면 다음을 입력합니다.

  ```
  # ipadm show-prop protocol
  ```

- 프로토콜의 특정 등록 정보를 표시하려면 다음을 입력합니다.

  ```
  # ipadm show-prop -p property protocol
  ```

- 특정 프로토콜 버전의 특정 등록 정보를 표시하려면 다음을 입력합니다.

  ```
  # ipadm show-prop -p property protocol-version
  ```

5 프로토콜 버전의 특정 등록 정보를 기본 설정으로 재설정하려면 다음을 입력합니다.

```
# ipadm reset-prop -p property protocol-version
```

예 9–5 IPv4 및 IPv6 패킷에 대해 전달 사용

다음 예는 인터페이스의 패킷 전달에 대한 위의 예와 유사합니다. ipadm show-prop의 두 사용은 지정된 등록 정보의 설정을 표시하거나 프로토콜의 모든 등록 정보 및 해당 설정을 표시합니다.

```
# ipadm show-prop -p forwarding ip
PROTO   PROPERTY     PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4    forwarding   rw     off       --           off       on,off
ipv6    forwarding   rw     off       --           off       on,off
#
# ipadm set-prop -p forwarding=on ipv4
# ipadm set-prop -p forwarding=on ipv6
#
# ipadm show-prop ip
PROTO   PROPERTY     PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4    forwarding   rw     on        on           off       on,off
ipv4    ttl          rw     255       --           255       1-255
ipv6    forwarding   rw     on        on           off       on,off
ipv6    hoplimit     rw     255       --           255       1-255#
```

프로토콜 등록 정보 관리

인터페이스와 별도로 ipadm 명령을 사용하여 조정 가능 항목이라고도 하는 프로토콜 등록 정보를 구성할 수 있습니다. ipadm은 이전 릴리스에서 조정 가능 항목을 설정하는 데 주로 사용되었던 ndd 명령을 대체합니다. 이 절에서는 선택한 TCP/IP 프로토콜 등록 정보를 사용자 정의하는 절차와 예를 제공합니다.

TCP/IP 등록 정보 설정

TCP/IP 등록 정보는 인터페이스 기반이거나 전역일 수 있습니다. 특정 인터페이스나 영역의 모든 인터페이스에 전역적으로 등록 정보를 적용할 수 있습니다. 전역 등록 정보는 각 비전역 영역에서 다른 설정을 사용할 수 있습니다. 지원되는 프로토콜 등록 정보 목록은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

일반적으로 TCP/IP 인터넷 프로토콜의 기본 설정으로도 네트워크가 작동하는 데 충분합니다. 하지만 기본 설정이 네트워크 토폴로지에 충분하지 않은 경우 다음 표의 절차에서 이러한 TCP/IP 등록 정보를 사용자 정의하는 방법을 보여줍니다.

이 표에서는 프로토콜의 특정 등록 정보를 구성하는 작업을 설명하고 해당 절차에 대한 링크를 제공합니다.

표 9–2 선택한 TCP/IP 등록 정보 설정

작업	설명	수행 방법
포트를 권한 부여됨으로 표시합니다.	root 사용자를 제외하고 액세스를 제한하기 위해 인터페이스의 포트를 예약합니다.	175 페이지 “포트 액세스를 root 사용자로만 제한하는 방법”
멀티홈 호스트에서 수신 또는 전송 중인 IP 패킷의 동작을 사용자 정의합니다.	멀티홈 호스트에서 대칭 경로 지정을 사용자 정의합니다.	177 페이지 “멀티홈 호스트에 대칭 경로 지정을 구현하는 방법”
프로토콜의 등록 정보에 대한 정보를 표시합니다.	프로토콜의 등록 정보 및 현재 설정을 표시합니다.	178 페이지 “IP 인터페이스 및 주소 모니터링”

주 – ipadm 도구를 사용하여 네트워크 인터페이스와 IP 주소를 구성하는 절차는 164 페이지 “IP 인터페이스 구성”을 참조하십시오.

▼ 포트 액세스를 root 사용자로만 제한하는 방법

TCP, UDP 및 SCTP와 같은 전송 프로토콜에서 포트 1–1023은 루트 권한으로 실행된 프로세스만 이러한 포트에 바인딩할 수 있도록 권한이 부여된 기본 포트입니다. ipadm 명령을 사용하면 권한이 부여된 포트가 되도록 이 지정된 기본 범위를 초과하여 포트를 예약할 수 있습니다. 따라서 루트 프로세스만 해당 포트에 바인딩할 수 있습니다. 이 절차에서는 다음 전송 프로토콜 등록 정보를 사용합니다.

- smallest_nonpriv_port
- extra_priv_ports

1 지정된 포트가 일반 포트의 범위 내에 있고 사용될 수 있는지 확인합니다.

```
# ipadm show-prop -p smallest_nonpriv_port protocol
```

여기서 *protocol*은 권한이 부여된 포트를 구성하려는 프로토콜 유형(예: IP, UDP, ICMP 등)입니다.

명령 출력 결과의 POSSIBLE 필드에는 일반 사용자가 바인딩할 수 있는 포트 번호의 범위가 표시됩니다. 지정된 포트가 이 범위 내에 있으면 권한이 부여된 포트로 설정할 수 있습니다.

2 예약하려는 포트가 사용 가능하며 권한이 부여된 포트로 이미 표시되지 않았는지 확인합니다.

```
# ipadm show-prop -p extra_priv_ports protocol
```

명령 출력 결과의 CURRENT 필드는 현재 권한 부여됨으로 표시된 포트를 나타냅니다. 지정된 포트가 이 필드 아래에 없는 경우 권한이 부여된 포트로 설정할 수 있습니다.

3 **지정된 포트를 권한이 부여된 포트로 추가합니다.**

```
# ipadm set-prop -p extra_priv_ports=port-number protocol
```

4 **권한이 부여된 포트로 추가하거나 제거하려는 각 포트에 대해 다음 중 하나를 반복합니다.**

- 포트를 권한이 부여된 포트로 추가하려면 다음 구문을 입력합니다.

  ```
  # ipadm set-prop -p extra_priv_ports+=portnumber protocol
  ```

  주 – 더하기 기호(+) 수식자를 통해 권한이 부여된 포트가 되도록 포트를 여러 개 할당할 수 있습니다. 더하기 기호 수식자를 사용하여 이러한 포트 목록을 작성할 수 있습니다. 이 구문에 수식자를 사용하면 목록에 포트를 개별적으로 추가할 수 있습니다. 수식자를 사용하지 않으면 할당한 포트가 이전에 권한 부여됨으로 나열된 다른 모든 포트를 대체합니다.

- 권한이 부여된 포트인 지정된 포트를 제거하려면 다음 구문을 입력합니다.

  ```
  # ipadm set-prop -p extra_priv_ports-=portnumber protocol
  ```

  주 – 빼기 기호(-) 수식자를 사용하면 현재 권한 부여됨으로 나열된 기존 포트에서 포트를 제거할 수 있습니다. 동일한 구문을 사용하여 기본 포트를 포함한 기타 권한이 부여된 포트를 모두 제거할 수 있습니다.

5 **지정된 포트의 새 상태를 확인합니다.**

```
# ipadm show-prop -p extra_priv_ports protocol
```

명령 출력 결과에서 지정된 포트가 CURRENT 필드에 포함되어 있는지 확인합니다.

예 9–6 권한이 부여된 포트 설정

이 예에서는 포트 3001 및 3050을 권한이 부여된 포트로 설정합니다. 또한 현재 권한이 부여된 포트로 나열된 포트 4045를 제거합니다.

smallest_nonpriv_port 등록 정보의 출력 결과에서 POSSIBLE 필드는 포트 1024가 권한이 부여되지 않은 최하위 포트이며 지정된 포트 3001과 3050이 사용 가능한 권한이 부여되지 않은 포트 범위 내에 있음을 나타냅니다. extra_priv_ports 등록 정보의 출력 결과에서 CURRENT 필드 아래의 포트 2049 및 4045는 권한 부여됨으로 표시됩니다. 따라서 포트 3001을 권한이 부여된 포트로 계속 설정할 수 있습니다.

```
# ipadm show-prop -p smallest_nonpriv_port tcp
PROTO PROPERTY                PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
tcp   smallest_nonpriv_port   rw     1024      --           1024      1024-32768

# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
```

```
tcp     extra_priv_ports    rw     2049,4045    --           2049,4045    1-65535

# ipadm set-prop -p extra_priv_ports+=3001 tcp
# ipadm set-prop -p extra_priv_ports+=3050 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO    PROPERTY            PERM   CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp      extra_priv_ports    rw     2049,4045    3001,3050    2049,4045    1-65535
                                    3001,3050

# ipadm set-prop -p extra_priv_ports-=4045 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO    PROPERTY            PERM   CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp      extra_priv_ports    rw     2049,3001    3001,3050    2049,4045    1-65535
                                    3050
```

▼ 멀티홈 호스트에 대칭 경로 지정을 구현하는 방법

기본적으로 여러 인터페이스가 있는 시스템은 **멀티홈 호스트**라고도 하며, 경로 지정 테이블에서 트래픽 대상까지 가장 긴 일치 경로를 기준으로 네트워크 트래픽의 경로를 지정합니다. 대상까지의 길이가 같은 경로가 여러 개 있을 경우 Oracle Solaris는 ECMP(Equal Cost Multipathing) 알고리즘을 적용하여 트래픽을 해당 경로에 분산시킵니다.

이런 방식의 트래픽 분산이 적합하지 않은 경우도 있습니다. 패킷의 IP 소스 주소와 동일한 서브넷에 없는 멀티홈 호스트의 인터페이스를 통해 IP 패킷이 전송될 수 있습니다. 또한 송신 패킷이 특정 수신 요청(예: ICMP 에코 요청)에 대한 응답인 경우 요청과 응답이 동일한 인터페이스를 순회할 수 없습니다. 이러한 트래픽 경로 지정 구성을 비대칭 경로 지정이라고 합니다. 인터넷 서비스 공급자가 RFC 3704(http://rfc-editor.org/rfc/bcp/bcp84.txt)에 설명된 대로 진입 필터링을 구현하는 경우 비대칭 경로 지정 구성으로 인해 공급자가 송신 패킷을 삭제할 수도 있습니다.

RFC 3704는 인터넷에서 서비스 거부 공격을 제한하기 위한 것입니다. 이 의도를 준수하려면 네트워크에서 대칭 경로 지정을 구성해야 합니다. Oracle Solaris에서 IP `hostmodel` 등록 정보를 사용하면 이 요구 사항을 충족할 수 있습니다. 이 등록 정보는 멀티홈 호스트를 통해 수신 또는 전송된 IP 패킷의 동작을 제어합니다.

다음 절차에서는 `ipadm` 명령을 사용하여 특정 경로 지정 구성의 `hostmodel` 등록 정보를 설정하는 방법을 보여줍니다.

1 멀티홈 호스트에서 관리자가 됩니다.

2 시스템에서 네트워크 패킷의 경로 지정을 구성합니다.

```
# ipadm set-prop -p hostmodel=value protocol
```

이 등록 정보를 다음 세 가지 설정 중 하나로 구성할 수 있습니다.

`strong` RFC 1122에 정의된 강력한 ES(엔드 시스템) 모델에 해당합니다. 이 설정은 대칭 경로 지정을 구현합니다.

`weak`	RFC 1122에 정의된 약한 ES 모델에 해당합니다. 이 설정에서는 멀티홈 호스트가 비대칭 경로 지정을 사용합니다.
`src-priority`	기본 경로를 사용하여 패킷 경로 지정을 구성합니다. 경로 지정 테이블에 대상 경로가 여러 개 있는 경우 기본 경로는 송신 패킷의 IP 소스 주소가 구성된 인터페이스를 사용하는 경로입니다. 해당 경로가 없는 경우 송신 패킷은 패킷의 IP 대상에 대한 가장 긴 일치 경로를 사용합니다.

3 **(옵션) `hostmodel` 등록 정보의 설정을 확인합니다.**

```
# ipadm show-prop protocol
```

예 9–7 멀티홈 호스트에 대칭 경로 지정 설정

이 예에서는 멀티홈 호스트에서 모든 IP 트래픽의 대칭 경로 지정을 적용하려고 합니다.

```
# ipadm set-prop -p hostmodel=strong ip
# ipadm show-prop -p hostmodel ip
PROTO  PROPERTY   PERM  CURRENT  PERSISTENT  DEFAULT   POSSIBLE
ipv6   hostmodel  rw    strong   --          weak      strong,
                                                      src-priority,
                                                      weak
ipv4   hostmodel  rw    strong   --          weak      strong,
                                                      src-priority,
                                                      weak
```

IP 인터페이스 및 주소 모니터링

또한 `ipadm` 명령은 IP 인터페이스와 해당 등록 정보나 매개변수에 대한 정보를 모니터하고 가져오는 기본 도구입니다. 인터페이스 정보를 가져오는 `ipadm` 하위 명령은 다음 기본 구문을 사용합니다.

`ipadm show-*` [*other-arguments*] [*interface*]

- 인터페이스 정보를 가져오려면 `ipadm show-if`를 사용합니다.
- 주소 정보를 가져오려면 `ipadm show-addr`을 사용합니다.
- 특정 인터페이스 등록 정보에 대한 정보를 가져오려면 `ipadm show-ifprop`을 사용합니다.
- 특정 주소 등록 정보에 대한 정보를 가져오려면 `ipadm show-addrprop`을 사용합니다.

이 절에서는 `ipadm` 명령을 사용하여 네트워크 인터페이스 정보를 가져오는 방법의 여러 예를 제공합니다. 네트워크에서 수행하는 기타 모니터링 작업 유형의 경우 **Oracle Solaris 관리: IP 서비스**의 5 장, “TCP/IP 네트워크 관리”를 참조하십시오.

주 – `ipadm show-*` 명령의 모든 필드에 대한 자세한 내용은 `ipadm(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 네트워크 인터페이스 정보를 가져오는 방법

이 절차에서는 인터페이스의 일반 상태, 주소 정보 및 IP 등록 정보에 대한 정보를 표시하는 방법에 대해 설명합니다.

1 관리자로 전환합니다.

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

2 인터페이스에 대한 상태 정보를 가져오려면 다음 명령을 입력합니다.

```
# ipadm show-if [interface]
```

인터페이스를 지정하지 않으면 시스템의 모든 인터페이스에 대한 정보가 제공됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

`IFNAME` 정보가 표시되는 인터페이스를 나타냅니다.

`CLASS` 다음 네 가지 중 하나일 수 있는 인터페이스 클래스를 나타냅니다.

- `ip`는 IP 인터페이스를 나타냅니다.
- `ipmp`는 IPMP 인터페이스를 나타냅니다.
- `vni`는 가상 인터페이스를 나타냅니다.
- `loopback`은 자동으로 생성되는 루프백 인터페이스를 나타냅니다. 루프백 인터페이스를 제외하고 나머지 세 인터페이스 클래스는 수동으로 만들 수 있습니다.

`STATE` `ok`, `offline`, `failed`, `down` 또는 `disabled`일 수 있는 인터페이스 상태를 나타냅니다.

`failed` 상태는 IPMP 그룹에 적용되며 작동 중지되고 트래픽을 호스트할 수 없는 데이터 링크 또는 IP 인터페이스를 나타낼 수 있습니다. IP 인터페이스가 IPMP 그룹에 속하는 경우 IPMP 인터페이스가 그룹의 다른 활성 IP 인터페이스를 사용하여 계속 트래픽을 받고 보낼 수 있습니다.

`down` 상태는 관리자가 오프라인 상태로 전환한 IP 인터페이스를 나타냅니다.

`disable` 상태는 `ipadm disable-if` 명령을 사용하여 연결 취소된 IP 인터페이스를 나타냅니다.

ACTIVE 인터페이스가 트래픽을 호스트하는 데 사용되는지 여부를 나타내며 yes 또는 no로 설정됩니다.

OVER 인터페이스의 IPMP 클래스에만 적용되며 IPMP 인터페이스 또는 그룹을 구성하는 기본 인터페이스를 나타냅니다.

3 인터페이스에 대한 주소 정보를 가져오려면 다음 명령을 입력합니다.

```
# ipadm show-addr [addrobj]
```

주소 식별자를 지정하지 않으면 시스템의 모든 주소 식별자에 대한 주소 정보가 제공됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

ADDROBJ 주소가 나열되는 주소 객체를 지정합니다.

TYPE IP 주소가 static, dhcp 또는 addrconf인지를 나타냅니다. addrconf 설정은 Stateless 또는 Stateful 주소 구성을 사용하여 주소를 가져왔음을 나타냅니다.

STATE 실제 활성 구성의 주소 객체에 대해 설명합니다. 이러한 값의 전체 목록은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

ADDR 인터페이스에 구성되는 IP 주소를 지정합니다. 주소는 IPv4 또는 IPv6일 수 있습니다. 터널 인터페이스는 로컬 및 원격 주소를 모두 표시합니다.

터널에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 6 장, "IP 터널 구성"을 참조하십시오.

4 인터페이스 등록 정보에 대한 정보를 가져오려면 다음 명령을 입력합니다.

```
# ipadm show-ifprop [-p property] interface
```

등록 정보를 지정하지 않으면 모든 등록 정보와 해당 설정이 표시됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

IFNAME 정보가 표시되는 인터페이스를 나타냅니다.

PROPERTY 인터페이스의 등록 정보를 나타냅니다. 한 인터페이스에 등록 정보가 여러 개 있을 수 있습니다.

PROTO 등록 정보가 적용되고 IPv4 또는 IPv6일 수 있는 프로토콜을 나타냅니다.

PERM 지정된 등록 정보에 대해 허용된 권한을 나타내며 읽기 전용, 쓰기 전용 또는 둘 다일 수 있습니다.

CURRENT 활성 구성에서 등록 정보의 현재 설정을 나타냅니다.

PERSISTENT 시스템을 재부트할 때 재적용되는 등록 정보의 설정을 나타냅니다.

DEFAULT 지정한 등록 정보의 기본 설정을 나타냅니다.

POSSIBLE 지정한 등록 정보에 할당될 수 있는 값의 목록을 나타냅니다. 숫자 설정의 경우 허용되는 값의 범위가 표시됩니다.

주 – 정보가 요청되는 등록 정보를 인터페이스가 지원하지 않는 경우와 같이 필드 값을 알 수 없는 경우 설정이 물음표(?)로 표시됩니다.

---

5 주소 등록 정보에 대한 정보를 가져오려면 다음 명령을 입력합니다.

```
# ipadm show-addrprop [-p property,...] [addrobj]
```

표시되는 정보는 사용하는 옵션에 따라 달라집니다.

- 등록 정보를 지정하지 않으면 모든 등록 정보가 나열됩니다.
- 등록 정보만 지정하면 모든 주소의 해당 등록 정보가 표시됩니다.
- 주소 객체만 지정하면 시스템에 있는 모든 기존 주소의 등록 정보가 표시됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

ADDROBJ 등록 정보가 나열되는 주소 객체를 나타냅니다.

PROPERTY 주소 객체의 등록 정보를 나타냅니다. 한 주소 객체에 등록 정보가 여러 개 있을 수 있습니다.

PERM 지정된 등록 정보에 대해 허용된 권한을 나타내며 읽기 전용, 쓰기 전용 또는 둘 다일 수 있습니다.

CURRENT 현재 구성에서 등록 정보의 실제 설정을 나타냅니다.

PERSISTENT 시스템을 재부트할 때 재적용되는 등록 정보의 설정을 나타냅니다.

DEFAULT 지정한 등록 정보의 기본 설정을 나타냅니다.

POSSIBLE 지정한 등록 정보에 할당될 수 있는 설정의 목록을 나타냅니다. 숫자 설정의 경우 허용되는 값의 범위가 표시됩니다.

예 9–8 ipadm 명령을 사용하여 인터페이스 모니터

이 예 세트에서는 `ipadm show-*` 하위 명령을 사용하여 가져올 수 있는 정보 유형을 보여줍니다. 먼저 일반 인터페이스 정보가 표시됩니다. 그런 다음 주소 정보가 제공됩니다. 최종적으로, `net1` 인터페이스의 특정 등록 정보 `MTU`에 대한 정보가 제공됩니다. 이 예에는 터널 인터페이스 및 사용자 정의 이름을 사용하는 인터페이스가 포함됩니다.

```
# ipadm show-if
IFNAME       CLASS        STATE     ACTIVE     OVER
lo0          loopback     ok        yes        --
net0         ip           ok        yes        --
net1         ip           ok        yes        --
tun0         ip           ok        yes        --
```

```
# ipadm show-addr
ADDROBJ           TYPE       STATE     ADDR
lo0/?             static     ok        127.0.0.1/8
net0/v4           static     ok        192.168.84.3/24
tun0/v4tunaddr    static     ok        173.129.134.1-->173.129.134.2
```

interface/?로 나열된 주소 객체는 libipadm API를 사용하지 않는 응용 프로그램이 인터페이스에 주소를 구성했음을 나타냅니다. 이러한 응용 프로그램은 ipadm 명령의 제어를 받지 않습니다. 이 명령을 사용하려면 주소 객체 이름이 *interface*/*user-defined-string* 형식을 사용해야 합니다. IP 주소 할당의 예는 164 페이지 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

```
# ipadm show-ifprop -p mtu net1
IFNAME  PROPERTY  PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net1    mtu       ipv4   rw    1500      --         1500     68-1500
net1    mtu       ipv6   rw    1500      --         1500     1280-1500

# ipadm show-addrprop net1/v4
ADDROBJ          PROPERTY    PERM  CURRENT          PERSISTENT  DEFAULT          POSSIBLE
net1/v4   broadcast   r-    192.168.84.255  --          192.168.84.255   --
net1/v4   deprecated  rw    off             --          off              on,off
net1/v4   prefixlen   rw    24              24          24               1-30,32
net1/v4   private     rw    off             --          off              on,off
net1/v4   transmit    rw    on              --          on               on,off
net1/v4   zone        rw    global          --          global           --
```

인터페이스 구성 문제 해결

이 절에서는 ipadm 명령을 사용하여 IP 인터페이스를 구성하는 동안 발생할 수 있는 일반적인 문제에 대해 설명합니다.

ipadm 명령이 작동하지 않습니다.

dladm 및 ipadm 명령을 사용한 수동 IP 인터페이스 구성은 DefaultFixed와 같은 수정된 유형의 NCP(네트워크 구성 프로파일)에서만 작동합니다. 시스템의 활성 NCP가 자동 유형 프로파일인 경우 dladm 및 ipadm 명령을 사용하기 전에 수정된 유형 프로파일로 전환합니다.

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

# netadm enable -p ncp defaultfixed
```

ipadm create-addr 명령을 사용하여 IP 주소를 할당할 수 없습니다.

기존의 `ifconfig` 명령을 사용하면 단일 명령 구문으로 IP 주소를 연결하고 할당할 수 있습니다. `ipadm create-addr` 명령을 사용하여 IP 주소를 구성하는 경우 먼저 별도의 명령으로 IP 인터페이스를 만들어야 합니다.

```
# ipadm create-ip interface
# ipadm create-addr -T addr-type -a address addrobj
```

IP 주소를 구성하는 동안 cannot create address object: Invalid argument provided 메시지가 표시됨

주소 객체는 IP 인터페이스에 바인딩된 특정 IP 주소를 식별합니다. 이 객체는 IP 인터페이스의 각 IP 주소에 대한 고유 식별자입니다. 동일한 IP 인터페이스에 할당할 두번째 IP 주소를 식별하려면 다른 주소 객체를 지정해야 합니다. 동일한 주소 객체 이름을 사용하려면 다른 IP 주소를 식별하기 위해 할당하기 전에 주소 객체의 첫번째 인스턴스를 삭제해야 합니다.

```
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADR
lo0       static   ok      127.0.0.1/10
net0/v4   static   ok      192.168.10.1

# ipadm create-addr -T static -a 192.168.10.5 net0/v4b
```

또는

```
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADR
lo0       static   ok      127.0.0.1/10
net0/v4   static   ok      192.168.10.1

# ipadm delete-addr net0/v4
# ipadm create-addr -T static -a 192.168.10.5 net0/v4
```

IP 인터페이스를 구성하는 동안 cannot create address: Persistent operation on temporary object 메시지가 표시됨

ipadm 명령은 지속 구성을 만듭니다. 구성 중인 IP 인터페이스가 임시 인터페이스로 생성된 경우 ipadm 명령을 사용하여 해당 인터페이스에 지속 설정을 구성할 수 없습니다. 구성 중인 인터페이스가 임시인지 확인한 후 해당 인터페이스를 삭제하고 지속 객체로 다시 만든 다음 구성을 계속합니다.

```
# ipadm show-if -o all
IFNAME   CLASS      STATE   ACTIVE   CURRENT         PERSISTENT   OVER
lo0      loopback   ok      yes      -m46-v------    46--         --
net0     ip         ok      yes      bm4---------    ----         --
```

PERSISTENT 필드에 IPv4 구성을 나타내는 4 플래그나 IPv6 구성을 나타내는 6 플래그가 없을 경우 net0이 임시 인터페이스로 생성된 것입니다.

```
# ipadm delete-ip net0
# ipadm create-ip net0
# # ipadm create-addr -T static -a 192.168.1.10 net0/v4
```

비교 테이블: ipadm 명령 및 기타 네트워킹 명령

ipadm 명령은 IP 인터페이스의 모든 구성 작업에 사용할 기본 도구입니다. 이 명령은 ifconfig 및 ndd 명령과 같이 네트워크 구성에 사용된 이전 릴리스의 명령을 대체합니다. 다음 표에서는 이러한 이전 도구의 선택한 명령 옵션 및 ipadm 명령의 동등한 옵션을 보여줍니다.

주 – ipadm 옵션의 전체 목록은 제공되지 않습니다. 전체 목록은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

---

ifconfig 명령 옵션 및 ipadm 명령 옵션

다음 표에서는 ifconfig 명령 옵션과 대략적인 해당 ipadm 하위 명령을 보여줍니다.

표 9–3 ifconfig 및 ipadm 명령 간의 구문 매핑

ifconfig 명령	ipadm 명령
plumb/unplumb	ipadm create-ip ipadm create-vni ipadm create-ipmp ipadm enable-addr ipadm delete-ip ipadm delete-vni ipadm delete-ipmp ipadm disable-addr
[*address*[/*prefix-length*] [*dest-address*]] [addif *address*[*prefix-length*]] [removeif *address*[*prefix-length*]][netmask *mask*][destination *dest-address*]{auto-dhcp|dhcp}[primary][wait *seconds*]extend | release | start	ipadm create-addr -T static ipadm create-addr -T dhcp ipadm create-addr -T addrconf ipadm show-addr ipadm delete-addr ipadm refresh-addr
[deprecated | -deprecated] [preferred | -preferred] [private | -private] [zone *zonename* | -zones | -all-zones][xmit | -xmit]	ipadm set-addprop ipadm reset-addprop ipadm show-addprop
up	ipadm up-addr
down	ipadm down-addr
[metric *n*] [mtu *n*] [nud | -nud] [arp | -arp] [usesrc [*name* | none] [router | -router]	ipadm set-ifprop ipadm show-ifprop ipadm reset-ifprop
[ipmp] [group [*name* | ""]] standby | -standby] [failover | -failover]	ipadm create-ipmp ipadm delete-ipmp ipadm add-ipmp ipadm remove-ipmp ipadm set-ifprop -p [standby] [group]

표 9–3 ifconfig 및 ipadm 명령 간의 구문 매핑 (계속)

ifconfig 명령	ipadm 명령
[tdst *tunnel-dest-addr*] [tsrc *tunnel-srcs-addr*] [encaplimit *n*| -encaplimit] [thoplimit *n*]	dladm *-iptun 명령 세트. 자세한 내용은 dladm(1M) 매뉴얼 페이지와 **Oracle Solaris 관리: IP 서비스**의 "dladm 명령을 통한 터널 구성 및 관리"를 참조하십시오.
[auth_algs *authentication algorithm*] [encr_algs *encryption algorithm*] [encr_auth_algs *encryption authentication algorithm*]	ipsecconf 자세한 내용은 ipsecconf(1M) 및 **Oracle Solaris 관리: IP 서비스**의 15 장, "IPsec 구성(작업)"을 참조하십시오.
[auth_revarp] [ether [*address*]] [index *if-index*] [subnet *subnet-address*] [broadcast *broadcast-address*] [token *address /prefix-length*] dhcp 옵션 – inform, ping, release, status, drop	현재 사용할 수 있는 동등한 하위 명령은 없습니다.
modlist] [modinsert *mod_name@ pos*] [modremove *mod_name@pos*]	현재 사용할 수 있는 동등한 하위 명령은 없습니다.

ndd 명령 옵션 및 ipadm 명령 옵션

다음 표에서는 ndd 명령 옵션과 대략적인 해당 ipadm 하위 명령을 보여줍니다.

표 9–4 ndd 및 ipadm 명령 간의 구문 매핑

ndd 명령	ipadm 명령
등록 정보 검색	

표 9–4 ndd 및 ipadm 명령 간의 구문 매핑 *(계속)*

ndd 명령	ipadm 명령
bash-3.2# **ndd -get /dev/ip ?** ip_def_ttl (read and write) ip6_def_hops (read and write) ip_forward_directed_broadcasts (read and write) ip_forwarding (read and write) ... ... bash-3.2# **ndd -get /dev/ip ** **ip_def_ttl** 100 bash-3.2# **ndd -get /dev/ip ** **ip6_def_hops** 255 bash-3.2# **ndd -get /dev/tcp ?** tcp_cwnd_max (read and write) tcp_strong_iss (read and write) tcp_time_wait_interval (read and write) tcp_tstamp_always (read and write) tcp_tstamp_if_wscale (read and write) ... ... bash-3.2# **ndd -get /dev/tcp ecn** 1 bash-3.2# **ndd -get /dev/tcp sack** 2	bash-3.2# **ipadm show-prop ip** PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE ipv4 forwarding rw off -- off on,off ipv4 ttl rw 255 -- 255 1-255 ipv6 forwarding rw off -- off on,off ipv6 hoplimit rw 255 -- 255 1-255 ... bash-3.2# **ipadm show-prop -p ttl,hoplimit ip** PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE ipv4 ttl rw 255 -- 255 1-255 ipv6 hoplimit rw 255 -- 255 1-255 bash-3.2# **ipadm show-prop tcp** PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE tcp ecn rw passive -- passive never,passive, active tcp extra_ rw 2049 2049,4045 2049,4045 1-65535 priv_ports tcp largest_ rw 65535 -- 65535 1024-65535 anon_port tcp recv_ rw 128000 -- 128000 2048-1073741824 maxbuf tcp sack rw active -- active never,passive, active tcp send_ rw 49152 -- 49152 4096-1073741824 maxbuf tcp smallest_ rw 32768 -- 32768 1024-65535 anon_port tcp smallest_ rw 1024 -- 1024 1024-32768 nonpriv_port ... ... bash-3.2# **ipadm show-prop -p ecn,sack tcp** PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE tcp ecn rw passive -- passive never,passive,active tcp sack rw active -- active never,passive,active
등록 정보 설정	

표 9–4 ndd 및 ipadm 명령 간의 구문 매핑 *(계속)*

ndd 명령	ipadm 명령
bash-3.2# **ndd -set /dev/ip ** **ip_def_ttl** 64 bash-3.2# **ndd -get /dev/ip ** **ip_def_ttl** 64	bash-3.2# **ipadm set-prop -p ttl=64 ipv4** bash-3.2# **ipadm show-prop -p ttl ip** PROTO PROPERTY FAMILY PERM VALUE DEFAULT POSSIBLE ip ttl inet rw 64 255 1-255 PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE ipv4 ttl rw 64 64 255 1-255 bash-3.2# **ipadm reset-prop -p ttl ip** bash-3.2# **ipadm show-prop -p ttl ip** PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE ipv4 ttl rw 255 255 255 1-255

◆ ◆ ◆ 10장

10 Oracle Solaris에서 무선 인터페이스 통신 구성

이 장에서는 Oracle Solaris에서 실행되는 랩탑에서 무선 인터페이스 통신을 구성하고 사용하는 방법에 대해 설명합니다. 다음 항목을 다룹니다.

- WiFi 인터페이스를 통한 통신
- WiFi 네트워크 찾기
- Oracle Solaris 시스템에서 WiFi 연결 및 사용
- 보안 WiFi 통신

WiFi 통신 작업 맵

작업	설명	수행 방법
시스템의 WiFi 통신을 계획합니다.	필요에 따라 WiFi를 지원하는 위치에 라우터를 포함하는 랩탑 또는 무선 네트워크 구성을 설치합니다.	191 페이지 “WiFi 통신을 위해 시스템을 준비하는 방법”
WiFi 네트워크에 연결합니다.	로컬 WiFi 네트워크를 설치하고 통신을 설정합니다.	192 페이지 “WiFi 네트워크에 연결하는 방법”
WiFi 링크에서 통신을 모니터합니다.	표준 Oracle Solaris 네트워킹 도구를 사용하여 WiFi 링크 상태를 확인합니다.	196 페이지 “WiFi 링크를 모니터하는 방법”
보안 WiFi 통신을 설정합니다.	WEP 키를 만들고 이 키를 사용하여 보안 WiFi 네트워크와 연결을 설정합니다.	198 페이지 “암호화된 WiFi 네트워크 연결을 설정하는 방법”

WiFi 인터페이스를 통한 통신

IEEE 802.11 사양은 LAN(Local Area Network)에 대한 무선 통신을 정의합니다. 이러한 사양과 사양에서 설명하는 네트워크를 총체적으로 *WiFi*라고 합니다. 이 용어는 Wi-Fi Alliance 무역 그룹의 상표입니다. WiFi 네트워크는 공급자와 잠재 클라이언트가 모두 쉽게 구성할 수 있습니다. 따라서 점차 인기가 증가하고 있으며 전세계에서 일반적으로 사용되고 있습니다. WiFi 네트워크는 휴대폰, TV 및 라디오와 동일한 전파 기술을 사용합니다.

Oracle Solaris에는 시스템을 WiFi 클라이언트로 구성할 수 있는 기능이 있습니다. 이 절에서는 `dladm` 명령의 WiFi 연결 옵션을 사용하여 랩탑 또는 홈 컴퓨터를 로컬 WiFi 네트워크에 연결하는 방법에 대해 설명합니다.

주 – WiFi 서버 또는 액세스 포인트를 구성하는 기능은 Oracle Solaris에 포함되어 있지 않습니다.

WiFi 네트워크 찾기

WiFi 네트워크는 일반적으로 다음 세 가지 변형으로 제공됩니다.

- 상용 WiFi 네트워크
- 공공 WiFi 네트워크
- 개인 WiFi 네트워크

WiFi가 서비스하는 위치를 **핫 스폿**이라고 합니다. 각 핫 스폿에는 액세스 포인트가 있습니다. **액세스 포인트**는 인터넷에 "유선"으로 연결된 라우터(예: 이더넷 또는 DSL)입니다. 인터넷 연결은 대체로 WISP(무선 인터넷 서비스 공급자) 또는 일반 ISP를 통해 제공됩니다.

상용 WiFi 네트워크

대부분의 호텔과 카페는 랩탑 컴퓨터를 소유한 고객에게 무선 인터넷 연결을 서비스로 제공합니다. 이러한 상용 핫 스폿은 해당 시설 내에 액세스 포인트가 있습니다. 액세스 포인트는 상용 위치를 서비스하는 WISP에 유선으로 연결된 라우터입니다. 일반적인 WISP에는 민간 공급자와 휴대폰 업체가 포함됩니다.

Oracle Solaris를 실행하는 랩탑을 사용하여 호텔이나 기타 상용 핫 스폿이 제공하는 WiFi 네트워크에 연결할 수 있습니다. 핫 스폿에서는 WiFi 네트워크에 연결하기 위한 지침을 요청합니다. 대체로 연결 프로세스에서 로그인 시 시작하는 브라우저에 키를 제공해야 합니다. 네트워크를 사용하기 위해 호텔이나 WISP에 요금을 지불해야 할 수도 있습니다.

대체로 인터넷 핫 스폿인 상용 위치에서 이 기능을 고객에게 알립니다. 또한 Wi-FiHotSpotList.com (`http://www.wi-fihotspotlist.com`) 등의 다양한 웹 사이트에서 무선 핫 스폿 목록을 찾을 수 있습니다.

공공 WiFi 네트워크

전세계 도시들은 시민들이 홈 시스템에서 액세스할 수 있는 무료 공공 WiFi 네트워크를 구성했습니다. 공공 WiFi는 전신주나 기타 옥외에 설치된 라디오 송신기를 사용하여 네트워크가 서비스하는 영역에 "망사형(mesh)"을 형성합니다. 이러한 송신기는 공공 WiFi 네트워크에 대한 액세스 포인트입니다. 해당 영역이 공공 WiFi 네트워크에서 서비스되는 경우 네트워크의 망사형에 홈이 포함될 수 있습니다.

공공 WiFi에 대한 액세스는 대체로 무료입니다. Oracle Solaris를 실행하는, 적절한 장비를 갖춘 랩탑이나 PC에서 공공 네트워크에 액세스할 수 있습니다. 시스템에서 공공 네트워크에 액세스할 때는 홈 라우터가 필요 없습니다. 하지만 공공 네트워크의 신호가 약한 영역에서는 홈 라우터를 구성하는 것이 좋습니다. 홈 라우터는 WiFi 네트워크를 통한 보안 연결이 필요한 경우에도 권장됩니다. 자세한 내용은 197 페이지 “보안 WiFi 통신”을 참조하십시오.

개인 WiFi 네트워크

WiFi 네트워크는 비교적 구성이 용이하므로 회사와 대학에서의 액세스를 직원이나 학생으로 제한하여 개인 WiFi 네트워크를 사용합니다. 일반적으로 개인 WiFi 네트워크에서는 연결할 때 키를 제공하거나 연결 후에 보안 VPN을 실행해야 합니다. 개인 네트워크에 연결하려면 Oracle Solaris를 실행하는, 올바른 장비를 갖춘 랩탑 또는 PC와 보안 기능 사용 권한이 필요합니다.

WiFi 통신 계획

시스템을 WiFi 네트워크에 연결하려면 먼저 다음 지침을 완료합니다.

▼ WiFi 통신을 위해 시스템을 준비하는 방법

1 시스템에 지원되는 WiFi 인터페이스를 장착합니다.

Oracle Solaris에서 지원하는 WiFi 카드(예: Atheros 칩셋을 지원하는 카드)가 시스템에 있어야 합니다. 현재 지원되는 드라이버 및 칩셋 목록은 Wireless Networking for OpenSolaris (`http://hub.opensolaris.org/bin/view/Community+Group+laptop/wireless`)를 참조하십시오.

시스템에 아직 인터페이스가 없는 경우 인터페이스 카드 설치에 대한 제조업체 지침을 따릅니다. 192 페이지 “WiFi 네트워크에 연결하는 방법” 절차 진행 중에 인터페이스 소프트웨어를 구성합니다.

2 상용, 공공 또는 개인 WiFi 네트워크가 서비스하는 장소에 시스템을 배치합니다.

시스템이 네트워크의 액세스 포인트 근처에 있어야 합니다. 일반적으로 상용 또는 개인 네트워크 핫스폿의 경우에는 이 점을 고려할 필요가 없습니다. 하지만 무료 공공 네트워크를 사용하려는 경우 해당 위치가 송신기 액세스 포인트 근처여야 합니다.

3 **(옵션) 추가 액세스 포인트로 사용되도록 무선 라우터를 설정합니다.**

해당 위치에서 WiFi 네트워크를 사용할 수 없는 경우 고유한 라우터를 설정합니다. 예를 들어, DSL 회선이 있는 경우 무선 라우터를 DSL 라우터에 연결합니다. 그러면 무선 라우터가 무선 장치의 액세스 포인트가 됩니다.

Oracle Solaris 시스템에서 WiFi 연결 및 사용

이 절에는 Oracle Solaris에서 실행되는 랩탑 또는 데스크탑 컴퓨터에 대한 WiFi 연결을 설정하고 모니터하는 작업이 포함되어 있습니다.

▼ WiFi 네트워크에 연결하는 방법

시작하기 전에 다음 절차에서는 191 페이지 "WiFi 통신을 위해 시스템을 준비하는 방법"의 지침을 준수했다고 가정합니다.

1 **관리자로 전환합니다.**

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

2 **사용 가능한 링크를 확인합니다.**

```
# dladm show-link
LINK        CLASS     MTU    STATE    BRIDGE     OVER
ath0        phys      1500   up       --         --
e1000g0     phys      1500   up       --         --
```

이 출력 결과 예는 두 개의 링크를 사용할 수 있음을 나타냅니다. `ath0` 링크는 WiFi 통신을 지원합니다. `e1000g` 링크는 시스템을 유선 네트워크에 연결합니다.

3 **WiFi 인터페이스를 구성합니다.**

다음 단계를 사용하여 인터페이스를 구성합니다.

- WiFi를 지원하는 인터페이스를 만듭니다.

  ```
  # ipadm create-ip ath0
  ```

- 링크가 연결(plumb)되었는지 확인합니다.

  ```
  # ipadm show-if
  IFNAME      CLASS       STATE     ACTIVE     OVER
  lo0         loopback    ok        yes        --
  e1000g0     ip          ok        yes        --
  ath0        ip          ok        yes        --
  ```

4 **사용 가능한 네트워크를 확인합니다.**

```
# dladm scan-wifi
 LINK       ESSID         BSSID/IBSSID       SEC     STRENGTH   MODE   SPEED
 ath0       net1        00:0e:38:49:01:d0  none     good       g      54Mb
```

```
ath0        net2          00:0e:38:49:02:f0  none     very weak  g       54Mb
ath0        net3          00:0d:ed:a5:47:e0  none     very good  g       54Mb
```

scan-wifi 명령의 출력 결과 예에는 현재 위치에서 사용 가능한 WiFi 네트워크에 대한 정보가 표시됩니다. 출력 결과에 포함되는 정보는 다음과 같습니다.

LINK	WiFi 연결에서 사용할 링크 이름입니다.
ESSID	Extended Service Set ID입니다. ESSID는 출력 결과 예에 나온 net1, net2, net3과 같은 WiFi 네트워크의 이름입니다.
BSSID/IBSSID	특정 ESSID의 고유 식별자인 Basic Service Set ID입니다. BSSID는 네트워크에 특정 ESSID를 제공하는 주변 액세스 포인트의 48비트 MAC 주소입니다.
SEC	네트워크 액세스에 필요한 보안 유형입니다. 값은 none 또는 WEP입니다. WEP에 대한 자세한 내용은 197 페이지 "보안 WiFi 통신"을 참조하십시오.
STRENGTH	해당 위치에서 사용할 수 있는 WiFi 네트워크의 라디오 신호 강도입니다.
MODE	네트워크에서 실행하는 802.11 프로토콜의 버전입니다. 모드는 a, b, g 또는 이러한 모드의 조합입니다.
SPEED	특정 네트워크의 초당 메가비트 속도입니다.

5 WiFi 네트워크에 연결합니다.

다음 중 하나를 수행합니다.

- 신호가 가장 강한 비보안 WiFi 네트워크에 연결합니다.

  ```
  # dladm connect-wifi
  ```

- ESSID를 지정하여 비보안 네트워크에 연결합니다.

  ```
  # dladm connect-wifi -e ESSID
  ```

  dladm의 connect-wifi 하위 명령에는 WiFi 네트워크에 연결하는 추가 옵션이 여러 개 있습니다. 자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

6 인터페이스의 IP 주소를 구성합니다.

다음 중 하나를 수행합니다.

- DHCP 서버에서 IP 주소를 받습니다.

  ```
  # ipadm create-addr -T dhcp addrobj
  ```

  여기서 *addrobj*는 이름 지정 규약 *interface*/*user-defined-string*을 사용합니다.

  WiFi 네트워크가 DHCP를 지원하지 않는 경우 다음 메시지가 수신됩니다.

```
ipadm: interface: interface does not exist or cannot be managed using DHCP
```

- 정적 IP 주소를 구성합니다.

  시스템에 전용 IP 주소가 있는 경우 이 옵션을 사용합니다.

  ```
  # ipadm create-addr -T static -a address addrobj
  ```

7 시스템이 연결되어 있는 WiFi 네트워크의 상태를 확인합니다.

```
# dladm show-wifi
LINK        STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
ath0        connected     net3          none    very good  g      36Mb
```

이 출력 결과 예는 시스템이 현재 net3 네트워크에 연결되어 있음을 나타냅니다. 이전 scan-wifi 출력 결과에는 사용 가능한 네트워크 중 net3의 신호가 가장 강한 것으로 표시되었습니다. 다른 네트워크를 직접 지정하지 않는 경우 dladm show-wifi 명령은 신호가 가장 강한 WiFi 네트워크를 자동으로 선택합니다.

8 WiFi 네트워크를 통해 인터넷에 액세스합니다.

시스템이 연결되어 있는 네트워크에 따라 다음 중 하나를 수행합니다.

- 액세스 포인트가 무료 서비스를 제공하는 경우 이제 선택한 브라우저나 응용 프로그램을 실행할 수 있습니다.
- 액세스 포인트가 요금을 지불해야 하는 상용 핫 스폿에 있는 경우 현재 위치에서 제공되는 지침을 따릅니다. 일반적으로 브라우저를 실행하고 키를 제공하고 네트워크 공급자에게 신용 카드 정보를 제공합니다.

9 세션을 종료합니다.

다음 중 하나를 수행합니다.

- WiFi 세션을 종료하지만 시스템이 계속 실행되도록 둡니다.

  ```
  # dladm disconnect-wifi
  ```

- 현재 여러 세션이 실행되고 있는 경우 특정 WiFi 세션을 종료합니다.

  ```
  # dladm disconnect-wifi link
  ```

  여기서 *link*는 세션에 사용된 인터페이스를 나타냅니다.

- WiFi 세션이 실행되는 동안 시스템을 정상적으로 종료합니다.

  ```
  # shutdown -g0 -i5
  ```

  shutdown 명령을 통해 시스템을 끄기 전에 WiFi 세션의 연결을 명시적으로 끊지 않아도 됩니다.

예 10–1 특정 WiFi 네트워크에 연결

다음 예에서는 인터넷 커피숍에서 Oracle Solaris를 실행하는 랩탑을 사용할 때 발생할 수 있는 일반적인 시나리오를 보여줍니다.

WiFi 링크를 사용할 수 있는지 여부를 확인합니다.

```
# dladm show-wifi
ath0            type: non-vlan    mtu: 1500         device: ath0
```

ath0 링크는 랩탑에 설치되어 있습니다. ath0 인터페이스를 구성하고 작동하는지 확인합니다.

```
# ipadm create-ip ath0
IFNAME     STATE    CURRENT     PERSISTENT
lo0        ok       -m-v-----46 ---
ath0       ok       bm-------46 -46
```

해당 위치에서 사용 가능한 WiFi 링크를 표시합니다.

```
# dladm scan-wifi
 LINK       ESSID         BSSID/IBSSID       SEC     STRENGTH   MODE   SPEED
 ath0       net1         00:0e:38:49:01:d0  none     weak       g      54Mb
 ath0       net2         00:0e:38:49:02:f0  none     very weak  g      54Mb
 ath0       net3         00:0d:ed:a5:47:e0  wep      very good  g      54Mb
 ath0       citinet      00:40:96:2a:56:b5  none     good       b      11Mb
```

출력 결과는 net3의 신호 상태가 가장 양호하다는 것을 나타냅니다. net3에는 커피숍 공급자가 요금을 청구하는 키가 필요합니다. citinet은 해당 지역에서 제공하는 무료 네트워크입니다.

citinet 네트워크에 연결합니다.

```
# dladm connect-wifi -e citinet
```

connect-wifi의 -e 옵션은 기본 WiFi 네트워크의 ESSID를 인수로 사용합니다. 이 명령의 인수는 무료 로컬 네트워크의 ESSID인 citinet입니다. dladm connect-wifi 명령은 WiFi 네트워크에 연결하는 여러 옵션을 제공합니다. 자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

WiFi 인터페이스의 IP 주소를 구성합니다.

```
# ipadm create-addr -T static -a 10.192.16.3/8 ath0/v4
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
e1000g0/v4        static   ok           129.146.69.34/24
ath0/v4static     static   ok           10.192.16.3/8
lo0/v6            static   ok           ::1/128
```

이 예에서는 정적 IP 주소 10.192.16.3/24가 랩탑에 구성되어 있다고 가정합니다.

```
# dladm show-wifi
LINK       STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
ath0       connected     citinet       none    good       g      11Mb
```

출력 결과는 이제 랩탑이 citinet 네트워크에 연결되었음을 나타냅니다.

```
# firefox
```

Firefox 브라우저의 홈 페이지가 표시됩니다.

브라우저나 다른 응용 프로그램을 실행하여 WiFi 네트워크를 통해 작업을 시작합니다.

```
# dladm disconnect-wifi
# dladm show-wifi
LINK        STATUS        ESSID         SEC     STRENGTH   MODE    SPEED
ath0        disconnected   --            --       --        --       --
```

show-wifi 출력 결과에서 ath0 링크가 WiFi 네트워크에서 연결이 끊어졌음을 확인합니다.

▼ WiFi 링크를 모니터하는 방법

이 절차에서는 표준 네트워킹 도구를 통해 WiFi 링크의 상태를 모니터하고 linkprop 하위 명령을 통해 링크 등록 정보를 변경하는 방법을 보여줍니다.

1 **관리자로 전환합니다.**

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

2 **192 페이지 "WiFi 네트워크에 연결하는 방법"에 설명된 대로 WiFi 네트워크에 연결합니다.**

3 **링크의 등록 정보를 봅니다.**

다음 구문을 사용하십시오.

```
# dladm show-linkprop interface
```

예를 들어, 다음 구문을 사용하여 ath0 링크에 설정된 연결의 상태를 표시합니다.

```
# dladm show-linkprop ath0
PROPERTY        VALUE           DEFAULT         POSSIBLE
channel         5               --              --
powermode       off             off             off,fast,max
radio           ?               on              on,off
speed           36               --              1,2,5.5,6,9,11,12,18,24,36,48,54
```

4 **링크에 고정 속도를 설정합니다.**

주의 – Oracle Solaris는 WiFi 연결의 최적 속도를 자동으로 선택합니다. 링크의 초기 속도를 수정하면 성능이 감소되거나 특정 WiFi 연결이 설정되지 않을 수 있습니다.

show-linkprop 출력 결과에 나열된 가능한 속도 값 중 하나로 링크 속도를 수정할 수 있습니다.

```
# dladm set-linkprop -p speed=value link
```

5 링크의 패킷 흐름을 확인합니다.

```
# netstat -I ath0 -i 5
   input   ath0      output       input  (Total)    output
packets errs  packets errs  colls  packets errs  packets errs  colls
317     0     106     0     0      2905    0     571     0     0
14      0     0       0     0      20      0     0       0     0
7       0     0       0     0      16      0     1       0     0
5       0     0       0     0      9       0     0       0     0
304     0     10      0     0      631     0     316     0     0
338     0     9       0     0      722     0     381     0     0
294     0     7       0     0      670     0     371     0     0
306     0     5       0     0      649     0     338     0     0
289     0     5       0     0      597     0     301     0     0
```

예 10–2 링크 속도 설정

이 예에서는 WiFi 네트워크에 연결한 후 링크 속도를 설정하는 방법을 보여줍니다.

```
# dladm show-linkprop -p speed ath0
PROPERTY        VALUE           DEFAULT         POSSIBLE
speed           24              --              1,2,5,6,9,11,12,18,24,36,48,54
# dladm set-linkprop -p speed=36 ath0

# dladm show-linkprop -p speed ath0
PROPERTY        VALUE           DEFAULT         POSSIBLE
speed           36              --              1,2,5,6,9,11,12,18,24,36,48,54
```

보안 WiFi 통신

전파 기술을 통해 WiFi 네트워크를 쉽게 사용할 수 있으며 여러 위치에서 사용자가 무료로 액세스할 수 있는 경우도 많습니다. 그 결과, WiFi 네트워크 연결이 안전하지 않을 수 있습니다. 하지만 특정 유형의 WiFi 연결은 더 안전합니다.

- 액세스가 제한된 개인 WiFi 네트워크에 연결

  회사나 대학에서 설정한 내부 네트워크와 같은 개인 네트워크는 올바른 보안 챌린지를 제공할 수 있는 사용자만 네트워크에 액세스할 수 있도록 제한합니다. 잠재적 사용자는 연결 시퀀스 도중 키를 제공하거나 보안 VPN을 통해 네트워크에 로그인해야 합니다.
- WiFi 네트워크에 대한 연결 암호화

보안 키를 사용하여 시스템과 WiFi 네트워크 간의 통신을 암호화할 수 있습니다. WiFi 네트워크에 대한 액세스 포인트는 보안 키 생성 기능이 있는 홈 또는 사무실 라우터여야 합니다. 시스템과 라우터는 보안 연결을 만들기 전에 키를 설정하고 공유합니다.

`dladm` 명령은 액세스 포인트를 통한 연결의 암호화를 위해 WEP(Wired Equivalent Privacy) 키를 사용할 수 있습니다. WEP 프로토콜은 무선 연결에 대한 IEEE 802.11 사양에서 정의됩니다. `dladm` 명령의 WEP 관련 옵션에 대한 자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 암호화된 WiFi 네트워크 연결을 설정하는 방법

다음 절차에서는 홈 시스템과 홈 라우터 간에 보안 통신을 설정하는 방법을 보여줍니다. 많은 유무선 홈 라우터에는 보안 키를 생성할 수 있는 암호화 기능이 있습니다. 이 절차에서는 이러한 라우터를 사용하며 해당 설명서를 사용할 수 있다고 가정합니다. 또한 시스템이 라우터에 이미 연결되어 있다고 가정합니다.

1 홈 라우터를 구성하기 위한 소프트웨어를 시작합니다.

지침은 제조업체 설명서를 참조하십시오. 일반적으로 라우터 제조업체는 라우터 구성을 위한 내부 웹 사이트 또는 그래픽 사용자 인터페이스를 제공합니다.

2 WEP 키의 값을 생성합니다.

라우터용 보안 키를 만들기 위한 제조업체 지침을 따릅니다. 라우터 구성 GUI에서 키에 대해 선택한 암호문을 제공하도록 요청할 수도 있습니다. 소프트웨어는 이 암호문을 사용하여 16진수 문자열을 생성합니다. 이 문자열은 대체로 길이가 5바이트 또는 13바이트입니다. 이 문자열이 WEP 키에 사용할 값이 됩니다.

3 키 구성을 적용하고 저장합니다.

지침은 제조업체 설명서를 참조하십시오.

4 관리자로 전환합니다.

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

5 WEP 키가 포함된 보안 객체를 만듭니다.

시스템에서 터미널 창을 열고 다음을 입력합니다.

```
# dladm create-secobj -c wep keyname
```

여기서 *keyname*은 키에 지정할 이름을 나타냅니다.

6 보안 객체에 WEP 키의 값을 제공합니다.

`create-secobj` 하위 명령이 키의 값을 요청하는 스크립트를 실행합니다.

```
provide value for keyname: 5 or 13 byte key
confirm value for keyname: retype key
```

이 값은 라우터에서 생성된 키입니다. 스크립트는 ASCII나 16진수의 5바이트 또는 13바이트 문자열을 키 값으로 허용합니다.

7 방금 만든 키의 컨텐츠를 확인합니다.

```
# dladm show-secobj
OBJECT              CLASS
keyname               wep
```

여기서 *keyname*은 보안 객체의 이름입니다.

8 WiFi 네트워크에 대해 암호화된 연결을 설정합니다.

```
# dladm connect-wifi -e network -k keyname interface
```

9 연결이 안전한지 확인합니다.

```
# dladm show-wifi
LINK       STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
ath0       connected     net1          wep     good       g      11Mb
```

`SEC` 제목 아래의 `wep` 값은 WEP 암호화가 연결에 적용되었음을 나타냅니다.

예 10–3 암호화된 WiFi 통신 설정

이 예에서는 이미 다음을 수행했다고 가정합니다.

- WEP 키를 만들 수 있는 홈 라우터에 시스템을 연결했습니다.
- 라우터 제조업체의 설명서에 따라 WEP 키를 만들었습니다.
- 시스템에 보안 객체를 만드는 데 사용할 수 있도록 키를 저장했습니다.

```
# dladm create-secobj -c wep mykey
provide value for mykey: *****
confirm value for mkey: *****
```

라우터에서 생성된 WEP 키를 제공할 때 입력한 값이 별표로 표시됩니다.

```
# dladm show-secobj
OBJECT              CLASS
 mykey              wep
# dladm connect-wifi -e citinet -k mykey ath0
```

이 명령은 보안 객체 `mykey`를 사용하여 WiFi 네트워크 `citinet`에 대한 암호화된 연결을 설정합니다.

```
# dladm show-wifi
LINK       STATUS        ESSID         SEC     STRENGTH   MODE   SPEED
```

```
ath0        connected     citinet        wep     good        g       36Mb
```

이 출력 결과는 WEP 암호화를 통해 citinet에 연결되었음을 나타냅니다.

◆ ◆ ◆ 11 장

11 브릿지 관리

이 장에서는 브릿지 및 브릿지 관리 방법에 대해 설명합니다.

이 장에서는 다음 항목을 다룹니다.

브릿징 개요

브릿지는 개별 네트워크 세그먼트를 연결하는 데 사용됩니다. 브릿지로 연결하면 연결된 네트워크 세그먼트가 단일 네트워크 세그먼트처럼 통신합니다. 브릿징은 네트워킹 스택의 데이터 링크 계층(L2)에 구현됩니다. 브릿지는 패킷 전달 방식을 사용하여 하위 네트워크를 함께 연결합니다.

브릿징과 경로 지정은 모두 네트워크의 리소스 위치에 대한 정보를 배포하는 데 사용될 수 있지만 여러 가지 차이점이 있습니다. 경로 지정은 IP 계층(L3)에 구현되고 경로 지정 프로토콜을 사용합니다. 데이터 링크 계층에서는 경로 지정 프로토콜이 사용되지 않습니다. 대신 브릿지에 연결되어 있는 링크에 수신된 네트워크 트래픽을 검사하여 전달된 패킷의 대상을 확인합니다.

패킷이 수신되면 소스 주소가 검사됩니다. 패킷의 소스 주소는 패킷이 전송된 노드를 패킷이 수신된 링크에 연결합니다. 그런 다음 수신된 패킷이 대상 주소와 동일한 주소를 사용하는 경우 브릿지가 링크를 통해 패킷을 해당 주소로 전달합니다.

소스 주소와 연결된 링크는 브릿징된 하위 네트워크의 다른 브릿지에 연결된 중간 링크일 수 있습니다. 시간이 경과하면 브릿징된 하위 네트워크의 모든 브릿지가 패킷을 지정된 노드로 보내는 링크를 "학습"합니다. 따라서 패킷의 대상 주소를 사용하여 hop 단위 브릿징을 통해 패킷을 최종 대상으로 보냅니다.

로컬 "링크 작동 중지" 알림은 지정된 링크의 모든 노드에 더 이상 연결할 수 없음을 나타냅니다. 이 경우 해당 링크에 대한 패킷 전달이 중지되며 이 링크를 통한 모든 전달 항목이 비워집니다. 또한 시간 경과에 따라 전달 항목이 오래됩니다. 링크를 복원하면

해당 링크를 통해 수신된 패킷이 새 항목으로 간주됩니다. 패킷의 소스 주소를 기반으로 하는 "학습" 프로세스가 다시 시작됩니다. 이 프로세스를 사용하면 주소가 대상 주소로 사용될 때 브릿지가 해당 링크를 통해 패킷을 제대로 전달할 수 있습니다.

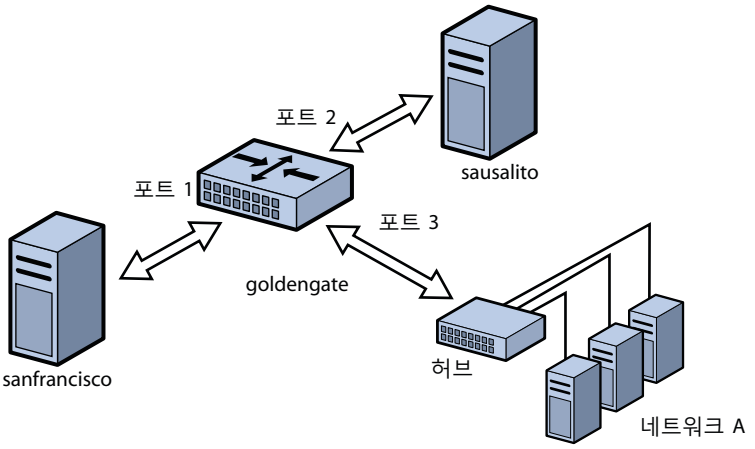
패킷을 대상으로 전달하려면 브릿지가 해당 브릿지에 연결된 모든 링크에서 무차별 모드로 수신 대기해야 합니다. 무차별 모드에서 수신 대기하면 브릿지에서 패킷이 전체 회선 속도로 무기한 순환하는 전달 루프가 발생할 수 있습니다. 따라서 브릿징은 STP(Spanning Tree Protocol) 방식을 사용하여 하위 네트워크를 사용할 수 없게 만드는 네트워크 루프를 방지합니다.

STP 및 RSTP(Rapid Spanning Tree Protocol)를 브릿지에 사용하는 것 외에도 Oracle Solaris는 향상된 TRILL 보호 기능을 지원합니다. 기본적으로 STP가 사용되지만 브릿징 명령에 `-P trill` 옵션을 지정하면 TRILL을 사용할 수 있습니다.

브릿지 구성을 사용하면 다양한 노드가 단일 네트워크로 연결되어 네트워크의 다양한 노드 관리가 단순화됩니다. 브릿지를 통해 이러한 세그먼트를 연결하면 모든 노드가 단일 브로드캐스트 네트워크를 공유합니다. 따라서 각 노드는 네트워크 세그먼트에 트래픽을 전달하기 위해 라우터를 사용하는 대신 IP와 같은 네트워크 트래픽을 사용하여 다른 노드에 연결할 수 있습니다. 브릿지를 사용하지 않는 경우 노드 간의 IP 트래픽 전달을 허용하도록 IP 경로 지정을 구성해야 합니다.

다음 그림에서는 브릿징된 단순 네트워크 구성을 보여줍니다. `goldengate` 브릿지는 브릿징이 구성되어 있는 Oracle Solaris 시스템입니다. `sanfrancisco` 및 `sausalito`는 브릿지에 물리적으로 연결된 시스템입니다. 네트워크 A에서 한쪽은 브릿지에, 다른 쪽은 컴퓨터 시스템에 물리적으로 연결된 허브를 사용합니다. 브릿지 포트는 `bge0`, `bge1` 및 `bge2`와 같은 링크입니다.

그림 11–1 브릿징된 단순 네트워크



브릿지 네트워크는 여러 브릿지를 물리적으로 연결하는 링으로 형성될 수 있습니다. 이러한 구성은 네트워크에서 일반적으로 사용됩니다. 이 유형의 구성을 사용하면 오래된 패킷이 링을 무기한 반복하여 네트워크 링크가 포화되는 문제가 발생할 수 있습니다. 이러한 반복 상태로부터 보호하기 위해 Oracle Solaris 브릿지는 STP 및 TRILL 프로토콜을 모두 구현합니다. 대부분의 하드웨어 브릿지는 STP 루프 방지도 구현합니다.

다음 그림에서는 링으로 구성된 브릿징된 네트워크를 보여줍니다. 이 구성에서는 세 개의 브릿지를 보여줍니다. `westminster`에는 두 시스템이 물리적으로 연결되어 있고, `waterloo`에는 한 시스템이 물리적으로 연결되어 있습니다. 또한 `tower`에도 한 시스템이 물리적으로 연결되어 있습니다. 각 브릿지는 브릿지 포트를 통해 서로 물리적으로 연결되어 있습니다.

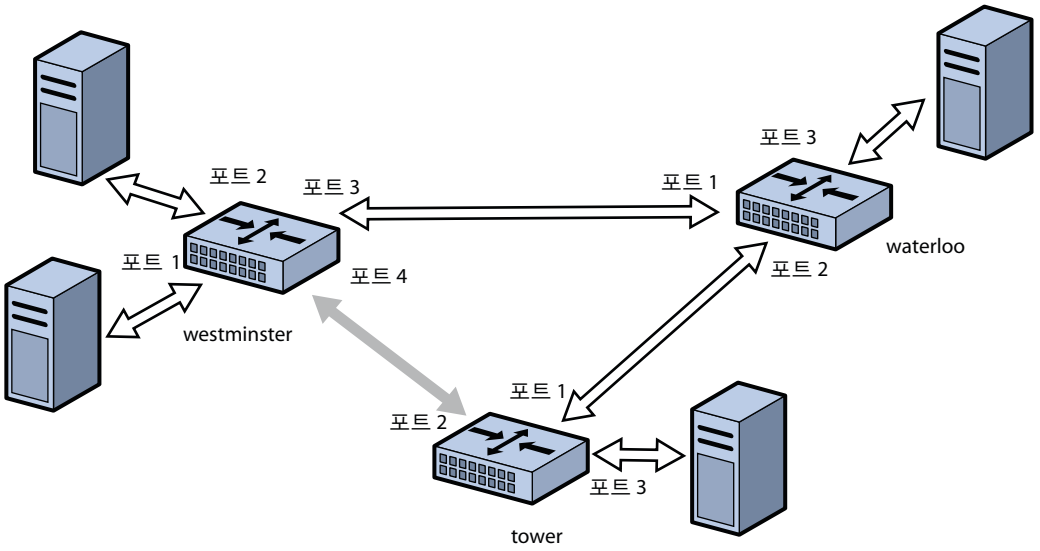
STP 또는 RSTP를 루프 방지에 사용하는 경우 루프의 연결 중 하나가 패킷을 전달할 수 없게 하여 물리적 루프를 줄여줍니다. 이 그림에서는 `westminster` 및 `tower` 브릿지 간의 물리적 링크가 패킷 전달에 사용되지 않음을 보여줍니다.

사용 가능한 물리적 링크를 종료하여 루프 방지를 수행하므로 STP 및 RSTP는 대역폭이 감소합니다.

STP 및 RSTP와 달리 TRILL은 루프 방지를 위해 물리적 링크를 종료하지 않습니다. 대신 TRILL은 네트워크의 각 TRILL 노드에 대해 최단 경로 정보를 계산하고 이 정보를 사용하여 패킷을 개별 대상으로 전달합니다.

따라서 TRILL을 사용하면 시스템의 **모든** 링크가 항상 사용될 수 있습니다. 루프는 IP에서 루프를 처리하는 것과 유사한 방식으로 처리되므로 문제가 되지 않습니다. 즉, TRILL은 필요에 따라 경로를 만들고 전달 hop 제한을 사용하여 일시적 루프 상태에 의한 문제를 방지합니다.

그림 11–2 브릿징된 네트워크 링



주의 – SPARC 플랫폼에서 `local-mac-address?=false`를 설정하지 **마십시오**. 설정하면 시스템이 여러 포트와 동일한 네트워크에서 동일한 MAC 주소를 잘못 사용하게 됩니다.

주 – 가능한 최고 레벨의 성능이 필요한 경우 링크를 브릿지로 구성하지 **마십시오**. 브릿징을 사용하려면 기본 인터페이스가 **반드시** 무차별 모드에 있어야 합니다. 이 경우 하드웨어, 드라이버 및 기타 시스템 계층에 있는 많은 중요한 최적화 기능이 사용 안함으로 설정됩니다. 이러한 성능 향상 기능이 사용 안함으로 설정되는 것은 브릿징 방식의 필연적인 결과입니다.

시스템 링크 중 **일부**가 브릿징되지 않아 이러한 제약 조건이 적용되지 않는 시스템에서 브릿지를 사용할 수 있습니다. 이러한 성능 문제는 브릿지의 일부로 구성된 링크에만 영향을 줍니다.

STP에 대한 자세한 내용은 IEEE 802.1D-1998을 참조하십시오. RSTP에 대한 자세한 내용은 IEEE 820.1Q-2004를 참조하십시오. TRILL에 대한 자세한 내용은 Internet Engineering Task Force (IETF) TRILL draft documents (`http://tools.ietf.org/wg/trill`)를 참조하십시오.

링크 등록 정보

dladm show-linkprop, dladm set-linkprop 및 reset-linkprop 명령을 사용하여 이러한 링크 등록 정보를 표시하고 수정할 수 있습니다.

default_tag
: 링크와 주고받는 태그 미지정된 패킷의 기본 VLAN(가상 LAN) ID를 정의합니다. 유효한 값은 0에서 4094 사이입니다. 기본값은 1입니다. 비VLAN 및 비VNIC(가상 네트워크 인터페이스 카드) 유형 링크에만 이 등록 정보가 있습니다. 이 값을 0으로 설정하면 포트와 주고받는 태그 미지정된 패킷이 전달되지 않습니다. 이것은 MAC 등록 정보입니다.

주 – 또한 브릿징 범위 외부에서 이 등록 정보를 사용하여 링크의 IEEE PVID(Port VLAN Identifier)를 지정합니다. default_tag가 0이 아닌 경우 기본 링크 자체가 PVID를 자동으로 나타내기 때문에 링크에 동일한 ID를 가진 VLAN을 만들 수 없습니다.

예를 들어, net0에서 PVID를 5로 설정한 경우 net0에 ID가 5인 VLAN을 만들 수 없습니다. 이 경우 VLAN 5를 지정하려면 net0을 사용합니다.

default_tag를 해당 링크에 생성된 기존 VLAN의 ID와 같도록 설정할 수 없습니다. 예를 들어, 다음 명령은 net0에 VLAN 22를 만듭니다.

