

Oracle® Solaris 11.1에서 IP 서비스 품질 관리

Copyright © 1999, 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	7
1 IPQoS 소개(개요)	11
IPQoS 기본	11
차별화 서비스란?	11
IPQoS 기능	12
QoS(Quality-of-Service) 이론 및 실제에 대한 추가 정보를 얻을 수 있는 위치	12
IPQoS에서 QoS 제공	14
서비스 단계 계약 구현	14
개별 조직에 대해 QoS 보장	14
QoS 정책 소개	14
IPQoS를 사용하여 네트워크 효율성 향상	15
대역폭이 네트워크 트래픽에 미치는 영향	15
서비스 클래스를 사용하여 트래픽 우선 순위 지정	15
차등화 서비스 모델	16
분류기(ipgpc) 개요	17
측정기(tokenmt 및 tswtclmt) 개요	18
표시기(dscpmk 및 dlcosmk) 개요	18
플로우 계산(flowacct) 개요	19
IPQoS 모듈을 통한 트래픽 플로우 방식	19
IPQoS 사용 네트워크에서 트래픽 전달	21
DS 코드 포인트	21
흡별 동작	21
2 IPQoS 사용 네트워크 계획(작업)	25
일반 IPQoS 구성 계획(작업 맵)	25
Diffserv 네트워크 토폴로지 계획	26

Diffserv 네트워크에 대한 하드웨어 전략	26
IPQoS 네트워크 토폴로지	27
서비스 품질 정책 계획	29
QoS 정책 계획 지원	29
QoS 정책 계획(작업 맵)	30
▼ 네트워크에서 IPQoS를 준비하는 방법	31
▼ QoS 정책에 대한 클래스 정의 방법	31
필터 정의	33
▼ QoS 정책에서 필터를 정의하는 방법	34
▼ 플로우 제어 계획 방법	35
▼ 전달 동작 계획 방법	38
▼ 플로우 계산 계획 방법	40
IPQoS 구성 예 소개	41
IPQoS 토폴로지	41
3 IPQoS 구성 파일 만들기(작업)	45
IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)	45
QoS 정책을 만들기 위한 도구	47
기본 IPQoS 구성 파일	47
웹 서버에 대한 IPQoS 구성 파일 만들기	47
▼ IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법	49
▼ IPQoS 구성 파일에서 필터를 정의하는 방법	52
▼ IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법	53
▼ IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법	56
▼ 최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법	57
애플리케이션 서버에 대한 IPQoS 구성 파일 만들기	60
▼ 애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법	62
▼ IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법	64
▼ IPQoS 구성 파일에서 플로우 제어를 구성하는 방법	66
라우터에서 차별화 서비스 제공	69
▼ IPQoS 사용 네트워크에서 라우터를 구성하는 방법	69
4 IPQoS 시작 및 유지 관리(작업)	71
IPQoS 관리(작업 맵)	71
IPQoS 구성 적용	72

▼ IPQoS 커널 모듈에 새 구성을 적용하는 방법	72
▼ 재부트 때마다 IPQoS 구성이 적용되도록 하는 방법	73
IPQoS 메시지에 대한 syslog 로깅 사용	73
▼ 부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법	73
IPQoS 오류 메시지를 사용하여 문제 해결	74
5 플로우 계산 및 통계 수집 사용(작업)	79
플로우 계산 설정(작업 맵)	79
트래픽 플로우에 대한 정보 기록	80
▼ 플로우 계산 데이터에 대한 파일을 만드는 방법	80
통계 정보 수집	82
6 IPQoS 세부 정보(참조)	85
IPQoS 아키텍처 및 Diffserv 모델	85
분류기 모듈	86
측정기 모듈	88
표시기 모듈	91
flowacct 모듈	95
IPQoS 구성 파일	97
action 명령문	99
모듈 정의	99
class 절	100
filter 절	100
params 절	101
ipqosconf 구성 유틸리티	101
색인	103

머리말

Oracle Solaris 11.1에서 IP 서비스 품질 관리를 시작합니다. 이 설명서는 Oracle Solaris 시스템 관리 정보의 중요한 부분을 다루고 있는 여러 권으로 구성된 세트의 일부입니다. 이 설명서에서는 Oracle Solaris를 이미 설치했다고 가정합니다. 따라서 네트워크를 구성하거나 네트워크에 필요한 네트워킹 소프트웨어를 구성할 준비가 되어 있어야 합니다.

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

이 설명서의 대상

이 설명서는 네트워크에 구성된 Oracle Solaris를 실행하는 시스템 관리 책임자를 대상으로 작성되었습니다. 이 설명서를 사용하려면 적어도 2년의 UNIX 시스템 관리 경험이 있어야 합니다. UNIX 시스템 관리 교육 과정에 참석하는 것도 도움이 될 수 있습니다.

Oracle Support에 액세스

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

활자체 규약

다음 표는 이 책에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체	설명	예
AaBbCc123	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오. machine_name% you have mail.
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	machine_name% su Password:
AaBbCc123	새로 나오는 용어, 강조 표시할 용어입니다. 명령줄 변수를 실제 이름이나 값으로 바꾸십시오.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.
AaBbCc123	책 제목, 장, 절	사용자 설명서 의 6장을 읽으십시오. 캐시는 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 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#

IPQoS 소개(개요)

IPQoS(IP Quality of Service)를 통해 계산 통계 우선 순위 지정, 제어 및 수집이 가능합니다. IPQoS를 사용하면 네트워크 사용자에게 일관된 레벨의 서비스를 제공할 수 있습니다. 또한 트래픽 관리를 통해 네트워크 정체를 피할 수 있습니다.

다음은 이 장에 포함된 항목 목록입니다.

- 11 페이지 “IPQoS 기본”
- 14 페이지 “IPQoS에서 QoS 제공”
- 15 페이지 “IPQoS를 사용하여 네트워크 효율성 향상”
- 16 페이지 “차등화 서비스 모델”
- 21 페이지 “IPQoS 사용 네트워크에서 트래픽 전달”

주 - IPQoS 기능은 이후 릴리스에서 제거될 수 있습니다. 비슷한 대역폭 리소스 제어 기능을 지원하는 `dladm`, `flowadm` 및 관련 명령을 대신 사용하는 것이 좋습니다. 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

IPQoS 기본

IPQoS는 IETF(Internet Engineering Task Force)의 Differentiated Services Working Group에서 정의한 Diffserv(차별화 서비스) 아키텍처를 사용으로 설정합니다. Oracle Solaris에서 IPQoS는 TCP/IP 프로토콜 스택의 IP 레벨에서 구현됩니다.

차별화 서비스란?

IPQoS를 사용으로 설정하면 선택한 고객 및 선택한 응용 프로그램에 대해 서로 다른 레벨의 네트워크 서비스를 제공할 수 있습니다. 서로 다른 레벨의 서비스를 총칭하여 **차별화 서비스**라고 합니다. 고객에게 제공하는 차별화 서비스는 회사에서 고객에게

제공하는 서비스 레벨의 구조를 기준으로 할 수 있습니다. 또한 네트워크의 응용 프로그램이나 사용자에게 대해 설정된 우선 순위를 기준으로 차별화 서비스를 제공할 수 있습니다.

QoS 제공에는 다음 작업이 포함됩니다.

- 서로 다른 그룹(예: 고객 또는 엔터프라이즈의 부서)에 서비스 레벨 지정
- 특정 그룹이나 응용 프로그램에 제공되는 네트워크 서비스 우선 순위 지정
- 네트워크 병목 영역 및 기타 형태의 정체 발견 및 제거
- 네트워크 성능 모니터링 및 성능 통계 제공
- 네트워크 리소스 간의 대역폭 규제

IPQoS 기능

IPQoS에는 다음과 같은 기능이 있습니다.

- QoS 정책 구성을 위한 `ipqosconf` 명령줄 도구
- 조직의 QoS 정책을 구성하는 필터를 기준으로 작업을 선택하는 분류기
- Diffserv 모델과 호환되어 네트워크 트래픽을 측정하는 측정 모듈
- 패킷의 IP 헤더를 전달 정보로 표시하는 기능을 기반으로 하는 서비스 차별화
- 트래픽 플로우에 대한 통계를 수집하는 플로우 계산 모듈
- UNIX® `kstat` 명령을 통해 트래픽 클래스에 대한 통계 수집
- SPARC® 및 x86 아키텍처 지원
- IPv4 및 IPv6 주소 지원
- IP 보안 아키텍처(IPsec)와 상호 운용성
- VLAN(virtual local area networks)에 대한 802.1D 사용자 우선 순위 표시 지원

QoS(Quality-of-Service) 이론 및 실체에 대한 추가 정보를 얻을 수 있는 위치

차별화 서비스 및 QoS에 대한 정보는 서적 및 온라인 소스에서 찾을 수 있습니다.

QoS 관련 서적

QoS 이론 및 실체에 대한 자세한 내용은 다음 서적을 참조하십시오.

- Ferguson, Paul 및 Geoff Huston. *Quality of Service*. John Wiley & Sons, Inc., 1998.
- Kilkki, Kalevi. *Differentiated Services for the Internet*. Macmillan Technical Publishing, 1999.

QoS에 대한 RFC(Requests for Comments)

IPQoS는 다음 RFC 및 다음 인터넷 초안에 기술된 사양을 준수합니다.

- RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (<http://www.ietf.org/rfc/rfc2474.txt?number=2474>) - 차별화 서비스 지원을 위한 IPv4 및 IPv6 패킷 헤더의 ToS(type of service) 필드 또는 DS 필드에 대한 향상된 기능을 설명합니다.
- RFC 2475, An Architecture for Differentiated Services (<http://www.ietf.org/rfc/rfc2475.txt?number=2475>) - Diffserv 구조의 구성 및 모듈에 대한 자세한 설명을 제공합니다.
- RFC 2597, Assured Forwarding PHB Group (<http://www.ietf.org/rfc/rfc2597.txt?number=2597>) - 홉별 AF(assured forwarding) 동작이 어떻게 작동하는지 설명합니다.
- RFC 2598, An Expedited Forwarding PHB (<http://www.ietf.org/rfc/rfc2598.txt?number=2598>) - 홉별 EF(expedited forwarding) 동작이 어떻게 작동하는지 설명합니다.
- 인터넷 초안, *An Informal Management Model for Diffserv Routers* - 라우터에서 Diffserv 아키텍처 구현을 위한 모델을 제공합니다.

QoS 정보 웹 사이트

IETF의 Differentiated Services Working Group은 Diffserv 인터넷 초안에 대한 링크를 제공하는 웹 사이트(<http://www.ietf.org/html.charters/diffserv-charter.html>)를 유지 관리합니다.

라우터 제조업체(예: Cisco Systems 및 Juniper Networks)는 차별화 서비스가 자사 제품에서 어떻게 구현되는지 설명하는 자사 웹 사이트에서 정보를 제공합니다.

IPQoS 매뉴얼 페이지

IPQoS 설명서에는 다음 매뉴얼 페이지가 포함되어 있습니다.

- [ipqosconf\(1M\)](#) - IPQoS 구성 파일 설정을 위한 명령을 설명합니다.
- [ipqos\(7ipp\)](#) - Diffserv 아키텍처 모델의 IPQoS 구현을 설명합니다.
- [ipgpc\(7ipp\)](#) - Diffserv 분류기의 IPQoS 구현을 설명합니다.
- [tokenmt\(7ipp\)](#) - IPQoS tokenmt 측정기를 설명합니다.
- [tswtclmt\(7ipp\)](#) - IPQoS tswtclmt 측정기를 설명합니다.
- [dscpmk\(7ipp\)](#) - DSCP 표시기 모듈을 설명합니다.
- [dlcosmk\(7ipp\)](#) - IPQoS 802.1D 사용자 우선 순위 표시기 모듈을 설명합니다.
- [flowacct\(7ipp\)](#) - IPQoS 플로우 계산 모듈을 설명합니다.
- [acctadm\(1M\)](#) - Oracle Solaris 확장 정산 기능을 구성하는 명령을 설명합니다. acctadm 명령에는 IPQoS 확장이 포함됩니다.

IPQoS에서 QoS 제공

IPQoS 기능을 통해 인터넷 서비스 제공자(ISP) 및 응용 프로그램 서비스 제공자(ASP)는 고객에게 서로 다른 레벨의 네트워크 서비스를 제공할 수 있습니다. 이러한 기능을 통해 개별 회사 및 교육 기관은 내부 조직이나 주요 응용 프로그램에 대한 서비스 우선 순위를 지정할 수 있습니다.

서비스 단계 계약 구현

조직이 ISP 또는 ASP인 경우 IPQoS 구성을 회사에서 고객에게 제공하는 SLA(서비스 단계 계약)를 기준으로 할 수 있습니다. SLA에서 서비스 제공자는 고객에게 가격 구조를 기준으로 특정 레벨의 네트워크 서비스를 보장합니다. 예를 들어, 높은 가격의 SLA는 고객이 모든 유형의 네트워크 트래픽에 대해 24시간 내내 가장 높은 우선 순위를 가지도록 보장할 수 있습니다. 반면, 중간 가격의 SLA는 고객이 업무 시간 중 전자 메일에 대해서만 높은 우선 순위를 가지도록 보장할 수 있습니다. 기타 모든 트래픽은 24시간 내내 중간 우선 순위를 가집니다.

개별 조직에 대해 QoS 보장

조직이 엔터프라이즈이거나 기관인 경우 해당 네트워크에 대한 QoS를 제공할 수도 있습니다. 특정 그룹이나 특정 응용 프로그램의 트래픽이 더 높거나 낮은 서비스 레벨을 가지도록 보장할 수 있습니다.

QoS 정책 소개

QoS(서비스 품질) 정책을 정의하여 서비스 품질을 구현합니다. QoS 정책은 고객 또는 응용 프로그램의 우선 순위 및 서로 다른 범주의 트래픽 처리를 위한 작업과 같이 다양한 네트워크 속성을 정의합니다. IPQoS 구성 파일에서 조직의 QoS 정책을 구현합니다. 이 파일은 Oracle Solaris 커널에 상주하는 IPQoS 모듈을 구성합니다. IPQoS 정책이 적용된 호스트는 *IPQoS 사용 시스템*으로 간주됩니다.

QoS 정책에서는 일반적으로 다음을 정의합니다.

- 서비스 클래스라고 하는 고유의 네트워크 트래픽 그룹
- 각 클래스에 대한 네트워크 트래픽의 양을 규제하기 위한 측정 단위. 이러한 측정 단위는 *측정*이라고 하는 트래픽 측정 프로세스를 제어합니다.
- IPQoS 시스템 및 Diffserv 라우터가 패킷 플로우에 적용해야 하는 작업. 이 유형의 작업을 *호출 동작(PHB)*이라고 합니다.
- 조직이 서비스 클래스에 대해 필요로 하는 통계 수집. 예를 들면 고객이나 특정 응용 프로그램이 생성하는 트래픽입니다.

패킷이 네트워크에 전달될 때 IPQoS 사용 시스템은 패킷 헤더를 검사합니다. IPQoS 시스템이 수행하는 작업은 QoS 정책으로 결정됩니다.

QoS 정책 설계를 위한 작업은 29 페이지 “서비스 품질 정책 계획”에 설명되어 있습니다.

IPQoS를 사용하여 네트워크 효율성 향상

IPQoS에는 서비스 품질을 구현할 때 네트워크 성능을 더욱 효율화할 수 있는 기능이 포함되어 있습니다. 컴퓨터 네트워크가 확장되면 사용자 수 및 더욱 강력한 프로세서 증가로 생성되는 네트워크 트래픽 관리의 필요성도 높아집니다. 과다 사용되는 네트워크의 증상으로는 데이터 손실 및 트래픽 정체를 들 수 있습니다. 두 증상은 모두 느린 응답 시간이라는 결과를 초래합니다.

과거에는 시스템 관리자가 더 많은 대역폭을 추가하여 네트워크 트래픽 문제를 처리했습니다. 링크에서 트래픽 레벨은 광범위하게 변동하는 경우가 많습니다. IPQoS를 사용하면 기존 네트워크의 트래픽을 관리하고 확장이 필요한지 여부 및 필요한 위치를 평가할 수 있습니다.

예를 들어, 엔터프라이즈나 기관의 경우 트래픽 병목 현상을 피하려면 효율적인 네트워크를 유지 관리해야 합니다. 또한 그룹이나 응용 프로그램에서 할당된 대역폭보다 많이 소비하지 않도록 해야 합니다. ISP 또는 ASP의 경우, 고객이 지불한 레벨의 네트워크 서비스를 받도록 네트워크 성능을 관리해야 합니다.

대역폭이 네트워크 트래픽에 미치는 영향

IPQoS를 사용하여 네트워크 대역폭(완전히 사용된 네트워크 링크나 장치에서 전송할 수 있는 최대 데이터 양)을 규제할 수 있습니다. QoS 정책에서 대역폭 사용 우선 순위를 지정하여 고객이나 사용자에게 QoS를 제공해야 합니다. IPQoS 측정 모듈을 통해 IPQoS 사용 호스트에서 다양한 트래픽 클래스 간에 대역폭 할당을 측정하고 제어할 수 있습니다.

네트워크의 트래픽을 효과적으로 관리할 수 있으려면 먼저 대역폭 사용에 대한 다음 질문에 답해야 합니다.

- 귀사의 로컬 네트워크에서 트래픽 문제가 있는 영역은 어디입니까?
- 사용 가능한 대역폭을 최대한 사용하기 위해 무엇을 해야 합니까?
- 우선 순위가 가장 높은 사이트의 중요 응용 프로그램은 무엇입니까?
- 정체에 민감한 응용 프로그램은 무엇입니까?
- 낮은 우선 순위로 지정해도 되는 덜 중요한 응용 프로그램은 무엇입니까?

서비스 클래스를 사용하여 트래픽 우선 순위 지정

서비스 품질을 구현하려면 네트워크 트래픽을 분석하여 트래픽을 분할할 수 있는 모든 포괄적인 그룹 지정은 식별합니다. 그런 다음 여러 그룹을 개별 특성 및 개별 우선 순위를 가지는 서비스 클래스로 조직합니다. 이러한 클래스는 조직에 대한 QoS 정책의 기준이 되는 기본 범주를 형성합니다. 서비스 클래스는 제어할 트래픽 그룹을 나타냅니다.

예를 들어, 제공자는 계단식 가격 구조로 프리미엄, 골드, 실버 및 브론즈 레벨의 서비스를 제공할 수 있습니다. 프리미엄 SLA는 ISP가 고객을 위해 호스트하는 웹 사이트를 대상으로 한 수신 트래픽에 대해 가장 높은 우선 순위를 보장합니다. 따라서 고객의 웹 사이트에 대한 수신 트래픽이 하나의 트래픽 클래스가 될 수 있습니다.

엔터프라이즈의 경우, 부서 요구 사항을 기준으로 서비스 클래스를 만들 수 있습니다. 또는 네트워크 트래픽에서 특정 응용 프로그램의 수를 기준으로 클래스를 만들 수 있습니다.

다음은 엔터프라이즈에 대한 트래픽 클래스의 몇 가지 예입니다.

- 특정 서버에 대한 전자 메일 및 나가는 FTP와 같이 자주 사용되는 응용 프로그램(둘 다 하나의 클래스가 될 수 있음). 직원들은 이러한 응용 프로그램을 지속적으로 사용하므로 QoS 정책에서 전자 메일 및 나가는 FTP에 대해 적은 양의 대역폭과 낮은 우선 순위를 보장할 수 있습니다.
- 하루 24시간 실행되어야 하는 주문 입력 데이터베이스. 엔터프라이즈에 대한 데이터베이스 응용 프로그램의 중요도에 따라 데이터베이스에 많은 양의 대역폭과 높은 우선 순위를 지정할 수 있습니다.
- 중요한 업무 또는 민감한 업무를 수행하는 부서(예: 급여 부서). 조직에 대한 부서의 중요도에 따라 해당 부서에 지정하는 우선 순위 및 대역폭의 양이 결정됩니다.
- 회사의 외부 웹 사이트로 들어오는 호출. 이 클래스에는 낮은 우선 순위로 실행되는 적당한 양의 대역폭을 지정할 수 있습니다.

차등화 서비스 모델

IPQoS에는 RFC 2475에 정의된 차등화 서비스(Diffserv) 아키텍처에 속하는 다음 모듈이 포함됩니다.

- 분류기
- 측정기
- 표시기

IPQoS는 Diffserv 모델에 다음 향상된 기능을 추가합니다.

- 플로우 계산 모듈
- 802.1D 데이터그램 표시기

이 절에서는 IPQoS에서 사용되는 Diffserv 모듈을 소개합니다. QoS 정책을 설정하려면 이러한 모듈, 이름 및 용도에 대해 알고 있어야 합니다. 각 모듈에 대한 자세한 내용은 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”을 참조하십시오.

