

Oracle® Solaris 11.1에서 가상 네트워크 사용

Copyright © 2011, 2012, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록 상표입니다.

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

목차

머리말	5
1 Oracle Solaris에서 네트워크 가상화 및 리소스 관리	7
네트워크 가상화 개요	7
네트워크 가상화의 구성 요소	8
가상 네트워크 구현 대상	10
가상화 구성 요소를 구성하기 위한 명령	10
네트워크 리소스 관리의 개요	11
리소스 제어를 위한 데이터 링크 등록 정보	11
플로우를 사용한 네트워크 리소스 관리	12
네트워크 리소스 관리 명령	13
2 Oracle Solaris에서 가상 네트워크 만들기 및 관리	15
가상 네트워크의 구성 요소 구성	15
▼ VNIC 및 Etherstub 구성 방법	15
▼ VLAN ID로 VNIC 구성 방법	17
가상 네트워크 구축	18
▼ 가상 네트워크용으로 영역을 구성하는 방법	19
▼ VNIC를 사용하도록 영역을 재구성하는 방법	21
▼ 개인 가상 네트워크를 만드는 방법	23
VNIC에 대한 기타 관리 작업	25
VNIC의 VLAN ID 수정	26
VNIC MAC 주소 수정	26
VNIC 마이그레이션	27
VNIC 정보 표시	29
▼ VNIC 삭제 방법	29

3 Oracle Solaris의 네트워크 리소스 관리	31
클라이언트, 전송 링 및 수신 링 작업	31
MAC 클라이언트 및 링 할당	31
링 할당에 대한 데이터 링크 등록 정보	33
수신 및 전송 링 작업을 위한 명령	33
링 정보 가져오기 및 해석	34
▼ 클라이언트 구성 및 링 할당 방법	38
풀 및 CPU 작업	40
▼ 데이터 링크에 대한 CPU 풀을 구성하는 방법	42
▼ 링크에 CPU를 할당하는 방법	43
플로우의 리소스 관리	44
▼ 플로우 구성 방법	45
4 Oracle Solaris에서 네트워크 트래픽 및 리소스 사용 모니터링	51
네트워크 트래픽 플로우 개요	51
트래픽 통계 모니터링 명령	54
링크의 네트워크 트래픽에 대한 통계 수집	54
네트워크 장치에서 네트워크 트래픽 통계 가져오기	55
라인에서 네트워크 트래픽 통계 가져오기	56
링크 집계에서 네트워크 트래픽 통계 가져오기	58
플로우의 네트워크 트래픽에 대한 통계 수집	58
네트워크 트래픽에 대한 네트워크 정산 구성	60
▼ 네트워크 정산 설정 방법	60
▼ 네트워크 트래픽에 대한 기록 통계를 가져오는 방법	62
색인	65

머리말

Oracle Solaris 11.1에서 가상 네트워크 사용을 시작합니다. 이 설명서는 Oracle Solaris 네트워크 구성을 위한 기본 항목 및 절차를 다루는 **Establishing an Oracle Solaris 11.1 Network** 시리즈의 일부입니다. 이 설명서에서는 Oracle Solaris를 이미 설치했다고 가정합니다.

이 설명서의 대상

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

Oracle Support에 액세스

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

활자체 규약

다음 표는 이 설명서에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체 또는 기호	설명	예제
AaBbCc123	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오. <code>machine_name% you have mail.</code>
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	<code>machine_name% su</code> <code>Password:</code>

표 P-1 활자체 규약 (계속)

활자체 또는 기호	설명	예제
<i>AaBbCc123</i>	위치 표시자: 실제 이름이나 값으로 바꿉니다.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.
<i>AaBbCc123</i>	설명서 제목, 새 용어, 강조 표시할 용어입니다.	<p>사용자 설명서의 6장을 읽으십시오.</p> <p>캐시는 로컬로 저장된 복사본입니다.</p> <p>파일을 저장하면 안 됩니다.</p> <p>주: 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.</p>

명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 슈퍼유저 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	\$
슈퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	#
C 셸	machine_name%
슈퍼유저용 C 셸	machine_name#

Oracle Solaris에서 네트워크 가상화 및 리소스 관리

이 장에서는 Oracle Solaris에서 네트워크 가상화 및 리소스 제어에 관련된 기본 개념에 대해 설명합니다. 다음 항목을 다룹니다.

- 7 페이지 “네트워크 가상화 개요”
- 11 페이지 “네트워크 리소스 관리의 개요”

이러한 기능은 플로우 제어를 관리하고 시스템 성능을 향상시키며 OS 가상화, 유틸리티 컴퓨팅, 서버 통합 실현 등에 필요한 네트워크 사용률을 구성하는 데 유용합니다.

네트워크 가상화 개요

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(가상 근거리 통신망) 소프트웨어 기술입니다. 외부 가상 네트워크의 예로 대규모 회사 네트워크와 데이터 센터가 있습니다.

이 설명서에서는 내부 가상 네트워크에 대해 자세히 설명합니다. **내부 가상 네트워크**는 하나 이상의 물리적 NIC에 네트워크 인터페이스가 구성된 가상 시스템 또는 영역을 사용하는 시스템 한 개로 구성됩니다. 이러한 네트워크 인터페이스를 **가상 네트워크 인터페이스 카드 또는 vNIC(가상 NIC)**라고 부릅니다. 이러한 컨테이너는 동일한 로컬 네트워크에 있는 것처럼 서로 통신하여 단일 호스트에서 효율적으로 가상 네트워크로 작동할 수 있습니다.

특별한 유형의 내부 가상 네트워크는 **개인 가상 네트워크**입니다. 개인 가상 네트워크는 VPN(가상 사설망)과는 다릅니다. VPN 소프트웨어는 두 끝점 시스템 간에 보안 P2P

연결을 만듭니다. 개인 가상 네트워크는 외부 시스템에서 액세스할 수 없는 시스템의 가상 네트워크입니다. 다른 외부 시스템에 대한 이러한 내부 네트워크의 격리는 etherstub을 통해 VNIC를 구성함으로써 얻을 수 있습니다. Etherstub은 다음 단원에서 설명합니다.

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.

네트워크 가상화의 구성 요소

Oracle Solaris에서 네트워크 가상화의 기본 구성 요소는 다음과 같습니다.

- VNIC(가상 네트워크 인터페이스 카드)
- 가상 스위치
- Etherstub

VNIC는 물리적 NIC와 데이터 링크 인터페이스가 동일한 가상 네트워크 장치입니다. VNIC는 기본 데이터 링크로 구성합니다. VNIC를 구성하면 물리적 NIC처럼 동작합니다. 또한 시스템 리소스가 VNIC를 물리적 NIC처럼 처리합니다. VNIC에는 자동으로 생성된 MAC 주소가 있습니다. 사용 중인 네트워크 인터페이스에 따라 [dladm\(1M\)](#) 매뉴얼 페이지에 설명된 대로 기본 주소가 아닌 MAC 주소를 VNIC에 명시적으로 지정할 수 있습니다.

VNIC를 지원하는 물리적 인터페이스의 현재 목록은 [Network Virtualization and Resource Control FAQ](#) (<http://hub.opensolaris.org/bin/view/Project+crossbow/faq>)를 참조하십시오.

VNIC를 만들면 **가상 스위치**가 자동으로 만들어집니다. 이더넷 설계에 따라 스위치 포트가 해당 포트에 연결된 호스트로부터 송신 패킷을 받는 경우 해당 패킷이 동일한 포트의 대상으로 이동할 수 없습니다. 가상 네트워크는 동일한 NIC를 공유하므로 이 설계는 가상 네트워크를 통해 구성된 시스템의 결점입니다. 송신 패킷은 스위치 포트에서 외부 네트워크로 이동합니다. 패킷은 전송된 포트와 동일한 포트를 통해 반환될 수 없기 때문에 수신 패킷이 대상 영역에 도달할 수 없습니다. 가상 스위치는 이러한 영역에 패킷 전달 방식을 제공합니다. 가상 스위치는 가상 네트워크가 서로 통신할 수 있는 데이터 경로를 열어 줍니다.

*Etherstub*은 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 개발자라면 액세스가 제한되는 네트워크 환경을 만들려는 경우, etherstub을 사용하여 그러한 환경을 만들 수 있습니다.

Etherstub 및 VNIC는 Oracle Solaris의 가상화 기능 중 일부에 지나지 않습니다. Oracle Solaris 영역에서는 일반적으로 이러한 구성 요소가 사용됩니다. 영역에서 사용할 VNIC 또는 etherstub을 지정하여 단일 시스템에 네트워크를 만들 수 있습니다. 영역에 대한 자세한 내용은 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리**를 참조하십시오.

이러한 구성 요소를 조합하고 이를 영역에 배치하여 다음 그림과 비슷한 시스템 네트워크를 구성할 수 있습니다.

그림 1-1 단일 인터페이스에 대한 VNIC 구성

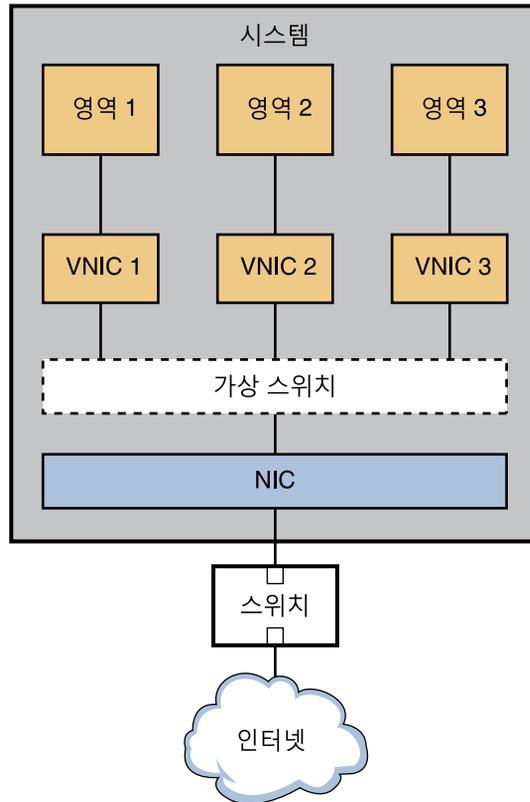


그림 1-1에서는 NIC 한 개가 있는 단일 시스템을 보여줍니다. NIC는 VNIC 세 개로 구성됩니다. 각 VNIC는 단일 영역을 지원합니다. 영역 1, 영역 2 및 영역 3은 단일 시스템 내에서 가상 네트워크를 구성합니다. 영역은 각각의 VNIC를 사용하여 각 영역 간에 그리고 외부 네트워크와 통신합니다. 그리고 세 개의 VNIC가 가상 스위치를 통해 기본 물리적 NIC에 연결됩니다. 가상 스위치의 기능은 외부 스위치가 스위치의 포트에 연결된 시스템에 제공하는 연결 기능과 동일합니다.

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.

영역은 또한 시스템 내에서 각 영역 간에 트래픽을 교환할 수도 있습니다. 예를 들어, 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.

또는 etherstub을 기반으로 가상 네트워크를 만들 수 있습니다. etherstub은 완전히 소프트웨어를 기반으로 하며 가상 네트워크의 기반으로 네트워크 인터페이스가 필요하지 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 상당한 양의 대역폭

분리, 보안 및 유연성의 큰 손실 없이 여러 영역이나 가상 시스템이 포함된 단일 시스템으로 많은 시스템을 교체할 수 있습니다.

네트워크 가상화의 이점에 대한 데모를 보려면 [Consolidating the Data Center With Network Virtualization \(http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html\)](http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html)을 참조하십시오.

가상화 구성 요소를 구성하기 위한 명령

VNIC를 만들려면 `dladm create-vnic` 명령을 사용합니다.

```
# dladm create-vnic -l link [-v vid] vnic
```

`-l link` VNIC가 구성된 데이터 링크의 이름입니다.

`-v vid` VNIC를 VLAN으로 만들려는 경우 VNIC에 대한 VLAN ID입니다. 이 옵션은 필수가 아닙니다. VLAN ID로 VNIC를 구성하려면 17 페이지 “VLAN ID로 VNIC 구성 방법”을 참조하십시오. VLAN에 대한 자세한 내용은 **Oracle Solaris 11.1 네트워크 성능 관리**의 3 장, “VLAN 작업”을 참조하십시오.

`vnic` VNIC의 이름을 나타냅니다.

주-MAC 주소, VNIC와 연결할 CPU 등과 같이 VNIC에 대한 다른 등록 정보를 구성할 수 있습니다. 이러한 등록 정보 목록을 보려면 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 특정 등록 정보 수정은 VNIC에서만 작동합니다. 예를 들어, `dladm create-vnic` 명령의 경우 MAC 주소를 구성할 뿐만 아니라 VLAN ID를 지정하여 VNIC를 VLAN으로 만들 수 있습니다. 하지만 `dladm create-vlan` 명령을 사용하여 VLAN에 대해 MAC 주소를 직접 구성할 수는 없습니다.

VNIC는 데이터 링크를 통해 한 번에 하나씩만 만들 수 있습니다. 데이터 링크로서 VNIC에는 필요에 따라 추가로 구성할 수 있는 링크 등록 정보가 포함됩니다. [11 페이지](#) “[리소스 제어를 위한 데이터 링크 등록 정보](#)”에는 시스템에서 네트워크 리소스 사용을 관리하기 위한 일부 등록 정보가 나열되어 있습니다.

etherstub을 만들려면 `dladm create-ether` 명령을 사용합니다.

```
# dladm create-ether etherstub
```

VNIC 또는 etherstub 만들기는 가상 네트워크 구성의 초기 단계입니다. 이러한 구성 요소를 사용하여 시스템에서 가상 네트워크를 만들려면 [2 장](#), “[Oracle Solaris에서 가상 네트워크 만들기 및 관리](#)”를 참조하십시오.

네트워크 리소스 관리의 개요

이 단원에서는 시스템에서 네트워크 리소스 사용을 관리하기 위해 사용할 수 있는 여러 가지 방법들에 대해 설명합니다.

리소스 제어를 위한 데이터 링크 등록 정보

Oracle Solaris 11에서는 네트워크 리소스를 관리하여 QoS(서비스 품질)를 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 네트워크킹 프로세스에 사용할 수 있는 양을 결정합니다. 예를 들어, 네트워크킹 프로세스 전용으로 예약된 특정 개수의 CPU와 링크를 연결할 수 있습니다. 또는 특정 유형의 네트워크 트래픽을 처리하도록 지정된 대역폭을 링크에 지정할 수 있습니다.

리소스 등록 정보를 정의한 후에는 새로운 값이 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. 예를 들어,

`dladm set-linkprop` 명령을 사용하여 네트워크 리소스와 관련된 등록 정보를 설정할 수 있습니다. 물리적 및 가상 데이터 링크 모두에 동일한 구문이 사용됩니다.