```
# dladm create-vlan -l net0 -v 22 myvlan0
```

이 경우 default_tag를 22로 설정할 수 없습니다. 설정하면 net0과 myvlan0이 모두 동일한 VLAN을 나타내게 됩니다.

default_tag를 0으로 설정하여 net0의 태그 미지정된 패킷이 VLAN과 연결되지 않도록 합니다. 이 경우 구성된 브릿지가 해당 패킷을 전달할 수 없습니다.

---

forward
: 브릿지를 통한 트래픽 전달을 사용 및 사용 안함으로 설정합니다. 이 등록 정보는 VNIC 링크를 제외한 모든 링크에 있습니다. 유효한 값은 1(true) 및 0(false)입니다. 기본값은 1입니다. 사용 안함으로 설정한 경우 링크 인스턴스와 연결된 VLAN이 브릿지를 통해 트래픽을 전달하지 않습니다. 전달을 사용 안함으로 설정하는 것은 일반 브릿지에 대한 "allowed set(허용 설정)"에서 VLAN을 제거하는 것과 같습니다. 즉, 로컬 클라이언트에서 기본 링크로의 VLAN 기반 I/O가 계속되지만 브릿지 기반 전달은 수행되지 않습니다.

stp
: STP와 RSTP를 사용 및 사용 안함으로 설정합니다. 유효한 값은 1(true) 및 0(false)입니다. 기본값은 1이며, STP와 RSTP가 사용으로 설정됩니다. 0으로 설정하면 링크가 Spanning Tree Protocol 유형을

사용하지 않으며 항상 전달 모드로 설정됩니다. 전달 모드에서는 BPDU(Bridge Protocol Data Unit) 보호 기능을 사용합니다. 끝 노드에 연결된 P2P 연결을 구성하려면 STP 및 RSTP를 사용 안함으로 설정합니다. 비VLAN 및 비VNIC 유형 링크에만 이 등록 정보가 있습니다.

`stp_cost` 링크 사용 시의 STP 및 RSTP 비용 값을 나타냅니다. 유효한 값은 `1`에서 `65535` 사이입니다. 기본값은 `0`으로, 링크 유형별로 비용이 자동으로 계산됨을 나타냅니다. 다음 값은 여러 링크 유형의 비용을 나타냅니다. 10Mbps의 경우 `100`이고, 100Mbps의 경우 `19`이고, 1Gbps의 경우 `4`이고, 10Gbps의 경우 `2`입니다.

`stp_edge` 포트가 다른 브릿지에 연결되었는지 여부를 지정합니다. 유효한 값은 `1`(true) 및 `0`(false)입니다. 기본값은 `1`입니다. `0`으로 설정하면 데몬이 임의 유형의 BPDU가 표시되지는 않지만 포트가 다른 브릿지에 연결되었다고 가정합니다.

`stp_p2p` 연결 모드 유형을 지정합니다. 유효한 값은 `true`, `false` 및 `auto`입니다. 기본값은 `auto`이며, 지점 간 연결이 자동으로 검색됩니다. 지점 간 모드로 강제 설정하려면 `true`를 지정하고, 일반 다지점 모드를 강제 적용하려면 `false`를 지정합니다.

`stp_priority` STP 및 RSTP 포트 우선 순위 값을 설정합니다. 유효한 값은 `0`에서 `255` 사이입니다. 기본값은 `128`입니다. STP 및 RSTP 포트 우선 순위 값은 포트 식별자 앞에 이 값을 추가하여 브릿지의 기본 루트 포트를 결정하는 데 사용됩니다. 숫자 값이 작을수록 우선 순위가 더 높습니다.

STP 데몬

`dladm create-bridge` 명령을 사용하여 만든 각 브릿지는 동일한 이름을 가진 `svc:/network/bridge`의 SMF 인스턴스로 표시됩니다. 각 인스턴스는 STP를 구현하는 `/usr/lib/bridged` 데몬의 복사본을 실행합니다.

다음 명령 예에서는 `pontevecchio`라는 브릿지를 만듭니다.

```
# dladm create-bridge pontevecchio
```

시스템은 `svc:/network/bridge:pontevecchio`라는 SMF 서비스와 `/dev/net/pontevecchio0`이라는 관찰 노드를 만듭니다.

안전을 위해 모든 포트가 기본적으로 표준 STP를 실행합니다. STP와 같이 특정 형태의 브릿징 프로토콜을 실행하지 않는 브릿지는 네트워크에 오래 지속되는 전달 루프를 형성할 수 있습니다. 이더넷은 패킷에 hop 수 또는 TTL이 없기 때문에 이러한 루프가 네트워크에 치명적입니다.

특정 포트가 다른 브릿지에 연결되지 않은 것을 아는 경우(예: 호스트 시스템에 대한 직접 지점 간 연결), 관리상 해당 포트에 대해 STP를 사용 안함으로 설정할 수 있습니다. 브릿지의 모든 포트에서 STP가 사용 안함으로 설정된 경우에도 STP 데몬이 계속 실행됩니다. 데몬은 다음과 같은 이유로 계속 실행됩니다.

- 추가된 새 포트를 처리하기 위해
- BPDU 보호 기능을 구현하기 위해
- 필요한 경우 포트에서 전달을 사용 또는 사용 안함으로 설정하기 위해

포트에서 STP가 사용 안함으로 설정된 경우 `bridged` 데몬이 BPDU를 계속 수신 대기합니다(BPDU 보호). 이 데몬은 `syslog`를 사용하여 오류에 플래그를 지정하고 포트에서 전달을 사용 안함으로 설정하여 잘못된 네트워크 구성을 나타냅니다. 링크 상태가 작동 중지되었다가 다시 작동하거나 수동으로 링크를 제거하고 재추가하면 링크가 사용으로 재설정됩니다.

브릿지에 대해 SMF 서비스 인스턴스를 사용 안함으로 설정하면 STP 데몬이 중지될 때 해당 포트에서 브릿지 전달이 중지됩니다. 인스턴스를 다시 시작하면 STP가 초기 상태부터 시작됩니다.

TRILL 데몬

`dladm create-bridge -P trill` 명령을 사용하여 만든 각 브릿지는 동일한 이름을 가진 `svc:/network/bridge` 및 `svc:/network/routing/trill`의 SMF 인스턴스로 표시됩니다. `svc:/network/routing/trill`의 각 인스턴스는 TRILL 프로토콜을 구현하는 `/usr/lib/trilld` 데몬의 복사본을 실행합니다.

다음 명령 예에서는 `bridgeofsighs`라는 브릿지를 만듭니다.

```
# dladm create-bridge -P trill bridgeofsighs
```

시스템은 `svc:/network/bridge:bridgeofsighs` 및 `svc:/network/routing/trill:bridgeofsighs`라는 두 개의 SMF 서비스를 만듭니다. 또한 시스템은 `/dev/net/bridgeofsighs0`이라는 관찰 노드를 만듭니다.

브릿지 디버깅

각 브릿지 인스턴스에 "관찰 노드"가 할당됩니다. 이 노드는 `/dev/net/` 디렉토리에 표시되며 브릿지 이름과 후행 `0`을 더한 값으로 이름이 지정됩니다.

관찰 노드는 `snoop` 및 `wireshark` 유틸리티에 사용됩니다. 이 노드는 자동으로 삭제되는 패킷 전송을 제외하고 표준 이더넷 인터페이스로 동작합니다. 관찰 노드 위에 IP를 연결할 수 없으며, 수동 옵션을 사용하지 않으면 바인드 요청(`DL_BIND_REQ`)을 수행할 수 없습니다.

사용할 경우 관찰 노드는 브릿지에서 처리되는 각 패킷의 수정되지 않은 복사본 한 개를 사용자가 사용할 수 있게 합니다. 이 동작은 일반 브릿지의 "모니터링" 포트와 유사하며 일반 DLPI "무차별 모드" 규칙이 적용됩니다. `pfmod` 또는 `snoop` 및 `wireshark` 유틸리티의 기능을 사용하여 VLAN ID를 기준으로 필터링할 수 있습니다.

전달된 패킷은 브릿지에 수신된 데이터를 나타냅니다.

주의 – 브릿징 프로세스가 VLAN 태그를 추가, 제거 또는 수정하는 경우 표시된 데이터에서 이 프로세스가 발생하기 전의 상태를 설명합니다. 드물긴 하지만 이 경우 여러 링크에서 사용되는 고유한 `default_tag` 값이 있으면 혼동을 줄 수 있습니다.

브릿징 프로세스가 완료된 후 특정 링크에서 전송 및 수신되는 패킷을 보려면 브릿지 관찰 노드 대신 개별 링크에서 `snoop`를 실행합니다.

관찰 노드에 대한 자세한 내용은 320 페이지 “네트워크 가상화 및 리소스 제어의 관찰 기능”을 참조하십시오.

기타 브릿지 동작

다음 절에서는 구성에 브릿지를 사용하는 경우 링크 동작이 어떻게 변경되는지에 대해 설명합니다.

표준 링크 동작에 대한 자세한 내용은 231 페이지 “VLAN(가상 LAN) 관리”를 참조하십시오.

DLPI 동작

다음은 브릿지를 사용으로 설정한 경우 링크 동작의 차이점에 대해 설명합니다.

- 링크 작동(`DL_NOTE_LINK_UP`) 및 링크 작동 중지(`DL_NOTE_LINK_DOWN`) 알림은 통합되어 전달됩니다. 즉, 모든 외부 링크가 링크 작동 중지 상태를 표시하는 경우 MAC 계층을 사용하는 상위 레벨 클라이언트도 링크 작동 중지 이벤트를 표시합니다. 브릿지의 외부 링크가 링크 작동 상태를 표시하는 경우 상위 레벨 클라이언트도 모두 링크 작동을 표시합니다.

이 통합 링크 작동 및 링크 작동 중지 보고는 다음과 같은 이유로 수행됩니다.

- 링크 작동 중지가 표시되는 경우 링크의 노드에 더 이상 연결할 수 없습니다. 브릿징 코드에서 다른 링크를 통해 패킷을 보내고 받을 수 있는 경우에는 해당하지 않습니다. 링크의 실제 상태가 필요한 관리 응용 프로그램은 기존의 MAC 계층 커널 통계를 사용하여 상태를 표시할 수 있습니다. 이러한 응용 프로그램은 하드웨어 상태 정보를 보고하며 전달에 참여하지 않는다는 점에서 IP 등의 일반 클라이언트와 다릅니다.
- 모든 외부 링크가 작동 중지된 경우 브릿지 자체가 종료된 것처럼 상태가 표시됩니다. 이 특수 사례에서는 시스템에서 연결할 수 있는 항목이 없다고 인식합니다. 단점은 모든 인터페이스가 "실제"(가상 아님)이고 모두 연결 해제된 경우 브릿지를 사용하여 로컬 전용 통신을 허용할 수 없다는 것입니다.

- 링크 관련 기능은 모두 일반적으로 생성됩니다. 특수 하드웨어 가속 기능을 지원하는 링크는 클라이언트가 전적으로 실제 출력 링크를 결정하지 않으므로 이러한 기능을 사용할 수 없습니다. 브릿지 전달 기능이 대상 MAC 주소를 기준으로 출력 링크를 선택해야 하며, 이 출력 링크는 브릿지의 임의 링크일 수 있습니다.

VLAN 관리

기본적으로 시스템에 구성된 VLAN은 브릿지 인스턴스의 모든 포트에 전달됩니다. `dladm create-vlan` 또는 `dladm create-vnic -v` 명령을 호출하면 기본 링크가 브릿지의 일부인 경우 이 명령이 해당 브릿지 링크에서 지정된 VLAN의 전달도 사용으로 설정합니다.

링크에서 VLAN을 구성하고 브릿지의 다른 링크와 주고받는 전달을 사용 안함으로 설정하려면 `dladm set-linkprop` 명령으로 `forward` 등록 정보를 설정하여 전달을 사용 안함으로 설정해야 합니다.

기본 링크가 브릿지의 일부로 구성된 경우 브릿징에 대해 VLAN을 자동으로 사용으로 설정하려면 `dladm create-vlan` 명령을 사용합니다.

표준 준수 STP에서는 VLAN이 무시됩니다. 브릿징 프로토콜은 태그가 없는 BPDU 메시지를 사용하여 루프가 없는 토폴로지 한 개만 계산하고 이 트리를 사용하여 링크를 사용 및 사용 안함으로 설정합니다. 링크가 STP에 의해 자동으로 사용 안함으로 설정될 때 구성된 VLAN의 연결이 해제되지 않도록 네트워크에 프로비전된 중복 링크를 구성해야 합니다. 즉, 브릿징된 백본 전체에서 모든 VLAN을 실행하거나 모든 중복 링크를 신중하게 검사해야 합니다.

TRILL은 복잡한 STP 규칙을 따르지 않아도 됩니다. 대신 TRILL은 VLAN 태그가 유지되는 패킷을 자동으로 캡슐화하고 네트워크를 통해 전달합니다. 즉, TRILL은 브릿징된 단일 네트워크 내에서 동일한 VLAN ID가 재사용된, 격리된 VLAN을 바인딩합니다.

이것은 네트워크의 격리된 섹션에서 VLAN 태그를 재사용하여 4094 제한보다 큰 VLAN 세트를 관리할 수 있는 STP와 다른 중요한 차이점입니다. TRILL을 사용하여 이런 방식으로 네트워크를 관리할 수는 없지만 공급자 기반 VLAN과 같은 다른 솔루션을 구현할 수 있습니다.

VLAN이 있는 STP 네트워크의 경우 STP가 "잘못된" 링크를 사용 안함으로 설정할 때 VLAN 분할을 방지하도록 페일오버 특성을 구성하기 어려울 수 있습니다. TRILL 모델의 견고성은 격리된 VLAN에서의 비교적 적은 기능 손실을 보상하기에 충분합니다.

VLAN 동작

브릿지는 허용되는 VLAN 세트와 각 링크의 `default_tag` 등록 정보를 검사하여 전달을 수행합니다. 일반 프로세스는 다음과 같습니다.

- **입력 VLAN 결정.** 이 작업은 패킷이 링크에 수신될 때 시작됩니다. 패킷이 수신되면 VLAN 태그가 확인됩니다. 해당 태그가 없거나 태그가 우선 순위 전용(태그 0)이면 해당 링크에 구성된 `default_tag`(0으로 설정되지 않은 경우)가 내부 VLAN 태그로 사용됩니다. 태그가 없거나 0이고 `default_tag`가 0이면 패킷이 무시됩니다. 태그 미지정된 전달은 수행되지 않습니다. 태그가 있고 `default_tag`와 같은 경우에도 패킷이 무시됩니다. 그렇지 않으면 입력 태그가 입력 VLAN으로 사용됩니다.
- **링크 구성원 검사.** 입력 VLAN이 이 링크에서 허용된 VLAN으로 구성되지 않은 경우 패킷이 무시됩니다. 그런 다음 전달이 계산되고 동일한 검사가 출력 링크에 대해 수행됩니다.
- **태그 업데이트.** 출력 링크에서 VLAN(이 시점에는 0이 아님)이 `default_tag`와 같으면 패킷의 태그(있는 경우)가 우선 순위에 관계없이 제거됩니다. 출력 링크에서 VLAN이 `default_tag`와 다른 경우 현재 태그가 없으면 추가되고, 현재 우선 순위를 패킷에 복사하여 출력 패킷에 대한 태그가 설정됩니다.

주 – 전달 시 여러 인터페이스로 보내는 경우(브로드캐스트, 멀티캐스트 및 알 수 없는 대상) 출력 링크 검사와 태그 업데이트를 각 출력 링크에 대해 독립적으로 수행해야 합니다. 일부 전송에는 태그가 지정되고 다른 전송에는 태그가 지정되지 않을 수 있습니다.

브릿지 구성 예

다음 예에서는 브릿지 구성 및 브릿징 서비스에 대한 정보를 확인하는 방법을 보여줍니다.

- 다음 명령을 실행하여 브릿지에 대한 정보를 가져올 수 있습니다.

```
# dladm show-bridge
BRIDGE      PROTECT ADDRESS            PRIORITY DESROOT
tonowhere   trill   32768/66:ca:b0:39:31:5d 32768 32768/66:ca:b0:39:31:5d
sanluisrey  stp     32768/ee:2:63:ed:41:94 32768 32768/ee:2:63:ed:41:94
```

```
pontoon     trill   32768/56:db:46:be:b9:62 32768 32768/56:db:46:be:b9:62
```

- 다음 명령을 실행하여 브릿지에 대한 TRILL 별명 정보를 가져올 수 있습니다.

```
# dladm show-bridge -t tonowhere
NICK FLAGS LINK        NEXTHOP
38628 --   simblue2    56:db:46:be:b9:62
58753 L    --          --
```

브릿지 관리(작업 맵)

Oracle Solaris는 dladm 명령과 SMF 기능을 사용하여 브릿지를 관리합니다. SMF 명령을 사용하면 svc:/network/bridge 인스턴스의 FMRI(오류 관리 자원 식별자)를 통해 브릿지 인스턴스를 사용 및 사용 안함으로 설정하고 모니터할 수 있습니다. dladm 명령을 사용하면 브릿지를 만들거나 삭제하고 브릿지에 링크를 할당하거나 제거할 수 있습니다.

다음 표에서는 브릿지 관리에 사용할 수 있는 작업을 보여줍니다.

작업	설명	수행 방법
구성된 브릿지에 대한 정보를 확인합니다.	dladm show-bridge 명령을 사용하여 시스템에 구성된 브릿지에 대한 정보를 확인합니다. 구성된 브릿지, 링크, 통계 및 커널 전달 항목에 대한 정보를 확인할 수 있습니다.	212 페이지 "구성된 브릿지에 대한 정보를 확인하는 방법"
브릿지에 연결된 링크에 대한 구성 정보를 확인합니다.	dladm show-link 명령을 사용하여 시스템에 구성된 링크에 대한 정보를 확인합니다. 링크가 브릿지에 연결된 경우 BRIDGE 필드의 출력을 참조하십시오.	214 페이지 "브릿지 링크에 대한 구성 정보를 확인하는 방법"
브릿지를 만듭니다.	dladm create-bridge 명령을 사용하여 브릿지를 만들고 선택적 링크를 추가합니다. 기본적으로 브릿지는 STP를 사용하여 생성됩니다. 대신 TRILL을 사용하여 브릿지를 만들려면 dladm create-bridge 명령줄에 -P trill을 추가하거나 dladm modify-bridge 명령을 사용하여 TRILL을 사용으로 설정합니다.	214 페이지 "브릿지를 만드는 방법"

작업	설명	수행 방법
브릿지에 대한 보호 유형을 수정합니다.	`dladm modify-bridge` 명령을 사용하여 브릿지에 대한 보호 유형을 수정합니다. 기본적으로 브릿지는 STP를 사용하여 생성됩니다. 대신 TRILL을 사용하여 브릿지를 만들려면 `dladm modify-bridge` 명령에 `-P trill`을 사용하여 TRILL을 사용으로 설정합니다.	215 페이지 "브릿지에 대한 보호 유형을 수정하는 방법"
브릿지에 링크를 추가합니다.	`dladm add-bridge` 명령을 사용하여 기존 브릿지에 링크를 하나 이상 추가합니다.	216 페이지 "기존 브릿지에 링크를 하나 이상 추가하는 방법"
브릿지에서 링크를 제거합니다.	`dladm remove-bridge` 명령을 사용하여 브릿지에서 링크를 제거합니다. 모든 링크가 제거될 때까지 브릿지를 삭제할 수 없습니다.	216 페이지 "브릿지에서 링크를 제거하는 방법"
시스템에서 브릿지를 삭제합니다.	`dladm delete-bridge` 명령을 사용하여 시스템에서 브릿지를 삭제합니다.	217 페이지 "시스템에서 브릿지를 삭제하는 방법"

▼ 구성된 브릿지에 대한 정보를 확인하는 방법

이 절차에서는 `dladm show-bridge` 명령에 다양한 옵션을 사용하여 구성된 브릿지에 대한 여러 종류의 정보를 표시하는 방법을 보여줍니다.

`dladm show-bridge` 명령 옵션에 대한 자세한 내용은 `dladm`(1M) 매뉴얼 페이지를 참조하십시오.

1 관리자로 전환합니다.

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

2 특정 브릿지나 구성된 모든 브릿지에 대한 정보를 확인합니다.

- 브릿지 목록을 확인합니다.

  ```
  # dladm show-bridge
  ```

- 브릿지의 링크 관련 상태를 표시합니다.

  ```
  # dladm show-bridge -l bridge-name
  ```

- 브릿지에 대한 통계를 표시합니다.

  ```
  # dladm show-bridge -s bridge-name
  ```

주 – 보고된 통계의 이름과 정의는 변경될 수 있습니다.

- 브릿지에 대한 링크 관련 통계를 표시합니다.

  # dladm show-bridge -ls *bridge-name*
- 브릿지에 대한 커널 전달 항목을 표시합니다.

  # dladm show-bridge -f *bridge-name*
- 브릿지에 대한 TRILL 정보를 표시합니다.

  # dladm show-bridge -t *bridge-name*

예 11–1 브릿지 정보 보기

다음은 dladm show-bridge 명령에 다양한 옵션을 사용하는 예입니다.

- 다음은 시스템에 구성된 모든 브릿지에 대한 정보를 보여줍니다.

```
# dladm show-bridge
BRIDGE       PROTECT ADDRESS           PRIORITY DESROOT
goldengate   stp     32768/8:0:20:bf:f 32768    8192/0:d0:0:76:14:38
baybridge    stp     32768/8:0:20:e5:8 32768    8192/0:d0:0:76:14:38
```

- 다음 dladm show-bridge -l 명령은 단일 브릿지 인스턴스 tower에 대한 링크 관련 상태 정보를 보여줍니다. 구성된 매개변수를 보려면 dladm show-linkprop 명령을 대신 사용합니다.

```
# dladm show-bridge -l tower
LINK         STATE        UPTIME   DESROOT
hme0         forwarding   117      8192/0:d0:0:76:14:38
qfe1         forwarding   117      8192/0:d0:0:76:14:38
```

- 다음 dladm show-bridge -s 명령은 지정한 브릿지 terabithia에 대한 통계를 보여줍니다.

```
# dladm show-bridge -s terabithia
BRIDGE       DROPS        FORWARDS
terabithia   0            302
```

- 다음 dladm show-bridge -ls 명령은 지정한 브릿지 london의 모든 링크에 대한 통계를 보여줍니다.

```
# dladm show-bridge -ls london
LINK         DROPS     RECV      XMIT
hme0         0         360832    31797
qfe1         0         322311    356852
```

- 다음 dladm show-bridge -f 명령은 지정한 브릿지 avignon에 대한 커널 전달 항목을 보여줍니다.

```
# dladm show-bridge -f avignon
DEST              AGE     FLAGS  OUTPUT
8:0:20:bc:a7:dc   10.860  --     hme0
8:0:20:bf:f9:69   --      L      hme0
```

```
8:0:20:c0:20:26   17.420  --      hme0
8:0:20:e5:86:11   --      L       qfe1
```

- 다음 dladm show-bridge -t 명령은 지정한 브릿지 key에 대한 TRILL 정보를 보여줍니다.

```
# dladm show-bridge -t key
NICK FLAGS LINK        NEXTHOP
38628 --   london      56:db:46:be:b9:62
58753 L    --          --
```

▼ 브릿지 링크에 대한 구성 정보를 확인하는 방법

dladm show-link 출력 결과에는 BRIDGE 필드가 포함됩니다. 링크가 브릿지의 구성원이면 이 필드에 구성원으로 속한 브릿지의 이름이 식별됩니다. 이 필드는 기본적으로 표시됩니다. 브릿지에 속하지 않는 링크의 경우 -p 옵션을 사용하면 필드가 비어 있습니다. 그렇지 않으면 필드에 --가 표시됩니다.

또한 브릿지 관찰 노드는 dladm show-link 출력 결과에 별도의 링크로 표시됩니다. 이 노드의 경우 기존 OVER 필드에 브릿지의 구성원인 링크가 나열됩니다.

1 **관리자로 전환합니다.**

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

2 **브릿지의 구성원인 링크에 대한 구성 정보를 확인합니다.**

```
# dladm show-link [-p]
```

-p 옵션은 구문 분석 가능한 형식으로 출력을 생성합니다.

▼ 브릿지를 만드는 방법

이 절차에서는 STP를 사용하여 기본값인 브릿지를 만드는 방법을 보여줍니다. 브릿지 만들기 옵션에 대한 자세한 내용은 dladm(1M) 매뉴얼 페이지의 dladm create-bridge 설명을 참조하십시오.

주 – 대신 TRILL을 사용하여 브릿지를 만들려면 -dladm create-bridge 명령줄에 P trill을 추가하거나 dladm modify-bridge 명령을 사용하여 TRILL을 사용으로 설정합니다.

---

dladm create-bridge 명령은 브릿지 인스턴스를 만들고 선택적으로 하나 이상의 네트워크 링크를 새 브릿지에 할당합니다. 기본적으로 시스템에 브릿지 인스턴스가 없으므로 Oracle Solaris는 기본적으로 네트워크 링크 간을 브릿징하지 않습니다.

링크 간을 브릿징하려면 브릿지 인스턴스를 하나 이상 만들어야 합니다. 각 브릿지 인스턴스를 별개입니다. 브릿지 간에는 전달 연결이 없으며 링크가 최대 한 개 브릿지의 구성원입니다.

*bridge-name*은 적합한 SMF 서비스 인스턴스 이름이어야 하는 임의 문자열입니다. 이 이름은 이스케이프 시퀀스가 없는 FMRI 구성 요소입니다. 즉, 공백, ASCII 제어 문자 및 다음 문자가 포함될 수 없습니다.

```
; / ? : @ & = + $ , % < > # "
```

SUNW 문자열로 시작하는 모든 이름과 마찬가지로 이름 default는 예약되어 있습니다. 후행 숫자가 있는 이름은 "관찰 장치" 만들기에 예약되어 있습니다. 관찰 장치의 사용으로 인해 적합한 브릿지 인스턴스의 이름은 적합한 dlpi(7P) 이름으로 다시 제한됩니다. 이름은 알파벳 문자나 밑줄 문자로 시작하고 끝나야 합니다. 이름의 나머지 부분에는 영숫자와 밑줄 문자를 사용할 수 있습니다.

1 **관리자로 전환합니다.**

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

2 **브릿지를 만듭니다.**

```
# dladm create-bridge [-l link]... bridge-name
```

-l *link* 옵션은 브릿지에 링크를 추가합니다. 지정한 링크를 추가할 수 없는 경우 명령이 실패하고 브릿지가 생성되지 않습니다.

다음 예에서는 hme0 및 qfe1 링크를 연결하여 brooklyn 브릿지를 만드는 방법을 보여줍니다.

```
# dladm create-bridge -l hme0 -l qfe1 brooklyn
```

▼ 브릿지에 대한 보호 유형을 수정하는 방법

이 절차에서는 dladm modify-bridge 명령을 사용하여 보호 유형을 STP에서 TRILL로 또는 TRILL에서 STP로 수정하는 방법을 보여줍니다.

- **브릿지에 대한 보호 유형을 수정합니다.**

```
# dladm modify-bridge -P protection-type bridge-name
```

-P *protection-type* 옵션은 사용할 보호 유형을 지정합니다. 기본적으로 보호 유형은 STP(-P stp)입니다. TRILL 보호 유형을 대신 사용하려면 -P trill 옵션을 사용합니다.

다음 예에서는 brooklyn 브릿지에 대한 보호 유형을 기본값인 STP에서 TRILL로 수정하는 방법을 보여줍니다.

```
# dladm modify-bridge -P trill brooklyn
```

▼ 기존 브릿지에 링크를 하나 이상 추가하는 방법

이 절차에서는 브릿지 인스턴스에 링크를 하나 이상 추가하는 방법을 보여줍니다.

링크는 최대 한 개 브릿지의 구성원일 수 있습니다. 따라서 한 브릿지 인스턴스에서 다른 브릿지 인스턴스로 링크를 이동하려면 먼저 현재 브릿지에서 링크를 제거한 후 다른 인스턴스에 추가해야 합니다.

브릿지에 할당되는 링크는 VLAN, VNIC 또는 터널일 수 없습니다. 통합의 일부로 허용되는 링크 또는 통합 자체인 링크만 브릿지에 할당할 수 있습니다.

브릿지에 할당되는 링크는 모두 동일한 MTU 값을 가져야 합니다. Oracle Solaris에서는 기존 링크의 MTU 값을 변경할 수 있습니다. 이 경우 브릿지를 다시 시작하기 전에 MTU 값이 일치하도록 할당된 링크를 제거하거나 변경할 때까지 브릿지 인스턴스가 유지 관리 상태로 전환됩니다.

브릿지에 할당되는 링크는 802.3 및 802.11 매체를 포함하는 이더넷 유형이어야 합니다.

1 관리자로 전환합니다.

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

2 기존 브릿지에 새 링크를 추가합니다.

```
# dladm add-bridge -l new-link bridge-name
```

다음 예에서는 기존 브릿지 `rialto`에 `qfe2` 링크를 추가하는 방법을 보여줍니다.

```
# dladm add-bridge -l qfe2 rialto
```

▼ 브릿지에서 링크를 제거하는 방법

이 절차에서는 브릿지 인스턴스에서 링크를 하나 이상 제거하는 방법을 보여줍니다. 브릿지를 삭제하려는 경우 이 절차를 사용합니다. 브릿지를 삭제하려면 먼저 해당 링크를 모두 제거해야 합니다.

1 관리자로 전환합니다.

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

2 브릿지에서 링크를 제거합니다.

```
# dladm remove-bridge [-l link]... bridge-name
```

다음 예에서는 `charles` 브릿지에서 `hme0`, `qfe1` 및 `qfe2` 링크를 제거하는 방법을 보여줍니다.

```
# dladm remove-bridge -l hme0 -l qfe1 -l qfe2 charles
```

▼ 시스템에서 브릿지를 삭제하는 방법

이 절차에서는 브릿지 인스턴스를 삭제하는 방법을 보여줍니다. 브릿지를 삭제하려면 먼저 `dladm remove-bridge` 명령을 실행하여 연결된 링크를 모두 비활성화해야 합니다. 216 페이지 "브릿지에서 링크를 제거하는 방법"을 참조하십시오.

1 관리자로 전환합니다.

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

2 시스템에서 브릿지를 삭제합니다.

```
# dladm delete-bridge bridge-name
```

다음 예에서는 먼저 `coronado` 브릿지에서 `hme0`, `qfe1` 및 `qfe2` 링크를 제거한 다음 시스템에서 브릿지 자체를 제거하는 방법을 보여줍니다.

```
# dladm remove-bridge -l hme0 -l qfe1 -l qfe2 coronado
# dladm delete-bridge coronado
```

◆ ◆ ◆ 12 장

12 링크 통합 관리

이 장에서는 링크 통합을 구성하고 유지 관리하는 절차에 대해 설명합니다. 이 절차에는 유연한 링크 이름 지원과 같은 새 기능을 사용하는 단계가 포함되어 있습니다.

링크 통합 개요

Oracle Solaris는 네트워크 인터페이스를 링크 통합으로 구성하는 기능을 지원합니다. **링크 통합**는 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 할당할 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 12–1에서는 인기 있는 웹 사이트를 호스트하는 서버에 대한 통합을 보여줍니다. 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 `192.168.50.32`인 `aggr1` 통합입니다. 이 통합은 `bge0`에서 `bge2`까지의 인터페이스 세 개로 구성됩니다. 이러한 인터페이스는 고객 질의에 대한 응답으로 트래픽을 보내는 데만 사용됩니다. 모든 인터페이스에서 보낸 패킷 트래픽의 송신 주소는 `aggr1`의 IP 주소인 `192.168.50.32`입니다.

그림 12–1 기본 링크 통합 토폴로지

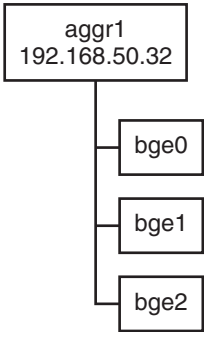
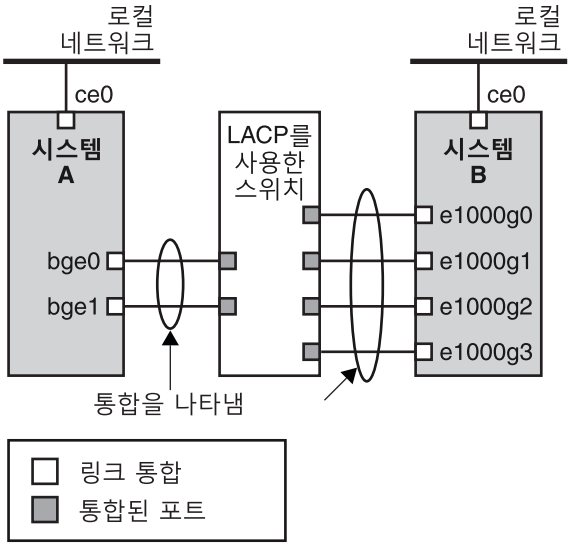


그림 12–2에서는 두 시스템이 포함된 로컬 네트워크를 보여주며, 각 시스템에 통합이 구성되어 있습니다. 두 시스템은 스위치로 연결되어 있습니다. 스위치를 통해 통합을 실행해야 하는 경우 해당 스위치가 통합 기술을 지원해야 합니다. 이 구성 유형은 특히 고가용성과 중복 시스템에 유용합니다.

이 그림에서 시스템 A에는 `bge0`과 `bge1`의 두 인터페이스로 구성된 통합이 있습니다. 이러한 인터페이스는 통합된 포트를 통해 스위치에 연결됩니다. 시스템 B에는 `e1000g0`에서 `e1000g3`까지 인터페이스 4개로 구성된 통합이 있습니다. 이러한 인터페이스도 스위치의 통합된 포트에 연결됩니다.

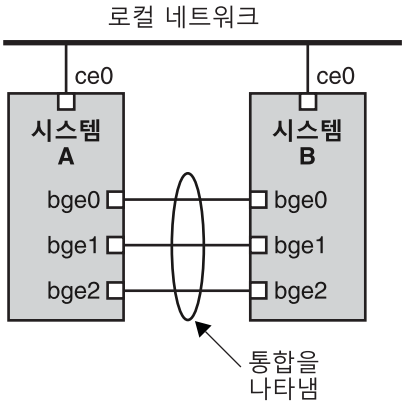
그림 12–2 스위치를 사용한 링크 통합 토폴로지



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 12–3 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 흐름에 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 흐름 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 임의 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.
- 생성된 인터페이스는 통합의 구성원이 될 수 없습니다.
- 통합의 모든 인터페이스가 동일한 속도 및 전이중 모드로 실행되어야 합니다.
- EEPROM 매개변수 `local-mac-address?`에서 MAC 주소의 값을 "true"로 설정해야 합니다. 지침은 인터페이스의 MAC 주소가 고유한지 확인하는 방법을 참조하십시오.

특정 장치는 링크 상태 알림을 지원하는 IEEE 802.3ad Link Aggregation Standard의 요구 사항을 충족하지 않습니다. 포트가 통합에 연결되거나 통합에서 분리되려면 이 지원이 있어야 합니다. 링크 상태 알림을 지원하지 않는 장치는 `dladm create-aggr` 명령의 `-f` 옵션을 사용해야만 통합할 수 있습니다. 이러한 장치의 경우 링크 상태가 항상 `UP`으로 보고됩니다. `-f` 옵션 사용에 대한 자세한 내용은 224 페이지 “링크 통합을 만드는 방법”을 참조하십시오.

링크 통합의 유연한 이름

링크 통합에 유연한 이름을 지정할 수 있습니다. 링크 통합에 의미 있는 임의 이름을 지정할 수 있습니다. 유연한 이름 및 사용자 정의 이름에 대한 자세한 내용은 24 페이지 “네트워크 장치 및 데이터 링크 이름”을 참조하십시오. 이전 Oracle Solaris 릴리스에서는 통합에 지정하는 **키**의 값으로 링크 통합을 식별합니다. 이 방법에 대한 설명은 링크 통합 개요를 참조하십시오. 해당 방법도 여전히 유효하지만 사용자 정의 이름을 사용하여 링크 통합을 식별하는 것이 좋습니다.

다른 모든 데이터 링크 구성과 유사하게, 링크 통합은 `dladm` 명령을 사용하여 관리됩니다.

링크 통합 관리(작업 맵)

다음 표에서는 링크 통합 관리 절차에 대한 링크를 제공합니다.

작업	설명	수행 방법
통합을 만듭니다.	여러 데이터 링크로 구성된 통합을 구성합니다.	224 페이지 “링크 통합을 만드는 방법”
통합을 수정합니다.	통합 정책과 모드를 변경합니다.	226 페이지 “통합을 수정하는 방법”

작업	설명	수행 방법
통합을 구성하는 링크를 수정합니다.	통합을 구성하는 데이터 링크 수를 늘리거나 줄입니다.	227 페이지 “통합에 링크를 추가하는 방법” 또는 228 페이지 “통합에서 링크를 제거하는 방법”
통합을 삭제합니다.	네트워크 구성에서 링크 통합을 완전히 제거합니다.	228 페이지 “통합을 삭제하는 방법”

▼ 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 관리자로 전환합니다.

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

2 네트워크 데이터 링크 정보를 표시합니다.

```
# dladm show-link
```

3 통합을 만드는 중인 링크는 아무 응용 프로그램에서도 열려 있지 않아야 합니다.

예를 들어, 링크에 IP 인터페이스를 만든 경우 인터페이스를 제거합니다.

a. 링크가 응용 프로그램에서 사용되고 있는지 확인하려면 `dladm show-link` 구문이나 `ipadm show-if` 구문의 출력 결과를 검사합니다.

- 데이터 링크가 사용 중인 경우 `dladm show-link` 출력 결과의 `STATE` 필드에 링크가 `up` 상태로 표시됩니다. 결과는 다음과 같습니다.

```
# dladm show-link
LINK      CLASS     MTU     STATE     BRIDGE   OVER
qfe3      phys      1500    up        --       --
```

- 데이터 링크가 사용 중인 경우 해당 링크의 IP 인터페이스가 `ipadm show-if` 구문의 출력 결과에 포함됩니다. 결과는 다음과 같습니다.

```
# ipadm show-if
IFNAME       CLASS        STATE     ACTIVE     OVER
lo0          loopback     ok        yes        --
qfe3         ip           ok        no         --
```

주 – 출력 결과에 `offline` 상태가 표시되는 경우에도 해당 링크에 IP 인터페이스가 있으므로 데이터 링크가 사용됩니다.

b. IP 인터페이스를 제거하려면 다음 명령을 입력합니다.

```
# ipadm delete-ip interface
```

구문 설명은 다음과 같습니다.

interface 링크에 생성된 IP 인터페이스를 지정합니다.

4 링크 통합을 만듭니다.

```
# dladm create-aggr [-f] -l link1 -l link2 [...] aggr
```

`-f` 통합을 강제로 만듭니다. 링크 상태 알림을 지원하지 않는 장치를 통합하려는 경우 이 옵션을 사용합니다.

linkn 통합하려는 데이터 링크를 지정합니다.

aggr 통합에 지정할 이름을 지정합니다.

5 통합에 IP 인터페이스를 만듭니다.

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

6 유효한 IP 주소로 IP 인터페이스를 구성합니다.

```
# ipadm create-addr interface -T static -a IP-address addrobj
```

여기서 *interface*는 통합 이름을 사용해야 하고 *addrobj*는 이름 지정 규약 *interface*/*user-defined-string*을 사용합니다.

7 방금 만든 통합의 상태를 확인합니다.

통합의 상태는 `UP`이어야 합니다.

```
# dladm show-aggr
```

예 12–1 링크 통합 만들기

이 예에서는 subvideo0 및 subvideo1이라는 데이터 링크 두 개가 있는 링크 통합을 만드는 데 사용되는 명령을 보여줍니다. 이 구성은 시스템 재부트 후에도 유지됩니다.

```
# dladm show-link
LINK          CLASS     MTU     STATE   BRIDGE     OVER
subvideo0     phys      1500    up      --         ----
subvideo1     phys      1500    up46    --         ----
```

```
# ipadm delete-ip subvideo0
# ipadm delete-ip subvideo1
# dladm create-aggr -l subvideo0 -l subvideo1 video0
# ipadm create-ip video0
# ipadm create-addr -T static -a 10.8.57.50/24 video/v4
# dladm show-aggr
LINK      POLICY  ADDRPOLICY      LACPACTIVITY   LACPTIMER  FLAGS
video0    L4      auto            off            short      -----
```

링크 정보를 표시하면 링크 통합이 목록에 포함됩니다.

```
# dladm show-link
LINK          CLASS     MTU     STATE   BRIDGE     OVER
subvideo0     phys      1500    up      --         ----
subvideo1     phys      1500    up      --         ----
video0        aggr      1500    up      --         subvideo0, subvideo1
```

▼ 통합을 수정하는 방법

이 절차에서는 통합 정의를 다음과 같이 변경하는 방법을 보여줍니다.

- 통합 정책 수정
- 통합 모드 변경

1 관리자로 전환합니다.

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

2 통합의 정책을 수정합니다.

 # dladm modify-aggr -P *policy-key aggr*

policy-key 222 페이지 "정책 및 로드 균형 조정"에 설명된 대로 L2, L3 및 L4 정책 중 하나 이상을 나타냅니다.

aggr 정책을 수정하려는 통합을 지정합니다.

3 통합의 LACP 모드를 수정합니다.

 # dladm modify-aggr -L *LACP-mode* **-T** *timer-value aggr*

-L *LACP-mode* 통합을 실행할 LACP 모드를 나타냅니다. 값은 `active`, `passive` 및 `off`입니다. 스위치가 passive 모드로 LACP를 실행하는 경우 통합에 대해 active 모드를 구성해야 합니다.

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

예 12–2 링크 통합 수정

이 예에서는 video0 통합의 정책을 L2로 수정한 다음 Active LACP 모드를 설정하는 방법을 보여줍니다.

```
# dladm modify-aggr -P L2 video0
# dladm modify-aggr -L active -T short video0
# dladm show-aggr
LINK      POLICY  ADDRPOLICY      LACPACTIVITY   LACPTIMER  FLAGS
video0    L2      auto            active         short      -----
```

▼ 통합에 링크를 추가하는 방법

1. **관리자로 전환합니다.**

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

2. **추가하려는 링크에 연결된 IP 인터페이스가 없는지 확인합니다.**

   # ipadm delete-ip *interface*

3. **통합에 링크를 추가합니다.**

   # dladm add-aggr -l *link* **[-l** *link***] [...]** *aggr*

   여기서 *link*는 통합에 추가할 데이터 링크를 나타냅니다.

4. **데이터 링크를 더 추가한 후 다른 작업을 수행하여 전체 링크 통합 구성을 수정합니다.**

   예를 들어, 그림 12–3에 설명된 구성의 경우 케이블 연결을 추가하거나 수정한 다음 추가 데이터 링크를 수용하도록 스위치를 재구성해야 할 수 있습니다. 스위치에서 재구성 작업을 수행하려면 스위치 설명서를 참조하십시오.

예 12–3 통합에 링크 추가

이 예에서는 video0 통합에 링크를 추가하는 방법을 보여줍니다.

```
# dladm show-link
LINK          CLASS     MTU     STATE     BRODGE   OVER
subvideo0     phys      1500    up        --       ----
subvideo1     phys      1500    up        --       ----
video0        aggr      1500    up        --       subvideo0, subvideo1
net3          phys      1500    unknown   --       ----

# ipadm delete-ip video0
# dladm add-aggr -l net3 video0
# dladm show-link
LINK          CLASS     MTU     STATE     BRIDGE   OVER
subvideo0     phys      1500    up        --       ----
```

```
subvideo1     phys      1500    up        --       ----
video0        aggr      1500    up        --       subvideo0, subvideo1, net3
net3          phys      1500    up        --       ----
```

▼ 통합에서 링크를 제거하는 방법

1 **관리자로 전환합니다.**

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

2 **통합에서 링크를 제거합니다.**

```
# dladm remove-aggr -l link  aggr-link
```

예 12–4 통합에서 링크 제거

이 예에서는 `video0` 통합에서 링크를 제거하는 방법을 보여줍니다.

```
dladm show-link
LINK          CLASS     MTU     STATE     OVER
subvideo0     phys      1500    up        --       ----
subvideo1     phys      1500    up        --       ----
video0        aggr      1500    up        --       subvideo0, subvideo1, net3
net3          phys      1500    up        --       ----

# dladm remove-aggr -l net3 video0
# dladm show-link
LINK          CLASS     MTU     STATE     BRIDGE   OVER
subvideo0     phys      1500    up        --       ----
subvideo1     phys      1500    up        --       ----
video0        aggr      1500    up        --       subvideo0, subvideo1
net3          phys      1500    unknown   --       ----
```

▼ 통합을 삭제하는 방법

1 **관리자로 전환합니다.**

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

2 **통합에 구성된 IP 인터페이스를 삭제합니다.**

```
# ipadm delete-ip IP-aggr
```

여기서 *IP-aggr*은 링크 통합의 IP 인터페이스입니다.

3 **링크 통합을 삭제합니다.**

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

예 12–5 통합 삭제

이 예에서는 `video0` 통합을 삭제합니다. 삭제는 지속적입니다.

```
# ipadm delete-ip video0
# dladm delete-aggr video0
```

◆ ◆ ◆ 13 장

13 VLAN 관리

이 장에서는 VLAN(가상 LAN)을 구성하고 유지 관리하는 절차에 대해 설명합니다. 이 절차에는 유연한 링크 이름 지원과 같은 기능을 사용하는 단계가 포함되어 있습니다.

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 할당하면 전체 로컬 네트워크에 대한 네트워크 관리와 보안을 향상시킬 수 있습니다. 동일한 시스템의 인터페이스를 서로 다른 VLAN에 할당할 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.

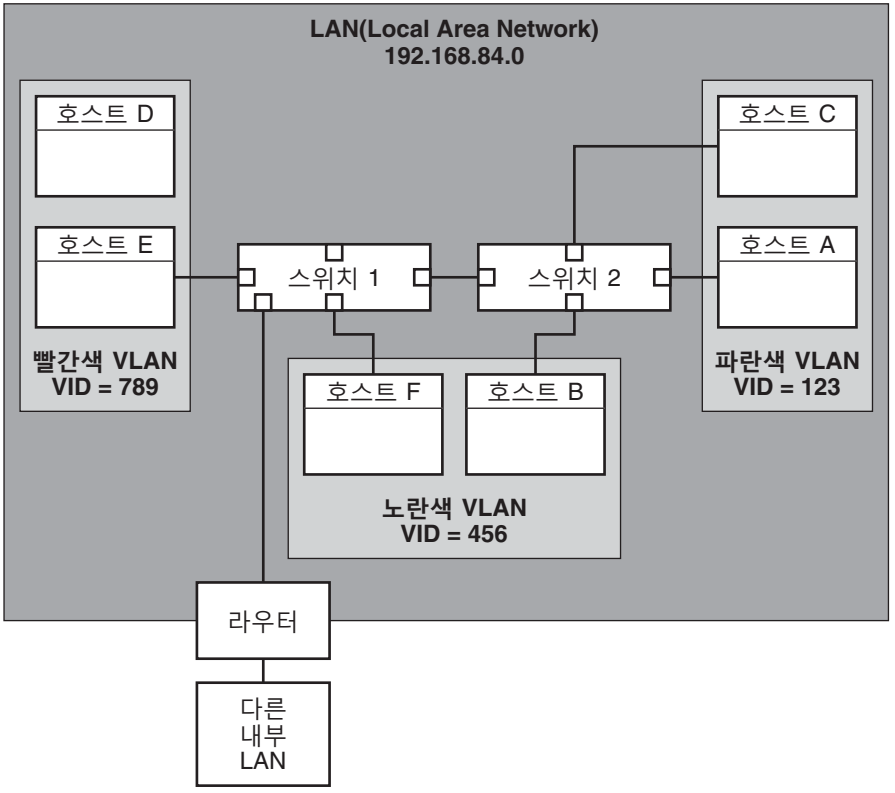
VLAN 토폴로지 개요

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 192.168.84.0인 LAN을 보여줍니다. 이 LAN은 빨간색, 노란색 및 파란색인 세 개의 VLAN으로 분할됩니다.

그림 13–1 VLAN 세 개가 있는 LAN(Local Area Network)

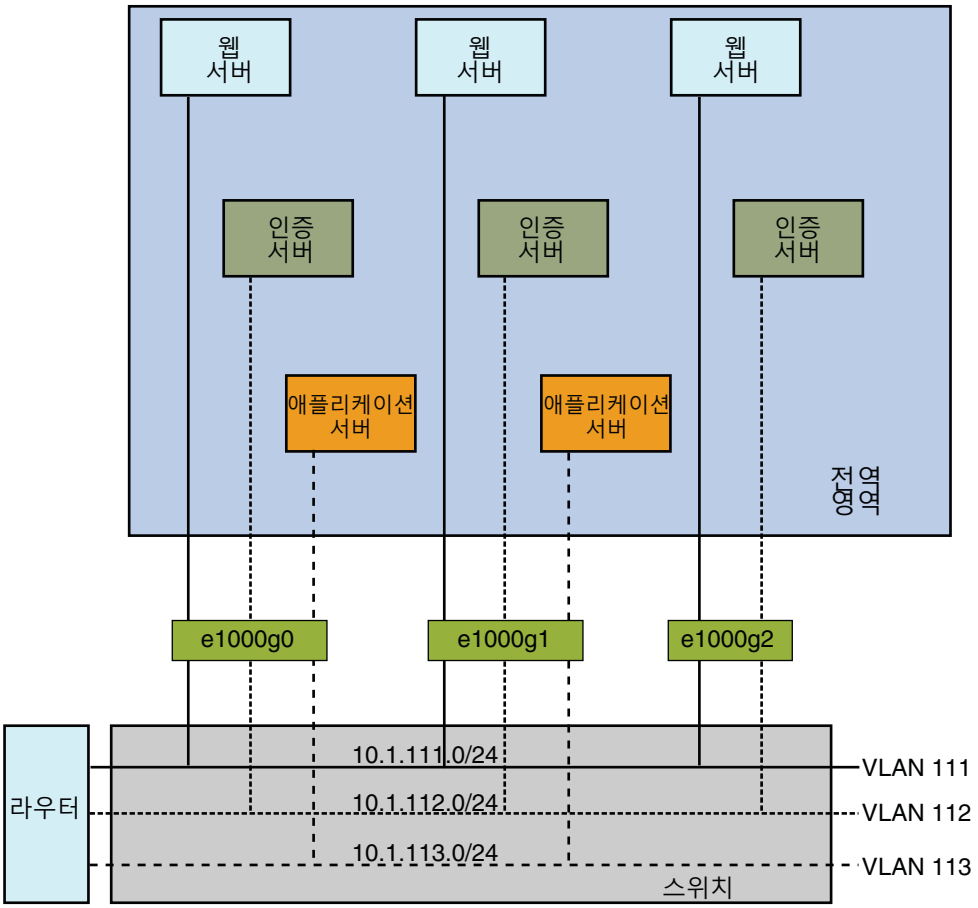


LAN 192.168.84.0의 연결은 스위치 1과 2에 의해 처리됩니다. 빨간색 VLAN에는 Accounting 작업 그룹의 시스템이 있습니다. Human Resources 작업 그룹의 시스템은 노란색 VLAN에 있습니다. Information Technologies 작업 그룹의 시스템은 파란색 VLAN에 할당됩니다.

VLAN을 사용하여 네트워크 통합

영역의 VLAN을 사용하면 스위치와 같은 단일 네트워크 장치 내에 여러 가상 네트워크를 구성할 수 있습니다. 물리적 NIC 세 개가 있는 시스템에 대한 다음 그림을 고려해 보십시오.

그림 13–2 여러 VLAN이 있는 시스템



VLAN을 사용하지 않을 경우 특정 기능을 수행하는 여러 시스템을 구성하고 이러한 시스템을 별도의 네트워크에 연결할 것입니다. 예를 들어, 웹 서버는 한 LAN에 연결되고, 인증 서버는 다른 LAN에 연결되고, 애플리케이션 서버는 세번째 네트워크에 연결됩니다. VLAN과 영역을 사용하면 모두 8개 시스템을 축소하여 단일 시스템의

영역으로 구성할 수 있습니다. 그런 다음 VLAN 태그 또는 VLAN ID(VID)를 사용하여 동일한 기능을 수행하는 각 영역 세트에 VLAN을 할당합니다. 그림에 제공된 정보를 다음과 같이 표로 작성할 수 있습니다.

기능	영역 이름	VLAN 이름	VID	IP 주소	NIC
웹 서버	webzone1	web1	111	10.1.111.0	e1000g0
인증 서버	authzone1	auth1	112	10.1.112.0	e1000g0
애플리케이션 서버	appzone1	app1	113	10.1.113.0	e1000g0
웹 서버	webzone2	web2	111	10.1.111.0	e1000g1
인증 서버	authzone2	auth2	112	10.1.112.0	e1000g1
애플리케이션 서버	appzone2	app2	113	10.1.113.0	e1000g1
웹 서버	webzone3	web3	111	10.1.111.0	e1000g2
인증 서버	authzone3	auth3	112	10.1.112.0	e1000g2

그림에 표시된 구성을 만들려면 예 13–1을 참조하십시오.

VLAN의 의미 있는 이름

Oracle Solaris에서는 VLAN 인터페이스에 의미 있는 이름을 지정할 수 있습니다. VLAN 이름은 링크 이름 및 VID(VLAN ID 번호)로 구성됩니다(예: `sales0`). VLAN을 만들 때 사용자 정의 이름을 지정해야 합니다. 사용자 정의 이름에 대한 자세한 내용은 24 페이지 “네트워크 장치 및 데이터 링크 이름”을 참조하십시오. 유효한 사용자 정의 이름에 대한 자세한 내용은 28 페이지 “유효한 링크 이름 규칙”을 참조하십시오.

VLAN 관리(작업 맵)

다음 표에서는 VLAN을 관리하는 여러 작업에 대한 링크를 제공합니다.

작업	설명	수행 방법
VLAN(가상 LAN)을 계획합니다.	VLAN을 만들기 전에 필요한 계획 작업을 수행합니다.	235 페이지 “VLAN 구성을 계획하는 방법”
VLAN을 구성합니다.	네트워크에 VLAN을 만듭니다.	236 페이지 “VLAN을 구성하는 방법”

작업	설명	수행 방법
통합에 VLAN을 구성합니다.	VLAN과 링크 통합을 모두 사용하는 결합된 기술을 배포합니다.	239 페이지 “링크 통합에 VLAN을 구성하는 방법”
VLAN 정보를 표시합니다.	VLAN과 해당 구성 요소에 대한 정보를 가져옵니다.	240 페이지 “VLAN 정보를 표시하는 방법”
VLAN을 제거합니다.	데이터 링크에 구성된 여러 VLAN에서 제거할 VLAN을 선택합니다.	241 페이지 “VLAN을 제거하는 방법”

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 13–1을 참조하십시오.

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

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 구성원이 될 인터페이스를 결정합니다.**

a. **시스템에 구성된 인터페이스를 확인합니다.**

```
# dladm show-link
```

b. **시스템의 각 데이터 링크와 연결할 VID를 식별합니다.**

c. **`dladm create-vlan` 명령을 사용하여 VLAN을 만듭니다.**

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

다음 절차에서는 VLAN을 만들고 구성하는 방법을 보여줍니다. Oracle Solaris에서는 모든 이더넷 장치가 VLAN을 지원할 수 있습니다. 하지만 특정 장치에는 몇 가지 제한 사항이 있습니다. 이러한 예외는 240 페이지 "레거시 장치의 VLAN"을 참조하십시오.

▼ VLAN을 구성하는 방법

시작하기 전에 VLAN을 만들기 전에 시스템에 데이터 링크가 이미 구성되어 있어야 합니다. 164 페이지 "IP 인터페이스를 구성하는 방법"을 참조하십시오.

1 관리자로 전환합니다.

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

2 시스템에서 사용 중인 링크 유형을 확인합니다.

```
# dladm show-link
```

3 데이터 링크에 VLAN 링크를 만듭니다.

```
# dladm create-vlan -l link -v VID vlan-link
```

link VLAN 인터페이스를 만들 링크를 지정합니다.

VID VLAN ID 번호를 나타냅니다.

vlan-link 관리상 선택한 이름일 수도 있는 VLAN의 이름을 지정합니다.

4 VLAN 구성을 확인합니다.

```
# dladm show-vlan
```

5 VLAN에 IP 인터페이스를 만듭니다.

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

여기서 *interface*는 VLAN 이름을 사용합니다.

6 IP 주소로 IP 인터페이스를 구성합니다.

```
# ipadm create-addr -T static -a IP-address addrobj
```

여기서 *addrobj*는 이름 지정 규약 *interface*/*user-defined-string*을 사용합니다.

예 13–1 VLAN 구성

이 예에서는 그림 13–2에 설명된 VLAN 구성을 만듭니다. 이 예는 시스템에 여러 영역을 이미 구성했다고 가정합니다. 영역 구성에 대한 자세한 내용은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 제II부, "Oracle Solaris Zones"을 참조하십시오.

```
global# dladm show-link
LINK        CLASS     MTU     STATE     BRIDGE     OVER
e1000g0     phys      1500    up        --         --
e1000g1     phys      1500    up        --         --
e1000g2     phys      1500    up        --         --

global# dladm create-vlan -l e1000g0 -v 111 web1
global# dladm create-vlan -l e1000g0 -v 112 auth1
global# dladm create-vlan -l e1000g0 -v 113 app1
global# dladm create-vlan -l e1000g1 -v 111 web2
global# dladm create-vlan -l e1000g1 -v 112 auth2
global# dladm create-vlan -l e1000g1 -v 113 app2
global# dladm create-vlan -l e1000g2 -v 111 web3
global# dladm create-vlan -l e1000g2 -v 112 auth3

global# dladm show-vlan
LINK      VID     OVER        FLAGS
web1      111     e1000g0     ----
auth1     112     e1000g0     ----
app1      113     e1000g0     ----
web2      111     e1000g1     ----
auth2     112     e1000g1     ----
app2      113     e1000g1     ----
web3      111     e1000g2     ----
auth3     113     e1000g2     ----
```

링크 정보가 표시되면 VLAN이 목록에 포함됩니다.

```
global# dladm show-link
LINK        CLASS    MTU      STATE     BRIDGE     OVER
e1000g0     phys     1500     up        --         --
e1000g1     phys     1500     up        --         --
e1000g2     phys     1500     up        --         --
web1        vlan     1500     up        --         e1000g0
auth1       vlan     1500     up        --         e1000g0
app1        vlan     1500     up        --         e1000g0
web2        vlan     1500     up        --         e1000g1
auth2       vlan     1500     up        --         e1000g1
app2        vlan     1500     up        --         e1000g1
web3        vlan     1500     up        --         e1000g2
auth3       vlan     1500     up        --         e1000g2
```

해당 영역에 VLAN을 할당합니다. 예를 들어, 개별 영역에 대한 네트워크 정보를 확인하면 각 영역에 대해 다음과 유사한 데이터가 표시됩니다.

```
global# zonecfg -z webzone1 info net
net:
     address not specified
     physical: web1

global# zonecfg -z authzone1 info net
net:
     address not specified
     physical: auth1

global# zonecfg -z appzone2 info net
```

```
net:
     address not specified
     physical: app2
```

physical 등록 정보의 값은 지정된 영역에 대해 설정된 VLAN을 나타냅니다.

각 비전역 영역에 로그인하여 IP 주소로 VLAN을 구성합니다.

webzone1:

```
webzone1# ipadm create-ip web1
webzone1# ipadm create-addr -T static -a 10.1.111.0/24 web1/v4
```

webzone2:

```
webzone2# ipadm create-ip web2
webzone2# ipadm create-addr -T static -a 10.1.111.0/24 web2/v4
```

webzone3:

```
webzone3# ipadm create-ip web3
webzone3# ipadm create-addr -T static -a 10.1.111.0/24 web3/v4
```

authzone1:

```
authzone1# ipadm create-ip auth1
authzone1# ipadm create-addr -T static -a 10.1.112.0/24 auth1/v4
```

authzone2:

```
authzone2# ipadm create-ip auth2
autzone2# ipadm create-addr -T static -a 10.1.112.0/24 auth2/v4
```

authzone3:

```
authzone3# ipadm create-ip auth3
authzone3# ipadm create-addr -T static -a 10.1.112.0/24 auth3/v4
```

appzone1:

```
appzone1# ipadm create-ip app1
appzone1# ipadm create-addr -T static -a 10.1.113.0/24 app1/v4
```

appzone2:

```
appzone2# ipadm create-ip app2
appzone2# ipadm create-addr -T static -a 10.1.113.0/24 app2/v4
```

▼ 링크 통합에 VLAN을 구성하는 방법

인터페이스에 VLAN을 구성하는 것과 동일한 방식으로 링크 통합에 VLAN을 만들 수도 있습니다. 링크 통합은 12 장, "링크 통합 관리"에서 설명합니다. 이 절에서는 VLAN과 링크 통합의 구성을 결합합니다.

시작하기 전에 먼저 링크 통합을 만들고 유효한 IP 주소로 구성합니다. 링크 통합을 만들려면 224 페이지 "링크 통합을 만드는 방법"을 참조하십시오.

1 시스템에 구성된 통합을 나열합니다.

```
# dladm show-link
```

2 통합에 만들려는 각 VLAN에 대해 다음 명령을 실행합니다.

```
# dladm create-vlan -l link -v VID vlan-link
```

구문 설명은 다음과 같습니다.

link VLAN 인터페이스를 만들 링크를 지정합니다. 이 특정 경우에서 링크는 링크 통합을 나타냅니다.

VID VLAN ID 번호를 나타냅니다.

vlan-link 관리상 선택한 이름일 수도 있는 VLAN의 이름을 지정합니다.

3 VLAN에 IP 인터페이스를 만듭니다.

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

여기서 *interface*는 VLAN 이름을 사용합니다.

4 VLAN의 IP 인터페이스를 유효한 IP 주소로 구성합니다.

```
# ipadm create-addr -T static -a IP-address addrobj
```

여기서 *addrobj*는 이름 지정 규약 *vlan-int*/*user-defined-string*을 따라야 합니다.

예 13–2 링크 통합에 여러 VLAN 구성

이 예에서는 링크 통합에 VLAN 두 개가 구성됩니다. VLAN에는 각각 VID 193과 194가 할당됩니다.