분류기(ipgpc) 개요

Diffserv 모델에서 **분류기**는 네트워크 트래픽 플로우에서 패킷을 선택합니다. **트래픽 플로우**는 다음 IP 헤더 필드에서 동일한 정보를 가지는 패킷 그룹을 구성합니다.

- 소스 주소
- 대상 주소
- 소스 포트
- 대상 포트
- 프로토콜 번호

IPQoS에서 이러한 필드를 **5-튜플**이라고 합니다.

IPQoS 분류기 모듈의 이름은 **ipgpc**로 지정됩니다. **ipgpc** 분류기는 트래픽 플로우를 IPQoS 구성 파일에서 구성하는 특성을 기준으로 클래스로 분류합니다.

ipgpc에 대한 자세한 내용은 **86 페이지 “분류기 모듈”**을 참조하십시오.

IPQoS 클래스

클래스는 유사한 특성을 공유하는 네트워크 플로우의 그룹입니다. 예를 들어, ISP는 고객에게 제공하는 서로 다른 서비스 레벨을 나타내기 위해 클래스를 정의할 수 있습니다. ASP는 다양한 응용 프로그램에 서로 다른 서비스 레벨을 제공하는 SLA를 정의할 수 있습니다. ASP QoS 정책의 경우, 클래스에는 특정 대상 IP 주소로 향하는 FTP 트래픽이 포함될 수 있습니다. 회사의 외부 웹 사이트의 송신 트래픽도 클래스로 정의될 수 있습니다.

트래픽을 클래스로 그룹화하는 것은 QoS 정책 계획에서 큰 부분을 차지합니다. **ipqosconf** 유틸리티를 사용하여 클래스를 만드는 경우 실제로는 **ipgpc** 분류기를 구성하는 것입니다.

클래스를 정의하는 방법에 대한 자세한 내용은 **31 페이지 “QoS 정책에 대한 클래스 정의 방법”**을 참조하십시오.

IPQoS 필터

필터는 **선택기**라고 부르는 매개변수가 포함된 일련의 규칙입니다. 각 필터는 클래스를 가리켜야 합니다. IPQoS는 패킷을 각 필터의 선택기에 대해 일치시켜 패킷이 해당 필터의 클래스에 속하는지 여부를 결정합니다. IPQoS 5-튜플 및 기타 공통 매개변수 등의 다양한 선택기를 사용하여 패킷을 필터링할 수 있습니다.

- 소스 주소 및 대상 주소
- 소스 포트 및 대상 포트
- 프로토콜 번호
- 사용자 ID
- 프로젝트 ID
- 차별화 서비스 코드 포인트(DSCP)

- 인터페이스 인덱스

예를 들어, 단순 필터에는 값이 80인 대상 포트가 포함될 수 있습니다. 그런 다음 ipgpc 분류기는 대상 포트 80(HTTP)으로 향하는 모든 패킷을 선택하고 QoS 정책에서 정의된 대로 패킷을 처리합니다.

필터 만들기에 대한 자세한 내용은 34 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.

측정기(tokenmt 및 tswtclmt) 개요

Diffserv 모델에서 측정기는 클래스별 기준에서 트래픽 플로우의 전송 속도를 추적합니다. 측정기는 플로우의 실제 속도가 구성된 속도를 얼마나 준수하는지 평가하여 해당하는 결과를 결정합니다. 트래픽 플로우의 결과를 기준으로 측정기는 후속 작업을 선택합니다. 후속 작업에는 다른 작업으로 패킷 보내기 또는 추가 처리 없이 네트워크로 패킷 돌려보내기가 포함될 수 있습니다.

IPQoS 측정기는 네트워크 플로우가 QoS 정책에서 해당 클래스에 대해 정의된 전송 속도를 준수하는지 여부를 결정합니다. IPQoS에는 두 측정 모듈이 포함됩니다.

- tokenmt - 두 토큰 버킷 측정 체계를 사용합니다.
- tswtclmt - 시간별 창 측정 체계를 사용합니다.

두 측정 모듈은 모두 빨간색, 노란색 및 녹색의 세 가지 결과를 인식합니다.

red_action_name, yellow_action_name 및 green_action_name 매개변수에서 각 결과에 대해 수행할 작업을 정의합니다.

또한 tokenmt가 색상을 인식하도록 구성할 수 있습니다. 색상 인식 측정 인스턴스에서는 패킷의 크기, DSCP, 트래픽 속도 및 구성된 매개변수를 사용하여 결과를 결정합니다. 측정기는 DSCP를 사용하여 패킷의 결과를 녹색, 노란색 또는 빨간색으로 매핑합니다.

IPQoS 측정기의 매개변수 정의에 대한 자세한 내용은 35 페이지 “플로우 제어 계획 방법”을 참조하십시오.

표시기(dscpmk 및 dlcosmk) 개요

Diffserv 모델에서 표시는 패킷에 전달 동작을 반영하는 값을 표시합니다. 표시는 패킷의 헤더에 값을 두어 패킷을 네트워크에 어떻게 전달할지 나타내는 프로세스입니다.

IPQoS에는 두 표시기 모듈이 포함됩니다.

- **dscpmk** - IP 패킷 헤더의 DS 필드를 **차별화 서비스 코드 포인트** 또는 *DSCP*라는 숫자 값으로 표시합니다. 그러면 Diffserv 인식 라우터에서 DS 코드 포인트를 사용하여 알맞은 전달 동작을 패킷에 적용할 수 있습니다.
- **dLcosmk** - 이더넷 프레임 헤더의 VLAN(virtual local area network) 태그를 **사용자 우선 순위**라는 숫자 값으로 표시합니다. 사용자 우선 순위는 데이터그램에 적용할 알맞은 전달 동작을 정의하는 *CoS(서비스 클래스)*를 나타냅니다.

dLcosmk는 IETF에서 설계한 Diffserv 모델의 일부가 아닌 IPQoS 추가 기능입니다.

QoS 정책의 표시기 전략 구현에 대한 자세한 내용은 **38 페이지** “전달 동작 계획 방법”을 참조하십시오.

플로우 계산(flowacct) 개요

IPQoS는 flowacct 정산 모듈을 Diffserv 모델에 추가합니다. flowacct를 사용하여 트래픽 플로우에 대한 통계를 수집하고 해당 SLA에 따라 고객에게 청구할 수 있습니다. 플로우 계산은 용량 계획 및 시스템 모니터링에도 유용합니다.

flowacct 모듈은 acctadm 명령과 함께 작동하여 계산 로그 파일을 만듭니다. 기본 로그에는 다음 목록에 나온 대로 IPQoS 5-튜플 및 두 가지 추가 속성이 포함됩니다.

- 소스 주소
- 소스 포트
- 대상 주소
- 대상 포트
- 프로토콜 번호
- 패킷 수
- 바이트 수

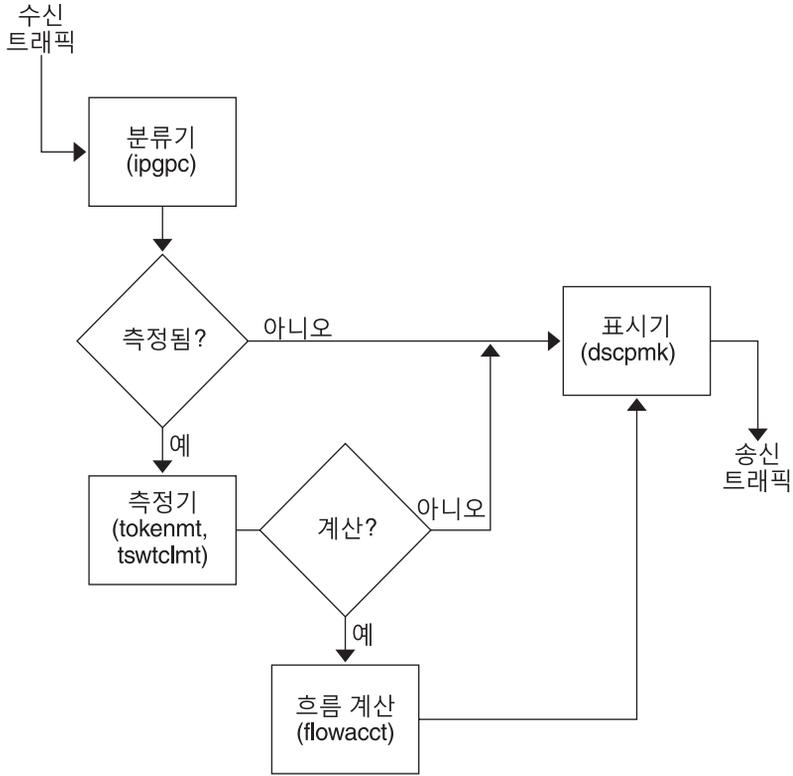
또한 **80 페이지** “트래픽 플로우에 대한 정보 기록”과 flowacct(7ipp) 및 acctadm(1M) 매뉴얼 페이지에 설명된 대로 다른 속성에 대한 통계도 수집할 수 있습니다.

플로우 계산 전략 계획에 대한 자세한 내용은 **40 페이지** “플로우 계산 계획 방법”을 참조하십시오.

IPQoS 모듈을 통한 트래픽 플로우 방식

다음 그림은 수신 트래픽이 몇 가지 IPQoS 모듈을 통과할 수 있는 경로를 보여줍니다.

그림 1-1 Diffserv 모델의 IPQoS 구현을 통한 트래픽 플로우



이 그림은 IPQoS 사용 시스템에서 일반적인 트래픽 플로우 시퀀스를 보여줍니다.

1. 분류기는 패킷 스트림에서 시스템 QoS 정책의 필터링 조건과 일치하는 모든 패킷을 선택합니다.
2. 그런 다음 선택된 패킷은 수행할 다음 작업에 대해 평가됩니다.
3. 분류기는 플로우 제어가 필요하지 않은 모든 트래픽을 표시기로 보냅니다.
4. 플로우 제어가 필요한 트래픽은 측정기로 보내집니다.
5. 측정기는 구성된 속도를 적용합니다. 그런 다음 측정기는 트래픽 준수 값을 플로우 제어 패킷에 지정합니다.
6. 그런 다음 플로우 제어 패킷은 평가되어 패킷에 계산이 필요한지 여부를 결정합니다.
7. 측정기는 플로우 계산이 필요하지 않은 모든 트래픽을 표시기로 보냅니다.
8. 플로우 계산 모듈은 수신된 패킷에 대한 통계를 수집합니다. 그런 다음 모듈은 패킷을 표시기로 보냅니다.
9. 표시기는 DS 코드 포인트를 패킷 헤더에 지정합니다. 이 DSCP는 Diffserv 인식 시스템에서 패킷에 적용해야 하는 홉별 동작을 나타냅니다.

IPQoS 사용 네트워크에서 트래픽 전달

이 절에서는 IPQoS 사용 네트워크에서 패킷 전달과 관련된 요소를 소개합니다. IPQoS 사용 시스템은 네트워크 스트림에서 시스템의 IP 주소를 대상으로 가지는 모든 패킷을 처리합니다. 그런 다음 IPQoS 시스템은 QoS 정책을 패킷에 적용하여 차별화 서비스를 설정합니다.

DS 코드 포인트

DSCP(DS 코드 포인트)는 패킷 헤더에서 Diffserv 인식 시스템이 표시된 패킷에 대해 수행해야 하는 작업을 정의합니다. Diffserv 아키텍처는 사용할 IPQoS 사용 시스템 및 Diffserv 라우터에 대한 DS 코드 포인트 세트를 정의합니다. 또한 Diffserv 아키텍처는 DSCP와 일치하는 **전달 동작**이라는 작업 세트를 정의합니다. IPQoS 사용 시스템은 패킷 헤더에서 DS 필드의 우선권 비트를 DSCP로 표시합니다. 라우터가 DSCP 값이 있는 패킷을 수신하면 라우터는 해당 DSCP와 연결된 전달 동작을 적용합니다. 그런 다음 패킷은 네트워크로 보내집니다.

주 - dlcosmk 표시기는 DSCP를 사용하지 않습니다. 대신 dlcosmk가 이더넷 프레임 헤더를 CoS 값으로 표시합니다. VLAN 장치를 사용하는 네트워크에서 IPQoS를 구성하려는 경우 [91 페이지](#) “표시기 모듈”을 참조하십시오.

홉별 동작

Diffserv 용어에서 DSCP에 지정된 전달 동작을 **PHB(홉별 동작)**이라고 합니다. PHB는 Diffserv 인식 시스템에서 다른 트래픽과 관련하여 표시된 패킷이 수신하는 전달 우선권을 정의합니다. 이 우선권은 최종적으로 IPQoS 사용 시스템이나 Diffserv 라우터가 표시된 패킷을 전달할지 또는 삭제할지 결정합니다. 전달되는 패킷의 경우, 대상으로 향하는 경로에서 패킷이 만나는 각 Diffserv 라우터는 동일한 PHB를 적용합니다. 다른 Diffserv 시스템이 DSCP를 변경할 경우는 예외입니다. PHB에 대한 자세한 내용은 [91 페이지](#) “패킷 전달을 위해 dscpmk 표시기 사용”을 참조하십시오.

PHB의 목적은 지정된 양의 네트워크 리소스를 인접 네트워크의 트래픽 클래스에 제공하는 것입니다. QoS 정책에서 이 목적을 달성할 수 있습니다. 트래픽 플로우가 IPQoS 사용 시스템을 떠날 때 트래픽 클래스에 대한 우선권 레벨을 나타내는 DSCP를 정의합니다. 우선권은 높은 우선권/낮은 삭제 가능성에서 낮은 우선권/높은 삭제 가능성의 범위에 있을 수 있습니다.

예를 들어, QoS 정책은 한 트래픽 클래스에 낮은 삭제 가능성의 PHB를 보장하는 DSCP를 지정할 수 있습니다. 그러면 이 트래픽 클래스는 Diffserv 인식 라우터에서 이 클래스의 패킷에 대역폭을 보장하는 낮은 삭제 우선권의 PHB를 수신합니다. 다양한 레벨의 우선권을 다른 트래픽 클래스에 지정하는 다른 DSCP를 QoS 정책에 추가할 수 있습니다. 낮은 우선권의 패킷에는 패킷의 DSCP에 표시된 우선 순위에 따라 Diffserv 시스템에서 대역폭을 제공합니다.

IPQoS는 Diffserv 아키텍처에서 정의된 빠른 전달 및 보장 전달의 두 가지 전달 동작 유형을 지원합니다.

빠른 전달

EF(빠른 전달) 홉별 동작은 EF 관련 DSCP를 가진 트래픽 클래스에 가장 높은 우선 순위가 부여되도록 합니다. EF DSCP를 가진 트래픽은 대기열에 두지 않습니다. EF는 낮은 손실, 대기 시간 및 지터를 제공합니다. EF에 권장되는 DSCP는 101110입니다. 101110으로 표시된 패킷은 대상으로 향하는 경로에서 Diffserv 인식 네트워크를 통과할 때 보장된 낮은 삭제 우선권을 받습니다. 프리미엄 SLA의 고객이나 응용 프로그램에 우선 순위를 지정할 때 EF DSCP를 사용합니다.

보장 전달

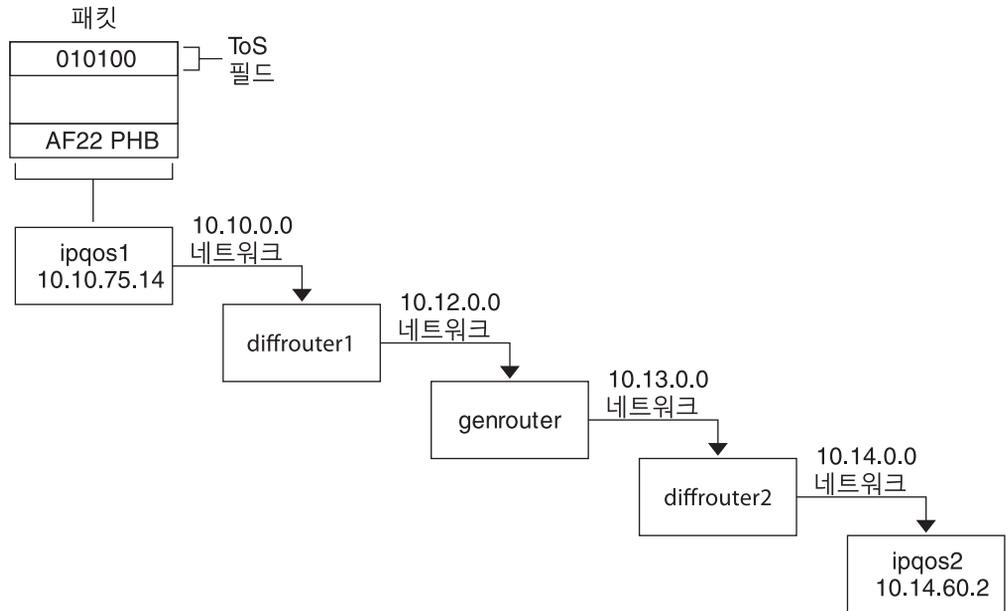
AF(보장 전달) 홉별 동작은 패킷에 지정할 수 있는 네 가지 서로 다른 전달 클래스를 제공합니다. 모든 전달 클래스는 표 6-2에 나온 대로 세 가지 삭제 우선권을 제공합니다.

다양한 AF 코드 포인트는 고객 및 응용 프로그램에 서로 다른 레벨의 서비스를 지정할 수 있는 기능을 제공합니다. QoS 정책에서는 QoS 정책을 계획할 때 네트워크에서 트래픽 및 서비스의 우선 순위를 지정할 수 있습니다. 그런 다음 서로 다른 AF 레벨을 우선 순위가 지정된 트래픽에 지정할 수 있습니다.

Diffserv 환경에서 패킷 전달

다음 그림은 부분적으로 Diffserv 사용 환경을 갖춘 회사 인트라넷의 일부를 보여줍니다. 이 시나리오에서 10.10.0.0 및 10.14.0.0 네트워크의 모든 호스트는 IPQoS가 사용으로 설정되어 있고, 두 네트워크의 로컬 라우터는 Diffserv를 인식합니다. 하지만 임시 네트워크는 Diffserv에 대해 구성되지 않았습니다.

그림 1-2 Diffserv 인식 네트워크 홉에서 패킷 전달



다음 단계에서는 이 그림에 표시된 패킷의 플로우를 추적합니다. 단계는 ipqos1 호스트에서 발생하는 패킷의 진행부터 시작됩니다. 그런 다음 단계는 여러 홉을 거쳐 ipqos2 호스트로 계속됩니다.

1. ipqos1의 사용자는 ftp 명령을 실행하여 세 홉 떨어진 ipqos2 호스트에 액세스합니다.
2. ipqos1은 QoS 정책을 결과 패킷 플로우에 적용합니다. 그런 다음 ipqos1은 ftp 트래픽을 성공적으로 분류합니다.

시스템 관리자는 로컬 네트워크 10.10.0.0에서 발생하는 모든 나가는 ftp 트래픽에 대한 클래스를 만들었습니다. ftp 클래스에 대한 트래픽에는 클래스 2인 중간 삭제 우선권의 AF22 홉별 동작이 지정되었습니다. ftp 클래스에 대해서는 2Mb/초의 트래픽 플로우 속도가 구성되었습니다.

3. ipqos-1은 ftp 플로우를 측정하여 플로우가 2Mb/초의 약정된 속도를 초과하는지 여부를 결정합니다.
4. ipqos1의 표시기는 나가는 ftp 패킷의 DS 필드를 AF22 PHB와 일치하는 010100 DSCP로 표시합니다.
5. diffrouter1 라우터는 ftp 패킷을 수신합니다. 그런 다음 diffrouter1은 DSCP를 검사합니다. diffrouter1가 정체된 경우 AF22로 표시된 패킷은 삭제됩니다.
6. ftp 트래픽은 diffrouter1의 파일에서 AF22에 대해 구성된 홉별 동작에 따라 다음 홉으로 전달됩니다.
7. ftp 트래픽은 10.12.0.0 네트워크를 통과하여 Diffserv를 인식하지 못하는 genrouter로 이동합니다. 결과적으로 트래픽은 “최선 조건” 전달 동작을 수신합니다.

8. genrouter는 ftp 트래픽을 10.13.0.0 네트워크에 전달합니다. 여기에서 트래픽은 diffrouter2로 수신됩니다.
9. diffrouter2는 Diffserv를 인식합니다. 따라서 라우터는 AF22 패킷에 대한 라우터 정책에서 정의된 PHB에 따라 ftp 패킷을 네트워크에 전달합니다.
10. ipqos2는 ftp 트래픽을 수신합니다. 그런 다음 ipqos2는 ipqos1의 사용자에게 사용자 이름과 암호를 물어봅니다.

IPQoS 사용 네트워크 계획(작업)

Oracle Solaris를 실행하는 모든 시스템에서 IPQoS를 구성할 수 있습니다. 그러면 IPQoS 시스템은 Diffserv 인식 라우터와 함께 작동하여 인터넷에서 차별화된 서비스와 트래픽 관리를 제공합니다.