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 지정할 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 [51 페이지](#) “네트워크 트래픽 플로우 개요”를 참조하십시오.

네트워크 리소스 관리는 다음 작업에 유용합니다.

- 네트워크 프로비저닝
- 서비스 단계 계약 설정
- 클라이언트 청구
- 보안 문제 진단

복잡한 QoS 규칙 정의 없이 개별 시스템에서 데이터 트래픽을 격리시키고 우선 순위를 지정하고 추적 및 제어할 수 있습니다.

플로우를 사용한 네트워크 리소스 관리

플로우는 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 플로우를 구성하며 특정 플로우 이름으로 레이블이 지정됩니다. 그런 다음 플로우에 특정 리소스를 지정할 수 있습니다.

플로우를 만들기 위한 기본으로 사용되는 속성은 패킷 헤더의 정보에서 파생됩니다. 다음 속성 중 하나에 따라 패킷 트래픽을 플로우로 구성할 수 있습니다.

- IP 주소
- 전송 프로토콜 이름(UDP, TCP 또는 SCTP)
- 응용 프로그램 포트 번호(예: FTP의 경우 포트 21)
- IPv6 패킷의 QoS에만 사용되는 DS 필드 속성. DS 필드에 대한 자세한 내용은 [Oracle Solaris 11.1에서 IP 서비스 품질 관리](#)를 참조하십시오.

플로우는 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 플로우를 만들 수 있습니다. 하지만 포트 번호 21에서 수신된 지정된 IP 주소의 패킷에 대해서는 플로우를 만들 수 없습니다. 마찬가지로, IP 주소 192.168.1.10의 모든 트래픽에 대해 플로우를 만든 다음 192.168.1.10의 전송 계층 트래픽에 대해 플로우를 만들 수는 없습니다. 따라서 각 플로우가 다른 속성을 기반으로 하는 여러 플로우를 시스템에 구성할 수 있습니다.

네트워크 리소스 관리 명령

네트워크 리소스를 할당하는 데 사용되는 명령은 데이터 링크 또는 플로우에서 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
```

다음은 리소스 할당에 설정할 수 있는 링크 등록 정보입니다.

- **대역폭** - 특정 링크 사용에 대해 하드웨어 대역폭을 제한할 수 있습니다.
- **NIC 링** - NIC가 링 할당을 지원할 경우 전송 및 수신 링을 데이터 링크에서 전용으로 사용할 수 있습니다. NIC 링은 [31 페이지 “클라이언트, 전송 링 및 수신 링 작업”](#)에서 설명합니다.
- **CPU 풀** - CPU 풀은 일반적으로 특정 영역으로 만들어지고 연결됩니다. 이러한 풀을 데이터 링크에 지정하여 연결된 영역의 네트워크 프로세스 관리에 CPU 세트를 예약할 수 있습니다. CPU 및 풀은 [40 페이지 “풀 및 CPU 작업”](#)에서 설명합니다.
- **CPU** - 여러 CPU가 있는 시스템에서는 특정 네트워크 처리를 위해 지정된 개수의 CPU를 전용으로 사용할 수 있습니다.
- 플로우에 대해 `flowadm` 하위 명령을 사용할 수 있습니다. 플로우에서 리소스를 관리하는 것은 데이터 링크에서 리소스를 관리하는 방법과 비슷합니다. 동시에 플로우를 만들고 리소스를 추가하려면 다음 구문을 사용합니다.

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] \
-p property=value[,property=value] flow
```

플로우의 특성을 결정하는 정의된 속성 세트가 시스템의 **플로우 제어 정책**을 구성합니다.

기존 플로우의 등록 정보를 설정하려면 다음 구문을 사용합니다.

```
# flowadm set-flowprop -p property=value[,property=value] flow
```

플로우에 지정할 수 있는 리소스 지정에 대한 등록 정보는 링크에 직접 지정된 등록 정보와 동일합니다. 하지만 현재 대역폭 등록 정보만 플로우와 연결될 수 있습니다. 데이터 링크와 플로우에서 등록 정보를 설정하는 명령은 서로 다르지만 구문은 비슷합니다. 대역폭 등록 정보를 구성하려면 [45 페이지 “플로우 구성 방법”](#)의 예를 참조하십시오.

flowadm 명령에 대한 자세한 내용은 [flowadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
flowadm 명령에 사용할 하위 명령 목록을 보려면 다음을 입력합니다.

```
# flowadm help
The following subcommands are supported:
Flow      : add-flow      remove-flow  reset-flowprop
            set-flowprop  show-flow    show-flowprop
For more info, run: flowadm help <subcommand>.
```

Oracle Solaris에서 가상 네트워크 만들기 및 관리

이 장에는 단일 시스템에서 가상 네트워크를 구성하는 작업이 포함되어 있습니다.

다음 항목을 다룹니다.

- 15 페이지 “가상 네트워크의 구성 요소 구성”
- 18 페이지 “가상 네트워크 구축”
- 25 페이지 “VNIC에 대한 기타 관리 작업”

가상 네트워크에 대한 소개는 1 장, “Oracle Solaris에서 네트워크 가상화 및 리소스 관리”를 참조하십시오.

가상 네트워크의 구성 요소 구성

Oracle Solaris 11에서 etherstub 및 VNIC는 네트워크 가상화의 기본 구성 요소입니다. 이 단원에서는 가상 네트워크를 구축하기 위한 준비로 이러한 구성 요소를 구성하는 단계를 설명합니다. 이러한 구성 요소에 대한 자세한 내용은 8 페이지 “네트워크 가상화의 구성 요소”를 참조하십시오.

다음 절차에 대해 설명합니다.

- 15 페이지 “VNIC 및 Etherstub 구성 방법”
- 17 페이지 “VLAN ID로 VNIC 구성 방법”

▼ VNIC 및 Etherstub 구성 방법

VNIC는 가상 네트워크를 외부 네트워크에 연결합니다. 또한 VNIC를 사용하면 VNIC로 자동으로 만들어지는 가상 스위치를 통해 영역이 서로 통신할 수 있습니다. 가상 네트워크가 영역 간에 내부적으로 그리고 외부 LAN 및 인터넷에서 트래픽을 호스트하기 위해서는 각 영역에 고유한 인터페이스가 포함되어야 합니다. 따라서 가상 네트워크에 속하는 영역 수만큼 이 절차를 여러 번 반복해야 합니다.

1 관리자로 전환합니다.

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

2 (선택 사항) etherstub을 만듭니다.

```
# dladm create-etherstub etherstub
```

외부 시스템에서 액세스하지 못하도록 제한하려는 개인 가상 네트워크를 만드는 경우에만 이 단계를 수행하십시오. 개인 가상 네트워크에 대한 자세한 내용은 7 페이지 “네트워크 가상화 개요”를 참조하십시오.

데이터 링크와 마찬가지로 네트워크 설정에 유용한 방식으로 etherstub 이름을 지정할 수 있습니다. 사용자 정의된 이름을 만드는 방법에 대한 지침은 **Oracle Solaris 11 네트워킹 소개**의 “유효한 링크 이름 규칙”을 참조하십시오.

3 VNIC를 만듭니다.

```
# dladm create-vnic -l datalink [-v vid] vnic
```

개인 가상 네트워크에 대해 VNIC를 만드는 경우 datalink에 대한 etherstub을 지정합니다. VNIC를 VLAN으로 만드는 경우에만 명령 구문에 -v vid를 포함합니다. 여기서 vid는 VNIC의 VLAN ID를 나타냅니다. 그렇지 않으면 이 옵션을 생략합니다.

VNIC를 VLAN으로 만드는 경우 17 페이지 “VLAN ID로 VNIC 구성 방법”에서 VNIC를 VLAN으로 만드는 것과 관련된 추가 단계를 참조하십시오.

VNIC 이름은 임의로 지정할 수 있습니다. VNIC에 사용자 정의된 이름을 지정하려면 **Oracle Solaris 11 네트워킹 소개**의 “유효한 링크 이름 규칙”을 참조하십시오.

4 VNIC를 통해 IP 인터페이스를 만듭니다.

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

5 VNIC 인터페이스에 정적 IP 주소를 지정합니다.

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

-a address IP 주소를 지정합니다. CIDR 표기법을 사용할 수 있습니다.

interface 이전 단계에서 만든 VNIC를 지정합니다.

정적 IP 주소는 IPv4 및 IPv6 주소일 수 있습니다. IP 주소 구성에 대한 자세한 내용은 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 “IP 인터페이스를 구성하는 방법”을 참조하십시오.

IP 주소 구성에 대한 자세한 내용은 **Oracle Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결**의 “IP 인터페이스를 구성하는 방법”을 참조하십시오.

6 /etc/hosts 파일에 주소 정보를 추가합니다.

▼ VLAN ID로 VNIC 구성 방법

가상 네트워크에서 VLAN 트래픽을 호스트하도록 VLAN ID를 사용하여 VNIC를 구성할 수 있습니다. 또한 개별 VNIC의 VLAN 구성을 네트워크에 전파하도록 링크 등록 정보 `vlan-announce`를 설정할 수도 있습니다.

일반 VLAN 링크와 달리 VLAN으로 구성된 VNIC는 고유한 MAC 주소를 갖고 있습니다. 비VNIC VLAN에 대한 자세한 내용은 [Oracle Solaris 11.1 네트워크 성능 관리의 3 장, “VLAN 작업”](#)을 참조하십시오.

주 - 다음 절차에는 VLAN ID로 VNIC를 만들고 VNIC가 VLAN 트래픽을 지원하도록 설정하기 위해 적합한 등록 정보를 설정하는 단계만 포함됩니다. 등록 정보를 사용하여 설정하면 중간 포트 및 스위치만 자동으로 업데이트되지만 이러한 포인트에서 VLAN을 정의하려면 끝점이 별도로 구성되어 있어야 합니다.

1 관리자로 전환합니다.

자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

2 VLAN ID를 사용하여 VNIC를 만듭니다.

```
# dladm create-vnic -l link -v vid vnic
```

3 VNIC의 VLAN 구성을 네트워크에 브로드캐스트합니다.

```
# dladm set-linkprop -p vlan-announce=gvrp link
```

이 단계에서는 연결된 스위치에 VLAN ID를 자동으로 등록하는 GVRP(GARP VLAN 등록 프로토콜) 클라이언트 시스템을 사용하여 설정합니다. 기본적으로 `vlan-announce` 등록 정보는 `off`로 설정되며 네트워크에 VLAN 브로드캐스트 메시지가 전송되지 않습니다. 등록 정보를 `gvrp`로 설정한 후에는 네트워크 장치의 자동 VLAN 포트 구성을 사용하여 설정하기 위해 해당 링크에 대한 VLAN 구성이 전달됩니다. 따라서 이러한 장치에서 VLAN 트래픽을 수락하고 전달할 수 있습니다.

4 (선택 사항) VLAN 브로드캐스트 간의 대기 시간을 구성하려면 `gvrp-timeout` 등록 정보를 설정합니다.

```
# dladm set-linkprop -p gvrp-timeout=time link
```

여기서 `time`은 밀리초 단위입니다. 기본값은 250밀리초입니다. 로드가 높은 시스템은 VLAN 정보를 다시 브로드캐스트할 때 간격이 더 짧아야 할 수 있습니다. 이 등록 정보를 사용하면 간격을 조정할 수 있습니다.

5 (선택 사항) `vlan-announce` 및 `gvrp-timeout` 등록 정보의 값을 표시하려면 다음 명령을 사용합니다.

```
# dladm show-linkprop -p vlan-announce,gvrp-timeout
```

예 2-1 VNIC를 VLAN으로 구성

이 예제에서는 VLAN ID를 사용하여 VNIC를 만들고 VLAN 구성이 네트워크에 전달되도록 설정합니다.

```
# dladm create-vnic -l net0 -v 123 vnic0
# dladm set-linkprop -p vlan-announce=gvrp net0
# dladm show-linkprop -p vlan-announce,gvrp-timeout net0
LINK      PROPERTY      PERM  VALUE  DEFAULT  POSSIBLE
net0      vlan-announce rw    gvrp   off      gvrp,off
net0      gvrp-timeout  rw    250    250      --
```

가상 네트워크 구축

가상 네트워크는 영역 및 가상화 구성 요소를 조합합니다. 필요한 만큼 그리고 시스템에서 지원할 수 있는 만큼 영역을 여러 개 만듭니다. 각 영역은 고유한 가상 인터페이스를 가집니다. 시스템의 영역은 서로 통신할 수 있습니다. 가상 네트워크는 전체적으로 보다 큰 외부 네트워크의 대상에 연결합니다.

가상 네트워크 구축은 etherstub 또는 VNIC 구성을 위한 하나 이상의 단계 및 영역 구성을 위한 단계들로 구성됩니다. 이러한 단계는 독립적인 절차 세트이지만 가상 네트워크 구성을 완료하기 위해서는 두 가지를 모두 완료해야 합니다.

이 단원의 절차는 다음과 같은 가정을 기반으로 수행됩니다.

- 시스템의 가상 네트워크가 세 개의 영역으로 구성됩니다. 영역은 구성 단계가 서로 다릅니다. 첫번째 영역은 새 영역으로 만들어졌으며, 두번째 영역은 이미 시스템에 존재하고 VNIC를 사용하도록 재구성되어야 합니다. 세번째 영역은 개인 가상 네트워크로 설계되었습니다. 따라서 이 절차에서는 가상 네트워크용으로 영역을 준비하기 위한 여러 방법들을 보여줍니다.
- 시스템의 물리적 인터페이스는 IP 주소 192.168.3.70으로 구성됩니다.
- 라우터의 IP 주소는 192.168.3.25입니다

이 단원의 각 절차에서는 보다 구체적인 단계 설명을 제공하기 위해 세부 사항이 추가되어 있습니다.

가상 네트워크를 구축할 때 일부 단계는 전역 영역에서 수행되며, 일부 단계는 비전역 영역에서 수행됩니다. 혼동하지 않도록 각 절차의 예제에서는 특정 명령이 실행되는 영역이 프롬프트로 표시됩니다. 하지만 프롬프트에 표시되는 실제 경로는 시스템에 지정된 프롬프트에 따라 다를 수 있습니다.

이 단원에서는 다음과 같은 절차에 대해 설명합니다.

- 19 페이지 “가상 네트워크용으로 영역을 구성하는 방법”
- 21 페이지 “VNIC를 사용하도록 영역을 재구성하는 방법”
- 23 페이지 “개인 가상 네트워크를 만드는 방법”

▼ 가상 네트워크용으로 영역을 구성하는 방법

이 절차에서는 새 VNIC를 사용하여 새 영역을 구성하는 방법을 설명합니다. 이 절차에는 네트워크 가상화와 관련된 단계만 포함되어 있습니다. 영역 구성에 대한 자세한 지침은 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리의 17 장, “비전역 영역 계획 및 구성(작업)”**을 참조하십시오.