```
# dladm show-link
LINK          CLASS     MTU     STATE     BRIDGE     OVER
subvideo0     phys      1500    up        --         ----
subvideo1     phys      1500    up        --         ----
video0        aggr      1500    up        --         subvideo0, subvideo1

# dladm create-vlan -l video0 -v 193 salesregion1
# dladm create-vlan -l video0 -v 194 salesregion2

# ipadm create-ip salesregion1
```

```
# ipadm create-ip salesregion2

# ipadm create-addr -T static -a 192.168.10.5/24 salesregion1/v4static
# ipadm create-addr -T static -a 192.168.10.25/24 salesregion2/v4static
```

레거시 장치의 VLAN

특정 레거시 장치는 최대 프레임 크기가 1514바이트인 패킷만 처리합니다. 프레임 크기가 최대 제한을 초과하는 패킷은 삭제됩니다. 이 경우 236 페이지 “VLAN을 구성하는 방법”에 나열된 것과 동일한 절차를 따릅니다. 하지만 VLAN을 만들 때는 `-f` 옵션을 사용하여 VLAN을 강제로 만듭니다.

일반적인 수행 단계는 다음과 같습니다.

1. `-f` 옵션을 사용하여 VLAN을 만듭니다.

   ```
   # dladm create-vlan -f -l link -v VID [vlan-link]
   ```

2. 최대 전송 단위(MTU)에 대해 더 작은 크기를 설정합니다(예: 1496바이트).

   ```
   # dladm set-linkprop -p default_mtu=1496 vlan-link
   ```

   MTU 값이 작으면 링크 계층에서 전송 전에 VLAN 헤더를 삽입할 수 있는 공간이 허용됩니다.

3. 동일한 단계를 수행하여 VLAN에 있는 각 노드의 MTU 크기에 대해 더 작은 값을 동일하게 설정합니다.

   링크 등록 정보 값 변경에 대한 자세한 내용은 141 페이지 “데이터 링크 구성(작업)”을 참조하십시오.

VLAN에서 기타 관리 작업 수행

이 절에서는 기타 VLAN 작업의 새 하위 명령 `dladm` 사용에 대해 설명합니다. 이러한 `dladm` 명령은 링크 이름에서도 작동합니다.

▼ VLAN 정보를 표시하는 방법

1 관리자로 전환합니다.

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

2 VLAN 정보를 표시합니다.

```
# dladm show-vlan [vlan-link]
```

VLAN 링크를 지정하지 않으면 이 명령은 구성된 모든 VLAN에 대한 정보를 표시합니다.

예 13–3 VLAN 정보 표시

다음 예는 그림 13–2와 같이 여러 VLAN이 있는 시스템을 기반으로 하며, 시스템에서 사용 가능한 VLAN을 보여줍니다.

```
# dladm show-vlan
LINK      VID     OVER        FLAGS
web1      111     e1000g0     ----
auth1     112     e1000g0     ----
app1      113     e1000g0     ----
web2      111     e1000g1     ----
auth2     112     e1000g1     ----
app2      113     e1000g1     ----
web3      111     e1000g2     ----
auth3     113     e1000g2     ----
```

구성된 VLAN은 `dladm show-link` 명령을 실행할 때도 표시됩니다. VLAN은 명령 출력 결과의 `CLASS` 열에서 적절하게 식별됩니다.

```
# dladm show-link
LINK        CLASS     MTU      STATE     BRIDGE     OVER
e1000g0     phys      1500     up        --         --
e1000g1     phys      1500     up        --         --
e1000g2     phys      1500     up        --         --
web1        vlan      1500     up        --         e1000g0
auth1       vlan      1500     up        --         e1000g0
app1        vlan      1500     up        --         e1000g0
web2        vlan      1500     up        --         e1000g1
auth2       vlan      1500     up        --         e1000g1
app2        vlan      1500     up        --         e1000g1
web3        vlan      1500     up        --         e1000g2
auth3       vlan      1500     up        --         e1000g2
```

▼ VLAN을 제거하는 방법

1 관리자로 전환합니다.

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

2 제거할 VLAN을 결정합니다.

```
# dladm show-vlan
```

3 VLAN의 IP 인터페이스를 연결 취소합니다.

```
# ipadm delete-ip vlan-interface
```

여기서 *vlan-interface*는 VLAN에 구성된 IP 인터페이스입니다.

주 – 현재 사용 중인 VLAN은 제거할 수 없습니다.

---

4 **다음 단계 중 하나를 수행하여 VLAN을 제거합니다.**

- 일시적으로 VLAN을 삭제하려면 다음과 같이 `-t` 옵션을 사용합니다.

  # dladm delete-vlan -t *vlan*

- 지속적으로 삭제하려면 다음을 수행합니다.

  a. VLAN을 제거합니다.

  # dladm delete-vlan *vlan*

예 13–4 VLAN 제거

```
# dladm show-vlan
LINK      VID     OVER        FLAGS
web1      111     e1000g0     ----
auth1     112     e1000g0     ----
app1      113     e1000g0     ----
web2      111     e1000g1     ----
auth2     112     e1000g1     ----
app2      113     e1000g1     ----
web3      111     e1000g2     ----
auth3     113     e1000g2     ----

# ipadm delete-ip web1
# dladm delete-vlan web1
```

사용자 정의 이름을 사용하는 동안 네트워크 구성 작업 결합

이 절에서는 사용자 정의 이름 사용 시 링크, 링크 통합 및 VLAN 구성에 대한 이전 장의 모든 절차를 결합하는 예를 제공합니다. 사용자 정의 이름을 사용하는 다른 네트워킹 시나리오에 대한 설명은 http://www.oracle.com/us/sun/index.htm의 문서를 참조하십시오.

예 13–5 링크, VLAN 및 링크 통합 구성

이 예에서는 NIC 4개를 사용하는 한 시스템을 8개의 개별 서브넷의 라우터로 구성해야 합니다. 이를 위해서 각 서브넷에 대해 하나씩 8개의 링크가 구성됩니다. 먼저 NIC 4개에서 모두 링크 통합이 생성됩니다. 이러한 태그 미지정된 링크는 기본 경로가 가리키는 네트워크에 대해 태그 미지정된 기본 서브넷이 됩니다.

그런 후에 다른 서브넷의 링크 통합에 VLAN 인터페이스가 구성됩니다. 색상으로 구분된 체계에 따라 서브넷의 이름이 지정됩니다. 마찬가지로 VLAN 이름은 해당 서브넷에 일치하도록 지정됩니다. 최종 구성은 8개 서브넷에 대한 8개 링크로 구성되며, 한 개는 태그 미지정된 링크이고 나머지 7개는 태그 지정된 VLAN 링크입니다.

예 13–5 링크, VLAN 및 링크 통합 구성 *(계속)*

재부트 후에도 구성이 유지되게 하려면 이전 Oracle Solaris 릴리스와 동일한 절차가 적용됩니다. 예를 들어, `/etc/inet/ndpd.conf`와 같이 구성 파일에 IP 주소를 추가해야 합니다. 또는 인터페이스에 대한 필터 규칙을 규칙 파일에 포함해야 합니다. 이러한 최종 단계는 예에 포함되어 있지 않습니다. 이 단계는 **Oracle Solaris 관리: IP 서비스**의 해당 장, 특히 **TCP/IP 관리** 및 **DHCP**를 참조하십시오.

```
# dladm show-link
LINK        CLASS      MTU  STATE    BRIDGE     OVER
nge0        phys      1500  up       --         --
nge1        phys      1500  up       --         --
e1000g0     phys      1500  up       --         --
e1000g1     phys      1500  up       --         --

# dladm show-phys
LINK        MEDIA               STATE      SPEED  DUPLEX   DEVICE
nge0        Ethernet            up         1000Mb  full     nge0
nge1        Ethernet            up         1000Mb  full     nge1
e1000g0     Ethernet            up         1000Mb  full     e1000g0
e1000g1     Ethernet            up         1000Mb  full     e1000g1

# ipadm delete-ip nge0
# ipadm delete-ip nge1
# ipadm delete-ip e1000g0
# ipadm delete-ip e1000g1

# dladm rename-link nge0 net0
# dladm rename-link nge1 net1
# dladm rename-link e1000g0 net2
# dladm rename-link e1000g1 net3

# dladm show-link
LINK        CLASS      MTU  STATE    BRIDGE     OVER
net0        phys      1500  up       --         --
net1        phys      1500  up       --         --
net2        phys      1500  up       --         --
net3        phys      1500  up       --         --

# dladm show-phys
LINK        MEDIA               STATE      SPEED  DUPLEX   DEVICE
net0        Ethernet            up         1000Mb  full     nge0
net1        Ethernet            up         1000Mb  full     nge1
net2        Ethernet            up         1000Mb  full     e1000g0
net3        Ethernet            up         1000Mb  full     e1000g1

# dladm create-aggr -P L2,L3 -l net0 -l net1 -l net2 -l net3 default0

# dladm show-link
LINK        CLASS      MTU  STATE    BRIDGE     OVER
net0        phys      1500  up       --         --
net1        phys      1500  up       --         --
net2        phys      1500  up       --         --
net3        phys      1500  up       --         --
default0    aggr      1500  up       --         net0 net1 net2 net3
```

예 13–5 링크, VLAN 및 링크 통합 구성 *(계속)*

```
# dladm create-vlan -v 2 -l default0 orange0
# dladm create-vlan -v 3 -l default0 green0
# dladm create-vlan -v 4 -l default0 blue0
# dladm create-vlan -v 5 -l default0 white0
# dladm create-vlan -v 6 -l default0 yellow0
# dladm create-vlan -v 7 -l default0 red0
# dladm create-vlan -v 8 -l default0 cyan0

# dladm show-link
LINK        CLASS      MTU  STATE    BRIDGE     OVER
net0        phys      1500  up       --         --
net1        phys      1500  up       --         --
net2        phys      1500  up       --         --
net3        phys      1500  up       --         --
default0    aggr      1500  up       --         net0 net1 net2 net3
orange0     vlan      1500  up       --         default0
green0      vlan      1500  up       --         default0
blue0       vlan      1500  up       --         default0
white0      vlan      1500  up       --         default0
yellow0     vlan      1500  up       --         default0
red0        vlan      1500  up       --         default0
cyan0       vlan      1500  up       --         default0

# dladm show-vlan
LINK          VID   OVER        FLAGS
orange0         2   default0    -----
green0          3   default0    -----
blue0           4   default0    -----
white0          5   default0    -----
yellow0         6   default0    -----
red0            7   default0    -----
cyan0           8   default0    -----

# ipadm create-ip orange0
# ipadm create-ip green0
# ipadm create-ip blue0
# ipadm create-ip white0
# ipadm create-ip yellow0
# ipadm create-ip red0
# ipadm create-ip cyan0

# ipadm create-addr -T static -a IP-address orange0/v4
# ipadm create-addr -T static -a IP-address green0/v4
# ipadm create-addr -T static -a IP-address blue0/v4
# ipadm create-addr -T static -a IP-address white0/v4
# ipadm create-addr -T static -a IP-address yellow0/v4
# ipadm create-addr -T static -a IP-address red0/v4
# ipadm create-addr -T static -a IP-address cyan0/v4
```

◆ ◆ ◆ 14장

14 IPMP 소개

IPMP(IP Network Multipathing)는 특정 LAN에 연결된 여러 인터페이스가 있는 시스템에 대해 물리적 인터페이스 실패 감지, 투명한 네트워크 액세스 페일오버 및 패킷 부하 분산 기능을 제공합니다.

이 장은 다음 내용으로 구성되어 있습니다.

주 – 이 장과 15 장, "IPMP 관리"의 IPMP 설명 전체, 특히 모든 내용에 나오는 **인터페이스** 용어는 *IP* **인터페이스**를 의미합니다. NIC(네트워크 인터페이스 카드)와 같이 용어 정의에서 용어의 다른 사용을 명시적으로 나타내지 않는 경우 이 용어는 항상 IP 계층에 구성된 인터페이스를 가리킵니다.

IPMP의 새로운 기능

이전 구현에 비해 달라진 현재 IPMP 구현의 기능은 다음과 같습니다.

- IPMP 그룹이 IPMP IP 인터페이스로 표현됩니다. 이 인터페이스는 네트워킹 스택의 IP 계층에 있는 다른 인터페이스와 동일하게 처리됩니다. 모든 IP 관리 작업, 경로 지정 테이블, ARP(Address Resolution Protocol) 테이블, 방화벽 규칙 및 기타 IP 관련 절차는 IPMP 인터페이스를 참조하여 IPMP 그룹과 함께 작동합니다.

- 시스템이 기본 활성 인터페이스에 데이터 주소를 배포합니다. 이전 IPMP 구현에서는 관리자가 처음에 IPMP 그룹을 만들 때 해당 인터페이스에 대한 데이터 주소의 바인딩을 결정합니다. 현재 구현에서는 IPMP 그룹을 만들 때 데이터 주소가 주소 풀로 IPMP 인터페이스에 속합니다. 그런 다음 커널이 데이터 주소를 그룹의 기본 활성 인터페이스에 임의로 자동 바인딩합니다.
- `ipmpstat` 도구는 IPMP 그룹에 대한 정보를 가져오는 주요 도구로 도입되었습니다. 이 명령은 그룹의 기본 IP 인터페이스, 테스트 및 데이터 주소, 사용되는 실패 감지 유형, 실패한 인터페이스 등 IPMP 구성의 모든 측면에 대한 정보를 제공합니다. `ipmpstat` 함수, 사용할 수 있는 옵션 및 각 옵션이 생성하는 출력 결과는 모두 289 페이지 "IPMP 정보 모니터링"에서 설명합니다.
- 네트워크 설정 중에 IPMP 그룹을 식별하기 쉽도록 IPMP 인터페이스에 사용자 정의 이름을 지정할 수 있습니다. 사용자 정의 이름으로 IPMP 그룹을 구성하는 절차는 273 페이지 "IPMP 그룹 구성"에서 IPMP 그룹 만들기를 설명하는 절차를 참조하십시오.

주 – IPMP를 사용하려면 시스템에서 `DefaultFixed` 프로파일이 사용으로 설정되었는지 확인합니다. 절차는 138 페이지 "프로파일 및 구성 도구"를 참조하십시오. 프로파일 관리 네트워크 구성에 대한 자세한 내용은 4 장, "NWAM 프로파일 구성(작업)"을 참조하십시오.

IPMP 배포

이 절에서는 IPMP 그룹 사용에 대한 다양한 내용에 대해 설명합니다.

IPMP 사용 이유

인터페이스를 사용할 수 없게 하는 여러 가지 요인이 있습니다. 일반적으로 IP 인터페이스가 실패할 수 있습니다. 또는 하드웨어 유지 관리를 위해 인터페이스를 오프라인 상태로 전환할 수 있습니다. 이 경우 IPMP 그룹이 없으면 사용할 수 없는 인터페이스와 연결된 IP 주소를 사용하여 시스템에 더 이상 연결할 수 없습니다. 또한 해당 IP 주소를 사용하는 기존 연결이 손상됩니다.

IPMP를 사용하면 하나 이상의 IP 인터페이스를 *IPMP* **그룹**으로 구성할 수 있습니다. 이 그룹은 네트워크 트래픽을 보내거나 받는 데이터 주소가 있는 IP 인터페이스로 작동합니다. 그룹의 기본 인터페이스가 실패할 경우 데이터 주소가 그룹의 나머지 기본 활성 인터페이스에 재배포됩니다. 따라서 인터페이스 실패 후에도 그룹이 네트워크 연결을 유지합니다. IPMP를 사용하면 그룹에 대해 사용할 수 있는 인터페이스가 하나만 있어도 항상 네트워크 연결을 사용할 수 있습니다.

또한 IPMP는 IPMP 그룹의 인터페이스 세트에 아웃바운드 네트워크 트래픽을 자동으로 분산하여 전체 네트워크 성능을 향상시킵니다. 이 프로세스를 아웃바운드 **부하**

분산이라고 합니다. 또한 시스템은 응용 프로그램에서 해당 IP 소스 주소가 지정되지 않은 패킷에 대해 소스 주소를 선택하여 인바운드 부하 분산을 간접적으로 제어합니다. 하지만 응용 프로그램이 IP 소스 주소를 명시적으로 선택한 경우 시스템에서 해당 소스 주소를 변경하지 않습니다.

IPMP 사용 시기

IPMP 그룹의 구성은 시스템 구성에 의해 결정됩니다. 다음 규칙을 확인합니다.

- 동일한 LAN의 여러 IP 인터페이스를 IPMP 그룹으로 구성해야 합니다. 광범위한 의미의 LAN은 해당 노드가 **동일한 링크 계층 브로드캐스트 도메인**에 속하는 유무선 로컬 네트워크와 VLAN을 포함하는 다양한 로컬 네트워크 구성을 나타냅니다.

  주 – 동일한 링크 계층(L2) 브로드캐스트 도메인의 여러 IPMP 그룹은 지원되지 않습니다. L2 브로드캐스트 도메인은 일반적으로 특정 서브넷에 매핑됩니다. 따라서 서브넷당 IPMP 그룹 한 개만 구성해야 합니다.

- IPMP 그룹의 기본 IP 인터페이스가 여러 LAN에 걸쳐 있지 않아야 합니다.

예를 들어, 세 개의 인터페이스가 있는 시스템이 두 개의 개별 LAN에 연결되어 있다고 가정합니다. 두 개의 IP 인터페이스가 하나의 LAN에 연결되고 남은 단일 IP 인터페이스가 다른 LAN에 연결됩니다. 이 경우 첫번째 규칙에 따라 첫번째 LAN에 연결한 두 개의 IP 인터페이스를 IPMP 그룹으로 구성해야 합니다. 두번째 규칙에 따라 두번째 LAN에 연결한 단일 IP 인터페이스는 해당 IPMP 그룹의 구성원이 될 수 없습니다. 단일 IP 인터페이스에 대한 IPMP 구성은 필요 없습니다. 하지만 인터페이스의 가용성을 모니터하기 위해 단일 인터페이스를 IPMP 그룹으로 구성할 수 있습니다. 단일 인터페이스 IPMP 구성은 255 페이지 “IPMP 인터페이스 구성 유형”에서 자세히 설명합니다.

첫번째 LAN에 대한 링크가 IP 인터페이스 세 개로 구성되고 다른 링크는 인터페이스 두 개로 구성된 또 다른 사례를 고려해 보십시오. 이 설치에서는 IPMP 그룹 두 개를 구성해야 합니다. 첫번째 LAN에 연결하는 세 개의 인터페이스가 한 그룹이고, 두번째 LAN에 연결하는 두 개의 인터페이스가 또 다른 그룹입니다.

IPMP 및 링크 통합 비교

IPMP 및 링크 통합은 네트워크 성능을 향상시키고 네트워크 가용성을 유지 관리하는 서로 다른 기술입니다. 일반적으로 링크 통합은 네트워크 성능 향상을 위해 배포하고 IPMP는 고가용성을 위해 사용합니다.

다음 표에서는 링크 통합과 IPMP의 일반적인 비교 내용을 보여줍니다.

	IPMP	링크 통합
네트워크 기술 유형	계층 3(IP 계층)	계층 2(링크 계층)
구성 도구	`ipadm`	`dladm`
링크 기반 실패 감지	지원됨	지원됨
검사 기반 실패 감지	ICMP 기반으로, 중간에 있는 계층 2 스위치의 여러 레벨에서 테스트 주소와 동일한 IP 서브넷에 정의된 모든 시스템을 대상으로 합니다.	LACP(Link Aggregation Control Protocol) 기반으로, 직접 피어 호스트 또는 스위치를 대상으로 합니다.
대기 인터페이스 사용	지원됨	지원되지 않음
여러 스위치에 걸쳐 있음	지원됨	일반적으로 지원되지 않음. 일부 공급업체는 여러 스위치에 걸쳐 있는 링크 통합에 대해 상호 운용 불가능한 고유의 솔루션을 제공합니다.
하드웨어 지원	필요하지 않음	필요함. 예를 들어, Oracle Solaris를 실행하는 시스템에서 링크 통합을 사용하려면 스위치의 해당 포트도 통합해야 합니다.
링크 계층 요구 사항	브로드캐스트 가능	이더넷 관련
드라이버 프레임워크 요구 사항	없음	GLDv3 프레임워크를 사용해야 합니다.
부하 분산 지원	있음. 커널에 의해 제어됩니다. 인바운드 부하 분산은 소스 주소 선택의 간접적인 영향을 받습니다.	`dladm` 명령을 사용하여 관리자가 아웃바운드 트래픽의 부하 분산을 세부적으로 제어할 수 있습니다. 인바운드 부하 분산이 지원됩니다.

링크 통합에서 수신 트래픽은 통합을 구성하는 여러 링크에 분산됩니다. 따라서 통합에 링크를 추가하기 위해 더 많은 NIC를 설치하면 네트워킹 성능이 향상됩니다. IPMP 트래픽은 사용 가능한 활성 인터페이스에 바인딩된 경우 IPMP 인터페이스의 데이터 주소를 사용합니다. 예를 들어, 모든 데이터 트래픽이 두 개의 IP 주소 간에만 전달되지만 반드시 동일한 연결을 사용하지는 않는 경우 IP 주소 두 개만 사용할 수 있기 때문에 NIC를 더 추가해도 IPMP에서 성능이 향상되지 않습니다.

두 기술은 서로를 보완하며 함께 배포하여 네트워크 성능과 가용성을 더욱 향상시킬 수 있습니다. 예를 들어, 특정 공급업체가 고유의 솔루션을 제공한 경우를 제외하고 현재 링크 통합은 여러 스위치에 걸쳐 있을 수 없습니다. 따라서 스위치가 스위치와 호스트 간 링크 통합의 단일 실패 포인트가 됩니다. 스위치가 실패하면 마찬가지로 링크 통합도 손실되고 네트워크 성능이 저하됩니다. IPMP 그룹에는 이러한 스위치 제한이 적용되지 않습니다. 따라서 여러 스위치를 사용하는 LAN 시나리오에서는 해당 스위치에 연결하는 링크 통합을 호스트의 IPMP 그룹으로 결합할 수 있습니다. 이 구성을 사용하면

네트워크 성능과 고가용성이 모두 향상됩니다. 스위치가 실패할 경우 실패한 스위치에 대한 링크 통합의 데이터 주소가 그룹의 나머지 링크 통합에 재배포됩니다.

링크 통합에 대한 기타 내용은 12 장, “링크 통합 관리”를 참조하십시오.

IPMP 구성에서 유연한 링크 이름 사용

사용자 정의 링크 이름을 지원할 경우 링크 구성이 링크가 연결된 물리적 NIC에 더 이상 바인딩되지 않습니다. 사용자 정의 링크 이름을 사용하면 IP 인터페이스를 훨씬 유연하게 관리할 수 있습니다. 이러한 유연성은 IPMP 관리로도 확장됩니다. IPMP 그룹의 기본 인터페이스가 실패하고 교체해야 하는 경우 인터페이스 교체 절차가 훨씬 쉬워집니다. 실패한 NIC와 동일한 유형인 경우 교체 NIC의 이름을 바꾸어 실패한 NIC의 구성을 상속받을 수 있습니다. IPMP 그룹에 새 인터페이스를 추가하기 전에 새 구성을 만들 필요가 없습니다. 실패한 NIC의 링크 이름을 새 NIC에 지정하면 새 NIC가 실패한 인터페이스와 동일한 설정으로 구성됩니다. 그런 다음 다중 경로 지정 데몬이 활성 및 대기 인터페이스의 IPMP 구성에 따라 인터페이스를 배포합니다.

따라서 네트워킹 구성을 최적화하고 IPMP 관리를 용이하게 하려면 일반 이름을 지정하여 인터페이스에 유연한 링크 이름을 활용해야 합니다. 다음 절인 249 페이지 “IPMP 작동 방식”의 모든 예에서는 IPMP 그룹과 기본 인터페이스에 유연한 링크 이름을 사용합니다. 사용자 정의 링크 이름을 사용하는 네트워킹 환경에서 NIC 교체의 이면 프로세스에 대한 자세한 내용은 261 페이지 “IPMP 및 동적 재구성”을 참조하십시오. 네트워킹 스택 및 사용자 정의 링크 이름의 사용에 대한 개요는 20 페이지 “Oracle Solaris의 네트워크 스택”을 참조하십시오.

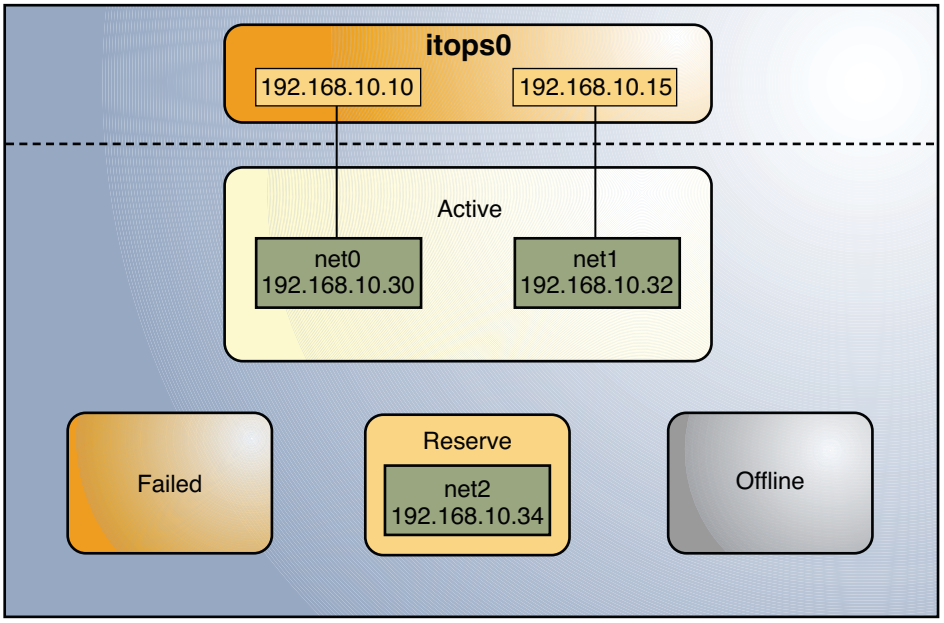
IPMP 작동 방식

IPMP는 그룹을 만들 때 활성 및 대기 인터페이스의 원래 개수를 보존하여 네트워크 가용성을 유지 관리합니다.

IPMP 실패 감지는 그룹에서 특정 기본 IP 인터페이스의 가용성을 확인하기 위해 링크 기반, 검사 기반 또는 둘 다일 수 있습니다. IPMP에서 기본 인터페이스가 실패했음을 확인하면 해당 인터페이스에 failed 플래그가 지정되며 더 이상 사용할 수 없습니다. 실패한 인터페이스와 연결된 데이터 IP 주소가 그룹에서 작동하는 다른 인터페이스에 재배포됩니다. 사용 가능한 경우 활성 인터페이스의 원래 개수를 유지 관리하기 위해 대기 인터페이스도 배포됩니다.

그림 14–1에 설명된 대로 활성-대기 구성을 사용하는 인터페이스가 세 개인 IPMP 그룹 `itops0`을 고려해 보십시오.

그림 14–1 IPMP 활성-대기 구성



itops0 그룹은 다음과 같이 구성됩니다.

- 192.168.10.10과 192.168.10.15라는 두 개의 데이터 주소가 그룹에 할당됩니다.
- 기본 인터페이스 두 개가 활성 인터페이스로 구성되고 net0과 net1이라는 유연한 링크 이름이 지정됩니다.
- 그룹에는 net2라는 유연한 링크 이름을 가진 대기 인터페이스 한 개가 있습니다.
- 검사 기반 실패 감지가 사용되므로 활성 및 대기 인터페이스가 다음과 같이 테스트 주소로 구성됩니다.
  - net0: 192.168.10.30
  - net1: 192.168.10.32
  - net2: 192.168.10.34

주 – 그림에서 Active, Offline, Reserve 및 Failed 영역은 물리적 위치가 아니라 기본 인터페이스의 상태만 나타냅니다. 이 IPMP 구현에서 인터페이스 또는 주소의 물리적 이동이나 IP 인터페이스의 전송은 발생하지 않습니다. 이 영역은 실패 또는 복구의 결과로 기본 인터페이스의 상태가 어떻게 변경되는지만 보여줍니다.

ipmpstat 명령에 여러 옵션을 사용하여 기존 IPMP 그룹에 대한 특정 유형의 정보를 표시할 수 있습니다. 추가 예를 보려면 289 페이지 “IPMP 정보 모니터링”을 참조하십시오.

다음의 ipmpstat 명령을 사용하면 그림 14–1의 IPMP 구성을 표시할 수 있습니다.

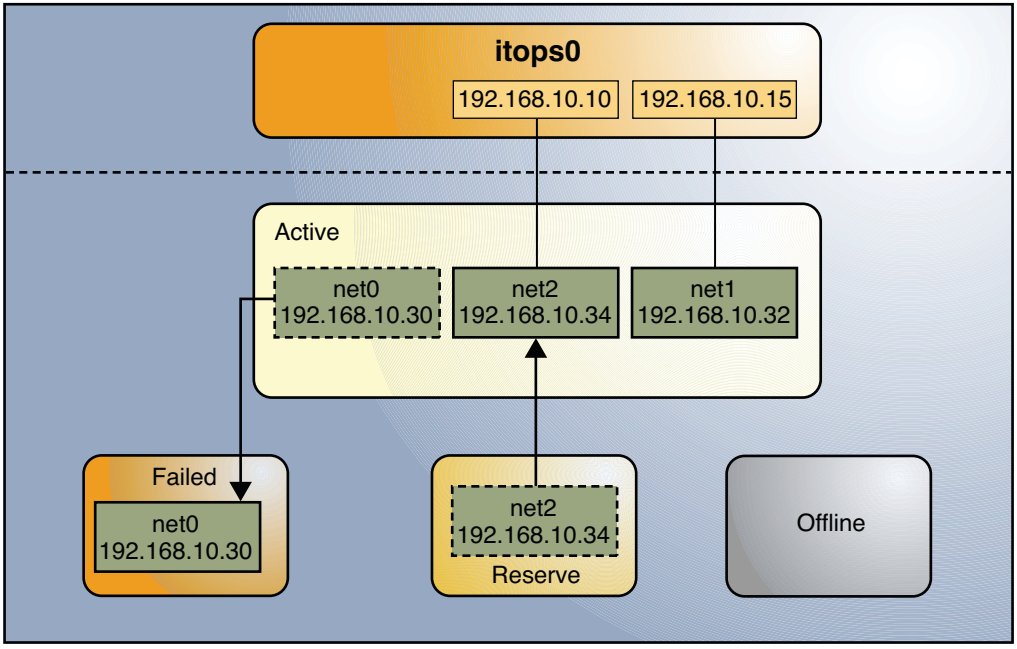
```
# ipmpstat -g
GROUP     GROUPNAME     STATE     FDT       INTERFACES
itops0    itops0        ok        10.00s    net1 net0 (net2)
```

그룹의 기본 인터페이스에 대한 정보를 표시하려면 다음을 입력합니다.

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS      LINK       PROBE     STATE
net0        yes        itops0    -------    up         ok        ok
net1        yes        itops0    --mb---    up         ok        ok
net2        no         itops0    is-----    up         ok        ok
```

IPMP는 기본 인터페이스 관리를 통해 활성 인터페이스의 원래 개수를 보존하여 네트워크 가용성을 유지 관리합니다. 따라서 net0이 실패할 경우 net2가 배포되어 그룹에서 활성 인터페이스 두 개가 유지되도록 합니다. net2 활성화는 그림 14–2에 나와 있습니다.

그림 14–2 IPMP의 인터페이스 실패



주 – 그림 14–2에 표시된 데이터 주소와 활성 인터페이스 간의 일대일 매핑은 그림을 단순화하기 위한 것일 뿐입니다. IP 커널 모듈은 데이터 주소와 인터페이스 간의 일대일 관계를 준수하지 않고 임의로 데이터 주소를 할당할 수 있습니다.

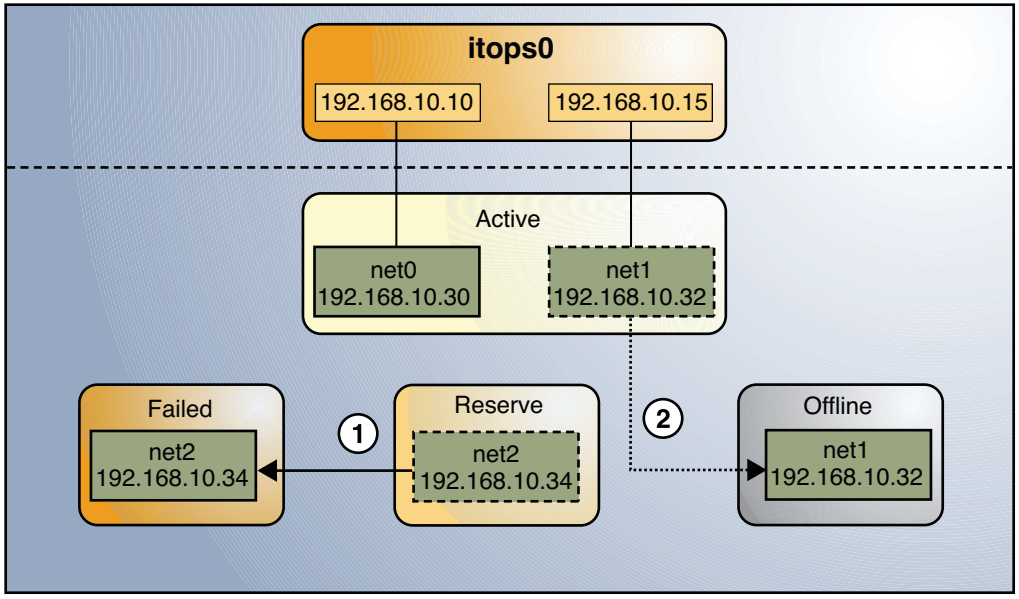
ipmpstat 유틸리티는 그림 14–2의 정보를 다음과 같이 표시합니다.

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS       LINK        PROBE     STATE
net0        no         itops0    -------     up          failed    failed
net1        yes        itops0    --mb---     up          ok        ok
net2        yes        itops0    -s-----     up          ok        ok
```

net0은 복구 후에 활성 인터페이스 상태로 돌아갑니다. net2는 원래의 대기 상태로 돌아갑니다.

그림 14–3에서는 대기 인터페이스 net가 실패하고(1), 나중에 관리자가 활성 인터페이스 net1을 오프라인 상태로 전환하는(2) 다른 실패 시나리오를 보여줍니다. 그 결과, IPMP 그룹에서 작동하는 인터페이스는 net0 한 개뿐입니다.

그림 14–3 IPMP의 대기 인터페이스 실패



ipmpstat 유틸리티는 그림 14–3의 정보를 다음과 같이 표시합니다.

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP      FLAGS       LINK        PROBE      STATE
net0        yes        itops0     -------     up          ok         ok
net1        no         itops0     --mb-d-     up          ok         offline
net2        no         itops0     is-----     up          failed     failed
```

이 특정 실패의 경우 인터페이스가 복구된 후의 복구가 다르게 동작합니다. 복구 후의 구성과 비교하여 복원은 IPMP 그룹의 원래 활성 인터페이스 수에 종속됩니다. 복구 프로세스는 그림 14–4에서 그래픽으로 나와 있습니다.

그림 14–4 IPMP 복구 프로세스

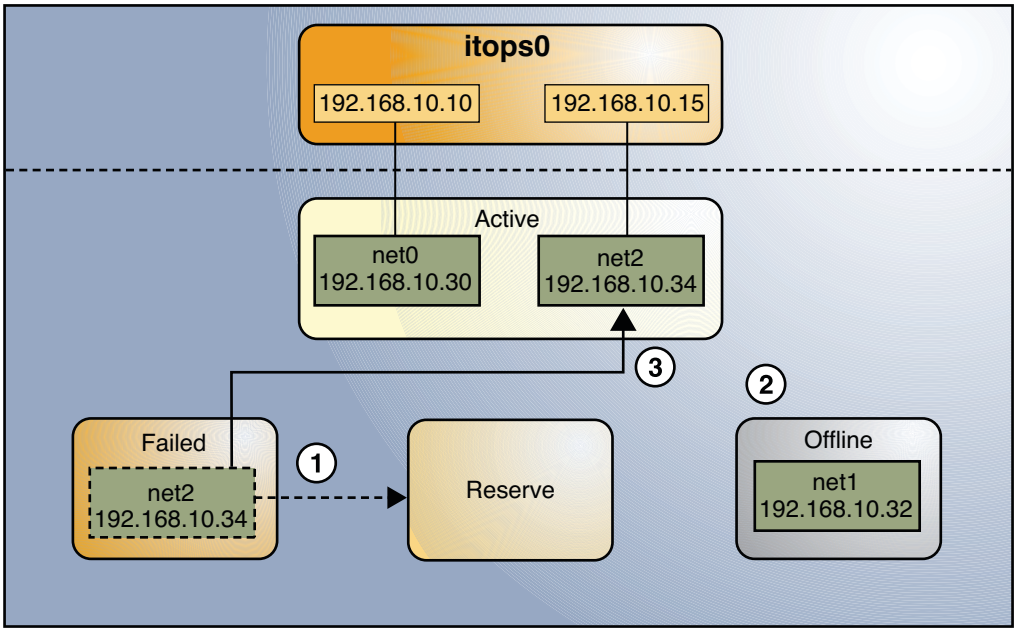


그림 14–4에서는 net2가 복구되면 정상적으로 원래 대기 인터페이스의 상태로 돌아갑니다(1). 하지만 net1이 계속 오프라인 상태로 유지되므로 IPMP 그룹이 원래 활성 인터페이스 두 개를 반영하지 않습니다(2). 따라서 IPMP가 net2를 활성 인터페이스로 대신 배포합니다(3).

ipmpstat 유틸리티는 복구 후 IPMP 시나리오를 다음과 같이 표시합니다.

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP      FLAGS       LINK        PROBE      STATE
net0        yes        itops0     -------     up          ok         ok
net1        no         itops0     --mb-d-     up          ok         offline
net2        yes        itops0     -s-----     up          ok         ok
```

실패한 활성 인터페이스가 복구 시 자동으로 활성 상태로 돌아가지 않는 FAILBACK=no 모드로 구성된 활성 인터페이스가 실패에 관련된 경우 유사한 복원 시퀀스가 발생합니다. 그림 14–2의 net0이 FAILBACK=no 모드로 구성되었다고 가정합니다. 이 모드에서 복구된 net0은 원래 활성 인터페이스였지만 대기 인터페이스의 예비 상태로 전환됩니다. net2 인터페이스가 활성 상태로 유지되어 IPMP 그룹의 원래 활성 인터페이스 두 개를 유지합니다. ipmpstat 유틸리티는 복구 정보를 다음과 같이 표시합니다.

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP      FLAGS      LINK       PROBE      STATE
net0        no         itops0     i------    up         ok         ok
net1        yes        itops0     --mb---    up         ok         ok
net2        yes        itops0     -s-----    up         ok         ok
```

이 유형의 구성에 대한 자세한 내용은 260 페이지 “FAILBACK=no 모드”를 참조하십시오.

Oracle Solaris의 IPMP 구성 요소

Oracle Solaris IPMP에서는 다음 소프트웨어가 사용됩니다.

다중 경로 지정 데몬 in.mpathd는 인터페이스 실패와 복구를 감지합니다. 기본 인터페이스에 대해 테스트 주소가 구성되어 있을 경우 이 데몬은 링크 기반 실패 감지와 검사 기반 실패 감지를 모두 수행합니다. 사용되는 실패 감지 방법의 유형에 따라 데몬은 인터페이스에서 해당 플래그를 설정하거나 지워 인터페이스가 실패했는지 또는 복구되었는지를 나타냅니다. 옵션으로, IPMP 그룹에 속하도록 구성되지 않은 인터페이스를 포함하여 모든 인터페이스의 가용성을 모니터하도록 데몬을 구성할 수도 있습니다. 실패 감지에 대한 설명은 257 페이지 “IPMP의 실패 및 복구 감지”를 참조하십시오.

또한 in.mpathd 데몬은 IPMP 그룹의 활성 인터페이스 지정을 제어합니다. 이 데몬은 IPMP 그룹을 만들 때 구성된 원래 활성 인터페이스 수를 동일하게 유지 관리합니다. 따라서 in.mpathd는 관리자가 구성한 정책에 따라 필요할 경우 기본 인터페이스를 활성화하거나 비활성화합니다. in.mpathd 데몬이 기본 인터페이스의 활성화를 관리하는 방식에 대한 자세한 내용은 249 페이지 “IPMP 작동 방식”을 참조하십시오. 데몬에 대한 자세한 내용은 in.mpathd(1M) 매뉴얼 페이지를 참조하십시오.

IP **커널 모듈**은 그룹에서 사용 가능한 IP 데이터 주소 세트를 그룹에서 사용 가능한 기본 IP 인터페이스 세트에 배포하여 아웃바운드 부하 분산을 관리합니다. 또한 이 모듈은 소스 주소를 선택하여 인바운드 부하 분산을 관리합니다. IP 모듈의 두 역할은 모두 네트워크 트래픽 성능을 향상시킵니다.

IPMP **구성 파일** /etc/default/mpathd는 데몬의 동작을 구성하는 데 사용됩니다. 예를 들어, 실패를 감지하기 위해 대상을 검사할 기간 또는 검사할 인터페이스를 설정하여 데몬이 검사 기반 실패 감지를 수행하는 방식을 지정할 수 있습니다. 인터페이스가 복구된 후의 실패한 인터페이스 상태를 지정할 수도 있습니다. 또한 이 파일의 매개변수를 설정하여 데몬이 IPMP 그룹에 속하도록 구성된 인터페이스뿐 아니라

시스템의 모든 IP 인터페이스를 모니터할 것인지를 지정합니다. 구성 파일을 수정하는 절차는 286 페이지 “IPMP 데몬의 동작을 구성하는 방법”을 참조하십시오.

ipmpstat **유틸리티**는 IPMP 전체의 상태에 대해 여러 유형의 정보를 제공합니다. 또한 이 도구는 그룹에 대해 구성된 데이터 및 테스트 주소뿐 아니라 각 그룹의 기본 IP 인터페이스에 대한 다른 특정 정보도 표시합니다. 이 명령의 사용에 대한 자세한 내용은 289 페이지 “IPMP 정보 모니터링” 및 `ipmpstat(1M)` 매뉴얼 페이지를 참조하십시오.

IPMP 인터페이스 구성 유형

IPMP 구성은 일반적으로 동일한 LAN에 연결된 동일한 시스템에 있는 둘 이상의 물리적 인터페이스로 구성됩니다. 이러한 인터페이스는 다음 구성 중 하나로 IPMP 그룹에 속할 수 있습니다.

- 활성-활성 구성 – 모든 기본 인터페이스가 활성 상태인 IPMP 그룹입니다. **활성 인터페이스**는 현재 IPMP 그룹이 사용할 수 있는 IP 인터페이스입니다. 기본적으로 IPMP 그룹에 속하도록 인터페이스를 구성하면 기본 인터페이스가 활성화됩니다. 활성 인터페이스 및 기타 IPMP 용어에 대한 자세한 내용은 263 페이지 “IPMP 용어 및 개념”을 참조하십시오.
- 활성-대기 구성 – 관리상 하나 이상의 인터페이스가 예비로 구성된 IPMP 그룹입니다. 예비 인터페이스를 **대기 인터페이스**라고 합니다. 유휴 상태이긴 하지만 대기 IP 인터페이스는 인터페이스 구성 방식에 따라 인터페이스의 가용성을 추적하기 위해 다중 경로 지정 데몬에 의해 모니터됩니다. 인터페이스에서 링크 실패 알림을 지원하는 경우 링크 기반 실패 감지가 사용됩니다. 인터페이스가 테스트 주소로 구성된 경우 검사 기반 실패 감지도 사용됩니다. 활성 인터페이스가 실패할 경우 대기 인터페이스가 필요에 따라 자동으로 배포됩니다. IPMP 그룹에 대해 원하는 개수만큼 대기 인터페이스를 구성할 수 있습니다.

단일 인터페이스를 고유한 IPMP 그룹으로 구성할 수도 있습니다. 단일 인터페이스 IPMP 그룹은 여러 인터페이스가 포함된 IPMP 그룹과 동일한 방식으로 동작합니다. 하지만 이 IPMP 구성은 네트워크 트래픽에 고가용성을 제공하지 않습니다. 기본 인터페이스가 실패할 경우 시스템에서 트래픽을 보내거나 받는 기능이 모두 손실됩니다. 단일 인터페이스 IPMP 그룹을 구성하는 목적은 실패 감지를 사용하여 인터페이스의 가용성을 모니터하기 위한 것입니다. 인터페이스에 테스트 주소를 구성하면 검사 기반 실패 감지를 사용하여 인터페이스를 추적할 데몬을 설정할 수 있습니다. 일반적으로 단일 인터페이스 IPMP 그룹 구성은 Oracle Solaris Cluster 소프트웨어와 같이 광범위한 페일오버 기능을 가진 다른 기술과 함께 사용됩니다. 시스템에서 기본 인터페이스의 상태를 계속 모니터할 수 있습니다. 그러나 Oracle Solaris Cluster 소프트웨어는 실패 시 네트워크의 가용성을 보장하는 기능을 제공합니다. Oracle Solaris Cluster 소프트웨어에 대한 자세한 내용은 **Sun Cluster Overview for Solaris OS** 를 참조하십시오.

기본 인터페이스가 제거된 그룹과 같이 기본 인터페이스가 없는 IPMP 그룹도 존재할 수 있습니다. IPMP 그룹은 삭제되지 않지만 이 그룹을 사용하여 트래픽을 보내고 받을 수는

없습니다. 그룹에 대해 기본 IP 인터페이스를 온라인 상태로 전환하면 IPMP 인터페이스의 데이터 주소가 해당 인터페이스에 할당되며 시스템이 네트워크 트래픽 호스트를 계속합니다.

IPMP 주소 지정

IPv4 네트워크와 이중 스택 IPv4 및 IPv6 네트워크에서 모두 IPMP 실패 감지를 구성할 수 있습니다. IPMP를 사용하여 구성된 인터페이스는 다음 두 가지 유형의 주소를 지원합니다.

- **데이터 주소**는 부트 시 DHCP 서버에 의해 동적으로 또는 `ipadm` 명령을 사용하여 수동으로 IP 인터페이스에 할당된 일반적인 IPv4 및 IPv6 주소입니다. IPMP 인터페이스에는 데이터 주소가 할당됩니다. 표준 IPv4 패킷 트래픽과 해당하는 경우 IPv6 패킷 트래픽은 **데이터 트래픽**으로 간주됩니다. 데이터 트래픽 흐름은 IPMP 인터페이스에 호스트된 데이터 주소를 사용하여 해당 그룹의 활성 인터페이스를 통해 전달됩니다.
- **테스트 주소**는 `in.mpathd` 데몬이 검사 기반 실패 및 복구 감지를 수행하기 위해 사용하는 IPMP 관련 주소입니다. 테스트 주소도 DHCP 서버에 의해 동적으로 또는 `ipadm` 명령을 사용하여 수동으로 할당될 수 있습니다. IPMP 인터페이스에는 데이터 주소가 할당되지만 그룹의 기본 인터페이스에는 테스트 주소만 할당됩니다. 이중 스택 네트워크에 있는 기본 인터페이스의 경우 IPv4 테스트 주소, IPv6 테스트 주소 또는 둘 다를 구성할 수 있습니다. 기본 인터페이스가 실패할 경우 `in.mpathd` 데몬은 해당 인터페이스의 테스트 주소를 검사 기반 실패 감지에 계속 사용하여 인터페이스의 후속 복구를 확인합니다.

주 – 구체적으로 검사 기반 실패 감지를 사용하려는 경우에만 테스트 주소를 구성해야 합니다. 그렇지 않으면 테스트 주소를 사용하지 않고 전이적 검사에서 실패를 감지하도록 할 수 있습니다. 테스트 주소를 사용하거나 사용하지 않는 검사 기반 실패 감지에 대한 자세한 내용은 257 페이지 “검사 기반 실패 감지”를 참조하십시오.

이전 IPMP 구현에서는 특히 인터페이스 실패 시 응용 프로그램이 사용하지 않도록 테스트 주소를 `DEPRECATED`로 표시해야 했습니다. 현재 구현에서는 테스트 주소가 기본 인터페이스에 있습니다. 따라서 IPMP를 인식하지 않는 응용 프로그램이 해당 주소를 실수로 사용할 수 없습니다. 하지만 이러한 주소가 가능한 데이터 패킷 소스로 간주되지 않도록 시스템에서 자동으로 `NOFAILOVER` 플래그가 설정된 주소를 `DEPRECATED`로 표시합니다.

IPv4 테스트 주소

일반적으로 서브넷의 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 검사를 교환합니다. RFC 1918 스타일 테스트 주소를 사용하는 경우 해당 RFC 1918 서브넷의 주소를 가진 네트워크에서 다른 시스템(특히 라우터)을 구성해야 합니다. 그러면 in.mpathd 데몬이 대상 시스템과 검사를 성공적으로 교환할 수 있습니다. RFC 1918 개인 주소에 대한 자세한 내용은 RFC 1918, Address Allocation for Private Internets (http://www.ietf.org/rfc/rfc1918.txt?number=1918)를 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 사용이 되거나 ipadm을 통해 인터페이스를 수동으로 구성하면 링크-로컬 주소가 자동으로 구성됩니다.

링크-로컬 주소에 대한 자세한 내용은 **System Administration Guide: IP Services**의 "Link-Local Unicast Address"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.

IPMP의 실패 및 복구 감지

네트워크가 계속해서 트래픽을 보내거나 받을 수 있도록 IPMP는 IPMP 그룹의 기본 IP 인터페이스에서 실패 감지를 수행합니다. 실패한 인터페이스는 복구할 때까지 사용할 수 없습니다. 나머지 활성 인터페이스는 계속 작동하고 기존의 대기 인터페이스가 필요에 따라 배포됩니다.

IPMP의 실패 감지 유형

in.mpathd 데몬은 다음 유형의 실패 감지를 처리합니다.

- 검사 기반 실패 감지. 다음 두 가지 유형이 있습니다.
  - 테스트 주소가 구성되지 않음(전이적 검사)
  - 테스트 주소가 구성됨
- 링크 기반 실패 감지. NIC 드라이버에서 지원되는 경우에 사용합니다.

검사 기반 실패 감지

검사 기반 실패 감지는 ICMP 검사를 사용하여 인터페이스 실패 여부를 확인하는 작업으로 구성됩니다. 이 실패 감지 방법의 구현은 테스트 주소의 사용 여부에 따라 달라집니다.

테스트 주소를 사용하지 않는 검사 기반 실패 감지

테스트 주소를 사용하지 않을 경우 이 방법은 다음 두 가지 검사 유형으로 구현됩니다.

- ICMP 검사

  ICMP 검사는 경로 지정 테이블에 정의된 대상을 검사하기 위해 그룹의 활성 인터페이스에 의해 전송됩니다. **활성** 인터페이스는 인터페이스의 링크 계층(L2) 주소가 지정된 인바운드 IP 패킷을 받을 수 있는 기본 인터페이스입니다. ICMP 검사는 데이터 주소를 검사의 소스 주소로 사용합니다. ICMP 검사가 대상에 도달하고 대상으로부터 응답을 받으면 활성 인터페이스가 작동합니다.

- 전이적 검사

  전이적 검사는 활성 인터페이스를 검사하기 위해 그룹의 대체 인터페이스에 의해 전송됩니다. 대체 인터페이스는 인바운드 IP 패킷을 받지 않는 기본 인터페이스입니다.

  예를 들어, 기본 인터페이스 4개로 구성된 IPMP 그룹을 고려해 보십시오. 이 그룹은 데이터 주소 한 개로 구성되었지만 테스트 주소가 없습니다. 이 구성에서 아웃바운드 패킷은 모두 기본 인터페이스를 사용할 수 있습니다. 하지만 인바운드 패킷은 데이터 주소가 바인딩된 인터페이스만 받을 수 있습니다. 인바운드 패킷을 받을 수 없는 나머지 기본 인터페이스 세 개가 **대체** 인터페이스입니다.

  대체 인터페이스가 성공적으로 활성 인터페이스에 검사를 보내고 응답을 받을 수 있으면 활성 인터페이스가 작동하며 검사를 보낸 대체 인터페이스도 작동하는 것입니다.

주 – 테스트 주소가 필요 없는 이 실패 감지 방법을 사용하려면 전이적 검사를 사용으로 설정해야 합니다.

테스트 주소를 사용하는 검사 기반 실패 감지

이 실패 감지 방법에서는 테스트 주소를 사용하는 ICMP 검사 메시지를 보내고 받습니다. **검사 트래픽** 또는 테스트 트래픽이라고도 하는 이 메시지는 인터페이스를 통해 동일한 로컬 네트워크에 있는 하나 이상의 대상 시스템으로 전송됩니다. 데몬은 검사 기반 실패 감지가 구성된 모든 인터페이스를 통해 모든 대상을 개별적으로 검사합니다. 지정된 인터페이스에 대한 5회 연속 검사에 대해 응답이 없을 경우 `in.mpathd`는 해당 인터페이스가 실패했다고 간주합니다. 검사 속도는 FDT(**실패 감지 시간**)에 따라 달라집니다. 실패 감지 시간의 기본값은 10초입니다. 하지만 IPMP 구성 파일에서 실패 감지 시간을 조정할 수 있습니다. 지침은 286 페이지 “IPMP 데몬의 동작을 구성하는 방법”을 참조하십시오. 검사 기반 실패 감지를 최적화하려면 다중 경로 지정 데몬에서 검사를 받을 대상 시스템을 여러 개 설정해야 합니다. 여러 대상 시스템을 사용하면 보고된 실패의 특성을 확인하는 데 도움이 됩니다. 예를 들어, 정의된 유일한 대상 시스템에서 응답이 없을 경우 시스템이 대상 시스템이나 IPMP 그룹의 인터페이스 중 하나에 실패를 표시할 수 있습니다. 반면, 여러 대상 시스템 중에서 한 시스템만 검사에 응답하지 않는 경우 실패가 IPMP 그룹 자체가 아니라 대상 시스템에서 발생했을 가능성이 큽니다.

in.mpathd 데몬은 동적으로 검사할 대상 시스템을 결정합니다. 먼저 데몬이 IPMP 그룹의 인터페이스와 연결된 테스트 주소와 동일한 서브넷에 있는 대상 시스템을 경로 지정 테이블에서 검색합니다. 이러한 대상이 있으면 데몬이 검사 대상으로 사용합니다. 동일한 서브넷에 대상 시스템이 없는 경우 in.mpathd는 멀티캐스트 패킷을 보내 링크에서 인접한 호스트를 검사합니다. 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 5개 호스트가 검사 대상으로 선택됩니다. in.mpathd가 멀티캐스트 검사에 응답한 라우터나 호스트를 찾을 수 없는 경우 ICMP 에코 패킷인 in.mpathd가 검사 기반 실패를 감지할 수 없습니다. 이 경우 ipmpstat -i 유틸리티는 검사 상태를 unknown으로 보고합니다.

호스트 경로를 사용하여 in.mpathd에서 사용할 대상 시스템 목록을 명시적으로 구성할 수 있습니다. 지침은 284 페이지 “검사 기반 실패 감지 구성”을 참조하십시오.

그룹 실패

IPMP 그룹의 모든 인터페이스가 동시에 실패하면 **그룹 실패**가 발생합니다. 이 경우 기본 인터페이스를 사용할 수 없습니다. 또한 모든 대상 시스템이 동시에 실패하고 검사 기반 실패 감지가 사용으로 설정된 경우 in.mpathd 데몬이 현재 대상 시스템을 모두 비우고 새 대상 시스템을 검사합니다.

테스트 주소가 없는 IPMP 그룹에서는 활성 인터페이스를 검사할 수 있는 단일 인터페이스가 검사자로 지정됩니다. 이 지정된 인터페이스에는 FAILED 플래그와 PROBER 플래그가 모두 설정됩니다. 인터페이스가 복구를 감지하기 위해 대상 검색을 계속할 수 있도록 데이터 주소가 이 인터페이스에 바인딩됩니다.

링크 기반 실패 감지

인터페이스가 이 유형의 실패 감지를 지원하는 경우 링크 기반 실패 감지가 항상 사용으로 설정됩니다.

타사 인터페이스가 링크 기반 실패 감지를 지원하는지 확인하려면 ipmpstat -i 명령을 사용합니다. 지정된 인터페이스에 대한 출력 결과에서 LINK 열이 unknown 상태로 표시되는 경우 해당 인터페이스는 링크 기반 실패 감지를 지원하지 않습니다. 장치에 대한 자세한 내용은 제조업체 설명서를 참조하십시오.

링크 기반 실패 감지를 지원하는 네트워크 드라이버는 인터페이스의 링크 상태를 모니터하고 해당 링크 상태가 변경될 경우 네트워킹 부속 시스템에 알려줍니다. 변경 알림을 받으면 네트워킹 부속 시스템이 해당 인터페이스에 대해 RUNNING 플래그를 적절하게 설정하거나 지웁니다. in.mpathd 데몬이 인터페이스의 RUNNING 플래그가 지워진 것을 감지하면 데몬이 즉시 인터페이스 실패를 발생시킵니다.

실패 감지 및 익명 그룹 기능

IPMP는 익명 그룹의 실패 감지를 지원합니다. 기본적으로 IPMP는 IPMP 그룹에 속하는 인터페이스의 상태만 모니터합니다. 하지만 IPMP 그룹에 속하지 않는 인터페이스의 상태도 추적하도록 IPMP 데몬을 구성할 수 있습니다. 따라서 이 인터페이스는 "익명

그룹"의 일부로 간주됩니다. ipmpstat -g 명령을 실행하면 익명 그룹이 이중 대시(--)로 표시됩니다. 익명 그룹에서 인터페이스의 데이터 주소는 테스트 주소 역할도 수행합니다. 이 인터페이스는 명명된 IPMP 그룹에 속하지 않으므로 해당 주소가 응용 프로그램에 표시됩니다. IPMP 그룹에 속하지 않는 인터페이스 추적을 사용으로 설정하려면 286 페이지 "IPMP 데몬의 동작을 구성하는 방법"을 참조하십시오.

물리적 인터페이스 복구 감지

복구 감지 시간은 실패 감지 시간의 두 배입니다. 실패 감지의 기본 시간은 10초입니다. 이에 따라 복구 감지의 기본 시간은 20초입니다. 실패한 인터페이스에 RUNNING 플래그가 다시 표시되고 실패 감지 방법이 복구된 것으로 감지되면 in.mpathd가 인터페이스의 FAILED 플래그를 지웁니다. 복구된 인터페이스는 관리자가 원래 설정한 활성 인터페이스 수에 따라 재배포됩니다.

기본 인터페이스가 실패하고 검사 기반 실패 감지가 사용되면 테스트 주소가 구성되지 않은 경우 지정된 검사자를 통해 또는 인터페이스의 테스트 주소를 사용하여 in.mpathd 데몬이 검사를 계속합니다. 인터페이스 복구 도중 실패한 인터페이스의 원래 구성에 따라 복원이 계속됩니다.

- 실패한 인터페이스가 원래 활성 인터페이스였던 경우 – 복구된 인터페이스가 원래 활성 상태로 돌아갑니다. 실패 시 대체 역할을 한 대기 인터페이스는 시스템 관리자가 정의한 개수의 활성 인터페이스가 그룹에 있는 경우 다시 대기 상태로 전환됩니다.

  주 – 이 단계의 예외는 복구된 활성 인터페이스가 FAILBACK=no 모드로 구성된 경우입니다. 자세한 내용은 260 페이지 "FAILBACK=no 모드"를 참조하십시오.

- 실패한 인터페이스가 원래 대기 인터페이스였던 경우 – IPMP 그룹이 원래 활성 인터페이스 수를 반영하는 경우 복구된 인터페이스가 원래 대기 상태로 돌아갑니다. 그렇지 않으면 대기 인터페이스가 활성 인터페이스로 전환됩니다.

인터페이스 실패 및 복구 중의 IPMP 동작에 대한 그래픽 표현을 보려면 249 페이지 "IPMP 작동 방식"을 참조하십시오.

FAILBACK=no 모드

기본적으로 실패 후 복구된 활성 인터페이스는 자동으로 그룹의 활성 인터페이스로 돌아갑니다. 이 동작은 데몬 구성 파일에서 FAILBACK 매개변수의 설정에 의해 제어됩니다. 하지만 데이터 주소를 복구된 인터페이스로 재매핑할 때 발생하는 사소한 중단조차 일부 관리자에게는 허용되지 않을 수 있습니다. 이러한 관리자는 활성화된 대기 인터페이스가 활성 인터페이스로 계속 작동하도록 할 수 있습니다. IPMP를 사용하면 관리자가 기본 동작을 대체하여 인터페이스가 복구 시 자동으로 활성화되지 않게 할 수 있습니다. 해당 인터페이스는 FAILBACK=no 모드로 구성해야 합니다. 관련 절차는 286 페이지 "IPMP 데몬의 동작을 구성하는 방법"을 참조하십시오.

FAILBACK=no 모드의 활성 인터페이스가 실패하고 이후에 복구되면 IPMP 데몬이 IPMP 구성을 다음과 같이 복원합니다.

- IPMP 그룹이 활성 인터페이스의 원래 구성을 반영하는 경우 데몬이 인터페이스의 INACTIVE 상태를 유지합니다.
- 복구 시 IPMP 구성이 활성 인터페이스에 대한 그룹의 원래 구성을 반영하지 않는 경우 FAILBACK=no 상태에 관계없이 복구된 인터페이스가 활성 인터페이스로 재배포됩니다.

주 – FAILBACK=NO 모드는 전체 IPMP 그룹에 대해 설정됩니다. 인터페이스별 조정 가능 매개변수는 아닙니다.

IPMP 및 동적 재구성

DR(동적 재구성) 기능을 사용하면 시스템이 실행되는 동안 인터페이스 같은 시스템 하드웨어를 재구성할 수 있습니다. DR은 이 기능을 지원하는 시스템에서만 사용할 수 있습니다.

일반적으로 cfgadm 명령을 사용하여 DR 작업을 수행합니다. 하지만 일부 플랫폼은 다른 방법을 제공합니다. DR 수행에 대한 자세한 내용은 플랫폼 설명서를 참조하십시오. Oracle Solaris를 사용하는 시스템의 경우 표 14–1에 나열된 리소스에서 DR에 대한 특정 설명서를 찾을 수 있습니다. DR에 대한 최신 정보는 http://www.oracle.com/technetwork/indexes/documentation/index.html에서도 제공되며 "dynamic reconfiguration" 항목을 검색하면 확인할 수 있습니다.

표 14–1 동적 재구성에 대한 설명서 리소스

설명	정보
cfgadm 명령에 대한 자세한 정보	cfgadm(1M) 매뉴얼 페이지
Oracle Solaris Cluster 환경의 DR에 대한 특정 정보	**Oracle Solaris Cluster 시스템 관리 설명서**
Oracle Sun 서버의 DR에 대한 특정 정보	특정 서버와 함께 제공된 설명서 참조
DR 및 cfgadm 명령에 대한 소개 정보	**Oracle Solaris 관리: 장치 및 파일 시스템**의 6 장, "동적으로 장치 구성(작업)"
DR을 지원하는 시스템에서 IPMP 그룹을 관리하는 작업	288 페이지 "동적 재구성을 사용하여 IPMP 구성 복구"

이후 절에서는 DR과 IPMP의 상호 운용 방식에 대해 설명합니다.

NIC의 DR을 지원하는 시스템에서는 IPMP를 사용하여 연결을 유지하고 기존 연결의 중단을 방지할 수 있습니다. IPMP는 RCM(Reconfiguration Coordination Manager)

프레임워크에 통합됩니다. 따라서 NIC를 안전하게 연결, 분리 또는 재연결할 수 있으며 RCM이 시스템 구성 요소의 동적 재구성을 관리합니다.

새 NIC 연결

DR 지원을 사용할 경우 새 인터페이스를 연결하고 기존 IPMP 그룹에 추가할 수 있습니다. 또는 해당하는 경우 새로 추가한 인터페이스를 고유한 IPMP 그룹으로 구성할 수 있습니다. IPMP 그룹을 구성하는 절차는 273 페이지 “IPMP 그룹 구성”을 참조하십시오. 이러한 인터페이스는 구성된 후 IPMP에서 즉시 사용할 수 있습니다. 하지만 사용자 정의 링크 이름의 사용 이점을 활용하려면 일반 링크 이름을 지정하여 인터페이스의 하드웨어 기반 링크 이름을 대체해야 합니다. 그런 다음 방금 지정한 일반 이름을 사용하여 해당 구성 파일을 만듭니다. 사용자 정의 링크 이름을 사용하여 단일 인터페이스를 구성하는 절차는 164 페이지 “IP 인터페이스를 구성하는 방법”을 참조하십시오. 인터페이스에 일반 링크 이름을 지정한 후 IPMP에 인터페이스를 사용하는 경우와 같이 인터페이스에서 추가 구성을 수행할 때는 항상 일반 이름을 참조해야 합니다.

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.

검사에 성공하면 데몬이 인터페이스에 대해 `OFFLINE` 플래그를 설정합니다. 인터페이스에서 모든 테스트 주소의 구성이 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 이벤트에 대한 상태 메시지가 표시됩니다. 그렇지 않으면 분리 요청이 성공적으로 완료됩니다. 시스템에서 구성 요소를 제거할 수 있습니다. 기존 연결은 중단되지 않습니다.

NIC 교체

IPMP 그룹의 기본 인터페이스가 실패할 경우 일반적인 솔루션은 새 NIC를 연결하여 실패한 인터페이스를 교체하는 것입니다. RCM은 실행 중인 시스템에서 분리된 NIC와 연결된 구성 정보를 기록합니다. 실패한 NIC를 **동일한** NIC로 교체하는 경우 RCM이 `ipadm` 명령을 사용하여 이전에 정의된 지속 구성에 따라 자동으로 인터페이스를 구성합니다.

예를 들어, 실패한 `bge0` 인터페이스를 다른 `bge0` 인터페이스로 교체한다고 가정합니다. `ipadm` 명령을 사용하여 정의된 실패한 `bge0`의 구성 설정은 지속 설정입니다. 교체 `bge`

NIC를 연결하면 RCM이 bge0 인터페이스를 연결한 후 지속 설정에 따라 구성합니다. 따라서 인터페이스가 테스트 주소로 올바르게 구성되며 IPMP 그룹에 추가됩니다.