이 장에는 Diffserv 인식 네트워크에 IPQoS 사용 시스템을 추가하는 계획 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

- 25 페이지 “일반 IPQoS 구성 계획(작업 맵)”
- 26 페이지 “Diffserv 네트워크 토폴로지 계획”
- 29 페이지 “서비스 품질 정책 계획”
- 30 페이지 “QoS 정책 계획(작업 맵)”
- 41 페이지 “IPQoS 구성 예 소개”

주 - IPQoS 기능은 이후 릴리스에서 제거될 수 있습니다. 비슷한 대역폭 리소스 제어 기능을 지원하는 `dladm`, `flowadm` 및 관련 명령을 대신 사용하는 것이 좋습니다. 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

일반 IPQoS 구성 계획(작업 맵)

네트워크에서 IPQoS를 비롯하여 차별화된 서비스를 구현하려면 광범위한 계획이 필요합니다. 각 IPQoS 사용 시스템의 위치 및 기능뿐 아니라 각 시스템과 로컬 네트워크에 있는 라우터의 관계도 고려해야 합니다. 다음 작업 맵에서는 네트워크에서 IPQoS를 구현하는 주요 계획 작업을 나열하고 작업을 완료하는 데 필요한 절차와 관련된 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS 사용 시스템을 통합하는 Diffserv 네트워크 토폴로지를 계획합니다.	다양한 Diffserv 네트워크 토폴로지를 조사하여 사이트에 가장 적합한 솔루션을 결정합니다.	26 페이지 “Diffserv 네트워크 토폴로지 계획”
2. IPQoS 시스템이 제공할 다양한 유형의 서비스를 계획합니다.	네트워크가 제공하는 서비스의 유형을 SLA(서비스 단계 계약)별로 구성합니다.	29 페이지 “서비스 품질 정책 계획”
3. 각 IPQoS 시스템에 대한 QoS 정책을 계획합니다.	각 SLA 구현에 필요한 클래스, 측정 및 계산 기능을 결정합니다.	29 페이지 “서비스 품질 정책 계획”
4. 해당하는 경우 Diffserv 라우터에 대한 정책을 계획합니다.	IPQoS 시스템에서 사용되는 Diffserv 라우터에 대한 일정 잡기 및 대기열 지정 정책을 결정합니다.	대기열 지정 및 일정 잡기 정책은 라우터 설명서를 참조하십시오.

Diffserv 네트워크 토폴로지 계획

네트워크에 대해 차별화된 서비스를 제공하려면 하나 이상의 IPQoS 사용 시스템 및 Diffserv 인식 라우터가 필요합니다. 이 절에 설명된 다양한 방법으로 이와 같은 기본 시나리오를 확장할 수 있습니다.

Diffserv 네트워크에 대한 하드웨어 전략

일반적으로 고객은 서버 및 서버 통합에서 IPQoS를 실행합니다. 반대로 네트워크 요구 사항에 따라 데스크탑 시스템에서도 IPQoS를 실행할 수 있습니다.

다음 목록에서는 IPQoS 구성이 가능한 시스템에 대해 설명합니다.

- 웹 서버, 데이터베이스 서버 등의 다양한 서비스를 제공하는 Oracle Solaris 시스템
- 전자 메일, FTP 또는 기타 많이 사용되는 네트워크 응용 프로그램을 제공하는 애플리케이션 서버
- 웹 캐시 서버 또는 프록시 서버
- Diffserv 인식 로드 밸런서가 관리하는 IPQoS 사용 서버 팜의 네트워크
- 단일 이기종 네트워크에 대한 트래픽을 관리하는 방화벽
- 가상 근거리 통신망(LAN)의 일부인 IPQoS 시스템

Diffserv 인식 라우터가 이미 작동되고 있는 네트워크 토폴로지에 IPQoS 시스템을 도입할 수 있습니다. 라우터가 현재 Diffserv를 제공하지 않을 경우 Cisco Systems, Juniper Networks 및 기타 라우터 제조업체에서 제공하는 Diffserv 솔루션을 고려해 보십시오. 로컬 라우터가 Diffserv를 구현하지 않은 경우 라우터는 표시를 평가하지 않은 상태로 표시된 패킷을 다음 홉으로 전달합니다.

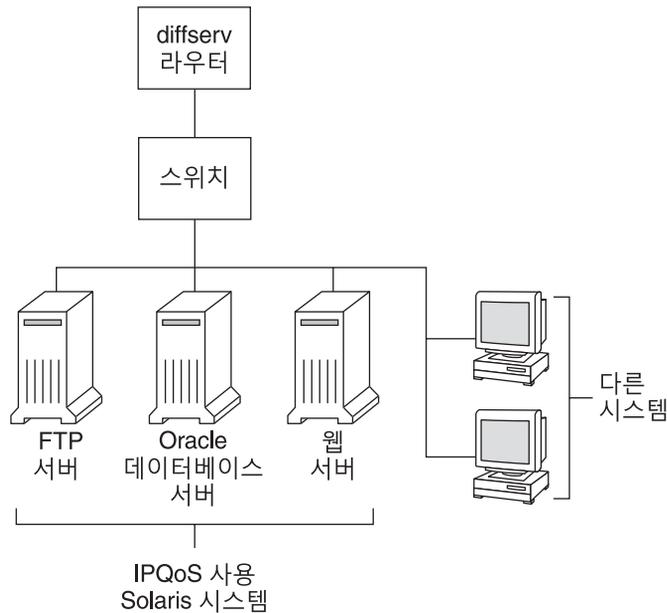
IPQoS 네트워크 토폴로지

이 단원에서는 여러 네트워크 요구에 대한 IPQoS 전략을 설명합니다.

개별 호스트의 IPQoS

다음 그림에서는 IPQoS 사용 시스템의 단일 네트워크를 보여 줍니다.

그림 2-1 네트워크 세그먼트의 IPQoS 시스템



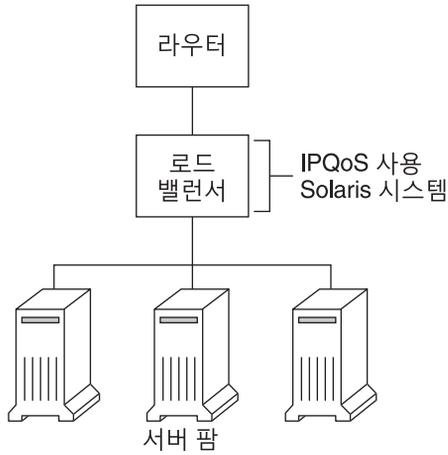
이 네트워크는 회사 인트라넷의 유일한 세그먼트입니다. 애플리케이션 서버 및 웹 서버에서 IPQoS를 사용으로 설정하면 각 IPQoS 시스템이 송신 트래픽을 릴리스하는 속도를 제어할 수 있습니다. 라우터가 Diffserv를 인식하도록 설정할 경우 추가로 수신 및 송신 트래픽을 제어할 수 있습니다.

본 설명서의 예에는 “개별 호스트의 IPQoS” 시나리오가 사용됩니다. 설명서 전체에서 사용되는 토폴로지 예는 [그림 2-4](#)를 참조하십시오.

서버 팜 네트워크의 IPQoS

다음 그림에서는 이기종 서버 팜이 여러 개인 네트워크를 보여 줍니다.

그림 2-2 IPQoS 사용 서버 팜의 네트워크



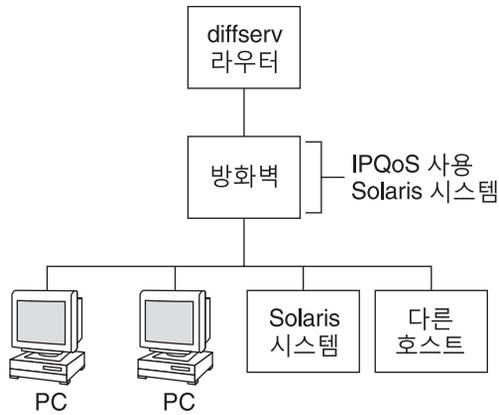
해당 토폴로지에서는 라우터가 Diffserv를 인식하므로 수신 트래픽과 송신 트래픽을 모두 대기열에 지정하고 속도를 제어할 수 있습니다. 로드 밸런서도 Diffserv를 인식하며 서버 팜에서 IPQoS가 사용됩니다. 로드 밸런서가 사용자 ID, 프로젝트 ID 등의 선택기를 사용하여 라우터 이외의 추가 필터링을 제공할 수 있습니다. 이러한 선택기는 응용 프로그램 데이터에 포함되어 있습니다.

이 시나리오는 로컬 네트워크의 혼잡을 관리할 수 있도록 플로우 제어 및 트래픽 전달을 제공합니다. 또한 이 시나리오는 서버 팜의 송신 트래픽으로 인해 인트라넷의 다른 부분이 과부화되지 않도록 합니다.

방화벽의 IPQoS

다음 그림에서는 방화벽을 통해 다른 세그먼트로부터 보호되는 회사 네트워크 세그먼트를 보여 줍니다.

그림 2-3 IPQoS 사용 방화벽으로 보호되는 네트워크



이 시나리오에서 트래픽은 패킷이 필터링되고 대기열에 지정되는 Diffserv 인식 라우터로 들어옵니다. 라우터가 전달한 모든 수신 트래픽은 IPQoS 사용 방화벽을 통과합니다. IPQoS를 사용하려면 방화벽이 IP 전달 스택을 무시하지 않아야 합니다.

방화벽의 보안 정책에 따라 수신 트래픽이 내부 네트워크로 들어오거나 나갈 수 있는지 여부가 결정됩니다. QoS 정책은 방화벽을 통과한 수신 트래픽의 서비스 레벨을 제어합니다. QoS 정책에 따라 송신 트래픽에 전달 동작을 표시할 수도 있습니다.

서비스 품질 정책 계획

서비스 품질(QoS) 정책을 계획할 때는 네트워크가 제공하는 서비스를 검토 및 분류하고 우선 순위를 설정해야 합니다. 또한 사용 가능한 대역폭을 평가하여 각 트래픽 클래스가 네트워크로 릴리스되는 속도를 결정해야 합니다.

QoS 정책 계획 지원

IPQoS 구성 파일에 필요한 정보를 포함하는 형식으로 QoS 정책을 계획하기 위한 정보를 수집합니다. 예를 들어, 다음 템플리트를 사용하여 IPQoS 구성 파일에서 사용될 주요 정보 범주를 나열할 수 있습니다.

표 2-1 QoS 계획 템플리트

클래스	우선 순위	필터	선택기	속도	전달 여부	계산 여부
클래스 1	1	필터 1 필터 3	선택기 1 선택기 2	측정기 유형에 따른 측정기 속도	표시자 삭제 우선 순위	플로우 계산 통계 필요

표 2-1 QoS 계획 템플리트 (계속)

클래스	우선순위	필터	선택기	속도	전달 여부	계산 여부
클래스 1	1	필터 2	선택기 1 선택기 2	해당 없음	해당 없음	해당 없음
클래스 2	2	필터 1	선택기 1 선택기 2	측정기 유형에 따른 측정기 속도	표시자 삭제 우선 순위	플로우 계산 통계 필요
클래스 2	2	필터 2	선택기 1 선택기 2	해당 없음	해당 없음	해당 없음

각 주요 범주를 구분하여 추가로 QoS 정책을 정의할 수 있습니다. 후속 절에서는 템플리트에 표시되는 범주에 대한 정보를 얻는 방법을 설명합니다.

QoS 정책 계획(작업 맵)

이 작업 맵에서는 QoS 정책 계획에 대한 주요 작업을 나열하고 각 작업에 대한 수행 지침과 관련된 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS를 지원하도록 네트워크 토폴로지를 설계합니다.	네트워크에서 차별화된 서비스를 제공할 호스트 및 라우터를 식별합니다.	31 페이지 “네트워크에서 IPQoS를 준비하는 방법”
2. 네트워크의 서비스를 구분해야 할 클래스를 정의합니다.	사이트에서 제공하는 서비스 유형 및 SLA를 확인하고 해당 서비스가 속한 고유한 트래픽 클래스를 결정합니다.	31 페이지 “QoS 정책에 대한 클래스 정의 방법”
3. 클래스에 대한 필터를 정의합니다.	특정 클래스의 트래픽과 네트워크 트래픽 플로우를 구분할 가장 적합한 방법을 결정합니다.	34 페이지 “QoS 정책에서 필터를 정의하는 방법”
4. 패킷이 IPQoS 시스템에서 나갈 때 트래픽을 측정할 플로우 제어 속도를 정의합니다.	각 트래픽 클래스에 대해 허용 가능한 플로우 속도를 결정합니다.	35 페이지 “플로우 제어 계획 방법”
5. QoS 정책에서 사용할 DSCP 또는 사용자 우선 순위 값을 정의합니다.	라우터 또는 스위치가 플로우를 처리할 때 트래픽 플로우에 지정되는 전달 동작을 결정할 체계를 계획합니다.	38 페이지 “전달 동작 계획 방법”
6. 해당하는 경우 네트워크의 트래픽 플로우에 대한 통계 모니터링 계획을 설정합니다.	트래픽 클래스를 평가하여 계산 또는 통계 용도로 모니터링해야 할 트래픽 플로우를 결정합니다.	40 페이지 “플로우 계산 계획 방법”

주 - 이 절의 나머지 부분에서는 IPQoS 사용 시스템의 QoS 정책을 계획하는 방법에 대해 설명합니다. Diffserv 라우터에 대한 QoS 정책을 계획하려면 라우터 설명서 및 라우터 제조업체 웹 사이트를 참조하십시오.

▼ 네트워크에서 IPQoS를 준비하는 방법

다음 절차에서는 QoS 정책을 만들기 전에 수행할 일반적인 계획 작업을 나열합니다.

- 1 **네트워크 토폴로지를 검토합니다. 그런 다음 IPQoS 시스템 및 Diffserv 라우터를 사용하는 전략을 계획합니다.**
토폴로지에는 26 페이지 “Diffserv 네트워크 토폴로지 계획”을 참조하십시오.
- 2 **토폴로지에서 IPQoS를 필요로 하거나 IPQoS 서비스로 사용 가능한 적합한 후보가 될 수 있는 호스트를 식별합니다.**
- 3 **동일한 QoS 정책을 사용할 수 있는 IPQoS 사용 시스템을 결정합니다.**
예를 들어, 네트워크의 모든 호스트에서 IPQoS를 사용으로 설정하려면 동일한 QoS 정책을 사용할 수 있는 호스트를 식별합니다. 각 IPQoS 사용 시스템에는 해당 IPQoS 구성 파일에서 구현되는 로컬 QoS 정책이 있어야 합니다. 하지만 특정 범위의 시스템에서 사용할 하나의 IPQoS 구성 파일을 만들 수 있습니다. 그런 다음 QoS 정책 요구 사항이 동일한 모든 시스템에 구성 파일을 복사할 수 있습니다.
- 4 **네트워크의 Diffserv 라우터에 필요한 계획 작업을 검토하고 수행합니다.**
자세한 내용은 라우터 설명서 및 라우터 제조업체 웹 사이트를 참조하십시오.

▼ QoS 정책에 대한 클래스 정의 방법

첫 번째 QoS 정책 정의 단계는 트래픽 플로우를 클래스로 구성하는 것입니다. Diffserv 네트워크에서 모든 유형의 트래픽에 대해 클래스를 만들 필요는 없습니다. 네트워크 토폴로지에 따라 각 IPQoS 사용 시스템에 대해 다른 QoS 정책을 만들어야 할 수도 있습니다.

주 - 클래스 개요는 17 페이지 “IPQoS 클래스”를 참조하십시오.

다음 절차에서는 31 페이지 “네트워크에서 IPQoS를 준비하는 방법”에 설명된 대로 IPQoS를 사용할 네트워크의 시스템을 결정했다고 가정합니다.

- 1 **QoS 정책 정보를 구성하는 데 필요한 QoS 계획 테이블을 만듭니다.**
제안 사항은 표 2-1을 참조하십시오.

2 네트워크에 있는 모든 QoS 정책에 대해 나머지 단계를 수행합니다.

3 QoS 정책에서 사용할 클래스를 정의합니다.

다음 질문은 가능한 클래스 정의를 위한 네트워크 트래픽 분석 지침입니다.

■ **회사에서 고객에게 서비스 단계 계약을 제공합니까?**

그럴 경우 회사에서 고객에게 제공하는 SLA의 상대적인 우선 순위 레벨을 평가합니다. 다른 우선 순위 레벨이 보장된 고객에게 동일한 응용 프로그램을 제공할 수 있습니다.

예를 들어, 회사에서 각 고객에게 웹 사이트 호스팅을 제공할 수 있습니다. 이 경우 각 고객 웹 사이트에 대한 클래스를 정의해야 합니다. 고급 웹 사이트를 하나의 서비스 레벨로 제공하는 SLA도 있을 수 있고, 할인 고객에게 "최상의" 개인 웹 사이트를 제공하는 SLA도 있을 수 있습니다. 이 요소는 다양한 웹 사이트를 클래스뿐만 아니라 웹 사이트 클래스에 지정되는 잠재적으로 다른 홉별 동작도 나타냅니다.

■ **IPQoS 시스템이 플로우 제어가 필요할 수 있는 많이 사용되는 응용 프로그램을 제공합니까?**

과도한 트래픽을 생성하는 많이 사용되는 응용 프로그램을 제공하는 서버에서 IPQoS를 사용으로 설정하여 네트워크 성능을 향상시킬 수 있습니다. 일반적인 예로 전자 메일, 네트워크 뉴스 및 FTP를 들 수 있습니다. 가능한 경우 서비스 유형별로 수신 및 송신 트래픽에 대해 별도의 클래스를 만드는 것이 좋습니다. 예를 들어, 메일 서버용 QoS 정책에 대해 mail-in 클래스와 mail-out 클래스를 만들 수 있습니다.

■ **네트워크에서 우선 순위가 가장 높은 전달 동작을 필요로 하는 특정 응용 프로그램이 실행됩니까?**

우선 순위가 가장 높은 전달 동작을 필요로 하는 중요한 응용 프로그램은 라우터 대기열에서 가장 높은 우선 순위를 받아야 합니다. 일반적인 예로 스트리밍 비디오 및 스트리밍 오디오를 들 수 있습니다.

이와 같이 우선 순위가 높은 응용 프로그램에 대해 수신 클래스와 송신 클래스를 정의합니다. 그런 다음 응용 프로그램을 제공하는 IPQoS 사용 시스템과 Diffserv 라우터의 QoS 정책에 클래스를 추가합니다.

■ **플로우에 대역폭이 많이 사용되어 네트워크에서 트래픽 플로우가 제어되어야 합니까?**

netstat, snoop 및 기타 네트워크 모니터링 유틸리티를 사용하여 네트워크에 문제를 일으키고 있는 트래픽의 유형을 검색할 수 있습니다. 지금까지 만든 클래스를 검토한 다음 정의되지 않은 문제 트래픽 범주에 대해 새 클래스를 만듭니다. 문제 트래픽 범주에 대한 클래스를 이미 정의한 경우 문제 트래픽을 제어할 측정기의 속도를 정의합니다.

네트워크에 있는 모든 IPQoS 사용 시스템의 문제 트래픽에 대한 클래스를 만듭니다. 그러면 각 IPQoS 시스템이 트래픽 플로우를 네트워크로 릴리스하는 속도를 제한하여 문제 트래픽을 처리할 수 있습니다. 또한 Diffserv 라우터에서 QoS 정책에 해당 문제 클래스를 정의해야 합니다. 그러면 라우터가 QoS 정책에 구성된 대로 문제 플로우를 대기열에 지정하고 일정을 잡을 수 있습니다.

■ 특정 유형의 트래픽에 대한 통계를 얻어야 합니까?

간단한 SLA 검토를 통해 계산해야 할 고객 트래픽의 유형을 확인할 수 있습니다. 사이트에서 SLA를 제공하는 경우 계산해야 할 트래픽에 대한 클래스가 이미 만들어진 상태일 것입니다. 모니터링하고 있는 트래픽 플로우에 대한 통계 수집을 사용하여 설정할 클래스를 정의할 수도 있습니다. 또한 보안상 액세스를 제한할 트래픽에 대한 클래스를 만들 수 있습니다.

4 1단계에서 만든 QoS 계획 테이블에서 정의한 클래스를 나열합니다.

5 각 클래스에 우선 순위 레벨을 지정합니다.

예를 들어, 우선 순위 레벨 1이 가장 높은 우선 순위의 클래스를 나타내도록 지정하고 나머지 클래스에 우선 순위를 내림차순으로 지정합니다. 지정한 우선 순위 레벨은 구조적인 용도로만 사용됩니다. QoS 정책 템플릿에서 설정한 우선 순위 레벨은 IPQoS에 실제로 사용되지 않습니다. QoS 정책에 적합한 경우 두 개 이상의 클래스에 동일한 우선 순위를 지정할 수도 있습니다.