이 절차에서는 가상 네트워크에 대한 이 첫번째 영역을 새로운 영역으로 만든다고 가정합니다.

1 관리자로 전환합니다.

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

2 VNIC를 구성합니다.

15 페이지 “**VNIC 및 Etherstub 구성 방법**”을 참조하십시오. 하지만 이 절차에서는 etherstub을 만드는 단계를 생략합니다.

3 영역을 만듭니다.

```
global# zonecfg -z zone
```

영역을 만들 때는 ip-type 매개변수를 **exclusive**로 설정하고 이전에 만든 VNIC를 영역의 물리적 인터페이스에 지정했는지 확인합니다.

4 영역 구성 모드를 종료하려면 구성을 확인한 후 커밋합니다.

5 영역을 설치합니다.

```
global# zoneadm -z zone install
```

주 - 설치 프로세스에 오랜 시간이 걸릴 수 있습니다.

6 영역을 시작합니다.

```
global# zoneadm -z zone boot
```

7 영역이 완전히 부트된 후 영역에 로그인합니다.

```
# zlogin -C zone
```

8 메시지가 표시되면 정보를 제공합니다.

대부분의 정보는 선택 항목 목록에서 선택하여 제공합니다. 일반적으로 기본 옵션만으로도 충분합니다. 가상 네트워크를 구성하려면 다음 정보를 제공하거나 확인해야 합니다.

- 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

필요한 정보를 제공한 후 영역이 다시 시작됩니다.

예 2-2 가상 네트워크용으로 영역 구성

이 예제에는 zone1을 만들기 위한 자세한 단계가 포함됩니다. 하지만 가상 네트워크를 만드는 것과 관련된 영역 매개변수만 나열됩니다.

```
global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> set 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> commit
zonecfg:zone1> exit

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 -z zone1 boot

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)
..
```

네트워크 정보에 대해서는 다음과 같은 정보가 제공됩니다.

```

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

```

▼ VNIC를 사용하도록 영역을 재구성하는 방법

이 절차에서는 가상 네트워크의 두번째 영역을 참조합니다. 영역이 존재하지만 현재 구성으로는 가상 네트워크에 포함될 수 없습니다. 특히 영역의 IP 유형이 공유 유형이고 현재 인터페이스가 net0입니다. 이러한 구성을 모두 변경해야 합니다.

1 관리자로 전환합니다.

자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

2 VNIC를 만듭니다.

```
global# dladm create-vnic [-v vid] -l datalink vnic
```

여기서 *vid*는 VNIC에 지정한 VLAN ID를 나타냅니다. VNIC를 VLAN으로 만들려는 경우에만 VLAN ID를 지정합니다.

VNIC의 인터페이스는 아직 구성하지 마십시오. 이 단계는 이 절차의 이후에 수행합니다.

3 영역의 IP 유형을 shared에서 exclusive로 변경합니다.

```
global# zonecfg -z zone
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1>
```

4 VNIC를 사용하도록 영역의 인터페이스를 변경합니다.

```
zonecfg:zone1> remove net physical=NIC
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic
zonecfg:zone1:net> end
zonecfg:zone1>
```

5 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:zone1 verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global#
```

6 영역을 재부트합니다.

```
global# zoneadm -z zone reboot
```

7 영역에 로그인합니다.

```
global# zlogin zone
```

8 유효한 IP 주소로 VNIC를 구성합니다.

VNIC에 정적 주소를 지정하는 경우 다음을 입력합니다.

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

여기서 *address*에는 CIDR 표기법을 사용할 수 있습니다.

9 전역 영역에서 주소 정보를 /etc/hosts 파일에 추가합니다.**예 2-3 VNIC를 사용하도록 영역 구성 재구성**

이 예제에서 zone2는 공유 영역으로 존재합니다. 이 영역은 또한 가상 링크가 아닌 시스템의 기본 인터페이스를 사용합니다. vnic2를 사용하도록 zone2를 수정해야 합니다. vnic2를 사용하려면 zone2의 IP 유형을 먼저 exclusive로 변경해야 합니다. 가상 네트워크와 관련된 정보에만 집중할 수 있도록 일부 출력 결과가 잘려 있습니다.

```
global# dladm create-vnic -l net0 vnic2
```

```
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# zoneadm -z zone2 reboot
```

```
global# zlogin zone2
zone2# ipadm create-ip vnic2
zone2# ipadm create-addr -a 192.168.3.85/24 vnic2
ipadm: vnic2/v4
```

```
zone2# exit
```

```
global# vi /etc/hosts
#
::1          localhost
127.0.0.1    localhost
192.168.3.70 loghost   #For net0
192.168.3.80 zone1     #using vnic1
192.168.3.85 zone2     #using vnic2
```

▼ 개인 가상 네트워크를 만드는 방법

다음 절차에서는 가상 네트워크의 세번째 영역을 구성하는 방법을 설명합니다. 영역이 가상 네트워크에 속하지만 외부 시스템에서 액세스할 수 없습니다. 격리된 영역이 시스템 외부로 네트워크 트래픽을 전송할 수 있도록 하려면 NAT(네트워크 주소 변환)를 사용해야 합니다. NAT는 VNIC의 개인 IP 주소를 물리적 네트워크 인터페이스의 경로 지정 가능한 IP 주소로 변환합니다. 하지만 개인 IP 주소 자체는 외부 네트워크에서 볼 수 없습니다. NAT에 대한 자세한 내용은 **Oracle Solaris 11.1의 네트워크 보안의 “IP 필터의 NAT 기능 사용”**을 참조하십시오.

etherstub 사용은 일반적인 가상 네트워크 및 개인 가상 네트워크 간에 중요한 차이점이 있습니다. 개인 가상 네트워크에서는 영역에 지정되는 VNIC가 etherstub을 통해 구성됩니다. 따라서 시스템을 통과하는 네트워크 트래픽과 격리되어 있습니다.

이 절차에서는 영역이 이미 있지만 현재까지는 연결된 인터페이스가 없다고 가정합니다.

1 관리자로 전환합니다.

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

2 etherstub을 만듭니다.

```
global# dladm create-etherstub etherstub
```

3 etherstub에 VNIC를 만듭니다.

```
global# dladm create-vnic -l etherstub vnic
```

VNIC의 인터페이스는 아직 구성하지 마십시오. 이 단계는 이 절차의 이후에 수행합니다.

4 영역에 VNIC를 지정합니다.

```
global# zonecfg -z zone
zone# set physical=vnic
```

5 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:zone1 verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global#
```

6 영역에 로그인합니다.

```
# zlogin zone
```

7 영역에서 영역에 이제 지정된 VNIC를 통해 IP 인터페이스를 만듭니다.

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

8 유효한 IP 주소로 VNIC를 구성합니다.

VNIC에 정적 주소를 지정하는 경우 다음을 입력합니다.

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

여기서 *address*에는 CIDR 표기법을 사용할 수 있습니다.

9 전역 영역에서 주소 정보를 /etc/hosts 파일에 추가합니다.**10 전역 영역에서 IP 전달을 수행하도록 기본 인터페이스를 설정합니다.**

```
# ipadm set-ifprop -p forwarding=on -m ipv4 primary-interface
```

주 - 일반적으로 Oracle Solaris 11에서 기본 인터페이스는 net0 이름을 사용합니다.

11 전역 영역에서 기본 인터페이스에 대해 /etc/ipnat.conf 파일에서 NAT(네트워크 주소 변환)를 구성합니다.**12 NAT를 사용으로 설정하도록 IP 필터 서비스를 시작합니다.**

```
# svcadm enable network/ipfilter
```

13 영역을 재부트합니다.

```
# zoneadm -z zone reboot
```

예 2-4 개인 가상 네트워크 구성 만들기

이 예제에서 zone3은 개인 네트워크로 격리되도록 구성됩니다. NAT 및 IP 전달도 가상 개인 네트워크가 호스트 외부로 패킷을 전송하도록 구성되지만 개인 주소를 외부 네트워크로부터 숨깁니다. 영역은 이미 배타적 IP 유형으로 구성되어 있습니다. 하지만 여기에 지정된 IP 주소가 없습니다.

```
global# dladm create-etherstub ether0
global# dladm create-vnic -l ether0 vnic3
global# zonecfg -z zone3
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic3
zonecfg:zone3:net> end
zonecfg:zone3> verify
zonecfg:zone3> commit
zonecfg:zone3> exit
global#

global# zlogin zone3
zone3# ipadm create-ip vnic3
zone3# ipadm create-addr -a 192.168.0.10/24 vnic3
ipadm: vnic3/v4
zone3# exit

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
192.168.3.85 zone2 #using vnic2
192.168.0.10 zone3 #using vnic3

global# ipadm set-ifprop -p forwarding=on -m ipv4 vnic3

global# vi /etc/ipf/ipnat.conf
map vnic3 192.168.0.0/24 -> 0/32 portmap tcp/udp auto
map vnic3 192.168.0.0/24 -> 0/32

global# svcadm enable network/ipfilter
global# zoneadm -z zone3 boot

```

VNIC에 대한 기타 관리 작업

이 단원에서는 기본 구성을 수행한 후 VNIC에서 수행할 수 있는 작업을 설명합니다. 이 단원에서는 다음 항목을 다룹니다.

- 26 페이지 “VNIC의 VLAN ID 수정”
- 26 페이지 “VNIC MAC 주소 수정”
- 27 페이지 “VNIC 마이그레이션”
- 29 페이지 “VNIC 정보 표시”
- 29 페이지 “VNIC 삭제 방법”

주 - VNIC는 VLAN으로 구성할 수 있습니다. 이와 비슷한 하위 명령인 `dladm modify-vlan`을 사용하면 `dladm create-vlan` 명령을 사용하여 만든 직접 VLAN을 수정할 수 있습니다. VLAN 또는 VLAN으로 구성된 VNIC를 수정하는지에 따라 올바른 하위 명령을 사용해야 합니다. `dladm show-vlan` 하위 명령으로 표시된 VLAN에서 `modify-vlan` 하위 명령을 사용합니다. VLAN ID를 포함하여 `dladm show-vnic` 하위 명령으로 표시된 VNIC에서 `modify-vnic` 하위 명령을 사용합니다. 직접 VLAN을 수정하려면 [Oracle Solaris 11.1 네트워크 성능 관리의 “VLAN 수정”](#)을 참조하십시오.

두 가지 유형의 VNIC 수정을 사용할 수 있습니다.

- 전역 수정을 사용하면 특정 데이터 링크의 모든 VNIC에 대한 속성이 한 번에 변경됩니다. `-L` 옵션을 사용하면 수정하려는 VNIC가 포함된 기본 데이터 링크를 식별할 수 있습니다.
- 선택적인 수정을 사용하면 선택한 VNIC의 속성을 수정할 수 있습니다. 기본 데이터 링크를 식별하기 위해 `-L` 옵션을 사용하는 대신 변경하려는 속성의 VNIC를 지정합니다.

수정할 수 있는 속성은 VLAN ID, MAC 주소 및 기본 링크입니다. 기본 링크를 수정하면 VNIC가 다른 데이터 링크로 이동됩니다. 다음 단원에서는 이러한 수정에 대해 자세히 설명합니다.

VNIC의 VLAN ID 수정

VNIC의 VLAN ID를 변경하려면 다음 명령 중 하나를 사용합니다.

- `dladm modify-vnic -v vid -L datalink`

이 명령에서 *vid*는 VNIC에 지정하는 새로운 VLAN ID를 나타냅니다. *datalink*는 VNIC가 구성된 기본 링크를 나타냅니다. 데이터 링크에 단일 VNIC가 존재하는 경우에만 이 명령을 사용할 수 있습니다. 이러한 VNIC는 고유한 VLAN ID를 가져야 하기 때문에 VNIC가 여러 개 구성된 데이터 링크에서는 이 명령이 실패합니다.

- `dladm modify-vnic -v vid vnic`

단일 데이터 링크를 통해 여러 VNIC의 고유한 VLAN ID를 변경하려면 이 명령을 사용합니다. 각 VLAN ID는 동일 데이터 링크의 VNIC에 대해 고유하므로 한 번에 하나씩 VLAN ID를 변경해야 합니다. `net0`을 통해 구성된 `vnic0`, `vnicb0` 및 `vnicc0`의 VLAN ID를 변경한다고 가정해보십시오. 그러면 다음을 수행해야 합니다.

```
# dladm modify-vnic -v 123 vnic0
# dladm modify-vnic -v 456 vnicb0
# dladm modify-vnic -v 789 vnicc0
```

- `dladm modify-vnic -v vid vnic, vnic, [...]`

각 VNIC가 서로 다른 데이터 링크에 있는 경우 VNIC의 VLAN ID를 하나의 그룹으로 변경하려면 이 명령을 사용합니다. `vnic0`, `vnic1` 및 `vnic2`의 VLAN ID를 변경한다고 가정해보십시오. 이러한 VNIC는 각각 `net0`, `net1` 및 `net2`으로 구성됩니다. 다음 명령을 사용합니다.

```
# dladm modify-vnic -v 123 vnic0,vnic1,vnic2
```

VNIC MAC 주소 수정

VNIC는 고유한 MAC 주소를 포함합니다. 이러한 주소를 수정하려면 특정 상황에 따라 다음 명령 중 하나를 사용합니다.

- `dladm modify-vnic -m mac-address vnic`

특정 MAC 주소를 특정 VNIC에 지정하려면 이 명령을 사용합니다.

- `dladm modify-vnic -m random -L datalink`

이 명령은 데이터 링크에서 모든 VNIC의 MAC 주소를 변경하는 전역 수정을 수행합니다. 시스템은 고유한 MAC 주소를 VNIC에 자동으로 지정합니다. 이 명령에서 `-m random` 옵션은 `-m auto` 옵션과 동일합니다.

■ `dladm modif-vnic -m random vnic, vnic,[...]`

이 명령은 선택적인 VNIC 수정을 수행합니다. 전역 수정 및 선택적 수정 모두 `-m` 옵션에 대해 `random`을 지정합니다.

단일 명령으로 VNIC의 VLAN ID 및 MAC 주소를 수정할 수 있습니다. 하지만 예상치 않은 동작이 발생할 수 있으므로 여러 VNIC 속성을 전역으로 수정하기 위한 명령을 사용할 때는 주의가 필요합니다. VNIC에서 여러 속성을 한 번에 변경하는 것이 VNIC 그룹의 여러 속성을 동시에 변경하는 것보다 좋습니다.