둘 다 동일한 유형(예: 이더넷)인 경우 실패한 NIC를 다른 NIC로 교체할 수 있습니다. 이 경우 RCM은 새 인터페이스가 연결된 후에 연결합니다. 인터페이스를 처음 구성할 때 사용자 정의 이름을 사용하지 않은 경우 IPMP 그룹에 인터페이스를 추가하려면 먼저 새 NIC를 구성해야 합니다.

하지만 사용자 정의 링크 이름을 사용한 경우 추가 구성 단계가 필요 없습니다. 실패한 인터페이스의 링크 이름을 새 인터페이스에 재지정하면 새 인터페이스가 제거된 인터페이스의 지속 설정에 지정된 구성을 얻게 됩니다. 그런 다음 RCM이 해당 설정에 따라 인터페이스를 구성합니다. 인터페이스가 실패할 경우 DR을 사용하여 IPMP 구성을 복구하는 절차는 288 페이지 "동적 재구성을 사용하여 IPMP 구성 복구"를 참조하십시오.

IPMP 용어 및 개념

이 절에서는 이 책의 전체 IPMP 장에서 사용되는 용어와 개념을 소개합니다.

활성 인터페이스

시스템에서 데이터 트래픽을 보내거나 받는 데 사용할 수 있는 기본 인터페이스를 나타냅니다. 다음 조건이 충족되면 인터페이스가 활성화됩니다.

- 인터페이스에서 하나 이상의 IP 주소가 UP입니다. UP 주소를 참조하십시오.
- 인터페이스에 FAILED, INACTIVE 또는 OFFLINE 플래그가 설정되어 있지 않습니다.
- 인터페이스에 중복 하드웨어 주소가 있다는 플래그가 지정되지 않았습니다.

사용할 수 없는 인터페이스인 INACTIVE 인터페이스와 비교합니다.

데이터 주소

데이터의 소스 또는 대상 주소로 사용할 수 있는 IP 주소를 나타냅니다. 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. 이전 IPMP 구현에서는 데이터 주소가 IPMP 그룹의 기본 인터페이스에 호스트되었습니다. 현재 구현에서는 데이터 주소가 IPMP 인터페이스에 호스트됩니다.

DEPRECATED 주소

데이터의 소스 주소로 사용할 수 없는 IP 주소를 나타냅니다. 일반적으로 NOFAILOVER 플래그가 지정된 IPMP 테스트 주소는 시스템에서 자동으로 DEPRECATED로 표시됩니다. 하지만 아무 주소나

	DEPRECATED로 표시하여 해당 주소가 소스 주소로 사용되지 않도록 할 수 있습니다.
동적 재구성	시스템이 실행되는 동안 진행 중인 작업에 거의 또는 전혀 영향을 주지 않고 시스템을 재구성할 수 있게 하는 기능을 나타냅니다. Oracle의 모든 Sun 플랫폼에서 DR을 지원하는 것은 아닙니다. 일부 플랫폼은 특정 하드웨어 유형의 DR만 지원합니다. NIC의 DR을 지원하는 플랫폼에서는 DR 도중 시스템에 대한 중단 없는 네트워크 액세스를 위해 IPMP를 사용할 수 있습니다. IPMP가 DR을 지원하는 방식에 대한 자세한 내용은 261 페이지 “IPMP 및 동적 재구성”을 참조하십시오.
명시적 IPMP 인터페이스 생성	현재 IPMP 구현에만 적용됩니다. 이 용어는 ipadm create-ipmp 명령을 사용하여 IPMP 인터페이스를 만드는 방법을 나타냅니다. 명시적 IPMP 인터페이스 생성은 IPMP 그룹을 만드는 기본 방법입니다. 이 방법을 사용하면 관리자가 IPMP 인터페이스 이름과 IPMP 그룹 이름을 설정할 수 있습니다. 암시적 IPMP 인터페이스 생성과 비교합니다.
FAILBACK=no 모드	인터페이스 복구 도중 재배포를 방지하여 인터페이스에 대한 수신 주소의 재바인딩을 최소화하는 기본 인터페이스의 설정을 나타냅니다. 특히, 인터페이스 복구가 감지될 때 인터페이스의 FAILED 플래그가 지워집니다. 하지만 복구된 인터페이스의 모드가 FAILBACK=no이면 작동하는 두번째 인터페이스가 있을 경우 INACTIVE 플래그가 설정되어 해당 인터페이스가 사용되지 않도록 합니다. IPMP 그룹의 두번째 인터페이스가 실패할 경우 INACTIVE 인터페이스가 인계할 수 있습니다. 현재 IPMP 구현에서는 페일백 개념이 더 이상 적용되지 않지만 관리 호환성을 위해 이 모드의 이름이 유지됩니다.
FAILED 인터페이스	in.mpathd 데몬이 오작동하는 것으로 확인한 인터페이스를 나타냅니다. 링크 기반 또는 검사 기반 실패 감지를 통해 확인됩니다. FAILED 플래그는 실패한 모든 인터페이스에 설정됩니다.
실패 감지	물리적 인터페이스 또는 인터페이스에서 인터넷 계층 장치로의 경로가 더 이상 작동하지 않을 경우 이를

	감지하는 프로세스를 나타냅니다. 링크 기반 실패 감지와 검사 기반 실패 감지의 두 가지 실패 감지 형태가 구현됩니다.
암시적 IPMP 인터페이스 생성	`ifconfig` 명령을 통해 존재하지 않는 IPMP 그룹에 기본 인터페이스를 배치하여 IPMP 인터페이스를 만드는 방법을 나타냅니다. 암시적 IPMP 인터페이스 생성은 이전 Oracle Solaris 릴리스의 IPMP 구현과 이전 버전과의 호환성을 위해 지원됩니다. 따라서 이 방법은 IPMP 인터페이스 이름 또는 IPMP 그룹 이름을 설정하는 기능을 제공하지 않습니다. 암시적 IPMP 인터페이스 생성은 `ipadm` 명령에서 지원되지 않습니다. 명시적 IPMP 인터페이스 생성과 비교합니다.
`INACTIVE` 인터페이스	작동하지만 관리 정책에 따라 사용되지 않는 인터페이스를 나타냅니다. `INACTIVE` 플래그는 모든 `INACTIVE` 인터페이스에 설정됩니다. 사용할 수 없는 인터페이스인 활성 인터페이스와 비교합니다.
IPMP 익명 그룹 지원	IPMP 데몬이 IPMP 그룹에 속하는지 여부에 관계없이 시스템에 있는 모든 네트워크 인터페이스의 상태를 추적하는 IPMP 기능을 나타냅니다. 하지만 인터페이스가 실제로 IPMP 그룹에 없는 경우 인터페이스 실패 시 이러한 인터페이스의 주소를 사용할 수 없습니다.
IPMP 그룹	네트워크 가용성과 사용률 향상을 위해 시스템에서 교환 가능한 것으로 처리되는 네트워크 인터페이스 세트를 나타냅니다. 각 IPMP 그룹에는 시스템이 그룹의 임의 활성 인터페이스 세트와 연결할 수 있는 데이터 주소 세트가 있습니다. 이 데이터 주소 세트를 사용하면 네트워크 가용성이 유지 관리되고 네트워크 사용률이 향상됩니다. 관리자는 IPMP 그룹에 배치할 인터페이스를 선택할 수 있습니다. 하지만 동일한 그룹의 모든 인터페이스가 동일한 링크에 연결되고 동일한 프로토콜 세트(예: IPv4 및 IPv6)로 구성되는 등 공통된 등록 정보 세트를 공유해야 합니다.
IPMP 그룹 인터페이스	IPMP 인터페이스를 참조하십시오.
IPMP 그룹 이름	`ipadm set-ifprop` 하위 명령으로 지정할 수 있는 IPMP 그룹의 이름을 나타냅니다. 동일한 IPMP 그룹 이름을 가진 모든 기본 인터페이스가 동일한 IPMP 그룹의

	일부로 정의됩니다. 현재 구현에서는 IPMP 그룹 이름보다 IPMP 인터페이스 이름이 강조됩니다. 관리자는 `ipadm create-ipmp` 하위 명령으로 IPMP 그룹을 만들어 IPMP 인터페이스와 그룹에 동일한 이름을 사용하는 것이 좋습니다.
IPMP 인터페이스	현재 IPMP 구현에만 적용됩니다. 이 용어는 지정된 IPMP 그룹, 인터페이스의 기본 인터페이스 중 하나 또는 모두, 모든 데이터 주소를 나타내는 IP 인터페이스를 가리킵니다. 현재 IPMP 구현에서 IPMP 인터페이스는 IPMP 그룹을 관리하기 위한 핵심 구성 요소이며 경로 지정 테이블, ARP 테이블, 방화벽 규칙 등에서 사용됩니다.
IPMP 인터페이스 이름	IPMP 인터페이스의 이름을 나타냅니다. 이 문서는 이름 지정 규약 `ipmp`*N*을 사용합니다. 시스템은 암시적 IPMP 인터페이스 생성 시에도 동일한 이름 지정 규약을 사용합니다. 하지만 관리자가 암시적 IPMP 인터페이스 생성을 사용하여 임의 이름을 선택할 수 있습니다.
IPMP 싱글톤	데이터 주소가 테스트 주소로도 사용될 수 있게 하는 IPMP 구성을 나타냅니다. 이 구성은 Oracle Solaris Cluster 소프트웨어에서 사용됩니다. 예를 들어, 이 구성은 인터페이스 한 개만 IPMP 그룹에 속해 있을 때 적용됩니다.
링크 기반 실패 감지	인터페이스 상태를 확인하기 위해 네트워크 카드의 링크 상태가 모니터되는 수동 형태의 실패 감지를 지정합니다. 링크 기반 실패 감지는 링크의 작동 여부만 테스트합니다. 이 유형의 실패 감지가 모든 네트워크 카드 드라이버에서 지원되지는 않습니다. 링크 기반 실패 감지는 명시적 구성이 필요 없으며 링크 실패를 즉시 감지합니다. 검사 기반 실패 감지와 비교합니다.
부하 분산	인터페이스 세트에 인바운드 또는 아웃바운드 트래픽을 배포하는 프로세스를 나타냅니다. 로드 균형 조정과 달리 부하 분산은 부하가 균일하게 배포되도록 보장하지 않습니다. 부하 분산을 사용하면 처리량이 증가합니다. 부하 분산은 네트워크 트래픽이 여러 연결을 사용하는 여러 대상으로 흐르는 경우에만 발생합니다.

	인바운드 부하 분산은 IPMP 그룹의 인터페이스 세트에 인바운드 트래픽을 배포하는 프로세스를 나타냅니다. IPMP를 사용하여 인바운드 부하 분산을 직접 제어할 수는 없습니다. 이 프로세스는 소스 주소 선택 알고리즘에 의해 간접적으로 조작됩니다. 아웃바운드 부하 분산은 IPMP 그룹의 인터페이스 세트에 아웃바운드 트래픽을 배포하는 프로세스를 나타냅니다. 아웃바운드 부하 분산은 IP 모듈에서 대상별로 수행되며 필요한 경우 IPMP 그룹의 인터페이스 상태와 구성원에 따라 조정됩니다.
NOFAILOVER 주소	이전 IPMP 구현에만 적용됩니다. 기본 인터페이스와 연결되므로 기본 인터페이스가 실패할 경우 사용할 수 없는 상태로 유지되는 주소를 나타냅니다. 모든 NOFAILOVER 주소에 NOFAILOVER 플래그가 설정됩니다. IPMP 테스트 주소는 NOFAILOVER로 지정해야 하고 IPMP 데이터 주소는 NOFAILOVER로 지정하면 안됩니다. 페일오버 개념은 IPMP 구현에 없습니다. 하지만 NOFAILOVER 용어는 관리 호환성을 위해 유지됩니다.
OFFLINE 인터페이스	일반적으로 시스템에서 제거 준비를 하는 동안 관리상 시스템에서 사용 안함으로 설정된 인터페이스를 나타냅니다. 해당 인터페이스에는 OFFLINE 플래그가 설정되어 있습니다. if_mpadm 명령을 사용하여 인터페이스를 오프라인 상태로 전환할 수 있습니다.
물리적 인터페이스	기본 인터페이스를 참조하십시오.
검사	ping 명령에서 사용되는 패킷과 유사하게, ICMP 패킷을 나타냅니다. 이 검사는 지정된 인터페이스의 송신 및 수신 경로를 테스트하는 데 사용됩니다. 검사 기반 실패 감지가 사용으로 설정된 경우 in.mpathd 데몬이 검사 패킷을 보냅니다. 검사 패킷은 IPMP 테스트 주소를 소스 주소로 사용합니다.
검사 기반 실패 감지	검사 대상과 검사를 교환하여 인터페이스 상태를 확인하는 능동 형태의 실패 감지를 나타냅니다. 사용으로 설정된 경우 검사 기반 실패 감지에서 각 인터페이스의 전체 송신 및 수신 경로를 테스트합니다. 하지만 이 감지 유형에서는 관리자가 각 인터페이스를 테스트 주소로 명시적으로 구성해야 합니다. 링크 기반 실패 감지와 비교합니다.

검사 대상	IPMP 그룹의 인터페이스와 동일한 링크에 있는 시스템을 나타냅니다. 검사 기반 실패 감지를 사용하여 지정된 인터페이스의 상태를 확인할 수 있도록 `in.mpathd` 데몬이 대상을 선택합니다. 검사 대상은 ICMP 검사를 보내고 받을 수 있는 링크의 임의 호스트일 수 있습니다. 검사 대상은 일반적으로 라우터입니다. 일반적으로 여러 검사 대상을 사용하여 검사 대상 자체의 실패로부터 실패 감지 논리를 보호합니다.
소스 주소 선택	IPMP 그룹의 데이터 주소를 특정 패킷에 대한 소스 주소로 선택하는 프로세스를 나타냅니다. 구체적으로 응용 프로그램이 사용할 소스 주소를 선택하지 않은 경우 항상 시스템에서 소스 주소를 선택합니다. 각 데이터 주소가 하드웨어 주소 한 개에만 연결되기 때문에 소스 주소 선택은 인바운드 부하 분산을 간접적으로 제어합니다.
`STANDBY` 인터페이스	관리상 그룹의 다른 인터페이스가 실패한 경우에만 사용되도록 구성된 인터페이스를 나타냅니다. 모든 `STANDBY` 인터페이스에 `STANDBY` 플래그가 설정됩니다.
대상 시스템	검사 대상을 참조하십시오.
테스트 주소	검사의 소스 또는 대상 주소로 사용해야 하고 데이터 트래픽의 소스 또는 대상 주소로 사용하면 안되는 IP 주소를 나타냅니다. 테스트 주소는 기본 인터페이스와 연결됩니다. 기본 인터페이스가 `UP` 테스트 주소로 구성된 경우 `in.mpathd` 데몬은 검사 기반 실패 감지를 사용하여 이 주소를 모니터합니다. 모든 테스트 주소를 `NOFAILOVER`로 지정해야 합니다. 또한 시스템은 이러한 주소를 `DEPRECATED`로 자동 표시하여 데이터 패킷의 가능한 소스 주소로 간주되지 않도록 합니다.
기본 인터페이스	IPMP 그룹에 속하며 실제 네트워크 장치와 직접 연결된 IP 인터페이스를 지정합니다. 예를 들어, `ce0`과 `ce1`이 IPMP 그룹 `ipmp0`에 배치된 경우 `ce0`과 `ce1`이 `ipmp0`의 기본 인터페이스를 구성합니다. 이전 구현에서는 IPMP 그룹이 기본 인터페이스로만 구성됩니다. 하지만 현재 구현에서는 이러한 인터페이스가 그룹을 나타내는 IPMP 인터페이스(예: `ipmp0`)의 기반을 이룹니다(이름).
실행 취소-오프라인 작업	이전에 오프라인 상태였던 인터페이스를 관리상 시스템에서 사용할 수 있게 하는 작업을 나타냅니다.

	if_mpadm 명령을 사용하여 실행 취소-오프라인 작업을 수행할 수 있습니다.
사용할 수 없는 인터페이스	현재 구성에서 데이터 트래픽을 보내거나 받는 데 사용할 수 없는 기본 인터페이스를 나타냅니다. 사용할 수 없는 인터페이스는 현재 사용되고 있지 않지만 그룹의 활성 인터페이스를 사용할 수 없을 경우 사용할 수 있는 INACTIVE 인터페이스와는 다릅니다. 다음 조건 중 하나가 있으면 인터페이스를 사용할 수 없습니다. ■ 인터페이스에 UP 주소가 없습니다. ■ 인터페이스에 FAILED 또는 OFFLINE 플래그가 설정되었습니다. ■ 인터페이스에 그룹의 다른 인터페이스와 동일한 하드웨어 주소가 있다는 플래그가 지정되었습니다.
UP 주소	UP 플래그를 설정하여 관리상 시스템에서 사용할 수 있게 된 주소를 나타냅니다. UP이 아닌 주소는 시스템에 속하지 않는 것으로 간주되므로 소스 주소 선택 도중 고려되지 않습니다.

◆ ◆ ◆ 15장

15 IPMP 관리

이 장에서는 IPMP(IP Network Multipathing)를 사용하여 인터페이스 그룹을 관리하는 작업을 제공합니다. 다음 주요 내용으로 구성되어 있습니다.

IPMP 관리 작업 맵

Oracle Solaris에서는 `ipmpstat` 명령이 IPMP 그룹 정보를 가져오는 데 사용하는 기본 도구입니다. 이 장에서는 `ipmpstat` 명령이 이전 Oracle Solaris 릴리스에서 IPMP 정보를 제공하는 데 사용된 `ifconfig` 명령의 특정 함수를 대체합니다.

`ipmpstat` 명령의 여러 옵션에 대한 자세한 내용은 289 페이지 “IPMP 정보 모니터링”을 참조하십시오.

다음 절에서는 이 장에 포함된 작업에 대한 링크를 제공합니다.

IPMP 그룹 만들기 및 구성(작업 맵)

작업	설명	수행 방법
IPMP 그룹을 계획합니다.	IPMP 그룹을 구성하기 전에 필요한 작업과 보조 정보를 모두 나열합니다.	273 페이지 “IPMP 그룹을 계획하는 방법”

작업	설명	수행 방법
DHCP를 사용하여 IPMP 그룹을 구성합니다.	DHCP를 사용하여 IPMP 그룹을 구성하는 대체 방법을 제공합니다.	275 페이지 “DHCP를 사용하여 IPMP 그룹을 구성하는 방법”
활성-활성 IPMP 그룹을 구성합니다.	모든 기본 인터페이스가 호스트 네트워크 트래픽에 배포되는 IPMP 그룹을 구성합니다.	277 페이지 “활성-활성 IPMP 그룹을 수동으로 구성하는 방법”
활성-대기 IPMP 그룹을 구성합니다.	기본 인터페이스 한 개가 비활성 예비 상태로 유지되는 IPMP 그룹을 구성합니다.	279 페이지 “활성-대기 IPMP 그룹을 수동으로 구성하는 방법”

IPMP 그룹 유지 관리(작업 맵)

작업	설명	수행 방법
IPMP 그룹에 인터페이스를 추가합니다.	기존 IPMP 그룹의 구성원으로 새 인터페이스를 구성합니다.	281 페이지 “IPMP 그룹에 인터페이스를 추가하는 방법”
IPMP 그룹에서 인터페이스를 제거합니다.	IPMP 그룹에서 인터페이스를 제거합니다.	281 페이지 “IPMP 그룹에서 인터페이스를 제거하는 방법”
IPMP 그룹에 IP 주소를 추가하거나 제거합니다.	IPMP 그룹의 주소를 추가하거나 제거합니다.	282 페이지 “IP 주소를 추가하거나 제거하는 방법”
인터페이스의 IPMP 구성원을 변경합니다.	IPMP 그룹 간에 인터페이스를 이동합니다.	283 페이지 “한 IPMP 그룹에서 다른 그룹으로 인터페이스를 이동하는 방법”
IPMP 그룹을 삭제합니다.	더 이상 필요하지 않은 IPMP 그룹을 삭제합니다.	284 페이지 “IPMP 그룹을 삭제하는 방법”
실패한 카드를 교체합니다.	IPMP 그룹에서 실패한 NIC를 제거하거나 교체합니다.	288 페이지 “실패한 물리적 카드를 교체하는 방법”

검사 기반 실패 감지 구성(작업 맵)

작업	설명	수행 방법
대상 시스템을 수동으로 지정합니다.	검사 기반 실패 감지의 대상 시스템을 식별하고 추가합니다.	285 페이지 “검사 기반 실패 감지의 대상 시스템을 수동으로 지정하는 방법”
검사 기반 실패 감지의 동작을 구성합니다.	매개변수를 수정하여 검사 기반 실패 감지의 동작을 결정합니다.	286 페이지 “IPMP 데몬의 동작을 구성하는 방법”

IPMP 그룹 모니터링(작업 맵)

작업	설명	수행 방법
그룹 정보를 가져옵니다.	IPMP 그룹에 대한 정보를 표시합니다.	289 페이지 "IPMP 그룹 정보를 가져오는 방법"
데이터 주소 정보를 가져옵니다.	IPMP 그룹에서 사용하는 데이터 주소에 대한 정보를 표시합니다.	290 페이지 "IPMP 데이터 주소 정보를 가져오는 방법"
IPMP 인터페이스 정보를 가져옵니다.	IPMP 인터페이스 또는 그룹의 기본 인터페이스에 대한 정보를 표시합니다.	291 페이지 "그룹의 기본 IP 인터페이스에 대한 정보를 가져오는 방법"
검사 대상 정보를 가져옵니다.	검사 기반 실패 감지의 대상에 대한 정보를 표시합니다.	293 페이지 "IPMP 검사 대상 정보를 가져오는 방법"
검사 정보를 가져옵니다.	시스템에서 진행 중인 검사에 대한 실시간 정보를 표시합니다.	294 페이지 "IPMP 검사를 관찰하는 방법"
IPMP 그룹을 모니터하기 위해 정보 표시를 사용자 정의합니다.	표시되는 IPMP 정보를 결정합니다.	295 페이지 "스크립트에서 `ipmpstat` 명령의 출력 결과를 사용자 정의하는 방법"

IPMP 그룹 구성

이 절에서는 IPMP 그룹을 계획하고 구성하는 데 사용되는 절차를 제공합니다. 14 장, "IPMP 소개"의 개요에서는 인터페이스로서 IPMP 그룹의 구현에 대해 설명합니다. 따라서 이 장에서 *IPMP* **그룹** 및 *IPMP* **인터페이스**란 용어는 같은 의미로 사용됩니다.

▼ IPMP 그룹을 계획하는 방법

다음 절차에는 IPMP 그룹을 구성하기 전에 필요한 계획 작업 및 수집할 정보가 포함되어 있습니다. 작업을 순서대로 수행할 필요는 없습니다.

주 – 각 서브넷 또는 L2 브로드캐스트 도메인에 대해 IPMP 그룹 한 개만 구성해야 합니다. 자세한 내용은 247 페이지 "IPMP 사용 시기"를 참조하십시오.

---

1 요구에 맞는 일반 IPMP 구성을 결정합니다.

IPMP 구성은 네트워크에서 시스템에 호스트된 트래픽 유형을 처리하는 데 필요한 사항에 따라 달라집니다. IPMP는 아웃바운드 네트워크 패킷을 IPMP 그룹의 인터페이스에 분산시키므로 네트워크 처리량이 향상됩니다. 하지만 지정된 TCP 연결에 대해 인바운드 트래픽은 잘못된 순서로 패킷을 처리하는 위험을 최소화하기 위해 대체로 하나의 물리적 경로만 따릅니다.

따라서 네트워크에서 많은 아웃바운드 트래픽을 처리하는 경우 IPMP 그룹에 다수의 인터페이스를 구성하면 네트워크 성능이 향상될 수 있습니다. 대신 시스템에서 많은 인바운드 트래픽을 호스트하는 경우 그룹에 포함된 인터페이스 수가 많아도 반드시 트래픽 부하 분산에 의해 성능이 향상되는 것은 아닙니다. 하지만 기본 인터페이스가 많으면 인터페이스 실패 시 네트워크 가용성을 보장하는 데 도움이 됩니다.

2 SPARC 기반 시스템의 경우 그룹의 각 인터페이스에 고유한 MAC 주소가 있는지 확인합니다.

시스템의 각 인터페이스에 대해 고유한 MAC 주소를 구성하려면 163 페이지 “SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법”을 참조하십시오.

3 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, `net0` 인터페이스의 `ifconfig` 출력 결과는 다음과 같습니다.

```
# ifconfig net0 modlist
    0 arp
    1 ip
    2 e1000g
```

출력 결과와 같이 인터페이스는 대체로 IP 모듈 바로 아래에 네트워크 드라이브로 존재합니다. 이러한 인터페이스에는 추가 구성이 필요 없어야 합니다.

하지만 특정 기술은 IP 모듈과 네트워크 드라이버 간에 STREAMS 모듈로 삽입됩니다. 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*`@`*position*

```
ifconfig net0 modinsert vpnmod@3
```

4 IPMP 그룹의 모든 인터페이스에서 동일한 IP 주소 형식을 사용합니다.

IPv4에 대해 한 인터페이스가 구성된 경우 그룹의 모든 인터페이스를 IPv4에 대해 구성해야 합니다. 예를 들어, 한 인터페이스에 IPv6 주소 지정을 추가하는 경우 IPMP 그룹의 모든 인터페이스에서 IPv6 지원을 구성해야 합니다.

5 **구현하려는 실패 감지 유형을 결정합니다.**

예를 들어, 검사 기반 실패 감지를 구현하려는 경우 기본 인터페이스에 테스트 주소를 구성해야 합니다. 관련 정보는 257 페이지 “IPMP의 실패 감지 유형”을 참조하십시오.

6 **IPMP 그룹의 모든 인터페이스가 동일한 로컬 네트워크에 연결되어 있는지 확인합니다.**

예를 들어, 동일한 IP 서브넷의 이더넷 스위치를 IPMP 그룹으로 구성할 수 있습니다. 임의 개수의 인터페이스를 IPMP 그룹으로 구성할 수 있습니다.

주 – 예를 들어, 시스템에 물리적 인터페이스가 하나뿐인 경우 단일 인터페이스 IPMP 그룹을 구성할 수도 있습니다. 관련 정보는 255 페이지 “IPMP 인터페이스 구성 유형”을 참조하십시오.

---

7 **IPMP 그룹에 서로 다른 네트워크 매체 유형의 인터페이스가 포함되지 않도록 합니다.**

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.

▼ DHCP를 사용하여 IPMP 그룹을 구성하는 방법

현재 IPMP 구현에서는 DHCP(Dynamic Host Configuration Protocol) 지원을 사용하여 IPMP 그룹을 구성할 수 있습니다.

활성-활성 인터페이스나 활성-대기 인터페이스를 사용하여 다중 인터페이스 IPMP 그룹을 구성할 수 있습니다. 관련 정보는 255 페이지 “IPMP 인터페이스 구성 유형”을 참조하십시오. 다음 절차에서는 DHCP를 사용하여 활성-대기 IPMP 그룹을 구성하는 단계에 대해 설명합니다.

시작하기 전에 IPMP 그룹에 포함될 IP 인터페이스가 시스템의 네트워크 데이터 링크에서 올바르게 구성되었는지 확인합니다. 기본 IP 인터페이스가 없는 경우에도 IPMP 인터페이스를 만들 수 있습니다. 하지만 이 IPMP 인터페이스의 후속 구성이 실패합니다.

링크와 IP 인터페이스를 구성하는 절차는 162 페이지 “IP 인터페이스 구성(작업)”을 참조하십시오. IPv6 인터페이스 구성에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 “IPv6 인터페이스 구성”을 참조하십시오.

또한 SPARC 시스템을 사용하는 경우 각 인터페이스에 고유한 MAC 주소를 구성합니다. 절차는 163 페이지 “SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법”을 참조하십시오.

마지막으로, DHCP를 사용하는 경우 기본 인터페이스에 무기한 임대가 있는지 확인합니다. 그렇지 않으면 그룹 실패 시 테스트 주소가 만료되고 IPMP 데몬이 검사 기반 실패 감지를 사용 안함으로 설정하여 링크 기반 실패 감지가 사용됩니다. 링크 기반 실패 감지에서 인터페이스가 작동 중으로 검색되면 인터페이스가 복구되었다고 데몬이 잘못 보고할 수 있습니다. DHCP 구성에 대한 자세한 내용은 **System Administration Guide: IP Services**의 13 장, "Planning for DHCP Service (Tasks)"을 참조하십시오.

주 – 시스템의 활성 네트워크 프로파일이 반응적 프로파일인 경우 IPMP를 사용할 수 없습니다. IPMP 그룹을 구성하기 전에 필요한 경우 `DefaultFixed` 프로파일을 사용으로 설정하여 수정된 네트워크 구성 프로파일로 전환합니다. 절차는 138 페이지 "프로파일 및 구성 도구"를 참조하십시오.

---

1 관리자로 전환합니다.

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

2 IPMP 인터페이스를 만듭니다.

```
# ipadm create-ipmp ipmp-interface
```

구문 설명은 다음과 같습니다.

*ipmp-interface*는 IPMP 인터페이스의 이름을 지정합니다. IPMP 인터페이스에 의미 있는 이름을 지정할 수 있습니다. 모든 IP 인터페이스와 마찬가지로 이름은 문자열과 숫자로 구성됩니다(예: `ipmp0`).

3 아직 없는 경우 기본 IP 인터페이스를 만듭니다.

```
# ipadm create-ip under-interface
```

여기서 *under-interface*는 IPMP 그룹에 추가할 IP 인터페이스를 나타냅니다.

4 테스트 주소가 포함될 기본 IP 인터페이스를 IPMP 그룹에 추가합니다.

```
# ipadm add-ipmp -i under-interface1 [-i under-interface2 ...] ipmp-interface
```

시스템에서 사용 가능한 개수만큼 IP 인터페이스를 IPMP 그룹에 대해 만들 수 있습니다.

5 DHCP가 IPMP 인터페이스의 데이터 주소를 구성하고 관리하게 합니다.

```
# ipadm create-addr -T dhcp addrobj
```

*addrobj*는 주소 객체를 나타내며 *interface*/*string* 형식을 사용합니다. 이 단계의 *interface*는 IPMP 인터페이스입니다. 문자열은 임의의 사용자 정의 문자열일 수 있습니다. 따라서 IPMP 인터페이스에 데이터 주소가 여러 개 있는 경우 해당 주소 객체는 *ipmp-interface*/*string1*, *ipmp-interface*/*string2*, *ipmp-interface*/*string3* 등이 됩니다.

6 **DHCP가 기본 인터페이스의 테스트 주소를 관리하게 합니다.**

IPMP 그룹의 각 기본 인터페이스에 대해 다음 명령을 실행해야 합니다.

```
# ipadm create-addr -T dhcp addrobj
```

*addrobj*는 주소 객체를 나타내며 *interface*/*string* 형식을 사용합니다. 이 단계의 *interface*는 기본 인터페이스입니다. 문자열은 임의의 사용자 정의 문자열일 수 있습니다. 따라서 IPMP 그룹에 대한 기본 인터페이스가 여러 개 있는 경우 해당 주소 객체는 *under-interface1*/*string*, *ipmp-interface2*/*string*, *ipmp-interface3*/*string* 등이 됩니다.

예 15–1 DHCP를 사용하여 IPMP 그룹 구성

이 예에서는 DHCP를 사용하여 활성-대기 IPMP 그룹을 구성하는 방법을 보여주며 다음 시나리오를 기반으로 합니다.

- IPMP 그룹의 기본 인터페이스 세 개가 IPMP 그룹의 지정된 구성원인 해당 데이터 링크 `net0`, `net1` 및 `net2`에 구성됩니다.
- IPMP 인터페이스 `itops0`은 IPMP 그룹과 동일한 이름을 공유합니다.
- `net2`는 지정된 대기 인터페이스입니다.
- 검사 기반 실패 감지를 사용하기 위해 모든 기본 인터페이스에 테스트 주소가 할당됩니다.

```
# ipadm create-ipmp itops0

# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2

# ipadm add-ipmp -i net0 -i net1 -i net2 itops0

# ipadm create-addr -T dhcp itops0/dhcp0
# ipadm create-addr -T dhcp itops0/dhcp1

# ipadm create-addr -T dhcp net0/test
# ipadm create-addr -T dhcp net2/test
# ipadm create-addr -T dhcp net3/test

# ipadm set-ifprop -p standby=on net2
```

▼ 활성-활성 IPMP 그룹을 수동으로 구성하는 방법

다음 절차에서는 활성-활성 IPMP 그룹을 수동으로 구성하는 단계에 대해 설명합니다.

시작하기 전에 잠재 IPMP 그룹에 포함될 IP 인터페이스가 시스템의 네트워크 데이터 링크에서 올바르게 구성되었는지 확인합니다. 링크와 IP 인터페이스를 구성하는 절차는 162 페이지 “IP 인터페이스 구성(작업)”을 참조하십시오. IPv6 인터페이스 구성에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 “IPv6 인터페이스 구성”을 참조하십시오.

기본 IP 인터페이스가 없는 경우에도 IPMP 인터페이스를 만들 수 있습니다. 하지만 이 IPMP 인터페이스의 후속 구성이 실패합니다.

또한 SPARC 시스템을 사용하는 경우 각 인터페이스에 고유한 MAC 주소를 구성합니다. 절차는 163 페이지 "SPARC: 인터페이스의 MAC 주소가 고유한지 확인하는 방법"을 참조하십시오.

1 관리자로 전환합니다.

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

2 IPMP 인터페이스를 만듭니다.

```
# ipadm create-ipmp ipmp-interface
```

구문 설명은 다음과 같습니다.

*ipmp-interface*는 IPMP 인터페이스의 이름을 지정합니다. IPMP 인터페이스에 의미 있는 이름을 지정할 수 있습니다. 모든 IP 인터페이스와 마찬가지로 이름은 문자열과 숫자로 구성됩니다(예: `ipmp0`).

3 기본 IP 인터페이스를 그룹에 추가합니다.

```
# ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
```

여기서 *under-interface*는 IPMP 그룹의 기본 인터페이스를 나타냅니다. 시스템에서 사용 가능한 개수만큼 IP 인터페이스를 추가할 수 있습니다.

주 – 이중 스택 환경에서 인터페이스의 IPv4 인스턴스를 특정 그룹 아래에 배치하면 IPv6 인스턴스도 동일한 그룹 아래에 자동으로 배치됩니다.

---

4 IPMP 인터페이스에 데이터 주소를 추가합니다.

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address*는 CIDR 표기법을 사용할 수 있습니다.

*addrobj*는 이름 지정 규약 *ipmp-interface*/*any-string*을 사용해야 합니다. 따라서 IPMP 인터페이스의 이름이 `ipmp0`인 경우 *addrobj*는 `ipmp0/dataaddr`일 수 있습니다.

5 기본 인터페이스에 테스트 주소를 추가합니다.

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address*는 CIDR 표기법을 사용할 수 있습니다.

*addrobj*는 이름 지정 규약 *under-interface*/*any-string*을 사용해야 합니다. 따라서 기본 인터페이스의 이름이 `net0`인 경우 *addrobj*는 `net0/testaddr`일 수 있습니다.

주 – 특정 인터페이스에서 검사 기반 실패 감지를 사용하려는 경우에만 테스트 주소를 구성해야 합니다.

IPMP 그룹의 모든 테스트 IP 주소는 동일한 네트워크 접두어를 사용해야 합니다. 테스트 IP 주소는 단일 IP 서브넷에 속해야 합니다.

▼ 활성-대기 IPMP 그룹을 수동으로 구성하는 방법

대기 인터페이스에 대한 자세한 내용은 255 페이지 "IPMP 인터페이스 구성 유형"을 참조하십시오. 다음 절차에서는 인터페이스 한 개가 예비 상태로 유지되는 IPMP 그룹을 구성합니다. 이 인터페이스는 그룹의 활성 인터페이스가 실패하는 경우에만 배포됩니다.

1 관리자로 전환합니다.

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

2 IPMP 인터페이스를 만듭니다.

```
# ipadm create-ipmp ipmp-interface
```

구문 설명은 다음과 같습니다.

*ipmp-interface*는 IPMP 인터페이스의 이름을 지정합니다. IPMP 인터페이스에 의미 있는 이름을 지정할 수 있습니다. 모든 IP 인터페이스와 마찬가지로 이름은 문자열과 숫자로 구성됩니다(예: `ipmp0`).

3 기본 IP 인터페이스를 그룹에 추가합니다.

```
# ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
```

여기서 *under-interface*는 IPMP 그룹의 기본 인터페이스를 나타냅니다. 시스템에서 사용 가능한 개수만큼 IP 인터페이스를 추가할 수 있습니다.

주 – 이중 스택 환경에서 인터페이스의 IPv4 인스턴스를 특정 그룹 아래에 배치하면 IPv6 인스턴스도 동일한 그룹 아래에 자동으로 배치됩니다.

4 IPMP 인터페이스에 데이터 주소를 추가합니다.

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address*는 CIDR 표기법을 사용할 수 있습니다.

*addrobj*는 이름 지정 규약 *ipmp-interface*/*any-string*을 사용해야 합니다. 따라서 IPMP 인터페이스의 이름이 `ipmp0`인 경우 *addrobj*는 `ipmp0/dataaddr`일 수 있습니다.

5 기본 인터페이스에 테스트 주소를 추가합니다.

```
# ipadm create-addr -T static IP-address addrobj
```

*IP-address*는 CIDR 표기법을 사용할 수 있습니다.

*addrobj*는 이름 지정 규약 *under-interface*/*any-string*을 사용해야 합니다. 따라서 기본 인터페이스의 이름이 `net0`인 경우 *addrobj*는 `net0/testaddr`일 수 있습니다.

주 – 특정 인터페이스에서 검사 기반 실패 감지를 사용하려는 경우에만 테스트 주소를 구성해야 합니다.

IPMP 그룹의 모든 테스트 IP 주소는 동일한 네트워크 접두어를 사용해야 합니다. 테스트 IP 주소는 단일 IP 서브넷에 속해야 합니다.

---

6 기본 인터페이스 중 하나를 대기 인터페이스로 구성합니다.

```
# ipadm set-ifprop -p standby=yes under-interface
```

예 15–2 활성-대기 IPMP 그룹 구성

이 예에서는 활성-대기 IPMP 구성을 수동으로 만드는 방법을 보여줍니다. 이 예는 기본 인터페이스를 만드는 작업으로 시작됩니다.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2

# ipadm create-ipmp itops0

# ipadm add-ipmp -i net0 -i net1 -i net2 itops0
# ipadm create-addr -T static -a 192.168.10.10/24 itops0/v4add1
# ipadm create-addr -T static -a 192.168.10.15/24 itops0/v4add2

# ipadm create-addr -T static -a 192.168.85.30/24 net0/test
# ipadm create-addr -T static -a 192.168.85.32/24 net1/test
# ipadm create-addr -T static -a 192.168.85.34/24 net2/test

# ipadm set-ifprop -p standby=yes net2

# ipmpstat -g
GROUP     GROUPNAME   STATE      FDT        INTERFACES
itops0    itops0      ok         10.00s     net0 net1 (net2)

# ipmpstat -t
INTERFACE  MODE     TESTADDR        TARGETS
net0       routes   192.168.10.30   192.168.10.1
net1       routes   192.168.10.32   192.168.10.1
net2       routes   192.168.10.34   192.168.10.5
```

IPMP 그룹 유지 관리

이 절에는 기존 IPMP 그룹과 해당 그룹 내의 인터페이스를 유지 관리하는 작업이 포함되어 있습니다. 이 작업에서는 273 페이지 "IPMP 그룹 구성"에 설명된 대로 IPMP 그룹을 이미 구성했다고 가정합니다.

▼ IPMP 그룹에 인터페이스를 추가하는 방법

시작하기 전에 그룹에 추가하는 인터페이스가 그룹에 속하기 위한 모든 제약 조건을 충족하는지 확인합니다. IPMP 그룹의 요구 사항 목록은 273 페이지 "IPMP 그룹을 계획하는 방법"을 참조하십시오.

1. **관리자로 전환합니다.**

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

2. **기본 IP 인터페이스가 없는 경우 인터페이스를 만듭니다.**

   # ipadm create-ip *interface*

3. **IPMP 그룹에 IP 인터페이스를 추가합니다.**

   # ipadm add-ipmp -i *under-interface ipmp-interface*

예 15–3 IPMP 그룹에 인터페이스 추가

IPMP 그룹 `itops0`에 `net4` 인터페이스를 추가하려면 다음 명령을 입력합니다.

```
# ipadm create-ip net4
# ipadm add-ipmp -i net4 itops0
# ipmpstat -g
GROUP    GROUPNAME   STATE      FDT       INTERFACES
itops0   itops0      ok         10.00s    net0 net1 net4
```

▼ IPMP 그룹에서 인터페이스를 제거하는 방법

1. **관리자로 전환합니다.**

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

2. **IPMP 그룹에서 인터페이스를 제거합니다.**

   # ipadm remove-ipmp -i *under-interface*[**, -i** *under-interface*, **...**] *ipmp-interface*

   단일 명령으로 기본 인터페이스를 필요한 개수만큼 제거할 수 있습니다. 기본 인터페이스를 모두 제거해도 IPMP 인터페이스가 삭제되지는 않습니다. 대신 빈 IPMP 인터페이스 또는 그룹으로 존재합니다.

예 15–4 그룹에서 인터페이스 제거

IPMP 그룹 itops0에서 net4 인터페이스를 제거하려면 다음 명령을 입력합니다.

```
# ipadm remove-ipmp net4 itops0
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
itops0  itops0      ok         10.00s    net0 net1
```

▼ IP 주소를 추가하거나 제거하는 방법

ipadm create-addr 하위 명령을 사용하여 주소를 추가하거나 ipadm delete-addr 하위 명령을 사용하여 인터페이스에서 주소를 제거합니다. 현재 IPMP 구현에서 테스트 주소는 기본 IP 인터페이스에 호스트되고 데이터 주소는 IPMP 인터페이스에 할당됩니다. 다음 절차에서는 테스트 주소 또는 데이터 주소인 IP 주소를 추가하거나 제거하는 방법에 대해 설명합니다.

1 관리자로 전환합니다.

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

2 데이터 주소를 추가하거나 제거합니다.

- IPMP 그룹에 데이터 주소를 추가하려면 다음 명령을 입력합니다.

  # ipadm create-addr -T static -a *ip-address addrobj*

  *addrobj*는 이름 지정 규약 *ipmp-interface*/*user-string*을 사용합니다.
- IPMP 그룹에서 주소를 제거하려면 다음 명령을 입력합니다.

  # ipadm delete-addr *addrobj*

  *addrobj*는 이름 지정 규약 *inder-interface*/*user-string*을 사용합니다.

3 테스트 주소를 추가하거나 제거합니다.

- IPMP 그룹의 기본 인터페이스에 테스트 주소를 할당하려면 다음 명령을 입력합니다.

  # ipadm create-addr -T static *ip-address adrobj*
- IPMP 그룹의 기본 인터페이스에서 테스트 주소를 제거하려면 다음 명령을 입력합니다.

  # ipadm delete-addr *addrobj*

예 15–5 인터페이스에서 테스트 주소 제거

다음 예에서는 예 15–2의 itops0 구성을 사용합니다. 이 단계는 net1 인터페이스에서 테스트 주소를 제거합니다. 이 예에서 테스트 주소는 net1/test1로 지정되었다고 가정합니다.

```
# ipmpstat -t
INTERFACE      MODE     TESTADDR         TARGETS
net1      routes   192.168.10.30   192.168.10.1

# ipadm show-addr
ADDROBJ           TYPE     STATE     ADDR
lo0               static   ok        127.0.0.1/8
...
net1/test1        static   ok        192.168.10.30

# ipadm delete-addr net1/test1
```

▼ 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 관리자로 전환합니다.

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

2 인터페이스를 새 IPMP 그룹으로 이동합니다.

 # ipadm add-ipmp -i *under-interface ipmp-interface*

여기서 *under-interface*는 이동할 기본 인터페이스를 나타내고 *ipmp-interface*는 기본 인터페이스를 이동할 IPMP 인터페이스 또는 그룹을 나타냅니다.

새 그룹에 인터페이스를 배치하면 기존 그룹에서 해당 인터페이스가 자동으로 제거됩니다.

예 15–6 다른 IPMP 그룹으로 인터페이스 이동

이 예에서는 그룹의 기본 인터페이스가 `net0`, `net11` 및 `net2`라고 가정합니다. `net0`을 IPMP 그룹 `cs-link1`로 이동하려면 다음을 입력합니다.

```
# ipadm add-ipmp -i net0 ca-link1
```

이 명령은 IPMP 그룹 `itops0`에서 `net0` 인터페이스를 제거하고 `net0`을 `cs-link1`에 넣습니다.

▼ IPMP 그룹을 삭제하는 방법

특정 IPMP 그룹이 더 이상 필요하지 않은 경우 이 절차를 사용합니다.

1 관리자로 전환합니다.

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

2 IPMP 그룹 및 기본 IP 인터페이스를 식별합니다.

```
# ipmpstat -g
```

3 현재 IPMP 그룹에 속하는 IP 인터페이스를 모두 삭제합니다.

```
# ipadm remove-ipmp -i under-interface[, -i under-interface, ...] ipmp-interface
```

주 – IPMP 인터페이스를 성공적으로 삭제하려면 IPMP 그룹에 속한 IP 인터페이스가 없어야 합니다.

---

4 IPMP 인터페이스를 삭제합니다.

```
# ipadm delete-ipmp ipmp-interface
```

IPMP 인터페이스를 삭제하면 해당 인터페이스와 연결된 모든 IP 주소가 시스템에서 삭제됩니다.

예 15–7 IPMP 인터페이스 삭제

기본 IP 인터페이스 `net0`과 `net1`이 있는 `itops0` 인터페이스를 삭제하려면 다음 명령을 입력합니다.

```
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT        INTERFACES
itops0  itops0      ok         10.00s     net0 net1

# ipadm remove-ipmp -i net0 -i net1 itops0

# ipadm delete-ipmp itops0
```

검사 기반 실패 감지 구성

검사 기반 실패 감지의 경우 257 페이지 "검사 기반 실패 감지"에 설명된 대로 대상 시스템을 사용해야 합니다. 검사 기반 실패 감지의 대상을 식별할 때 `in.mpathd` 데몬은 라우터 대상 모드나 멀티캐스트 대상 모드의 두 가지 모드로 작동합니다. 라우터 대상 모드에서는 다중 경로 데몬이 경로 지정 테이블에 정의된 대상을 검사합니다. 대상을 정의하지 않은 경우 데몬이 멀티캐스트 대상 모드로 작동합니다. 이 모드에서는 멀티캐스트 패킷이 전송되어 LAN에서 인접한 호스트를 검사합니다.

검사할 in.mpathd 데몬에 대해 호스트 대상을 설정해야 합니다. 일부 IPMP 그룹의 경우 기본 라우터를 대상으로 사용해도 됩니다. 하지만 검사 기반 실패 감지에 대해 특정 대상을 구성해야 하는 IPMP 그룹도 있습니다. 대상을 지정하려면 경로 지정 테이블의 호스트 경로를 검사 대상으로 설정합니다. 경로 지정 테이블에 구성된 호스트 경로는 기본 라우터 앞에 나열됩니다. IPMP는 명시적으로 정의된 호스트 경로를 대상 선택으로 사용합니다. 따라서 기본 라우터를 사용하는 대신 호스트 경로를 설정하여 특정 검사 대상을 구성해야 합니다.

경로 지정 테이블에 호스트 경로를 설정하려면 route 명령을 사용합니다. 이 명령과 함께 -p 옵션을 사용하여 지속 경로를 추가할 수 있습니다. 예를 들어, route -p add는 시스템을 재부트한 후에도 경로 지정 테이블에 유지될 경로를 추가합니다. 따라서 -p 옵션을 사용하면 시스템을 시작할 때마다 경로를 다시 만드는 특수 스크립트 없이도 지속 경로를 추가할 수 있습니다. 검사 기반 실패 감지를 최적으로 사용하려면 검사를 받을 대상을 여러 개 설정해야 합니다.

뒤에 나오는 샘플 절차에서는 검사 기반 실패 감지를 위해 대상에 지속 경로를 추가하는 정확한 구문을 보여줍니다. route 명령의 옵션에 대한 자세한 내용은 route(1M) 매뉴얼 페이지를 참조하십시오.

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 검사 기반 실패 감지를 사용하는 인터페이스의 ICMP 요청에 대답할 수 있어야 합니다.

▼ 검사 기반 실패 감지의 대상 시스템을 수동으로 지정하는 방법

1 검사 기반 실패 감지를 구성 중인 시스템에 사용자 계정으로 로그인합니다.

2 검사 기반 실패 감지에서 대상으로 사용할 특정 호스트에 경로를 추가합니다.

```
$ route -p add -host destination-IP gateway-IP -static
```

여기서 *destination-IP* 및 *gateway-IP*는 대상으로 사용할 호스트의 IPv4 주소입니다. 예를 들어, IPMP 그룹 itops0의 인터페이스와 동일한 서브넷에 있는 대상 시스템 192.168.10.137을 지정하려면 다음을 입력합니다.

```
$ route -p add -host 192.168.10.137 192.168.10.137 -static
```

이 새로운 경로는 시스템이 다시 시작될 때마다 자동으로 구성됩니다. 검사 기반 실패 감지의 대상 시스템에 대해 임시 경로만 정의하려는 경우 -p 옵션을 사용하지 마십시오.

3 **네트워크에서 대상 시스템으로 사용할 추가 호스트에 경로를 추가합니다.**

▼ 사용할 실패 감지 방법을 선택하는 방법

기본적으로 검사 기반 실패 감지는 테스트 주소를 사용해야만 수행할 수 있습니다. NIC 드라이버가 지원하는 경우 링크 기반 실패 감지도 자동으로 사용으로 설정됩니다.

이 방법이 NIC 드라이버에서 지원되는 경우 링크 기반 실패 감지를 사용 안함으로 설정할 수 없습니다. 하지만 구현할 검사 기반 실패 감지 유형을 선택할 수 있습니다.

1 **전이적 검사만 사용하려면 다음 단계를 수행합니다.**

a. **적절한 SMF 명령을 사용하여 IPMP 등록 정보 transitive-probing을 설정합니다.**

```
# svccfg -s svc:/network/ipmp setprop config/transitive-probing=true
# svcadm refresh  svc:/network/ipmp:default
```

이 등록 정보 설정에 대한 자세한 내용은 in.mpathd(1M) 매뉴얼 페이지를 참조하십시오.

b. **IPMP 그룹에 대해 구성된 기존 테스트 주소를 모두 제거합니다.**

2 **테스트 주소만 사용하여 실패를 검사하려면 다음 단계를 수행합니다.**

a. **필요한 경우 전이적 검사를 해제합니다.**

```
# svccfg -s svc:/network/ipmp setprop config/transitive-probing=false
# svcadm refresh  svc:/network/ipmp:default
```

b. **IPMP 그룹의 기본 인터페이스에 테스트 주소를 할당합니다.**

▼ IPMP 데몬의 동작을 구성하는 방법

IPMP 구성 파일 /etc/default/mpathd를 사용하여 IPMP 그룹에 대해 다음과 같은 시스템 차원 매개변수를 구성합니다.

- FAILURE_DETECTION_TIME
- TRACK_INTERFACES_ONLY_WITH_GROUPS
- FAILBACK

1 **관리자로 전환합니다.**

자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 "관리 권한을 얻는 방법"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 검사에서 인터페이스 실패가 발생했는지 여부를 감지하는 데 걸리는 시간(초)입니다. 기본값은 10초입니다.

b. FAILBACK 매개변수의 새 값을 입력합니다.

```
FAILBACK=[yes | no]
```

- *yes* – *yes* 값은 IPMP 페일백 동작의 기본값입니다. 실패한 인터페이스의 복구가 감지되면 네트워크 액세스가 260 페이지 "물리적 인터페이스 복구 감지"에 설명된 대로 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. 검사 트래픽에는 계속 인터페이스를 사용할 수 있습니다.

  예를 들어, IPMP 그룹 `ipmp0`은 두 인터페이스 `net0`과 `net1`로 구성됩니다. `/etc/default/mpathd` 파일에서 *FAILBACK=no* 매개변수가 설정되었습니다. `net0`이 실패하면 `FAILED` 플래그가 지정되고 사용할 수 없게 됩니다. 복구 후에는 인터페이스에 `INACTIVE` 플래그가 지정되며 *FAILBACK=no* 설정 때문에 사용할 수 없는 상태로 유지됩니다.

  `net1`이 실패하고 `net0`만 `INACTIVE` 상태이면 `net0`의 `INACTIVE` 플래그가 지워지고 인터페이스를 사용할 수 있게 됩니다. IPMP 그룹에도 `INACTIVE` 상태의 다른 인터페이스가 있는 경우 `net1`이 실패하면 이러한 `INACTIVE` 인터페이스 중 하나(`net0`일 필요는 없음)가 지워지고 사용할 수 있게 됩니다.

c. TRACK_INTERFACES_ONLY_WITH_GROUPS 매개변수의 새 값을 입력합니다.

```
TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
```

주 – 이 매개변수와 익명 그룹 기능에 대한 자세한 내용은 259 페이지 "실패 감지 및 익명 그룹 기능"을 참조하십시오.

- *yes* – *yes* 값은 IPMP 동작의 기본값입니다. 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. 하지만 IPMP 그룹으로 구성되지 않은 인터페이스에서 실패 또는 복구가 감지될 경우 해당 인터페이스의 네트워킹 기능을 유지 관리하기 위해 IPMP에서 아무 작업도 트리거되지 않습니다. 따라서 *no* 값은 실패 보고에만 유용하며 네트워크 가용성을 직접 향상시키지는 않습니다.

3 **in.mpathd 데몬을 다시 시작합니다.**

```
# pkill -HUP in.mpathd
```

동적 재구성을 사용하여 IPMP 구성 복구

이 절에는 DR(동적 재구성)을 지원하는 시스템 관리와 관련된 절차가 포함되어 있습니다.

▼ 실패한 물리적 카드를 교체하는 방법

이 절차에서는 DR을 지원하는 시스템에서 물리적 카드를 교체하는 방법에 대해 설명합니다. 이 절차는 다음 조건을 가정합니다.

- 시스템의 활성 NCP는 `DefaultFixed`입니다. 시스템의 활성 NCP가 `DefaultFixed`가 아닌 경우 DR 사용에 대한 자세한 내용은 38 페이지 “NWAM이 다른 Oracle Solaris 네트워킹 기술과 함께 작동하는 방식”의 **동적 재구성 및 네트워크 구성 프로파일**을 참조하십시오.
- 시스템의 IP 인터페이스는 `net0`과 `net1`입니다.
- 두 인터페이스는 모두 IPMP 그룹 `itops0`에 속합니다.
- 기본 인터페이스 `net0`에 테스트 주소가 포함되어 있습니다.
- 기본 인터페이스 `net0`이 실패했으며 `net0`의 카드 `bge`를 제거해야 합니다.
- `bge` 카드를 `e1000g` 카드로 교체하고 있습니다.

시작하기 전에 DR 수행 절차는 시스템 유형에 따라 달라집니다. 따라서 다음을 완료해야 합니다.

- 시스템이 DR을 지원하는지 확인합니다.
- 시스템의 DR 절차에 대해 설명하는 해당 매뉴얼을 참조하십시오. Oracle Sun 하드웨어의 경우 DR을 지원하는 모든 시스템은 서버입니다. Sun 시스템에 대한 현재 DR 설명서를 찾으려면 `http://www.oracle.com/technetwork/indexes/documentation/index.html`에서 "dynamic reconfiguration"을 검색합니다.

주 – 다음 절차의 단계에서는 특히 IPMP 및 링크 이름 사용과 관련된 DR 측면만 참조합니다. DR을 수행하는 전체 단계는 이 절차에 포함되지 않습니다. 예를 들어, 구성이 자동화되지 않은 경우 ATM 및 기타 서비스와 같은 IP 계층 위의 일부 계층에 수동 구성 단계가 필요합니다. 시스템에 해당하는 DR 설명서를 따릅니다.

NIC를 교체하는 자세한 절차는 155 페이지 “동적 재구성을 사용하여 네트워크 인터페이스 카드를 교체하는 방법”을 참조하십시오.

---

1. **관리자로 전환합니다.**
   자세한 내용은 **Oracle Solaris 관리: 보안 서비스**의 "관리 권한을 얻는 방법"을 참조하십시오.

2. **적절한 DR 단계를 수행하여 시스템에서 실패한 NIC를 제거합니다.**
   예를 들어, `bge` 카드를 제거합니다.

3. **교체 NIC를 시스템에 연결합니다.**
   예를 들어, `bge` 카드가 차지했던 곳과 동일한 위치에 `e1000g` 카드를 설치합니다. `e1000g`의 데이터 링크에 `net0`이라는 이름이 지정되며 해당 데이터 링크의 구성이 상속됩니다.

4. **새 NIC의 리소스를 사용할 수 있게 하여 DR 프로세스를 완료합니다.**
   예를 들어, `cfgadm` 명령을 사용하여 이 단계를 수행합니다. 자세한 내용은 `cfgadm(1M)` 매뉴얼 페이지를 참조하십시오.

   이 단계 후에는 새 인터페이스가 `net0`의 지속 구성에 따라 테스트 주소로 구성되고, IPMP 그룹의 기본 인터페이스로 추가되며, 활성 또는 대기 인터페이스로 배포됩니다. 그런 다음 커널이 IPMP 인터페이스 `itops0`의 지속 구성에 따라 이 새로운 인터페이스에 데이터 주소를 할당할 수 있습니다.

IPMP 정보 모니터링

다음 절차에서는 `ipmpstat` 명령을 사용하여 시스템에서 IPMP 그룹의 여러 측면을 모니터할 수 있게 합니다. IPMP 그룹 전체나 해당 기본 IP 인터페이스의 상태를 관찰할 수 있습니다. 또한 그룹에 대한 데이터 및 테스트 주소의 구성을 확인할 수 있습니다. `ipmpstat` 명령을 사용하여 실패 감지에 대한 정보를 가져올 수도 있습니다. `ipmpstat` 명령 및 해당 옵션에 대한 자세한 내용은 `ipmpstat(1M)` 매뉴얼 페이지를 참조하십시오.

기본적으로 호스트 이름이 있을 경우 숫자 IP 주소 대신 호스트 이름이 출력 결과에 표시됩니다. 출력 결과에 숫자 IP 주소를 나열하려면 `-n` 옵션을 다른 옵션과 함께 사용하여 특정 IPMP 그룹 정보를 표시합니다.

주 – 달리 명시되지 않은 경우 다음 절차에서는 `ipmpstat` 명령을 사용하는 데 시스템 관리자 권한이 필요 없습니다.

▼ IPMP 그룹 정보를 가져오는 방법

이 절차를 사용하면 기본 인터페이스의 상태를 비롯하여 시스템에서 다양한 IPMP 그룹의 상태를 나열할 수 있습니다. 특정 그룹에 대해 검사 기반 실패 감지가 사용으로 설정된 경우 이 명령에 해당 그룹에 대한 실패 감지 시간도 포함됩니다.

- **IPMP 그룹 정보를 표시합니다.**

```
$ ipmpstat -g
GROUP    GROUPNAME   STATE      FDT        INTERFACES
itops0   itops0      ok         10.00s     net0 net1
acctg1   acctg1      failed     --         [net3 net4]
field2   field2      degraded   20.00s     net2 net5 (net7) [net6]
```

GROUP — IPMP 인터페이스 이름을 지정합니다. 익명 그룹의 경우 이 필드가 비어 있습니다. 익명 그룹에 대한 자세한 내용은 in.mpathd(1M) 매뉴얼 페이지를 참조하십시오.

GROUPNAME — IPMP 그룹의 이름을 지정합니다. 익명 그룹의 경우 이 필드가 비어 있습니다.

STATE — 그룹의 현재 상태를 나타내며 다음 중 하나일 수 있습니다.

- ok는 IPMP 그룹의 모든 기본 인터페이스를 사용할 수 있음을 나타냅니다.
- degraded는 그룹의 일부 기본 인터페이스를 사용할 수 없음을 나타냅니다.
- failed는 그룹의 모든 인터페이스를 사용할 수 없음을 나타냅니다.

FDT — 실패 감지가 사용으로 설정된 경우 실패 감지 시간을 지정합니다. 실패 감지가 사용 안함으로 설정된 경우 이 필드가 비어 있습니다.

INTERFACES — 그룹에 속하는 기본 인터페이스를 지정합니다. 이 필드에는 활성 인터페이스, 비활성 인터페이스 및 사용할 수 없는 인터페이스가 차례로 나열됩니다. 인터페이스 상태는 나열된 방식으로 표시됩니다.

- *interface*(괄호 또는 대괄호 없음)는 활성 인터페이스를 나타냅니다. 활성 인터페이스는 시스템에서 데이터 트래픽을 보내거나 받는 데 사용하는 인터페이스입니다.
- *(interface)*(괄호 있음)는 작동하지만 비활성 인터페이스를 나타냅니다. 인터페이스가 관리 정책에 정의된 대로 사용되고 있지 않습니다.
- *[interface]*(대괄호 있음)는 인터페이스가 실패했거나 오프라인 상태이므로 인터페이스를 사용할 수 없음을 나타냅니다.

▼ IPMP 데이터 주소 정보를 가져오는 방법

이 절차를 사용하면 데이터 주소 및 각 주소가 속하는 그룹을 표시할 수 있습니다. 표시되는 정보에는 ipadm [up-addr/down-addr] 명령으로 주소가 토글되었는지 여부에 따라 사용할 수 있는 주소도 포함됩니다. 주소를 사용할 수 있는 인바운드 또는 아웃바운드 인터페이스를 결정할 수도 있습니다.

- **IPMP 주소 정보를 표시합니다.**

```
$ ipmpstat -an
ADDRESS         STATE    GROUP      INBOUND     OUTBOUND
192.168.10.10   up       itops0     net0   net0 net1
192.168.10.15   up       itops0     net1   net0 net1
192.0.0.100     up       acctg1     --          --
192.0.0.101     up       acctg1     --          --
128.0.0.100     up       field2     net2       net2 net7
128.0.0.101     up       field2     net7       net2 net7
128.0.0.102     down     field2     --          --
```

ADDRESS -n 옵션을 -a 옵션과 함께 사용하는 경우 호스트 이름 또는 데이터 주소를 지정합니다.

STATE IPMP 인터페이스의 주소가 up(사용 가능) 또는 down(사용 불가능) 상태인지를 나타냅니다.

GROUP 특정 데이터 주소를 호스트하는 IPMP IP 인터페이스를 지정합니다.

INBOUND 지정된 주소에 대한 패킷을 받는 인터페이스를 식별합니다. 외부 이벤트에 따라 필드 정보가 변경될 수도 있습니다. 예를 들어, 데이터 주소가 작동 중지되었거나 IPMP 그룹에 활성 IP 인터페이스가 남아 있지 않은 경우 이 필드가 비어 있습니다. 빈 필드는 시스템이 지정된 주소로 전송된 IP 패킷을 허용하지 않음을 나타냅니다.

OUTBOUND 지정된 주소를 소스 주소로 사용하는 패킷을 보내는 인터페이스를 식별합니다. INBOUND 필드와 마찬가지로 OUTBOUND 필드 정보도 외부 이벤트에 따라 변경될 수 있습니다. 빈 필드는 시스템이 지정된 소스 주소로 패킷을 보내지 않음을 나타냅니다. 필드가 비어 있는 것은 주소가 작동 중지되었거나 그룹에 활성 IP 인터페이스가 남아 있지 않기 때문일 수 있습니다.

▼ 그룹의 기본 IP 인터페이스에 대한 정보를 가져오는 방법

이 절차를 사용하면 IPMP 그룹의 기본 IP 인터페이스에 대한 정보를 표시할 수 있습니다. NIC, 데이터 링크 및 IP 인터페이스 간의 해당 관계에 대한 설명은 20 페이지 “Oracle Solaris의 네트워크 스택”을 참조하십시오.

- **IPMP 인터페이스 정보를 표시합니다.**

```
$ ipmpstat -i
INTERFACE   ACTIVE   GROUP      FLAGS      LINK       PROBE      STATE
net0        yes      itops0     --mb---    up         ok         ok
net1        yes      itops0     -------    up         disabled   ok
net3        no       acctg1     -------    unknown    disabled   offline
net4        no       acctg1     is-----    down       unknown    failed
net2        yes      field2     --mb---    unknown    ok         ok
net6        no       field2     -i-----    up         ok         ok
```

```
net5        no      filed2     -------    up          failed     failed
net7        yes     field2     --mb---    up          ok         ok
```

INTERFACE 각 IPMP 그룹의 각 기본 인터페이스를 지정합니다.

ACTIVE 인터페이스가 작동하며 사용 여부(yes 또는 no)를 나타냅니다.

GROUP IPMP 인터페이스 이름을 지정합니다. 익명 그룹의 경우 이 필드가 비어 있습니다. 익명 그룹에 대한 자세한 내용은 in.mpathd(1M) 매뉴얼 페이지를 참조하십시오.

FLAGS 기본 인터페이스의 상태를 나타내며, 다음 중 하나 또는 임의 조합일 수 있습니다.

- i는 INACTIVE 플래그가 인터페이스에 설정되었으므로 데이터 트래픽을 보내거나 받는 데 해당 인터페이스가 사용되지 않음을 나타냅니다.
- s는 인터페이스가 대기 인터페이스로 구성되었음을 나타냅니다.
- m은 시스템에서 IPMP 그룹에 대한 IPv4 멀티캐스트 트래픽을 보내고 받는 데 해당 인터페이스를 지정했음을 나타냅니다.
- b는 시스템에서 IPMP 그룹에 대한 브로드캐스트 트래픽을 받는 데 해당 인터페이스를 지정했음을 나타냅니다.
- M은 시스템에서 IPMP 그룹에 대한 IPv6 멀티캐스트 트래픽을 보내고 받는 데 해당 인터페이스를 지정했음을 나타냅니다.
- d는 인터페이스가 작동 중지되었으므로 사용할 수 없음을 나타냅니다.
- h는 인터페이스가 다른 인터페이스와 중복 물리적 하드웨어 주소를 공유하며 오프라인 상태로 전환되었음을 나타냅니다. h 플래그는 인터페이스를 사용할 수 없음을 나타냅니다.

LINK 링크 기반 실패 감지의 상태를 나타내며 다음 상태 중 하나입니다.

- up 또는 down은 링크의 사용 가능 여부를 나타냅니다.
- unknown은 링크가 up 또는 down인지에 대한 알림을 드라이버가 지원하지 않으므로 링크 상태 변경을 감지하지 못함을 나타냅니다.

PROBE 테스트 주소로 구성된 인터페이스에 대한 검사 기반 실패 감지 상태를 다음과 같이 지정합니다.

- ok는 검사가 작동하며 활성 상태임을 나타냅니다.
- failed는 검사 기반 실패 감지에서 인터페이스가 작동하지 않는 것이 감지되었음을 나타냅니다.
- unknown은 적합한 검사 대상을 찾을 수 없으므로 검사를 보낼 수 없음을 나타냅니다.
- disabled는 인터페이스에 IPMP 테스트 주소가 구성되어 있지 않음을 나타냅니다. 따라서 검사 기반 실패 감지가 사용 안함으로 설정됩니다.

STATE 인터페이스의 전체 상태를 다음과 같이 지정합니다.

- ok는 인터페이스가 온라인 상태이며 실패 감지 방법의 구성에 따라 정상적으로 작동하고 있음을 나타냅니다.
- failed는 인터페이스의 링크가 작동 중지되었거나 검사 감지에서 인터페이스가 트래픽을 보내거나 받을 수 없음이 확인되어 인터페이스가 작동하지 않음을 나타냅니다.
- offline은 인터페이스를 사용할 수 없음을 나타냅니다. 일반적으로 인터페이스는 다음과 같은 상황에서 오프라인으로 전환됩니다.
  - 인터페이스를 테스트하고 있습니다.
  - 동적 재구성을 수행하고 있습니다.
  - 인터페이스가 다른 인터페이스와 중복 하드웨어 주소를 공유합니다.
- unknown은 검사 기반 실패 감지에 대해 검사 대상을 찾을 수 없어서 IPMP 인터페이스의 상태를 확인할 수 없음을 나타냅니다.

▼ IPMP 검사 대상 정보를 가져오는 방법

이 절차를 사용하면 IPMP 그룹의 각 IP 인터페이스와 연결된 검사 대상을 모니터할 수 있습니다.

- **IPMP 검사 대상을 표시합니다.**

```
$ ipmpstat -nt
INTERFACE   MODE          TESTADDR         TARGETS
net0        routes        192.168.85.30    192.168.85.1 192.168.85.3
net1        disabled      --               --
net3        disabled      --               --
net4        routes        192.1.2.200      192.1.2.1
net2        multicast     128.9.0.200      128.0.0.1 128.0.0.2
net6        multicast     128.9.0.201      128.0.0.2 128.0.0.1
net5        multicast     128.9.0.202      128.0.0.1 128.0.0.2
net7        multicast     128.9.0.203      128.0.0.1 128.0.0.2

$ ipmpstat -nt
INTERFACE   MODE         TESTADDR         TARGETS
net3        transitive   <net1>           <net1> <net2> <net3>
net2        transitive   <net1>           <net1> <net2> <net3>
net1        routes       172.16.30.100    172.16.30.1
```

INTERFACE IPMP 그룹의 기본 인터페이스를 지정합니다.

MODE 검사 대상을 가져오는 방법을 지정합니다.

- routes는 시스템 경로 지정 테이블이 검사 대상을 찾는 데 사용됨을 나타냅니다.
- mcast는 멀티캐스트 ICMP 검사가 대상을 찾는 데 사용됨을 나타냅니다.

- `disabled`는 인터페이스에 대해 검사 기반 실패 감지가 사용 안함으로 설정되었음을 나타냅니다.
- `transitive`는 두번째 예와 같이 전이적 검사가 실패 감지에 사용됨을 나타냅니다. 전이적 검사와 테스트 주소를 동시에 사용하여 검사 기반 실패 감지를 구현할 수 없습니다. 테스트 주소를 사용하지 않으려는 경우 전이적 검사로 전환해야 합니다. 전이적 검사를 사용하지 않으려는 경우 테스트 주소를 구성해야 합니다. 개요는 257 페이지 "검사 기반 실패 감지"를 참조하십시오.

`TESTADDR` `-n` 옵션을 `-t` 옵션과 함께 사용하는 경우 검사를 보내고 받기 위해 인터페이스에 할당되는 IP 주소 또는 호스트 이름을 지정합니다.

전이적 검사를 사용하는 경우 인터페이스 이름은 데이터를 받는 데 사용되지 않는 기본 IP 인터페이스를 나타냅니다. 또한 이 이름은 지정된 인터페이스의 소스 주소로 전이적 테스트 검사가 전송되고 있음을 나타냅니다. 데이터를 받는 활성 기본 IP 인터페이스의 경우 표시되는 IP 주소는 송신 ICMP 검사의 소스 주소를 나타냅니다.

주 – IP 인터페이스가 IPv4 및 IPv6 테스트 주소로 구성된 경우 검사 대상 정보가 각 테스트 주소에 대해 별도로 표시됩니다.

`TARGETS` 현재 검사 대상을 공백으로 구분된 목록으로 나열합니다. `-n`을 `-t` 옵션과 함께 사용하는 경우 검사 대상이 호스트 이름 또는 IP 주소로 표시됩니다.

▼ IPMP 검사를 관찰하는 방법

이 절차를 사용하면 진행 중인 검사를 관찰할 수 있습니다. 검사를 관찰하는 명령을 실행하면 `Ctrl-C`로 명령을 종료할 때까지 시스템의 검사 작업에 대한 정보가 계속 표시됩니다. 이 명령을 실행하려면 기본 관리자 권한이 있어야 합니다.

1 관리자로 전환합니다.

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

2 진행 중인 검사에 대한 정보를 표시합니다.