6 클래스 정의가 완료되면 34 페이지 “QoS 정책에서 필터를 정의하는 방법”에 설명된 대로 각 클래스에 대한 필터를 정의합니다.

자세한 정보 클래스 우선 순위 설정

클래스를 만들면 우선 순위가 가장 높은 클래스, 우선 순위가 중간인 클래스, 우선 순위가 최상인 클래스를 빠르게 파악할 수 있습니다. 적합한 클래스 우선 순위 설정 체계는 38 페이지 “전달 동작 계획 방법”에 설명된 대로 송신 트래픽에 흡별 동작을 지정할 때 특히 중요합니다.

클래스에 PHB를 지정하는 것 외에 클래스에 대한 필터에 우선 순위 선택기를 정의할 수도 있습니다. 우선 순위 선택기는 IPQoS 사용 호스트에서만 활성화됩니다. 속도와 DSCP가 동일한 여러 클래스가 IPQoS 시스템에서 나갈 때 대역폭 경합이 발생하는 경우가 있다고 가정합니다. 이 경우 각 클래스의 우선 순위 선택기가 동일한 값의 클래스에 지정된 서비스 레벨의 순서를 추가로 지정할 수 있습니다.

필터 정의

특정 클래스의 멤버로 패킷 플로우를 식별하기 위한 필터를 만듭니다. 각 필터에는 패킷 플로우 평가 기준을 정의하는 선택기가 포함되어 있습니다. 그러면 IPQoS 사용 시스템이 선택기의 기준을 사용하여 트래픽 플로우에서 패킷을 추출합니다. 그런 다음 IPQoS 시스템이 패킷을 클래스와 연관시킵니다. 필터 소개는 17 페이지 “IPQoS 필터”를 참조하십시오.

다음 표에서는 가장 일반적으로 사용되는 선택기를 나열합니다. 처음 다섯 개의 선택기는 IPQoS 시스템이 패킷을 플로우 구성원으로 식별하는 데 사용하는 IPQoS 5 튜플을 나타냅니다. 전체 선택기 목록은 표 6-1을 참조하십시오.

표 2-2 일반적인 IPQoS 선택기

이름	정의
saddr	소스 주소입니다.
daddr	대상 주소입니다.
sport	소스 포트 번호입니다. /etc/services에 정의된 잘 알려진 포트 번호 또는 사용자 정의 포트 번호를 사용할 수 있습니다.
dport	대상 포트 번호입니다.
protocol	/etc/protocols의 트래픽 플로우 유형에 지정된 IP 프로토콜 번호 또는 프로토콜 이름입니다.
ip_version	사용할 주소 지정 스타일입니다. IPv4 또는 IPv6을 사용하십시오. IPv4가 기본값입니다.
dsfield	DS 필드 내용, 즉 DSCP입니다. 이미 특정 DSCP가 표시된 수신 패킷을 추출하려면 이 선택기를 사용하십시오.
priority	클래스에 지정된 우선 순위 레벨입니다. 자세한 내용은 31 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
user	상위 레벨 응용 프로그램이 실행될 때 사용되는 UNIX 사용자 ID 또는 사용자 이름입니다.
projid	상위 레벨 응용 프로그램이 실행될 때 사용되는 프로젝트 ID입니다.
direction	트래픽 플로우 방향입니다. 값은 LOCAL_IN, LOCAL_OUT, FWD_IN 또는 FWD_OUT입니다.

주 - 선택기를 선택할 때는 신중하십시오. 클래스에 대한 패킷을 추출하는 데 필요한 만큼만 선택기를 사용하십시오. 선택기를 많이 정의할수록 IPQoS 성능에 끼치는 영향이 커집니다.

▼ QoS 정책에서 필터를 정의하는 방법

시작하기 전에 다음 단계를 수행하려면 31 페이지 “QoS 정책에 대한 클래스 정의 방법” 절차를 완료해야 합니다.

- 1 31 페이지 “QoS 정책에 대한 클래스 정의 방법”에서 만든 QoS 계획 테이블에 각 클래스에 대한 필터를 하나 이상 만듭니다.

가능한 경우 클래스별로 수신 및 송신 트래픽에 대해 별도의 필터를 만드는 것이 좋습니다. 예를 들어, IPQoS 사용 FTP 서버의 QoS 정책에 ftp-in 필터 및 ftp-out 필터를 추가합니다. 그런 다음 기본 선택기 외에 적합한 direction 선택기도 정의할 수 있습니다.

- 2 클래스의 각 필터에 대한 선택기를 하나 이상 정의합니다.

표 2-1에서 소개된 QoS 계획 테이블을 사용하여 정의한 클래스에 대한 필터를 채웁니다.

예 2-1 FTP 트래픽에 대한 필터 정의

다음 표는 송신 FTP 트래픽에 대한 필터를 정의하는 방법을 보여주는 예입니다.

클래스	우선 순위	필터	선택기
ftp-traffic	4	ftp-out	saddr 10.190.17.44 daddr 10.100.10.53 sport 21 direction LOCAL_OUT

- 참조
- 플로우 제어 체계를 정의하려면 35 페이지 “플로우 제어 계획 방법”을 참조하십시오.
 - 플로우가 네트워크 스트림으로 반환될 때의 플로우에 대한 전달 동작을 정의하려면 38 페이지 “전달 동작 계획 방법”을 참조하십시오.
 - 특정 유형의 트래픽에 대한 플로우 계산을 계획하려면 40 페이지 “플로우 계산 계획 방법”을 참조하십시오.
 - QoS 정책에 다른 클래스를 추가하려면 31 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
 - QoS 정책에 다른 필터를 추가하려면 34 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.

▼ 플로우 제어 계획 방법

플로우 제어 과정에서는 클래스에 대한 트래픽 플로우가 측정되고 정의된 속도로 패킷이 네트워크로 릴리스됩니다. 플로우 제어를 계획할 때 IPQoS 측정 모듈에 사용할 매개변수를 정의합니다. 측정기는 트래픽이 네트워크로 릴리스되는 속도를 결정합니다. 측정 모듈 소개는 18 페이지 “측정기(tokenmt 및 tswtclmt) 개요”를 참조하십시오.

다음 절차에서는 34 페이지 “QoS 정책에서 필터를 정의하는 방법”에 설명된 대로 필터 및 선택기를 정의했다고 가정합니다.

- 1 네트워크에 대한 최대 대역폭을 확인합니다.
- 2 네트워크에서 지원되는 SLA를 검토합니다. 고객 및 각 고객에게 보장되는 서비스의 유형을 식별합니다.
특정 레벨의 서비스를 보장하려면 고객이 생성한 특정 트래픽 클래스를 측정해야 할 수도 있습니다.
- 3 31 페이지 “QoS 정책에 대한 클래스 정의 방법”에서 만든 클래스 목록을 검토합니다. SLA와 연관된 클래스 이외의 다른 클래스를 측정해야 할지 여부를 결정합니다.

IPQoS 시스템이 높은 레벨의 트래픽을 생성하는 응용 프로그램을 실행한다고 가정합니다. 응용 프로그램의 트래픽을 분류한 후 플로우를 측정하여 플로우의 패킷이 네트워크로 반환되는 속도를 제어합니다.

주 - 모든 클래스를 측정해야 하는 것은 아닙니다. 클래스 목록을 검토할 때 이 지침을 염두에 두십시오.

4 플로우 제어가 필요한 트래픽을 선택하는 각 클래스의 필터를 결정합니다. 그런 다음 측정이 필요한 클래스 목록을 세분화합니다.

필터가 두 개 이상인 클래스의 경우 하나의 필터에 대해서만 측정해야 합니다. 특정 클래스의 수신 및 송신 트래픽에 대한 필터를 정의한 것으로 가정합니다. 한 방향의 트래픽만 플로우 제어가 필요한 것으로 결론지을 수 있습니다.

5 플로우를 제어할 각 클래스에 대한 측정기 모듈을 선택합니다.

QoS 계획 테이블의 측정기 열에 모듈 이름을 추가합니다.

6 구조적 테이블에 측정할 각 클래스에 대한 속도를 추가합니다.

tokenmt 모듈을 사용하는 경우 다음 속도(비트/초)를 정의해야 합니다.

- 커밋 속도
- 최고 속도

이러한 속도가 특정 클래스를 측정하기에 충분할 경우 tokenmt에 대한 커밋 속도 및 커밋 버스트만 정의할 수 있습니다.

필요한 경우 다음 속도도 정의할 수 있습니다.

- 커밋 버스트
- 최고 버스트

tokenmt 속도에 대한 전체 정의는 89 페이지 “두 속도 측정기로 tokenmt 구성”을 참조하십시오. tokenmt(7ipp) 매뉴얼 페이지에서도 자세한 내용을 확인할 수 있습니다.

tswtclmt 모듈을 사용할 경우 다음 속도(비트/초)를 정의해야 합니다.

- 커밋 속도
- 최고 속도

창 크기(밀리초)도 정의할 수 있습니다. 이러한 속도는 90 페이지 “tswtclmt 측정 모듈” 및 twstclmt(7ipp) 매뉴얼 페이지에 정의되어 있습니다.

7 측정된 트래픽에 대한 트래픽 준수 결과를 추가합니다.

두 측정 모듈의 결과는 녹색, 빨간색 및 노란색입니다. 정의한 속도에 적용되는 트래픽 준수 결과를 QoS 구조적 테이블에 추가합니다. 측정기 결과는 88 페이지 “측정기 모듈”에서 자세히 설명됩니다.

커밋 속도를 준수하는 트래픽 또는 준수하지 않는 트래픽에 대해 수행해야 할 작업을 결정해야 합니다. 항상은 아니지만 이 작업은 패킷 헤더에 해당 동작을 표시하는 경우가 많습니다. 트래픽 플로우가 커밋 속도를 초과하지 않는 상태에서 녹색 레벨의 트래픽에 대해 허용 가능한 작업 중 하나는 처리를 계속하는 것일 수 있습니다. 플로우가 최고 속도를 초과할 경우 클래스의 패킷을 삭제하는 작업을 수행할 수도 있습니다.

예 2-2 측정기 정의

다음 표는 전자 메일 트래픽의 클래스에 대한 측정기 항목을 보여 주는 예입니다. IPQoS 시스템이 있는 네트워크의 총 대역폭은 100메가비트/초 또는 10000000비트/초입니다. QoS 정책은 전자 메일 클래스에 낮은 우선 순위를 지정합니다. 또한 이 클래스는 최상의 전달 동작을 수신합니다.

클래스	우선순위	필터	선택기	속도
email	8	mail_in	daddr10.50.50.5 dport imap direction LOCAL_IN	
email	8	mail_out	saddr10.50.50.5 sport imap direction LOCAL_OUT	meter=tokenmt 커밋 속도=5000000 커밋 버스트=5000000 최고 속도=10000000 최고 버스트=1000000 녹색 우선 순위=처리 계속 노란색 우선 순위=노란색 PHB 표시 빨간색 우선 순위=삭제

- 참조
- 패킷이 네트워크 스트림으로 반환될 때의 플로우에 대한 전달 동작을 정의하려면 38 페이지 “전달 동작 계획 방법”을 참조하십시오.
 - 특정 유형의 트래픽에 대한 플로우 계산을 계획하려면 40 페이지 “플로우 계산 계획 방법”을 참조하십시오.
 - QoS 정책에 다른 클래스를 추가하려면 31 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
 - QoS 정책에 다른 필터를 추가하려면 34 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.

- 다른 플로우 제어 체계를 정의하려면 35 페이지 “플로우 제어 계획 방법”을 참조하십시오.
- IPQoS 구성 파일을 만들려면 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

▼ 전달 동작 계획 방법

전달 동작에 따라 네트워크로 전달될 트래픽 플로우의 우선 순위 및 삭제 우선 순위가 결정됩니다. 두 가지 주요 전달 동작(다른 트래픽 클래스와 관계가 있는 클래스의 플로우 우선 순위 설정 또는 전체 플로우 삭제)을 선택할 수 있습니다.

Diffserv 모델은 표시자를 사용하여 선택된 전달 동작을 트래픽 플로우에 지정합니다. IPQoS는 다음 표시자 모듈을 제공합니다.

- `dscpmk` - IP 패킷의 DS 필드에 DSCP를 표시하는 데 사용됩니다.
- `dlcosmk` - 데이터그램의 VLAN 태그에 CoS(class-of-service) 값을 표시하는 데 사용됩니다.

주 - 이 절의 제안 사항은 IP 패킷에 해당하는 것입니다. IPQoS 시스템에 VLAN 장치가 포함된 경우 `dlcosmk` 표시자를 사용하여 데이터그램에 대한 전달 동작을 표시할 수 있습니다. 자세한 내용은 93 페이지 “VLAN 장치에서 `dlcosmk` 표시기 사용”을 참조하십시오.

IP 트래픽의 우선 순위를 설정하려면 각 패킷에 DSCP를 지정해야 합니다. `dscpmk` 표시자는 패킷의 DS 필드에 DSCP를 표시합니다. 전달 동작 유형과 연관된 잘 알려진 코드점 그룹에서 클래스에 대한 DSCP를 선택합니다. 이러한 잘 알려진 코드점은 EF PHB의 경우 46(101110)이며 AF PHB의 경우 코드점 범위입니다. DSCP 및 전달에 대한 개요 정보는 21 페이지 “IPQoS 사용 네트워크에서 트래픽 전달”을 참조하십시오.

시작하기 전에 다음 단계에서는 QoS 정책에 대한 클래스 및 필터를 정의했다고 가정합니다. 측정기와 표시자를 함께 사용하여 트래픽을 제어하는 경우가 많기는 하지만 표시자만으로도 전달 동작을 정의할 수 있습니다.

- 1 **지금까지 만든 클래스 및 각 클래스에 지정한 우선 순위를 검토합니다.**
모든 트래픽 클래스를 표시해야 하는 것은 아닙니다.
- 2 **우선 순위가 가장 높은 클래스에 EF 응답 동작을 지정합니다.**
EF PHB는 EF DSCP 46(101110)이 지정된 패킷이 AF PHB가 지정된 패킷보다 먼저 네트워크에 릴리스되도록 합니다. 우선 순위가 가장 높은 트래픽에 EF PHB를 사용합니다. EF에 대한 자세한 내용은 91 페이지 “EF(빠른 전달) PHB”를 참조하십시오.
- 3 **측정할 트래픽이 있는 클래스에 전달 동작을 지정합니다.**

4 클래스에 지정한 우선 순위에 따라 나머지 클래스에 DS 코드점을 지정합니다.

예 2-3 게임 응용 프로그램에 대한 QoS 정책

일반적으로 트래픽은 다음 이유로 측정됩니다.

- 네트워크 사용량이 많을 때 SLA가 이 클래스의 패킷에 대한 서비스 레벨을 보장합니다.
- 우선 순위가 보다 낮은 클래스가 네트워크의 혼잡을 야기할 수 있습니다.

표시자와 측정기를 함께 사용하여 이러한 클래스에 차별화된 서비스 및 대역폭 관리를 제공합니다. 예를 들어, 다음 표에서는 QoS 정책의 일부를 보여 줍니다. 이 정책은 높은 레벨의 트래픽을 생성하는 많이 사용되는 게임 응용 프로그램에 대한 클래스를 정의합니다.

클래스	우선 순위	필터	선택기	속도	전달 여부
games_app	9	games_in	sport 6080	해당 없음	해당 없음
games_app	9	games_out	dport 6081	meter=tokenmt 커밋 속도=5000000 커밋 버스트=5000000 최고 속도=10000000 최고 버스트=15000000 녹색 우선 순위=처리 계속 노란색 우선 순위=노란색 PHB 표시 빨간색 우선 순위=삭제	녹색=AF31 노란색=AF42 빨간색=drop

전달 동작은 커밋 속도를 준수하거나 최고 속도에 미치지 않는 games_app 트래픽에 우선 순위가 낮은 DSCP를 지정합니다. games_app 트래픽이 최고 속도를 초과하면 QoS 정책은 games_app의 패킷이 삭제되도록 합니다. 모든 AF 코드점은 표 6-2에서 나열됩니다.

- 참조
- 특정 유형의 트래픽에 대한 플로우 계산을 계획하려면 40 페이지 “플로우 계산 계획 방법”을 참조하십시오.
 - QoS 정책에 다른 클래스를 추가하려면 31 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.

- QoS 정책에 다른 필터를 추가하려면 34 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.
- 플로우 제어 체계를 정의하려면 35 페이지 “플로우 제어 계획 방법”을 참조하십시오.
- 패킷이 네트워크 스트림으로 반환될 때의 플로우에 대한 추가 전달 동작을 정의하려면 38 페이지 “전달 동작 계획 방법”을 참조하십시오.
- IPQoS 구성 파일을 만들려면 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

▼ 플로우 계산 계획 방법

IPQoS flowacct 모듈을 사용하여 청구 또는 네트워크 관리 용도로 트래픽 플로우를 추적할 수 있습니다. 다음 절차에 따라 QoS 정책에 플로우 계산이 포함되어야 할지 여부를 결정하십시오.

1 회사에서 고객에게 SLA를 제공합니까?

그럴 경우 플로우 계산을 사용해야 합니다. SLA를 검토하여 회사에서 고객에게 청구할 네트워크 트래픽의 유형을 결정합니다. 그런 다음 QoS 정책을 검토하여 청구할 트래픽을 선택하는 클래스를 결정합니다.

2 네트워크 문제가 발생하지 않도록 모니터링하거나 테스트해야 할 응용 프로그램이 있습니까?

있을 경우 플로우 계산을 사용하여 이러한 응용 프로그램의 동작을 관찰하는 것이 좋습니다. QoS 정책을 검토하여 모니터링해야 할 트래픽에 지정한 클래스를 확인합니다.

3 QoS 계획 테이블에서 플로우 계산이 필요한 각 클래스에 대해 플로우 계산 열에 Y를 표시합니다.

- 참조
- QoS 정책에 다른 클래스를 추가하려면 31 페이지 “QoS 정책에 대한 클래스 정의 방법”을 참조하십시오.
 - QoS 정책에 다른 필터를 추가하려면 34 페이지 “QoS 정책에서 필터를 정의하는 방법”을 참조하십시오.
 - 플로우 제어 체계를 정의하려면 35 페이지 “플로우 제어 계획 방법”을 참조하십시오.
 - 패킷이 네트워크 스트림으로 반환될 때의 플로우에 대한 전달 동작을 정의하려면 38 페이지 “전달 동작 계획 방법”을 참조하십시오.
 - 특정 유형의 트래픽에 대한 추가 플로우 계산을 계획하려면 40 페이지 “플로우 계산 계획 방법”을 참조하십시오.
 - IPQoS 구성 파일을 만들려면 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

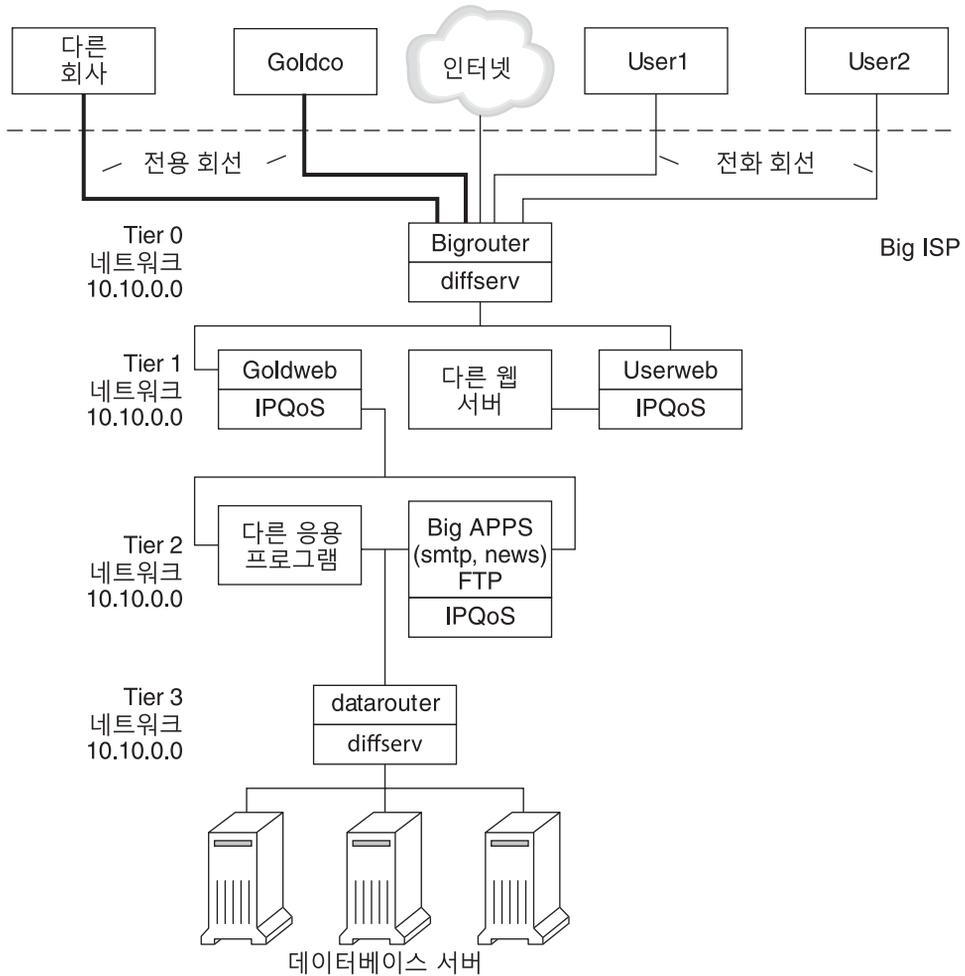
IPQoS 구성 예 소개

본 설명서의 나머지 장에 나오는 작업에서는 이 절에 소개된 IPQoS 구성 예를 사용합니다. 예에서는 가상 서비스 제공업체인 BigISP의 공용 인터넷에 있는 차별화된 서비스 솔루션을 보여 줍니다. BigISP는 전용 회선을 통해 BigISP에 연결하는 대규모 회사에 서비스를 제공합니다. 모뎀을 통한 전화 접속을 사용하는 개인도 BigISP에서 서비스를 구입할 수 있습니다.