다음 예제에서는 VNIC의 VLAN ID 및 MAC 주소를 수정하기 전과 수정한 후의 출력 결과를 보여줍니다.

```
# dladm show-vnic vnic0
LINK      OVER      SPEED    MACADDRESS      MACADDRTYPE     VID
vnic0     net0      1000    2:8:20:ec:c4:1d random           0
# dladm modify-vnic -m random -v 123 vnic0
# dladm show-vnic vnic0
LINK      OVER      SPEED    MACADDRESS      MACADDRTYPE     VID
vnic0     net0      1000    2:8:20:0:1:2   random           123
```

VNIC 마이그레이션

VNIC를 삭제하고 재구성하지 않고 하나 이상의 VNIC를 한 기본 데이터 링크에서 다른 기본 데이터 링크로 이동할 수 있습니다. 기본 링크는 물리적 링크, 링크 집계 또는 etherstub일 수 있습니다.

VNIC를 성공적으로 마이그레이션하기 위해서는 VNIC를 이동할 기본 데이터 링크가 VNIC의 데이터 링크 등록 정보를 수용할 수 있어야 합니다. 이러한 등록 정보가 지원되지 않을 경우 마이그레이션이 실패하고 사용자에게 알림이 제공됩니다. 마이그레이션을 성공한 후 VNIC가 네트워크에 연결된 상태로 유지되면 VNIC를 사용하는 모든 응용 프로그램이 정상적으로 작동합니다.

데이터 링크 상태, 링크 속도, MTU 크기 등의 특정 하드웨어에 종속된 등록 정보는 VNIC 마이그레이션 이후 변경될 수 있습니다. 이러한 등록 정보의 값은 VNIC가 마이그레이션된 데이터 링크에서 상속됩니다.

또한 VNIC를 전역으로 또는 선택적으로 마이그레이션할 수도 있습니다. 전역 마이그레이션은 모든 VNIC를 데이터 링크를 통해 다른 데이터 링크로 마이그레이션하는 것을 의미합니다. 전역 마이그레이션을 수행하기 위해서는 소스 데이터 링크와 대상 데이터 링크만 지정하면 됩니다. 다음 예제에서는 모든 VNIC를 `ether0`에서 `net1`로 이동합니다.

```
# dladm modify-vnic -l net1 -L ether0
```

구문 설명은 다음과 같습니다.

- `-l datalink`는 VNIC가 마이그레이션되는 대상 데이터 링크를 나타냅니다.
- `-L datalink`는 VNIC가 구성된 원본 데이터 링크를 나타냅니다.

주 - 소스 데이터 링크보다 대상 데이터 링크를 먼저 지정해야 합니다.

선택적 VNIC 마이그레이션을 수행하려면 이동하려는 VNIC를 지정합니다. 다음 예제에서는 선택한 VNIC를 `net0`에서 `net1`로 이동합니다.

```
# dladm modify-vnic -l net1 vnic0,vnic1,vnic2
```

주 - `-L` 옵션은 전역 수정으로만 제한됩니다.

VNIC 그룹을 마이그레이션할 때는 해당 VLAN ID도 동시에 수정할 수 있습니다. 하지만 새 VLAN ID를 지정하려면 다음 예제에 표시된 것처럼 한 번에 하나씩 VNIC를 마이그레이션해야 합니다.

```
# dladm modify-vnic -l net1 -v 123 vnic0
# dladm modify-vnic -l net1 -v 456 vnic1
# dladm modify-vnic -l net1 -v 789 vnic2
```

MAC 주소에 대한 마이그레이션 효과는 VNIC가 소스 데이터 링크에서 출하시 MAC 주소를 사용 중인지 여부에 따라 달라집니다.

- 마이그레이션 중에 `-m` 옵션을 지정하지 않으면 마이그레이션 후에 출하시 MAC 주소가 대상 데이터 링크로부터 임의로 지정된 주소로 바뀝니다.
- 마이그레이션 중 `-m address` 옵션을 사용할 경우 마이그레이션 후에 주소가 VNIC에 지정됩니다.

임의로 지정된 MAC 주소는 영향을 받지 않고 마이그레이션 후에도 해당 VNIC에 보존됩니다.

다음 예제에서는 여러 VNIC를 마이그레이션하는 방법을 보여줍니다. VNIC가 임의로 지정된 MAC 주소를 사용 중인지 확인하십시오. 따라서 이러한 주소는 마이그레이션 후에 변경되지 않습니다.

```
# dladm show-vnic
LINK      OVER      SPEED    MACADDRESS      MACADDRTYPE    VID
vnic1     net0      1000     2:8:20:c2:39:38 random           0
vnic2     net0      1000     2:8:20:5f:84:ff random           0

# dladm modify-vnic -l net1 -L net0
# dladm show-vnic vnic0
LINK      OVER      SPEED    MACADDRESS      MACADDRTYPE    VID
vnic1     net1      1000     2:8:20:c2:39:38 random           0
vnic2     net1      1000     2:8:20:5f:84:ff random           0
```

VNIC 정보 표시

시스템에서 VNIC에 대한 정보를 가져오려면 `dladm show-vnic` 명령을 사용합니다.

```
# 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
```

VNIC도 또한 데이터 링크입니다. 따라서 VNIC가 시스템에 존재할 경우 VNIC에 대한 정보를 포함하기 위해 데이터 링크에 대한 정보를 표시하는 모든 `dladm` 명령을 사용할 수도 있습니다. 예를 들어, `dladm show-link`는 목록에 VNIC를 포함합니다. 또는 `dladm show-linkprop` 명령을 사용하여 VNIC의 등록 정보를 확인할 수 있습니다. 단일 VNIC에 대한 등록 정보를 가져오려면 링크 등록 정보를 표시할 때 VNIC를 지정합니다.

```
# dladm show-linkprop [-p property] vnic
```

▼ VNIC 삭제 방법

이 절차에서는 시스템에서 VNIC 구성을 삭제하는 방법을 보여줍니다. 이 단계에서는 VNIC가 영역에 연결되어 있다고 가정합니다. 이 절차를 수행하려면 사용자가 전역 영역에 있어야 합니다.

- 1 VNIC가 영역에 연결되어 있으므로 영역을 중지합니다.

```
global# zoneadm -z zone halt
```

주 - 영역에 사용된 링크를 확인하려면 `dladm show-link` 명령을 사용합니다.

- 2 영역에서 VNIC를 제거하거나 분리합니다.

```
global# zonecfg -z zone remove net physical=vnic
```

- 3 시스템에서 VNIC를 삭제합니다.

```
global# dladm delete-vnic vnic
```

- 4 영역을 재부트합니다.

```
global# zonecfg -z zone boot
```

예 2-5 시스템에서 VNIC 삭제

이 예제에서 `vnic1`은 `zoneB` 및 시스템에서 제거됩니다.

```
Global# dladm show-link
LINK      CLASS  MTU    STATE  OVER
```

```
net0          phys  1500  up    --
net2          phys  1500  up    --
net1          phys  1500  up    --
net3          phys  1500  up    --
zoneA/net0    vnic  1500  up    net0
zoneB/net0    vnic  1500  up    net0
vnic0         vnic  1500  up    net1
zoneA/vnic0   vnic  1500  up    net1
vnic1         vnic  1500  up    net1
zoneB/vnic1   vnic  1500  up    net1
```

```
Global# zoneadm -z zoneB halt
Global# zonecfg -z zoneB remove net physical=vnic1
Global# dladm delete-vnic vnic1
Global# zonecfg -z zoneB reboot
```

Oracle Solaris의 네트워크 리소스 관리

이 장에서는 VNIC와 같은 가상 링크를 비롯한 데이터 링크의 리소스를 관리하는 방법에 대해 설명합니다. 네트워크 리소스 관리는 IP QoS(서비스 품질)를 구현하여 특히 가상 네트워크에서 성능을 향상시켜 줍니다.

다음 항목을 다룹니다.

- 31 페이지 “클라이언트, 전송 링 및 수신 링 작업”
- 40 페이지 “풀 및 CPU 작업”
- 44 페이지 “플로우의 리소스 관리”

클라이언트, 전송 링 및 수신 링 작업

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 시스템과 다른 네트워크 노드 간의 통신을 사용으로 설정하기 위해 NIC를 통해 구성됩니다. 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 클라이언트는 소프트웨어 기반 MAC 클라이언트입니다.

하드웨어 기반 클라이언트는 NIC에 따라 배타적으로 사용하도록 링을 지정할 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 기본 인터페이스(예: `net0`)에 최적화됩니다. 주 인터페이스를 **주 클라이언트**라고도 합니다. 다른 클라이언트의 배타적 사용에 지정되지 않은 사용 가능한 링은 모두 주 인터페이스에 자동으로 지정됩니다.

`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 11.1 조정 가능 매개변수 참조 설명서**를 참조하십시오.

소프트웨어 기반 클라이언트는 링을 배타적으로 사용하지 않습니다. 대신 링을 다른 기존의 소프트웨어 기반 클라이언트 또는 기본 클라이언트와 공유합니다. 소프트웨어 기반 클라이언트가 사용하는 링은 링 할당에서 우선순위를 갖는 하드웨어 기반 클라이언트 수에 따라 달라집니다.

기본 클라이언트 및 다른 **보조 클라이언트** 사이의 차이점을 이해하는 것이 중요합니다. 기본 클라이언트는 NIC의 물리적 데이터 링크입니다. 설치 중 Oracle Solaris에서 제공하는 일반 이름에 따라 기본 클라이언트는 이름이 `netN`으로 지정됩니다. 여기서 `N`은 인스턴스 번호입니다. 데이터 링크의 일반 이름에 대한 설명은 **Oracle Solaris 11 네트워킹 소개**의 “네트워크 장치 및 데이터 링크 이름”을 참조하십시오. VNIC는 물리적 데이터 링크를 통해 생성된 **보조 클라이언트**입니다. 이러한 클라이언트가 하드웨어 기반 클라이언트이면 링을 배타적으로 사용할 수 있습니다. 그렇지 않으면 클라이언트가 소프트웨어 기반입니다.

VLAN의 링 할당

VLAN을 사용하면 VLAN을 만드는 방법에 따라 링 지정이 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 지정할 수 있습니다.

링 할당에 대한 데이터 링크 등록 정보

링을 관리하려면 `dladm` 명령을 사용하여 두 개의 링 등록 정보를 설정할 수 있습니다.

- `rxrings`는 지정한 링크에 지정된 Rx 링 수를 나타냅니다.
- `txrings`는 지정한 링크에 지정된 Tx 링 수를 나타냅니다.

각 등록 정보를 세 가지 가능한 값 중 하나로 설정할 수 있습니다.

- `sw`는 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.
- `hw`도 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 할당을 지원하는 경우에만 `*rings` 등록 정보를 `hw`로 설정합니다.

현재 링 지정 및 사용에 대한 정보를 제공하려면 다음과 같은 추가 읽기 전용 링 등록 정보를 사용할 수 있습니다.

- `rxrings-available` 및 `txrings-available`은 할당할 수 있는 Rx 및 Tx 링 수를 나타냅니다.
- `rxhwclnt-available` 및 `txhwclnt-available`은 NIC에 구성할 수 있는 Rx 및 Tx 하드웨어 기반 클라이언트 수를 나타냅니다.

수신 및 전송 링 작업을 위한 명령

데이터 링크의 수신 및 전송 링 사용을 관리하려면 다음 주요 `dladm` 하위 명령을 사용합니다.

- `dladm show-linkprop` - Rx 및 Tx 링을 포함하여 링크 등록 정보의 현재 값을 표시합니다. 출력 결과에는 데이터 링크의 링 지원 기능에 대한 다음 정보가 제공됩니다. Rx 및 Tx 링을 사용하도록 구성할 수 있는 클라이언트 유형을 확인하려면 이 정보가 필요합니다.
 - 만들 수 있는 사용 가능한 클라이언트
 - 사용 가능한 클라이언트에 할당할 수 있는 사용 가능한 링
 - 동적 및 정적 링 할당을 지원할 수 있는 기능
 - 정적 링 할당만 지원될 경우 기존 클라이언트에 대한 현재 링 배포

33 페이지 “링 할당에 대한 데이터 링크 등록 정보”에서는 이 명령의 출력 결과를 해석하는 방법에 대해 보다 자세히 설명합니다.

- `dladm show-phys -H datalink` - 물리적 데이터 링크의 링이 기존 클라이언트에서 현재 사용되는 방식을 표시합니다.
- `dladm create-vnic -p ring-properties vnic` - 트래픽 서비스를 위해 사용할 특정 개수의 Tx 또는 Rx 링이 포함된 클라이언트를 만듭니다.
- `dladm set-linkprop -p ring-properties datalink` - 해당 링을 사용할 수 있고 링 할당이 지원된다는 가정하에 특정 클라이언트에 링을 할당합니다.

링 정보가져오기 및 해석

이 단원에서는 데이터 링크의 링 관련 등록 정보를 표시하는 `dladm show-linkprop` 출력 결과에 대해 설명합니다.

데이터 링크의 링 할당 기능 표시

이 단원에서는 링 관련 등록 정보에 대한 명령 출력 결과 예제를 제공하고 가져올 수 있는 정보 유형을 설명합니다. 예제에 사용된 NIC는 다음과 같습니다.

- `net0`(`nxge` 사용)
- `net1`(`ixgbe` 사용)
- `net2`(`e1000g` 사용)

예 3-1 nxge 링 정보

다음 예제에서는 `nxge`에 대한 링 정보를 보여줍니다.

```
# dladm show-linkprop net0
LINK      PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net0      rxrings           rw    --    --       sw,<1-7>
...
net0      txrings           rw    --    --       sw,<1-7>
...
net0      rxrings-available r-    5     --       --
net0      txrings-available r-    5     --       --
net0      rxhwcCnt-available r-    2     --       --
net0      txhwcCnt-available r-    2     --       --
...
```

`net0`에서는 POSSIBLE 필드의 값이 `rxrings` 및 `txrings`에 대해 `sw` 및 `<1-7>`입니다. 이러한 값은 `nxge`가 하드웨어 기반 클라이언트 및 소프트웨어 기반 클라이언트를 모두 지원함을 나타냅니다. `<1-7>` 범위는 클라이언트에 대해 설정할 수 있는 Rx 링 또는 Tx 링의 개수 한도를 나타냅니다. 범위는 또한 NIC가 수신 및 전송측에서 모두 동적 링 할당을 지원한다는 것을 나타냅니다.

또한 `*rings-available` 등록 정보는 Rx 링 5개와 Tx 링 5개를 하드웨어 기반 클라이언트에 할당할 수 있음을 나타냅니다.

예 3-1 nxge 링 정보 (계속)

하지만 *clnt-available 등록 정보는 사용 가능한 Rx 링을 배타적으로 사용할 수 있는 두 클라이언트만 구성할 수 있음을 보여줍니다. 마찬가지로, 사용 가능한 Tx 링을 배타적으로 사용할 수 있는 두 클라이언트만 구성할 수 있습니다.