```
# ipmpstat -pn
TIME    INTERFACE   PROBE     NETRTT   RTT      RTTAVG     TARGET
0.11s   net0        589       0.51ms   0.76ms   0.76ms     192.168.85.1
0.17s   net4        612       --       --       --         192.1.2.1
0.25s   net2        602       0.61ms   1.10ms   1.10ms     128.0.0.1
0.26s   net6        602       --       --       --         128.0.0.2
0.25s   net5        601       0.62ms   1.20ms   1.00ms     128.0.0.1
0.26s   net7        603       0.79ms   1.11ms   1.10ms     128.0.0.1
```

```
1.66s   net4        613       --        --        --         192.1.2.1
1.70s   net0        603       0.63ms    1.10ms    1.10ms     192.168.85.3
^C

# ipmpstat -pn
TIME    INTERFACE   PROBE     NETRTT    RTT       RTTAVG     TARGET
1.39S   net4        t28       1.05ms    1.06ms    1.15ms     <net1>
1.39s   net1        i29       1.00ms    1.42ms    1.48ms     172.16.30.1
```

TIME — ipmpstat 명령이 실행된 시간을 기준으로 검사가 전송된 시간을 지정합니다. ipmpstat를 시작하기 전에 검사를 시작한 경우 명령이 실행된 시간을 기준으로 시간이 음수 값으로 표시됩니다.

INTERFACE — 검사가 전송되는 인터페이스를 지정합니다.

PROBE — 검사를 나타내는 식별자를 지정합니다. 전이적 검사가 실패 감지에 사용되는 경우 식별자 앞에 t(전이적 검사) 또는 i(ICMP 검사)가 추가됩니다.

NETRTT — 검사의 총 네트워크 라운드 트립 시간을 지정하며 밀리초 단위로 측정됩니다. NETRTT는 IP 모듈이 검사를 보내는 순간과 IP 모듈이 대상으로부터 ack 패킷을 받는 순간 사이의 시간을 나타냅니다. in.mpathd 데몬이 검사가 손실되었음을 확인하면 필드가 비워집니다.

RTT — 검사의 총 라운드 트립 시간을 지정하며 밀리초 단위로 측정됩니다. RTT는 데몬이 검사를 보내는 코드를 실행하는 순간과 데몬이 대상의 ack 패킷 처리를 완료하는 순간 사이의 시간을 나타냅니다. in.mpathd 데몬이 검사가 손실되었음을 확인하면 필드가 비워집니다. NETRTT에 없는 RTT에서 발생하는 스파이크는 로컬 시스템이 과부하되었음을 나타낼 수 있습니다.

RTTAVG — 로컬 시스템과 대상 간의 인터페이스에서 검사의 평균 라운드 트립 시간을 지정합니다. 평균 라운드 트립 시간은 느린 대상을 식별하는 데 도움이 됩니다. 데이터가 부족하여 평균을 계산할 수 없는 경우 이 필드가 비워집니다.

TARGET — -n 옵션을 -p 옵션과 함께 사용하는 경우 검사가 전송되는 대상 주소 또는 호스트 이름을 지정합니다.

▼ 스크립트에서 ipmpstat 명령의 출력 결과를 사용자 정의하는 방법

ipmpstat를 사용하는 경우 기본적으로 80개 열에 들어가는 가장 의미 있는 필드가 표시됩니다. ipmpstat -p 구문의 경우를 제외하고 ipmpstat 명령과 함께 사용하는 옵션과 관련된 모든 필드가 출력 결과에 표시됩니다. 표시할 필드를 지정하려는 경우 명령의 출력 모드를 결정하는 다른 옵션과 함께 -o 옵션을 사용합니다. 이 옵션은 스크립트에서 또는 명령 별칭을 사용하여 명령을 실행할 때 특히 유용합니다.

- **출력 결과를 사용자 정의하려면 다음 명령 중 하나를 실행합니다.**

  - ipmpstat 명령의 선택한 필드를 표시하려면 특정 출력 옵션과 함께 -o 옵션을 사용합니다. 예를 들어, 그룹 출력 모드의 GROUPNAME 및 STATE 필드만 표시하려면 다음을 입력합니다.

    ```
    $ ipmpstat -g -o groupname,state

    GROUPNAME  STATE
    itops0     ok
    accgt1     failed
    field2     degraded
    ```

  - 지정된 ipmpstat 명령의 모든 필드를 표시하려면 다음 구문을 사용합니다.

    ```
    # ipmpstat -o all
    ```

▼ ipmpstat 명령의 시스템 구문 분석 가능 출력 결과를 생성하는 방법

ipmpstat -P 구문을 사용하여 시스템 구문 분석 가능 정보를 생성할 수 있습니다. -P 옵션은 특히 스크립트에서 사용하기 위한 것입니다. 시스템 구문 분석 가능 출력 결과와 일반적인 출력 결과의 차이점은 다음과 같습니다.

- 헤더가 생략됩니다.
- 필드가 콜론(:)으로 구분됩니다.
- 빈 값이 포함된 필드가 이중 대시(--)로 채워지는 대신 비어 있습니다.
- 여러 필드가 요청되는 경우 필드에 리터럴 콜론(:) 또는 백슬래시(\)가 포함되어 있으면 이러한 문자 앞에 백슬래시(\)를 추가하여 이스케이프하거나 제외할 수 있습니다.

ipmpstat -P 구문을 올바르게 사용하려면 다음 규칙을 관찰합니다.

- -o *option fields*를 -P 옵션과 함께 사용합니다.
- -o all을 - P 옵션과 함께 사용하지 않습니다.

이러한 규칙을 하나라도 무시하면 ipmpstat -P가 실패합니다.

- **그룹 이름, 실패 감지 시간 및 기본 인터페이스를 시스템 구문 분석 가능 형식으로 표시하려면 다음을 입력합니다.**

```
$ ipmpstat -P -o -g groupname,fdt,interfaces
itops0:10.00s:net0 net1
acctg1::[net3 net4]
field2:20.00s:net2 net7 (net5) [net6]
```

그룹 이름, 실패 감지 시간 및 기본 인터페이스는 그룹 정보 필드입니다. 따라서 -o -g 옵션을 -P 옵션과 함께 사용합니다.

예 15–8 스크립트에서 ipmpstat -P 사용

이 샘플 스크립트는 특정 IPMP 그룹의 실패 감지 시간을 표시합니다.

```
getfdt() {
         ipmpstat -gP -o group,fdt | while IFS=: read group fdt; do
             [[ "$group" = "$1" ]] && { echo "$fdt"; return; }
         done
     }
```

◆ ◆ ◆ 16장

16 LLDP를 사용하여 네트워크 연결 정보 교환

이 장에서는 LLDP(Link Layer Discovery Protocol)를 사용하여 로컬 네트워크 전체에서 시스템이 시스템 및 네트워크 연결 정보를 교환할 수 있게 하는 방법에 대해 설명합니다.

Oracle Solaris의 LLDP 개요

LLDP는 토폴로지 검색을 위해 로컬 네트워크 전체에 정보를 알리는 데 사용됩니다. 이 프로토콜을 사용할 경우 시스템이 네트워크의 다른 시스템에 연결 및 관리 정보를 알릴 수 있습니다. 이 정보에는 시스템 기능, 관리 주소 및 기타 관련 정보가 포함될 수 있습니다. 또한 이 프로토콜을 사용하면 동일한 시스템이 동일한 로컬 네트워크에 있는 다른 시스템에 대한 유사한 정보를 받을 수 있습니다.

Oracle Solaris에서는 LLDP 지원에 PFC(우선 순위 기반 흐름 제어) 및 응용 프로그램 TLV와 같은 DCB 기능에 대한 구성 정보를 교환하기 위한 DCB(Data Center Bridging)도 포함됩니다.

LLDP를 사용할 경우 시스템 관리자가 특히 VLAN(가상 LAN), 링크 통합 및 기타 링크 유형을 포함하는 복잡한 네트워크에서 결함이 있는 시스템 구성을 쉽게 감지할 수 있습니다.

LLDP 구현의 구성 요소

LLDP는 다음 구성 요소로 구현됩니다.

- LLDP 기능을 사용으로 설정하려면 LLDP 패키지를 설치해야 합니다. 이 패키지는 LLDP 데몬, 명령줄 유틸리티, 서비스 매니페스트 및 스크립트와 LLDP 작동에 필요한 기타 구성 요소를 제공합니다.
- `lldp` 서비스는 `svcadm` 명령에 의해 사용으로 설정됩니다. 이 서비스는 LLDP 데몬을 관리하며 데몬 시작, 중지, 다시 시작 또는 새로 고침을 담당합니다. 이 서비스는 기본적으로 사용 안함으로 설정됩니다. 따라서 LLDP를 사용하려면 먼저 시스템에

대해 전역적으로 서비스를 사용으로 설정해야 합니다. `lldp` 서비스가 사용으로 설정되고 데몬이 시작된 후 시스템 관리자의 결정에 따라 개별 링크에서 LLDP 기능을 사용으로 설정할 수 있습니다.

- `lldpadm` 명령은 개별 링크에서 LLDP를 관리하며, LLDP의 작동 모드를 구성하고 전송될 TLV(Time-Length-Value) 단위를 지정하고 DCB 응용 프로그램 정보를 구성하는 데 사용됩니다. 특히, 이 명령은 에이전트별 LLDP 등록 정보와 전역 LLDP 등록 정보를 설정하는 데 사용됩니다. `lldpadm` 명령의 일반 하위 명령은 `dladm` 및 `ipadm` 명령의 일반 하위 명령과 유사합니다.
  - `lldpadm set-*`는 수행할 작업을 지정하며, 이 작업에서 지정한 LLDP 등록 정보에 대해 하나 이상의 값이 설정됩니다.
  - `lldpadm show-*`는 지정한 LLDP 등록 정보에 대해 설정되는 값을 표시합니다.
  - `lldpadm reset-*`는 지정한 LLDP 등록 정보의 구성을 기본값으로 되돌립니다.

  이러한 하위 명령의 사용은 이후 절에서 설명합니다. `lldpadm` 명령에 대한 자세한 내용은 `lldpadm(1M)` 매뉴얼 페이지를 참조하십시오.
- LLDP 라이브러리(`liblldp.so`)는 링크의 LLDP 정보를 검색하고 LLDP 패킷을 구문 분석하고 기타 기능을 수행하는 데 사용할 수 있는 API를 제공합니다.
- LLDP 에이전트는 LLDP가 사용으로 설정된 물리적 NIC와 연결된 LLDP 인스턴스입니다. LLDP 에이전트는 연결된 NIC의 LLDP 동작을 제어합니다. 물리적 NIC에서만 LLDP 에이전트를 구성할 수 있습니다.
- LLDP 데몬(`lldpd`)은 시스템에서 LLDP 에이전트의 관리자 역할을 수행합니다. 또한 SNMP(Simple Network Management Protocol)를 통해 시스템에 수신된 LLDP 정보를 검색하는 SNMP의 데몬인 `snmpd`와 상호 작용합니다. 또한 이 데몬은 `sysevents` 정보를 게시하고 LLDP 라이브러리의 질의에 응답합니다.

다음 절에서는 LLDP 에이전트에 대해 자세히 설명합니다.

LLDP 에이전트의 기능

LLDP 에이전트는 *PDU(Protocol Data Unit)*라고도 하는 LLDP 패킷을 전송하고 수신합니다. 이 에이전트는 해당 패킷에 포함된 정보를 관리하고 두 가지 유형의 데이터 저장소에 저장합니다.

- 로컬 관리 정보 데이터베이스, 즉 로컬 MIB. 이 데이터 저장소는 LLDP 에이전트가 사용으로 설정된 특정 링크와 관련된 네트워크 정보를 포함합니다. 로컬 MIB에는 공통 정보와 고유 정보가 모두 포함됩니다. 예를 들어, 섀시 ID는 시스템의 모든 LLDP 에이전트에서 공유되는 공통 정보입니다. 하지만 포트 번호는 시스템의 데이터 링크마다 다릅니다. 따라서 각 에이전트가 해당 로컬 MIB를 관리합니다.
- 원격 MIB. 이 데이터 저장소의 정보는 로컬 네트워크의 다른 시스템과 관련이 있습니다.

LLDP 에이전트의 작동 방식 구성

LLDP 에이전트가 다음 모드로 작동하도록 구성할 수 있습니다.

- 전송 전용(txonly) 모드에서는 에이전트가 수신 LLDP 패킷을 처리하지 않습니다. 따라서 원격 MIB가 비어 있습니다.
- 수신 전용(rxonly) 모드에서는 에이전트가 수신 LLDP 패킷만 처리하고 정보를 원격 MIB에 저장합니다. 하지만 로컬 MIB의 정보는 전송되지 않습니다.
- 전송 및 수신(both) 모드에서는 에이전트가 LLDP 패킷을 전송하고 수신합니다. 두 유형의 MIB가 모두 사용됩니다. 또한 이 모드에서는 기본 링크가 지원하는 DCB 기능이 사용으로 자동 설정됩니다.
- 사용 안함(disable) 모드에서는 에이전트가 없습니다.

▼ LLDP를 사용으로 설정하는 방법

이 절차에서는 처음으로 시스템에서 LLDP를 사용으로 설정합니다.

1 LLDP 패키지를 설치합니다.

```
# pkg install lldp
```

주 – Oracle Solaris 패키지 및 설치 방법에 대한 개요는 **Oracle Solaris 관리: 일반 작업**의 12장, “소프트웨어 패키지 관리(작업)”를 참조하십시오.

---

2 시스템에서 LLDP 서비스를 시작합니다.

```
# svcadm enable svc:/network/lldp:default
```

3 LLDP를 사용으로 설정할 데이터 링크를 식별합니다.

4 해당 데이터 링크의 LLDP 에이전트에 대해 작동 모드를 설정합니다.

```
# lldpadm set-agentprop -p mode=value agent
```

여기서 *value*는 작동 모드 중 하나일 수 있으며 *agent*는 LLDP가 사용으로 설정된 데이터 링크의 이름을 사용합니다.

주 – 명령을 사용하기 쉽도록 `lldpadm` 명령의 하위 명령을 약어 형태로 입력할 수 있습니다. 예를 들어, `lldpadm set-agentprop` 대신 `lldpadm set-ap`를 입력할 수 있습니다. 하위 명령 및 약어 형태는 `lldpadm`(1M) 매뉴얼 페이지를 참조하십시오.

---

5 LLDP 에이전트의 작동 모드를 확인하려면 다음 명령을 입력합니다.

```
# lldpadm show-agentprop -p mode agent
```

6 **LLDP 에이전트를 사용 안함으로 설정하려면 다음 명령 중 하나를 사용합니다.**

- **lldpadm set-agentprop -p mode=disable** *agent*
- **lldpadm reset-agentprop -p mode** *agent*

7 **전체 시스템에서 LLDP를 해제하려면 다음을 입력합니다.**

```
# svcadm disable svc:/network/lldp:default
```

예 16–1 여러 데이터 링크에서 LLDP 사용

이 예에서는 시스템에 `net0`과 `net1`이라는 데이터 링크 두 개가 있으며 각 LLDP 에이전트에 대해 LLDP가 서로 다른 모드에서 사용으로 설정되었습니다. 한 에이전트는 LLDP 패킷을 전송 및 수신하여 작동하고, 다른 에이전트는 LLDP 패킷을 전송만 합니다.

```
# svcadm enable svc:/network/lldp:default
# lldpadm set-agentprop -p mode=both net0
# lldpadm set-agentprop -p mode=txonly net1
```

알릴 정보 구성

LLDP 에이전트는 시스템 및 연결 정보를 LLDP 패킷이나 LLDPDU로 전송합니다. 이러한 패킷에는 TLV(Type-Length-Value) 형식이 개별적으로 지정된 정보 단위가 포함됩니다. 따라서 정보 단위를 TLV 단위라고도 합니다. 특정 TLV 단위는 필수이며 LLDP를 사용으로 설정한 경우 기본적으로 LLDP 패킷에 포함됩니다. 필수 TLV 단위는 다음과 같습니다.

- 섀시 ID
- 포트 ID
- TTL(활성 시간)
- PDU의 끝

섀시 ID는 `hostid` 명령에서 생성되는 정보이고 포트 ID는 물리적 NIC의 MAC 주소입니다. 링크 수에 따라 단일 시스템에서 여러 LLDP 에이전트를 사용으로 설정할 수 있습니다. 결합된 섀시 ID와 포트 ID는 에이전트를 고유하게 식별하며 시스템의 다른 에이전트와 구분합니다.

`lldpadm` 명령을 사용하여 LLDP 패킷에서 필수 TLV 단위를 제외할 수 없습니다.

선택적 TLV 단위를 LLDP 패킷에 추가할 수 있습니다. 이러한 선택적 TLV 단위는 공급업체가 알릴 공급업체 관련 TLV 단위를 삽입할 수 있는 수단입니다. TLV 단위는 개별 OUI(Organization Unique Identifier)로 식별되며 이러한 OUI가 IEEE 802.1 사양인지 또는 IEEE 802.3 사양인지에 따라 입력됩니다. 각 TLV 유형에 해당하는 LLDP 에이전트 등록 정보는 각 유형의 값을 설정할 수 있도록 생성됩니다.

다음 표에서는 TLV 유형이나 그룹, 해당 등록 정보 이름, 각 등록 정보에 대한 TLV 단위 및 해당 설명을 보여줍니다.

표 16–1 LLDP 에이전트에 대해 사용으로 설정할 수 있는 TLV 단위

TLV 유형	등록 정보 이름	TLV	설명
기본 관리	basic-tlv	sysname, portdesc, syscapab, sysdesc, mgmtaddr	알릴 시스템 이름, 포트 설명, 시스템 기능, 시스템 설명 및 관리 주소를 지정합니다.
802.1 OUI	dot1-tlv	vlanname, pvid, linkaggr, pfc, appln	알릴 VLAN 이름, 포트 VLAN ID, 링크 통합, 포트 설명 및 응용 프로그램 TLV를 지정합니다.
802.3 OUI	dot3-tlv	max-framesize	알릴 최대 프레임 크기를 지정합니다.
Oracle 관련 OUI(0x0003BA로 정의됨)	virt-tlv	vnic	가상 네트워크가 구성된 경우 알릴 VNIC를 지정합니다.

이러한 등록 정보 중 하나를 구성하여 LLDP를 사용으로 설정한 경우 패킷에 포함될 TLV 단위를 지정합니다.

▼ LLDP 패킷에 대한 TLV 단위를 지정하는 방법

이 절차에서는 LLDP 패킷에 알릴 TLV 단위를 추가하는 방법을 보여줍니다. LLDP 패킷에 대해 TLV 단위를 설정하려면 `lldpadm set-agentprop` 하위 명령을 사용합니다.

1 필요한 경우 추가할 TLV 단위를 포함할 수 있는 LLDP 에이전트 등록 정보를 식별합니다.

이 하위 명령은 각 등록 정보에 대해 이미 설정된 TLV 단위도 표시합니다.

```
# lldpadm show-agentprop agent
```

이 등록 정보를 지정하지 않을 경우 이 하위 명령은 모든 LLDP 에이전트 등록 정보와 해당 TLV 값을 표시합니다.

2 등록 정보에 TLV 단위를 추가합니다.

```
# lldpadm set-agentprop -p property[+|-]=value[,...] agent
```

+|- 수식자는 여러 값을 허용하는 등록 정보에 사용됩니다. 이러한 수식자를 사용하여 목록에서 값을 추가(+)하거나 제거(-)할 수 있습니다. 수식자를 사용하지 않으면 설정한 값이 이전에 등록 정보에 대해 정의된 모든 값을 대체합니다.

3 (옵션) 등록 정보의 새 값을 표시합니다.

```
# lldpadm show-agentprop -p property agent
```

예 16–2 LLDP 패킷에 선택적 TLV 단위 추가

이 예에서는 LLDP 에이전트 net0이 패킷의 VLAN 정보를 알리도록 이미 구성되었습니다. 이 알림에 시스템 기능, 링크 통합 및 네트워크 가상화 정보도 포함하려고 합니다. 하지만 패킷에서 VLAN 설명은 제거하려고 합니다.

```
# lldpadm show-agentprop net0
# lldpadm set-agentprop -p dot1-tlv+=linkaggr net0
AGENT   PROPERTY   PERM   VALUE          DEFAULT   POSSIBLE
net0    mode       rw     both           disable   txonly,rxonly,both,
                                                   disable
net0    basic-tlv  rw     sysname,        none     none,portdesc,
                          sysdesc                  sysname,sysdesc,
                                                   syscapab,mgmtaddr,
                                                   all
net0    dot1-tlv   rw     vlanname,       none     none,vlanname,pvid,
                          pvid,pfc                 linkaggr,pfc,appln,
                                                   all
net0    dot3-tlv   rw     max-framesize   none     none, max-framesize,
                                                   all
net0    virt-tlv   rw     none           none     none,vnic,all


# lldpadm set-agentprop -p basic-tlv+=syscapab,dot1-tlv+=linkaggr,virt-tlv=vnic net0
# lldpadm set-agentprop -p dot1-tlv-=pfc net0
# lldpadm show-agentprop -p net0
AGENT   PROPERTY   PERM   VALUE          DEFAULT   POSSIBLE
net0    mode       rw     both           disable   txonly,rxonly,both,
                                                   disable
net0    basic-tlv  rw     sysname,        none     none,portdesc,
                          sysdesc,                 sysname,sysdesc,
                          syscapab                 syscapab,mgmtaddr,
                                                   all
net0    dot1-tlv   rw     vlanname,       none     none,vlanname,pvid,
                          pvid,                    linkaggr,pfc,appln,
                          linkaggr                 all
net0    dot3-tlv   rw     max-framesize   none     none, max-framesize,
                                                   all
net0    virt-tlv   rw     vnic           none     none,vnic,all
```

TLV 단위 관리

각 TLV 단위에는 특정 값으로 구성할 수 있는 등록 정보가 있습니다. TLV 단위가 LLDP 에이전트의 등록 정보로 설정된 경우 TLV 단위는 네트워크에서 지정한 값으로만 알려집니다. 예를 들어, 시스템의 기능을 알리는 TLV 값 syscapab를 고려해 보십시오. 이러한 기능은 라우터, 브릿지, 반복기, 전화 및 기타 장치에 대한 지원을 포함할 수 있습니다. 하지만 라우터 및 브릿지와 같이 실제로 특정 시스템에서 지원되는 기능만 알려지도록 syscapab를 설정할 수 있습니다.

TLV 관리 절차는 전역 TLV 또는 에이전트별 TLV를 구성하는지에 따라 달라집니다.

전역 *TLV*는 시스템의 모든 LLDP 에이전트에 적용됩니다. 다음 표에서는 전역 TLV 값과 가능한 해당 구성을 표시합니다.

표 16–2 전역 TLV 및 해당 등록 정보

TLV 이름	TLV 등록 정보 이름	가능한 등록 정보 값	값 설명
syscapab	supported	other, repeater, bridge, wlan-ap, router, telephone, docsis-cd, station, cvlan, sylvan, tpmr	시스템에서 지원되는 주요 기능을 나타냅니다. 기본값은 router, station 및 bridge입니다.
	enabled	supported에 대해 나열되는 값의 일부입니다.	시스템에서 사용으로 설정된 기능을 나타냅니다.
mgmtaddr	ipaddr	ipv4 또는 ipv6	로컬 LLDP 에이전트와 연결될 IP 주소의 유형을 지정합니다. 이 주소는 상위 계층 엔티티에 도달하는 데 사용되며 네트워크 관리에 의한 검색을 지원합니다. 한 개의 유형만 지정할 수 있습니다.

전역 값을 가질 수 없는 TLV 단위는 LLDP 에이전트 레벨에서 관리됩니다. **에이전트별 *TLV* 단위**를 사용할 경우 제공한 값은 특정 LLDP 에이전트에서 TLV 단위를 전송할 수 있을 때 사용됩니다.

다음 표에서는 LLDP 에이전트에 대한 TLV 값과 가능한 해당 구성을 표시합니다.

표 16–3 에이전트별 TLV 단위 및 해당 등록 정보

TLV 이름	TLV 등록 정보 이름	가능한 등록 정보 값	값 설명
pfc	willing	on, off	원격 시스템의 구성 정보를 허용하거나 거부하도록 LLDP 에이전트를 설정합니다.
appln	apt	응용 프로그램 우선 순위 테이블에 정의된 정보에서 값을 가져옵니다.	응용 프로그램 우선 순위 테이블을 구성합니다. 이 표에는 응용 프로그램의 TLV 단위 및 해당 우선 순위 목록이 포함되어 있습니다. 응용 프로그램은 id/selector 쌍으로 식별됩니다. 테이블 내용은 다음 형식을 사용합니다. id/selector/priority

다음 절차에서는 전역 TLV 값을 정의하는 방법을 보여줍니다. 에이전트별 TLV 단위를 정의하는 방법에 대한 자세한 내용은 306 페이지 “DCB(Data Center Bridging)”를 참조하십시오.

▼ 전역 TLV 값을 정의하는 방법

이 절차에서는 특정 TLV 단위에 전역 값을 제공하는 방법을 보여줍니다. 전역 TLV 값을 설정하려면 `lldpadm set-tlvprop` 하위 명령을 사용합니다.

1 알리려는 값을 포함하도록 적절한 TLV 등록 정보를 구성합니다.

참조는 표 16–2를 참조하십시오.

```
# lldpadm set-tlvprop -p tlv-property=value[,value,value,...] tlv
```

2 (옵션) 방금 구성한 등록 정보의 값을 표시합니다.

```
# lldpadm show-tlvprop
```

예 16–3 시스템의 기능 및 관리 IP 주소 지정

이 예에서는 다음 두 가지 목적을 달성합니다.

- LLDP 패킷에 알릴 시스템 기능에 대한 특정 정보를 제공합니다. 이를 위해서는 `syscapab` TLV 단위의 `supported` 및 `enabled` 등록 정보를 모두 구성해야 합니다.
- 알림에 사용되는 관리 IP 주소를 제공합니다.

```
# lldpadm set-tlvprop -p supported=bridge,router,repeater syscapab
# lldpadm set-tlvprop -p enabled=router syscapab
# lldpadm set-tlvprop -p ipaddr=192.168.1.2 mgmtaddr
# lldpadm show-tlvprop
TLVNAME    PROPERTY    PERM  VALUE        DEFAULT         POSSIBLE
syscapab   supported   rw    bridge,      bridge,router,  other,router,
                             router,      station         repeater,bridge,
                             repeater                     wlan-ap,telephone,
                                                          docis-cd,station,
                                                          cvlan,svlan,tpmr
syscapab   enabled     rw    router       none            bridge,router,
                                                          repeater
mgmtaddr   ipaddr      rw    192.162.1.2  none            --
```

DCB(Data Center Bridging)

FCoE(Fibre Channel over Ethernet) 트래픽을 지원하기 위해 Oracle Solaris의 LLDP 구현에는 DCB(Data Center Bridging) 지원이 포함됩니다.

트래픽 교환에 일반 이더넷을 사용하는 네트워크에서는 네트워크 사용량이 많을 때 패킷이 삭제될 수 있는 위험이 있습니다. FCoE 트래픽의 주요 요구 사항은 전송 도중 패킷이 삭제될 수 없도록 하는 것입니다. DCBx(Data Center Bridging Exchange), PFC(우선 순위 기반 흐름 제어) TLV 및 응용 프로그램 TLV를 지원하면 패킷 삭제가 방지됩니다.

PFC는 패킷에 대한 우선 순위 정보를 포함하도록 표준 PAUSE 프레임을 확장합니다. 일반적으로 PAUSE 프레임은 트래픽이 많을 때 링크에서 전송되어 수신 끝에서 이미 수신된 패킷을 처리할 수 있게 합니다. PFC를 사용하면 PAUSE 프레임을 전송하여

링크의 모든 트래픽을 중지하는 대신 트래픽이 패킷에 대해 정의된 우선 순위에 따라 일시 중지됩니다. 트래픽을 일시 중지해야 하는 우선 순위에 대해 PFC 프레임을 보낼 수 있습니다. 보낸 사람은 해당 특정 우선 순위의 트래픽을 중지하지만 다른 우선 순위의 트래픽은 영향을 받지 않습니다. 지정한 시간 후에 다른 PFC 프레임이 전송되어 일시 중지된 트래픽을 계속할 수 있다고 알립니다.

PFC 구성 정보는 DCBx를 통해 피어 스테이션 간에 교환됩니다. 트래픽 교환의 피어에 일치하는 PFC 구성이 있을 경우 PFC는 필요에 따라 트래픽 전송을 일시 중지하거나 계속할 수 있습니다. 각 우선 순위에 다른 패킷을 할당할 수 있도록 응용 프로그램 TLV를 사용하여 우선 순위 정보를 정의합니다. 피어에 일치하지 않는 PFC 구성이 있을 경우 다음에 나오는 절차에 따라 다른 피어의 구성을 허용하도록 PFC TLV를 사용자 정의할 수 있습니다.

DCB(Data Center Bridging)는 304 페이지 “TLV 단위 관리”에 설명된 대로 에이전트별 TLV 단위를 구성하는 방법을 보여주는 특정 사례입니다.

▼ 에이전트별 TLV 값을 설정하는 방법

이 절차에서는 `llpadm set-agenttlvprop` 하위 명령을 사용하여 LLDP 에이전트 레벨에서 TLV 값을 설정하는 방법을 보여줍니다.

1 지정된 LLDP 에이전트에서 알리려는 값을 포함하도록 적절한 TLV 등록 정보를 구성합니다.

참조는 표 16–3을 참조하십시오.

```
# lldpadm set-agenttlvprop -p tlv-property[+|-]=value[,value,value,...] -a agent tlv-name
```

2 (옵션) 방금 구성한 등록 정보의 값을 표시합니다.

```
# lldpadm show-agenttlvprop
```

예 16–4 LLDP 에이전트가 정보를 허용하도록 설정 및 TLV 응용 프로그램 우선 순위 지정

이 예에서는 `pfc` 및 `appln` TLV 값이 사용자 정의되는 방식을 보여줍니다. 이 예의 TLV 단위는 DCB가 FCoE 트래픽에 대해 작동하는 방식을 지정합니다. 로컬 구성이 피어 구성과 일치하지 않는 경우 시스템이 피어의 PFC 구성을 허용하도록 구성되었습니다. 또한 이 예에서는 LLDP 에이전트의 응용 프로그램 TLV에 대해 우선 순위가 설정된 방식을 보여줍니다.

```
# lldpadm set-agenttlvprop -p willing=on -a net0 pfc
# lldpadm set-agenttlvprop -p apt=8906/1/4 -a net0 appln
# lldpadm show-agenttlvprop
AGENT   TLVNAME   PROPERTY   PERM   VALUE      DEFAULT   POSSIBLE
net0    pfc       willing    rw     on         off       on,off
net0    appln     apt        rw     8906/1/4   --        --
```

LLDP 에이전트 모니터링

lldpadm show-agent 하위 명령은 LLDP 에이전트가 알리는 전체 정보를 표시합니다. 지정된 시스템을 기준으로 알림은 네트워크의 나머지 부분으로 전송되는 로컬 시스템 정보일 수 있습니다. 또는 알림이 동일한 네트워크의 다른 시스템에서 시스템에 수신되는 정보일 수 있습니다.

▼ 알림을 표시하는 방법

이 절차에서는 LLDP 에이전트가 알리는 정보를 표시하는 방법을 보여줍니다. 정보는 로컬 또는 원격일 수 있습니다. **로컬** 정보는 로컬 시스템에서 제공됩니다. **원격** 정보는 네트워크의 다른 시스템에서 제공되며 로컬 시스템에 수신됩니다.

- **lldpadm show-agent 하위 명령에 적절한 옵션을 사용하여 원하는 정보를 표시합니다.**
  - **LLDP 에이전트가 알리는 로컬 정보를 표시하려면 다음 명령을 입력합니다.**

    # lldpadm show-agent -l *agent*
  - **LLDP 에이전트에 수신되는 원격 정보를 표시하려면 다음 명령을 입력합니다.**

    # lldpadm show-agent -r *agent*
  - **로컬 또는 원격 정보를 자세히 표시하려면 다음 명령을 입력합니다.**

    # lldpadm show-agent -[l|r]v *agent*

예 16–5 알리는 LLDP 에이전트 정보 가져오기

다음 예에서는 LLDP 에이전트가 로컬 또는 원격으로 알리는 정보를 표시하는 방법을 보여줍니다. 기본적으로 이 정보는 간결한 형태로 표시됩니다. -v 옵션을 사용하면 상세 정보 표시 또는 자세한 정보를 가져올 수 있습니다.

```
# lldpadm show-agent -l net0
AGENT   CHASSISID   PORTID
net0    004bb87f    00:14:4f:01:77:5d

# lldpadm show-agent -lv net0
                     Agent: net0
         Chassis ID Subtype: Local(7)
            Port ID Subtype: MacAddress(3)
                    Port ID: 00:14:4f:01:77:5d
           Port Description: net0
               Time to Live: 81 (seconds)
                System Name: hosta.example.com
         System Description: SunOS 5.11 dcb-clone-x-01-19-11 i86pc
     Supported Capabilities: bridge,router
       Enabled Capabilities: router
         Management Address: 192.168.1.2
         Maximum Frame Size: 3000
```

```
              Port VLAN ID: --
              VLAN Name/ID: vlan25/25
        VNIC PortID/VLAN ID: 02:08:20:72:71:31
    Aggregation Information: Capable, Not Aggregated
                PFC Willing: --
                    PFC Cap: --
                    PFC MBC: --
                 PFC Enable: --
Application(s) (ID/Sel/Pri): --
    Information Valid Until: 117 (seconds)

# lldpdadm show-agent -r net0
AGENT   SYSNAME   CHASSISID   PORTID
net0    hostb     0083b390    00:14:4f:01:59:ab

# llpdadm show-agent -rv net0
                      Agent: net0
         Chassis ID Subtype: Local(7)
            Port ID Subtype: MacAddress(3)
                    Port ID: 00:14:4f:01:59:ab
          Port Description: net0
              Time to Live: 121 (seconds)
               System Name: hostb.example.com
         System Description: SunOS 5.11 dcb-clone-x-01-19-11 i86pc
     Supported Capabilities: bridge,router
       Enabled Capabilities: router
         Management Address: 192.168.1.3
         Maximum Frame Size: 3000
               Port VLAN ID: --
               VLAN Name/ID: vlan25/25
        VNIC PortID/VLAN ID: 02:08:20:72:71:31
    Aggregation Information: Capable, Not Aggregated
                PFC Willing: --
                    PFC Cap: --
                    PFC MBC: --
                 PFC Enable: --
Application(s) (ID/Sel/Pri): --
    Information Valid Until: 117 (seconds)
```

▼ LLDP 통계를 표시하는 방법

LLDP 통계를 표시하여 로컬 시스템이나 원격 시스템이 알리는 LLDP 패킷에 대한 정보를 가져올 수 있습니다. 이 통계는 LLDP 패킷 전송 및 수신과 관련된 중요한 이벤트를 나타냅니다.

1 LLDP 패킷 전송 및 수신에 대한 모든 통계를 표시하려면 다음 명령을 사용합니다.

```
# lldpadm show-agent -s agent
```

2 선택한 통계 정보를 표시하려면 -o 옵션을 사용합니다.

```
# lldpadm show-agent -s -o field[,field,...]agent
```

여기서 *field*는 show-agent -s 명령의 출력 결과에 있는 임의 필드 이름을 나타냅니다.

예 16–6 LLDP 패킷 통계 표시

이 예에서는 LLDP 패킷 알림에 대한 정보를 표시하는 방법을 보여줍니다.