IPQoS 토폴로지

다음 그림에서는 BigISP의 공용 인터넷에 사용되는 네트워크 토폴로지를 보여 줍니다.

그림 2-4 IPQoS 예 토폴로지



BigISP는 공용 인터넷에서 다음 네 계층을 구현했습니다.

- **Tier 0 - 10.10.0.0** 네트워크에는 외부 인터페이스와 내부 인터페이스가 모두 있는 Bigrouter라는 큰 Diffserv 라우터가 있습니다. Goldco라는 대규모 조직을 비롯하여 여러 회사가 Bigrouter에서 종료되는 전용 회선 서비스를 임대했습니다. Tier 0은 전화 회선 또는 ISDN을 통해 연결하는 개인 고객도 처리합니다.
- **Tier 1 - 10.11.0.0** 네트워크는 웹 서비스를 제공합니다. Goldweb 서버는 Goldco가 BigISP로부터 구매한 고급 서비스에 포함된 웹 사이트를 호스팅합니다. Userweb 서버는 개인 고객이 구매한 작은 웹 사이트를 호스팅합니다. Goldweb과 Userweb에는 모두 IPQoS가 사용됩니다.

- **Tier 2 - 10.12.0.0** 네트워크는 모든 고객에게 사용할 응용 프로그램을 제공합니다. 애플리케이션 서버 중 하나인 BigAPPS에는 IPQoS가 사용됩니다. BigAPPS는 SMTP, 뉴스 및 FTP 서비스를 제공합니다.
- **Tier 3 - 10.13.0.0** 네트워크는 큰 데이터베이스 서버를 다룹니다. Tier 3에 대한 액세스는 Diffserv 라우터인 datarouter를 통해 제어됩니다.

IPQoS 구성 파일 만들기(작업)

이 장에서는 IPQoS 구성 파일을 만드는 방법을 설명합니다. 이 장에서는 다음 항목을 다룹니다.

- 45 페이지 “IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)”
- 47 페이지 “QoS 정책을 만들기 위한 도구”
- 47 페이지 “웹 서버에 대한 IPQoS 구성 파일 만들기”
- 60 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일 만들기”
- 69 페이지 “라우터에서 차별화 서비스 제공”

이 장에서는 완전한 QoS 정책이 정의되어 있고, 이 정책을 IPQoS 구성 파일에 대한 기준으로 사용할 준비가 되어 있다고 가정합니다. QoS 정책 계획에 대한 자세한 내용은 29 페이지 “서비스 품질 정책 계획”을 참조하십시오.

주 - IPQoS 기능은 이후 릴리스에서 제거될 수 있습니다. 비슷한 대역폭 리소스 제어 기능을 지원하는 `dladm`, `flowadm` 및 관련 명령을 대신 사용하는 것이 좋습니다. 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)

이 작업 맵에서는 IPQoS 구성 파일을 만들기 위한 일반적인 작업을 나열하고 작업 수행 단계를 설명하는 각 절에 대한 링크를 제공합니다.

작업	설명	수행 방법
1. IPQoS 사용 네트워크 구성을 계획합니다.	로컬 시스템에서 IPQoS 사용 시스템이 되어야 하는 시스템을 결정합니다.	31 페이지 “네트워크에서 IPQoS를 준비하는 방법”

작업	설명	수행 방법
2. 네트워크에서 IPQoS 시스템에 대한 QoS 정책을 계획합니다.	트래픽 플로우를 고유의 서비스 클래스로 식별합니다.그런 다음 트래픽 관리가 필요한 플로우를 결정합니다.	29 페이지 “서비스 품질 정책 계획”
3. IPQoS 구성 파일을 만들고 첫번째 작업을 정의합니다.	IPQoS 파일을 만들고 IP 분류기를 호출한 다음 처리할 클래스를 정의합니다.	49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”
4. 클래스에 대한 필터를 만듭니다.	어떤 클래스가 선택되고, 클래스로 구성되는지 제어하는 필터를 추가합니다.	52 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”
5. 더 많은 클래스와 필터를 IPQoS 구성 파일에 추가합니다.	IP 분류기로 처리할 더 많은 클래스와 필터를 만듭니다.	57 페이지 “최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법”
6. 측정 모듈을 구성하는 매개변수와 함께 action 명령문을 추가합니다.	QoS 정책에서 플로우 제어를 요구하는 경우 플로우 제어 속도 및 준수 레벨을 측정기에 지정합니다.	66 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”
7. 표시기를 구성하는 매개변수와 함께 action 명령문을 추가합니다.	QoS 정책에서 차별화된 전달 동작을 요구하는 경우 트래픽 클래스가 전달되는 방식을 정의합니다.	53 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”
8. 플로우 계산 모듈을 구성하는 매개변수와 함께 action 명령문을 추가합니다.	QoS 정책에서 트래픽 플로우에 대한 통계 수집을 요구하는 경우 계산 통계가 수집되는 방식을 정의합니다.	56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”
9. IPQoS 구성 파일을 적용합니다.	지정된 IPQoS 구성 파일의 내용을 해당하는 커널 모듈에 추가합니다.	72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”
10. 라우터 파일에서 전달 동작을 구성합니다.	네트워크의 IPQoS 구성 파일에서 전달 동작을 정의하는 경우 결과 DSCP를 라우터의 해당하는 일정 파일에 추가합니다.	69 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”

QoS 정책을 만들기 위한 도구

네트워크에 대한 QoS 정책은 IPQoS 구성 파일에 상주합니다. 이 구성 파일은 텍스트 편집기를 사용하여 만듭니다. 그런 다음 파일을 `ipqosconf`(IPQoS 구성 유틸리티)에 인수로 제공합니다. `ipqosconf`가 구성 파일에서 정의된 정책을 적용하도록 지시하면 정책이 커널 IPQoS 시스템에 쓰여집니다. `ipqosconf` 명령에 대한 자세한 내용은 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오. `ipqosconf` 사용에 대한 자세한 내용은 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.

기본 IPQoS 구성 파일

IPQoS 구성 파일은 29 페이지 “서비스 품질 정책 계획”에서 정의한 QoS 정책을 구현하는 `action` 명령문 트리로 구성됩니다. IPQoS 구성 파일은 IPQoS 모듈을 구성합니다. 각 작업 명령문에는 작업 명령문에서 호출된 모듈로 처리될 클래스, 필터 또는 매개변수 세트가 포함됩니다.

IPQoS 구성 파일의 전체 구문은 예 6-3 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

IPQoS 예제 토폴로지 구성

이 장의 작업에서는 세 IPQoS 사용 시스템에 대한 IPQoS 구성 파일을 만드는 방법을 설명합니다. 이러한 시스템은 그림 2-4에 소개된 BigISP 회사의 네트워크 토폴로지에 속합니다.

- **Goldweb** - 프리미엄 레벨 SLA를 구매한 고객을 위한 웹 사이트를 호스트하는 웹 서버입니다.
- **Userweb** - “최선 조건” SLA를 구매한 가정 사용자를 위한 개인용 웹 사이트를 호스트하는 덜 강력한 웹 서버입니다.
- **BigAPPS** - 골드 레벨 및 최선 조건 고객을 위한 메일, 네트워크 뉴스 및 FTP를 서비스하는 애플리케이션 서버입니다.

이러한 세 구성 파일은 가장 일반적인 IPQoS 구성을 보여줍니다. 다음 절에 나오는 샘플 파일을 고유의 IPQoS 구현을 위한 템플릿으로 사용할 수 있습니다.

웹 서버에 대한 IPQoS 구성 파일 만들기

이 절에서는 프리미엄 웹 서버에 대한 구성을 만드는 방법을 통해 IPQoS 구성 파일을 소개합니다. 그런 다음 개인용 웹 사이트를 호스트하는 서버에 대한 다른 구성 파일에서 완전히 다른 레벨의 서비스를 구성하는 방법을 보여줍니다. 두 서버는 그림 2-4에 나온 네트워크 예의 일부입니다.

다음 구성 파일은 **Goldweb** 서버에 대한 IPQoS 작업을 정의합니다. 이 서버는 프리미엄 SLA를 구매한 회사인 **Goldco**에 대한 웹 사이트를 호스트합니다.

예 3-1 프리미엄 웹 서버에 대한 샘플 IPQoS 구성 파일

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name goldweb
        next_action markAF11
        enable_stats FALSE
    }
    class {
        name video
        next_action markEF
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class goldweb
    }
    filter {
        name videoout
        sport videosrv
        direction LOCAL_OUT
        class video
    }
}

action {
    module dscpmk
    name markAF11
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}

action {
    module dscpmk
    name markEF
    params {
        global_stats TRUE
        dscp_map{0-63:46}
        next_action acct
    }
}

action {
    module flowacct
    name acct
    params {
        enable_stats TRUE
        timer 10000
        timeout 10000
        max_limit 2048
    }
}
```

예 3-1 프리미엄 웹 서버에 대한 샘플 IPQoS 구성 파일 (계속)

```
}
}
```

다음 구성 파일은 Userweb에 대한 IPQoS 작업을 정의합니다. 이 서버는 낮은 가격 또는 **최선 조건 SLA**의 개인을 위한 웹 사이트를 호스트합니다. 이 레벨의 서비스는 IPQoS 시스템에서 더 높은 가격 SLA의 고객 트래픽을 처리한 후 최선 조건 고객에게 제공할 수 있는 최상의 서비스를 보장합니다.

예 3-2 최선 조건 웹 서버에 대한 샘플 구성

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
    class {
        name Userweb
        next_action markAF12
        enable_stats FALSE
    }
    filter {
        name webout
        sport 80
        direction LOCAL_OUT
        class Userweb
    }
}

action {
    module dscpmk
    name markAF12
    params {
        global_stats FALSE
        dscp_map{0-63:12}
        next_action continue
    }
}
```

▼ IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법

유지 관리하기 가장 쉬운 디렉토리에서 첫번째 IPQoS 구성 파일을 만들 수 있습니다. 이 장의 작업에서는 IPQoS 구성 파일에 대한 위치로 /var/ipqos 디렉토리를 사용합니다. 다음 절차에서는 예 3-1에 소개된 IPQoS 구성 파일의 초기 세그먼트를 만듭니다.

주 - IPQoS 구성 파일을 만들 때 각 action 명령문과 절을 중괄호({})로 묶는 경우 주의하십시오. 중괄호 사용 예는 예 3-1을 참조하십시오.

1 프리미엄 웹 서버에 로그인하고 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

모든 IPQoS 구성 파일은 버전 번호 `fmt_version 1.0`이 첫번째 주석 처리되지 않은 행으로 시작되어야 합니다.

2 일반 IP 분류기 `ipgpc`를 구성하는 초기 action 명령문에서 여는 매개변수를 따릅니다.

이 초기 작업은 IPQoS 구성 파일을 구성하는 action 명령문 트리를 시작합니다. 예를 들어, `/var/ipqos/Goldweb.qos` 파일은 `ipgpc` 분류기를 호출하는 초기 action 명령문으로 시작됩니다.

```
fmt_version 1.0
```

```
action {
  module ipgpc
  name ipgpc.classify
```

`fmt_version 1.0` IPQoS 구성 파일을 시작합니다.

`action {` 작업 명령문을 시작합니다.

`module ipgpc` `ipgpc` 분류기를 구성 파일의 첫번째 작업으로 구성합니다.

`name ipgpc.classify` 항상 `ipgpc.classify`가 되어야 하는 분류기 action 명령문의 이름을 정의합니다.

action 명령문에 대한 자세한 구문 정보는 99 페이지 “action 명령문” 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

3 통계 매개변수 `global_stats`와 함께 `params` 절을 추가합니다.

```
params {
  global_stats TRUE
}
```

`ipgpc.classify` 명령문의 `global_stats TRUE` 매개변수는 해당 작업에 대한 통계 수집을 사용으로 설정합니다. 또한 `global_stats TRUE`는 클래스 절 정의에서 `enable_stats TRUE`를 지정할 때마다 클래스별 통계 수집을 사용으로 설정합니다.

통계를 설정하면 성능이 영향을 받습니다. 새 IPQoS 구성 파일에 대한 통계를 수집하여 IPQoS가 제대로 작동하는지 확인할 수 있습니다. 나중에 `global_stats` 인수를 `FALSE`로 변경하여 통계 수집을 해제할 수 있습니다.

전역 통계는 `params` 절에서 정의할 수 있는 유일한 매개변수 유형입니다. `params` 절에 대한 구문 및 기타 자세한 내용은 101 페이지 “params 절” 및 `ipqosconf(1M)` 매뉴얼 페이지를 참조하십시오.

4 프리미엄 서버로 향하는 트래픽을 식별하는 클래스를 정의합니다.

```
class {
    name goldweb
    next_action markAF11
    enable_stats FALSE
}
```

이 명령문을 클래스 절이라고 합니다. class 절에는 다음과 같은 내용이 있습니다.

- name goldweb Goldweb 서버로 향하는 트래픽을 식별하는 goldweb 클래스를 만듭니다.
- next_action markAF11 ipgpc 모듈이 goldweb 클래스의 패킷을 markAF11 작업 명령문에 전달하도록 지시합니다. markAF11 작업 명령문은 dscpmk 표시기를 호출합니다.
- enable_stats FALSE goldweb 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계는 설정되지 않습니다.

class 절의 구문에 대한 자세한 내용은 100 페이지 “class 절” 및 ipqosconf(1M) 매뉴얼 페이지를 참조하십시오.

5 가장 높은 우선 순위 전달을 가져야 하는 응용 프로그램을 식별하는 클래스를 정의합니다.

```
class {
    name video
    next_action marKEF
    enable_stats FALSE
}
```

- name video Goldweb 서버에서 나가는 스트리밍 비디오 트래픽을 식별하는 video 클래스를 만듭니다.
- next_action marKEF ipgpc가 처리를 완료한 후 ipgpc 모듈이 video 클래스의 패킷을 marKEF 명령문에 전달하도록 지시합니다. marKEF 명령문은 dscpmk 표시기를 호출합니다.
- enable_stats FALSE video 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계 수집은 설정되지 않습니다.

- 참조
- 방금 만든 클래스에 대한 필터를 정의하려면 52 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
 - 구성 파일에 대한 다른 클래스 절을 만들려면 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 필터를 정의하는 방법

다음 절차에서는 IPQoS 구성 파일에서 클래스에 대한 필터를 정의하는 방법을 보여줍니다.

시작하기 전에 이 절차에서는 이미 파일 만들기를 시작하고 클래스를 정의했다고 가정합니다. 단계는 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”에서 만든 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

주-IPQoS 구성 파일을 만들 때 각 class 절 및 각 filter 절을 중괄호({})로 묶는 경우 주의하십시오. 중괄호 사용 예는 예 3-1을 참조하십시오.

1 IPQoS 구성 파일을 열고 정의한 마지막 클래스의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 class 절 이후 시작할 수 있습니다.

```
class {
    name video
    next_action markEF
    enable_stats FALSE
}
```

2 IPQoS 시스템의 송신 트래픽을 선택하는 filter 절을 정의합니다.

```
filter {
    name webout
    sport 80
    direction LOCAL_OUT
    class goldweb
}
```

name webout webout 이름을 필터에 제공합니다.

sport 80 HTTP(웹) 트래픽에 대해 잘 알려진 포트인 소스 포트 80의 트래픽을 선택합니다.

direction LOCAL_OUT 로컬 시스템의 송신 트래픽을 추가로 선택합니다.

class goldweb 필터가 속한 클래스(이 경우 goldweb 클래스)를 식별합니다.

IPQoS 구성 파일의 filter 절에 대한 구문 및 자세한 내용은 100 페이지 “filter 절”을 참조하십시오.

3 IPQoS 시스템에서 스트리밍 비디오 트래픽을 선택하는 filter 절을 정의합니다.

```
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
```

name videoout	videoout 이름을 필터에 제공합니다.
sport videosrv	이 시스템의 스트리밍 비디오 응용 프로그램에 대해 이전에 정의한 포트인 소스 포트 videosrv의 트래픽을 선택합니다.
direction LOCAL_OUT	로컬 시스템의 송신 트래픽을 추가로 선택합니다.
class video	필터가 속한 클래스(이 경우 video 클래스)를 식별합니다.

- 참조
- 표시기 모듈에 대한 전달 동작을 정의하려면 53 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”을 참조하십시오.
 - 측정 모듈에 대한 플로우 제어 매개변수를 정의하려면 66 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.
 - IPQoS 구성 파일을 활성화하려면 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
 - 추가 필터를 정의하려면 52 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
 - 응용 프로그램의 트래픽 플로우에 대한 클래스를 만들려면 62 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법

다음 절차에서는 IPQoS 구성 파일에 클래스에 대한 홈별 동작을 추가하여 트래픽 전달을 정의하는 방법을 보여줍니다.

시작하기 전에 이 절차에서는 이미 정의된 클래스와 필터가 있는 기존 IPQoS 구성 파일이 있다고 가정합니다. 단계는 예 3-1에서 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

주 - 이 절차에서는 dscpmk 표시기 모듈을 사용하여 트래픽 전달을 구성하는 방법을 보여줍니다. dlclosmk 표시기를 사용하여 VLAN 시스템에서 트래픽 전달에 대한 자세한 내용은 93 페이지 “VLAN 장치에서 dlcsmk 표시기 사용”을 참조하십시오.

1 IPQoS 구성 파일을 열고 정의한 마지막 필터의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 filter 절 이후 시작할 수 있습니다.

```
filter {
    name videoout
    sport videosrv
    direction LOCAL_OUT
    class video
}
```

이 filter 절은 ipgpc 분류기 action 명령문의 끝에 있습니다. 그러므로 필터를 종료하는 닫는 중괄호와 action 명령문을 종료하는 두번째 닫는 중괄호가 필요합니다.

2 다음 action 명령문으로 표시기를 호출합니다.

```
action {
    module dscpmk
    name markAF11
```

module dscpmk 표시기 모듈 dscpmk를 호출합니다.

name markAF11 markAF11 이름을 action 명령문에 제공합니다.

이전에 정의한 클래스 goldweb에는 next_action markAF11 명령문이 포함되어 있습니다. 이 명령문은 분류기가 처리를 완료한 후 트래픽 플로우를 markAF11 작업 명령문에 보냅니다.

3 표시기가 트래픽 플로우에 대해 수행할 작업을 정의합니다.

```
    params {
        global_stats FALSE
        dscp_map{0-63:10}
        next_action continue
    }
}
```

global_stats FALSE markAF11 표시기 action 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 통계는 수집되지 않습니다.

dscp_map{0-63:10} 표시기에서 현재 처리 중인 트래픽 클래스 goldweb의 패킷 헤더에 DSCP 10을 지정합니다.

next_action continue 트래픽 클래스 goldweb의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

DSCP 10은 표시기가 dscp 맵의 모든 항목을 십진수 값 10(이진수 001010)으로 설정하도록 지시합니다. 이 코드 포인트는 goldweb 트래픽 클래스의 패킷이 AF11 홉별 동작에 종속된다는 것을 나타냅니다. AF11은 DSCP 10의 모든 패킷이 낮은 삭제, 높은 우선 순위의 서비스를 받도록 보장합니다. 따라서 Goldweb의 프리미엄 고객에 대한 송신 트래픽에는 AF(보장 전달) PHB에 대해 사용 가능한 가장 높은 우선 순위가 제공됩니다. AF에 대해 가능한 DSCP 표는 표 6-2를 참조하십시오.

4 다른 표시기 action 명령문을 시작합니다.

```
action {
    module dscpmk
    name markEF
```

module dscpmk 표시기 모듈 dscpmk를 호출합니다.

name markEF markEF 이름을 action 명령문에 제공합니다.