예 3-2 ixgbe 링 정보

다음 예제에서는 ixgbe에 대한 링 정보를 보여줍니다.

```
# dladm show-linkprop net1
LINK  PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net1  rxrings           rw    --    --       sw,hw
...
net1  txrings           rw    --    --       sw,hw,<1-7>
...
net1  rxrings-available r-    0     --     --
net1  txrings-available r-    5     --     --
net1  rxhwclnt-available r-    0     --     --
net1  txhwclnt-available r-    7     --     --
...
```

net1의 경우 rxrings 및 txrings 모두에 대한 POSSIBLE 필드 값 sw 및 hw는 ixgbe가 하드웨어 기반 클라이언트와 소프트웨어 기반 클라이언트를 모두 지원함을 나타냅니다. Rx 링에는 정적 링 지정만 지원되며, 이 경우 하드웨어가 각 하드웨어 기반 클라이언트에 고정된 Rx 링 세트를 지정합니다. 하지만 Tx 링에서 <1-7> 범위는 동적 할당이 지원됨을 나타냅니다. 이 예제에서 하드웨어 기반 클라이언트를 최대 7개의 링에 지정하기 위해 Tx 링의 개수를 확인할 수 있습니다.

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 지원되지 않기 때문에 Rx 링을 배타적으로 사용하는 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 필드를 rxrings-available 및 txrings-available의 VALUE 필드와 비교하여 0의 의미를 확인할 수 있습니다.

예를 들어, 다음과 같이 txrings-available이 0이라고 가정합니다.

예 3-2 ixgbe 링 정보 (계속)

```
# dladm show-linkprop net1
LINK  PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net1  rxrings           rw    --    --       sw,hw
net1  txrings           rw    --    --       sw,hw,<1-7>
net1  rxrings-available r-    0     --     --
net1  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 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 3-3 e1000g 링 정보

다음 예제에서는 e1000g에 대한 링 정보를 보여줍니다.

```
# dladm show-linkprop net2
LINK  PROPERTY          PERM  VALUE  DEFAULT  POSSIBLE
...
net2  rxrings           rw    --    --       --
...
net2  txrings           rw    --    --       --
...
net2  rxrings-available r-    0     --     --
net2  txrings-available r-    0     --     --
net2  rxhwclnt-available r-    0     --     --
net2  txhwclnt-available r-    0     --     --
...
```

출력 결과에 따르면 e1000g에서 링 할당이 지원되지 않으므로 링과 하드웨어 기반 클라이언트를 모두 구성할 수 없습니다.

데이터 링크에 대한 링 사용 및 링 지정 표시

두 개의 읽기 전용 데이터 링크 등록 정보는 데이터 링크의 기존 클라이언트에서 링이 현재 사용되는 방법에 대한 정보를 제공합니다.

- rxrings-effective
- txrings-effective

링 사용 및 링이 클라이언트에 배포된 링에 대한 정보를 가져오려면 dladm show-linkprop 및 dladm show-phys -H 하위 명령을 모두 사용합니다.

다음 예제에서는 Rx 및 Tx 링 사용 및 이러한 링이 클라이언트 사이에 배포된 방식과 관련하여 두 명령으로 생성되는 서로 다른 출력 결과 유형을 보여줍니다.

예 3-4 기본 클라이언트의 링 사용

기본 클라이언트는 NIC의 물리적 데이터 링크를 통해 구성된 인터페이스입니다. 이 예제에서 NIC는 ixgbe 카드입니다. 기본적으로 해당 데이터 링크는 net0입니다. net0을 통한 IP 인터페이스가 기본 클라이언트입니다.

```
# dladm show-linkprop net0
LINK   PROPERTY                PERM VALUE  DEFAULT  POSSIBLE
...
net0   rxrings                 rw   --    --       sw,hw
net0   rxrings-effective      r-   2     --       --
net0   txrings                 rw   --    --       sw,hw,<1-7>
net0   txrings-effective      r-   8     --       --
net0   txrings-available      r-   7     --       --
net0   rxrings-available      r-   0     --       --
net0   rxhwclnt-available     r-   3     --       --
net0   txhwclnt-available     r-   7     --       --
...
# dladm show-phys -H net0
LINK   RINGTYPE  RINGS  CLIENTS
net0   RX        0-1    <default,mcast>
net0   TX        0-7    <default>net0
net0   RX        2-3    net0
net0   RX        4-5    --
net0   RX        6-7    --
```

출력 결과는 기본 클라이언트 net0에서 링의 사용 및 배포에 대한 다음 정보를 제공합니다.

- rxrings-effective는 net0이 두 개의 Rx 링을 자동으로 수신함을 나타냅니다. txrings-effective는 net0에서 8개의 Tx 링이 사용됨을 나타냅니다. 기본적으로 사용되지 않은 모든 링은 주 클라이언트에 자동으로 지정됩니다.
- dladm show-phys -H 명령을 기반으로 net0에 할당된 두 개의 Rx 링은 링 2 및 3입니다. Tx 링의 경우 net0은 링 0~7을 사용합니다.

예 3-5 보조 클라이언트의 링 사용

이 예제에서는 net0을 통해 VNIC 클라이언트 vnic1이 구성되었고 ixgbe 카드의 물리적 데이터 링크가 구성된 것으로 가정합니다.

```
# 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-   1     --       --
...
# dladm show-linkprop net0
LINK   PROPERTY                PERM VALUE  DEFAULT  POSSIBLE
...
net0   rxrings                 rw   --    --       sw,hw
net0   rxrings-effective      r-   2     --       --
net0   txrings                 rw   --    --       sw,hw,<1-7>
```

예 3-5 보조 클라이언트의 링 사용 (계속)

```

net0 txrings-effective r- -- -- --
net0 txrings-available r- 6 -- --
net0 rxrings-available r- 0 -- --
net0 rxhwclnt-available r- 3 -- --
net0 txhwclnt-available r- 6 -- --
...
# dladm show-phys -H net0
LINK RINGTYPE RINGS CLIENTS
net0 RX 0-1 <default,mcast>
net0 TX 0,2-7 <default>net0
net0 RX 2-3 net0
net0 RX 4-5 vnic1
net0 RX 6-7 --
net0 TX 1 vnic1

```

세 가지 명령으로 조합된 출력 결과는 다음 정보를 제공합니다.

- vnic1의 rxrings-effective는 이 VNIC가 두 개의 Rx 링을 자동으로 수신함을 나타냅니다. txrings-effective는 vnic1에서 한 개의 Tx 링이 사용됨을 나타냅니다. 이러한 링은 *ring 등록 정보에 대해 설정된 hw 값으로 표시된 것처럼 정적으로 할당됩니다.
- dladm show-phys -H 명령을 기반으로 net0에 할당된 두 개의 Rx 링은 링 2 및 3입니다. Tx 링의 경우 net0은 링 0 및 링 2~7을 사용합니다. vnic1은 Tx 링에 대해 링 1을 사용하고 Rx 링에 대해 링 4 및 5를 사용합니다.

vnic1은 정적 링 할당의 하드웨어 기반 클라이언트로 구성됩니다. 따라서 net0을 통해 만들 수 있는 사용 가능한 Tx 하드웨어 클라이언트(txhwclnt-available)의 개수는 6개로 줄어듭니다.

▼ 클라이언트 구성 및 링 할당 방법

이 절차에서는 링 할당에 대한 지원 유형을 기반으로 데이터 링크에 클라이언트를 구성하는 방법을 설명합니다. 34 페이지 “데이터 링크의 링 할당 기능 표시” 및 36 페이지 “데이터 링크에 대한 링 사용 및 링 지정 표시”에 설명된 대로 데이터 링크 링 등록 정보를 표시하는 dladm 명령의 출력 결과를 해석할 수 있는지 확인합니다. 이 정보는 클라이언트 구성 방식을 안내합니다.

1 데이터 링크의 링 등록 정보를 표시합니다.

```
# dladm show-linkprop datalink
```

출력 결과에서 다음을 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

2 이전 단계의 정보에 따라 다음 중 하나를 수행합니다.

- NIC가 동적 링 할당을 지원하는 경우 다음 구문을 통해 하드웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vnic -p rxrings=number[,txrings=number] -l link vnic
```

클라이언트가 이전에 만들어진 경우 다음 구문을 사용합니다.

```
# dladm set-linkprop -p rxrings=number[,txrings=number] vnic
```

주 - 일부 NIC는 Rx 링 또는 Tx 링에서 동적 링 할당을 지원하지만 두 유형에서 모두 지원하지는 않습니다. 동적 링 할당이 지원되는 링 유형에 대해 *number*를 지정합니다.

- NIC가 정적 링 할당을 지원하는 경우 다음 구문을 통해 하드웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vnic -p rxrings=hw[,txrings=hw] -l link vnic
```

클라이언트가 이전에 만들어진 경우 다음 구문을 사용합니다.

```
# dladm set-linkprop -p rxrings=hw[,txrings=hw] vnic
```

주 - 일부 NIC는 Rx 링 또는 Tx 링에서 정적 링 할당을 지원하지만 두 유형에서 모두 지원하지는 않습니다. 정적 링 할당이 지원되는 링 유형에 대해 *hw*를 지정합니다.

- NIC가 소프트웨어 기반 클라이언트만 지원하는 경우 다음 구문을 통해 하드웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vnic -p rxrings=sw[,txrings=sw] -l link vnic
```

클라이언트가 이전에 만들어진 경우 다음 구문을 사용합니다.

```
# dladm set-linkprop -p rxrings=sw[,txrings=sw] vnic
```

3 (옵션) 새로 만든 클라이언트의 링 정보를 확인합니다.

```
# dladm show-linkprop vnic
```

4 (선택 사항) 데이터 링크의 링이 여러 클라이언트 간에 배포된 방식을 확인합니다.

```
# dladm show-phys -H datalink
```

참조 Rx 및 Tx 링을 포함하여 가상 네트워크에서 네트워크 트래픽을 처리하기 위해 플로우를 사용하고 시스템 리소스를 할당하는 방법을 표시하는 예제를 보려면 [예 3-8](#)을 참조하십시오.

풀 및 CPU 작업

pool 링크 등록 정보를 사용하면 네트워크 처리를 CPU 풀에 바인딩할 수 있습니다. 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 전용으로 지정하여 네트워크 프로세스도 관리하려면 dladm set-linkprop 명령을 사용하여 링크의 pool 등록 정보를 구성합니다. 그런 다음 해당 링크를 영역에 지정합니다.

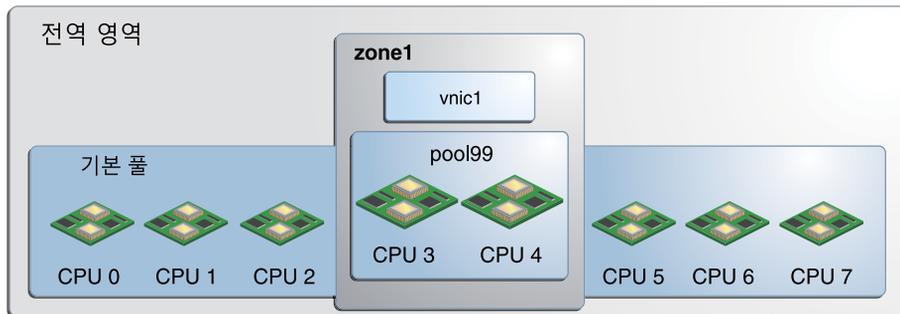
링크에 pool 등록 정보를 설정하고 링크를 영역의 네트워크 인터페이스로 지정하면 해당 링크가 영역의 풀에도 바인딩됩니다. 해당 영역이 배타적 영역이 되도록 설정된 경우 이 영역에 지정되지 않은 다른 데이터 링크가 풀의 CPU 리소스를 더 이상 사용할 수 없습니다.

주 - 특정 CPU를 데이터 링크에 지정하도록 개별 등록 정보인 cpu를 설정할 수 있습니다. cpu와 pool 등록 정보는 상호 배타적입니다. 두 등록 정보를 지정된 데이터 링크에 모두 설정할 수는 없습니다. cpu 등록 정보를 사용하여 데이터 링크에 CPU 리소스를 지정하려면 43 페이지 “링크에 CPU를 할당하는 방법”을 참조하십시오.

영역 내의 풀에 대한 자세한 내용은 [Oracle Solaris 11.1 관리: Oracle Solaris 영역](#), [Oracle Solaris 10 영역 및 리소스 관리의 13 장](#), “리소스 풀 만들기 및 관리(작업)”를 참조하십시오. 풀 만들기 및 풀에 CPU 세트 지정에 대한 자세한 내용은 poolcfg(1M) 매뉴얼 페이지를 참조하십시오.

다음 그림에서는 pool 등록 정보가 데이터 링크에 지정된 경우 풀 작동 방식을 보여줍니다.

그림 3-1 영역에 지정된 VNIC의 pool 등록 정보



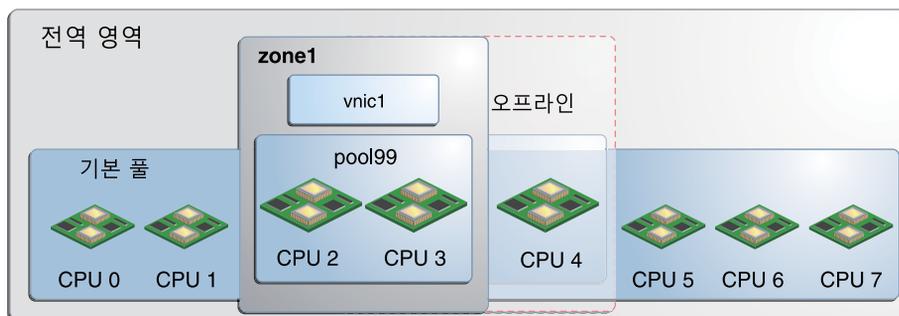
그림에서 시스템은 8개 CPU를 포함합니다. 시스템에 구성된 풀이 없을 경우 모든 CPU가 기본 풀에 속하며 전역 영역에서 사용됩니다. 하지만 이 예에서는 pool99 풀이

생성되었으며 CPU 3과 CPU 4로 구성됩니다. 이 풀은 배타적 영역인 zone1과 연결됩니다. pool99를 vnic1의 등록 정보로 설정하면 pool99는 vnic1의 네트워킹 프로세스 관리 전용이 됩니다. vnic1이 zone1의 네트워킹 인터페이스에 지정된 후 pool99의 CPU가 zone1의 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 지정되는 CPU가 결정됩니다. 데이터 링크에 대한 풀 변경 사항은 자동으로 구현되므로 해당 링크의 풀 관리가 간소화됩니다. 반면, cpu 등록 정보를 사용하여 링크에 특정 CPU를 지정하려면 지정할 CPU를 지정해야 합니다. 풀의 CPU 구성 요소를 변경할 때마다 cpu 등록 정보를 설정해야 합니다.

예를 들어, 그림 3-1에 시스템 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 3-2 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.