```
# lldpadm show-agent -s net0
AGENT IFRAMES IEER IDISCARD OFRAMES OLENERR TLVDISCARD TLVUNRECOG AGEOUT
net0        9    0        0      14       0          4          5      0
```

명령 출력 결과에서 다음 정보를 제공합니다.

- AGENT는 LLDP 에이전트가 사용으로 설정된 데이터 링크와 동일한 LLDP 에이전트의 이름을 지정합니다.
- IFRAMES, IEER 및 IDISCARD는 수신 중인 패킷, 오류가 있는 수신 패킷 및 삭제된 수신 패킷에 대한 정보를 표시합니다.
- OFRAMES 및 OLENERR은 송신 패킷 및 길이 오류가 있는 패킷을 나타냅니다.
- TLVDISCARD 및 TLVUNRECOG는 삭제된 TLV 단위와 인식할 수 없는 TLV 단위에 대한 정보를 표시합니다.
- AGEOUT은 시간 초과된 패킷을 나타냅니다.

이 예에서는 표준을 준수하지 않아서 시스템에 수신된 9개 프레임 중 5개 TLV를 인식할 수 없음을 나타냅니다. 또한 이 예에서는 로컬 시스템에서 네트워크로 14개 프레임이 전송되었음을 보여줍니다.

제 3 부

네트워크 가상화 및 리소스 관리

◆ ◆ ◆ 17 장

17 네트워크 가상화 및 리소스 제어 소개(개요)

이 장에서는 네트워크 가상화 및 리소스 제어에 관련된 기본 개념에 대해 설명합니다. 다음 항목을 다룹니다.

- 네트워크 가상화
- 가상 네트워크 유형
- 가상 시스템 및 영역
- 흐름 관리를 비롯한 리소스 제어
- 향상된 네트워크 관찰 기능

이러한 기능은 흐름 제어를 관리하고 시스템 성능을 향상시키며 OS 가상화, 유틸리티 컴퓨팅, 서버 통합 실현 등에 필요한 네트워크 사용률을 구성하는 데 유용합니다.

특정 작업의 경우 다음 장을 참조하십시오.

- 19 장, “가상 네트워크 구성(작업)”
- 22 장, “네트워크 트래픽 및 리소스 사용 모니터링”
- 20 장, “가상화된 환경에서 링크 보호 사용”
- 21 장, “네트워크 리소스 관리”

네트워크 가상화 및 가상 네트워크

네트워크 가상화는 하드웨어 네트워크 리소스 및 소프트웨어 네트워크 리소스를 단일 관리 단위로 결합하는 프로세스입니다. 네트워크 가상화의 목적은 시스템과 사용자에게 네트워크 리소스의 효율적이고 제어된 보안 공유를 제공하는 것입니다.

네트워크 가상화의 최종 결과물은 **가상 네트워크**입니다. 가상 네트워크는 두 가지 광범위한 유형인 외부와 내부로 분류됩니다. **외부 가상 네트워크**는 소프트웨어에서 단일 엔티티로 관리하는 여러 로컬 네트워크로 구성됩니다. 클래식 외부 가상 네트워크의 빌딩 블록은 스위치 하드웨어 및 VLAN 소프트웨어 기술입니다. 외부 가상 네트워크의 예로 대규모 회사 네트워크와 데이터 센터가 있습니다.

내부 가상 네트워크는 하나 이상의 의사 네트워크 인터페이스에 구성된 가상 시스템 또는 영역을 사용하는 시스템 한 개로 구성됩니다. 이러한 컨테이너는 동일한 로컬 네트워크에 있는 것처럼 서로 통신하여 단일 호스트에 가상 네트워크를 제공할 수 있습니다. 가상 네트워크의 빌딩 블록은 *VNIC(**가상 네트워크 인터페이스 카드 또는 가상 NIC)* 및 가상 스위치입니다. Oracle Solaris 네트워크 가상화는 내부 가상 네트워크 솔루션을 제공합니다.

네트워킹 리소스를 결합하여 내부 및 외부 가상 네트워크를 모두 구성할 수 있습니다. 예를 들어, 내부 가상 네트워크가 있는 개별 시스템을 대규모 외부 가상 네트워크의 일부인 LAN으로 구성할 수 있습니다. 이 부에서 설명하는 네트워크 구성에는 내부 및 외부 가상 네트워크 조합의 예도 포함되어 있습니다.

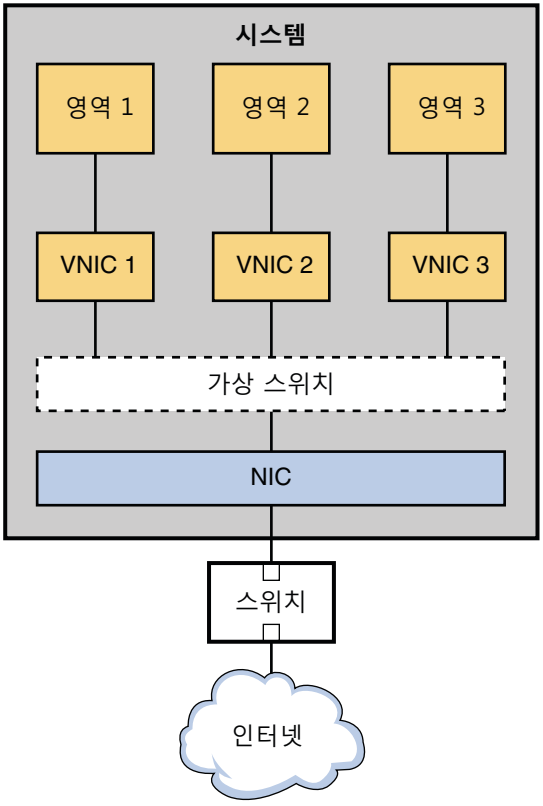
내부 가상 네트워크의 부분

Oracle Solaris에 작성된 내부 가상 네트워크는 다음 부분으로 구성되어 있습니다.

- 하나 이상의 NIC(네트워크 인터페이스 카드)
- 네트워크 인터페이스 위에 구성된 VNIC(가상 NIC)
- 인터페이스의 첫번째 VNIC와 동시에 구성된 가상 스위치
- VNIC 위에 구성된 컨테이너(예: 영역 또는 가상 시스템)

다음 그림에서는 이러한 부분과 단일 시스템에서 부분이 결합되는 방식을 보여줍니다.

그림 17–1 단일 인터페이스에 대한 VNIC 구성



이 그림에서는 NIC 한 개가 있는 단일 시스템을 보여줍니다. NIC는 VNIC 세 개로 구성됩니다. 각 VNIC는 단일 영역을 지원합니다. 따라서 영역 1, 영역 2 및 영역 3은 각각 VNIC 1, VNIC 2 및 VNIC 3에 구성됩니다. VNIC 세 개는 하나의 가상 스위치에 가상으로 연결되어 있습니다. 이 스위치는 VNIC와 VNIC가 작성된 물리적 NIC 간에 연결을 제공합니다. 물리적 인터페이스는 시스템에 외부 네트워크 연결을 제공합니다.

또는 etherstub을 기반으로 가상 네트워크를 만들 수 있습니다. etherstub은 순수하게 소프트웨어이며 가상 네트워크의 기반으로 네트워크 인터페이스가 필요하지 않습니다.

*VNIC*는 물리적 인터페이스와 동일한 데이터 링크 인터페이스가 있는 가상 네트워크 장치입니다. 물리적 인터페이스 위에 VNIC를 구성합니다. VNIC를 지원하는 물리적 인터페이스의 현재 목록은 Network Virtualization and Resource Control FAQ (`http://hub.opensolaris.org/bin/view/Project+crossbow/faq`)를 참조하십시오. 단일 물리적 인터페이스에 VNIC를 최대 900개까지 구성할 수 있습니다. VNIC를 구성하면 물리적 NIC처럼 동작합니다. 또한 시스템 리소스가 VNIC를 물리적 NIC처럼 처리합니다.

각 VNIC는 물리적 인터페이스에 해당하는 **가상 스위치**에 암시적으로 연결됩니다. 가상 스위치는 스위치 하드웨어가 스위치 포트에 연결된 시스템에 제공하는 가상 네트워크의 VNIC 간에 동일한 연결을 제공합니다.

이더넷 설계에 따라 스위치 포트가 해당 포트에 연결된 호스트로부터 송신 패킷을 받는 경우 해당 패킷이 동일한 포트의 대상으로 이동할 수 없습니다. 이 설계는 영역이나 가상 시스템으로 구성된 시스템의 결점입니다. 네트워크 가상화가 없을 경우 가상 시스템이나 배타적 스택이 있는 영역에서 송신 패킷을 동일한 시스템의 다른 가상 시스템이나 영역으로 전달할 수 없습니다. 송신 패킷은 스위치 포트에서 외부 네트워크로 이동합니다. 패킷은 전송된 포트와 동일한 포트를 통해 반환될 수 없기 때문에 수신 패킷이 대상 영역이나 가상 시스템에 도달할 수 없습니다. 따라서 동일한 시스템의 가상 시스템과 영역이 통신해야 하는 경우 컨테이너 간의 데이터 경로가 로컬 시스템에서 열려 있어야 합니다. 가상 스위치는 이러한 컨테이너에 패킷 전달 방식을 제공합니다.

가상 네트워크를 통해 데이터가 이동하는 방식

그림 17–1에서는 단일 시스템의 가상 네트워크에 대한 단순한 VNIC 구성을 보여줍니다.

가상 네트워크가 구성된 경우 영역에서 가상 네트워크가 없는 시스템과 동일한 방식으로 외부 호스트에 트래픽을 보냅니다. 트래픽은 영역에서 VNIC를 통해 가상 스위치와 물리적 인터페이스 순서로 흐르며, 여기서 데이터를 네트워크로 보냅니다.

그러나 앞에서 언급한 이더넷 제한 사항이 있다면 가상 네트워크의 한 영역에서 가상 네트워크의 다른 영역으로 패킷을 보내려는 경우 어떤 작업이 수행될까요? 그림 17–1과 같이 영역 1에서 트래픽을 영역 3으로 보내질까요? 이 경우는 패킷이 영역 1에서 전용 VNIC 1을 통해 전달됩니다. 그런 다음 트래픽이 가상 스위치를 통해 VNIC 3으로 전달됩니다. VNIC 3은 트래픽을 영역 3으로 전달합니다. 트래픽이 시스템을 벗어나지 않았으므로 이더넷 제한 사항에 위반되지도 않습니다.

가상 네트워크 구현 대상

Oracle Sun 서버에서 리소스를 통합해야 하는 경우 VNIC 및 가상 네트워크를 구현해 보십시오. ISP, 텔레콤 회사 및 대규모 금융 기관의 통합자는 다음 네트워크 가상화 기능을 사용하여 서버와 네트워크의 성능을 향상시킬 수 있습니다.

- 하드웨어 링을 지원하는 강력한 새 인터페이스를 비롯한 NIC 하드웨어
- VNIC의 여러 MAC 주소
- 최신 인터페이스가 제공하는 대량 대역폭

분리, 보안 및 유연성의 큰 손실 없이 여러 영역이나 가상 시스템 실행을 구현하는 단일 시스템으로 많은 시스템을 교체할 수 있습니다.

리소스 제어

리소스 제어는 시스템 리소스를 제어된 방식으로 할당하는 프로세스입니다. Oracle Solaris 리소스 제어 기능을 사용하면 시스템 가상 네트워크의 VNIC 간에 대역폭을 공유할 수 있습니다. 리소스 제어 기능을 사용하면 VNIC 및 가상 시스템 없이 물리적 인터페이스에서 대역폭을 할당하고 관리할 수도 있습니다. 이 절에서는 리소스 제어의 주요 기능을 소개하고 이러한 기능의 작동 방식을 간략하게 설명합니다.

대역폭 관리 및 흐름 제어의 작동 방식

Searchnetworking.com (`http://searchnetworking.techtarget.com`)에서는 "정해진 기간(대체로 1초) 동안 한 지점에서 다른 지점으로 전달될 수 있는 데이터 양"으로 대역폭을 정의합니다. **대역폭 관리**를 사용하면 물리적 NIC의 사용 가능한 대역폭 중 일부를 응용 프로그램이나 고객 등의 소비자에게 할당할 수 있습니다. 응용 프로그램별, 포트별, 프로토콜별 및 주소별로 대역폭을 제어할 수 있습니다. 대역폭 관리는 새 GLDv3 네트워크 인터페이스에서 사용 가능한 대량 대역폭을 효율적으로 사용할 수 있게 합니다.

리소스 제어 기능을 사용하면 인터페이스의 사용 가능한 대역폭에서 일련의 제어를 구현할 수 있습니다. 예를 들어, 인터페이스 대역폭의 **보장**을 특정 소비자에 설정할 수 있습니다. 이 보장은 응용 프로그램 또는 엔터프라이즈에 할당되는 최소 보장 대역폭 양입니다. 할당된 대역폭 부분을 **공유**라고 합니다. 보장을 설정하면 특정 양의 대역폭이 없을 경우 제대로 작동할 수 없는 응용 프로그램에 충분한 대역폭을 할당할 수 있습니다. 예를 들어, 스트리밍 매체와 VoIP(Voice over IP)는 많은 대역폭을 사용합니다. 리소스 제어 기능을 사용하면 이러한 두 응용 프로그램을 성공적으로 실행하는 데 충분한 대역폭을 보장할 수 있습니다.

또한 공유에 **제한**을 설정할 수 있습니다. 제한은 공유에서 사용할 수 있는 최대 대역폭 할당입니다. 제한을 사용하여 중요하지 않은 서비스가 중요 서비스로부터 대역폭을 빼앗지 않도록 제한할 수 있습니다.

마지막으로, 소비자에게 할당된 다양한 공유에 우선 순위를 지정할 수 있습니다. 클러스터의 하트비트 패킷과 같은 중요 트래픽에는 최고 우선 순위를 지정하고 덜 중요한 응용 프로그램에는 낮은 우선 순위를 지정할 수 있습니다.

예를 들어, ASP(응용 프로그램 서비스 공급자)는 고객이 구입하는 대역폭 공유를 기반으로 하는 유료 서비스 레벨을 고객에게 제공할 수 있습니다. SLA(서비스 단계 계약)의 일부로, 구입한 제한을 초과하지 않는 양의 대역폭이 각 공유에 보장됩니다. 서비스 단계 계약에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 "서비스 단계 계약 구현"을 참조하십시오. 우선 순위 제어는 SLA의 여러 계층 또는 SLA 고객이 지불한 가격을 기반으로 할 수도 있습니다.

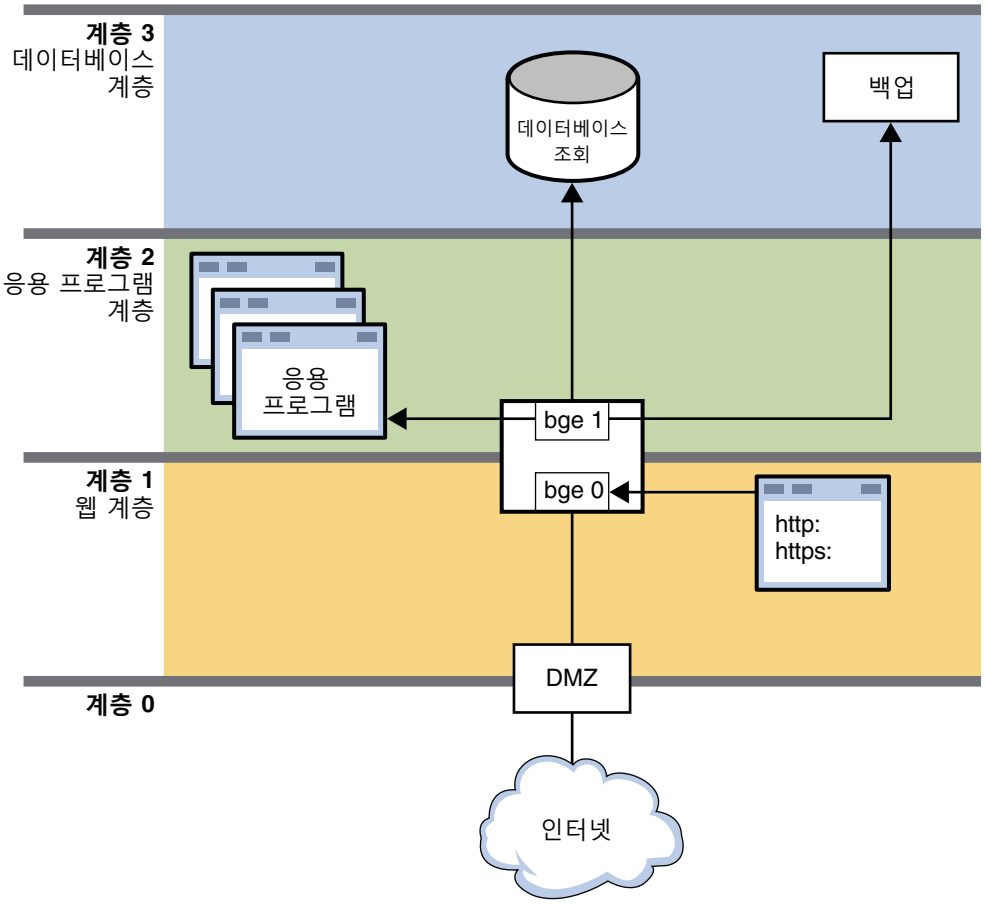
대역폭 사용은 흐름 관리를 통해 제어됩니다. **흐름**은 모두 포트 번호나 대상 주소와 같은 특정 특성을 가진 패킷 스트림입니다. 이러한 흐름은 영역을 비롯한 전송, 서비스 또는 가상 시스템에서 관리됩니다. 흐름은 응용 프로그램 또는 고객이 구입한 공유에 보장된 대역폭 양을 초과할 수 없습니다.

VNIC 또는 흐름에 보장이 할당되면 다른 흐름이나 VNIC에서 인터페이스를 사용하는 경우에도 VNIC에 지정된 대역폭이 보장됩니다. 하지만 할당된 보장은 물리적 인터페이스의 최대 대역폭을 초과하지 않는 경우에만 작동합니다.

네트워크에 리소스 제어 및 대역폭 관리 할당

다음 그림에서는 리소스 제어를 사용하여 다양한 응용 프로그램을 관리하는 회사 네트워크 토폴로지를 보여줍니다.

그림 17–2 리소스 제어가 적용된 네트워크



이 그림에서는 리소스 제어를 사용하여 네트워크 효율성과 성능을 향상시키는 일반적인 네트워크 토폴로지를 보여줍니다. 네트워크는 배타적 영역 및 가상 시스템과 같은 컨테이너와 VNIC를 구현하지 않습니다. 하지만 이 네트워크에서 통합 및 기타 용도로 VNIC와 컨테이너를 사용할 수 있습니다.

네트워크는 다음 4개의 계층으로 나뉩니다.

- **계층 0**은 DMZ(완충 영역)으로, 외부와의 액세스를 제어하는 소규모 로컬 네트워크입니다. DMZ의 시스템에서는 리소스 제어가 사용되지 않습니다.
- **계층 1**은 웹 계층이며 두 시스템을 포함합니다. 첫번째 시스템은 필터링을 수행하는 프록시 서버입니다. 이 서버에는 두 개의 인터페이스 `bge0`과 `bge1`이 있습니다. `bge0` 링크는 프록시 서버를 계층 0의 DMZ에 연결합니다. 또한 `bge0` 링크는 프록시 서버를 두번째 시스템인 웹 서버에 연결합니다. `http` 및 `https` 서비스는 웹 서버의 대역폭을 다른 표준 응용 프로그램과 공유합니다. 웹 서버의 크기 및 중요성 때문에 `http` 및 `https`에는 보장과 우선 순위가 필요합니다.
- **계층 2**는 응용 프로그램 계층이며 역시 두 시스템을 포함합니다. 프록시 서버의 두번째 인터페이스인 `bge1`은 웹 계층과 응용 프로그램 계층 간의 연결을 제공합니다. 스위치를 통해 애플리케이션 서버가 프록시 서버의 `bge1`에 연결합니다. 애플리케이션 서버에서는 리소스 제어를 통해 실행 중인 여러 응용 프로그램에 제공되는 대역폭 공유를 관리해야 합니다. 많은 대역폭이 필요한 중요 응용 프로그램은 더 작고 덜 중요한 응용 프로그램보다 높은 보장과 우선 순위가 지정되어야 합니다.
- **계층 3**은 데이터베이스 계층입니다. 이 계층의 두 시스템은 스위치를 통해 프록시 서버의 `bge1` 인터페이스에 연결합니다. 첫번째 시스템인 데이터베이스 서버는 보장을 실행하고 데이터베이스 조회에 관련된 다양한 프로세스에 우선 순위를 지정해야 합니다. 두번째 시스템은 네트워크의 백업 서버입니다. 이 시스템은 백업 도중 많은 대역폭을 사용해야 합니다. 하지만 백업 작업은 대체로 야간에 수행됩니다. 리소스 제어를 사용하면 백업 프로세스에 최고 대역폭 보장과 최고 우선 순위가 지정되는 시간을 제어할 수 있습니다.

리소스 제어 기능 구현 대상

시스템의 효율성과 성능을 향상시키려는 시스템 관리자는 리소스 제어 기능의 구현을 고려해야 합니다. 통합자는 VNIC와 함께 대역폭 공유를 위임하여 대규모 서버의 부하를 균형 있게 조정할 수 있습니다. 서버 관리자는 공유 할당 기능을 사용하여 ASP가 제공하는 SLA 등의 SLA를 구현할 수 있습니다. 일반적으로 시스템 관리자는 대역폭 관리 기능을 사용하여 특정 응용 프로그램을 격리시키고 우선 순위를 지정할 수 있습니다. 마지막으로, 공유 할당을 사용하면 개별 소비자의 대역폭 사용을 쉽게 관찰할 수 있습니다.

네트워크 가상화 및 리소스 제어의 관찰 기능

네트워크 가상화 및 리소스 제어에는 VNIC, 흐름 등의 제어를 설정하기 전에 리소스 사용을 확인할 수 있는 관찰 기능이 포함되어 있습니다. Oracle Solaris 확장 계정과 함께 리소스 제어 관찰 기능을 사용하여 시스템 통계를 로그에 누적할 수 있습니다. 네트워크 가상화 및 리소스 제어의 관찰 기능에는 다음이 포함됩니다.

- 실행 중인 시스템을 모니터하는 기능
- 통계를 기록하고 보고하는 기능
- 기록 데이터를 기록하는 확장 계정 기능

새 `flowadm` 명령과 `dladm` 및 `netstat` 명령에 대한 확장이 네트워크 가상화 관찰 기능을 구현합니다. 이러한 명령을 사용하여 현재 시스템 사용을 모니터하고 통계 데이터를 로그에 수집할 수 있습니다.

기록 로그를 분석하여 다음을 확인할 수 있습니다.

- 여러 시스템의 네트워크 리소스가 차세대 네트워크 인터페이스를 통해 많은 대역폭을 가진 단일 시스템으로 통합될 수 있는 위치. 이 작업은 VNIC와 가상 시스템 또는 배타적 영역을 설정하기 전에 수행합니다.
- 가장 많은 대역폭을 사용할 수 있는 응용 프로그램. 이 정보는 특정 시간 슬롯 내에서 중요 응용 프로그램에 가장 많은 대역폭이 보장되도록 대역폭 관리를 설정하는 데 유용할 수 있습니다. 예를 들어, 하루 20시간 동안 가장 많은 인터페이스 대역폭을 비디오 스트림에 보장할 수 있습니다. 하루 4시간 동안은 시스템의 백업 프로그램에 최고 우선 순위를 지정할 수 있습니다. 이 작업은 대역폭 관리 구현의 일부로 수행합니다.
- 사용한 대역폭에 대해 고객에게 청구할 금액. 응용 프로그램 서비스 공급자 및 시스템 공간을 대여한 기타 업체에서는 리소스 제어 관찰 기능을 사용하여 유료 고객의 사용을 확인할 수 있습니다. 일부 업체는 고객이 공급자로부터 보장된 대역폭 비율을 구입하는 서비스 단계 계약을 고객에게 제공합니다. 관찰 기능을 통해 각 고객이 사용하는 대역폭 양을 확인하고 추가 사용에 대해 요금을 청구할 수 있습니다. 기타 업체는 사용별 기준으로 고객에게 대역폭을 제공합니다. 이 경우 관찰 기능이 요금 청구에 직접 도움이 됩니다. 이 작업은 시스템에서 리소스 제어와 VNIC 및 가상 시스템을 구현한 후에 수행합니다.

다음 장인 18 장, “네트워크 가상화 및 리소스 제어 계획”에는 관찰 기능이 통합 및 리소스 제어 계획에 사용되는 위치를 보여주는 시나리오가 포함되어 있습니다.

◆ ◆ ◆ 18 장

18 네트워크 가상화 및 리소스 제어 계획

이 장에는 평가 후에 사이트의 네트워크 가상화 및 리소스 제어 솔루션을 설계할 수 있도록 도와주는 정보와 시나리오 예가 포함되어 있습니다. 이 장에서 설명하는 시나리오는 다음과 같습니다.

- 322 페이지 “단일 시스템의 기본 가상 네트워크”
- 324 페이지 “단일 시스템의 개인 가상 네트워크”
- 328 페이지 “일반 네트워크에 대한 인터페이스 기반 리소스 제어”

각 시나리오에는 특정 시나리오가 가장 유용한 네트워크 유형을 설명하는 "최상의 사용 방법"에 대한 제안 사항이 포함됩니다.

네트워크 가상화 및 리소스 제어 작업 맵

다음 표에서는 가상 네트워크를 구성하고 해당 네트워크에서 리소스 제어를 구현하는 작업에 대해 설명합니다.

작업	설명	수행 방법
단일 호스트에서 가상 네트워크를 설계하고 계획합니다.	로컬 네트워크가 제공하는 네트워크 서비스와 응용 프로그램을 단일 호스트에 통합합니다. 이 시나리오는 통합자와 서비스 공급자에게 특히 유용합니다.	322 페이지 “가상 네트워크 계획 및 설계”
단일 호스트에서 개인 가상 네트워크를 설계하고 계획합니다.	공용 액세스를 허용하지 않는 가상 네트워크를 실행합니다. 이 시나리오는 개발 환경을 실행해야 하는 시스템 관리자에게 권장됩니다.	324 페이지 “단일 시스템의 개인 가상 네트워크”

작업	설명	수행 방법
인터페이스별 기준으로 시스템에 대역폭 관리 및 리소스 제어를 제공합니다.	패킷 트래픽에 대해 특정 인터페이스 대역폭 양을 격리시키고, 우선 순위를 지정하고, 할당합니다. 이 시나리오는 웹 서비스나 데이터베이스 서버와 같은 특정 서비스의 많은 트래픽을 처리하는 시스템에 유용합니다.	328 페이지 “일반 네트워크에 대한 인터페이스 기반 리소스 제어”

가상 네트워크 계획 및 설계

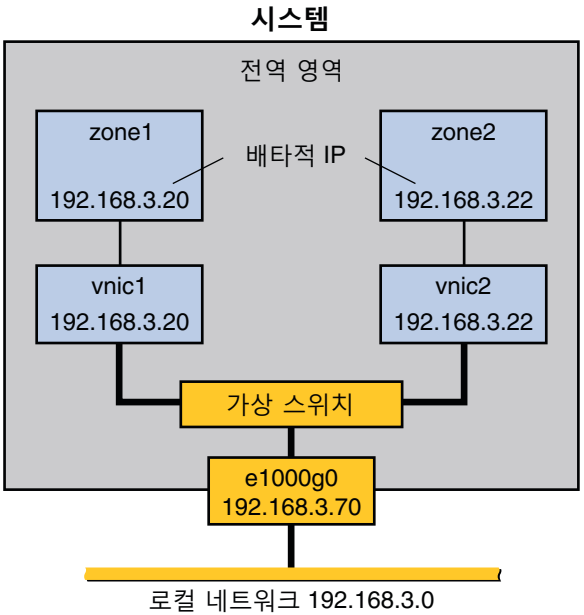
이 절에서는 가상 네트워크를 구성하는 두 가지 시나리오에 대해 설명합니다. 시나리오를 살펴보고 사이트의 요구에 가장 맞는 시나리오를 결정합니다. 그런 다음 이 시나리오를 특정 가상화 솔루션 설계의 기준으로 사용합니다. 시나리오에는 다음이 포함되어 있습니다.

- 특히 로컬 네트워크의 네트워크 서비스를 단일 호스트로 통합하는 데 유용한 두 영역의 기본 가상 네트워크
- 응용 프로그램과 서비스를 공용 네트워크에서 격리시키는 개발 환경에 유용한 개인 가상 네트워크

단일 시스템의 기본 가상 네트워크

그림 18–1에서는 332 페이지 “Oracle Solaris에서 네트워크 가상화의 구성 요소 구성” 절 전체의 예에서 사용된 기본 가상 네트워크, 즉 "시스템 내 네트워크"를 보여줍니다.

그림 18–1 단일 호스트의 가상 네트워크



이 가상 네트워크는 다음 요소로 구성됩니다.

- 단일 GLDv3 네트워크 인터페이스 `e1000g0`. 이 인터페이스는 공용 네트워크 `192.168.3.0/24`에 연결합니다. `e1000g0` 인터페이스의 IP 주소는 `192.168.3.70`입니다.
- 첫번째 VNIC를 만들 때 자동으로 구성되는 가상 스위치
- VNIC 두 개. `vnic1`의 IP 주소는 `192.168.3.20`이고 `vnic2`의 IP 주소는 `192.168.3.22`입니다.
- VNIC가 할당되는 배타적 IP 영역 두 개. `vnic1`은 `zone1`에 할당되고 `vnic2`는 `zone2`에 할당됩니다.

이 구성의 VNIC와 영역은 공용 액세스를 허용합니다. 따라서 영역이 `e1000g0` 인터페이스 이외의 영역까지 트래픽을 전달할 수 있습니다. 마찬가지로, 외부 네트워크의 사용자가 영역에서 제공하는 응용 프로그램과 서비스에 연결할 수 있습니다.

기본 가상 네트워크의 최상의 사용 방법

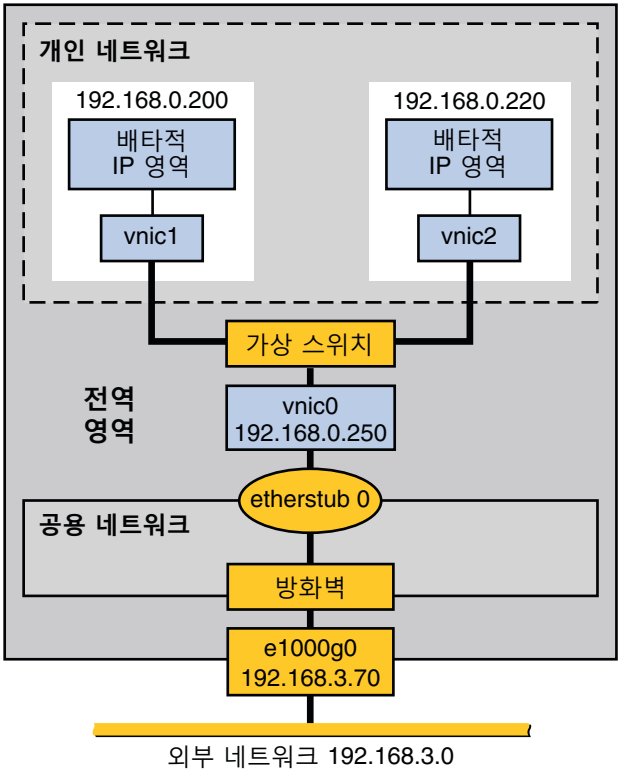
시스템 내 네트워크 시나리오를 사용하면 프로세스와 응용 프로그램을 단일 호스트의 개별 가상 시스템이나 영역에 격리시킬 수 있습니다. 또한 이 시나리오는 각각 완전히 격리된 응용 프로그램 세트를 실행할 수 있는 여러 컨테이너를 포함하도록 확장할 수 있습니다. 이 시나리오는 시스템의 효율성 및 확장을 통해 로컬 네트워크의 효율성도 향상시켜 줍니다. 따라서 이 시나리오는 다음 사용자에게 적합합니다.

- 네트워크 통합자 및 LAN의 서비스를 단일 시스템으로 통합하려는 사용자
- 고객에게 서비스를 대여하는 사이트. 개별 영역이나 가상 시스템을 대여하고, 트래픽을 관찰하고, 가상 네트워크의 각 영역에서 성능 측정 또는 요금 청구 용도로 통계를 수집할 수 있습니다.
- 프로세스와 응용 프로그램을 개별 컨테이너에 격리시켜 시스템 효율성을 향상시키려는 관리자.

단일 시스템의 개인 가상 네트워크

그림 18–2에서는 NAT(Network Address Translation)를 수행하는 패킷 필터링 소프트웨어 뒤에 개인 네트워크가 있는 단일 시스템을 보여줍니다. 이 그림에서는 예 19–5에서 작성된 시나리오를 보여줍니다.

그림 18–2 단일 호스트의 개인 가상 네트워크



이 토폴로지는 방화벽을 비롯한 공용 네트워크와 개인 네트워크가 etherstub 의사 인터페이스에 작성된 단일 시스템을 사용합니다. 공용 네트워크는 전역 영역에서 실행되며 다음 요소로 구성됩니다.

- IP 주소가 `192.168.3.70`인 GLDv3 네트워크 인터페이스 `e1000g0`
- IP 필터 소프트웨어에서 구현된 방화벽. IP 필터에 대한 소개는 **Oracle Solaris 관리: IP 서비스**의 "IP 필터 소개"를 참조하십시오.
- 가상 네트워크 토폴로지가 작성된 의사 인터페이스인 `etherstub0`. *etherstub*은 호스트에 가상 네트워크를 만드는 기능을 제공합니다. 해당 네트워크는 외부 네트워크로부터 완전히 격리됩니다.

개인 네트워크는 다음 요소로 구성됩니다.

- 개인 네트워크의 VNIC 간에 패킷 전달을 제공하는 가상 스위치
- 전역 영역에 대한 VNIC이며 IP 주소가 `192.168.0.250`인 `vnic0`
- IP 주소가 `192.168.0.200`인 `vnic1` 및 IP 주소가 `192.168.0.220`인 `vnic2`. 세 VNIC는 모두 `etherstub0`에 구성됩니다.
- `vnic1`은 `zone1`에 할당되고 `vnic2`는 `zone2`에 할당됩니다.

개인 가상 네트워크의 최상의 사용 방법

개발 환경에서 사용되는 호스트의 경우 개인 가상 네트워크를 만들어 볼 수 있습니다. etherstub 프레임워크를 사용하면 개발 중인 소프트웨어나 기능을 개인 네트워크의 컨테이너에 완전히 격리시킬 수 있습니다. 또한 개인 네트워크의 컨테이너에서 시작된 송신 패킷의 NAT(Network Address Translation)를 위해 방화벽 소프트웨어를 사용할 수 있습니다. 개인 네트워크는 궁극적인 배치 환경의 축소된 버전입니다.

자세한 정보

- 가상 네트워크를 구성하고 이 장에 설명된 시나리오를 구현하는 절차는 345 페이지 "개인 가상 네트워크 만들기"를 참조하십시오.
- VNIC 및 가상 네트워크에 대한 개념 정보는 313 페이지 "네트워크 가상화 및 가상 네트워크"를 참조하십시오.
- 영역에 대한 개념 정보는 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 15 장, "Oracle Solaris Zones 소개"를 참조하십시오.
- IP 필터에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 "IP 필터 소개"를 참조하십시오.

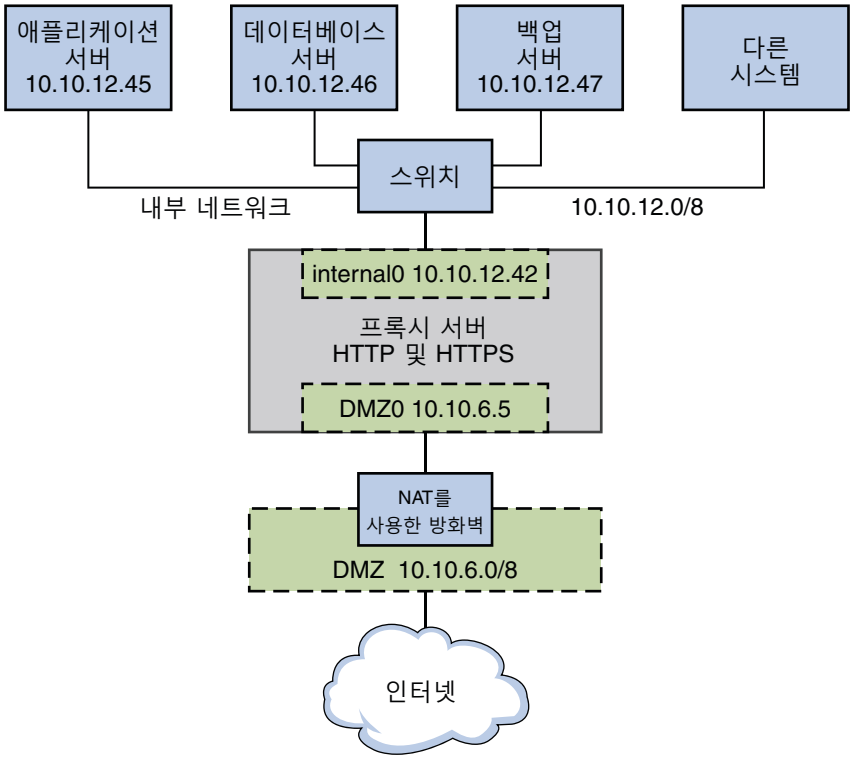
네트워크 리소스에 대한 제어 구현

네트워크 가상화를 사용하면 시스템 내 네트워크를 구성하여 더 낮은 비용으로 보다 효율적으로 네트워크 설정을 구현할 수 있습니다. 효율성을 증가시키기 위해 네트워킹 프로세스의 리소스 사용 방식을 결정하는 제어를 구현할 수도 있습니다. 특히 링, CPU 등의 네트워크 리소스와 관련된 링크 등록 정보를 사용자 정의하여 네트워크 패킷을 처리할 수 있습니다. 또한 네트워크 사용을 관리할 흐름을 만들 수 있습니다. 네트워크 리소스 제어는 21 장, "네트워크 리소스 관리"에서 자세히 설명합니다.

그림 18–3에서는 프록시 서버의 대역폭을 관리해야 하는 중소업체의 네트워크 토폴로지를 보여줍니다. 프록시 서버는 사이트의 내부 네트워크에 있는 여러 서버의 서비스가 필요한 내부 클라이언트에 프록시와 공용 웹 사이트를 제공합니다.

주 – 이 시나리오는 가상 네트워크에 대한 흐름 제어를 구현하는 방법을 보여주지 않으므로 VNIC를 포함하지 않습니다. 가상 네트워크의 흐름 제어는 가상 네트워크에 대한 흐름 제어를 참조하십시오.

그림 18–3 기존 네트워크의 프록시 서버에 대한 리소스 제어



이 그림에서는 회사에 DMZ(완충 영역) 역할도 수행하는 공용 네트워크 `10.10.6.0/8`이 있음을 보여줍니다. DMZ의 시스템은 IP 필터 방화벽을 통해 NAT(Name-to-Address Translation)를 제공합니다. 이 회사에는 프록시 서버 역할을 수행하는 대규모 시스템이 있습니다. 시스템에 유선 인터페이스 두 개와 ID가 0-16인 프로세서 세트 16개가 있습니다. 이 시스템은 IP 주소가 `l0 10.6.5`인 `nge0` 인터페이스를 통해 공용 네트워크에 연결됩니다. 인터페이스의 링크 이름은 `DMZ0`입니다. `DMZ0`을 통해 프록시 서버는 회사의 공용 웹 사이트 전체에 HTTP 및 HTTPS 서비스를 제공합니다.

이 그림에서는 회사의 내부 네트워크인 `10.10.12.0/24`도 보여줍니다. 프록시 서버는 IP 주소가 `10.10.12.42`인 `nge1` 인터페이스를 통해 내부 `10.10.12.0/8` 네트워크에 연결합니다. 이 인터페이스의 링크 이름은 `internal0`입니다. `internal0` 데이터 링크를 통해 프록시 서버는 애플리케이션 서버 `10.10.12.45`, 데이터베이스 서버 `10.10.12.46` 및 백업 서버 `10.10.12.47`의 서비스를 요청하는 내부 클라이언트 대신 작업합니다.

일반 네트워크에 대한 인터페이스 기반 리소스 제어

일반 네트워크에 대한 인터페이스 기반 리소스 제어 최상의 사용 방법

많이 사용하는 시스템, 특히 사용 가능한 대량의 대역폭을 제공하는 최신 GLDv3 인터페이스가 있는 시스템에 대해 흐름 제어를 설정해 보십시오. 인터페이스 기반 흐름 제어는 인터페이스, 시스템 및 잠재적으로 네트워크의 효율성을 향상시켜 줍니다. 모든 네트워크 유형의 임의 시스템에 흐름 제어를 적용할 수 있습니다. 또한 네트워크 효율성 향상을 목적으로 하는 경우 다양한 서비스를 개별 흐름으로 분리할 수 있습니다. 이 작업은 개별 흐름에 별도의 하드웨어 및 소프트웨어 리소스를 할당하므로 특정 시스템의 다른 서비스로부터 흐름이 격리됩니다. 흐름을 설정한 후 각 흐름의 트래픽을 관찰하고 통계를 수집할 수 있습니다. 그런 후에 대역폭 양과 우선 순위를 할당하여 인터페이스의 사용을 제어할 수 있습니다.

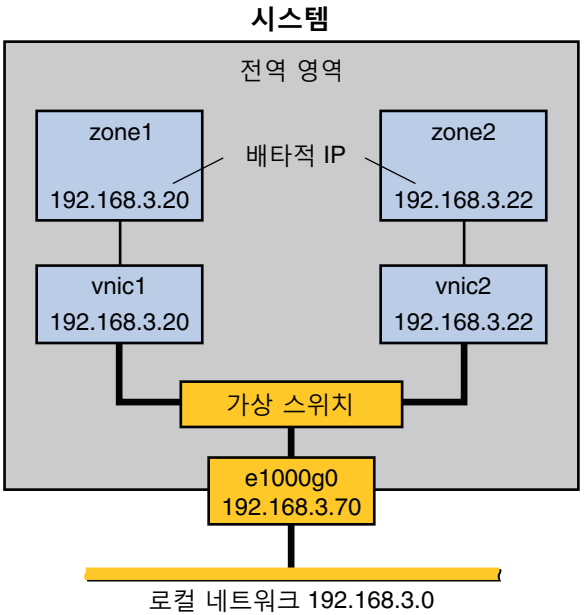
자세한 정보

- 흐름 제어를 구현하는 작업은 21 장, “네트워크 리소스 관리”를 참조하십시오.
- 대역폭 관리 및 리소스 제어에 대한 개념 정보는 317 페이지 “리소스 제어”을 참조하십시오.
- 자세한 기술 정보는 `dladm`(1M) 및 `flowadm`(1M) 매뉴얼 페이지를 참조하십시오.

가상 네트워크에 대한 흐름 제어

이 시나리오에서는 322 페이지 “단일 시스템의 기본 가상 네트워크”에서 소개된 기본 가상 네트워크 등의 가상 네트워크 내에서 흐름 제어를 사용하는 방식을 보여줍니다.

그림 18–4 흐름 제어가 적용된 기본 가상 네트워크



토폴로지는 322 페이지 “단일 시스템의 기본 가상 네트워크”에서 설명합니다. 여기서 호스트에는 VNIC 두 개(`vnic1` 및 `vnic2`)가 포함된 네트워크 인터페이스 한 개(`e1000g0`)가 있습니다. `zone1`은 `vnic1`에 구성되고 `zone2`는 `vnic2`에 구성됩니다. 가상 네트워크에 대한 리소스 관리에는 VNIC별 기준으로 흐름을 만드는 작업이 포함됩니다. 이러한 흐름은 송신 호스트의 포트 번호나 IP 주소와 같은 유사한 특성을 가진 패킷을 정의하고 격리시킵니다. 시스템에 대한 사용 정책에 따라 대역폭을 할당합니다.

VNIC 트래픽에 대한 흐름 제어의 다른 일반적인 사용은 영역을 대여하는 회사에서 이루어집니다. 고객별로 다른 서비스 단계 계약을 만들고 보장된 대역폭 양으로 영역을 대여합니다. 영역별 기준으로 흐름을 만드는 경우 각 고객의 트래픽을 격리 및 관찰하고 대역폭 사용을 모니터할 수 있습니다. 서비스 단계 계약이 엄격히 사용을 기반으로 하는 경우 통계 및 계정 기능을 사용하여 고객에게 요금을 청구할 수 있습니다.

흐름 제어는 영역의 트래픽에 대해 대역폭 관리가 필요한 네트워크에 효율적입니다. ASP(응용 프로그램 서비스 공급자) 또는 ISP(인터넷 서비스 공급자)와 같은 대규모 조직은 데이터 센터 및 멀티 프로세서 시스템의 VNIC에 대해 리소스 제어를 활용할 수 있습니다. 개별 영역을 고객에게 서로 다른 서비스 레벨로 대여할 수 있습니다. 따라서 `zone1`을 표준 가격으로 대여하고 표준 대역폭을 제공할 수 있습니다. 그런 다음 `zone2`를 프리미엄 가격으로 대여하고 해당 고객에게 높은 레벨의 대역폭을 제공할 수 있습니다.

▼ 가상 네트워크에서 응용 프로그램에 대한 사용 정책을 만드는 방법

1. **호스트에서 실행하려는 응용 프로그램을 나열합니다.**
2. **기록상 가장 많은 대역폭을 사용했거나 가장 많은 대역폭이 필요한 응용 프로그램을 확인합니다.**

   예를 들어, `telnet` 응용 프로그램은 시스템에서 많은 대역폭을 사용하지 않지만 자주 사용됩니다. 반면, 데이터베이스 응용 프로그램은 많은 대역폭을 사용하지만 가끔 사용됩니다. 영역에 응용 프로그램을 할당하기 전에 해당 응용 프로그램에 대한 트래픽을 모니터해 보십시오. 387 페이지 “링크의 네트워크 트래픽에 대한 통계 수집”에 설명된 대로 `dladm show-link` 명령의 통계 옵션을 사용하여 통계를 수집할 수 있습니다.
3. **이러한 응용 프로그램을 개별 영역에 할당합니다.**
4. **트래픽을 격리 및 제어하려는 `zone1`에서 실행 중인 모든 응용 프로그램에 대해 흐름을 만듭니다.**
5. **사이트에 적용 중인 사용 정책에 따라 흐름에 대역폭을 할당합니다.**

▼ 가상 네트워크에 대한 서비스 단계 계약을 만드는 방법

1. **서로 다른 가격으로 다양한 서비스 레벨을 제공하는 정책을 설계합니다.**

   예를 들어, 기본, 상위 및 높은 서비스 레벨을 만들고 각 레벨에 적절한 가격을 책정할 수 있습니다.
2. **월별 또는 서비스 레벨별 기준으로 고객에게 요금을 청구할 것인지, 아니면 실제 사용한 대역폭을 기준으로 요금을 청구할 것인지 결정합니다.**

   후자의 가격 책정 구조를 선택하는 경우 각 고객의 사용에 대한 통계를 수집해야 합니다.
3. **각 고객에 대한 컨테이너가 있는 가상 네트워크를 호스트에 만듭니다.**

   일반적인 구현은 각 고객에게 VNIC에서 실행되는 고유한 영역을 제공하는 것입니다.
4. **각 영역에 대한 트래픽을 격리시키는 흐름을 만듭니다.**

   영역의 모든 트래픽을 격리시키려면 영역의 VNIC에 할당된 IP 주소를 사용합니다.
5. **VNIC의 영역에 할당된 고객이 구입한 서비스 레벨을 기준으로 각 VNIC에 대역폭을 할당합니다.**

◆ ◆ ◆ 19장

19 가상 네트워크 구성(작업)

이 장에는 내부 가상 네트워크 또는 "시스템 내 네트워크"를 구성하는 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

가상 네트워크 작업 맵

이 표에서는 특정 작업의 링크를 포함하여 가상 네트워크를 구성하는 작업을 보여줍니다. 일부 작업은 가상 네트워크 시나리오에 적용되지 않습니다.

작업	설명	수행 방법
시스템에 VNIC를 만듭니다.	VNIC(가상 네트워크 인터페이스)를 하나 이상 만듭니다. VNIC는 가상 네트워크가 작성되는 의사 인터페이스입니다.	332 페이지 “가상 네트워크 인터페이스를 만드는 방법”
시스템에 etherstub을 만듭니다.	etherstub을 하나 이상 만듭니다. etherstub은 대규모 네트워크에서 격리된 개인 가상 네트워크를 만들 수 있는 가상 스위치입니다.	335 페이지 “etherstub을 만드는 방법”
VNIC를 사용할 영역을 만듭니다.	VNIC와 새 영역을 만들고 이러한 항목을 구성하여 기본 가상 네트워크를 만듭니다.	337 페이지 “VNIC에 사용할 새 영역 만들기”

작업	설명	수행 방법
VNIC를 사용할 영역을 수정합니다.	가상 네트워크가 되도록 기존 영역을 변경합니다.	342 페이지 "VNIC를 사용하도록 기존 영역의 구성 수정"
개인 가상 네트워크를 만듭니다.	etherstub 및 VNIC를 사용하여 대규모 네트워크에서 격리된 개인 네트워크를 구성합니다.	345 페이지 "개인 가상 네트워크 만들기"
VNIC를 제거합니다.	영역 자체를 삭제하지 않고 영역에 할당된 VNIC를 제거합니다.	347 페이지 "영역을 제거하지 않고 가상 네트워크를 제거하는 방법"

Oracle Solaris에서 네트워크 가상화의 구성 요소 구성

이 절에는 Oracle Solaris에서 네트워크 가상화의 빌딩 블록을 구성하는 작업이 포함되어 있습니다. 다음은 기본 구성 요소로 이루어져 있습니다.

- VNIC(가상 네트워크 인터페이스 카드)
- etherstub

*VNIC*는 데이터 링크 위에 만드는 의사 인터페이스입니다. VNIC에는 자동으로 생성된 MAC 주소가 있습니다. 사용 중인 네트워크 인터페이스에 따라 `dladm(1M)` 매뉴얼 페이지에 설명된 대로 기본 주소가 아닌 MAC 주소를 VNIC에 명시적으로 할당할 수 있습니다. 데이터 링크에 VNIC를 필요한 개수만큼 만들 수 있습니다.

*etherstub*은 시스템 관리자가 관리하는 의사 이더넷 NIC입니다. 물리적 링크 대신 etherstub에 VNIC를 만들 수 있습니다. etherstub의 VNIC는 시스템의 물리적 NIC와 독립적입니다. etherstub을 사용하면 시스템의 다른 가상 네트워크와 외부 네트워크에서 모두 격리된 개인 가상 네트워크를 생성할 수 있습니다. 예를 들어, 네트워크 전체가 아니라 회사 개발자로만 액세스가 제한되는 네트워크 환경을 만들려고 합니다. etherstub을 사용하여 이러한 환경을 만들 수 있습니다.

etherstub 및 VNIC는 Oracle Solaris 가상화 기능의 일부일 뿐입니다. 일반적으로 이러한 구성 요소는 Oracle Solaris 컨테이너 또는 영역과 함께 사용합니다. 영역에서 사용할 VNIC 또는 etherstub을 할당하여 단일 시스템에 네트워크를 만들 수 있습니다.

▼ 가상 네트워크 인터페이스를 만드는 방법

이 절차에서는 VNIC(가상 네트워크 인터페이스 카드)를 만드는 방법을 보여줍니다.

1 관리자로 전환합니다.

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

2 **(옵션) 시스템에서 사용 가능한 물리적 인터페이스에 대한 정보를 보려면 다음 명령을 입력합니다.**

```
# dladm show-phys
```

이 명령은 시스템의 물리적 NIC와 해당 데이터 링크 이름을 표시합니다. 데이터 링크의 사용자 정의 이름을 만들지 않으면 데이터 링크에 네트워크 인터페이스 장치 이름과 동일한 이름이 지정됩니다. 예를 들어, `e1000g0` 장치는 링크 이름을 다른 이름으로 바꿀 때까지 데이터 링크 이름 `e1000g0`을 사용합니다. 사용자 정의 데이터 링크 이름에 대한 자세한 내용은 24 페이지 "네트워크 장치 및 데이터 링크 이름"을 참조하십시오.

3 **(옵션) 시스템의 데이터 링크에 대한 정보를 보려면 다음 명령을 입력합니다.**

```
# dladm show-link
```

이 명령은 데이터 링크 및 현재 상태를 나열합니다. 데이터 링크의 STATE 필드에 데이터 링크가 up 상태로 표시되는지 확인합니다. 상태가 up인 데이터 링크에만 VNIC를 구성할 수 있습니다.

4 **(옵션) 구성된 인터페이스에 대한 IP 주소 정보를 보려면 다음 명령을 입력합니다.**

```
# ipadm show-addr
```

이 명령은 해당 IP 주소를 포함하여 시스템에 구성된 인터페이스를 나열합니다.

5 **데이터 링크에 VNIC를 만듭니다.**

```
# dladm create-vnic -l link vnic
```

- *link*는 VNIC가 구성되어 있는 데이터 링크의 이름입니다.
- *vnic*는 사용자 정의 이름으로도 레이블을 지정할 수 있는 VNIC입니다.

6 **링크에 VNIC IP 인터페이스를 만듭니다.**

```
# ipadm create-ip vnic
```

7 **유효한 IP 주소로 VNIC를 구성합니다.**

정적 IP 주소를 할당하는 경우 다음 구문을 사용합니다.

```
# ipadm create-addr -T static -a address addrobj
```

여기서 *addrobj*는 명명 형식 *interface/user-defined-string*(예: `e1000g0/v4globalz`)을 사용합니다. 이 명령을 사용하는 경우의 기타 옵션은 `ipadm(1M)` 매뉴얼 페이지를 참조하십시오.

8 **정적 IP 주소를 사용하는 경우 `/etc/hosts` 파일에 주소 정보를 추가합니다.**

9 **(옵션) VNIC의 주소 구성을 표시하려면 다음을 입력합니다.**

```
# ipadm show-addr
```

10 **(옵션) VNIC 정보를 표시하려면 다음을 입력합니다.**

```
# dladm show-vnic
```

예 19–1 가상 네트워크 인터페이스 만들기

이 예에는 VNIC를 만드는 명령이 포함되어 있습니다. 명령을 실행하려면 수퍼유저 또는 이와 동등한 역할로 시스템에 로그인해야 합니다.

```
# dladm show-phys
LINK      MEDIA               STATE      SPEED DUPLEX   DEVICE
net0      Ethernet            up         1000 full      e1000g0
net1      Ethernet            unknown    0    half      e1000g1

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

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

# ipadm show-addr
ADDROBJ        TYPE       STATE     ADDR
lo0/?          static     ok        127.0.0.1/8
net0/v4addr    static     ok        192.168.3.70/24

# dladm create-vnic -l net0 vnic0
# dladm create-vnic -l net0 vnic1

# dladm show-vnic
LINK        OVER          SPEED  MACADDRESS         MACADDRTYPE
vnic0       net0      1000 Mbps  2:8:20:c2:39:38    random
vnic1       net0      1000 Mbps  2:8:20:5f:84:ff    random
#
# ipadm create-ip vnic0
# ipadm create-ip vnic1

# ipadm create-addr -T static -a 192.168.3.80/24 vnic0/v4address
# ipadm create-addr -T static -a 192.168.3.85/24 vnic1/v4address
# ipadm show-addr
ADDROBJ            TYPE       STATE     ADDR
lo0/?              static     ok        127.0.0.1/8
net0/v4addr        static     ok        192.168.3.70/24
vnic0/v4address    static     ok        192.168.3.80/24
vnic1/v4address    static     ok        192.168.3.85/24
```

시스템의 `/etc/hosts` 파일에는 다음과 유사한 정보가 들어 있습니다.

```
# cat /etc/hosts
#
::1             localhost
127.0.0.1       localhost
192.168.3.70    loghost   #For e1000g0
192.168.3.80    vnic1
192.168.3.85    vnic2
```

▼ etherstub을 만드는 방법

etherstub을 사용하여 시스템의 나머지 가상 네트워크와 시스템이 연결된 외부 네트워크에서 가상 네트워크를 격리시킵니다. etherstub만 단독으로 사용할 수는 없습니다. 대신 VNIC를 etherstub과 함께 사용하여 가상 사설망이나 격리된 가상 네트워크를 만듭니다. etherstub을 필요한 개수만큼 만들 수 있습니다. 각 etherstub에 VNIC를 필요한 개수만큼 만들 수도 있습니다.

1 관리자로 전환합니다.

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

2 etherstub을 만듭니다.

```
# dladm create-etherstub etherstub
```

3 etherstub에 VNIC를 만듭니다.

```
# dladm create-vnic -l etherstub vnic
```

4 개인 주소로 VNIC를 구성합니다.

주 – etherstub에 VNIC를 구성 중인 네트워크를 격리시키려면 외부 네트워크의 기본 라우터에서 전달할 수 없는 개인 IP 주소를 사용해야 합니다. 예를 들어, 물리적 인터페이스의 주소가 시스템이 `192.168.3.x` 네트워크에 있음을 나타내는 `192.168.3.0/24`라고 가정합니다. 따라서 기본 라우터에 알려지지 않은 다른 주소(예: `192.168.0.x`)를 할당합니다.

---

5 (옵션) VNIC에 대한 정보를 표시하려면 다음 명령을 입력합니다.

```
# dladm show-vnic
```

이 명령은 시스템의 모든 VNIC와 VNIC가 생성된 데이터 링크 또는 etherstub을 나열합니다.

6 (옵션) 시스템의 모든 물리적 링크 및 가상 링크에 대한 정보를 표시하려면 다음 명령을 입력합니다.

```
# dladm show-link
```

예 19–2 etherstub 만들기

다음 예에서는 etherstub을 만들고 etherstub에 VNIC를 구성하는 방법을 보여줍니다. 이 예에서는 etherstub에 구성된 세번째 VNIC를 추가하여 이전 예를 개발합니다.

다음 명령을 실행하려면 수퍼유저 또는 이와 동등한 역할로 시스템에 로그인해야 합니다.

```
# dladm create-etherstub stub0
#
dladm show-vnic
LINK        OVER          SPEED  MACADDRESS         MACADDRTYPE
vnic1       net9      1000 Mbps  2:8:20:c2:39:38    random
vnic2       net0      1000 Mbps  2:8:20:5f:84:ff    random
#
# dladm create-vnic -l stub0 vnic3
# ipadm create-vnic vnic3
# ipadm create-addr -T static -a 192.168.0.10/24 vnic3/privaddr
#
# dladm show-vnic
LINK        OVER          SPEED  MACADDRESS         MACADDRTYPE
vnic1       net0      1000 Mbps  2:8:20:c2:39:38    random
vnic2       net0      1000 Mbps  2:8:20:5f:84:ff    random
vnic3       stub0        1000 Mbps  2:8:20:54:f4:74    random
#
# ipadm show-addr
ADDROBJ            TYPE       STATE     ADDR
lo0/?              static     ok        127.0.0.1/8
net0/v4addr        static     ok        192.168.3.70/24
vnic1/v4address    static     ok        192.168.3.80/24
vnic2/v4address    static     ok        192.168.3.85/24
vnic3/privaddr     static     ok        192.168.0.10/24
```

시스템의 `/etc/hosts` 파일에는 다음과 유사한 정보가 들어 있습니다.

```
# cat /etc/hosts
#
::1             localhost
127.0.0.1       localhost
192.168.3.70    loghost   #For e1000g0
192.168.3.80    vnic1
192.168.3.85    vnic2
192.168.0.10    vnic3
```

VNIC 및 영역 작업

이 절에서는 영역에서 사용되도록 네트워크 가상화 구성 요소를 구성하여 이러한 구성 요소를 배포하는 방법을 보여줍니다. 또한 영역에서 VNIC를 사용하도록 하는 경우 다음 두 가지 방법을 제공합니다.

- 완전히 새로운 영역을 만들고 이러한 영역에서 VNIC 구성
- VNIC를 사용하도록 기존 영역 구성 수정

시스템에 처음 로그인하면 자동으로 **전역 영역**에 있습니다. 전역 영역에 VNIC를 만듭니다. 그런 다음 전역 영역 또는 비전역 배타적 유형 영역에서 사용할 것인지에 따라 이러한 VNIC를 추가로 구성합니다. 영역에 대한 소개는 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 "영역 개요"를 참조하십시오.

VNIC에 사용할 새 영역 만들기

구성된 영역이 시스템에 없는 경우 또는 VNIC를 사용할 새 영역을 만들려는 경우 이 방법을 사용합니다.

VNIC를 사용하려면 영역을 배타적 IP 영역으로 구성해야 합니다. 이후 단계에서는 `vnic1`을 사용하여 `zone1`을 구성합니다. 동일한 단계를 수행하여 `zone2`를 구성해야 합니다. 알아보기 쉽도록 프롬프트에서 특정 명령이 실행되는 영역을 나타냅니다. 하지만 프롬프트에 표시되는 실제 경로가 특정 시스템의 프롬프트 설정에 따라 다를 수도 있습니다.

▼ 배타적 IP 영역을 만들고 구성하는 방법

영역을 만들 때 여러 매개변수를 설정할 수 있습니다. 이 장 전체에 나오는 영역 절차는 영역이 VNIC로 작동하도록 하는 매개변수에만 중점을 둡니다. 영역 구성에 대한 자세한 내용은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 제II부, "Oracle Solaris Zones"을 참조하십시오.

시작하기 전에 다음 작업을 수행했는지 확인합니다.

- 332 페이지 "가상 네트워크 인터페이스를 만드는 방법"에 설명된 대로 영역에 대한 VNIC를 생성했습니다.
- 영역 이름을 정의했습니다.
- 영역 홈 디렉토리를 결정했습니다.
- 특정 영역에 연결될 특정 VNIC를 결정했습니다.
- VNIC의 IP 주소를 결정했습니다.
- 영역에 제공할 라우터 주소와 같은 기타 네트워크 정보를 가져왔습니다.

1 관리자로 전환합니다.

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

2 만드는 각 영역에 대해 다음 단계를 수행합니다.

a. 영역 구성 유틸리티를 시작하고 영역을 만듭니다.

```
global# zonecfg -z zone
zonecfg:zone> create
```

b. `zonepath` 매개변수를 정의하여 홈 디렉토리를 설정합니다.

```
zonecfg:zone> set zonepath=/home/export/zone
```

c. 자동 부트를 사용으로 설정합니다.

```
zonecfg:zone> set autoboot=true
```

d. 영역을 배타적 IP 영역으로 구성합니다.

```
zonecfg:zone> set ip-type=exclusive
```

e. 영역의 인터페이스를 지정된 VNIC로 설정합니다.

```
zonecfg:zone> add net
zonecfg:zone:net> set physical=vnic
zonecfg:zone:net> end
zonecfg:zone>
```

f. 설정을 확인하고 커밋한 다음 영역 구성 유틸리티를 종료합니다.

```
zonecfg:zone>verify
zonecfg:zone> commit
zonecfg:zone> exit
global#
```

g. (옵션) 영역에 대한 정보가 올바른지 확인하려면 다음을 입력합니다.

```
global# zonecfg -z zone info
```

주 – 다음을 입력하여 영역 구성 유틸리티를 실행하는 동안 동일한 정보를 표시할 수 있습니다.

```
zonecfg:zone> info
```

---

3 영역을 설치합니다.

```
global# zoneadm -z zone install
```

주 – 설치 프로세스에 오랜 시간이 걸릴 수 있습니다.

---

4 (옵션) 영역이 완전히 설치된 후 영역의 상태를 확인합니다.

```
zoneadm list -iv
```

주 – `-iv` 옵션은 실행 여부에 관계없이 구성된 모든 영역을 나열합니다. 이 단계에서는 방금 만든 영역의 상태가 "`running`"이 아니라 "`installed`"입니다. `-v` 옵션을 사용하면 실행 중인 영역만 나열되고 방금 만든 영역은 제외됩니다.

---

5 영역을 시작합니다.

```
global# zoneadm -z zone boot
```

6 (옵션) 이제 영역이 실행되고 있는지 확인합니다.

```
global# zoneadm list -v
```

7 영역이 완전히 부트된 후 영역의 콘솔에 연결합니다.

```
# zlogin -C zone
```

8 메시지가 표시되면 정보를 제공합니다.

정보 중 일부는 터미널 유형, 지역, 언어 등입니다. 대부분의 정보는 선택 항목 목록에서 선택하여 제공합니다. 일반적으로 시스템 구성에 다른 옵션이 필요하지 않은 경우 기본 옵션으로 충분합니다.

다음 정보는 제공하거나 확인해야 하는 현재 절차와 관련된 항목입니다.

- 영역의 호스트 이름(예: `zone1`)
- 영역 VNIC의 IP 주소를 기반으로 하는 영역의 IP 주소
- IPv6이 사용으로 설정되었는지 여부
- 가상 네트워크가 있는 시스템이 서브넷의 일부인지 여부
- IP 주소의 넷마스크
- 가상 네트워크가 작성된 물리적 인터페이스의 IP 주소일 수 있는 기본 경로

영역에 대한 필수 정보를 제공하면 영역이 다시 시작됩니다.

예 19–3 영역과 VNIC를 만들어 기본 가상 네트워크 구성

이 예에서는 영역 및 VNIC를 만들기 위해 제공된 위의 모든 단계를 통합하여 가상 네트워크를 구성합니다. `zone1`이 예의 샘플 영역으로 사용됩니다.

이 예는 다음 가정을 기반으로 합니다.

- VNIC: `vnic1`
- 영역 이름: `zone1`
- 영역 홈 디렉토리: `/home/export/`*zone-name*
- VNIC 영역 할당: `zone1`의 경우 `vnic1`
- IP 주소: `vnic1`은 `192.168.3.80` 사용
- 물리적 인터페이스 IP 주소: `192.168.3.70`
- 라우터 주소: `192.168.3.25`

```
global# dladm show-phys
LINK    MEDIA       STATE       SPEED  DUPLEX    DEVICE
net0    Ethernet    up          1000   full      e1000g0
net1    Ethernet    unknown     1000   full      bge0

global# dladm show-lnk
LINK       CLASS     MTU    STATE     BRIDGE   OVER
net0    phys      1500   up        --       --
net1       phys      1500   unknown   --       --

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

global # ipadm show-addr
ADDROBJ         TYPE        STATE     ADDR
lo0/?           static      ok        127.0.0.1/8
net0/v4addr     static      ok        192.168.3.70/24
```

```
global # dladm create-vnic -l net0 vnic1

global # dladm show-vnic
LINK      OVER         SPEED        MACADDRESS      MACADDRTYPE
vnic1     net0     1000 Mbps    2:8:20:5f:84:ff     random

global # ipadm create-ip vnic1
global # ipadm create-addr -T static -a 192.168.3.80/24 vnic1/v4address
global # ipadm show-addr
ADDROBJ           TYPE       STATE     ADDR
lo0/?             static     ok        127.0.0.1/8
net0/v4addr       static     ok        192.168.3.70/24
vnic1/v4address   static     ok        192.168.3.80/24

global # cat /etc/hosts
::1             localhost
127.0.0.1       localhost
192.168.3.70    loghost   #For net0
192.168.3.80    zone1  #using vnic1

global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> seet autoboot=true
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify

zonecfg:zone1> info
zonename: zone1
zonepath: /export/home/zone1
brand:    native
autoboot: true
net:
          address not specified
          physical: vnic1

zonecfg:zone1> commit
zonecfg:zone1> exit
global#
global# zoneadm -z zone1 verify
WARNING: /export/home/zone1 does not exist, so it could not be verified.
When 'zoneadm install' is run, 'install' will try to create
/export/home/zone1, and 'verify' will be tried again,
but the 'verify' may fail if:
the parent directory of /export/home/zone1 is group- or other-writable
or
/export/home/zone1 overlaps with any other installed zones.

global# zoneadm -z zone1 install
Preparing to install zone <zone1>
Creating list of files to copy from the global zone.
.
.
Zone <zone1> is initialized.

global# zoneadm list -iv
```

```
ID NAME     STATUS      PATH                 BRAND     IP
0  global   running     /                    native   shared
-  zone1    installed   /export/home/zone1   native   excl

global# zoneadm -z zone1 boot
global# zoneadm list -v
ID NAME     STATUS    PATH                 BRAND     IP
0  global   running   /                    native   shared
1  zone1    running   /export/home/zone1   native   excl

zlogin -C zone1
What type of terminal are you using?
.
.
.
8) Sun Workstation
9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (dtterm)
14) Other
Type the number of your choice and press Return: 13
.
(More prompts)
..
```

메시지가 표시되면 정보를 제공합니다. 네트워크 정보에 대해 다음을 제공합니다.

```
Hostname: zone1
IP address: 192.168.3.80
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default route: 192.168.3.70
Router IP address: 192.168.3.25
```

다음 순서 여러 도구를 사용하여 네트워크 트래픽을 관찰하고 영역 사용에 대한 통계를 수집할 수 있습니다.

- 네트워크가 제대로 구성되었는지 확인하려면 **Oracle Solaris 관리: IP 서비스**의 5 장, “TCP/IP 네트워크 관리”를 참조하십시오.
- 네트워크를 통해 트래픽을 관찰하려면 **Oracle Solaris 관리: IP 서비스**의 “snoop 명령으로 패킷 전송 모니터링”을 참조하십시오.
- 네트워크에서 시스템 리소스를 사용하는 방법을 관리하려면 21 장, “네트워크 리소스 관리”를 참조하십시오.
- 회계용으로 통계를 얻으려면 22 장, “네트워크 트래픽 및 리소스 사용 모니터링”을 참조하십시오.

가상 네트워크를 역어셈블해야 하는 경우 347 페이지 “영역을 제거하지 않고 가상 네트워크를 제거하는 방법”을 참조하십시오.

VNIC를 사용하도록 기존 영역의 구성 수정

기존 영역에서 VNIC를 사용하려면 이 방법을 사용합니다. 이 경우 영역에 이미 영역 이름과 홈 디렉토리가 있거나 zonepaths가 이미 정의되었습니다.

▼ VNIC를 사용하도록 영역을 재구성하는 방법

시작하기 전에 다음 작업을 수행했는지 확인합니다.

- 332 페이지 "가상 네트워크 인터페이스를 만드는 방법"에 설명된 대로 영역에 대한 VNIC를 생성했습니다.
- 특정 영역에 연결될 특정 VNIC를 결정했습니다.
- VNIC의 IP 주소를 결정했습니다.
- 영역에 제공할 라우터 주소와 같은 기타 네트워크 정보를 가져왔습니다.

1 관리자로 전환합니다.

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

2 영역이 시스템에서 올바르게 구성되고 실행되고 있는지 확인합니다.

```
global# zoneadm list -v
```

주 – -v 옵션은 실행 중인 영역만 나열합니다. 시작되지 않은 영역을 포함하여 구성된 모든 영역을 나열하려면 -iv 옵션을 사용합니다.

---

3 VNIC로 구성하려는 각 영역에 대해 다음 단계를 수행합니다.

a. 영역에 대한 정보를 확인합니다.

```
global# zonecfg -z zone info
```

IP 유형 및 네트워크 인터페이스에 대한 정보를 확인합니다. 네트워크 인터페이스는 *physical* 매개변수로 지정됩니다. VNIC로 영역을 구성하려면 영역이 배타적 IP 영역이어야 하며 네트워크 인터페이스가 VNIC를 지정해야 합니다.

b. 필요한 경우 공유 영역을 배타적 IP 영역으로 변경합니다.

```
global# zonecfg -z zone
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1>
```

c. VNIC를 사용하도록 영역의 인터페이스를 변경합니다.

```
zonecfg:zone1> remove net physical=non-vnic-interface
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic
zonecfg:zone1:net> end
zonecfg:zone1>
```

d. 다른 매개변수 값을 적절하게 변경합니다.

e. 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:zone1 verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global#
```

f. 영역을 재부트합니다.

```
global# zoneadm -z zone reboot
```

g. 영역이 재부트된 후 `ip-type` 및 `physical`에 대한 영역 정보가 올바른지 확인합니다.

```
global# zonecfg -z zone info ip-type
global# zonecfg -z zone info net
```

이 정보에 영역의 IP 유형이 배타적이며 지정된 VNIC를 사용한다고 표시되어야 합니다.

4 영역에 로그인합니다.

```
global# zlogin zone
```

5 유효한 IP 주소로 VNIC를 구성합니다.

VNIC에 정적 주소를 할당하는 경우 다음을 입력합니다.

zone# **ipadm create-addr -T static -a** *address addrobj*

여기서 *address*는 CIDR 표기법을 사용할 수 있는 반면 *addrobj*는 이름 지정 규약 *interface*/*user-defined-string*을 따릅니다.

6 (옵션) 영역 내의 인터페이스 구성을 확인합니다.

zone# **ipadm show-if**

또는

zone# **ipadm show-addr**

예 19–4 VNIC를 사용하도록 영역 구성을 수정하여 기본 가상 네트워크 구성

이 예에서는 위의 예와 동일한 시스템을 사용하며 동일한 가정으로 작업합니다. 이 시스템에 `zone2`가 공유 영역으로 이미 존재한다고 가정합니다. `vnic2`를 사용하도록 `zone2`를 수정하려고 합니다.

```
global# dladm show-link
LINK    CLASS     MTU    STATE     BRIDGE   OVER
net0    phys      1500   up        --       --
net1    phys      1500   unknown   --       --
vnic1   vnic      1500   up        --       e1000g0

global# ipadm show-if
```

```
IFNAME   CLASS        STATE     ACTIVE     OVER
lo0      loopback     ok        yes        --
net0     ip           ok        yes        --
vnic1    ip           ok        yes        --

global # ipadm show-addr
ADDROBJ           TYPE       STATE     ADDR
lo0/?             static     ok        127.0.0.1/8
net0/v4addr       static     ok        192.168.3.70/24
vnic1/v4address   static     ok        192.168.3.80/24

global # dladm create-vnic -l net0 vnic2
global # dladm show-vnic
LINK      OVER        SPEED        MACADDRESS          MACADDRTYPE
vnic1     net0     1000 Mbps    2:8:20:5f:84:ff     random
vnic2     net0     1000 Mbps    2:8:20:54:f4:74     random

global# zoneadm list -v
ID NAME     STATUS    PATH                  BRAND     IP
0  global   running   /                     native   shared
1  zone1    running   /export/home/zone1    native   excl
2  zone2    running   /export/home/zone2    native   shared

global# zonecfg -z zone2 info
zonename: zone2
zonepath: /export/home/zone2
brand: native
autoboot: true
bootargs:
pool: z2-pool
limitpriv:
scheduling-class:
ip-type: shared
hostid:
inherit-pkg-dir:
        dir: /lib
inherit-pkg-dir:
        dir: /platform
inherit-pkg-dir:
        dir: /sbin
inherit-pkg-dir:
        dir: /usr
inherit-pkg-dir:
        dir: /etc/crypto
net:
        address not specified
        physical: e1000g0
        defrouter not specified
global#


global# zonecfg -z zone2
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> remove net physical=net0
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic2
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
```

```
zonecfg:zone1> exit
global#

global# zonecfg -z zone2 info ip-type
ip-type: exclusive
global#

global# zonecfg -z zone2 info net
net:
        address ot specified
        physical: vnic2
        defrouter not specified
global#

global# zlogin zone2
zone2# ipadm create-ip vnic2
zone2# ipadm create-addr -T static -a 192.168.3.85/24 vnic2/v4address

zone2# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
vnic2/v4address   static   ok           192.168.3.85/24

zone1# exit
global#

global# vi /etc/hosts
#
::1             localhost
127.0.0.1       localhost
192.168.3.70    loghost   #For e1000g0
192.168.3.80    zone1   #using vnic1
192.168.3.85    zone2   #using vnic2
```

다음 순서 네트워크 설정을 추가로 구성하여 시스템 리소스 사용을 사용자 정의하거나, 여러 도구를 사용하여 네트워크 트래픽을 관찰하고 리소스 사용에 대한 통계를 수집할 수 있습니다.

- 네트워크가 올바르게 구성되었는지 확인하려면 다음을 참조하십시오.
- 네트워크의 트래픽을 관찰하려면 다음을 참조하십시오.
- 네트워크가 시스템 리소스를 사용하는 방식을 관리하려면 다음을 참조하십시오.
- 계정 용도로 통계를 가져오려면 다음을 참조하십시오.

가상 네트워크를 역어셈블해야 하는 경우 347 페이지 “영역을 제거하지 않고 가상 네트워크를 제거하는 방법”을 참조하십시오.

개인 가상 네트워크 만들기

이 절의 예에서는 단일 시스템에 **개인 가상 네트워크**를 구성하는 방법을 보여줍니다. 개인 가상 네트워크는 VPN(가상 사설망)과는 다릅니다. VPN 소프트웨어는 두 끝점 시스템 간에 보안 P2P 연결을 만듭니다. 이 절의 작업에서 구성된 개인 네트워크는 외부 시스템에서 액세스할 수 없는 시스템의 가상 네트워크입니다.

개인 네트워크의 영역에서 호스트를 벗어나서 패킷을 보낼 수 있게 하려면 NAT(Network Address Translation) 장치를 구성합니다. NAT는 VNIC의 개인 IP 주소를 물리적 네트워크 인터페이스의 경로 지정 가능 IP 주소로 변환하지만 개인 IP 주소를 외부 네트워크에 노출하지 않습니다. 다음 예에는 경로 지정 구성도 포함되어 있습니다.

예 19–5 개인 가상 네트워크 구성 만들기

다음 예에서는 위의 예와 동일한 시스템을 사용하며 동일한 가정으로 진행합니다. 구체적으로, zone1과 zone2가 이제 가상 네트워크로 구성됩니다. zone3이 시스템에 이미 있다고 가정합니다. 네트워크의 나머지 부분에서 개인 네트워크가 격리되도록 zone3을 수정합니다. 그런 다음 가상 개인 네트워크가 호스트 외부로 패킷을 보낼 수 있도록 NAT 및 IP 전달을 구성하지만 개인 주소를 외부 네트워크로부터 숨깁니다.

```
global# dladm create-etherstub stub0

global# dladm create-vnic -l etherstub0 vnic3
global# dladm show-vnic
LINK      OVER        SPEED        MACADDRESS          MACADDRTYPE
vnic1     net0     1000 Mbps    2:8:20:5f:84:ff     random
vnic2     net0     1000 Mbps    2:8:20:54:f4:74     random
vnic3     stub0          0 Mbps    2:8:20:6b:8:ab      random

global# vi /etc/hosts
#
::1             localhost
127.0.0.1       localhost
192.168.3.70    loghost   #For e1000g0
192.168.3.80    zone1   #using vnic1
192.168.3.85    zone2   #using vnic2
```

이 단계에서 zone3이 vnic3에서 배타적 IP 영역이 되도록 수정합니다.

```
global# zonecfg -z zone3
zonecfg:zone3> set ip-type=exclusive
zonecfg:zone3> remove net physical=e1000g0
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic3
zonecfg:zone3:net> end
zonecfg:zone3> vereify
zonecfg:zone3> commit
zonecfg:zone3> exit
global#

global# zonecfg -z zone3 info ip-type
ip-type: exclusive
global#

global# zonecfg -z zone3 info net
net:
        address ot specified
        physical: vnic3
        defrouter not specified
global#

global# zlogin zone3
```

예 19–5 개인 가상 네트워크 구성 만들기 *(계속)*

```
zone3# ipadm create-ip vnic3
zone3# ipadm create-addr -T static -a 192.168.0.10/24 vnic3/privaddr

zone3# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
vnic3/privaddr    static   ok           192.168.0.10/24
zone3# exit

global# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4addr       static   ok           192.168.3.70/24
vnic1/v4address   static   ok           192.168.3.80/24
vnic2/v4address   static   ok           192.168.3.85/24
vnic3/privaddr    static   ok           192.168.0.10/24

global# vi /etc/hosts
::1             localhost
127.0.0.1       localhost
192.168.3.70    loghost   #For e1000g0
192.168.3.80    zone1   #using vnic1
192.168.3.85    zone2   #using vnic2
192.168.0.10    zone3   #using vnic3

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

           Routing services   "route:default ripng:default"

global# ipadm set-ifprop -p forwarding=yes -m ipv4 e1000g0

global# vi /etc/ipf/ipnat.conf
map e1000g0 192.168.0.0/24 -> 0/32  portmap tcp/udp auto
map e1000g0 192.168.0.0/24 -> 0/32

global# svcadm enable network/ipfilter

global# zoneadm -z zone1 boot
global# zoneadm -z zone2 boot
global# zoneadm -z zone3 boot
```

▼ 영역을 제거하지 않고 가상 네트워크를 제거하는 방법

다음 절차에서는 영역의 가상 네트워크를 사용 안함으로 설정하지만 영역을 그대로 유지하는 방법을 보여줍니다.

다음 중 하나를 수행해야 하는 경우 이 절차를 사용합니다.

- 다른 구성에 기존 영역을 사용합니다. 예를 들어, etherstub을 사용하여 영역을 만들어야 하는 개인 네트워크의 일부로 영역을 구성해야 할 수도 있습니다.
- 영역을 다른 네트워크로 마이그레이션합니다.
- 영역을 다른 영역 경로로 이동합니다.
- **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 "동일한 시스템에서 비전역 영역 복제"에 설명된 대로 영역을 복제합니다.

시작하기 전에 이 작업에서는 배타적 IP 영역으로 구성된 실행 중인 가상 네트워크가 있다고 가정합니다.

1 관리자로 전환합니다.

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

2 현재 구성된 영역의 상태를 확인합니다.

```
# zoneadm list -v
```

다음과 유사한 정보가 표시됩니다.

```
ID  NAME     STATUS       PATH                          BRAND           IP
 0  global   running      /                             native          shared
 1  zone1    running      /export/home/zone1            native          excl
 2  zone2    running      /export/home/zone2            native          excl
 3  zone3    running      /export/home/zone3            native          excl
```

3 가상 네트워크의 배타적 IP 영역을 중지합니다.

중지할 각 영역에 대해 다음 명령을 개별적으로 실행합니다.

```
global# zoneadm -z zone-name halt
```

영역을 중지하는 경우 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 "영역 정지"에 설명된 대로 영역의 응용 프로그램 환경을 제거하고 많은 시스템 작업을 종료합니다.

4 영역이 중지되었는지 확인합니다.

```
# zoneadm list -iv
ID NAME             STATUS     PATH                           BRAND    IP
   0 global           running    /                              native   shared
   - zone1            installed  /export/home/zone1             native   excl
   - zone2            installed  /export/home/zone2             native   excl
   - zone3            installed  /export/home/zone3             native   excl
```

영역이 설치된 상태이지만 더 이상 실행되지 않습니다. 중지된 영역을 재부트하려면 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 "영역 부트 방법"을 참조하십시오.

5 중지된 영역에 대해 구성된 VNIC를 나열합니다.

```
# dladm show-vnic
LINK        OVER              SPEED  MACADDRESS         MACADDRTYPE
vnic1       net0       1000 Mbps  2:8:20:5f:84:ff    random
vnic2       net1      1000 Mbps  2:8:20:54:f4:74    random
vnic3       stub0     1000 MBps  2:8:20:c2:39:38    random
```

VNIC가 전역 영역에 데이터 링크로 여전히 구성되어 있다고 결과 출력 결과에 표시됩니다. 하지만 해당 IP 인터페이스는 전역 영역이 아니라 이러한 VNIC가 연결된 영역에서 생성되고 사용으로 설정되었습니다. 이러한 비전역 영역이 이제 중지되었습니다.

6 VNIC를 삭제합니다.

```
# dladm delete-vnic vnic
```

예를 들어, 다음을 입력하여 그림 18–1의 영역에서 VNIC를 삭제합니다.

```
# dladm delete-vnic vnic1
# dladm delete-vnic vnic2
```

◆ ◆ ◆ 20장

20 가상화된 환경에서 링크 보호 사용

이 장에서는 링크 보호 및 Oracle Solaris 시스템에서 링크 보호를 구성하는 방법에 대해 설명합니다. 이 장에서는 다음 항목을 다룹니다.

링크 보호 개요

시스템 구성에서 가상화의 채택이 증가함에 따라 호스트 관리자가 게스트 VM(가상 시스템)에 물리적 또는 가상 링크에 대한 배타적 액세스 권한을 부여할 수 있습니다. 이 구성은 가상 환경의 네트워크 트래픽이 호스트 시스템에서 보내거나 받는 광범위한 트래픽으로부터 격리되도록 하여 네트워크 성능을 향상시킵니다. 동시에 이 구성은 게스트 환경에서의 유해한 패킷 생성 위험에 시스템 및 전체 네트워크를 노출시킬 수 있습니다.

링크 보호는 잠재적 악성 게스트 VM이 네트워크에 초래할 수 있는 손상을 방지하기 위한 것입니다. 이 기능은 다음과 같은 기본 위협으로부터 보호합니다.

- IP 및 MAC 스푸핑
- BPDU(Bridge Protocol Data Unit) 공격과 같은 L2 프레임 스푸핑

주 – 특히 더 복잡한 필터링 요구 사항이 있는 구성의 경우 링크 보호로 방화벽 배포를 대체해서는 안됩니다.

링크 보호 유형

링크 보호 방식은 기본적으로 사용 안함으로 설정됩니다. 링크 보호를 사용으로 설정하려면 다음 보호 유형 중 하나 이상을 `protection` 링크 등록 정보의 값으로 지정합니다.

`mac-nospoof` MAC 스푸핑에 대한 보호를 사용으로 설정합니다. 아웃바운드 패킷의 소스 MAC 주소가 데이터 링크에 구성된 MAC 주소와 일치해야 합니다. 그렇지 않으면 패킷이 삭제됩니다. 링크가 영역에 속하는 경우 `mac-nospoof`를 사용으로 설정하면 영역의 소유자가 해당 링크의 MAC 주소를 수정할 수 없습니다.

`ip-nospoof` IP 스푸핑에 대한 보호를 사용으로 설정합니다. 송신 IP, ARP 또는 NDP 패킷에는 DHCP 구성 IP 주소나 `allowed-ips` 링크 등록 정보에 나열된 주소 중 하나와 일치하는 주소 필드가 있어야 합니다. 그렇지 않으면 패킷이 삭제됩니다.

`allowed-ips` 링크 등록 정보는 `ip-nospoof` 보호 유형으로 작동합니다. 기본적으로 이 등록 정보에 지정된 목록은 비어 있습니다. 등록 정보가 비어 있거나 구성되지 않은 경우 다음 IP 주소가 등록 정보에 암시적으로 포함됩니다. 이러한 IP 주소를 송신 패킷의 IP 주소와 일치시켜 패킷이 통과할 수 있는지 또는 삭제되는지를 확인합니다.

- 동적으로 학습된 DHCP 구성 IPv4 또는 IPv6 주소
- 링크의 MAC 주소에서 파생되고 RFC 2464를 준수하는 링크 로컬 IPv6 주소

다음 목록에서는 `allowed-ips` 등록 정보의 주소와 일치해야 하는 프로토콜 및 해당 송신 패킷의 연결된 주소 필드를 보여줍니다. 이 등록 정보가 비어 있으면 패킷의 주소가 DHCP 구성 IP 주소와 일치해야 합니다.

- IP(IPv4 또는 IPv6) - 패킷의 소스 주소
- ARP - 패킷의 보낸 사람 프로토콜 주소

`restricted` 송신 패킷을 IPv4, IPv6 및 ARP 프로토콜 유형의 패킷으로만 제한합니다. 나열된 유형이 아닌 기타 패킷은 삭제됩니다. 이 보호 유형을 사용하면 링크가 잠재적으로 유해한 L2 제어 프레임을 생성하지 않습니다.

주 – 링크 보호로 인해 삭제되는 패킷은 커널 통계 `mac_spoofed`, `ip_spoofed` 및 `restricted`에 의해 추적됩니다. 이러한 통계는 세 가지 보호 유형에 해당합니다. `kstat` 명령을 사용하면 이러한 링크별 통계를 검색할 수 있습니다. 통계 검색에 대한 자세한 내용은 `kstat(1M)` 매뉴얼 페이지를 참조하십시오.

링크 보호 구성(작업 맵)

링크 보호를 사용하려면 dladm 명령의 옵션 중 하나를 사용하여 링크 등록 정보를 설정합니다. 보호 유형이 다른 구성 파일과 함께 작동하는 경우(예: ip-nospoof가 allowed-ips와 함께 작동) 두 가지 일반적인 작업을 수행합니다. 먼저 링크 보호를 사용으로 설정합니다. 그런 다음 구성 파일을 사용자 정의하여 링크 보호의 작동 방식을 결정합니다.

주 – 전역 영역에서 링크 보호를 구성해야 합니다.

다음은 Oracle Solaris 서버에서 링크 보호를 구성하는 데 사용할 수 있는 작업을 보여줍니다.

작업	설명	수행 방법
링크 보호 방식을 사용으로 설정합니다.	dladm set-linkprop 명령을 사용하여 링크에 대해 링크 보호 유형을 사용으로 설정합니다.	353 페이지 "링크 보호 방식을 사용으로 설정하는 방법"
링크 보호 방식을 사용 안함으로 설정합니다.	dladm reset-linkprop 명령을 사용하여 링크 보호를 사용 안함으로 설정합니다.	354 페이지 "링크 보호를 사용 안함으로 설정하는 방법"
IP 링크 보호 유형을 사용자 정의합니다.	dladm set-linkprop 명령을 사용하여 allowed-ips 등록 정보의 값을 구성하거나 수정합니다.	354 페이지 "IP 스푸핑에 대한 보호를 위해 IP 주소를 지정하는 방법"
링크 보호 구성을 확인합니다.	dladm show-linkprop 명령을 통해 protection 및 allowed-ips 등록 정보 이름을 지정하여 링크 보호 구성을 확인합니다.	355 페이지 "링크 보호 구성을 확인하는 방법"

▼ 링크 보호 방식을 사용으로 설정하는 방법

이 절차에서는 링크 보호 유형 mac-nospoof, ip-nospoof 및 restricted 중 하나 이상을 사용으로 설정합니다.

1 **관리자로 전환합니다.**

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

2 **하나 이상의 보호 유형을 지정하여 링크 보호를 사용으로 설정합니다.**

```
# dladm set-linkprop -p protection=value[,value,...] link
```

다음 예에서는 vnic0 링크의 세 가지 링크 보호 유형이 모두 사용으로 설정됩니다.

```
# dladm set-linkprop -p protection=mac-nospoof,ip-nospoof,restricted vnic0
```

▼ 링크 보호를 사용 안함으로 설정하는 방법

이 절차에서는 링크 보호를 사용 안함으로 설정하는 기본값으로 링크 보호를 재설정합니다.