5 표시기가 트래픽 플로우에 대해 수행할 작업을 정의합니다.

```

params {
    global_stats TRUE
    dscp_map{0-63:46}
    next_action acct
}

```

- global_stats TRUE 스트리밍 비디오 패킷을 선택하는 video 클래스에 대한 통계 수집을 사용으로 설정합니다.
- dscp_map{0-63:46} 표시기에서 현재 처리 중인 트래픽 클래스 video의 패킷 헤더에 DSCP 46을 지정합니다.
- next_action acct dscpmk가 처리를 완료한 후 dscpmk 모듈이 video 클래스의 패킷을 acct action 명령문에 전달하도록 지시합니다. acct action 명령문은 flowacct 모듈을 호출합니다.

DSCP 46은 dscpmk 모듈이 dscp 맵의 모든 항목을 DS 필드에서 십진수 값 46(이진수 101110)으로 설정하도록 지시합니다. 이 코드 포인트는 video 트래픽 클래스의 패킷이 EF(빠른 전달) 홉별 동작에 종속된다는 것을 나타냅니다.

주 - EF에 대해 권장되는 코드 포인트는 46(이진수 101110)입니다. 기타 DSCP는 AF PHB를 패킷에 지정합니다.

EF PHB는 DSCP 46의 패킷이 IPQoS 및 Diffserv 인식 시스템에서 가장 높은 우선권을 받도록 보장합니다. 스트리밍 응용 프로그램에는 가장 높은 우선 순위의 서비스가 필요하므로 QoS 정책에서 스트리밍 응용 프로그램에 EF PHB를 지정하게 됩니다. 빠른 전달 PHB에 대한 자세한 내용은 91 페이지 “EF(빠른 전달) PHB”를 참조하십시오.

6 방금 만든 DSCP를 Diffserv 라우터의 해당하는 파일에 추가합니다.

자세한 내용은 69 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.

- 참조
- 트래픽 플로우에 대한 플로우 계산 통계 수집을 시작하려면 56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
 - 표시기 모듈에 대한 전달 동작을 정의하려면 53 페이지 “IPQoS 구성 파일에서 트래픽 전달을 정의하는 방법”을 참조하십시오.
 - 측정 모듈에 대한 플로우 제어 매개변수를 정의하려면 66 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.
 - IPQoS 구성 파일을 활성화하려면 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
 - 추가 필터를 정의하려면 52 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
 - 응용 프로그램의 트래픽 플로우에 대한 클래스를 만들려면 62 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법

다음 절차에서는 IPQoS 구성 파일에서 트래픽 클래스에 대한 계산을 사용으로 설정하는 방법을 보여줍니다. 절차는 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”에 소개된 video 클래스에 대한 플로우 계산을 정의하는 방법을 보여줍니다. 이 클래스는 프리미엄 고객의 SLA의 일부로 청구되어야 하는 스트리밍 비디오 트래픽을 선택합니다.

시작하기 전에 이 절차에서는 이미 정의된 클래스, 필터, 측정 작업(해당하는 경우) 및 표시 작업(해당하는 경우)이 있는 기존 IPQoS 구성 파일이 있다고 가정합니다. 단계는 예 3-1에서 /var/ipqos/Goldweb.qos 파일 만들기를 계속합니다.

1 IPQoS 구성 파일을 열고 정의한 마지막 action 명령문의 끝을 찾습니다.

예를 들어, IPQoS 사용 서버 Goldweb에서 /var/ipqos/Goldweb.qos의 다음 markEF action 명령문 이후 시작할 수 있습니다.

```
action {
  module dscpmk
  name markEF
  params {
    global_stats TRUE
    dscp_map{0-63:46}
    next_action acct
  }
}
```

2 플로우 정산을 호출하는 action 명령문을 시작합니다.

```
action {
  module flowacct
  name acct
```

module flowacct 플로우 계산 모듈 flowacct를 호출합니다.

name acct acct 이름을 action 명령문에 제공합니다.

3 트래픽 클래스에 대한 정산을 제어하는 params 절을 정의합니다.

```
params {
  global_stats TRUE
  timer 10000
  timeout 10000
  max_limit 2048
  next_action continue
}
```

global_stats TRUE 스트리밍 비디오 패킷을 선택하는 video 클래스에 대한 통계 수집을 사용으로 설정합니다.

timer 10000	시간 초과된 플로우에 대해 플로우 테이블이 검사되는 간격(밀리초)를 지정합니다. 이 매개변수에서 간격은 10000밀리초입니다.
timeout 10000	최소 간격 시간 초과 값을 지정합니다. 플로우 패킷이 시간 초과 간격 동안 보이지 않으면 플로우가 “시간 초과”됩니다. 이 매개변수에서 패킷은 10000밀리초 후 시간 초과됩니다.
max_limit 2048	이 작업 인스턴스에 대한 플로우 테이블에서 최대 활성 플로우 레코드 수를 설정합니다.
next_action continue	트래픽 클래스 video의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

flowacct 모듈은 지정된 timeout 값에 도달할 때까지 특정 클래스의 패킷 플로우에 대한 통계 정보를 수집합니다.

- 참조
- 라우터에 대한 휴별 동작을 구성하려면 69 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.
 - IPQoS 구성 파일을 활성화하려면 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
 - 응용 프로그램의 트래픽 플로우에 대한 클래스를 만들려면 62 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.

▼ 최선 조건 웹 서버에 대한 IPQoS 구성 파일을 만드는 방법

최선 조건 웹 서버에 대한 IPQoS 구성 파일은 프리미엄 웹 서버에 대한 IPQoS 구성 파일과 약간 다릅니다. 예로 절차에서는 예 3-2의 구성 파일을 사용합니다.

- 1 최선 조건 웹 서버에 로그인합니다.
- 2 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

```

fmt_vesion 1.0
action {
    module ipgpc
    name ipgpc.classify
    params {
        global_stats TRUE
    }
}
    
```

/var/ipqos/userweb.qos 파일은 ipgpc 분류기를 호출하는 부분 action 명령문으로 시작되어야 합니다. 또한 action 명령문에는 통계 수집을 설정하는 params 절도 있어야 합니다. 이 action 명령문에 대한 설명은 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

3 최선 조건 웹 서버로 향하는 트래픽을 식별하는 클래스를 정의합니다.

```
class {
    name userweb
    next_action markAF12
    enable_stats FALSE
}
```

- name userweb 사용자의 웹 트래픽 전달을 위한 userweb이라는 클래스를 만듭니다.
- next_action markAF1 ipgpc가 처리를 완료한 후 ipgpc 모듈이 userweb 클래스의 패킷을 markAF12 action 명령문에 전달하도록 지시합니다. markAF12 action 명령문은 dscpmk 표시기를 호출합니다.
- enable_stats FALSE userweb 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 이 클래스에 대한 통계 수집은 발생하지 않습니다.

class 절 작업에 대한 설명은 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

4 userweb 클래스에 대한 트래픽 플로우를 선택하는 filter 절을 정의합니다.

```
filter {
    name webout
    sport 80
    direction LOCAL_OUT
    class userweb
}
}
```

- name webout webout 이름을 필터에 제공합니다.
 - sport 80 HTTP(웹) 트래픽에 대해 잘 알려진 포트인 소스 포트 80의 트래픽을 선택합니다.
 - direction LOCAL_OUT 로컬 시스템의 송신 트래픽을 추가로 선택합니다.
 - class userweb 필터가 속한 클래스(이 경우 userweb 클래스)를 식별합니다.
- filter 절 작업에 대한 설명은 52 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.

5 dscpmk 표시기를 호출하는 action 명령문을 시작합니다.

```
action {
    module dscpmk
    name markAF12
}
```

`module dscpmk` 표시기 모듈 `dscpmk`를 호출합니다.

`name markAF12` `markAF12` 이름을 `action` 명령문에 제공합니다.

이전에 정의한 클래스 `userweb`에는 `next_action markAF12` 명령문이 포함되어 있습니다. 이 명령문은 분류기가 처리를 완료한 후 트래픽 플로우를 `markAF12 action` 명령문에 보냅니다.

6 표시기가 트래픽 플로우 처리를 위해 사용할 매개변수를 정의합니다.

```
params {
    global_stats FALSE
    dscp_map{0-63:12}
    next_action continue
}
```

`global_stats FALSE` `markAF12` 표시기 `action` 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 `FALSE`이므로 통계 수집은 발생하지 않습니다.

`dscp_map{0-63:12}` 표시기에서 현재 처리 중인 트래픽 클래스 `userweb`의 패킷 헤더에 DSCP 12를 지정합니다.

`next_action continue` 트래픽 클래스 `userweb`의 패킷에 추가 처리가 필요하지 않으며 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있음을 나타냅니다.

DSCP 12는 표시기가 `dscp` 맵의 모든 항목을 십진수 값 12(이진수 001100)로 설정하도록 지시합니다. 이 코드 포인트는 `userweb` 트래픽 클래스의 패킷이 AF12 홉별 동작에 종속된다는 것을 나타냅니다. AF12는 DS 필드에서 DSCP 12의 모든 패킷이 중간 삭제, 높은 우선 순위의 서비스를 받도록 보장합니다.

7 IPQoS 구성 파일을 완료한 경우 구성을 적용합니다.

- 참조
- 응용 프로그램의 트래픽 플로우에 대한 클래스 및 기타 구성을 추가하려면 62 페이지 “애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법”을 참조하십시오.
 - 라우터에 대한 홉별 동작을 구성하려면 69 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.
 - IPQoS 구성 파일을 활성화하려면 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.

애플리케이션 서버에 대한 IPQoS 구성 파일 만들기

이 절에서는 고객에게 주요 응용 프로그램을 제공하는 애플리케이션 서버에 대한 구성 파일을 만드는 방법을 설명합니다. 이 절차에서는 예로 [그림 2-4](#)의 BigAPPS 서버를 사용합니다.

다음 구성 파일은 BigAPPS 서버에 대한 IPQoS 작업을 정의합니다. 이 서버는 고객을 위한 FTP, 전자 메일(SMTP) 및 네트워크 뉴스(NNTP)를 호스트합니다.

예 3-3 애플리케이션 서버에 대한 샘플 IPQoS 구성 파일

```
fmt_version 1.0

action {
  module ipgpc
  name ipgpc.classify
  params {
    global_stats TRUE
  }
  class {
    name smtp
    enable_stats FALSE
    next_action markAF13
  }
  class {
    name news
    next_action markAF21
  }
  class {
    name ftp
    next_action meterftp
  }
  filter {
    name smtpout
    sport smtp
    class smtp
  }
  filter {
    name newsout
    sport nntp
    class news
  }
  filter {
    name ftpout
    sport ftp
    class ftp
  }
  filter {
    name ftpdata
    sport ftp-data
    class ftp
  }
}
action {
  module dscpmk
```

예 3-3 애플리케이션 서버에 대한 샘플 IPQoS 구성 파일 (계속)

```

    name markAF13
    params {
        global_stats FALSE
        dscp_map{0-63:14}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF21
    params {
        global_stats FALSE
        dscp_map{0-63:18}
        next_action continue
    }
}
action {
    module tokenmt
    name meterftp
    params {
        committed_rate 50000000
        committed_burst 50000000
        red_action_name AF31
        green_action_name markAF22
        global_stats TRUE
    }
}
action {
    module dscpmk
    name markAF31
    params {
        global_stats TRUE
        dscp_map{0-63:26}
        next_action continue
    }
}
action {
    module dscpmk
    name markAF22
    params {
        global_stats TRUE
        dscp_map{0-63:20}
        next_action continue
    }
}
}

```

▼ 애플리케이션 서버에 대한 IPQoS 구성 파일을 구성하는 방법

- 1 IPQoS 사용 애플리케이션 서버에 로그인하고 .qos 확장자로 새 IPQoS 구성 파일을 만듭니다.

예를 들어, 애플리케이션 서버에 대해 /var/ipqos/BigAPPS.qos 파일을 만듭니다. 다음 필수 문구로 시작하여 ipgpc 분류기를 호출하는 action 명령문을 시작합니다.

```
fmt_version 1.0
```

```
action {
  module ipgpc
  name ipgpc.classify
  params {
    global_stats TRUE
  }
}
```

여는 action 명령문에 대한 설명은 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

- 2 BigAPPS 서버에서 세 응용 프로그램의 트래픽을 선택하는 클래스를 만듭니다.

action 명령문을 연 후 클래스 정의를 추가합니다.

```
class {
  name smtp
  enable_stats FALSE
  next_action markAF13
}
class {
  name news
  next_action markAF21
}
class {
  name ftp
  enable_stats TRUE
  next_action meterftp
}
```

`name smtp` SMTP 응용 프로그램에서 처리할 전자 메일 트래픽 플로우를 포함하는 smtp라는 클래스를 만듭니다.

`enable_stats FALSE` smtp 클래스에 대한 통계 수집을 사용으로 설정합니다. 하지만 `enable_stats`의 값이 FALSE이므로 이 클래스에 대한 통계는 수집되지 않습니다.

`next_action markAF13` ipgpc가 처리를 완료한 후 ipgpc 모듈이 smtp 클래스의 패킷을 markAF13 action 명령문에 전달하도록 지시합니다.

`name news` NNTP 응용 프로그램에서 처리할 네트워크 뉴스 트래픽 플로우를 포함하는 news라는 클래스를 만듭니다.

next_action markAF21	ipgpc가 처리를 완료한 후 ipgpc 모듈이 news 클래스의 패킷을 markAF21 작업 명령문에 전달하도록 지시합니다.
name ftp	FTP 응용 프로그램에서 처리할 송신 트래픽을 처리하는 ftp라는 클래스를 만듭니다.
enable_stats TRUE	ftp 클래스에 대한 통계 수집을 사용으로 설정합니다.
next_action meterftp	ipgpc가 처리를 완료한 후 ipgpc 모듈이 ftp 클래스의 패킷을 meterftp action 명령문에 전달하도록 지시합니다.

클래스 정의에 대한 자세한 내용은 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.

3 2단계에서 정의한 클래스의 트래픽을 선택하는 filter 절을 정의합니다.

```

filter {
  name smtpout
  sport smtp
  class smtp
}
filter {
  name newsout
  sport nntp
  class news
}
  filter {
    name ftpout
    sport ftp
    class ftp
  }
  filter {
    name ftpdata
    sport ftp-data
    class ftp
  }
}

```

name smtpout	smtpout 이름을 필터에 제공합니다.
sport smtp	sendmail(SMTP) 응용 프로그램에 대해 잘 알려진 포트인 소스 포트 25의 트래픽을 선택합니다.
class smtp	필터가 속한 클래스(이 경우 smtp 클래스)를 식별합니다.
name newsout	newsout 이름을 필터에 제공합니다.
sport nntp	네트워크 뉴스(NNTP) 응용 프로그램에 대해 잘 알려진 포트 이름인 소스 포트 이름 nntp의 트래픽을 선택합니다.
class news	필터가 속한 클래스(이 경우 news 클래스)를 식별합니다.
name ftpout	ftpout 이름을 필터에 제공합니다.

sport ftp	FTP 트래픽에 대해 잘 알려진 포트 번호인 소스 포트 21의 제어 데이터를 선택합니다.
name ftpdata	ftpdata 이름을 필터에 제공합니다.
sport ftp-data	FTP 데이터 트래픽에 대해 잘 알려진 포트 번호인 소스 포트 20의 트래픽을 선택합니다.
class ftp	ftpout 및 ftpdata 필터가 속한 클래스(이 경우 ftp 클래스)를 식별합니다.

- 참조
- 필터를 정의하려면 52 페이지 “IPQoS 구성 파일에서 필터를 정의하는 방법”을 참조하십시오.
 - 응용 프로그램 트래픽에 대한 전달 동작을 정의하려면 64 페이지 “IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법”을 참조하십시오.
 - 측정 모듈을 사용하여 플로우 제어를 구성하려면 66 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”을 참조하십시오.
 - 플로우 계산을 구성하려면 56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 응용 프로그램 트래픽에 대한 전달을 구성하는 방법

다음 절차에서는 응용 프로그램 트래픽에 대한 전달을 구성하는 방법을 보여줍니다. 이 절차에서는 네트워크의 다른 트래픽보다 낮은 우선권을 가질 수 있는 응용 프로그램 트래픽 클래스에 대한 홉별 동작을 정의합니다. 단계는 예 3-3의 /var/ipqos/BigAPPS.qos 파일 만들기를 계속합니다.

시작하기 전에 이 절차에서는 표시할 응용 프로그램에 대해 이미 정의된 클래스와 필터가 있는 기존 IPQoS 구성 파일이 있다고 가정합니다.

- 1 애플리케이션 서버에 대해 만든 IPQoS 구성 파일을 열고 마지막 filter 절의 끝을 찾습니다.

/var/ipqos/BigAPPS.qos 파일에서 마지막 필터는 다음과 같습니다.

```
filter {
    name ftpdata
    sport ftp-data
    class ftp
}
}
```

2 표시기를 다음과 같이 호출합니다.

```
action {
  module dscpmk
  name markAF13
```

module dscpmk 표시기 모듈 dscpmk를 호출합니다.

name markAF13 markAF13 이름을 action 명령문에 제공합니다.

3 전자 메일 트래픽 플로우에 대해 표시할 흡별 동작을 정의합니다.

```
  params {
    global_stats FALSE
    dscp_map{0-63:14}
    next_action continue
  }
}
```

global_stats FALSE markAF13 표시기 action 명령문에 대한 통계 수집을 사용으로 설정합니다. 하지만 enable_stats의 값이 FALSE이므로 통계는 수집되지 않습니다.

dscp_map{0-63:14} 표시기에서 현재 처리 중인 트래픽 클래스 smtp의 패킷 헤더에 DSCP 14를 지정합니다.

next_action continue 트래픽 클래스 smtp의 패킷에 추가 처리가 필요하지 않음을 나타냅니다. 그러면 이러한 패킷은 네트워크 스트림으로 돌아갈 수 있습니다.

DSCP 14는 표시기가 dscp 맵의 모든 항목을 십진수 값 14(이진수 001110)로 설정하도록 지시합니다. DSCP 14는 AF13 흡별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 14의 smtp 트래픽 클래스 패킷을 표시합니다.

AF13은 DSCP 14의 모든 패킷을 높은 삭제 우선권으로 지정합니다. 하지만 AF13은 클래스 1 우선 순위도 보장하므로 라우터는 대기열에서 나가는 전자 메일 트래픽을 높은 우선 순위로 보장합니다. 가능한 AF 코드 포인트 표는 표 6-2를 참조하십시오.

4 네트워크 뉴스 트래픽에 대한 흡별 동작을 정의하는 표시기 action 명령문을 추가합니다.

```
action {
  module dscpmk
  name markAF21
  params {
    global_stats FALSE
    dscp_map{0-63:18}
    next_action continue
  }
}
```

name markAF21 markAF21 이름을 action 명령문에 제공합니다.

dscp_map{0-63:18} 표시기에서 현재 처리 중인 트래픽 클래스 nntp의 패킷 헤더에 DSCP 18을 지정합니다.

DSCP 18은 표시기가 dscp 맵의 모든 항목을 십진수 값 18(이진수 010010)로 설정하도록 지시합니다. DSCP 18은 AF21 휴별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 18의 news 트래픽 클래스 패킷을 표시합니다.

AF21은 DSCP 18의 모든 패킷이 낮은 삭제 우선권을 받도록 보장하지만 클래스 2 우선 순위를 가집니다. 따라서 네트워크 뉴스 트래픽이 삭제될 가능성은 낮습니다.

- 참조
- 웹 서버에 대한 구성 정보를 추가하려면 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.
 - 측정 모듈을 사용하여 플로우 제어룰 구성하려면 66 페이지 “IPQoS 구성 파일에서 플로우 제어룰 구성하는 방법”을 참조하십시오.
 - 플로우 계산을 구성하려면 56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
 - 라우터에 대한 전달 동작을 구성하려면 69 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.
 - IPQoS 구성 파일을 활성화하려면 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.

▼ IPQoS 구성 파일에서 플로우 제어를 구성하는 방법

특정 트래픽 플로우가 네트워크로 전송되는 속도를 제어하려면 측정기에 대한 매개변수를 정의해야 합니다. IPQoS 구성 파일에서 두 가지 측정기 모듈 tokenmt 또는 tswtclmt를 사용할 수 있습니다.

다음 절차에서는 예 3-3의 애플리케이션 서버에 대한 IPQoS 구성 파일 만들기를 계속합니다. 이 절차에서는 측정기뿐 아니라 측정기 action 명령문 내에서 호출되는 두 표시기 작업도 구성합니다.

시작하기 전에 단계에서는 플로우를 제어할 응용 프로그램에 대한 클래스 및 필터를 이미 정의했다고 가정합니다.

1 애플리케이션 서버에 대해 만든 IPQoS 구성 파일을 엽니다.