VNIC를 만들 때 pool 등록 정보를 설정하려면 다음 구문을 사용합니다.

```
# dladm create-vnic -p pool=pool-name -l link vnic
```

기존 VNIC의 pool 등록 정보를 설정하려면 다음 구문을 사용합니다.

```
# dladm setlinkprop -p pool=pool-name vnic
```

다음 절차에서는 VNIC에 대해 CPU 풀을 구성하는 방법을 설명합니다.

시작하기 전에 다음을 완료한 상태여야 합니다.

- 지정된 개수의 CPU를 사용하여 프로세서 세트를 생성했습니다.
- 프로세서 세트가 연결될 풀을 생성했습니다.
- 프로세서 세트와 풀을 연결했습니다.

주 - 이러한 필수 조건을 완료하는 지침은 [Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리의 “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 pool은 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 지정하는 방법을 설명하는 지침은 [Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리의 “영역 구성, 확인 및 커밋”](#)을 참조하십시오.

예 3-6 배타적 IP 유형을 사용하여 영역에 링크의 CPU 풀 지정

이 예에서는 영역의 데이터 링크에 풀이 지정되는 방식을 보여줍니다. 이 시나리오는 [그림 3-1](#)의 구성을 기반으로 합니다. 이 예에서는 pool99라는 CPU 풀이 영역에 대해 이미 구성되었다고 가정합니다. 그런 다음 VNIC에 풀이 지정됩니다. 최종적으로, 비전역 영역 zone1이 VNIC를 네트워크 인터페이스로 사용하도록 설정됩니다.

```
# dladm create-vnic -l net1 -p pool99 vnic1

# zonecfg -c zone1
zonecfg:zone1> set ip-type=exclusive
zonecfg:zone1> add net
zonecfg:zone1>net> set physical=vnic1
zonecfg:zone1>net> end
zonecfg:zone1> exit
```

▼ 링크에 CPU를 할당하는 방법

다음 절차에서는 cpu 등록 정보를 구성하여 데이터 링크를 순회하는 트래픽을 처리하도록 특정 CPU를 지정하는 방법에 대해 설명합니다.

1 인터페이스에 대한 CPU 지정을 확인합니다.

```
# dladm show-linkprop -p cpus link
```

기본적으로 특정 인터페이스에는 CPU가 지정되지 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 지정합니다.

CPU는 링크의 인터럽트가 연결된 CPU를 포함할 수 있습니다.

```
# dladm set-linkprop -p cpus=cpu1,cpu2,... link
```

여기서 *cpu1*은 링크에 지정할 CPU 번호입니다. 링크 전용으로 여러 CPU를 지정할 수 있습니다.

4 링크 인터럽트를 검사하여 새 CPU 지정을 확인합니다.

```
# echo ::interrupts | mdb -k
```

5 (옵션) 링크와 연결된 CPU를 표시합니다.

```
# dladm show-linkprop -p cpus link
```

예 3-7 링크에 CPU 할당

이 예제에서는 특정 CPU를 데이터 링크 net0에 전용으로 지정하는 방법을 보여줍니다.

여러 명령으로 생성된 출력 결과에서 다음 정보를 확인합니다. 알아보기 쉽도록 출력 결과에서 중요한 정보가 강조 표시됩니다.

- 기본적으로 net0에는 전용 CPU가 없습니다. 따라서 VALUE는 --입니다.
- net0의 인터럽트는 CPU 18과 연결됩니다.
- CPU가 할당된 후 net0은 VALUE 아래에 새 CPU 목록을 표시합니다.

```
# dladm show-linkprop -p cpus net0
LINK      PROPERTY  PERM    VALUE    DEFAULT  POSSIBLE
net0      cpus      rw      --       --       --

# echo ::interrupts | mdb -k
Device Shared Type  MSG #   State  INO   Mondo  Pil  CPU
net#0  no     MSI   2      enbl   0x1a  0x1a   6   18

# dladm set-linkprop -p cpus=14,18,19,20 net0

# dladm show-linkprop -p cpus net0
LINK  PROPERTY  PERM  VALUE    DEFAULT  POSSIBLE
net0  cpus      rw    14,18,19,20  --       --
```

인터럽트를 포함하여 모든 지원 스레드가 이제 새로 지정된 CPU 세트로 제한됩니다.

참조 CPU 및 CPU 풀을 포함하여 가상 네트워크에서 네트워크 트래픽을 처리하기 위해 플로우를 사용하고 시스템 리소스를 할당하는 방법을 표시하는 예제를 보려면 [예 3-8](#)을 참조하십시오.

플로우의 리소스 관리

플로우는 속성에 따라 구성된 네트워크 패킷으로 구성됩니다. 플로우를 사용하여 네트워크 리소스를 더 할당할 수 있습니다. 플로우 개요는 [12 페이지 “플로우를 사용한 네트워크 리소스 관리”](#)를 참조하십시오.

리소스 관리 플로우 사용 시에는 다음과 같은 일반 단계가 포함됩니다.

1. [12 페이지 “플로우를 사용한 네트워크 리소스 관리”](#)에 나열된 대로 특정 속성에 기반을 두도록 플로우를 만듭니다.
2. 네트워크 리소스와 관련된 등록 정보를 설정하여 플로우의 리소스 사용을 사용자 정의합니다. 현재 패킷 처리를 위한 대역폭만 설정할 수 있습니다.

▼ 플로우 구성 방법

- 1 필요에 따라 사용 가능한 링크를 나열하여 플로우를 구성할 링크를 확인합니다.

```
# dladm show-link
```

- 2 선택한 링크의 IP 인터페이스가 IP 주소로 올바르게 구성되었는지 확인합니다.

```
# ipadm show-addr
```

- 3 각 플로우에 대해 확인한 속성에 따라 플로우를 만듭니다.

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] flow
```

link 플로우를 구성 중인 링크를 나타냅니다.

attribute 네트워크 패킷을 플로우로 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 11.1에서 IP 서비스 품질 관리의 “DS 코드 포인트”](#)를 참조하십시오.

flow 특정 플로우에 지정하는 이름을 나타냅니다.

플로우 및 플로우 속성에 대한 자세한 내용은 [flowadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

- 4 (선택 사항) 데이터 링크의 대역폭에 대한 사용 가능한 값 범위를 표시합니다.

```
# dladm show-linkprop -p maxbw link
```

여기서 *link*는 플로우가 구성된 데이터 링크입니다.

값 범위는 POSSIBLE 필드에 나열됩니다.

- 5 플로우에 대역폭 공유를 할당합니다.

```
# flowadm set-flowprop -p maxbw=value flow
```

설정된 값이 링크 대역폭에 허용되는 값의 범위 내에 있어야 합니다.

주 - 현재 플로우의 대역폭만 사용자 정의할 수 있습니다.

- 6 (선택 사항) 해당 링크로 만든 플로우를 표시합니다.

```
# flowadm
```

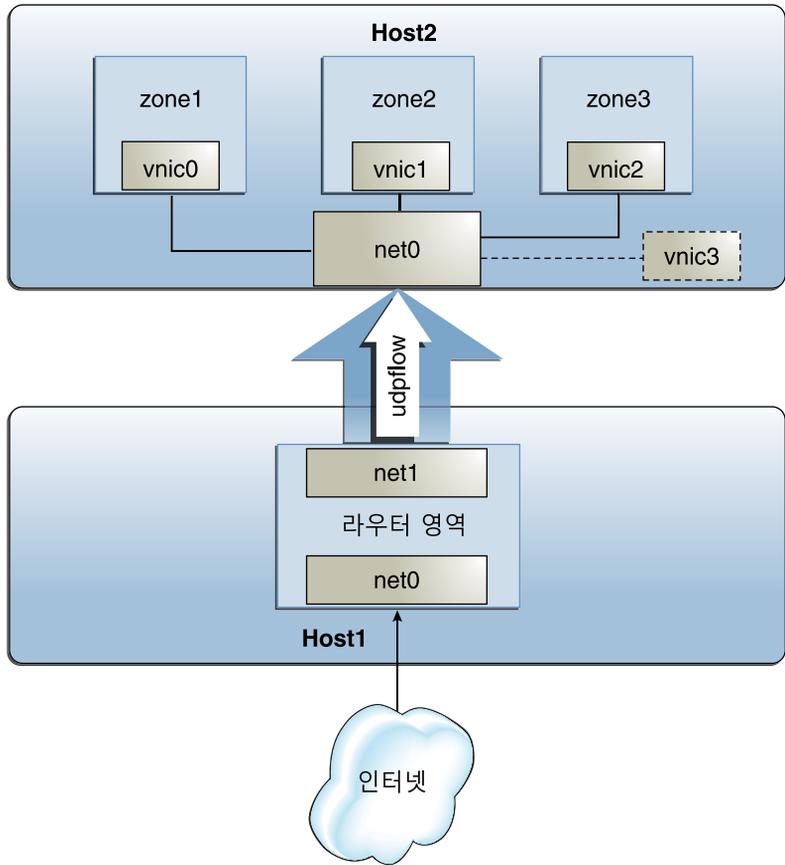
주 - flowadm 명령을 하위 명령 없이 사용하면 flowadm show-flow 명령과 동일한 정보를 제공합니다.

7 (선택 사항) 지정된 플로우에 대한 등록 정보 값을 표시합니다.

```
# flowadm show-flowprop flow
```

예 3-8 링크 및 플로우 등록 정보를 설정하여 리소스 관리

이 예에서는 데이터 링크와 플로우 모두에 네트워크 리소스를 할당하는 단계를 결합합니다. 이 예는 다음 그림에 표시된 구성을 기반으로 합니다.



또한 서로 연결된 두 개의 물리적 호스트를 보여줍니다.

- Host1의 구성은 다음과 같습니다.
 - 라우터 영역으로 사용되는 비전역 영역이 한 개 있습니다. 이 영역에는 인터페이스가 두 개 지정됩니다. net0은 인터넷에 연결하고 net1은 두번째 호스트를 포함하는 내부 네트워크에 연결합니다.
 - udpflow는 UDP 트래픽을 격리시키고 UDP 패킷의 리소스 사용 방식에 대한 제어를 구현하도록 net0으로 구성된 플로우입니다. 플로우 구성에 대한 자세한 내용은 44 페이지 “플로우의 리소스 관리”를 참조하십시오.
- Host2의 구성은 다음과 같습니다.
 - 비전역 영역 세 개와 해당 VNIC가 있습니다. VNIC는 해당 카드가 동적 링 할당을 지원하는 net0으로 구성됩니다. 링 할당에 대한 자세한 내용은 31 페이지 “클라이언트, 전송 링 및 수신 링 작업”을 참조하십시오.
 - 각 영역의 네트워크 처리 부하는 서로 다릅니다. 이 예제에서 zone1는 로드가 높고, zone2는 로드가 보통이고, zone3은 로드가 낮습니다. 해당 부하에 따라 이러한 영역에 리소스가 지정됩니다.
 - 별도의 VNIC가 소프트웨어 기반 클라이언트로 구성됩니다. MAC 클라이언트의 개요는 31 페이지 “MAC 클라이언트 및 링 할당”을 참조하십시오.

이 예의 작업은 다음과 같습니다.

- 플로우 만들기 및 플로우 제어 구성 - net1에 플로우가 생성되어 Host2에 수신되는 UDP 패킷에 대해 별도의 리소스 제어를 만듭니다.
- Host2에서 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 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리**의 17 장, “비전역 영역 계획 및 구성(작업)”을 참조하십시오.

먼저 Host1에서 링 및 IP 인터페이스에 대한 정보를 확인합니다.

```
# ipadm
NAME          CLASS/TYPE STATE   UNDER   ADDR
lo0           loopback  ok      --       --
  lo0/v4      static    ok      --       127.0.0.1/8
net0          ip        ok      --       --
  net0/v4     static    ok      --       10.10.6.5/24
net1          ip        failed  ipmp0    --
  net1/v4     static    ok      --       10.10.12.42/24
```

다음은 net1에 플로우를 만들어 UDP 트래픽을 Host2로 격리시킵니다. 플로우에 대한 리소스 제어를 구현합니다.

```
# flowadm add-flow -l net1 -a transport=udp udpflow
# flowadm set-flowprop -p maxbw=80 udpflow
```

만든 플로우에 대한 정보를 확인합니다.

```
flowadm
FLOW      LINK  IPADDR  PROTO  LPORT  RPORT  DFSLD
udpflow   net1  --      udp    --      --      --

# flowadm show-flowprop
FLOW      PROPERTY  VALUE      DEFAULT  POSSIBLE
udpflow   maxbw     80         --       --
```

Host2에서 net0에 각 영역에 대한 VNIC를 구성합니다. 각 VNIC에 대한 리소스 제어를 구현합니다. 해당 영역에 VNIC를 지정합니다.