1 **관리자로 전환합니다.**

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

2 **protection 등록 정보를 기본값으로 재설정하여 링크 보호를 사용 안함으로 설정합니다.**

```
# dladm reset-linkprop -p protection link
```

▼ IP 스푸핑에 대한 보호를 위해 IP 주소를 지정하는 방법

allowed-ips 등록 정보는 protection 등록 정보가 ip-nospoof 보호 유형을 사용으로 설정하는 경우에만 사용됩니다.

1 **관리자로 전환합니다.**

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

2 **IP 스푸핑으로부터 보호를 사용으로 설정했는지 확인합니다.**

이 링크 보호 유형을 사용으로 설정하지 않은 경우 다음 명령을 실행합니다.

```
# dladm set-linkprop -p protection=ip-nospoof
```

3 **allowed-ips 링크 등록 정보의 값으로 IP 주소 목록을 지정합니다.**

```
# dladm set-linkprop -p allowed-ips=IP-addr[,IP-addr,...] link
```

다음 예에서는 10.0.0.1 및 10.0.0.2 IP 주소를 vnic0 링크에 대한 allowed-ips 등록 정보의 값으로 지정하는 방법을 보여줍니다.

```
# dladm set-linkprop -p allowed-ips=10.0.0.1,10.0.0.2 vnic0
```

▼ 링크 보호 구성을 확인하는 방법

protection 및 allowed-ips 등록 정보의 값은 링크 보호의 구성 방식을 나타냅니다. allowed-ips 등록 정보는 protection 등록 정보가 ip-nospoof 보호 유형을 지정하는 경우에만 사용됩니다.

1 관리자로 전환합니다.

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

2 링크 보호 등록 정보 값을 확인합니다.

```
# dladm show-linkprop -p protection,allowed-ips link
```

다음 예에서는 vnic0 링크에 대한 protection 및 allowed-ips 등록 정보의 값을 보여줍니다.

```
# dladm show-linkprop -p protection,allowed-ips vnic0
LINK      PROPERTY        PERM     VALUE           DEFAULT     POSSIBLE
vnic0     protection      rw       ip-nospoof      --          --
                                   mac-nospoof
                                   restricted
vnic0     allowed-ips     rw       10.0.0.1,       --          --
                                   10.0.0.2
```

◆ ◆ ◆ 21 장

21 네트워크 리소스 관리

이 장에서는 VNIC와 같은 가상 링크를 비롯한 데이터 링크의 리소스를 관리하는 방법에 대해 설명합니다. 네트워크 리소스 관리는 서비스 품질을 구현하여 특히 가상 네트워크에서 성능을 향상시켜 줍니다.

이 장에서는 다음 항목을 다룹니다.

네트워크 리소스 관리의 개요

이 절에서는 네트워크 레인을 소개하여 네트워크 리소스 관리에 대해 설명합니다. 또한 데이터 링크 등록 정보를 설정하여 네트워크 리소스 관리를 구현하는 방법에 대해 설명합니다. 흐름도 네트워크 트래픽을 처리할 리소스 제어를 설정하는 또 다른 방법으로 정의됩니다.

리소스 제어를 위한 데이터 링크 등록 정보

이전 Oracle Solaris 릴리스에서는 서비스 품질 구현이 복잡한 프로세스였습니다. 이 프로세스는 대기열 원칙, 클래스 및 필터 규칙 정의와 이러한 모든 구성 요소 간의 관계 표시로 구성됩니다. 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 제V부, "IPQoS(IP Quality of Service)"를 참조하십시오.

이 릴리스에서는 네트워크 리소스를 관리하여 서비스 품질을 보다 쉽고 동적으로 얻을 수 있습니다. 네트워크 리소스 관리는 네트워크 리소스와 관련된 데이터 링크 등록 정보 설정으로 구성됩니다. 이러한 등록 정보를 설정하여 지정된 리소스 중 네트워킹 프로세스에 사용할 수 있는 양을 결정합니다. 예를 들어, 네트워킹 프로세스 전용으로 예약된 특정 개수의 CPU와 링크를 연결할 수 있습니다. 또는 특정 유형의 네트워크

트래픽을 처리하도록 지정된 대역폭을 링크에 할당할 수 있습니다. 리소스 등록 정보가 정의된 후 새 설정이 즉시 적용됩니다. 이 방법을 사용하면 리소스를 유연하게 관리할 수 있습니다. 링크를 만들 때 리소스 등록 정보를 설정할 수 있습니다. 또는 오랫동안 리소스 사용을 조사하고 리소스를 보다 효율적으로 할당하는 방법을 확인한 후와 같이 나중에 이러한 등록 정보를 설정할 수 있습니다. 리소스 할당 절차는 가상 네트워크 환경과 기존 물리적 네트워크에 모두 적용됩니다.

네트워크 리소스 관리는 트래픽의 전용 레인을 만드는 것에 비유됩니다. 여러 리소스를 결합하여 특정 유형의 네트워크 패킷을 처리하는 경우 해당 리소스가 이러한 패킷의 **네트워크 레인**을 형성합니다. 각 네트워크 레인에 대해 리소스를 다르게 할당할 수 있습니다. 예를 들어, 네트워크 트래픽이 가장 많은 레인에 리소스를 더 할당할 수 있습니다. 리소스가 실제 요구에 따라 분산되는 네트워크 레인을 구성하면 시스템의 패킷 처리 효율성이 증가합니다. 네트워크 레인에 대한 자세한 내용은 383 페이지 “네트워크 트래픽 흐름 개요”를 참조하십시오.

네트워크 리소스 관리는 다음 작업에 유용합니다.

- 네트워크 프로비저닝
- 서비스 단계 계약 설정
- 클라이언트 청구
- 보안 문제 진단

이전 릴리스의 복잡한 QoS 규칙 정의 없이 개별 시스템에서 데이터 트래픽을 격리시키고 우선 순위를 지정하고 추적 및 제어할 수 있습니다.

흐름을 사용한 네트워크 리소스 관리

흐름은 리소스를 사용하여 패킷을 처리하는 방식을 추가로 제어하기 위해 이러한 패킷을 분류하는 사용자 정의 방법입니다. 네트워크 패킷은 **속성**에 따라 분류될 수 있습니다. 속성을 공유하는 패킷은 흐름을 구성하며 특정 흐름 이름으로 레이블이 지정됩니다. 그런 다음 흐름에 특정 리소스를 할당할 수 있습니다.

흐름을 만드는 기본 역할을 하는 속성은 패킷 헤더의 정보에서 파생됩니다. 다음 속성 중 하나에 따라 패킷 트래픽을 흐름으로 구성할 수 있습니다.

- IP 주소
- 전송 프로토콜 이름(UDP, TCP 또는 SCTP)
- 응용 프로그램 포트 번호(예: FTP의 경우 포트 21)
- IPv6 패킷의 서비스 품질에만 사용되는 DS 필드 속성. DS 필드에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 “DS 코드 포인트”을 참조하십시오.

흐름은 목록의 속성 중 하나만 기반으로 할 수 있습니다. 예를 들어, 사용 중인 포트(예: FTP의 경우 포트 21) 또는 IP 주소(예: 특정 소스 IP 주소의 패킷)에 따라 흐름을 만들 수 있습니다. 하지만 포트 번호 21(FTP)에서 수신된 지정한 IP 주소의 패킷에 대해서는 흐름을 만들 수 없습니다. 마찬가지로, IP 주소 `192.168.1.10`의 모든 트래픽에 대해

흐름을 만든 다음 192.168.1.10의 전송 계층 트래픽에 대해 흐름을 만들 수는 없습니다. 따라서 각 흐름이 다른 속성을 기반으로 하는 여러 흐름을 시스템에 구성할 수 있습니다.

네트워크 리소스 관리 명령

네트워크 리소스 할당 명령은 데이터 링크 또는 흐름에서 직접 작업하는지에 따라 달라집니다.

- 데이터 링크의 경우 링크를 만들거나 기존 링크의 등록 정보를 설정하는 동안 등록 정보를 설정하는지 여부에 따라 적절한 dladm 하위 명령을 사용합니다. 동시에 링크를 만들고 리소스를 할당하려면 다음 구문을 사용합니다.

  ```
  # dladm create-vnic -l link -p property=value[,property=value] vnic
  ```

  여기서 *link*는 물리적 링크 또는 가상 링크일 수 있습니다.

  기존 링크의 등록 정보를 설정하려면 다음 구문을 사용합니다.

  ```
  # dladm set-linkprop -p property=value[,property=value] link
  ```

  dladm 명령과 이 명령이 관리하는 등록 정보에 대한 자세한 내용은 dladm(1M) 매뉴얼 페이지를 참조하십시오.

  다음은 리소스 할당에 설정할 수 있는 링크 등록 정보입니다.

  - 대역폭 - 특정 링크 사용에 대해 하드웨어 대역폭을 제한할 수 있습니다.
  - NIC 링 - NIC가 링 할당을 지원하는 경우 데이터 링크 전용으로 전송 및 수신 링을 할당할 수 있습니다. NIC 링은 360 페이지 “전송 및 수신 링”에서 설명합니다.
  - CPU 풀 - CPU 풀은 일반적으로 특정 영역으로 생성되고 연결됩니다. 이러한 풀을 데이터 링크에 할당하여 연결된 영역의 네트워크 프로세스 관리에 CPU 세트를 예약할 수 있습니다. CPU와 풀은 373 페이지 “풀 및 CPU”에서 설명합니다.
  - CPU – 여러 CPU가 있는 시스템에서는 특정 네트워크 처리를 위해 지정된 개수의 CPU를 전용으로 사용할 수 있습니다.
- 흐름의 경우 flowadm 하위 명령을 사용합니다. 먼저 flowadm add-flow 하위 명령을 사용하여 흐름을 만듭니다. 그런 다음 flowadm set-flowprop 하위 명령을 사용하여 흐름에 리소스를 할당합니다. 흐름의 특성을 결정하는 정의된 속성 세트가 시스템의 **흐름 제어 정책**을 구성합니다.

  주 – 흐름에 할당될 수 있는 리소스 할당의 등록 정보는 링크에 직접 할당된 등록 정보와 같습니다. 하지만 현재 대역폭 등록 정보만 흐름과 연결될 수 있습니다. 데이터 링크와 흐름에서 등록 정보를 설정하는 명령은 서로 다르지만 구문은 비슷합니다. 대역폭 등록 정보를 구성하려면 379 페이지 “흐름을 구성하는 방법”의 예를 참조하십시오.

  자세한 내용은 flowadm(1M) 매뉴얼 페이지를 참조하십시오.

네트워크 리소스 관리(작업 맵)

다음 표에서는 리소스 제어를 설정하고 이러한 리소스가 네트워크 처리에 할당되는 방식을 결정하는 여러 가지 방법을 보여줍니다.

작업	설명	수행 방법
MAC 클라이언트에 링을 할당합니다.	데이터 링크의 MAC 클라이언트가 링을 사용하도록 구성합니다.	361 페이지 “링 할당의 등록 정보”
데이터 링크에 CPU 풀을 할당합니다.	`pool` 등록 정보를 사용하여 영역의 네트워크 프로세스를 관리할 CPU 세트를 할당합니다.	376 페이지 “데이터 링크에 대한 CPU 풀을 구성하는 방법 ”
정의된 데이터 링크에 CPU 세트를 할당합니다.	여러 CPU가 있는 시스템에서 네트워킹 용도로 CPU 세트를 예약합니다.	377 페이지 “링크에 CPU를 할당하는 방법”
물리적 네트워크에서 흐름을 사용하여 네트워크 리소스 관리를 구현합니다.	네트워크 트래픽을 개별 흐름으로 격리시킵니다. 그런 다음 흐름 간에 정해진 양의 인터페이스 대역폭을 흐름에 할당합니다.	379 페이지 “흐름을 구성하는 방법”

데이터 링크의 리소스 관리

이 절에서는 물리적 네트워크나 가상 네트워크의 네트워크 성능을 향상시키기 위해 설정할 수 있는 선택한 링크 등록 정보에 대해 설명합니다.

전송 및 수신 링

NIC에서 수신(Rx) 링과 전송(Tx) 링은 각각 시스템이 네트워크 패킷을 받고 보내는 하드웨어 리소스입니다. 다음 절에서는 링의 개요 및 네트워킹 프로세스에 링을 할당하는 데 사용되는 절차를 차례로 제공합니다. 링을 할당하는 명령을 실행할 때 작동하는 방식을 보여주는 예도 제공됩니다.

MAC 클라이언트 및 링 할당

VNIC 및 기타 데이터 링크와 같은 MAC 클라이언트는 NIC 위에 구성되어 시스템과 기타 네트워크 노드 간의 통신을 사용할 수 있게 합니다. 클라이언트는 구성된 후 Rx 및 Tx 링을 모두 사용하여 각각 네트워크 패킷을 수신하거나 전송합니다. MAC 클라이언트는 하드웨어 기반이거나 소프트웨어 기반일 수 있습니다. 하드웨어 기반 클라이언트는 다음 조건 중 하나를 충족합니다.

- 하나 이상의 Rx 링을 전용으로 사용합니다.

- 하나 이상의 Tx 링을 전용으로 사용합니다.
- 하나 이상의 Rx 링과 하나 이상의 Tx 링을 전용으로 사용합니다.

이러한 조건을 하나도 충족하지 않는 클라이언트를 소프트웨어 기반 MAC 클라이언트라고 합니다.

하드웨어 기반 클라이언트에는 NIC에 따라 배타적 사용을 위한 링을 할당할 수 있습니다. `nxge`와 같은 NIC는 **동적 링 할당**을 지원합니다. 이러한 NIC에서는 하드웨어 기반 클라이언트를 구성할 수 있을 뿐만 아니라 링을 할당할 수 있는 경우 해당 클라이언트에 할당할 링 수를 결정하는 유연성도 있습니다. 링 사용은 항상 주 인터페이스(예: `nxge0`)에 최적화됩니다. 주 인터페이스를 **주 클라이언트**라고도 합니다. 다른 클라이언트의 배타적 사용에 할당되지 않은 사용 가능한 링은 모두 주 인터페이스에 자동으로 할당됩니다.

`ixge`와 같은 기타 NIC는 **정적 링 할당**만 지원합니다. 이러한 NIC에서는 하드웨어 기반 클라이언트만 만들 수 있습니다. 클라이언트는 클라이언트당 고정된 링 세트를 사용하여 자동으로 구성됩니다. 고정된 세트는 NIC 드라이버의 초기 구성 도중 결정됩니다. 정적 링 할당과 관련된 드라이버의 초기 구성에 대한 자세한 내용은 **Oracle Solaris 조정 가능 매개변수 참조 설명서**을 참조하십시오.

VLAN의 링 할당

VLAN을 사용하면 VLAN 생성 방식에 따라 링 할당이 다르게 진행됩니다. VLAN은 다음 두 가지 방식 중 하나로 생성됩니다.

- `dladm create-vlan` 하위 명령 사용:

  ```
  # dladm create-vlan -l link -v VID vlan
  ```

- `dladm create-vnic` 하위 명령 사용:

  ```
  # dladm create-vnic -l link -v VID vnic
  ```

`dladm create-vlan` 하위 명령으로 생성된 VLAN은 기본 인터페이스와 동일한 MAC 주소를 공유합니다. 따라서 이 VLAN은 기본 인터페이스의 Rx 및 Tx 링도 공유합니다. `dladm create-vnic` 명령을 사용하여 VNIC로 생성된 VLAN에는 기본 인터페이스와 다른 MAC 주소가 있습니다. 이러한 VLAN에 대한 링 할당은 기본 링크에 대한 할당과 독립적입니다. 따라서 NIC가 하드웨어 기반 클라이언트를 지원하는 경우 이 VLAN에 고유한 전용 링을 할당할 수 있습니다.

링 할당의 등록 정보

링을 관리하려면 `dladm` 명령을 사용하여 두 개의 링 등록 정보를 설정할 수 있습니다.

- `rxrings`는 지정한 링크에 할당된 Rx 링 수를 나타냅니다.
- `txrings`는 지정한 링크에 할당된 Tx 링 수를 나타냅니다.

각 등록 정보를 세 가지 가능한 값 중 하나로 설정할 수 있습니다.

- sw는 소프트웨어 기반 클라이언트를 구성하고 있음을 나타냅니다. 클라이언트는 링을 배타적으로 사용하지 않습니다. 대신 클라이언트는 유사하게 구성된 다른 기존 클라이언트와 링을 공유합니다.
- *n* > *0*(0보다 큰 수)은 하드웨어 기반 클라이언트의 구성에만 적용됩니다. 이 숫자는 배타적 사용을 위해 클라이언트에 할당되는 링의 수량을 나타냅니다. 기본 NIC가 동적 링 할당을 지원하는 경우에만 숫자를 지정할 수 있습니다.
- hw도 하드웨어 기반 클라이언트의 구성에 적용됩니다. 하지만 이러한 클라이언트의 경우 실제 전용 링 수를 지정할 수 없습니다. 대신 클라이언트당 고정된 링 수가 NIC 드라이버의 초기 구성에 따라 이미 설정되어 있습니다. 기본 NIC가 정적 링 할당만 지원하는 경우 *rings 등록 정보를 hw로 설정합니다.

현재 링 할당 및 사용에 대한 정보를 제공하려면 다음과 같은 추가 읽기 전용 링 등록 정보를 사용할 수 있습니다.

- rxrings-available 및 txrings-available은 할당할 수 있는 Rx 및 Tx 링 수를 나타냅니다.
- rxhwclnt-available 및 txhwclnt-available은 NIC에 구성할 수 있는 Rx 및 Tx 하드웨어 기반 클라이언트 수를 나타냅니다.

하드웨어 기반 클라이언트 구성 준비

하드웨어 기반 클라이언트를 구성하려면 시스템에 있는 NIC의 링 할당 기능을 알고 있어야 합니다. 필요한 정보를 가져오려면 다음 명령을 사용합니다.

```
# dladm show-linkprop link
```

여기서 *link*는 특정 NIC의 데이터 링크를 나타냅니다.

특정 등록 정보를 표시하려면 다음 명령을 사용합니다.

```
# dladm show-linkprop -p property[,property,...] link
```

하드웨어 기반 클라이언트를 올바르게 구성하려면 다음을 확인해야 합니다.

- NIC가 하드웨어 기반 클라이언트를 지원하는지 여부

  명령 출력 결과의 rxrings 및 txrings 등록 정보는 NIC가 하드웨어 기반 클라이언트를 지원하는지 여부를 나타냅니다. 동일한 데이터에서 NIC가 지원하는 링 할당 유형을 확인할 수도 있습니다.
- 하드웨어 기반 클라이언트에 할당할 링의 가용성

  명령 출력 결과의 rxrings-available 및 txrings-available 등록 정보는 하드웨어 기반 클라이언트에 할당할 수 있는 사용 가능한 Rx 링과 Tx 링을 나타냅니다.
- 링크에 구성할 수 있는 하드웨어 기반 클라이언트의 가용성

링은 세트로 할당됩니다. 사용 가능한 링 수와 전용 링을 사용할 수 있는 클라이언트 수 간에 일대일 관계는 없습니다. 따라서 링을 할당하려면 링의 가용성뿐 아니라 전용 링을 사용하도록 구성할 수 있는 추가 하드웨어 기반 클라이언트 수도 확인해야 합니다. 링과 하드웨어 기반 클라이언트를 모두 사용할 수 있는 경우에만 링을 할당할 수 있습니다.

명령 출력 결과의 `rxhwclnt-available` 및 `txhwclnt-available` 등록 정보는 전용 Rx 및 Tx 링을 사용하도록 구성할 수 있는 하드웨어 기반 클라이언트 수를 나타냅니다.

NIC가 링 할당을 지원하고 링과 하드웨어 기반 클라이언트를 사용할 수 있는 경우 365 페이지 “하드웨어 기반 클라이언트를 구성하는 방법”에 설명된 대로 시스템에 이 유형의 클라이언트를 구성할 수 있습니다. 또는 366 페이지 “소프트웨어 기반 클라이언트를 만드는 방법”에 설명된 대로 소프트웨어 기반 클라이언트를 대신 구성할 수 있습니다.

다음 예에서는 `nxge` NIC, `ixgbe` NIC 및 `e1000g` NIC의 링 관련 링크 등록 정보에 대해 표시되는 서로 다른 정보를 보여줍니다.

예 21–1 nxge NIC 링 정보

다음 예에서는 `nxge` NIC에 대한 링 정보를 보여줍니다.

```
# dladm show-linkprop nxge0
LINK     PROPERTY             PERM  VALUE  DEFAULT  POSSIBLE
...
nxge0    rxrings              rw    --     --       sw,<1-7>
...
nxge0    txrings              rw    --     --       sw,<1-7>
...
nxge0    rxrings-available    r-    5      --       --
nxge0    txrings-available    r-    5      --       --
nxge0    rxhwclnt-available   r-    2      --       --
nxge0    txhwclnt-available   r-    2      --       --
...
```

`POSSIBLE` 필드에는 `sw` 및 `<1-7>`이 `rxrings` 및 `txrings` 등록 정보에 허용되는 값으로 나열됩니다. 이러한 값은 `nxge`가 하드웨어 기반 클라이언트 및 소프트웨어 기반 클라이언트를 모두 지원함을 나타냅니다. `<1-7>` 범위는 설정하는 Rx 링 또는 Tx 링 수가 지정한 범위 내에 있어야 함을 나타냅니다. 범위에서 NIC가 수신 및 전송측에서 모두 동적 링 할당을 지원하는 것을 추론할 수도 있습니다.

또한 `*rings-available` 등록 정보는 Rx 링 5개와 Tx 링 5개를 하드웨어 기반 클라이언트에 할당할 수 있음을 나타냅니다.

하지만 `*clnt-available` 등록 정보를 기반으로 사용 가능한 Rx 링을 배타적으로 사용할 수 있는 두 클라이언트만 구성할 수 있습니다. 마찬가지로, 사용 가능한 Tx 링을 배타적으로 사용할 수 있는 두 클라이언트만 구성할 수 있습니다.

예 21–2 ixgbe NIC 링 정보

다음 예에서는 `ixgbe` NIC에 대한 링 정보를 보여줍니다.

예 21–2 ixgbe NIC 링 정보 *(계속)*

```
# dladm show-linkprop ixgbe0
LINK     PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0   rxrings             rw     --     --       sw,hw
...
ixgbe0   txrings             rw     --     --       sw,hw,<1-7>
...
ixgbe0   rxrings-available   r-     0      --       --
ixgbe0   txrings-available   r-     5      --       --
ixgbe0   rxhwclnt-available  r-     0      --       --
ixgbe0   txhwclnt-available  r-     7      --       --
...
```

rxrings 및 txrings 등록 정보의 POSSIBLE 필드는 ixgbe0에서 하드웨어 기반 클라이언트와 소프트웨어 기반 클라이언트를 모두 구성할 수 있음을 나타냅니다. Rx 링에는 정적 링 할당만 지원되며, 이 경우 하드웨어가 각 하드웨어 기반 클라이언트에 고정된 Rx 링 세트를 할당합니다. 하지만 Tx 링은 동적으로 할당할 수 있으므로 하드웨어 기반 클라이언트에 할당할 Tx 링 수를 결정할 수 있습니다(이 예에서는 최대 7개 링).

또한 *rings-available 등록 정보는 Tx 링 5개를 하드웨어 기반 클라이언트에 할당할 수 있지만 Rx 링은 할당할 수 없음을 나타냅니다.

최종적으로, *hwclnt-available 등록 정보를 기반으로 Tx 링을 배타적으로 사용할 하드웨어 기반 Tx 클라이언트를 7개 구성할 수 있습니다. 하지만 동적 Rx 링 할당은 ixgbe 카드에서 지원되지 않습니다. 따라서 지정한 전용 Rx 링 세트를 사용하여 하드웨어 기반 클라이언트를 만들 수 없습니다.

*rings-available 등록 정보 중 하나의 VALUE 필드 아래에 0이 있으면 다음 중 하나를 의미할 수 있습니다.

- 클라이언트에 할당할 수 있는 링이 더 이상 없습니다.
- 동적 링 할당이 지원되지 않습니다.

rxrings 및 txrings의 POSSIBLE 필드와 rxrings-available 및 txrings-available의 VALUE 필드를 비교하여 0의 의미를 확인할 수 있습니다.

예를 들어, 다음과 같이 txrings-available이 0이라고 가정합니다.

```
# dladm show-linkprop ixgbe0
LINK     PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0   rxrings             rw    --     --       sw,hw
ixgbe0   txrings             rw    --     --       sw,hw,<1-7>
ixgbe0   rxrings-available   r-    0      --       --
ixgbe0   txrings-available   r-    0      --       --
...
```

이 출력 결과에서 rxrings-available의 VALUE 필드는 0이고 rxrings의 POSSIBLE 필드는 sw,hw입니다. 결합된 정보는 NIC가 동적 링 할당을 지원하지 않으므로 Rx 링을 사용할 수 없음을 의미합니다. 전송측에서 txrings-available의 VALUE 필드는 0이고 txrings의

예 21–2 ixgbe NIC 링 정보 *(계속)*

POSSIBLE 필드는 sw,hw,<1-7>입니다. 결합된 정보는 모든 Tx 링이 이미 할당되었기 때문에 Tx 링을 사용할 수 없음을 나타냅니다. 하지만 txrings의 POSSIBLE 필드에 따라 동적 링 할당은 지원됩니다. 따라서 Tx 링을 사용할 수 있게 되면 해당 링을 할당할 수 있습니다.

예 21–3 e1000g NIC 링 정보

다음 예에서는 e1000g NIC에 대한 링 정보를 보여줍니다.

```
# dladm show-linkprop e1000g0
LINK     PROPERTY             PERM  VALUE  DEFAULT  POSSIBLE
...
e1000g0   rxrings             rw     --       --     --
...
e1000g0   txrings             rw     --       --     --
...
e1000g0   rxrings-available   r-     0        --     --
e1000g0   txrings-available   r-     0        --     --
e1000g0   rxhwclnt-available  r-     0        --     --
e1000g0   txhwclnt-available  r-     0        --     --
...
```

출력 결과에 따르면 e1000g NIC에서 링 할당이 지원되지 않으므로 링과 하드웨어 기반 클라이언트를 모두 구성할 수 없습니다.

▼ 하드웨어 기반 클라이언트를 구성하는 방법

이 절차에서는 동적 링 할당을 지원하는 NIC 또는 정적 링 할당을 지원하는 NIC에서 하드웨어 기반 클라이언트를 구성하는 방법을 보여줍니다.

시작하기 전에 시스템의 NIC에 대해 다음 정보가 있는지 확인합니다.

- NIC가 하드웨어 기반 클라이언트를 지원하는지 여부
- NIC가 지원하는 링 할당 유형
- 하드웨어 기반 클라이언트에 할당할 링의 가용성
- 링크에 구성할 수 있는 하드웨어 기반 클라이언트의 가용성

이 정보를 가져오려면 362 페이지 “하드웨어 기반 클라이언트 구성 준비”를 참조하십시오.

1 NIC가 지원하는 링 할당 유형에 따라 다음 단계 중 하나를 수행합니다.

- NIC가 동적 링 할당을 지원하는 경우 다음 구문을 사용합니다.

  # dladm create-vnic -p rxrings=*number***[,txrings=***number***] -l** *link vnic*

  number 클라이언트에 할당하는 Rx 링과 Tx 링 수를 나타냅니다. 이 숫자는 할당할 수 있는 링 수의 범위 내에 있어야 합니다.

주 – 일부 NIC는 Rx 링 또는 Tx 링에서 동적 할당을 지원하지만 두 유형에서 모두 지원하지는 않습니다. 동적 링 할당이 지원되는 링 유형에 대해 *number*를 지정합니다.

link 클라이언트를 만드는 데 사용 중인 데이터 링크를 나타냅니다.

vnic 구성 중인 클라이언트를 나타냅니다.

- NIC가 정적 링 할당을 지원하는 경우 다음 구문을 사용합니다.

```
# dladm create-vnic -p rxrings=hw[,txrings=hw] -l link vnic
```

주 – 일부 NIC는 Rx 링 또는 Tx 링에서 정적 할당을 지원하지만 두 유형에서 모두 지원하지는 않습니다. 정적 링 할당이 지원되는 링 유형에 대해 `hw`를 지정합니다.

2 (옵션) 새로 만든 클라이언트의 링 정보를 확인합니다.

```
# dladm show-linkprop vnic
```

▼ 소프트웨어 기반 클라이언트를 만드는 방법

소프트웨어 기반 클라이언트는 링을 배타적으로 사용하지 않습니다. 대신 클라이언트는 주 클라이언트 또는 다른 기존 소프트웨어 기반 클라이언트가 있는 인터페이스와 링 사용을 공유하지 않습니다. 소프트웨어 기반 클라이언트의 링 수는 기존 하드웨어 기반 클라이언트 수에 따라 달라집니다.

● **다음 단계 중 하나를 수행합니다.**

- 새 소프트웨어 기반 클라이언트를 만들려면 다음 명령을 입력합니다.

```
# dladm create-vnic -p rxrings=sw[,txrings=sw] -l link vnic
```

link 클라이언트를 만드는 데 사용 중인 데이터 링크를 나타냅니다.

vnic 구성 중인 클라이언트를 나타냅니다.

- 기존 클라이언트가 다른 클라이언트와 링을 공유하도록 구성하려면 다음 명령을 입력합니다.

```
# dladm set-linkprop -p rxrings=sw[,txrings=sw] vnic
```

예 21–4 하드웨어 기반 클라이언트 및 소프트웨어 기반 클라이언트 구성

이 예에서는 `ixgbe` NIC가 있는 시스템에서 하드웨어 기반 클라이언트 및 소프트웨어 기반 클라이언트를 모두 구성하는 방법을 보여줍니다. 링 할당이 구현되는 방식을 보여주기 위해 이 예는 여러 부분으로 나뉘어져 있습니다. 링 관련 정보는 구성 프로세스의 각 단계에서 표시되고 설명됩니다. 구성은 다음과 같이 진행됩니다.

1. 클라이언트 구성 전에 시스템의 링크 및 링 사용을 표시합니다.
2. 주 클라이언트를 구성합니다.
3. 소프트웨어 기반 클라이언트를 구성합니다.
4. 전용 링 없이 다른 클라이언트를 구성합니다.
5. 새로 구성된 클라이언트에 링을 정적으로 할당합니다.
6. 동적으로 할당된 전용 링이 있는 세번째 클라이언트를 구성합니다.

먼저 링크, 링 사용 및 링 관련 등록 정보를 표시합니다.

```
# dladm show-link
LINK     CLASS   MTU    STATE   BRIDGE   OVER
ixgbe0   phys    1500   down    --       --

# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS   CLIENTS
ixgbe0   RX         0-1     <default,mcast>
ixgbe0   TX         0-7     <default>
ixgbe0   RX         2-3     --
ixgbe0   RX         4-5     --
ixgbe0   RX         6-7     --

# dladm show-linkprop ixgbe0
LINK    PROPERTY              PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0   rxrings              rw    --     --       sw,hw
ixgbe0   rxrings-effective    r     --     --       --
ixgbe0   txrings              rw    --     --       sw,hw,<1-7>
ixgbe0   txrings-effective    r     --     --       --
ixgbe0   txrings-available    r-    7      --       --
ixgbe0   rxrings-available    r-    0      --       --
ixgbe0   rxhwclnt-available   r-    3      --       --
ixgbe0   txhwclnt-available   r-    7      --       --
...
```

명령 출력 결과에 시스템의 단일 링크 `ixgbe0`이 표시되지만 기존 클라이언트는 표시되지 않습니다. 또한 이 출력 결과에서 다음 정보도 얻을 수 있습니다.

- NIC에 Rx 링 8개와 Tx 링 8개가 있습니다(링 `0` - `7`).
- 하드웨어 기반 클라이언트의 경우 Rx 링에 정적 링 할당만 지원되지만 Tx 링에는 정적 및 동적 링 할당이 모두 지원됩니다.
- Rx 링과 Tx 링에 대해 모두 소프트웨어 기반 클라이언트를 구성할 수 있습니다.
- 1에서 7까지 Tx 링 7개를 다른 클라이언트에 동적으로 할당할 수 있습니다. 일반적으로 링 `0`은 주 클라이언트에 예약되었습니다. Rx 링에는 동적 링 할당이 지원되지 않으므로 Rx 링을 사용할 수 없습니다.
- 하드웨어 기반 클라이언트 3개는 Rx 링을 사용하도록 구성하고 하드웨어 기반 클라이언트 7개는 Tx 링을 사용하도록 구성할 수 있습니다.

`*rings-effective` 등록 정보에 대한 자세한 내용은 372 페이지 “정적 링 할당에서 링 할당을 식별하는 방법”을 참조하십시오.

다음은 주 클라이언트를 구성합니다.

```
# ipadm create-ip ixgbe0
# ipadm create-addr -T static -a 192.168.10.10/24 ixgbe0/v4
# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS   CLIENTS
ixgbe0   RX         0-1     <default,mcast>
ixgbe0   TX         0-7     <default>ixgbe0
ixgbe0   RX         2-3     ixgbe0
ixgbe0   RX         4-5     --
ixgbe0   RX         6-7     --

# dladm show-linkprop ixgbe0
LINK    PROPERTY              PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0   rxrings              rw    --     --       sw,hw
ixgbe0   rxrings-effective    r     2      --       --
ixgbe0   txrings              rw    --     --       sw,hw,<1-7>
ixgbe0   txrings-effective    r     8      --       --
ixgbe0   txrings-available    r-    7      --       --
ixgbe0   rxrings-available    r-    0      --       --
ixgbe0   rxhwclnt-available   r-    3      --       --
ixgbe0   txhwclnt-available   r-    7      --       --
...
```

출력 결과에서 다음 정보를 제공합니다.

- 주 클라이언트인 ixgbe0은 전용 사용을 위해 Rx 링 두 개(링 2와 3)를 자동으로 받습니다. 하지만 ixgbe0은 모든 Tx 링을 사용합니다. 기본적으로 사용되지 않은 모든 링은 주 클라이언트에 자동으로 할당됩니다.
- 다른 클라이언트에 할당할 수 있는 사용 가능한 Tx 링 수는 7개로 유지됩니다.
- Rx 링으로 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수는 3개로 유지됩니다. Tx 링으로 동적으로 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수는 7개로 유지됩니다.

다음은 VNIC를 소프트웨어 기반 클라이언트로 만듭니다.

```
# dladm create-vnic -l ixgbe0 -p rxrings=sw,txrings=sw vnic0
# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS   CLIENTS
ixgbe0   RX         0-1     <default,mcast>,vnic0
ixgbe0   TX         0-7     <default>vnic0,ixgbe0
ixgbe0   RX         2-3     ixgbe0
ixgbe0   RX         4-5     --
ixgbe0   RX         6-7     --

# dladm show-linkprop vnic0
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
vnic0   rxrings             rw    sw     --       sw,hw
...
vnic0   txrings             rw    sw     --       sw,hw,<1-7>
...
```

```
# dladm show-linkprop ixgbe0
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0   rxrings             rw    --     --       --
ixgbe0   rxrings-effective   r     2      --       --
ixgbe0   txrings             rw    --     --        sw,hw,<1-7>
ixgbe0   txrings-effective   r     --     --        --
ixgbe0   txrings-available   r-    7      --        --
ixgbe0   rxrings-available   r-    0      --        --
ixgbe0   rxhwclnt-available  r-    3      --        --
ixgbe0   txhwclnt-available  r-    7      --        --
...
```

출력 결과에서 다음 정보를 제공합니다.

- 소프트웨어 기반 클라이언트인 `vnic0`은 Rx 링 `0`과 `1`을 사용하도록 자동으로 할당됩니다. 이후에 생성된 Rx 링이 있는 기타 소프트웨어 기반 클라이언트는 기본적으로 이 쌍을 사용하도록 할당됩니다. 기본적으로 `vnic0`에는 Tx 링 8개(링 `0` - `7`) 사용도 모두 할당됩니다. 이후에 생성된 Tx 링이 있는 기타 소프트웨어 기반 클라이언트는 기본적으로 이 링 세트를 사용하도록 할당됩니다.
- 소프트웨어 기반 클라이언트인 `vnic0`의 `rxrings` 및 `txrings` 등록 정보가 `sw`로 적절하게 설정됩니다.
- Tx 링은 할당되지 않습니다. 따라서 다른 클라이언트에 할당할 수 있는 사용 가능한 Tx 링 수는 7개로 유지됩니다.
- Rx 링으로 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수는 3개로 유지됩니다. Tx 링으로 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수는 7개로 유지됩니다.

다음은 링 할당 없이 다른 클라이언트를 구성합니다.

```
# dladm create-vnic -l ixgbe0 vnic1
# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS     CLIENTS
ixgbe0   RX         0-1       <default,mcast>,vnic0
ixgbe0   TX         0,2-7     <default>vnic0,ixgbe0
ixgbe0   RX         2-3       ixgbe0
ixgbe0   RX         4-5       vnic1
ixgbe0   RX         6-7       --
ixgbe0   TX         1         vnic1

# dladm show-linkprop vnic1
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
vnic1   rxrings             rw    --     --       sw,hw
vnic1   rxrings-effective   r-    2      --       --
vnic1   txrings             rw    --     --       sw,hw,<1-7>
vnic1   txrings-effective   r-    --     --       --
...

# dladm show-linkprop ixgbe0
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
```

```
ixgbe0  rxrings             rw    --      --        sw,hw
ixgbe0  rxrings-effective   r-    2       --        --
ixgbe0  txrings             rw    --      --        sw,hw,<1-7>
ixgbe0  txrings-effective   r-    --      --        --
ixgbe0  txrings-available   r-    7       --        --
ixgbe0  rxrings-available   r-    0       --        --
ixgbe0  rxhwclnt-available  r-    3       --        --
ixgbe0  txhwclnt-available  r-    7       --        --
...
```

출력 결과에서 다음 정보를 제공합니다.

- 링 할당이 지원되는 경우 rxrings 및 txrings 등록 정보가 설정되지 않았어도 구성된 클라이언트가 하드웨어 기반 클라이언트로 간주됩니다. 따라서 vnic1은 해당 사용을 위해 전용 Rx 링 두 개(링 4와 5)를 자동으로 받습니다. 마찬가지로, vnic1은 전용 Tx 링(링 1)도 받습니다.
- 8개 Tx 링 중에서 ixgbe0과 vnic0은 이제 7개 링(링 0과 링 2 - 7)을 공유합니다. 링 1은 vnic1의 전용 Tx 링이 되었습니다.
- Tx 링은 할당되지 않습니다. 따라서 다른 클라이언트에 할당할 수 있는 사용 가능한 Tx 링 수는 7개로 유지됩니다.
- Rx 링으로 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수는 3개로 유지됩니다. Tx 링으로 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수는 7개로 유지됩니다.

다음은 새로 구성된 클라이언트 vnic1에 링을 정적으로 할당합니다.

```
# dladm set-linkprop -p rxrings=hw,txrings=hw vnic1
# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS     CLIENTS
ixgbe0   RX         0-1       <default,mcast>,vnic0
ixgbe0   TX         0,2-7     <default>vnic0,ixgbe0
ixgbe0   RX         2-3       ixgbe0
ixgbe0   RX         4-5       vnic1
ixgbe0   RX         6-7       --
ixgbe0   TX         1         vnic1

# dladm show-linkprop vnic1
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
vnic1   rxrings             rw    hw     --       sw,hw
vnic1   rxrings-effective   r-    2      --       --
vnic1   txrings             rw    hw     --       sw,hw,<1-7>
vnic1   txrings-effective   r-    --     --       --
...
# dladm show-linkprop ixgbe0
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0  rxrings             rw     --    --       sw,hw
ixgbe0  rxrings-effective   r-     2     --       --
ixgbe0  txrings             rw     --    --       sw,hw,<1-7>
ixgbe0  txrings-effective   r-     --    --       --
ixgbe0  txrings-available   r-     6     --       --
```

```
ixgbe0  rxrings-available   r-     0     --       --
ixgbe0  rxhwclnt-available  r-     3     --       --
ixgbe0  txhwclnt-available  r-     6     --       --
...
```

출력 결과에서 다음 정보를 제공합니다.

- vnic1에 대한 Rx 및 Tx 배포는 링 할당 없이 vnic1이 생성된 경우와 동일하게 유지됩니다.
- 마찬가지로, 링 정보는 링 할당 없이 vnic1이 생성된 경우와 동일하게 유지됩니다.
- vnic1의 rxrings 및 txrings 등록 정보가 명시적으로 hw로 설정되었습니다. 따라서 동적 할당에 사용 가능한 Tx 링 수가 6개로 감소했습니다. 마찬가지로, 구성할 수 있는 사용 가능한 하드웨어 기반 클라이언트 수도 6개로 감소했습니다.

다음은 동적으로 할당된 Tx 링이 있는 하드웨어 기반 클라이언트를 구성합니다.

```
# dladm create-vnic -l ixgbe0 -p txrings=2 vnic2
# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS     CLIENTS
ixgbe0   RX         0-1       <default,mcast>,vnic0
ixgbe0   TX         0,4-7     <default>vnic0,ixgbe0
ixgbe0   RX         2-3       ixgbe0
ixgbe0   RX         4-5       vnic1
ixgbe0   RX         6-7       vnic2
ixgbe0   TX         1         vnic1
ixgbe0   TX         2-3       vnic2

# dladm show-linkprop vnic2
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
vnic2   rxrings             rw    --     --       sw,hw
vnic2   rxrings-effective   r-    2      --       --
vnic2   txrings             rw    2      --       sw,hw,<1-7>
vnic2   txrings-effective   r-    2      --       --
...
# dladm show-linkprop ixgbe0
LINK    PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
ixgbe0  rxrings             rw    --     --       sw,hw
ixgbe0  rxrings-effective   r-    2      --       --
ixgbe0  txrings             rw    --     --       sw,hw,<1-7>
ixgbe0  txrings-effective   r-    --     --       --
ixgbe0  txrings-available   r-    4      --       --
ixgbe0  rxrings-available   r-    0      --       --
ixgbe0  rxhwclnt-available  r-    3      --       --
ixgbe0  txhwclnt-available  r-    5      --       --
...
```

출력 결과에서 다음 정보를 제공합니다.

- 하드웨어가 배타적 사용을 위해 한 쌍의 Rx 링(링 6 및 7)을 vnic2에 자동으로 할당했습니다. 하지만 관리자가 vnic2의 전용 Tx 링 두 개(링 2와 3)를 할당했습니다.

- 관리 차원에서 Tx 링 두 개가 vnic2에 할당되었으므로 다른 클라이언트에 할당할 수 있는 사용 가능한 Tx 링 수가 네 개로 감소했습니다.
- vnic2는 Tx 링 두 개가 있는 하드웨어 기반 클라이언트로 구성되었으므로 구성할 수 있는 사용 가능한 클라이언트 수가 5개로 감소했습니다.

▼ 정적 링 할당에서 링 할당을 식별하는 방법

정적 링 할당을 사용하여 하드웨어 기반 클라이언트를 구성하는 경우 하드웨어가 할당할 링 수를 결정합니다. 하지만 rxrings 및 txrings 등록 정보는 hw로 설정되며 실제로 할당된 링 수를 나타내지 않습니다. 대신 rxrings-effective 및 txrings-effective 등록 정보를 확인하여 이 개수를 얻을 수 있습니다.

1 다음 단계 중 하나를 수행하여 정적 링 할당으로 하드웨어 기반 클라이언트를 구성합니다.

- 정적 링 할당으로 클라이언트를 만들려면 다음 명령을 입력합니다.

  ```
  # dladm create-vnic -l link -p rxrings=hw[,txrings=hw] vnic
  ```

  link 클라이언트를 만드는 데 사용 중인 데이터 링크를 나타냅니다.

  vnic 구성 중인 클라이언트를 나타냅니다.
- 기존 클라이언트에 링을 정적으로 할당하려면 다음 명령을 입력합니다.

  ```
  # dladm set-linkprop -p rxrings=hw[,txrings=hw] vnic
  ```

2 할당된 링 수를 식별하려면 다음 하위 단계를 수행합니다.

a. 클라이언트의 등록 정보를 표시합니다.

```
# dladm show-linkprop link
```

여기서 *link*는 하드웨어 기반 클라이언트 또는 VNIC를 나타냅니다.

b. 정적으로 할당한 링 유형에 해당하는 *rings-effective 등록 정보의 값을 확인합니다.

예를 들어, Rx 링을 정적으로 할당한 경우 rxrings-effective 등록 정보를 확인합니다. Tx 링을 정적으로 할당한 경우 txrings-effective 등록 정보를 확인합니다. 이 숫자는 하드웨어가 할당한 링 수를 나타냅니다.

3 정적으로 할당된 링을 확인하려면 다음 하위 단계를 수행합니다.

a. NIC의 링 사용을 표시합니다.

```
# dladm show-phys -H link
```

여기서 *link*는 주 클라이언트를 나타냅니다.

b. 명령 출력 결과를 통해 첫번째 단계에서 구성한 하드웨어 기반 클라이언트에 할당된 Rx 링 또는 Tx 링을 확인합니다.

예 21–5 정적으로 할당된 링 식별

이 예에서는 ixgbe NIC에 구성된 클라이언트에 Rx 링이 정적으로 할당된 방식을 보여줍니다. 해당 NIC에서 Rx 링에는 정적 할당만 지원됩니다. 이 예는 다음과 같이 진행됩니다.

1. 시스템의 링크를 표시합니다. 이 예에서는 시스템에 ixgbe0이라는 링크 하나만 있습니다.
2. 정적으로 할당된 Rx 링이 있는 하드웨어 기반 클라이언트로 vnic1을 만듭니다.
3. 링 정보를 표시하여 하드웨어가 할당한 링 수를 확인합니다.
4. 링 사용을 표시하여 할당된 링을 식별합니다.

```
# dladm show-link
LINK     CLASS    MTU    STATE   BRIDGE   OVER
ixgbe0   phys     1500   down    --       --

# dladm create-vnic -l ixgbe0 -p rxrings=hw vnic1
# dladm show-linkprop vnic1
LINK     PROPERTY            PERM  VALUE  DEFAULT  POSSIBLE
...
vnic1    rxrings             rw    hw     --       sw,hw
vnic1    rxrings-effective   r-    2      --       --
vnic1    txrings             rw    --     --       sw,hw,<1-7>
vnic1    txrings-effective   r-    --     --       --

# dladm show-phys -H ixgbe0
LINK     RINGTYPE   RINGS     CLIENTS
ixgbe0   RX         0-1       <default,mcast>
ixgbe0   TX         0,2-7     <default>
ixgbe0   RX         2-3       vnic1
ixgbe0   RX         4-5       --
ixgbe0   RX         6-7       --
ixgbe0   TX         1         vnic1
...
```

출력 결과에 따르면 vnic1이 Rx 링으로 구성된 후 하드웨어가 rxrings-effective 등록 정보에 반영된 대로 전용 Rx 링 두 개를 할당했습니다. **dladm show-phys -H** 명령의 출력 결과를 기준으로 Rx 링 2와 3이 vnic1 전용으로 지정되었습니다.

클라이언트로 구성된 결과, vnic1은 전용 사용을 위해 Tx 링 1도 자동으로 받았습니다. 하지만 txrings 등록 정보가 명시적으로 설정되지 않았으므로 txrings-effective 등록 정보가 아무 값도 표시하지 않습니다.

풀 및 CPU

풀은 네트워크 처리를 CPU 풀에 바인딩할 수 있는 링크 등록 정보입니다. 이 등록 정보를 사용하면 네트워크 리소스 관리와 영역의 CPU 바인딩 및 관리를 보다 효율적으로 통합할 수 있습니다. Oracle Solaris에서 영역 관리에는 zonecfg 또는 poolcfg 명령을

사용하여 CPU 리소스 풀에 비네트워킹 프로세스를 바인딩하는 작업이 포함됩니다. 동일한 리소스 풀을 바인딩하여 네트워크 프로세스도 관리하려면 `dladm set-linkprop` 명령을 사용하여 링크의 `pool` 등록 정보를 구성합니다. 그런 다음 해당 링크를 영역에 할당합니다.

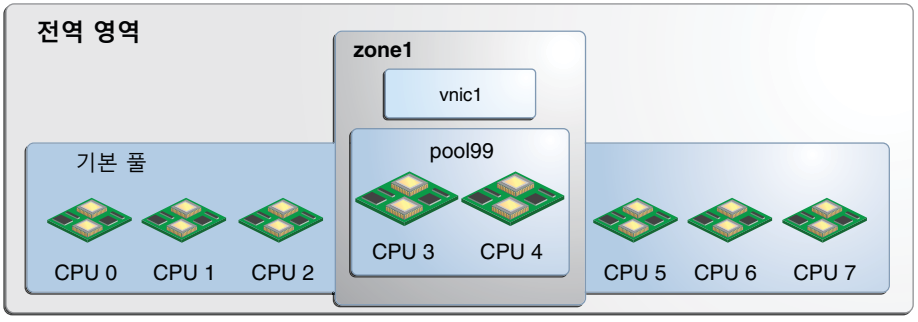
링크에 `pool` 등록 정보를 설정하고 링크를 영역의 네트워크 인터페이스로 할당하면 해당 링크가 영역의 풀에도 바인딩됩니다. 해당 영역이 배타적 영역이 되도록 설정된 경우 이 영역에 할당되지 않은 다른 데이터 링크가 풀의 CPU 리소스를 더 이상 사용할 수 없습니다.

주 – 별도의 등록 정보인 `cpu`를 설정하여 데이터 링크에 특정 CPU를 할당할 수 있습니다. `cpu`와 `pool` 등록 정보는 상호 배타적입니다. 두 등록 정보를 지정된 데이터 링크에 모두 설정할 수는 없습니다. `cpu` 등록 정보를 사용하여 데이터 링크에 CPU 리소스를 할당하려면 377 페이지 “링크에 CPU를 할당하는 방법”을 참조하십시오.

영역 내의 풀에 대한 자세한 내용은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 13 장, “리소스 풀 만들기 및 관리(작업)”를 참조하십시오. 풀 만들기 및 풀에 CPU 세트 할당에 대한 자세한 내용은 `poolcfg(1M)` 매뉴얼 페이지를 참조하십시오.

다음 그림에서는 `pool` 등록 정보가 데이터 링크에 할당된 경우 풀 작동 방식을 보여줍니다.

그림 21–1 영역에 할당된 VNIC의 pool 등록 정보

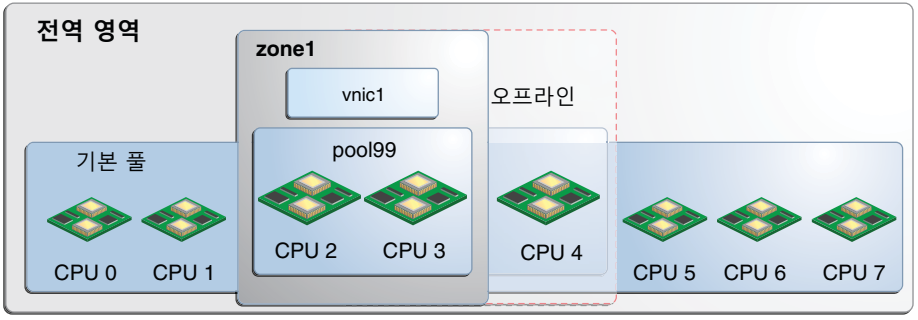


이 그림의 시스템에는 CPU가 8개 있습니다. 시스템에 구성된 풀이 없을 경우 모든 CPU가 **기본 풀**에 속하며 전역 영역에서 사용됩니다. 하지만 이 예에서는 `pool99` 풀이 생성되었으며 `CPU 3`과 `CPU 4`로 구성됩니다. 이 풀은 배타적 영역인 `zone1`과 연결됩니다. `pool99`를 `vnic1`의 등록 정보로 설정하면 `pool99`는 `vnic1`의 네트워킹 프로세스 관리 전용이 됩니다. `vnic1`이 `zone1`의 네트워킹 인터페이스에 할당된 후 `pool99`의 CPU가 `zone1`의 네트워킹 및 비네트워킹 프로세스를 모두 관리하도록 예약됩니다.

pool 등록 정보는 기본적으로 동적입니다. 일정 범위의 CPU로 영역 풀을 구성할 수 있으며 커널에 따라 풀의 CPU 세트에 할당되는 CPU가 결정됩니다. 데이터 링크에 대한 풀 변경 사항은 자동으로 구현되므로 해당 링크의 풀 관리가 간소화됩니다. 반면, cpu 등록 정보를 사용하여 링크에 특정 CPU를 할당하려면 할당할 CPU를 지정해야 합니다. 풀의 CPU 구성 요소를 변경할 때마다 cpu 등록 정보를 설정해야 합니다.

예를 들어, 그림 21–1의 시스템에서 CPU 4는 오프라인 상태로 전환되었습니다. pool 등록 정보는 동적이기 때문에 소프트웨어가 추가 CPU를 풀과 자동으로 연결합니다. 따라서 CPU 두 개로 이루어진 풀의 원래 구성이 보존됩니다. vnic1의 경우 변경 작업이 투명합니다. 다음 그림에서는 조정된 구성을 보여줍니다.

그림 21–2 pool 등록 정보의 자동 재구성



추가 풀 관련 등록 정보는 데이터 링크의 CPU 또는 CPU 풀 사용에 대한 정보를 표시합니다. 이러한 등록 정보는 읽기 전용이며 관리자가 설정할 수 없습니다.

- pool-effective는 네트워크 프로세스에 사용 중인 풀을 표시합니다.
- cpus-effective는 네트워크 프로세스에 사용 중인 CPU 목록을 표시합니다.

일반적으로 영역의 CPU 리소스를 관리하기 위해 초기 단계로 데이터 링크의 pool 등록 정보를 설정하지는 않습니다. 대체로 zonecfg 및 poolcfg와 같은 명령은 리소스 풀을 사용하도록 영역을 구성하는 데 사용됩니다. cpu 및 pool 링크 등록 정보 자체는 설정되지 않습니다. 이 경우 데이터 링크의 pool-effective 및 cpus-effective 등록 정보는 영역을 부트할 때 해당 영역의 구성에 따라 자동으로 설정됩니다. 기본 풀은 pool-effective 아래에 표시되고 cpus-effective 값은 시스템에서 선택합니다. 따라서 dladm show-linkprop 명령을 사용하는 경우 pool 및 cpu 등록 정보는 비어 있지만 pool-effective 및 cpus-effective 등록 정보에는 값이 포함됩니다.

영역의 CPU 풀을 네트워킹 프로세스에 바인딩하는 대신 데이터 링크의 pool 및 cpu 등록 정보를 직접 설정할 수도 있습니다. 이러한 등록 정보를 구성하면 해당 값이 pool-effective 및 cpus-effective 등록 정보에도 반영됩니다. 하지만 이 대체 단계는 영역의 네트워크 리소스 관리에 자주 사용되지 않습니다.

▼ 데이터 링크에 대한 CPU 풀을 구성하는 방법

다른 링크 등록 정보와 마찬가지로, 링크를 만들 때 또는 나중에 링크의 추가 구성이 필요할 때 데이터 링크에 대해 `pool` 등록 정보를 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
# dladm create-vnic -p pool=pool-name -l link vnic
```

VNIC를 만드는 동안 `pool` 등록 정보를 설정합니다. 기존 VNIC의 `pool` 등록 정보를 설정하려면 다음 구문을 사용합니다.

```
# dladm setlinkprop -p pool=pool-name vnic
```

다음 절차에서는 VNIC에 대해 CPU 풀을 구성하는 단계를 제공합니다.

시작하기 전에 다음을 완료한 상태여야 합니다.

- 할당된 개수의 CPU를 사용하여 프로세서 세트를 생성했습니다.
- 프로세서 세트가 연결될 풀을 생성했습니다.
- 프로세서 세트와 풀을 연결했습니다.

주 – 이러한 필수 조건을 완료하는 지침은 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 "구성을 수정하는 방법"을 참조하십시오.

---

1 링크의 `pool` 등록 정보를 영역에 대해 만든 CPU 풀로 설정합니다. VNIC의 존재 여부에 따라 다음 단계 중 하나를 수행합니다.

- VNIC가 아직 생성되지 않은 경우 다음 구문을 사용합니다.

  ```
  # dladm create-vnic -l link -p pool=pool vnic
  ```

  여기서 *pool*은 영역에 대해 생성된 풀의 이름을 나타냅니다.
- VNIC가 있는 경우 다음 구문을 사용합니다.

  ```
  # dladm setlinkprop -p pool=pool vnic
  ```

2 VNIC를 사용하도록 영역을 설정합니다.

```
zonecfg>zoneid:net> set physical=vnic
```

주 – 영역에 네트워킹 인터페이스를 할당하는 방법을 설명하는 전체 단계는 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 "영역 구성, 확인 및 커밋"을 참조하십시오.

예 21–6 배타적 IP 유형을 사용하여 영역에 링크의 CPU 풀 할당

이 예에서는 영역의 데이터 링크에 풀이 할당되는 방식을 보여줍니다. 이 시나리오는 그림 21–1의 구성을 기반으로 합니다. 이 예에서는 pool99라는 CPU 풀이 영역에 대해 이미 구성되었다고 가정합니다. 그런 다음 VNIC에 풀이 할당됩니다. 최종적으로, 비전역 영역 zone1이 VNIC를 네트워킹 인터페이스로 사용하도록 설정됩니다.

```
# dladm create-vnic -l e1000g0 -p pool99 vnic0

# zonecfg -c zone1
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1>net> set physical=vnic0
zonecfg:zone1>net> end
zonecfg:zone1> exit
```

▼ 링크에 CPU를 할당하는 방법

다음 절차에서는 cpu 등록 정보를 구성하여 데이터 링크를 순회하는 트래픽을 처리하도록 특정 CPU를 할당하는 방법에 대해 설명합니다.

1 인터페이스에 대한 CPU 할당을 확인합니다.

```
# dladm show-linkprop -p cpus link
```

기본적으로 특정 인터페이스에는 CPU가 할당되지 않습니다. 따라서 명령 출력 결과의 VALUE 매개변수에는 항목이 포함되지 않습니다.

2 인터럽트 및 인터럽트가 연결된 CPU를 나열합니다.

```
# echo ::interrupts | mdb -k
```

CPU 번호를 포함하여 시스템의 각 링크에 대한 매개변수가 출력 결과에 나열됩니다.

3 링크에 CPU를 할당합니다.

CPU는 링크의 인터럽트가 연결된 CPU를 포함할 수 있습니다.

```
# dladm set-linkprop -p cpus=cpu1,cpu2,... link
```

여기서 *cpu1*은 링크에 할당할 CPU 번호입니다. 링크 전용으로 여러 CPU를 지정할 수 있습니다.

4 링크 인터럽트를 검사하여 새 CPU 할당을 확인합니다.

```
# echo ::interrupts | mdb -k
```

5 (옵션) 링크와 연결된 CPU를 표시합니다.

```
# dladm show-linkprop -p cpus link
```

예 21–7 인터페이스에 CPU 할당

이 예에서는 그림 18–3의 internal0 인터페이스에 특정 CPU를 전용으로 지정하는 방법을 보여줍니다.

여러 명령으로 생성된 출력 결과에서 다음 정보를 확인합니다. 알아보기 쉽도록 출력 결과에서 중요한 정보가 강조 표시됩니다.

- 기본적으로 internal0에는 전용 CPU가 없습니다. 따라서 VALUE는 -- 입니다.
- internal0의 인터럽트는 CPU 18과 연결됩니다.
- CPU가 할당된 후 internal0은 VALUE 아래에 새 CPU 목록을 표시합니다.

```
# dladm show-linkprop -p cpus internal0
LINK          PROPERTY     PERM     VALUE     DEFAULT     POSSIBLE
internal0     cpus         rw       --        --          --

# echo ::interrupts | mdb -k
    Device   Shared   Type   MSG #   State   INO    Mondo   Pil   CPU
external#0   no       MSI    3       enbl    0x1b   0x1b    6     0
internal#0   no       MSI    2       enbl    0x1a   0x1a    6     18

# dladm set-linkprop -p cpus=14,18,19,20 internal0

# dladm show-linkprop -p cpus internal0
LINK        PROPERTY   PERM   VALUE        DEFAULT   POSSIBLE
internal0   cpus       rw     14,18,19,20  --        --
```

인터럽트가 포함된 모든 지원 스레드가 이제 새로 할당된 CPU 세트로 제한됩니다.

흐름의 리소스 관리

흐름은 속성에 따라 구성된 네트워크 패킷으로 구성됩니다. 흐름을 사용하여 네트워크 리소스를 더 할당할 수 있습니다. 흐름 개요는 358 페이지 “흐름을 사용한 네트워크 리소스 관리”를 참조하십시오.

리소스 관리 흐름을 사용하려면 다음 일반 단계를 수행합니다.

1. 358 페이지 “흐름을 사용한 네트워크 리소스 관리”에 나열된 대로 특정 속성에 기반을 두도록 흐름을 만듭니다.
2. 네트워크 리소스와 관련된 등록 정보를 설정하여 흐름의 리소스 사용을 사용자 정의합니다. 현재 패킷 처리를 위한 대역폭만 설정할 수 있습니다.

네트워크의 흐름 구성

물리적 네트워크와 가상 네트워크에 흐름을 만들 수 있습니다. 흐름을 구성하려면 `flowadm` 명령을 사용합니다. 자세한 기술 정보는 `flowadm(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 흐름을 구성하는 방법

1 (옵션) 흐름을 구성할 링크를 결정합니다.

```
# dladm show-link
```

2 선택한 링크의 IP 인터페이스가 IP 주소로 올바르게 구성되었는지 확인합니다.

```
# ipadm show-addr
```

3 각 흐름에 대해 결정한 속성에 따라 흐름을 만듭니다.

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] flow
```

attribute 네트워크 패킷을 흐름으로 구성할 수 있는 다음 분류 중 하나를 나타냅니다.

- IP 주소
- 전송 프로토콜(UDP, TCP 또는 SCTP)
- 응용 프로그램의 포트 번호(예: FTP의 경우 포트 21)
- IPv6 패킷의 서비스 품질에만 사용되는 DS 필드 속성. DS 필드에 대한 자세한 내용은 **Oracle Solaris 관리: IP 서비스**의 "DS 코드 포인트"을 참조하십시오.

flow 특정 흐름에 지정하는 이름을 나타냅니다.

흐름 및 흐름 속성에 대한 자세한 내용은 `flowadm(1M)` 매뉴얼 페이지를 참조하십시오.

4 적절한 흐름 등록 정보를 설정하여 흐름에 리소스 제어를 구현합니다.

```
# flowadm set-flowprop -p property=value[,property=value,...] flow
```

리소스를 제어하는 다음 흐름 등록 정보를 지정할 수 있습니다.

`maxbw` 이 흐름으로 식별된 패킷이 사용할 수 있는 링크 대역폭의 최대 크기입니다. 설정한 값이 링크 대역폭에 허용되는 값의 범위 내에 있어야 합니다. 링크 대역폭에 가능한 값의 범위를 표시하려면 다음 명령으로 생성된 출력 결과의 `POSSIBLE` 필드를 확인합니다.

```
# dladm show-linkprop -p maxbw link
```

주 – 현재 흐름의 대역폭만 사용자 정의할 수 있습니다.

---

5 **(옵션) 링크에 만든 흐름을 표시합니다.**

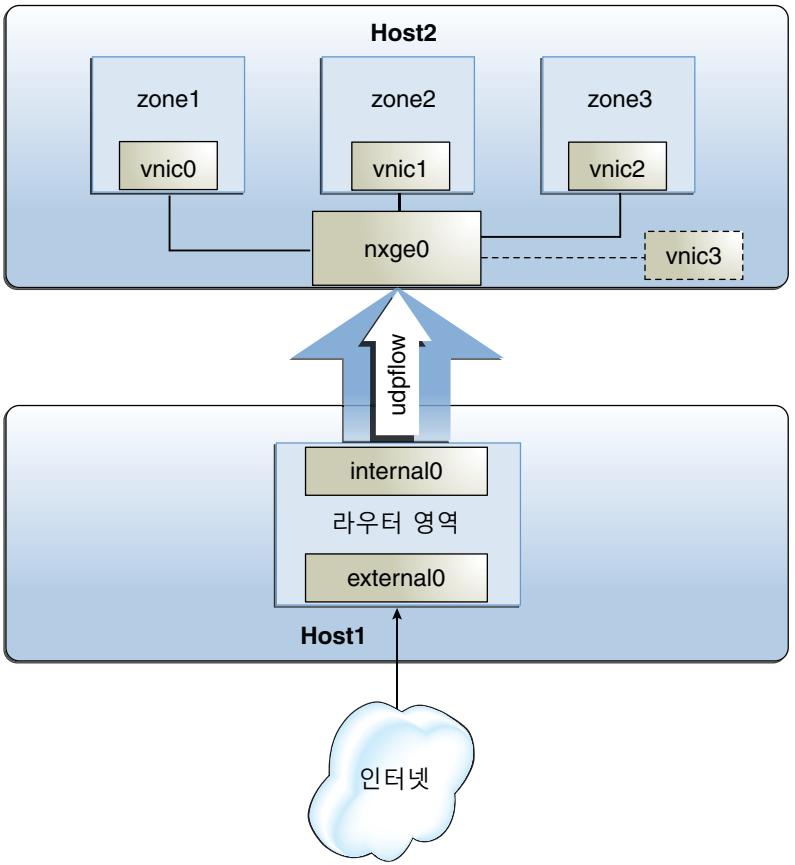
```
# flowadm show-flow -l link
```

6 **(옵션) 지정한 흐름에 대한 등록 정보 설정을 표시합니다.**

```
# flowadm show-flowprop flow
```

예 21–8 링크 및 흐름 등록 정보를 설정하여 리소스 관리

이 예에서는 데이터 링크와 흐름 모두에 네트워크 리소스를 할당하는 단계를 결합합니다. 이 예는 다음 그림에 표시된 구성을 기반으로 합니다.