/var/ipqos/BigAPPS.qos 파일에서 다음 표시기 작업 이후 시작합니다.

```
action {
  module dscpmk
  name markAF21
  params {
    global_stats FALSE
    dscp_map{0-63:18}
    next_action continue
  }
}
```

- 2 ftp 클래스의 트래픽을 플로우 제어하는 측정기 action 명령문을 만듭니다.

```
action {
  module tokenmt
  name meterftp
```

module tokenmt tokenmt 측정기를 호출합니다.

name meterftp meterftp 이름을 action 명령문에 제공합니다.

- 3 측정기의 속도를 구성하는 매개변수를 추가합니다.

```
params {
  committed_rate 50000000
  committed_burst 50000000
```

committed_rate 50000000 ftp 클래스의 트래픽에 전송 속도 50,000,000bps를 지정합니다.

committed_burst 50000000 ftp 클래스의 트래픽에 버스트 크기 50,000,000비트를 커밋합니다.

tokenmt 매개변수에 대한 설명은 89 페이지 “두 속도 측정기로 tokenmt 구성”을 참조하십시오.

- 4 트래픽 준수 우선권을 구성하는 매개변수를 추가합니다.

```
red_action markAF31
green_action_name markAF22
global_stats TRUE
}
```

red_action_name markAF31 ftp 클래스의 트래픽 플로우가 약정된 속도를 초과할 경우 패킷이 markAF31 표시기 action 명령문으로 보내짐을 나타냅니다.

green_action_name markAF22 ftp의 트래픽 플로우가 약정된 속도를 준수할 경우 패킷이 markAF22 작업 명령문으로 보내짐을 나타냅니다.

global_stats TRUE ftp 클래스에 대한 측정 통계를 사용으로 설정합니다.

트래픽 준수에 대한 자세한 내용은 88 페이지 “측정기 모듈”을 참조하십시오.

- 5 홉별 동작을 ftp 클래스의 비준수 트래픽 플로우에 지정하는 표시기 action 명령문을 추가합니다.

```
action {
  module dscpmk
  name markAF31
  params {
    global_stats TRUE
    dscp_map{0-63:26}
    next_action continue
```

```

    }
}
module dscpmk          표시기 모듈 dscpmk를 호출합니다.
name markAF31          markAF31 이름을 action 명령문에 제공합니다.
global_stats TRUE     ftp 클래스에 대한 통계를 사용으로 설정합니다.
dscp_map{0-63:26}    이 트래픽이 약정된 속도를 초과할 때마다 트래픽 클래스
                    ftp의 패킷 헤더에 DSCP 26을 지정합니다.
next_action continue  트래픽 클래스 ftp의 패킷에 추가 처리가 필요하지 않음을
                    나타냅니다. 그러면 이러한 패킷은 네트워크 스트림으로
                    돌아갈 수 있습니다.

```

DSCP 26은 표시기가 dscp 맵의 모든 항목을 십진수 값 26(이진수 011010)으로 설정하도록 지시합니다. DSCP 26은 AF31 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 26의 ftp 트래픽 클래스 패킷을 표시합니다.

AF31은 DSCP 26의 모든 패킷이 낮은 삭제 우선권을 받도록 보장하지만 클래스 3 우선 순위를 가집니다. 따라서 비준수 FTP 트래픽이 삭제될 가능성은 낮습니다. 가능한 AF 코드 포인트 표는 표 6-2를 참조하십시오.

6 약정된 속도를 준수하는 ftp 트래픽 플로우에 홉별 동작을 지정하는 표시기 action 명령문을 추가합니다.

```

action {
  module dscpmk
  name markAF22
  params {
    global_stats TRUE
    dscp_map{0-63:20}
    next_action continue
  }
}
name markAF22          markAF22 이름을 marker 작업에 제공합니다.
dscp_map{0-63:20}    ftp 트래픽이 구성된 속도를 준수할 때마다 트래픽 클래스 ftp의
                    패킷 헤더에 DSCP 20을 지정합니다.

```

DSCP 20은 표시기가 dscp 맵의 모든 항목을 십진수 값 20(이진수 010100)으로 설정하도록 지시합니다. DSCP 20은 AF22 홉별 동작을 설정합니다. 표시기는 DS 필드에서 DSCP 20의 ftp 트래픽 클래스 패킷을 표시합니다.

AF22는 DSCP 20의 모든 패킷이 클래스 2 우선 순위로 중간 삭제 우선권을 받도록 보장합니다. 따라서 준수하는 FTP 트래픽은 IPQoS 시스템에서 동시에 전송되는 플로우 중에서 중간 삭제 우선권이 보장됩니다. 하지만 라우터는 클래스 1 중간 삭제 우선권 표시 이상의 트래픽 클래스에 더 높은 전달 우선 순위를 제공합니다. 가능한 AF 코드 포인트 표는 표 6-2를 참조하십시오.

7 애플리케이션 서버에 대해 만든 DSCP를 Diffserv 라우터의 해당하는 파일에 추가합니다.

- 참조
- IPQoS 구성 파일을 활성화하려면 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”을 참조하십시오.
 - 웹 서버에 대한 구성 정보를 추가하려면 49 페이지 “IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법”을 참조하십시오.
 - 플로우 계산을 구성하려면 56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
 - 라우터에 대한 전달 동작을 구성하려면 69 페이지 “IPQoS 사용 네트워크에서 라우터를 구성하는 방법”을 참조하십시오.

라우터에서 차별화 서비스 제공

진정한 차별화 서비스를 제공하려면 26 페이지 “Diffserv 네트워크에 대한 하드웨어 전략”에 설명된 대로 네트워크 토폴로지에 Diffserv 인식 라우터를 포함시켜야 합니다. 라우터에서 Diffserv를 구성하고 해당 라우터의 파일을 업데이트하기 위한 실제 단계는 이 설명서의 범위를 벗어납니다.

이 절에서는 네트워크의 다양한 IPQoS 사용 시스템 및 Diffserv 라우터 사이에서 전달 정보를 조정하기 위한 일반적인 단계를 설명합니다.

▼ IPQoS 사용 네트워크에서 라우터를 구성하는 방법

다음 절차에서는 그림 2-4의 토폴로지를 예로 사용합니다.

시작하기 전에 다음 절차에서는 이 장의 이전 작업을 수행하여 네트워크에서 IPQoS 시스템을 이미 구성했다고 가정합니다.

- 1 네트워크의 모든 IPQoS 사용 시스템에 대한 구성 파일을 검토합니다.
- 2 QoS 다양한 정책에서 사용되는 각 코드 포인트를 식별합니다.

코드 포인트 및 코드 포인트가 적용되는 시스템과 클래스를 나열합니다. 다음 표는 동일한 코드 포인트를 사용했을 수 있는 영역을 나타냅니다. 이 연습은 수용할 수 있습니다. 하지만 동일하게 표시된 클래스의 우선권을 결정하려면 IPQoS 구성 파일에서 다른 조건(예: precedence 선택기)을 제공해야 합니다.

예를 들어, 이 장의 절차에서 사용된 샘플 네트워크의 경우 다음 코드 포인트 표를 만들 수 있습니다.

시스템	클래스	PHB	DS 코드 포인트
Goldweb	video	EF	46 (101110)
Goldweb	goldweb	AF11	10 (001010)
Userweb	webout	AF12	12 (001100)
BigAPPS	sntp	AF13	14 (001110)
BigAPPS	news	AF18	18 (010010)
BigAPPS	ftp 준수 트래픽	AF22	20 (010100)
BigAPPS	ftp 비준수 트래픽	AF31	26 (011010)

3 네트워크의 IPQoS 구성 파일에서 코드 포인트를 Diffserv 라우터의 해당하는 파일에 추가합니다.

제공하는 코드 포인트는 라우터의 Diffserv 일정 예약 방식을 구성하는 데 도움이 되어야 합니다. 자세한 내용은 라우터 제조업체의 설명서 및 웹 사이트를 참조하십시오.

IPQoS 시작 및 유지 관리(작업)

이 장에는 IPQoS 구성 파일을 활성화하고 IPQoS 관련 이벤트를 기록하기 위한 작업이 포함되어 있습니다. 다음 항목을 다룹니다.

- 71 페이지 “IPQoS 관리(작업 맵)”
- 72 페이지 “IPQoS 구성 적용”
- 73 페이지 “IPQoS 메시지에 대한 syslog 로깅 사용”
- 74 페이지 “IPQoS 오류 메시지를 사용하여 문제 해결”

주 - IPQoS 기능은 이후 릴리스에서 제거될 수 있습니다. 비슷한 대역폭 리소스 제어 기능을 지원하는 `dladm`, `flowadm` 및 관련 명령을 대신 사용하는 것이 좋습니다. 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

IPQoS 관리(작업 맵)

이 절에서는 Oracle Solaris 시스템에서 IPQoS를 시작하고 유지 관리하기 위한 작업을 나열합니다. 이 작업을 사용하기 전에 45 페이지 “IPQoS 구성 파일에서 QoS 정책 정의(작업 맵)”에 설명된 대로 완성된 IPQoS 구성 파일을 가지고 있어야 합니다.

다음 표에서는 이러한 작업을 나열하고 설명하며 이러한 작업을 완료하는 방법을 자세히 설명하는 링크를 제공합니다.

작업	설명	수행 방법
1. 시스템에서 IPQoS를 구성합니다.	<code>ipqosconf</code> 명령을 사용하여 시스템에서 IPQoS 구성 파일을 활성화합니다.	72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”
2. 각 시스템 부트 후 Oracle Solaris 시작 스크립트가 디버깅된 IPQoS 구성 파일을 적용하도록 합니다.	시스템이 재부트할 때마다 IPQoS 구성이 적용되도록 합니다.	73 페이지 “재부트 때마다 IPQoS 구성이 적용되도록 하는 방법”

작업	설명	수행 방법
3. IPQoS에 대해 syslog 로깅을 사용으로 설정합니다.	항목을 추가하여 IPQoS 메시지의 syslog 로깅을 사용으로 설정합니다.	73 페이지 “부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법”
4. 발생하는 IPQoS 문제를 해결합니다.	오류 메시지를 사용하여 IPQoS 문제를 해결합니다.	표 4-1의 오류 메시지를 참조하십시오.

IPQoS 구성 적용

ipqosconf 명령을 사용하여 IPQoS 구성을 활성화하고 조작합니다.

▼ IPQoS 커널 모듈에 새 구성을 적용하는 방법

ipqosconf 명령을 사용하여 IPQoS 구성 파일을 읽고 UNIX 커널에서 IPQoS 모듈을 구성합니다. 다음 절차에서는 47 페이지 “웹 서버에 대한 IPQoS 구성 파일 만들기”에서 만든 /var/ipqos/Goldweb.qos 파일을 예로 사용합니다. 자세한 내용은 ipqosconf (1M) 매뉴얼 페이지를 참조하십시오.

1 관리자로 전환합니다.

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

2 새 구성을 적용합니다.

```
# /usr/sbin/ipqosconf -a/var/ipqos/Goldweb.qos
```

ipqosconf는 지정된 IPQoS 구성 파일의 정보를 Oracle Solaris 커널의 IPQoS 모듈에 씁니다. 이 예에서는 /var/ipqos/Goldweb.qos의 내용이 현재 Oracle Solaris 커널에 적용됩니다.

주 - -a 옵션을 사용하여 IPQoS 구성 파일을 적용할 경우 파일의 작업이 현재 세션에 대해서만 활성화됩니다.

3 새 IPQoS 구성을 테스트하고 디버깅합니다.

UNIX 유틸리티를 사용하여 IPQoS 동작을 추적하고 IPQoS 구현에 대한 통계를 수집합니다. 이 정보는 구성이 예상한 대로 작동하는지 여부를 결정하는데 도움이 됩니다.

- 참조 ■ IPQoS 모듈이 어떻게 작동하는지에 대한 통계를 보려면 82 페이지 “통계 정보 수집”을 참조하십시오.

- ipqosconf 메시지를 기록하려면 73 페이지 “IPQoS 메시지에 대한 syslog 로깅 사용”을 참조하십시오.
- 각 부트 후 현재 IPQoS 구성이 적용되도록 하려면 73 페이지 “재부트 때마다 IPQoS 구성이 적용되도록 하는 방법”을 참조하십시오.

▼ 재부트 때마다 IPQoS 구성이 적용되도록 하는 방법

재부트되더라도 IPQoS 구성을 명시적으로 유지해야 합니다. 그렇지 않으면 시스템이 재부트할 때까지 현재 구성이 적용됩니다. IPQoS가 시스템에서 올바르게 작동하는 경우 다음을 수행하여 재부트되더라도 구성이 유지되도록 하십시오.

1 관리자로 전환합니다.

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

2 커널 모듈에 IPQoS 구성이 존재하는지 테스트합니다.

```
# ipqosconf -l
```

구성이 존재하는 경우 ipqosconf가 화면에 구성을 표시합니다. 출력을 받지 못할 경우 72 페이지 “IPQoS 커널 모듈에 새 구성을 적용하는 방법”에 설명된 대로 구성을 적용합니다.

3 IPQoS 시스템이 재부트될 때마다 기존 IPQoS 구성이 적용되도록 합니다.

```
# /usr/sbin/ipqosconf -c
```

-c 옵션은 현재 IPQoS 구성이 부트 시 구성 파일 /etc/inet/ipqosinit.conf에 다시 나타나도록 합니다.

IPQoS 메시지에 대한 syslog 로깅 사용

IPQoS 부트 시 메시지를 기록하려면 다음 절차에 나온 대로 /etc/syslog.conf 파일을 수정해야 합니다.

▼ 부트 중 IPQoS 메시지 로깅을 사용으로 설정하는 방법

1 관리자로 전환합니다.

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

2 /etc/syslog.conf 파일을 엽니다.

3 다음 텍스트를 파일에 최종 항목으로 추가합니다.

```
user.info /var/adm/messages
```

열 사이에는 공백 대신 탭을 사용합니다.

이 항목은 IPQoS로 생성되는 모든 부트 시 메시지를 /var/adm/messages 파일에 기록합니다.

4 시스템을 재부트하여 메시지를 적용합니다.

예 4-1 /var/adm/messages의 IPQoS 출력

시스템이 재부트된 후 /var/adm/messages를 보면 출력에 다음과 유사한 IPQoS 로깅 메시지가 포함되어 있습니다.

```
May 14 10:44:33 ipqos-14 ipqosconf: [ID 815575 user.info]
New configuration applied.
May 14 10:44:46 ipqos-14 ipqosconf: [ID 469457 user.info]
Current configuration saved to init file.
May 14 10:44:55 ipqos-14 ipqosconf: [ID 435810 user.info]
Configuration flushed.
```

또한 IPQoS 시스템의 /var/adm/messages 파일에서 다음과 유사한 IPQoS 오류 메시지도 볼 수 있습니다.

```
May 14 10:56:47 ipqos-14 ipqosconf: [ID 123217 user.error]
Missing/Invalid config file fmt_version.
May 14 10:58:19 ipqos-14 ipqosconf: [ID 671991 user.error]
No ipgpc action defined.
```

이러한 오류 메시지에 대한 설명은 표 4-1을 참조하십시오.

IPQoS 오류 메시지를 사용하여 문제 해결

이 절에서는 IPQoS로 생성되는 오류 메시지 및 가능한 해결 방법에 대한 표가 포함되어 있습니다.

표 4-1 IPQoS 오류 메시지

오류 메시지	설명	해결 방법
Undefined action in parameter <i>parameter-name's</i> action <i>action-name</i>	IPQoS 구성 파일에서 <i>parameter-name</i> 에 지정한 작업 이름이 구성 파일에 존재하지 않습니다.	작업을 만듭니다. 또는 매개변수의 다른 기존 작업을 참조합니다.
action <i>action-name</i> involved in cycle	IPQoS 구성 파일에서 <i>action-name</i> 이 작업 순환의 일부이며, 이는 IPQoS에서 허용되지 않습니다.	작업 순환을 확인합니다. 그런 다음 IPQoS 구성 파일에서 순환 참조 중 하나를 제거합니다.

표 4-1 IPQoS 오류 메시지 (계속)

오류 메시지	설명	해결 방법
Action <i>action-name</i> isn't referenced by any other actions	비 <i>ipgpc</i> 작업 정의가 IPQoS에서 정의된 다른 작업에 의해 참조되지 않으며, 이는 IPQoS에서 허용되지 않습니다.	참조되지 않는 작업을 제거합니다. 또는 다른 작업이 현재 참조되지 않는 작업을 참조하도록 만듭니다.
Missing/Invalid config file <i>fmt_version</i>	구성 파일의 형식이 파일의 첫 번째 항목으로 지정되지 않았으며, 이는 IPQoS에서 필수입니다.	49 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"에 설명된 대로 형식 버전을 추가합니다.
Unsupported config file format version	구성 파일에 지정된 형식 버전이 IPQoS에서 지원되지 않습니다.	IPQoS의 Solaris 9/02 릴리스로 시작하는 데 필요한 <i>fmt_version 1.0</i> 으로 형식 버전을 변경합니다.
No <i>ipgpc</i> action defined.	구성 파일에서 <i>ipgpc</i> 분류기에 대한 작업을 정의하지 않았으며, 이는 IPQoS 필수 사항입니다.	49 페이지 "IPQoS 구성 파일을 만들고 트래픽 클래스를 정의하는 방법"에 나온 대로 <i>ipgpc</i> 에 대한 작업을 정의합니다.
Can't commit a null configuration	<i>ipqosconf -c</i> 를 실행하여 구성을 커밋할 때 해당 구성이 비어 있었으며, 이는 IPQoS에서 허용되지 않습니다.	구성 커밋을 시도하기 전에 구성 파일을 적용합니다. 자세한 내용은 72 페이지 "IPQoS 커널 모듈에 새 구성을 적용하는 방법"을 참조하십시오.
Invalid CIDR mask on line <i>line-number</i>	구성 파일에서 CIDR 마스크를 IP 주소에 대한 유효한 범위를 벗어난 IP 주소의 일부로 사용했습니다.	마스크 값이 IPv4의 경우 1-32 및 IPv6의 경우 1-128 범위에 있도록 변경합니다.
Address masks aren't allowed for host names line <i>line-number</i>	구성 파일에서 호스트 이름에 대한 CIDR 마스크를 정의했으며, 이는 IPQoS에서 허용되지 않습니다.	마스크를 제거하거나 호스트 이름을 IP 주소로 변경합니다.
Invalid module name line <i>line-number</i>	구성 파일에서 작업 명령문에 지정한 모듈 이름이 잘못되었습니다.	모듈 이름의 철자를 확인합니다. IPQoS 모듈 목록은 표 6-5를 참조하십시오.
<i>ipgpc</i> action has incorrect name line <i>line-number</i>	구성 파일에서 <i>ipgpc</i> 작업에 제공한 이름이 필요한 <i>ipgpc.classify</i> 가 아닙니다.	<i>ipgpc.classify</i> 작업 이름을 바꿉니다.
Second parameter clause not supported line <i>line-number</i>	구성 파일에서 단일 작업에 대해 두 매개변수 절을 지정했으며, 이는 IPQoS에서 허용되지 않습니다.	작업에 대한 모든 매개변수를 단일 매개변수 절로 합칩니다.
Duplicate named action	구성 파일에서 두 작업에 동일한 이름을 제공했습니다.	작업 중 하나의 이름을 바꾸거나 제거합니다.
Duplicate named filter/class in action <i>action-name</i>	동일 작업에서 두 필터 또는 두 클래스에 동일한 이름을 제공했으며, 이는 IPQoS 구성 파일에서 허용되지 않습니다.	필터 또는 클래스 중 하나의 이름을 바꾸거나 제거합니다.
Undefined class in filter <i>filter-name</i> in action <i>action-name</i>	구성 파일에서 필터가 작업에 정의되지 않은 클래스를 참조합니다.	클래스를 만들거나 기존 클래스에 대한 필터 참조를 변경합니다.