```
# dladm create-vnic -l net0 vnic0
# dladm create-vnic -l net0 vnic1
# dladm create-vnic -l net0 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

# zonecfg -z zone1
# zonecfg:zone1> add net
# zonecfg:zone1:net> set physical=vnic0
# zonecfg:zone1:net> end
# zonecfg:zone1> commit
# zonecfg:zone1> exit
# zoneadm -z zone1 reboot

# zonecfg -z zone2
# zonecfg:zone2> add net
# zonecfg:zone2:net> set physical=vnic1
# zonecfg:zone2:net> end
# zonecfg:zone2> commit
# zonecfg:zone2> exit
# zoneadm -z zone2 reboot
#

# zonecfg -z zone3
# zonecfg:zone3> add net
# zonecfg:zone3:net> set physical=vnic2
# zonecfg:zone3:net> end
# zonecfg:zone3> commit
# zonecfg:zone3> exit
# zoneadm -z zone3 reboot
#
```

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=pool1 vnic0
```

최종적으로, 주 인터페이스인 net0과 링을 공유하는 소프트웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vnic -p rxrings=sw,txrings=sw -l net0 vnic3
```


Oracle Solaris에서 네트워크 트래픽 및 리소스 사용 모니터링

이 장에서는 물리적 및 가상 네트워크 환경 Oracle Solaris 11에서 네트워크 리소스의 사용에 대한 통계를 모니터링하고 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.

- 51 페이지 “네트워크 트래픽 플로우 개요”
- 54 페이지 “트래픽 통계 모니터링 명령”
- 54 페이지 “링크의 네트워크 트래픽에 대한 통계 수집”
- 58 페이지 “플로우의 네트워크 트래픽에 대한 통계 수집”
- 60 페이지 “네트워크 트래픽에 대한 네트워크 정산 구성”

네트워크 트래픽 플로우 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.

이 단원에서는 네트워크 레인의 개념을 소개합니다. **네트워크 레인**은 네트워크 트래픽을 관리하도록 할당된 시스템 리소스의 조합입니다. 따라서 네트워크 레인은 특정 네트워크 트래픽 유형에 대한 사용자 정의 경로입니다. 각 레인은 **하드웨어 레인** 또는 **소프트웨어 레인**일 수 있습니다. 또한 각 레인 유형은 **수신 레인** 또는 **전송 레인**일 수 있습니다.

하드웨어 및 소프트웨어 레인의 구분은 31 페이지 “클라이언트, 전송 링 및 수신 링 작업”에 설명된 대로 링 및 링 할당을 지원하는 NIC 기능에 따라 달라집니다. 이 장에서는 주로 수신 레인을 통해 받은 수신 트래픽에 중점을 둡니다.

주 - Rx 및 Tx 링과 기타 네트워크 리소스는 데이터 링크의 등록 정보를 설정하여 지정됩니다. 따라서 데이터 링크는 시스템에서 네트워크 레인입니다.

하드웨어 레인에서는 패킷이 해당 레인에 지정되는 링을 배타적으로 사용합니다. 반면, 소프트웨어 레인의 링은 해당 레인의 모든 네트워크 패킷에서 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.

다음 그림에서는 서로 다른 하드웨어 레인을 보여줍니다.

그림 4-1 하드웨어 레인

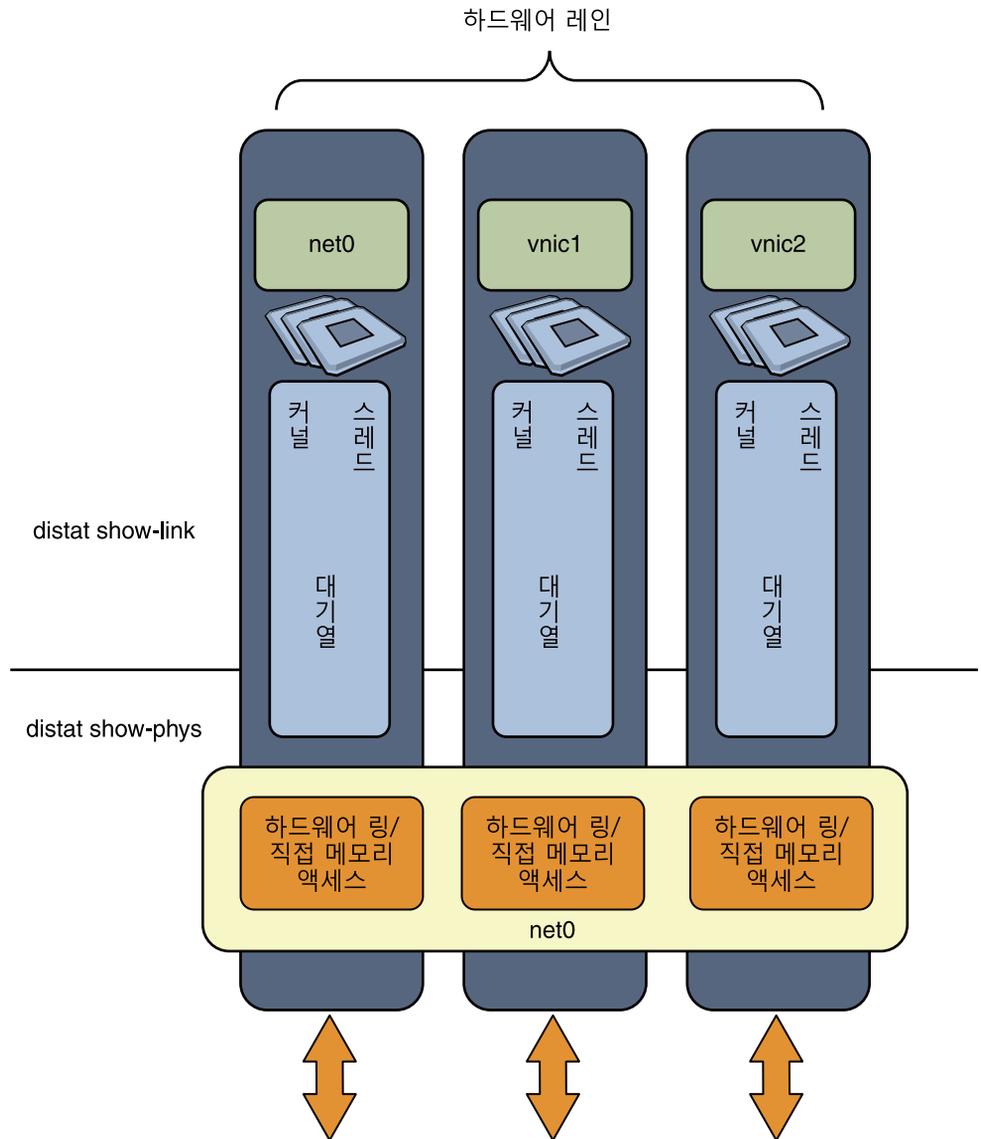


그림 4-1은 다음과 같은 구성을 보여줍니다.

- vnic1 및 vnic2의 두 VNIC가 net0으로 구성되어 있습니다. 기본 클라이언트인 net0은 기본 레인이기도 합니다. 보조 클라이언트인 VNIC도 네트워크 레인입니다.
- 보조 클라이언트에는 하드웨어 링 세트가 지정됩니다. 기본 클라이언트를 포함한 각 클라이언트는 하드웨어 레인으로 작동합니다.

- 각 라인에는 CPU 세트가 할당됩니다.

이후의 단원에서는 이러한 라인을 통해 이동하는 트래픽을 모니터링하는 방법을 설명합니다.

트래픽 통계 모니터링 명령

dlstat 및 flowstat 명령을 사용하면 각각 데이터 링크 및 플로우의 네트워크 트래픽에 대한 통계를 모니터링하고 가져올 수 있습니다. 이러한 명령은 dladm 및 flowadm 명령과 동일합니다. 다음 표에서는 *adm 명령 쌍의 기능과 *stat 명령 쌍의 기능을 비교해서 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	플로우를 구성 및 관리하기 위한 사용자 인터페이스 및 도구입니다.	flowstat 명령 옵션	플로우의 트래픽 통계를 가져오기 위한 사용자 인터페이스 및 도구입니다.

이후 단원에서는 각 모니터링 명령에 대해 보다 자세히 설명합니다.

링크의 네트워크 트래픽에 대한 통계 수집

dlstat 명령의 다음 변형을 사용하여 네트워크 트래픽 정보를 수집할 수 있습니다.

명령	제공되는 정보
dlstat [link] dlstat -rt [link] dlstat show-link [link]	라인당 인바운드 및 아웃바운드 트래픽 통계
dlstat show-link -rt [link]	라인당 링별 인바운드 및 아웃바운드 트래픽 통계
dlstat show-phys [link]	네트워크 물리 장치당 인바운드 및 아웃바운드 트래픽 통계

명령	제공되는 정보
<code>dlstat show-phys -rt</code> [link]	네트워크 물리 장치당 링별 인바운드 및 아웃바운드 트래픽 통계
<code>dlstat show-aggr</code> [link] <code>dlstat show-aggr -rt</code> [link]	집계당 포트별 인바운드 및 아웃바운드 트래픽 통계

`dlstat` 명령에 `-r` 옵션 또는 `-t` 옵션을 사용하여 통계 정보를 각각 수신측 또는 전송측으로 제한할 수 있습니다. 또한 `dlstat` 명령에 다른 옵션도 사용할 수 있습니다. 자세한 내용은 [dlstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

네트워크 장치에서 네트워크 트래픽 통계 가져오기

`dlstat show-phys` 하위 명령은 물리적 네트워크 장치를 나타내는 통계를 제공합니다. [그림 4-1](#)에 표시된 것처럼 이 하위 명령은 네트워크 스택의 장치 계층에 있는 하드웨어 링에서 작동합니다. 이와 동등한 하위 명령인 `dladm show-phys`도 동일한 레벨의 스택에서 작동합니다. [그림 4-1](#)을 [Oracle Solaris 11 네트워크링 소개의 “Oracle Solaris 11 구현”](#)에 설명된 네트워크 스택과 비교해보십시오.

다음 예제에서는 시스템의 모든 물리적 링크에 대한 통계를 보여줍니다.

```
# dlstat show-phys
LINK  IPKTS  RBYTES  OPKTS  OBYTES
net0  2.14M  257.48M  3.19M  210.88M
net1  1.15M  120.32M  1.00M  98.70M
net2  1.10M  110.10M  1.28   183.00M
...
```

출력 결과는 시스템의 각 링크에 대한 수신 및 송신 트래픽 통계를 모두 보여줍니다. 패킷 수 및 바이트 크기가 표시됩니다.

다음 예제에서는 `net0`의 각 하드웨어 링에 대한 수신측 통계를 보여줍니다.

```
# dlstat show-phys -r net0
LINK  TYPE  ID  INDEX  IPKTS  RBYTES
net0  rx    local  --    0      0
net0  rx    hw     1      0      0
net0  rx    hw     2    1.73M  2.61G
net0  rx    hw     3      0      0
net0  rx    hw     4    8.44M  12.71G
net0  rx    hw     5    5.68M  8.56G
net0  rx    hw     6    4.99M  7.38G
net0  rx    hw     7      0      0
```

두번째 출력에서 `net0` 장치는 `INDEX` 필드 아래에 식별된 8개 수신 링을 포함합니다. 링당 패킷의 균일한 배포는 링이 링크의 부하에 따라 링크에 올바르게 할당되었음을 나타내는 이상적인 구성입니다. 균일하지 않은 배포는 링크당 링의 부적절한 배포를

나타낼 수 있습니다. 불균등 배포는 NIC가 동적 링 할당을 지원하는지 여부에 따라 확인할 수 있습니다. 이 경우 패킷을 보다 고르게 처리할 수 있도록 링크당 링을 재배포할 수 있습니다. 동적 링 할당에 대한 자세한 내용은 31 페이지 “클라이언트, 전송 링 및 수신 링 작업”을 참조하십시오.

다음 예제에서는 장치에서 매초 수신되는 트래픽에 대한 정보를 보여줍니다. 간격은 -i 옵션을 사용하여 지정됩니다. 디스플레이의 새로 고침을 중지하려면 Ctrl-C를 누릅니다.

```
# dlstat show-phys -r -i 1
LINK  TYPE  INDEX  IPKTS  RBYTES
net0   rx     0     101.91K 32.86M
net1   rx     0      9.61M 14.47G
net2   rx     8       336K    0
net0   rx     0         0    0
net1   rx     0      82.13K 123.69M
net2   rx     0         0    0
...
^C
```

이 예제에서는 네트워크 장치로서 net1의 전송 링 사용량을 보여줍니다.

```
# dlstat show-phys -t net1
LINK  TYPE  INDEX  OPKTS  OBYTES
net1   tx     0        44  3.96K
net1   tx     1         0    0
net1   tx     2    1.48M 121.68M
net1   tx     3    2.45M 201.11M
net1   tx     4    1.47M 120.82M
net1   tx     5         0    0
net1   tx     6    1.97M 161.57M
net1   tx     7    4.59M 376.21M
net1   tx     8    2.43M 199.24M
net1   tx     9         0    0
net1   tx    10    3.23M 264.69M
net1   tx    11    1.88M 153.96M
```

라인에서 네트워크 트래픽 통계 가져오기

dlstat show-link 하위 명령은 물리적 링크를 통해 구성된 라인을 참조하는 통계를 제공합니다. 라인은 데이터 링크로 구성됩니다. 그림 4-1에 표시된 것처럼 이 하위 명령은 네트워크 스택의 데이터 링크 계층에서 작동합니다. 이와 동등한 하위 명령인 dladm show-link도 동일한 레벨의 스택에서 작동합니다. 그림 4-1을 **Oracle Solaris 11 네트워크 소개의 “Oracle Solaris 11 구현”**에 설명된 네트워크 스택과 비교해보십시오.

다음 예에서는 vnic0에 대한 수신측 트래픽 통계를 보여줍니다.

```
# dlstat show-link -r vnic0
LINK  TYPE  ID  INDEX  IPKTS  RBYTES  INTRS  POLLS  IDROPS
vnic0  rx    hw   2     1.73M  2.61G   1.33M  400.22K  0
vnic0  rx    hw   4     8.44M 12.71G   4.35M  4.09M   0
```

이전 출력 결과는 라인 `vnic0`의 트래픽 통계를 보여줍니다. 이 라인에는 두 개의 수신 링(링 2 및 링 4)을 배타적으로 사용하도록 할당되었습니다. 출력 결과에서는 수신 네트워크 트래픽에서 이러한 두 링이 어떻게 사용되는지를 보여줍니다. 하지만 데이터가 대역폭 제한 및 우선 순위 처리와 같은 다른 리소스 할당의 구현을 반영할 수도 있습니다.

기본 라인인 `net0`에 대해 다음 정보가 표시된다고 가정해보십시오.

```
# dlstat show-link -r net0
LINK TYPE ID INDEX IPKTS RBYTES INTRS POLLS IDROPS
net0 rx local -- 0 0 0 0 0
net0 rx sw -- 794.28K 1.19G 794.28K 0 0
...
```

출력 결과에 따라 Rx 링 중 하나인 링 0이 다른 클라이언트와 공유되었습니다. 보조 클라이언트가 할당된 링 없이 구성된 경우 링이 공유됩니다. 링은 다음과 같은 이유로 인해 할당되지 않을 수 있습니다.

- 링크를 통해 만들 수 있는 하드웨어 클라이언트가 더 이상 없습니다.
- 하드웨어 링을 더 이상 할당할 수 없습니다.
- 관리자가 의도적으로 소프트웨어 클라이언트를 구성했습니다.

인터럽트(INTRS) 및 삭제(*DROPS)에 대한 통계도 중요합니다. 인터럽트 수가 적고 패킷 삭제가 0이면 성능 효율성이 더 큼니다. 인터럽트 수 또는 패킷 삭제 수가 크면 라인에 리소스를 더 추가해야 할 수 있습니다.

다음 예제에서는 기본 라인인 `net1`에서 사용되는 링에서 아웃바운드 패킷에 대한 통계를 보여줍니다. 출력 결과에서는 `net1`이 모든 Tx 링을 사용하는 것을 보여줍니다.

```
# dlstat show-link -t net1
LINK TYPE ID INDEX OPKTS OBYTES ODROPS
net1 tx hw 0 32 1.44K 0
net1 tx hw 1 0 0 0
net1 tx hw 2 1.48M 97.95M 0
net1 tx hw 3 2.45M 161.87M 0
net1 tx hw 4 1.47M 97.25M 0
net1 tx hw 5 0 276 0
net1 tx hw 6 1.97M 130.25M 0
net1 tx hw 7 4.59M 302.80M 0
net1 tx hw 8 2.43M 302.80M 0
net1 tx hw 9 0 0 0
net1 tx hw 10 3.23M 213.05M 0
net1 tx hw 11 1.88M 123.93M 0
```

다음 명령에서는 링크 `net1`에 대한 수신측 통계 사용을 보여줍니다. 또한 명령에 `-F` 옵션을 사용하면 출력 결과에서 팬아웃 정보도 제공합니다. 구체적으로 팬아웃 수는 2(0 및 1)입니다. 링 0을 사용하는 하드웨어 라인에서 수신된 네트워크 트래픽은 분할되어 두 개의 팬아웃을 통해 전달됩니다. 마찬가지로, 링 1을 사용하는 하드웨어 라인에서 수신된 네트워크 트래픽도 분할되어 두 개의 팬아웃으로 나뉩니다.