또한 서로 연결된 두 개의 물리적 호스트를 보여줍니다.

- `Host1`의 구성은 다음과 같습니다.

- 라우터 영역으로 사용되는 비전역 영역이 한 개 있습니다. 이 영역에는 인터페이스가 두 개 할당됩니다. `external0`은 인터넷에 연결하고 `internal0`은 두번째 호스트를 포함하는 내부 네트워크에 연결합니다.
- 사용자 정의 이름을 사용하도록 IP 인터페이스의 이름이 바뀌었습니다. 필수는 아니지만 링크와 인터페이스에 사용자 정의 이름을 사용하면 네트워크를 관리할 때 도움이 됩니다. 24 페이지 “네트워크 장치 및 데이터 링크 이름”을 참조하십시오.
- `internal0`에 흐름을 구성하여 UDP 트래픽을 격리시키고 UDP 패킷의 리소스 사용 방식에 대한 제어를 구현합니다. 흐름 구성에 대한 자세한 내용은 378 페이지 “흐름의 리소스 관리”를 참조하십시오.

- `Host2`의 구성은 다음과 같습니다.
  - 비전역 영역 세 개와 해당 VNIC가 있습니다. VNIC는 동적 링 할당을 지원하는 `nxge` 카드에 구성됩니다. 링 할당에 대한 자세한 내용은 360 페이지 “전송 및 수신 링”을 참조하십시오.
  - 각 영역의 네트워크 처리 부하는 서로 다릅니다. 이 예에서는 `zone1`의 부하는 높고 `zone2`의 부하는 중간이고 `zone3`의 부하는 낮습니다. 해당 부하에 따라 이러한 영역에 리소스가 할당됩니다.
  - 별도의 VNIC가 소프트웨어 기반 클라이언트로 구성됩니다. MAC 클라이언트의 개요는 360 페이지 “MAC 클라이언트 및 링 할당”을 참조하십시오.

이 예의 작업은 다음과 같습니다.

- 흐름 만들기 및 흐름 제어 구성 – `internal0`에 흐름이 생성되어 `Host2`에 수신되는 UDP 패킷에 대해 별도의 리소스 제어를 만듭니다.
- `Host2`에서 VNIC에 대한 네트워크 리소스 등록 정보 구성 – 각 영역의 처리 부하를 기준으로 각 영역의 VNIC가 전용 링 세트로 구성됩니다. 또한 소프트웨어 기반 클라이언트의 예로 전용 링 없이 별도의 VNIC가 구성됩니다.

이 예에 영역 구성에 대한 절차는 포함되지 않습니다. 영역을 구성하려면 **Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리**의 17 장, “비전역 영역 계획 및 구성(작업)”을 참조하십시오.

먼저 `Host1`의 링크 및 IP 인터페이스에 대한 정보를 확인합니다.

```
# dladm show-phys
LINK         MEDIA      STATE      SPEED DUPLEX   DEVICE
internal0    Ethernet   up         1000 full      nge1
e1000g0      n          unknown    0    half      e1000g0
e1000g1      n          unknown    0    half      e1000g1
external0    Ethernet   up         1000 full      nge0

# dladm show-link
LINK        CLASS    MTU    STATE    BRIDGE  OVER
internal0   phys     1500   up       --      nge1
e1000g0     phys     1500   unknown  --      --
```

```
e1000g1     phys     1500   unknown  --       --
external0   phys     1500   up       --       nge0

# ipadm show-addr
ADDROBJ     TYPE     STATE     ADDR
lo0/4       static   ok        127.0.0.1/8
external0   static   ok        10.10.6.5/24
internal0   static   ok        10.10.12.42/24
```

다음은 internal0에 흐름을 만들어 UDP 트래픽을 Host2로 격리시킵니다. 흐름에 대한 리소스 제어를 구현합니다.

```
# flowadm add-flow -l external0 -a transport=udp udpflow
# flowadm set-flowprop -p maxbw=80 udpflow
```

만든 흐름에 대한 정보를 확인합니다.

```
flowadm show-flow
FLOW        LINK        IPADDR   PROTO   PORT   DFSLD
udpflow     internal0   --       udp     --     --

# flowadm show-flowprop
SECURE OUTPUT FOR THIS
```

Host2에서 nxge0에 각 영역에 대한 VNIC를 구성합니다. 각 VNIC에 대한 리소스 제어를 구현합니다. 해당 영역에 VNIC를 할당합니다.

```
# dladm create-vnic -l nxge0 vnic0
# dladm create-vnic -l nxge0 vnic1
# dladm create-vnic -l nxge0 vnic2

# dladm set-prop -p rxrings=4,txrings=4 vnic0
# dladm set-prop -p rxrings=2,txrings=2 vnic1
# dladm set-prop -p rxrings=1,txrings=1 vnic2

# zone1>zonecfg>net> set physical=vnic0
# zone2>zonecfg>net> set physical=vnic1
# zone3>zonecfg>net> set physical=vnic2
```

Host2의 CPU 세트인 pool1이 이전에 zone1에서 사용하도록 구성되었다고 가정합니다. 다음과 같이 해당 CPU 풀을 바인딩하여 zone1에 대한 네트워크 프로세스도 관리합니다.

```
# dladm set-prop -p pool=pool01 vnic0
```

최종적으로, 주 인터페이스인 nxge0과 링을 공유하는 소프트웨어 기반 클라이언트를 만듭니다.

```
dladm create-vnic -p rxrings=sw,txrings=sw -l nxge0 vnic3
```

◆ ◆ ◆ 22장

22 네트워크 트래픽 및 리소스 사용 모니터링

이 장에서는 물리적 및 가상 네트워크 환경에서 네트워크 리소스의 사용에 대한 통계를 모니터하고 수집하는 작업에 대해 설명합니다. 이 정보는 프로비저닝, 통합 및 청구 용도에 대한 리소스 할당을 분석하는 데 도움이 됩니다. 이 장에서는 통계를 표시하는 데 사용하는 두 가지 명령인 `dlstat` 및 `flowstat`를 소개합니다.

다음 항목을 다룹니다.

네트워크 트래픽 흐름 개요

패킷은 시스템에 들어오고 나갈 때 경로를 순회합니다. 세부적인 레벨에서 패킷은 NIC의 수신(Rx) 링과 전송(Tx) 링을 통해 수신 및 전송됩니다. 이러한 링에서 수신된 패킷은 추가 처리를 위해 네트워크 스택의 위로 전달되고 아웃바운드 패킷은 네트워크로 전송됩니다.

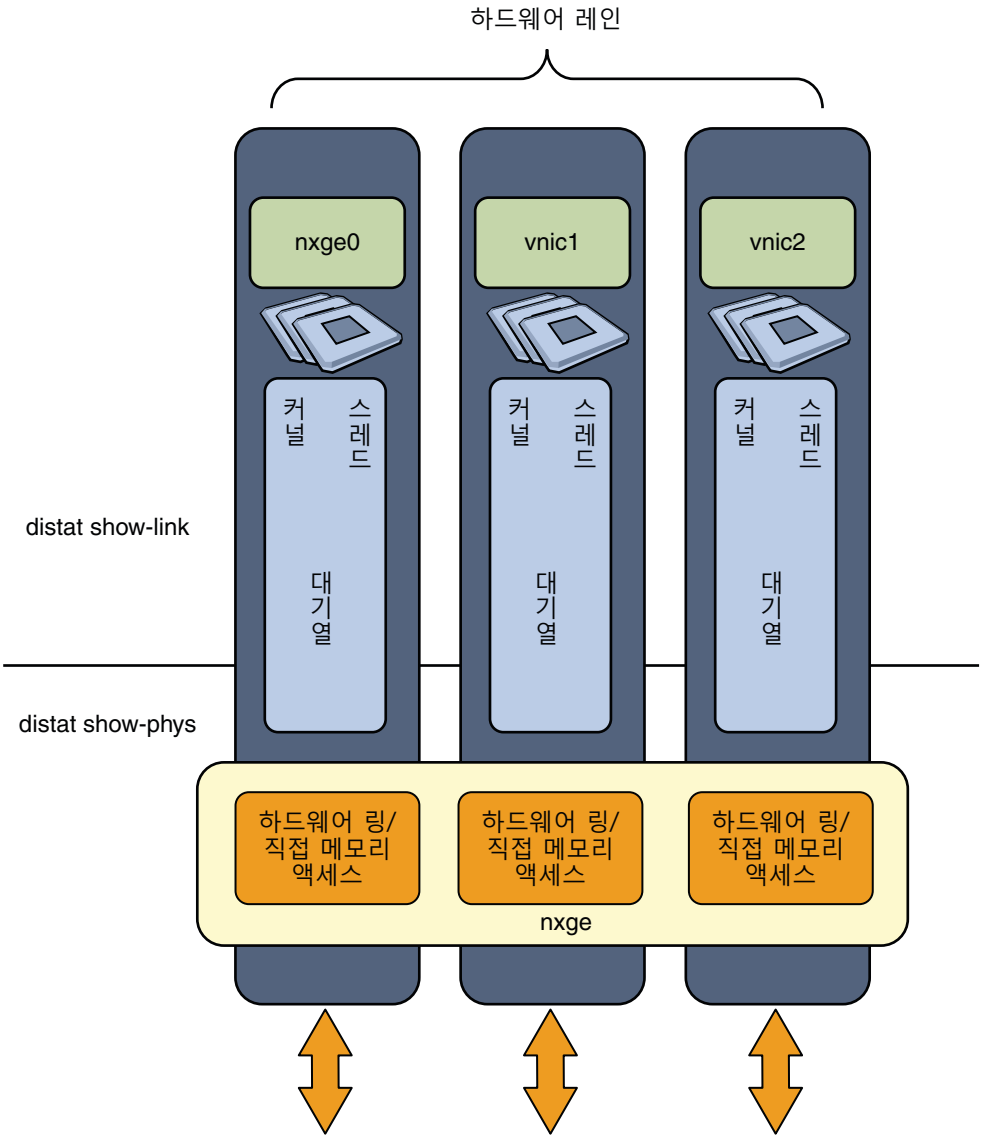
21 장, "네트워크 리소스 관리"에서는 네트워크 레인의 개념을 소개합니다. 네트워크 트래픽을 관리하도록 할당된 시스템 리소스의 조합이 네트워크 레인을 형성합니다. 따라서 **네트워크 레인**은 특정 네트워크 트래픽 유형에 대한 사용자 정의 경로입니다. 각 레인은 **하드웨어** 레인 또는 **소프트웨어** 레인일 수 있습니다. 또한 각 레인 유형은 **수신** 레인 또는 **전송** 레인일 수 있습니다. 하드웨어 및 소프트웨어 레인은 NIC가 링 할당을 지원할 수 있는지 여부에 따라 구분됩니다. 링 할당에 대한 자세한 내용은 360 페이지 "전송 및 수신 링"을 참조하십시오. 이 장에서는 주로 수신 레인을 통해 받은 수신 트래픽에 중점을 둡니다.

하드웨어 레인에서는 링이 해당 레인을 사용하는 패킷에 전용으로 지정됩니다. 반면, 소프트웨어 레인의 링은 데이터 링크 간에 공유됩니다. 데이터 링크는 다음과 같은 이유로 링을 공유하도록 구성됩니다.

- 관리 의도. 데이터 링크가 전용 링이 필요한 집중적 프로세스를 수행하지 않을 수도 있습니다.
- NIC는 링 할당을 지원하지 않습니다.
- 링 할당을 지원하지만 배타적 사용을 위해 링을 더 이상 할당할 수 없습니다.

다양한 하드웨어 레인을 보여주는 다음 그림을 고려해 보십시오.

그림 22–1 하드웨어 레인



이 그림에서는 다음 구성을 보여줍니다.

- 시스템에 `nxge`라는 단일 NIC가 있습니다.
- 물리적 장치에 `nxge0`, `vnic1` 및 `vnic2` 링크가 구성됩니다. 데이터 링크인 `nxge0`에는 사용자 정의 이름이 지정될 수 있습니다. 하지만 이 그림에서는 링크가 기본 장치 이름을 유지합니다.

- 시스템에 CPU가 여러 개 있습니다.
- NIC가 동적 링 할당을 지원합니다. 따라서 각 링크에 하드웨어 링 세트를 할당하여 하드웨어 레인을 형성할 수 있습니다. 또한 각 레인에 CPU 세트가 할당됩니다.

트래픽 및 리소스 사용 모니터링(작업 맵)

네트워크 레인의 패킷 흐름을 관찰하여 패킷이 네트워크 리소스를 사용하는 방식에 대한 정보를 가져올 수 있습니다. `dlstat` 명령은 데이터 링크에 대한 이 정보를 제공합니다. `flowstat` 명령은 기존 흐름에 대해 유사한 기능을 수행합니다.

다음 표에서는 네트워크 트래픽 및 시스템의 리소스 사용에 대한 통계를 가져오는 데 사용할 수 있는 여러 방법을 보여줍니다.

작업	설명	수행 방법
네트워크 트래픽에 대한 통계 정보를 가져옵니다.	시스템의 네트워크 인터페이스에서 수신 및 송신 트래픽을 확인합니다.	387 페이지 "네트워크 트래픽에 대한 기본 통계를 가져오는 방법"
링 사용에 대한 통계 정보를 가져옵니다.	수신 및 송신 트래픽이 NIC의 링에 분산되는 방식을 확인합니다.	389 페이지 "링 사용에 대한 통계를 가져오는 방법"
특정 레인의 네트워크 트래픽에 대한 통계 정보를 가져옵니다.	시스템의 네트워크 인터페이스에 구성된 네트워크 레인을 패킷이 순회할 때 수신 및 송신 트래픽에 대한 자세한 정보를 확인합니다.	390 페이지 "레인의 네트워크 트래픽에 대한 통계를 가져오는 방법"
흐름의 트래픽에 대한 통계 정보를 가져옵니다.	사용자 정의 흐름을 순회하는 수신 및 송신 트래픽에 대한 정보를 확인합니다.	393 페이지 "흐름에 대한 통계를 가져오는 방법"
네트워크 트래픽의 계정을 구성합니다.	계정 용도로 트래픽 정보를 캡처하도록 네트워크 계정을 구성합니다.	395 페이지 "확장 네트워크 계정을 구성하는 방법"
네트워크 트래픽에 대한 기록 통계를 가져옵니다.	확장 네트워크 계정의 로그 파일에서 정보를 추출하여 레인 및 흐름의 네트워크 트래픽에 대한 기록 통계를 가져옵니다.	396 페이지 "네트워크 트래픽에 대한 기록 통계를 가져오는 방법"

흐름 구성 단계에 대한 자세한 내용은 378 페이지 "흐름의 리소스 관리"를 참조하십시오. 이러한 두 명령에 대한 자세한 내용은 `dlstat(1M)` 및 `flowstat(1M)` 매뉴얼 페이지를 참조하십시오.

링크의 네트워크 트래픽에 대한 통계 수집

dlstat 및 flowstat 명령은 각각 데이터 링크와 흐름의 네트워크 트래픽에 대한 통계를 모니터하고 가져오기 위한 도구입니다. 이러한 명령은 dladm 및 flowadm 명령과 비슷합니다. 다음 표에서는 *adm 명령 쌍과 *stat 명령 쌍의 유사점과 해당 기능을 보여줍니다.

관리 명령		모니터링 명령	
명령	**기능**	**명령**	**기능**
dladm 명령 옵션	데이터 링크를 구성 및 관리하기 위한 사용자 인터페이스 및 도구입니다.	dlstat 명령 옵션	데이터 링크의 트래픽 통계를 가져오기 위한 사용자 인터페이스 및 도구입니다.
flowadm 명령 옵션	흐름을 구성 및 관리하기 위한 사용자 인터페이스 및 도구입니다.	flowstat 명령 옵션	흐름의 트래픽 통계를 가져오기 위한 사용자 인터페이스 및 도구입니다.

dlstat 명령의 다음 변형을 사용하여 네트워크 트래픽 정보를 수집할 수 있습니다.

- dlstat – 시스템에서 수신 또는 전송되는 패킷에 대한 일반 정보를 표시합니다.
- dlstat show-phys – 수신 및 전송 링의 사용 정보를 표시합니다. 이 명령은 네트워크 물리적 장치에 대한 비트래픽 정보를 표시하는 dladm show-phys 명령에 해당합니다. 이 명령이 적용되는 네트워크 레인의 레벨을 보려면 그림 22–1을 참조하십시오.
- dlstat show-link – 지정된 레인의 트래픽 흐름에 대한 자세한 정보를 표시합니다. 레인은 해당 데이터 링크로 식별됩니다. 이 명령은 데이터 링크에 대한 비트래픽 정보를 표시하는 dladm show-link 및 dladm show-vnic 명령에 해당합니다. dlstat show-link 명령이 적용되는 네트워크 레인의 레벨을 보려면 그림 22–1을 참조하십시오.
- dlstat show-aggr – 링크 통합의 포트 사용 정보를 표시합니다. 이 명령은 링크 통합에 대한 비트래픽 정보를 표시하는 dladm show-aggr 명령에 해당합니다.

▼ 네트워크 트래픽에 대한 기본 통계를 가져오는 방법

1 관리자로 전환합니다.

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

2 모든 데이터 링크에서 기본 트래픽 흐름을 관찰합니다.

```
# dlstat [-r|-t] [-i interval] [link]
```

`[-r|-t]` 수신측 통계만 표시(`-r` 옵션)하거나 전송측 통계만 표시(`-t` 옵션)합니다. 이러한 옵션을 사용하지 않으면 수신측과 전송측의 통계가 모두 표시됩니다.

`-i` *interval* 표시된 통계를 새로 고칠 시간(초)을 지정합니다. 이 옵션을 사용하지 않으면 정적 출력 결과가 표시됩니다.

link 지정한 데이터 링크의 통계만 모니터함을 나타냅니다. 이 옵션을 사용하지 않으면 모든 데이터 링크에 대한 정보가 표시됩니다.

독립적으로 사용될 경우 `dlstat` 명령은 구성된 모든 데이터 링크의 수신 및 송신 패킷에 대한 정보를 표시합니다.

다음 정보는 `dlstat` 명령과 함께 사용하는 대부분의 옵션에서 표시됩니다.

- IP 인터페이스로 구성되고 트래픽을 수신 또는 전송할 수 있는 시스템의 링크
- 패킷 및 바이트 크기
- 인터럽트 및 MAC 폴링 통계
- 패킷 체인 길이

예 22–1 기본 수신측 및 전송측 통계 표시

이 예에서는 시스템에 구성된 모든 데이터 링크에서 수신 및 전송되는 네트워크 트래픽에 대한 정보를 보여줍니다.

```
# dlstat
   LINK    IPKTS   RBYTES    OPKTS   OBYTES
e1000g0  101.88K   32.86M   40.16K    4.37M
  nxge1    4.50M    6.78G    1.38M   90.90M
  vnic1        8      336        0        0
```

예 22–2 1초 간격으로 수신측 통계 표시

이 예에서는 모든 데이터 링크에서 수신되는 트래픽에 대한 정보를 보여줍니다. 1초마다 정보가 새로 고쳐집니다. 디스플레이의 새로 고침을 중지하려면 Ctrl-C를 누릅니다.

```
# dlstat -r -i 1
   LINK   IPKTS  RBYTES   INTRS   POLLS   CH<10 CH10-50   CH>50
e1000g0 101.91K  32.86M  87.56K  14.35K   3.70K     205       5
  nxge1   9.61M  14.47G   5.79M   3.82M 379.98K  85.66K   1.64K
  vnic1       8     336       0       0       0       0       0
e1000g0       0       0       0       0       0       0       0
  nxge1  82.13K 123.69M  50.00K  32.13K   3.17K     724      24
  vnic1       0       0       0       0       0       0       0
...
^C
```

이 출력 결과에서는 인터럽트(`INTRS`)에 대한 통계가 중요합니다. 인터럽트 수가 적으면 성능 효율성이 더 큽니다. 인터럽트 수가 크면 특정 링크에 리소스를 더 추가해야 할 수 있습니다.

예 22–3 5초 간격으로 전송측 통계 표시

이 예에서는 모든 데이터 링크에서 전송되는 트래픽에 대한 정보를 표시합니다. 5초마다 정보가 새로 고쳐집니다.

```
# dlstat -t -i 5
   LINK   OPKTS  OBYTES  BLKCNT UBLKCNT
e1000g0  40.24K   4.37M       0       0
  nxge1   9.76M 644.14M       0       0
  vnic1       0       0       0       0
e1000g0       0       0       0       0
  nxge1  26.82K   1.77M       0       0
  vnic1       0       0       0       0
...
^C
```

▼ 링 사용에 대한 통계를 가져오는 방법

1 관리자로 전환합니다.

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

2 링 통계를 표시합니다.

```
# dlstat show-phys [-r|-t] [-i interval] [link]
```

`[-r|-t]` 수신측 통계만 표시(`-r` 옵션)하거나 전송측 통계만 표시(`-t` 옵션)합니다. 이러한 옵션을 사용하지 않으면 수신측과 전송측의 통계가 모두 표시됩니다.

`-i` *interval* 표시된 통계를 새로 고칠 시간(초)을 지정합니다. 이 옵션을 사용하지 않으면 정적 출력 결과가 표시됩니다.

link 지정한 데이터 링크의 통계만 모니터함을 나타냅니다. 이 옵션을 사용하지 않으면 모든 데이터 링크에 대한 정보가 표시됩니다.

독립적으로 사용될 경우 `dlstat show-phys` 명령은 구성된 모든 데이터 링크의 수신 및 송신 패킷에 대한 정보를 표시합니다.

예 22–4 데이터 링크에 대한 수신 링 통계 표시

이 예에서는 데이터 링크에 대한 수신 링의 사용을 보여줍니다.

```
# dlstat show-phys -r nxge1
     LINK TYPE INDEX   IPKTS  RBYTES
nxge1   rx     0      21   1.79K
nxge1   rx     1       0       0
nxge1   rx     2   1.39M   2.10G
nxge1   rx     3       0       0
```

```
nxge1   rx     4   6.81M  10.26G
nxge1   rx     5   4.63M   6.97G
nxge1   rx     6   3.97M   5.98G
nxge1   rx     7       0       0
```

nxge 장치에는 수신 링 8개가 있으며, 각 링은 INDEX 필드에서 식별됩니다. 링당 패킷의 균일한 배포는 링이 링크의 부하에 따라 링크에 올바르게 할당되었음을 나타내는 이상적인 구성입니다. 균일하지 않은 배포는 링크당 링의 부적절한 배포를 나타낼 수 있습니다. 해결 방법은 NIC가 링크당 링의 재배포를 허용하는 동적 링 할당을 지원하는지 여부에 따라 달라집니다. 동적 링 할당에 대한 자세한 내용은 360 페이지 "전송 및 수신 링"을 참조하십시오.

예 22–5 데이터 링크에 대한 전송 링 통계 표시

이 예에서는 데이터 링크에 대한 전송 링의 사용을 보여줍니다.

```
# dlstat show-phys -t nxge1
 LINK TYPE INDEX   OPKTS   OBYTES
nxge1   tx     0      44    3.96K
nxge1   tx     1       0        0
nxge1   tx     2   1.48M  121.68M
nxge1   tx     3   2.45M  201.11M
nxge1   tx     4   1.47M  120.82M
nxge1   tx     5       0        0
nxge1   tx     6   1.97M  161.57M
nxge1   tx     7   4.59M  376.21M
nxge1   tx     8   2.43M  199.24M
nxge1   tx     9       0        0
nxge1   tx     10  3.23M  264.69M
nxge1   tx     11  1.88M  153.96M
```

▼ 레인의 네트워크 트래픽에 대한 통계를 가져오는 방법

1 관리자로 전환합니다.

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

2 네트워크 레인에 대한 통계를 표시합니다.

```
# dlstat show-link [-r [F]|-t] [-i interval] [link]
```

`[-r|-t]` 수신측 통계만 표시(`-r` 옵션)하거나 전송측 통계만 표시(`-t` 옵션)합니다. 이러한 옵션을 사용하지 않으면 수신측과 전송측의 통계가 모두 표시됩니다.

`-i` *interval* 표시된 통계를 새로 고칠 시간(초)을 지정합니다. 이 옵션을 사용하지 않으면 정적 출력 결과가 표시됩니다.

link 지정한 데이터 링크의 통계만 모니터함을 나타냅니다. 이 옵션을 사용하지 않으면 모든 데이터 링크에 대한 정보가 표시됩니다.

링 그룹화가 지원되며 전용 링이 구성된 경우 하드웨어 레인 통계가 표시됩니다. 전용 링이 구성되지 않은 경우 소프트웨어 레인 통계가 표시됩니다.

예 22–6 레인에 대한 수신측 통계 표시

이 예에서는 다음 정보를 보여줍니다.

- 하드웨어 레인에서 패킷이 수신되는 방식
- 소프트웨어 레인에서 패킷이 수신되는 방식
- 소프트웨어 레인에서 패킷이 수신되고 할당된 CPU로 팬아웃되는 방식

다음 명령에서는 특정 링크에 대한 수신측 통계를 보여줍니다. 이 정보는 링 사용을 나타냅니다. 하지만 데이터가 대역폭 제한 및 우선 순위 처리와 같은 다른 리소스 할당의 구현을 반영할 수도 있습니다.

```
# dlstat show-link -r nxge1
 LINK TYPE    ID INDEX  IPKTS   RBYTES  INTRS   POLLS   CH<10 CH10-50  CH>50
nxge1   rx local    --      0        0      0       0       0       0      0
nxge1   rx    hw     1      0        0      0       0       0       0      0
nxge1   rx    hw     2  1.73M    2.61G  1.33M 400.22K  67.03K   7.49K     38
nxge1   rx    hw     3      0        0      0       0       0       0      0
nxge1   rx    hw     4  8.44M   12.71G  4.35M   4.09M 383.28K  91.24K  2.09K
nxge1   rx    hw     5  5.68M    8.56G  3.72M   1.97M 203.68K  43.94K    854
nxge1   rx    hw     6  4.90M    7.38G  3.11M   1.80M 168.59K  42.34K    620
nxge1   rx    hw     7      0        0      0       0       0       0      0
```

다음 명령에서는 특정 링크에 대한 수신측 통계를 보여줍니다. 출력 결과에서 `ID` 필드는 하드웨어 링이 배타적으로 할당되었는지 또는 클라이언트 간에 공유되는지를 나타냅니다. `ixgbe` 카드에서 Rx 링은 VNIC와 같은 다른 클라이언트가 링크에 구성된 경우에도 공유됩니다. 따라서 이 특정 예에서는 Rx 링이 `ID` 필드에 `sw` 값으로 표시된 대로 공유됩니다.

```
# dlstat show-link -r ixgbe0
  LINK TYPE    ID INDEX   IPKTS  RBYTES    INTRS  POLLS  CH<10 CH10-50  CH>50
ixgbe0   rx local    --       0       0        0      0      0       0      0
ixgbe0   rx    sw    -- 794.28K   1.19G  794.28K      0      0       0      0
```

다음 명령에서는 특정 링크에 대한 수신측 통계의 사용을 보여줍니다. 또한 명령에 `-F` 옵션을 사용하면 출력 결과에서 팬아웃 정보도 제공합니다. 구체적으로 팬아웃 수는 2(`0` 및 `1`)입니다. 링 `0`을 사용하는 하드웨어 레인에서 수신된 네트워크 트래픽은 분할되어 두 개의 팬아웃을 통해 전달됩니다. 마찬가지로, 링 `1`을 사용하는 하드웨어 레인에서 수신된 네트워크 트래픽도 분할되어 두 개의 팬아웃으로 나뉩니다.

```
# dlstat show-link -r -F nxge1
 LINK     ID   INDEX   FOUT   IPKTS
```

```
nxge1  local      --      0       0
nxge1     hw      0       0  382.47K
nxge1     hw      0       1        0
nxge1     hw      1       0  367.50K
nxge1     hw      1       1  433.24K
```

예 22–7 레인에 대한 전송측 통계 표시

다음 예에서는 특정 레인의 송신 패킷에 대한 통계를 보여줍니다.

```
# dlstat show-link -t nxge1
 LINK    TYPE   ID  INDEX    OPKTS    OBYTES   BLKCNT   UBLKCNT
nxge1     tx    hw      0       32     1.44K        0         0
nxge1     tx    hw      1        0         0        0         0
nxge1     tx    hw      2    1.48M    97.95M        0         0
nxge1     tx    hw      3    2.45M   161.87M        0         0
nxge1     tx    hw      4    1.47M    97.25M        0         0
nxge1     tx    hw      5        0       276        0         0
nxge1     tx    hw      6    1.97M   130.25M        0         0
nxge1     tx    hw      7    4.59M   302.80M        0         0
nxge1     tx    hw      8    2.43M   302.80M        0         0
nxge1     tx    hw      9        0         0        0         0
nxge1     tx    hw      10   3.23M   213.05M        0         0
nxge1     tx    hw      11   1.88M   123.93M        0         0
```

흐름의 네트워크 트래픽에 대한 통계 수집

흐름 통계는 시스템에 정의된 모든 흐름의 패킷 트래픽을 평가하는 데 도움이 됩니다. 흐름 정보를 가져오려면 `flowstat` 명령을 사용합니다. 이 명령에 대한 자세한 내용은 `flowstat(1M)` 매뉴얼 페이지를 참조하십시오.

자주 사용하는 `flowstat` 명령 구문은 다음과 같습니다.

`# flowstat [-r|-t] [-i` *`interval`*`] [-l` *`link flow`*`]`

`[-r|-t]` 수신측 통계만 표시(`-r` 옵션)하거나 전송측 통계만 표시(`-t` 옵션)합니다. 이러한 옵션을 사용하지 않으면 수신측과 전송측의 통계가 모두 표시됩니다.

`-i` *`interval`* 표시된 통계를 새로 고칠 시간(초)을 지정합니다. 이 옵션을 사용하지 않으면 정적 출력 결과가 표시됩니다.

`link` 지정한 데이터 링크의 모든 흐름에 대한 통계를 모니터함을 나타냅니다. 이 옵션을 사용하지 않으면 모든 데이터 링크의 모든 흐름에 대한 정보가 표시됩니다.

`flow` 지정한 흐름의 통계만 모니터함을 나타냅니다. 이 옵션을 사용하지 않으면 링크를 지정했는지 여부에 따라 모든 흐름 통계가 표시됩니다.

▼ 흐름에 대한 통계를 가져오는 방법

시작하기 전에 네트워크 구성에 흐름이 있는 경우에만 `flowstat` 명령을 사용할 수 있습니다. 흐름을 구성하려면 21 장, "네트워크 리소스 관리"를 참조하십시오.

1 이전에 흐름 제어를 구성한 시스템에서 전역 영역의 관리자가 됩니다.

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

2 흐름의 네트워크 트래픽을 관찰하는 방법의 샘플링을 보려면 다음 명령 중 하나를 수행합니다.

- 모든 흐름의 수신 및 송신 패킷에 대한 통계를 표시합니다.

  ```
  # flowstat
  ```

  이 명령은 구성된 모든 흐름의 트래픽 정보를 정적으로 표시합니다.
- 지정한 간격으로 모든 흐름의 기본 네트워크 트래픽 통계를 표시합니다.

  ```
  # flowstat -i interval
  ```

  Ctrl-C를 눌러 출력 결과 생성을 중지할 때까지 지정한 간격으로 통계 표시가 새로 고쳐집니다.
- 지정한 데이터 링크에 구성된 모든 흐름의 수신 패킷에 대한 통계를 표시합니다.

  ```
  # flowstat -r -l link
  ```
- 지정한 간격으로 지정한 흐름의 송신 패킷에 대한 통계를 표시합니다.

  ```
  # flowstat -t -i interval flow
  ```

예 22–8 1초 간격으로 모든 흐름에 대한 트래픽 통계 표시

이 예에서는 시스템에 구성된 모든 흐름의 수신 및 송신 트래픽에 대한 정보를 매 1초마다 보여줍니다.

```
# flowstat -i 1
FLOW     IPKTS   RBYTES  IERRS    OPKTS  OBYTES  OERRS
flow1  528.45K  787.39M      0  179.39K  11.85M      0
flow2  742.81K    1.10G      0        0       0      0
flow3        0        0      0        0       0      0
flow1   67.73K  101.02M      0   21.04K    1.39M      0
flow2        0        0      0        0       0      0
flow3        0        0      0        0       0      0
...
^C
```

예 22–9 모든 흐름에 대한 전송측 통계 표시

```
# flowstat -t
 FLOW    OPKTS   OBYTES   OERRS
flow1   24.37M    1.61G       0
flow2        0        0       0
flow1        4      216       0
```

예 22–10 지정한 링크의 모든 흐름에 대한 수신측 통계 표시

이 예에서는 net0 데이터 링크에 생성된 모든 흐름에서 하드웨어 레인의 수신 트래픽을 보여줍니다.

```
# flowstat -r -i 2 -l net0
    FLOW    IPKTS   RBYTES    IERRS
tcp-flow  183.11K  270.24M        0
udp-flow        0        0        0
tcp-flow  373.83K  551.52M        0
udp-flow        0        0        0
tcp-flow  372.35K  549.04M        0
udp-flow        0        0        0
tcp-flow  372.87K  549.61M        0
udp-flow        0        0        0
tcp-flow  371.57K  547.89M        0
udp-flow        0        0        0
tcp-flow  191.92K  282.95M        0
udp-flow  206.51K  310.70M        0
tcp-flow        0        0        0
udp-flow  222.75K  335.15M        0
tcp-flow        0        0        0
udp-flow  223.00K  335.52M        0
tcp-flow        0        0        0
udp-flow  160.22K  241.07M        0
tcp-flow        0        0        0
udp-flow  167.89K  252.61M        0
tcp-flow        0        0        0
udp-flow    9.52K   14.32M        0
^C
```

네트워크 계정 설정

확장 계정 기능을 사용하여 네트워크 트래픽에 대한 통계를 로그 파일에 캡처할 수 있습니다. 이런 방식으로 트래픽 레코드를 추적, 프로비저닝, 통합 및 청구 용도로 유지 관리할 수 있습니다. 나중에 로그 파일을 참조하여 일정 기간의 네트워크 사용에 대한 기록 정보를 확인할 수 있습니다.

확장 계정 기능을 구성하려면 acctadm 명령을 사용합니다.

▼ 확장 네트워크 계정을 구성하는 방법

1 네트워크 사용을 추적하려는 인터페이스가 있는 시스템에서 관리자가 됩니다.

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

2 시스템에서 확장 네트워크 계정의 상태를 확인합니다.

```
# acctadm net
```

`acctadm` 명령을 통해 네 가지 유형의 확장 계정을 사용으로 설정할 수 있습니다.

- 프로세스 계정
- 작업 계정
- IPQoS(IP 서비스 품질)의 흐름 계정
- 링크 및 흐름의 네트워크 계정

`net`을 지정하면 네트워크 계정의 상태가 표시됩니다. `net`을 사용하지 않으면 네 가지 계정 유형의 상태가 모두 표시됩니다.

주 – 네트워크 계정은 378 페이지 "흐름의 리소스 관리"에 설명된 대로 `flowadm` 및 `flowstat` 명령을 통해 관리되는 흐름에도 적용됩니다. 따라서 이러한 흐름의 계정을 설정하려면 `acctadm` 명령에 `net` 옵션을 사용합니다. 흐름 계정을 사용으로 설정하고 IPQoS 구성에 적용되는 `flow` 옵션을 사용하지 **마십시오**.

---

3 네트워크 트래픽에 대해 확장 계정을 사용으로 설정합니다.

```
# acctadm -e extended -f filename net
```

여기서 *filename*에는 네트워크 트래픽 통계를 캡처할 로그 파일의 전체 경로가 포함됩니다. 지정한 임의 디렉토리에 로그 파일을 만들 수 있습니다.

4 확장 네트워크 계정이 활성화되었는지 확인합니다.

```
# acctadm net
```

예 22–11 네트워크 트래픽에 대해 확장 계정 구성

이 예에서는 시스템에 구성된 흐름 및 데이터 링크의 네트워크 트래픽에 대한 기록 정보를 캡처하고 표시하는 방법을 보여줍니다.

먼저 다음과 같이 모든 계정 유형의 상태를 확인합니다.

```
# acctadm
           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
             Flow accounting: inactive
        Flow accounting file: none
      Tracked flow resources: none
    Untracked flow resources: extended
             Network accounting: inactive
        Network accounting file: none
      Tracked Network resources: none
    Untracked Network resources: extended
```

네트워크 계정이 활성화되지 않았다고 출력 결과에 표시됩니다.

다음은 확장 네트워크 계정을 사용으로 설정합니다.

```
# acctadm -e extended -f /var/log/net.log net
# acctadm net
             Net accounting: active
        Net accounting file: /var/log/net.log
      Tracked net resources: extended
    Untracked net resources: none
```

네트워크 계정을 사용으로 설정한 후 `dlstat` 및 `flowstat` 명령을 사용하여 로그 파일에서 정보를 추출할 수 있습니다. 다음 절차에서 단계에 대해 설명합니다.

▼ 네트워크 트래픽에 대한 기록 통계를 가져오는 방법

시작하기 전에 네트워크의 기록 데이터를 표시하려면 먼저 네트워크에 대해 확장 계정을 사용으로 설정해야 합니다. 또한 흐름의 트래픽에 대한 기록 데이터를 표시하려면 먼저 378 페이지 "흐름의 리소스 관리"에 설명된 대로 시스템에서 흐름을 구성해야 합니다.

1 네트워크 사용을 추적하려는 인터페이스가 있는 시스템에서 관리자가 됩니다.

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

2 데이터 링크의 리소스 사용에 대한 기록 정보를 추출하고 표시하려면 다음 명령을 사용합니다.

 # dlstat show-link -h [-a] -f *filename* **[-d** *date***] [-F** *format***] [-s** *start-time***] [-e** *end-time***] [**_link_**]**

`-h` 데이터 링크의 수신 및 송신 패킷별로 리소스 사용에 대한 기록 정보의 요약을 표시합니다.

`-a` 데이터 캡처 후에 이미 삭제된 데이터 링크를 포함하여 모든 데이터 링크의 리소스 사용을 표시합니다.

`-f` *filename* `acctadm` 명령을 사용하여 네트워크 계정을 사용으로 설정할 때 정의된 로그 파일을 지정합니다.

`-d` 정보를 사용할 수 있는 날짜에 기록된 정보를 표시합니다.

`-F` *format* 데이터를 특정 형식으로 표시합니다. 현재 지원되는 유일한 형식은 `gnuplot`입니다.

`-s` *start-time*, `-e` *end-time* 특정 날짜와 시간 범위에 대해 사용 가능한 기록된 정보를 표시합니다. `MM/DD/YYY,hh:mm:ss` 형식을 사용합니다. `hour(hh)`는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 현재 날짜의 시간 범위 데이터가 표시됩니다.

link 지정한 데이터 링크에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 데이터 링크에 대한 기록 네트워크 데이터가 표시됩니다.

3 구성된 흐름의 네트워크 트래픽에 대한 기록 정보를 추출하고 표시하려면 다음 명령을 사용합니다.

 # flowstat -h [-a] -f *filename* **[-d** *date***] [-F** *format***] [-s** *start-time***] [-e** *end-time***] [***flow***]**

`-h` 데이터 링크의 수신 및 송신 패킷별로 리소스 사용에 대한 기록 정보의 요약을 표시합니다.

`-a` 데이터 캡처 후에 이미 삭제된 데이터 링크를 포함하여 모든 데이터 링크의 리소스 사용을 표시합니다.

`-f` *filename* `acctadm` 명령을 사용하여 네트워크 계정을 사용으로 설정할 때 정의된 로그 파일을 지정합니다.

`-d` 정보를 사용할 수 있는 날짜에 기록된 정보를 표시합니다.

`-F` *format* 데이터를 특정 형식으로 표시합니다. 현재 지원되는 유일한 형식은 `gnuplot`입니다.

`-s` *start-time*, `-e` *end-time* 특정 날짜와 시간 범위에 대해 사용 가능한 기록된 정보를 표시합니다. `MM/DD/YYY,hh:mm:ss` 형식을 사용합니다. `hour(hh)`는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 현재 날짜의 시간 범위 데이터가 표시됩니다.

link 지정한 데이터 링크에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 데이터 링크에 대한 기록 네트워크 데이터가 표시됩니다.

flow 지정한 흐름에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 흐름에 대한 기록 네트워크 데이터가 표시됩니다.

예 22–12 데이터 링크의 리소스 사용에 대한 기록 정보 표시

다음 예에서는 지정한 데이터 링크의 네트워크 트래픽 및 해당 리소스 사용에 대한 기록 통계를 보여줍니다.

```
# dlstat show-link -h -f /var/log/net.log
LINK      DURATION  IPACKETS RBYTES      OPACKETS OBYTES      BANDWIDTH
e1000g0   80        1031     546908      0        0           2.44 Kbps
```

예 22–13 흐름의 리소스 사용에 대한 기록 정보 표시

다음 예에서는 흐름의 네트워크 트래픽 및 해당 리소스 사용에 대한 기록 통계를 표시하는 여러 가지 방법을 보여줍니다.

흐름의 트래픽별 리소스 사용에 대한 기록 통계를 표시합니다.

```
# flowstat -h -f /var/log/net.log
FLOW      DURATION  IPACKETS RBYTES      OPACKETS OBYTES      BANDWIDTH
flowtcp   100       1031     546908      0        0            43.76Kbps
flowudp   0         0        0           0        0            0.00Mbps
```

지정된 날짜 및 시간 범위에서 흐름의 트래픽별 리소스 사용에 대한 기록 통계를 표시합니다.

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-f /var/log/net.log flowtcp

FLOW      START       END         RBYTES   OBYTES     BANDWIDTH
flowtcp   10:39:06    10:39:26    1546     6539        3.23 Kbps
flowtcp   10:39:26    10:39:46    3586     9922        5.40 Kbps
flowtcp   10:39:46    10:40:06    240      216       182.40 bps
flowtcp   10:40:06    10:40:26    0        0           0.00 bps
```

지정된 날짜 및 시간 범위에서 흐름의 트래픽별 리소스 사용에 대한 기록 통계를 표시합니다. gnuplot 형식을 사용하여 정보를 표시합니다.

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
10:39:46 0.18
10:40:06 0.00
```

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 32비트에서 448비트까지 가변 길이 키를 사용하는 대칭 블록 암호 알고리즘입니다. 작성자인 Bruce Schneier에 따르면 Blowfish는 키가 자주 변경되지 않는 응용 프로그램에 최적화되었다고 합니다.

CA CA(인증 기관)를 참조하십시오.

CA (인증 기관) 디지털 서명과 공개-개인 키 쌍을 만드는 데 사용되는 디지털 인증서를 발급하는 인증된 타사 조직이나 회사입니다. CA는 고유 인증서가 부여된 개인의 ID를 보장합니다.

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.

CRL (인증서 해지 목록) CA에서 해지한 공개 키 인증서 목록입니다. CRL은 IKE를 통해 유지 관리되는 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. DES는 56비트 키를 사용합니다.

Diffie-Hellman 프로토콜 공개 키 암호화라고도 합니다. 1976년 Diffie와 Hellman이 개발한 비대칭 암호화 키 계약 프로토콜입니다. 이 프로토콜을 사용하면 두 사용자가 사전 보안 없이 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) 아키텍처 표준입니다. 주요 모듈은 분류자, 측정기, 표시자, 스케줄러 및 드롭퍼입니다. IPQoS는 분류자, 측정기 및 표시자 모듈을 구현합니다. diffserv 모델은 RFC 2475, *An Architecture for Differentiated Services*에서 설명합니다.

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 및 암호화 모드가 있습니다.

DR (동적 재구성) 시스템이 실행되는 동안 진행 중인 작업에 거의 또는 전혀 영향을 주지 않고 시스템을 재구성할 수 있게 하는 기능입니다. Oracle의 모든 Sun 플랫폼에서 DR을 지원하는 것은 아닙니다. 일부 플랫폼은 NIC와 같은 특정 하드웨어 유형의 DR만 지원합니다.

DSA Digital Signature Algorithm입니다. 512비트에서 4096비트 사이의 가변 키 크기를 사용하는 공개 키 알고리즘입니다. 미국 정부 표준인 DSS는 최대 1024비트까지 지정합니다. DSA는 입력으로 SHA-1을 사용합니다.

DSCP (DS 코드점) IP 헤더의 DS 필드에 포함될 경우 패킷 전달 방식을 나타내는 6비트 값입니다.

ESP (Encapsulating Security Payload) 데이터그램에 무결성과 기밀성을 제공하는 확장 헤더입니다. ESP는 IPsec(IP 보안 아키텍처)의 5가지 구성 요소 중 하나입니다.

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.

hop 두 호스트를 구분하는 라우터 수를 식별하는 데 사용되는 측정값입니다. 세 개의 라우터가 소스와 대상을 구분하는 경우 호스트는 서로 네 개 hop만큼 떨어져 있습니다.

ICMP Internet Control Message Protocol입니다. 오류를 처리하고 제어 메시지를 교환하는 데 사용됩니다.

ICMP 에코 요청 패킷 응답을 요청하기 위해 인터넷을 통해 컴퓨터로 전송되는 패킷입니다. 이러한 패킷을 일반적으로 "핑" 패킷이라고 합니다.

IKE Internet Key Exchange입니다. IKE는 IPsec SA(보안 연관)에 대한 인증된 키 관련 자료의 규정을 자동화합니다.

IP IP(인터넷 프로토콜), IPv4, IPv6을 참조하십시오.

IP-in-IP 캡슐화 IP 패킷 내의 IP 패킷을 터널링하기 위한 방식입니다.

IP 데이터그램 IP를 통해 전달되는 정보 패킷입니다. IP 데이터그램에는 헤더와 데이터가 포함됩니다. 헤더에는 데이터그램의 소스 및 대상 주소가 포함됩니다. 헤더의 다른 필드는 대상에서 데이터와 수반 데이터그램을 식별하고 재결합할 수 있도록 도와줍니다.

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 브릿지됨) 또는 ATM 네트워크가 있습니다. IP 링크 한 개에 IPv4 서브넷 번호 또는 접두어가 한 개 이상 할당됩니다. 서브넷 번호 또는 접두어 한 개를 여러 IP 링크에 할당할 수는 없습니다. ATM LANE에서 IP 링크는 에뮬레이트된 단일 LAN입니다. ARP를 사용하는 경우 ARP 프로토콜의 범위는 단일 IP 링크입니다.

IP 스택 대체로 TCP/IP를 "스택"이라고 합니다. 이것은 데이터 교환의 클라이언트 및 서버 끝에서 모든 데이터가 통과하는 계층(TCP, IP 등)을 나타냅니다.

IP (인터넷 프로토콜) 인터넷을 통해 한 컴퓨터에서 다른 컴퓨터로 데이터가 전송되는 방식 또는 프로토콜입니다.

IP 헤더 인터넷 패킷을 고유하게 식별하는 20바이트의 데이터입니다. 헤더에는 패킷의 소스 및 대상 주소가 포함됩니다. 헤더 내에 바이트를 더 추가할 수 있도록 하는 옵션이 있습니다.

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 흐름 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 보안입니다. IP 데이터그램에 보호 기능을 제공하는 보안 아키텍처입니다.

IPv4 Internet Protocol, version 4입니다. IPv4를 IP라고도 합니다. 이 버전은 32비트 주소 공간을 지원합니다.

IPv6 Internet Protocol, version 6입니다. IPv6은 128비트 주소 공간을 지원합니다.

MAC (메시지 인증 코드) MAC는 데이터 무결성을 보장하고 데이터 출처를 인증합니다. MAC는 도청에 대한 보호 기능을 제공하지 않습니다.

MD5 디지털 서명을 포함하여 메시지 인증에 사용되는 반복적인 암호화 해시 기능입니다. 이 기능은 1991년 Rivest가 개발했습니다.

MTU Maximum Transmission Unit입니다. 링크를 통해 전송할 수 있는 옥테트(크기)입니다. 예를 들어, 이더넷의 MTU는 1500옥테트입니다.

NAT Network Address Translation을 참조하십시오.

Network Address Translation NAT의 전체 이름입니다. 한 네트워크에서 사용되는 IP 주소를 다른 네트워크에서 알려진 다른 IP 주소로 변환합니다. 필요한 전역 IP 주소 수를 제한하는 데 사용됩니다.

NIC (네트워크 인터페이스 카드) 네트워크에 대한 인터페이스인 네트워크 어댑터 카드입니다. `igb` 카드와 같은 일부 NIC는 물리적 인터페이스를 여러 개 사용할 수 있습니다.

PFS (Perfect Forward Secrecy) PFS에서 데이터 전송을 보호하는 데 사용되는 키는 추가 키 파생에 사용되지 않습니다. 또한 데이터 전송을 보호하는 데 사용되는 키의 소스는 추가 키 파생에 사용되지 않습니다.

PFS는 인증된 키 교환에만 적용됩니다. Diffie-Hellman 프로토콜을 참조하십시오.

PHB (Per-Hop Behavior) 트래픽 클래스에 할당되는 우선 순위입니다. PHB는 다른 트래픽 클래스와 관련하여 해당 클래스의 흐름에 지정된 우선 순위를 나타냅니다.

PKI Public Key Infrastructure입니다. 디지털 인증서, 인증 기관 및 인터넷 트랜잭션에 관련된 각 당사자의 유효성을 확인 및 인증하는 기타 등록 기관으로 이루어진 시스템입니다.

RSA 디지털 서명 및 공개 키 암호 방식을 가져오는 방법입니다. 이 방법은 1978년 개발자 Rivest, Shamir 및 Adleman에 의해 처음 설명되었습니다.

SA SA(보안 연관)를 참조하십시오.

SA (보안 연관) 한 호스트의 보안 등록 정보를 두번째 호스트에 지정하는 연관입니다.

SADB Security Associations Database입니다. 암호화 키와 암호화 알고리즘을 지정하는 표입니다. 키와 알고리즘은 보안 데이터 전송에 사용됩니다.

SCTP Streams Control Transport Protocol을 참조하십시오.

SCTP (Stream Control Transmission Protocol) TCP와 유사한 방식으로 연결 지향 통신을 제공하는 전송 계층 프로토콜입니다. 또한 SCTP는 연결의 끝점 중 하나에서 IP 주소를 두 개 이상 사용할 수 있는 멀티홈을 지원합니다.

SHA-1 Secure Hashing Algorithm입니다. 이 알고리즘은 2^{64} 미만의 입력 길이에서 작동하여 메시지 다이제스트를 생성합니다. SHA-1 알고리즘은 DSA에 대한 입력입니다.

SPD SPD(보안 정책 데이터베이스)를 참조하십시오.

SPD (보안 정책 데이터베이스) 패킷에 적용할 보호 레벨을 지정하는 데이터베이스입니다. SPD는 IP 트래픽을 필터링하여 패킷을 무시할지, 일반 텍스트로 전달할지 또는 IPsec으로 보호할지를 결정합니다.

SPI SPI(Security Parameter Index)를 참조하십시오.

SPI (Security Parameter Index) 수신자가 수신된 패킷의 암호를 해독하는 데 사용해야 하는 SADB(Security Associations Database)의 행을 지정하는 정수입니다.

Stateful 패킷 필터 활성 연결의 상태를 모니터하고 얻은 정보를 사용하여 방화벽을 통해 허용할 네트워크 패킷을 결정할 수 있는 패킷 필터입니다. Stateful 패킷 필터는 요청과 응답을 추적하고 일치시켜 요청과 일치하지 않는 응답을 걸러낼 수 있습니다.

Stateless 자동 구성 호스트가 로컬 IPv6 라우터에서 알린 IPv6 접두어와 MAC 주소를 결합하여 고유한 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.

VLAN (가상 LAN) 장치 IP 프로토콜 스택의 이더넷(데이터 링크) 레벨에서 트래픽 전달을 제공하는 네트워크 인터페이스입니다.

VNIC (가상 네트워크 인터페이스) 물리적 네트워크 인터페이스에서 구성되었는지 여부에 관계없이 가상 네트워크 연결을 제공하는 의사 인터페이스입니다. 배타적 IP 영역이나 xVM 도메인과 같은 컨테이너가 VNIC 위에 구성되어 가상 네트워크를 형성합니다.

VPN (가상 사설망) 인터넷과 같은 공용 네트워크에서 터널을 사용하는 단일 보안 논리적 네트워크입니다.

가상 네트워크 단일 소프트웨어 엔티티로 함께 관리되는 소프트웨어 및 하드웨어 네트워크 리소스와 기능의 조합입니다. **내부** 가상 네트워크는 네트워크 리소스를 단일 시스템에 통합하며 "시스템 내 네트워크"라고도 합니다.

개인 주소 인터넷을 통해 경로를 지정할 수 없는 IP 주소입니다. 개인 주소는 인터넷 연결이 필요 없는 호스트의 내부 네트워크에서 사용할 수 있습니다. 이러한 주소는 Address Allocation for Private Internets (`http://www.ietf.org/rfc/rfc1918.txt?number=1918`)에서 정의되며 "1918" 주소라고도 합니다.

공개 키 암호화 서로 다른 두 개의 키를 사용하는 암호화 시스템입니다. 공개 키는 모든 사용자에게 알려집니다. 개인 키는 메시지 수신자에게만 알려집니다. IKE는 IPsec에 대해 공개 키를 제공합니다.

노드 IPv6에서 호스트 또는 라우터에 관계없이 IPv6을 사용할 수 있는 모든 시스템입니다.

대기 다른 물리적 인터페이스에서 오류가 발생하지 않은 경우 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 패킷 필터 Stateful 패킷 필터를 참조하십시오.

디지털 서명 전자 방식으로 전송된 메시지에 첨부되고 보낸 사람을 고유하게 식별하는 디지털 코드입니다.

라우터 대체로 인터페이스가 두 개 이상 있고 경로 지정 프로토콜을 실행하며 패킷을 전달하는 시스템입니다. 시스템이 PPP 링크의 끝점인 경우 인터페이스 한 개만 라우터로 사용되는 시스템을 구성할 수 있습니다.

라우터 검색 호스트가 연결된 링크에 있는 라우터를 찾는 프로세스입니다.

라우터 알림 라우터가 주기적으로 또는 라우터 요청 메시지에 대한 응답으로 다양한 링크 및 인터넷 매개변수와 함께 현재 상태를 알리는 프로세스입니다.

라우터 요청 호스트가 라우터에서 예약된 다음 시간이 아니라 즉시 라우터 알림을 생성하도록 요청하는 프로세스입니다.

로컬-사용 주소 서브넷 또는 가입자 네트워크 내에서 로컬 경로 지정 가능 범위만 있는 유니캐스트 주소입니다. 이 주소에 로컬 또는 전역 고유성 범위가 있을 수도 있습니다.

링크 계층 IPv4/IPv6 바로 아래에 있는 계층입니다.

링크-로컬 주소 IPv6에서 자동 주소 구성과 같은 용도로 단일 링크의 주소 지정에 사용되는 지정입니다. 기본적으로 링크-로컬 주소는 시스템의 MAC 주소에서 생성됩니다.

멀티캐스트 주소 특정 방식으로 인터페이스 그룹을 식별하는 IPv6 주소입니다. 멀티캐스트 주소로 전송된 패킷은 그룹의 모든 인터페이스에 전달됩니다. IPv6 멀티캐스트 주소는 IPv4 브로드캐스트 주소와 기능이 비슷합니다.

멀티홈 호스트 물리적 인터페이스가 두 개 이상 있고 패킷 전달을 수행하지 않는 시스템입니다. 멀티홈 호스트는 경로 지정 프로토콜을 실행할 수 있습니다.

물리적 인터페이스 링크에 대한 시스템 연결입니다. 이 연결은 대체로 장치 드라이버와 NIC(네트워크 인터페이스 카드)로 구현됩니다. `igb`와 같은 일부 NIC는 여러 연결점을 사용할 수 있습니다.

방화벽 조직의 개인 네트워크 또는 인트라넷을 인터넷에서 격리시켜 외부 침입으로부터 보호하는 장치 또는 소프트웨어입니다. 방화벽에는 패킷 필터링, 프록시 서버 및 NAT(Network Address Translation)가 포함될 수 있습니다.

복구 감지 NIC 또는 NIC에서 일부 계층 3 장치로의 경로가 오류 발생 후에 올바르게 작동하기 시작할 때 이를 감지하는 프로세스입니다.

부하 분산 인터페이스 세트에 인바운드 또는 아웃바운드 트래픽을 분산시키는 프로세스입니다. 부하 분산을 사용하면 처리량이 증가합니다. 부하 분산은 네트워크 트래픽이 여러 연결을 사용하는 여러 대상으로 흐르는 경우에만 발생합니다. 두 가지 유형의 부하 분산이 있습니다. 인바운드 부하 분산은 인바운드 트래픽에 사용되고 아웃바운드 부하 분산은 아웃바운드 트래픽에 사용됩니다.

브로드캐스트 주소 주소의 호스트 부분이 모두 0(10.50.0.0)이거나 모두 1비트(10.50.255.255)인 IPv4 네트워크 주소입니다. 로컬 네트워크의 컴퓨터에서 브로드캐스트 주소로 전송된 패킷은 해당 네트워크의 모든 컴퓨터에 전달됩니다.

비대칭 키 암호화 메시지를 보낸 사람과 받는 사람이 서로 다른 키를 사용하여 메시지를 암호화하고 암호를 해독하는 암호화 시스템입니다. 비대칭 키는 대칭 키 암호화에 대해 보안 채널을 설정하는 데 사용됩니다. Diffie-Hellman 프로토콜은 비대칭 키 프로토콜의 예입니다. 대칭 키 암호화와 대조됩니다.

사용자-우선 순위 VLAN 장치 네트워크에서 이더넷 데이터그램 전달 방식을 정의하는 즉, 서비스 클래스 표시를 구현하는 3비트 값입니다.

사이트-로컬-사용 주소 단일 사이트의 주소 지정에 사용되는 지정입니다.

선택기 네트워크 스트림에서 트래픽을 선택하기 위해 특정 클래스의 패킷에 적용할 기준을 구체적으로 정의하는 요소입니다. IPQoS 구성 파일의 필터 절에서 선택기를 정의합니다.

속임수 인증된 호스트에서 메시지를 보냈음을 나타내는 IP 주소를 사용하여 메시지를 보내 컴퓨터에 허용되지 않은 액세스를 시도합니다. IP 스푸핑을 실행하는 해커는 먼저 여러 가지 기술을 시도하여 인증된 호스트의 IP 주소를 찾은 다음 해당 호스트에서 패킷을 보낸 것처럼 표시되도록 패킷 헤더를 수정합니다.

스니프 컴퓨터 네트워크를 도청하며 일반 텍스트 암호와 같은 정보를 무선으로 조사하는 자동화된 프로그램의 일부로 자주 사용됩니다.

스머프 공격 원격 위치에서 IP 브로드캐스트 주소 또는 여러 브로드캐스트 주소로 보내는 ICMP 에코 요청 패킷을 사용하여 심각한 네트워크 혼잡 또는 정전을 발생시킵니다.

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 검사 기반(옵션)의 두 가지 실패 감지 유형이 포함됩니다.

애니캐스트 그룹 동일한 애니캐스트 IPv6 주소를 가진 인터페이스 그룹입니다. IPv6의 Oracle Solaris 구현에서는 애니캐스트 주소와 그룹 생성을 지원하지 않습니다. 하지만 Oracle Solaris IPv6 노드는 애니캐스트 그룹에 트래픽을 보낼 수 있습니다.

애니캐스트 주소 일반적으로 여러 노드에 속하는 인터페이스 그룹에 할당되는 IPv6 주소입니다. 애니캐스트 주소로 전송된 패킷은 해당 주소를 가진 **가장 가까운** 인터페이스로 경로 지정됩니다. 패킷의 경로는 경로 지정 프로토콜의 거리 측정을 준수합니다.

양방향 터널 데이터그램을 양방향으로 모두 전송할 수 있는 터널입니다.

역방향 터널 모바일 노드의 주소 관리에서 시작되고 홈 에이전트에서 종료되는 터널입니다.

유니캐스트 주소 IPv6 사용 노드의 단일 인터페이스를 식별하는 IPv6 주소입니다. 유니캐스트 주소는 사이트 접두어, 서브넷 ID 및 인터페이스 ID로 구성됩니다.

이중 스택 네트워크 계층에 IPv4와 IPv6이 모두 있고 스택의 나머지 부분은 동일한 TCP/IP 프로토콜 스택입니다. Oracle Solaris 설치 도중 IPv6을 사용으로 설정하면 호스트에 이중 스택 버전의 TCP/IP가 수신됩니다.

인접 라우터 검색 호스트가 연결된 링크에 있는 다른 호스트를 찾을 수 있도록 하는 IP 방식입니다.

인접 라우터 알림 인접 라우터 요청 메시지에 대한 응답 또는 링크-계층 주소 변경을 알리기 위해 노드에서 요청하지 않은 인접 라우터 알림을 보내는 프로세스입니다.

인접 라우터 요청 노드에서 인접 라우터의 링크-계층 주소를 확인하기 위해 보내는 요청입니다. 인접 라우터 요청은 캐시된 링크-계층 주소를 통해 인접 라우터에 여전히 연결할 수 있는지도 확인합니다.

인증 헤더 기밀성 없이 IP 데이터그램에 인증과 무결성을 제공하는 확장 헤더입니다.

자동 구성 호스트가 사이트 접두어와 로컬 MAC 주소에서 IPv6 주소를 자동으로 구성하는 프로세스입니다.

재생 공격 IPsec에서 침입자가 패킷을 캡처하는 공격입니다. 나중에 저장된 패킷으로 원본을 바꾸거나 반복합니다. 이러한 공격으로부터 보호하기 위해 패킷을 보호하는 보안 키의 수명 동안 증분하는 필드를 패킷에 포함할 수 있습니다.

재지정 라우터에서 특정 대상에 도달하는 데 보다 효율적인 첫번째 hop 노드를 호스트에 알립니다.

최소 캡슐화 홈 에이전트, 외부 에이전트 및 모바일 노드에서 지원할 수 있는 IPv4-in-IPv4 터널링의 선택적 형태입니다. 최소 캡슐화는 IP-in-IP 캡슐화보다 오버헤드가 8 또는 12바이트 더 적습니다.

출력 트래픽 측정 결과로 수행할 작업입니다. IPQoS 측정기에는 IPQoS 구성 파일에서 정의하는 세 가지 출력 결과(빨간색, 노란색 및 녹색)가 있습니다.

측정기 특정 클래스에 대한 트래픽 흐름 속도를 측정하는 diffserv 아키텍처의 한 모듈입니다. IPQoS 구현에는 `tokenmt`와 `tswtclmt`라는 두 개의 측정기가 있습니다.

캡슐화 헤더와 페이로드가 첫번째 패킷에 배치된 후 이 패킷이 두번째 패킷의 페이로드에 배치되는 프로세스입니다.

클래스 IPQoS에서 유사한 특성을 공유하는 네트워크 흐름 그룹입니다. IPQoS 구성 파일에서 클래스를 정의합니다.

키 관리 SA(보안 연관)를 관리하는 방식입니다.

키 저장소 이름 관리자가 NIC(네트워크 인터페이스 카드)의 저장소 영역, 즉 키 저장소에 지정하는 이름입니다. 키 저장소 이름을 토큰 또는 토큰 ID라고도 합니다.

터널 캡슐화되는 동안 뒤에 데이터그램이 오는 경로입니다. 캡슐화를 참조하십시오.

테스트 주소 검사의 소스 또는 대상 주소로 사용해야 하고 데이터 트래픽의 소스 또는 대상 주소로 사용하면 안되는 IPMP 그룹의 IP 주소입니다.

패킷 통신 회선을 통해 하나의 단위로 전송되는 정보 그룹입니다. IP 헤더 및 페이로드를 포함합니다.

패킷 필터 방화벽을 통해 지정한 패킷을 허용하거나 허용하지 않도록 구성할 수 있는 방화벽 기능입니다.

패킷 헤더 IP 헤더를 참조하십시오.

페이로드 패킷에서 전달되는 데이터입니다. 패킷을 대상에 전달하는 데 필요한 헤더 정보는 페이로드에 포함되지 않습니다.

표시자 1. 패킷 전달 방식을 나타내는 값으로 IP 패킷의 DS 필드를 표시하는 diffserv 아키텍처 및 IPQoS의 한 모듈입니다. IPQoS 구현에서 표시자 모듈은 `dscpmk`입니다.

2. 사용자 우선 순위 값으로 이더넷 데이터그램의 가상 LAN 태그를 표시하는 IPQoS 구현의 한 모듈입니다. 사용자 우선 순위 값은 VLAN 장치가 있는 네트워크에서 데이터그램을 전달하는 방식을 나타냅니다. 이 모듈을 `dlcosmk`라고 합니다.

프로토콜 스택 IP 스택을 참조하십시오.

프록시 서버 웹 브라우저와 같은 클라이언트 응용 프로그램과 다른 서버 사이에 있는 서버입니다. 특정 웹 사이트에 대한 액세스 방지 등의 용도로 요청을 필터링하는 데 사용됩니다.

필터 IPQoS 구성 파일에서 클래스의 특성을 정의하는 규칙 세트입니다. IPQoS 시스템은 IPQoS 구성 파일의 필터를 준수하는 트래픽 흐름을 모두 처리하도록 선택합니다. 패킷 필터를 참조하십시오.

해시 값 텍스트 문자열에서 생성된 숫자입니다. 해시 기능은 전송된 메시지가 조작되지 않았음을 보장하는 데 사용됩니다. 단방향 해시 기능의 예로 MD5와 SHA-1이 있습니다.

헤더 IP 헤더를 참조하십시오.

호스트 패킷 전달을 수행하지 않는 시스템입니다. Oracle Solaris를 설치하면 한 시스템이 기본적으로 호스트가 됩니다. 즉, 해당 시스템은 패킷을 전달할 수 없습니다. 일반적으로 호스트는 여러 인터페이스를 사용할 수 있지만 물리적 인터페이스는 하나뿐입니다.

흐름 계정 IPQoS에서 트래픽 흐름에 대한 정보를 누적 및 기록하는 프로세스입니다. IPQoS 구성 파일에서 `flowacct` 모듈의 매개변수를 정의하여 흐름 계정을 설정합니다.

색인

L

M