```
# dlstat show-link -r -F net1
LINK ID INDEX FOUT IPKTS
```

```
net1 local -- 0 0
net1 hw 0 0 382.47K
net1 hw 0 1 0
net1 hw 1 0 367.50K
net1 hw 1 1 433.24K
```

링크 집계에서 네트워크 트래픽 통계 가져오기

`dlstat show-aggr` 명령은 트래픽이 시스템의 집계를 통과할 때 각 집계 포트에 대한 네트워크 패킷 통계를 보여줍니다.

```
# dlstat show-aggr
LINK      PORT      IPKTS      RBYTES      OPKTS      OBYTES
aggr1     --        0          0           0          0
aggr1     net0      0          0           0          0
aggr1     net1      0          0           0          0
```

출력 결과는 두 개의 기본 링크인 `net0` 및 `net1`이 포함된 링크 집계 `aggr1`의 구성을 나타냅니다. 네트워크 트래픽이 수신되거나 집계를 통해 시스템에서 전송되면 수신 및 송신 패킷과 해당 크기에 대한 정보가 모든 포트에 대해 보고됩니다. 포트는 집계의 기본 링크로 식별됩니다.

플로우의 네트워크 트래픽에 대한 통계 수집

플로우 통계는 시스템에 정의된 모든 플로우의 패킷 트래픽을 평가하는 데 도움이 됩니다. 플로우 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. 수신측 및 전송측 모두에 대한 통계를 표시하려면 옵션을 생략합니다.

`-i interval` 표시된 통계를 새로 고칠 시간(초)을 지정합니다. 이 옵션을 사용하지 않으면 정적 출력 결과가 표시됩니다.

`-l link` 지정된 데이터 링크의 모든 플로우에 대한 통계를 모니터링을 나타냅니다. 이 옵션을 사용하지 않으면 모든 데이터 링크의 모든 플로우에 대한 정보가 표시됩니다.

`flow` 지정된 플로우의 통계만 모니터링을 나타냅니다. 이 옵션을 사용하지 않으면 링크를 지정했는지 여부에 따라 모든 플로우 통계가 표시됩니다.

다음 예제에서는 시스템에서 구성된 플로우에 대한 정보를 표시하는 여러 방법을 보여줍니다.

예 4-1 1초 간격으로 모든 플로우에 대한 트래픽 통계 표시

이 예에서는 시스템에 구성된 모든 플로우의 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
```

이 예제에서는 구성된 모든 플로우의 송신 트래픽에 대한 통계를 보여줍니다.

예 4-2 모든 플로우에 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 4-3 지정한 링크의 모든 플로우에 대한 수신측 통계 표시

이 예에서는 net0 데이터 링크에 생성된 모든 플로우에서 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
```

예 4-3 지정한 링크의 모든 플로우에 대한 수신측 통계 표시 (계속)

```
udp-flow    9.52K    14.32M    0
^C
```

네트워크 트래픽에 대한 네트워크 정산 구성

확장 정산 기능을 사용하여 시스템에서 네트워크 정산을 설정할 수 있습니다. 네트워크 정산에는 로그 파일에서 네트워크 트래픽에 대한 통계 수집이 포함됩니다. 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.

네트워크 정산을 설정하려면 확장 정산 기능의 `acctadm` 명령을 사용합니다. 네트워크 정산 설정을 마친 후에는 `flowstat` 명령을 사용하여 트래픽 통계를 기록합니다.

이 단원에서는 다음 절차에 대해 설명합니다.

- 60 페이지 “네트워크 정산 설정 방법”
- 62 페이지 “네트워크 트래픽에 대한 기록 통계를 가져오는 방법”

▼ 네트워크 정산 설정 방법

- 1 네트워크 사용을 추적하려는 인터페이스가 있는 시스템에서 관리자가 됩니다. 자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.
- 2 확장 정산 기능으로 사용으로 설정될 수 있는 정산 유형의 상태를 확인합니다.

```
# acctadm [process | task | flow | net]
```

확장 정산 기능은 4가지 유형의 정산을 사용으로 설정할 수 있습니다. `acctadm` 명령의 선택적 피연산자는 이러한 정산 유형에 따라 달라집니다. 명령에 피연산자를 사용하면 특정 유형의 정산을 구성할 수 있습니다.

- 프로세스 정산
- 작업 정산
- IPQoS의 플로우 정산
- 네트워크 정산 링크 및 플로우

주 - 네트워크 계정은 44 페이지 “플로우의 리소스 관리”에 설명된 대로 `flowadm` 및 `flowstat` 명령을 통해 관리되는 플로우에도 적용됩니다. 따라서 이러한 플로우의 계정을 설정하려면 `acctadm` 명령에 `net` 옵션을 사용합니다. IPQoS 구성에 대한 플로우 정산을 사용으로 설정하는 `flow` 옵션을 사용하지 **마십시오**.

`net`을 지정하면 네트워크 계정의 상태가 표시됩니다. `net`을 사용하지 않으면 네 가지 계정 유형의 상태가 모두 표시됩니다.

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 4-4 시스템에서 네트워크 정산 설정

이 예제에서는 시스템에서 트래픽 내역 정보를 캡처하고 표시하기 위해 네트워크 정산을 구성하는 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 트래픽에 대한 기록 통계를 가져오는 방법

네트워크 계정을 사용으로 설정한 후 `dlstat` 및 `flowstat` 명령을 사용하여 로그 파일에서 정보를 추출할 수 있습니다. 이 절차에서는 단계를 설명합니다.

시작하기 전에 네트워크의 기록 데이터를 표시하려면 먼저 네트워크에 대해 확장 계정을 사용으로 설정해야 합니다. 또한 플로우의 트래픽에 대한 기록 데이터를 표시하려면 먼저 [44 페이지 “플로우의 리소스 관리”](#)에 설명된 대로 시스템에서 플로우를 구성해야 합니다.

- 1 **네트워크 사용을 추적하려는 인터페이스가 있는 시스템에서 관리자가 됩니다.**
자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 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	<code>acctadm</code> 명령을 사용하여 네트워크 계정을 사용으로 설정할 때 정의된 로그 파일을 지정합니다.
-d date	지정된 날짜에 대해 기록된 정보를 표시합니다.
-F format	분석용으로 구성할 수 있는 특정 형식으로 데이터를 표시합니다. 현재 지원되는 유일한 형식은 <code>gnuplot</code> 입니다.
-s start-time, -e end-time	특정 날짜와 시간 범위에 대해 사용 가능한 기록된 정보를 표시합니다. <code>MM/DD/YYYY, hh:mm:ss</code> 형식을 사용합니다. <code>hour(hh)</code> 는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 시간 범위가 현재 날짜로 지정된 데이터가 표시됩니다.
link	지정된 데이터 링크에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 데이터 링크에 대한 기록 네트워크 데이터가 표시됩니다.

3 구성된 플로우의 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	구성된 플로우의 수신 및 송신 패킷별로 리소스 사용에 대한 기록 정보의 요약을 표시합니다.
-a	데이터 캡처 후에 이미 삭제된 데이터 링크를 포함하여 모든 구성된 플로우의 리소스 사용을 표시합니다.
-f filename	acctadm 명령을 사용하여 네트워크 계정을 사용으로 설정할 때 정의된 로그 파일을 지정합니다.
-d	지정된 날짜에 대해 기록된 정보를 표시합니다.
-F format	데이터를 특정 형식으로 표시합니다. 현재 지원되는 유일한 형식은 gnuplot입니다.
-s start-time, -e end-time	특정 날짜와 시간 범위에 대해 사용 가능한 기록된 정보를 표시합니다. MM/DD/YYYY, hh:mm:ss 형식을 사용합니다. hour(hh)는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 시간 범위가 현재 날짜로 지정된 데이터가 표시됩니다.
flow	지정한 플로우에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 플로우에 대한 기록 네트워크 데이터가 표시됩니다.

예 4-5 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 net0
LINK DURATION IPACKETS RBYTES OPACKETS OBYTES BANDWIDTH
net0 80 1031 546908 0 0 2.44 Kbps
```

예 4-6 플로우의 리소스 사용에 대한 기록 정보 표시

다음 예에서는 플로우의 네트워크 트래픽 및 해당 리소스 사용에 대한 기록 통계를 표시하는 여러 가지 방법을 보여줍니다.

다음 예제에서는 플로우의 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
```

다음 예제에서는 지정된 날짜 및 시간 범위에서 플로우의 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

다음 예제에서는 지정된 날짜 및 시간 범위에서 플로우의 트래픽별 리소스 사용에 대한 기록 통계를 표시합니다. 정보는 gnuplot 형식을 사용하여 표시됩니다.

```
# 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
```

색인

A

acctadm 명령, 60-64

C

CPU, 13, 40
 CPU 할당, 43-44
CPU 풀, 13, 40
 기본 풀, 40
 링크에 지정, 42

D

dladm 명령
 create-etherstub, 11, 16
 create-vlan, 32
 create-vnic, 10-11, 13, 16
 delete-vnic, 29-30
 modify-vnic, 26
 set-linkprop, 11, 13
 show-linkprop, 33-34
 show-phys, 34
 show-vnic, 29
dlstat 명령, 51, 54
 show-aggr, 58
 show-link, 56-58
 show-phys, 55-56
DS 필드, 12

E

etherstub, 8-10, 15-16
 개인 가상 네트워크, 23-25
 만들기, 11, 16

F

flowadm 명령, 44-49
 add-flow, 13, 45
 help, 14
 set-flowprop, 13, 45
 show-flowprop, 46
flowstat 명령, 51, 54, 58-60

G

GVRP(GARP VLAN 등록 프로토콜), 17

I

interrupts, 44
ipadm 명령
 create-addr, 16
 create-ip, 16
 IP 정보 표시, 47

M

MAC 주소, VNIC, 8

MAC 클라이언트, 31
 구성, 38-39
 기본, 32, 37
 링 할당, 38-39
 보조, 32, 37
 소프트웨어 기반, 31
 하드웨어 기반, 31

N

NIC 링, 13
 “링 그룹화”참조

Q

QoS(서비스 품질), 11-12

S

SCTP, 12
SLA(서비스 단계 계약), 12

T

TCP, 12

U

UDP, 12

V

VLAN
 VNIC로, 17-18
 VNIC의 VLAN ID 수정, 26
VNIC, 7-11
 CPU 풀 리소스 지정, 42
 MAC 주소 수정, 26-27
 VLAN ID, 10

VNIC(계속)

 VLAN ID 사용, 17-18
 VLAN ID 수정, 26
 VNIC IP 인터페이스 만들기, 16
 개인 가상 네트워크, 23-25
 구성, 19-21
 기본 링크 변경, 27-28
 대역폭, 13
 등록 정보 설정, 13
 마이그레이션, 27-28
 만들기, 10-11, 16
 삭제, 29-30
 영역에 사용, 21-22
 영역에 지정, 21
 정보 표시, 29
VNIC 마이그레이션, 27-28

가

가상 네트워크
 “VNIC”참조
 etherstub, 15-16
 VNIC 구성, 15-16
 개인, 7-11
 구축, 18-25
 배타적 IP 유형 영역, 22
 영역 구성, 19-21, 21-22
 영역에 사용, 9
 외부 및 내부, 7-11
가상 네트워크 카드, “VNIC”참조
가상 스위치, 8-10
가상화 및 서비스 품질, 31

개

개인 가상 네트워크, 7-11
 etherstub으로 구성, 23-25

기

기본 클라이언트, 32, 37

내

내부 가상 네트워크, 7-11

네

네트워크 가상화, 7-11
 dladm 하위 명령 구현, 13
 etherstub, 8-10
 VNIC, 8-10
 가상 스위치, 8-10
 구성 요소, 8-10
 데이터 센터 통합, 10
 네트워크 레인, 11, 12
 소프트웨어 레인, 51
 하드웨어 레인, 51
 네트워크 리소스 관리, 31
 CPU, 40
 CPU 풀, 40
 dladm 하위 명령 구현, 13
 링크에, 11
 플로우 사용, 12
 네트워크 사용 모니터링, 51
 네트워크 정산, 60-64
 네트워크 통계, 54-58
 네트워크 사용 모니터링, 51
 네트워크 트래픽에 대한 장치 사용, 55-56
 레인의 네트워크 트래픽, 56-58
 집계 포트의 트래픽, 58
 트래픽 내역 정보, 61, 62-64
 플로우에 대한 리소스 사용, 58-60

대

대역폭
 VNIC, 13
 플로우, 13
 플로우에 설정, 45

데

데이터 링크
 리소스 제어 등록 정보, 11-12, 13

데이터 링크, 리소스 제어 등록 정보, 13

데이터 센터 통합, 10

동

동적 링 그룹화, 31-39

리

리소스 제어, 11-12, 31

링

링, 전송 및 수신, 31-39
 링 그룹화
 NIC에서 지원, 34-36
 VLAN, 위치에, 32
 데이터 링크 등록 정보, 33
 동적 및 정적, 31-39
 링 사용 및 링 지정, 36-38
 링 정보 해석, 34, 35, 36
 수신 및 전송 링, 33-34
 정보 표시, 33
 할당 단계, 38-39
 링 할당, “링 그룹화” 참조

보

보조 클라이언트, 32, 37

소

소프트웨어 기반 클라이언트, 31

속

속성, 플로우, 12

영

영역

- VNIC 지정, 21
- 가상 네트워크용으로 구성, 21-22
- 네트워크 가상화용으로 구성, 19-21

외

- 외부 가상 네트워크, 7-11

의

- 의사 이더넷 NIC, “etherstub”참조

인

- 인터럽트, 57
- 인터페이스 전용 CPU, 43-44

전

- 전송 프로토콜, 12

정

- 정적 링 그룹화, 31-39

패

- 패킷 삭제, 57

플

- 플로우, 12, 44-49
 - 대역폭 설정, 13, 45
 - 등록 정보 설정, 13
 - 만들기, 13, 45
 - 속성, 12

플로우 (계속)

- 정보 표시, 45
- 플로우 제어, 44-49

하

- 하드웨어 기반 클라이언트, 31
- 하드웨어 링, 31-39

확

- 확장 정산 기능, 60-64
 - IPQoS의 플로우 정산, 60
 - 링크 및 플로우에 대한 네트워크 정산, 60
 - 작업 정산, 60
 - 프로세스 정산, 60