표 4-1 IPQoS 오류 메시지 (계속)

오류 메시지	설명	해결 방법
Undefined action in class <i>class-name</i> action <i>action-name</i>	클래스가 구성 파일에서 정의되지 않은 작업을 참조합니다.	작업을 만들거나 기존 작업에 대한 참조를 변경합니다.
Invalid parameters for action <i>action-name</i>	구성 파일에서 매개변수 중 하나가 잘못되었습니다.	이름이 지정된 작업으로 호출되는 모듈의 경우 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 항목을 참조합니다. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조할 수 있습니다.
Mandatory parameter missing for action <i>action-name</i>	구성 파일에서 작업에 대한 필수 매개변수를 정의하지 않았습니다.	이름이 지정된 작업으로 호출되는 모듈의 경우 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 항목을 참조합니다. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조할 수 있습니다.
Max number of classes reached in <code>ipgpc</code>	IPQoS 구성 파일의 <code>ipgpc</code> 작업에서 허용되는 것보다 많은 클래스를 지정했습니다. 최대 수는 10007입니다.	구성 파일을 검토하고 불필요한 클래스를 제거합니다. 또는 <code>/etc/system</code> 파일에 <code>ipgpc_max_classes</code> <i>class-number</i> 항목을 추가하여 최대 클래스 수를 늘릴 수 있습니다.
Max number of filters reached in action <code>ipgpc</code>	IPQoS 구성 파일의 <code>ipgpc</code> 작업에서 허용되는 것보다 많은 필터를 지정했습니다. 최대 수는 10007입니다.	구성 파일을 검토하고 불필요한 필터를 제거합니다. 또는 <code>/etc/system</code> 파일에 <code>ipgpc_max_filters</code> <i>filter-number</i> 항목을 추가하여 최대 필터 수를 늘릴 수 있습니다.
Invalid/missing parameters for filter <i>filter-name</i> in action <code>ipgpc</code>	구성 파일에 <i>filter-name</i> 필터에 잘못되거나 누락된 매개변수가 있습니다.	유효한 매개변수 목록은 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조하십시오.
Name not allowed to start with '!', line <i>line-number</i>	작업, 필터 또는 클래스 이름에 느낌표(!)로 시작했으며, 이는 IPQoS 파일에서 허용되지 않습니다.	느낌표를 제거하거나 작업, 클래스 또는 필터 이름을 바꿉니다.
Name exceeds the maximum name length line <i>line-number</i>	최대 길이 23자를 초과하는 작업, 클래스 또는 필터 이름을 구성 파일에 정의했습니다.	작업, 클래스 또는 필터에 더 짧은 이름을 제공합니다.
Array declaration line <i>line-number</i> is invalid	구성 파일에서 <i>line-number</i> 행의 매개변수에 대한 배열 선언이 잘못되었습니다.	잘못된 배열의 <code>action</code> 명령문으로 호출되는 배열 선언의 올바른 구문은 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”을 참조하십시오. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조하십시오.
Quoted string exceeds line, <i>line-number</i>	동일 행에서 문자열에 닫는 인용 부호가 없으며, 이는 구성 파일에서 필수입니다.	구성 파일에서 인용 문자열은 동일 행에서 시작되고 끝나야 합니다.
Invalid value, line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 제공된 값이 매개변수에 대해 지원되지 않습니다.	<code>action</code> 명령문으로 호출되는 모듈에 대해 허용되는 값은 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 설명을 참조하십시오. 또는 <code>ipqosconf(1M)</code> 매뉴얼 페이지를 참조할 수 있습니다.

표 4-1 IPQoS 오류 메시지 (계속)

오류 메시지	설명	해결 방법
Unrecognized value, line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 대한 값이 해당 매개변수에 대해 지원되는 열거 값이 아닙니다.	열거 값이 매개변수에 대해 올바른지 확인합니다. 인식할 수 없는 행 번호의 action 명령문으로 호출되는 모듈에 대한 설명은 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”을 참조하십시오. 또는 ipqosconf(1M) 매뉴얼 페이지를 참조할 수 있습니다.
Malformed value list line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 지정된 열거가 사양 구문을 준수하지 않습니다.	값 목록 형식이 잘못된 action 명령문으로 호출되는 모듈에 대한 올바른 구문은 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”의 모듈 설명을 참조하십시오. 또는 ipqosconf(1M) 매뉴얼 페이지를 참조할 수 있습니다.
Duplicate parameter line <i>line-number</i>	<i>line-number</i> 에 지정된 매개변수가 중복되었으며, 이는 구성 파일에서 허용되지 않습니다.	중복된 매개변수 중 하나를 제거합니다.
Invalid action name line <i>line-number</i>	구성 파일의 <i>line-number</i> 에 대한 작업에 사전 정의된 이름 “continue” 또는 “drop”을 사용하는 이름을 제공했습니다.	작업에서 사전 정의된 이름을 사용하지 않도록 작업 이름을 바꿉니다.
Failed to resolve src/dst host name for filter at line <i>line-number</i> , ignoring filter	ipqosconf가 구성 파일에 제공된 필터에 대해 정의된 소스 또는 대상 주소를 확인할 수 없습니다. 따라서 필터가 무시되었습니다.	필터가 중요한 경우 나중에 구성 적용을 시도합니다.
Incompatible address version line <i>line-number</i>	<i>line-number</i> 에서 주소의 IP 버전이 이전에 지정된 IP 주소 또는 ip_version 매개변수의 버전과 호환되지 않습니다.	두 충돌 항목이 호환되도록 변경합니다.
Action at line <i>line-number</i> has the same name as currently installed action, but is for a different module	시스템의 IPQoS 구성에 존재하는 작업의 모듈을 변경하려고 시도했으며, 이는 허용되지 않습니다.	새 구성을 적용하기 전에 현재 구성을 지웁니다.

플로우 계산 및 통계 수집 사용(작업)

이 장에서는 IPQoS 시스템이 처리하는 트래픽에 대한 계산 및 통계 정보를 얻는 방법을 설명합니다. 다음 항목을 다룹니다.

- 79 페이지 “플로우 계산 설정(작업 맵)”
- 80 페이지 “트래픽 플로우에 대한 정보 기록”
- 82 페이지 “통계 정보 수집”

주 - IPQoS 기능은 이후 릴리스에서 제거될 수 있습니다. 비슷한 대역폭 리소스 제어 기능을 지원하는 `dladm`, `flowadm` 및 관련 명령을 대신 사용하는 것이 좋습니다. 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

플로우 계산 설정(작업 맵)

다음 작업 맵에서는 `flowacct` 모듈을 사용하여 트래픽 플로우에 대한 정보를 얻는 일반적인 작업을 나열합니다. 이 맵에서는 해당 작업을 수행하는 절차와 관련된 링크도 제공합니다.

작업	설명	수행 방법
1. 트래픽 플로우에 대한 계산 정보를 포함할 파일을 만듭니다.	<code>acctadm</code> 명령을 사용하여 <code>flowacct</code> 를 통한 처리 결과를 보관할 파일을 만듭니다.	80 페이지 “플로우 계산 데이터에 대한 파일을 만드는 방법”
2. IPQoS 구성 파일에서 <code>flowacct</code> 매개변수를 정의합니다.	<code>timer</code> , <code>timeout</code> 및 <code>max_limit</code> 매개변수에 대한 값을 정의합니다.	56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”

`/var/ipqos/goldweb/account.info` flowacct의 플로우 레코드를 보관할 파일의 정규화된 경로 이름을 지정합니다.

`flow` 플로우 계산을 사용으로 설정하도록 `acctadm`에 지시합니다.

- 3 인수 없이 `acctadm`을 입력하여 IPQoS 시스템에서 플로우 계산에 대한 정보를 확인합니다.**
`acctadm`은 다음 출력을 생성합니다.

```
Task accounting: inactive
  Task accounting file: none
  Tracked task resources: none
  Untracked task resources: extended
    Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
  Untracked process resources: extended,host,mstate
    Flow accounting: active
    Flow accounting file: /var/ipqos/goldweb/account.info
  Tracked flow resources: basic
  Untracked flow resources: dsfield,ctime,lseen,projid,uid
```

마지막 네 개를 제외한 모든 항목은 Oracle Solaris 리소스 관리자 기능에 사용됩니다. 다음 표에서는 IPQoS와 관련된 항목에 대해 설명합니다.

항목	설명
Flow accounting: active	플로우 계산이 켜져 있음을 나타냅니다.
Flow accounting file: /var/ipqos/goldweb/account.info	현재 플로우 계산 파일의 이름을 지정합니다.
Tracked flow resources: basic	기본 플로우 속성만 추적됨을 나타냅니다.
Untracked flow resources: dsfield,ctime,lseen,projid,uid	파일에서 추적되지 않는 flowacct 속성을 나열합니다.

- 4 (옵션) 정산 파일에 확장 속성을 추가합니다.**

```
# acctadm -e extended -f /var/ipqos/goldweb/account.info flow
```

- 5 (옵션) 계산 파일에 기본 속성만 기록하도록 되돌립니다.**

```
# acctadm -d extended -e basic -f /var/ipqos/goldweb/account.info
-d 옵션은 확장 계산을 사용 안함으로 설정합니다.
```

- 6 플로우 계산 파일의 내용을 확인합니다.**

플로우 계산 파일 내용 확인 지침은 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리의 “libexacct에 대한 Perl 인터페이스”**에서 확인할 수 있습니다.

- 참조
- 확장 계산 기능에 대한 자세한 내용은 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리**의 4장, “확장 계정(개요)”을 참조하십시오.
 - IPQoS 구성 파일에서 flowacct 매개변수를 정의하려면 56 페이지 “IPQoS 구성 파일에서 클래스에 대한 계산을 사용으로 설정하는 방법”을 참조하십시오.
 - acctadm으로 만들어진 파일의 데이터를 인쇄하려면 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리**의 “libexacct에 대한 Perl 인터페이스”를 참조하십시오.

통계 정보수집

kstat 명령을 사용하여 IPQoS 모듈에서 통계 정보를 생성할 수 있습니다. 다음 구문을 사용하십시오.

```
/bin/kstat -m ipqos-module-name
```

표 6-5와 같이 유효한 IPQoS 모듈 이름을 지정할 수 있습니다. 예를 들어, dscpmk 표시자가 생성한 통계를 보려면 다음 형식의 kstat를 사용합니다.

```
/bin/kstat -m dscpmk
```

자세한 기술 정보는 kstat(1M) 매뉴얼 페이지를 참조하십시오.

예 5-1 IPQoS에 대한 kstat 통계

다음은 flowacct 모듈에 대한 통계를 얻기 위해 kstat를 실행하여 발생할 수 있는 결과의 예입니다.

```
# kstat -m flowacct
module: flowacct                instance: 3
name: Flowacct statistics       class:   flacct
    bytes_in_tbl                 84
    crtime                       345728.504106363
    epackets                      0
    flows_in_tbl                 1
    nbytes                        84
    npackets                      1
    snaptime                      345774.031843301
    usedmem                       256
```

class: flacct 트래픽 플로우가 속한 클래스의 이름(이 예의 경우 flacct)을 지정합니다.

bytes_in_tbl 플로우 테이블의 총 바이트수입니다. 총 바이트 수는 현재 플로우 테이블에 상주하는 모든 플로우 레코드의 합계(바이트)입니다. 이 플로우 테이블의 총 바이트 수는 84입니다. 테이블에 플로우가 없을 경우 bytes_in_tbl에 대한 값은 0입니다.

crtime 마지막으로 이 kstat 출력이 만들어진 시간입니다.

예 5-1 IPQoS에 대한 kstat 통계 (계속)

epackets	처리 중 오류가 발생한 패킷 수(이 예의 경우 0)입니다.
flows_in_tbl	플로우 테이블의 플로우 레코드 수(이 예의 경우 1)입니다. 테이블에 레코드가 없을 경우 flows_in_tbl에 대한 값은 0입니다.
nbytes	이 flowacct 작업 인스턴스가 확인한 총 바이트 수(이 예의 경우 84)입니다. 값에는 현재 플로우 테이블에 있는 바이트가 포함됩니다. 또한 값에는 시간이 초과되었으며 플로우 테이블에 더 이상 존재하지 않는 바이트가 포함됩니다.
npackets	이 flowacct 작업 인스턴스가 확인한 총 패킷 수(이 예의 경우 1)입니다. npackets에는 현재 플로우 테이블에 있는 패킷이 포함됩니다. 또한 npackets에는 시간이 초과되었으며 플로우 테이블에 더 이상 존재하지 않는 패킷이 포함됩니다.
usedmem	이 flowacct 인스턴스가 유지 관리하는 플로우 테이블에서 사용 중인 메모리(바이트)입니다. 이 예의 경우 usedmem 값은 256입니다. 플로우 테이블에 플로우 레코드가 없을 경우 usedmem에 대한 값은 0입니다.

IPQoS 세부 정보(참조)

이 장에는 다음 IPQoS 항목에 대한 세부 정보를 제공하는 참조 자료가 포함되어 있습니다.

- 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”
- 97 페이지 “IPQoS 구성 파일”
- 101 페이지 “ipqosconf 구성 유틸리티”

개요는 1 장, “IPQoS 소개(개요)”를 참조하십시오. 계획 정보는 2 장, “IPQoS 사용 네트워크 계획(작업)”을 참조하십시오. IPQoS 구성 절차는 3 장, “IPQoS 구성 파일 만들기(작업)”를 참조하십시오.

주 - IPQoS 기능은 이후 릴리스에서 제거될 수 있습니다. 비슷한 대역폭 리소스 제어 기능을 지원하는 `dladm`, `flowadm` 및 관련 명령을 대신 사용하는 것이 좋습니다. 자세한 내용은 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

IPQoS 아키텍처 및 Diffserv 모델

이 절에서는 IPQoS 아키텍처 및 IPQoS가 RFC 2475, *An Architecture for Differentiated Services* (<http://www.ietf.org/rfc/rfc2475.txt?number=2475>)에 정의된 차별화 서비스(Diffserv) 모델을 어떻게 구현하는지 설명합니다. IPQoS에는 Diffserv 모델의 다음 요소가 포함됩니다.

- 분류기
- 측정기
- 표시기

또한 IPQoS에는 플로우 정산 모듈 및 VLAN(virtual local area network) 장치와 함께 사용하기 위한 `dlcosmk` 표시기가 포함됩니다.

분류기 모듈

Diffserv 모델에서 **분류기**는 선택된 트래픽 플로우를 서로 다른 서비스 레벨이 적용되는 그룹으로 구성하기 위한 작업을 수행합니다. RFC 2475에서 정의된 분류기는 원래 경계 라우터를 위해 고안되었습니다. 반면, IPQoS 분류기 **ipgpc**은 로컬 네트워크의 내부에 있는 호스트의 트래픽 플로우를 처리하기 위해 고안되었습니다. 그러므로 IPQoS 시스템과 Diffserv 라우터가 모두 있는 네트워크는 더욱 뛰어난 차별화 서비스를 제공할 수 있습니다. **ipgpc**의 기술적인 설명은 **ipgpc(7ipp)** 매뉴얼 페이지를 참조하십시오.

ipgpc 분류기는 다음을 수행합니다.

1. IPQoS 사용 시스템의 IPQoS 구성 파일에 지정된 조건을 충족하는 트래픽 플로우를 선택합니다.
QoS 정책은 패킷 헤더에 있어야 하는 다양한 조건을 정의합니다. 이러한 조건을 **선택기**라고 합니다. **ipgpc** 분류기는 이러한 선택기를 IPQoS 시스템에서 수신한 패킷의 헤더와 비교한 다음 **ipgpc**는 모든 일치하는 패킷을 선택합니다.
2. IPQoS 구성 파일에 정의된 대로 패킷 플로우를 동일 특성을 가진 네트워크 트래픽인 **클래스**로 구분합니다.
3. 패킷의 DS(차별화 서비스) 필드 값에 DSCP(차별화 서비스 코드 포인트)가 있는지 검사합니다.
DSCP가 있으면 수신 트래픽이 전달 동작으로 발신자에 의해 표시되었는지 여부를 나타냅니다.
4. 특정 클래스의 패킷에 대해 IPQoS 구성 파일에서 지정된 추가 작업을 확인합니다.
5. 패킷을 IPQoS 구성 파일에서 지정된 다음 IPQoS 모듈에 전달하거나 패킷을 네트워크 스트림으로 돌려 보냅니다.

분류기의 개요는 **17 페이지 “분류기 (ipgpc) 개요”**를 참조하십시오. IPQoS 구성 파일에서 분류기 호출에 대한 자세한 내용은 **97 페이지 “IPQoS 구성 파일”**을 참조하십시오.

IPQoS 선택기

ipgpc 분류기는 IPQoS 구성 파일의 **filter** 절에서 사용할 수 있는 다양한 선택기를 지원합니다. 필터를 정의할 경우 항상 특정 클래스의 트래픽을 성공적으로 검색하는 데 필요한 최소수의 선택기를 사용하십시오. 정의하는 필터 수에 따라 IPQoS 성능이 영향을 받을 수 있습니다.

다음 표는 **ipgpc**에 대해 사용 가능한 선택기를 나열합니다.

표 6-1 IPQoS 분류기에 대한 필터 선택기

선택기	인수	선택되는 정보
saddr	IP 주소 번호	소스 주소입니다.
daddr	IP 주소 번호	대상 주소입니다.

표 6-1 IPQoS 분류기에 대한 필터 선택기 (계속)

선택기	인수	선택되는 정보
sport	/etc/services에서 정의된 포트 번호 또는 서비스 이름	트래픽 클래스가 발생한 소스 포트
dport	/etc/services에서 정의된 포트 번호 또는 서비스 이름	트래픽 클래스가 향하는 대상 포트
protocol	/etc/protocols에서 정의된 프로토콜 번호 또는 프로토콜 이름	이 트래픽 클래스에서 사용할 프로토콜
dsfield	0-63 값의 DSCP(DS 코드 포인트)	패킷에 적용할 전달 동작을 정의하는 DSCP. 이 매개변수가 지정되면 dsfield_mask 매개변수도 지정되어야 합니다.
dsfield_mask	0-255 값의 비트 마스크	dsfield 선택기와 함께 사용됨. dsfield_mask는 dsfield 선택기에 적용되어 일치시킬 해당비트를 결정합니다.
if_name	인터페이스 이름	특정 클래스의 수신 또는 송신 트래픽에 사용될 인터페이스
user	선택할 UNIX 사용자 ID 또는 사용자 이름 수. 패킷에 사용자 ID 또는 사용자 이름이 없으면 기본값 -1이 사용됩니다.	응용 프로그램에 제공된 사용자 ID
projid	선택할 프로젝트 ID 수	응용 프로그램에 제공된 프로젝트 ID
priority	우선 순위 번호. 가장 낮은 우선 순위는 0입니다.	이 클래스의 패킷에 제공된 우선 순위. 우선 순위는 동일 클래스에 대해 필터의 중요도 순서를 정렬하는데 사용됩니다.
direction	인수는 다음 중 하나가 될 수 있습니다.	IPQoS 시스템에서 패킷 플로우의 방향
	LOCAL_IN	IPQoS 시스템에 로컬 입력 트래픽
	LOCAL_OUT	IPQoS 시스템에 로컬 출력 트래픽
	FWD_IN	전달할 입력 트래픽
	FWD_OUT	전달할 출력 트래픽
precedence	우선권 값. 가장 높은 우선권은 0입니다.	우선권은 동일 우선 순위를 가진 필터 순서를 정렬하는데 사용됩니다.
ip_version	V4 또는 V6	패킷에서 사용되는 주소 지정 체계(IPv4 또는 IPv6)

측정기 모듈

측정기는 패킷별 기준에서 플로우의 전송 속도를 추적합니다. 그런 다음 측정기는 패킷이 구성된 매개변수를 준수하는지 여부를 확인합니다. 측정기 모듈은 패킷 크기, 구성된 매개변수 및 플로우 속도에 의존하는 작업 세트에서 패킷에 대한 다음 작업을 결정합니다.

측정기는 `tokenmt` 및 `tswtclmt`의 두 측정 모듈로 구성되며, IPQoS 구성 파일에서 구성합니다. 한 클래스에 대해 둘 중 하나의 모듈 또는 둘 다 구성할 수 있습니다.

측정 모듈을 구성할 때 속도에 대해 두 매개변수를 정의할 수 있습니다.

- `committed-rate` - 특정 클래스의 패킷에 대해 수용할 만한 전송 속도(bps, 초당 비트)를 정의합니다.
- `peak-rate` - 특정 클래스의 패킷에 대해 허용되는 최대 전송 속도(bps, 초당 비트)를 정의합니다.

패킷에 대한 측정 작업 결과는 세 가지 결과 중 하나가 될 수 있습니다.

- `green` - 패킷으로 인해 플로우가 약정된 속도 내에서 유지됩니다.
- `yellow` - 패킷으로 인해 플로우가 약정된 속도를 초과하지만 최대 속도를 초과하지는 않습니다.
- `red` - 패킷으로 인해 플로우가 최대 속도를 초과합니다.

IPQoS 구성 파일에서 서로 다른 작업으로 각 결과를 구성할 수 있습니다. 약정된 속도 및 최대 속도는 다음 절에서 설명합니다.

tokenmt 측정기 모듈

`tokenmt` 모듈은 **토큰 버킷**을 사용하여 플로우의 전송 속도를 측정합니다. `tokenmt`가 단일 속도 또는 두 가지 속도 측정기로 작동하도록 구성할 수 있습니다. `tokenmt` 작업 인스턴스는 트래픽 플로우가 구성된 매개변수를 준수하는지 여부를 결정하는 두 토큰 버킷을 유지 관리합니다.

[tokenmt\(7ipp\)](#) 매뉴얼 페이지에서 IPQoS가 토큰 측정기 패러다임을 구현하는 방식을 설명합니다. 토큰 버킷에 대한 일반적인 정보는 Kalevi Kilkki의 *Differentiated Services for the Internet* 및 여러 웹 사이트에서 찾을 수 있습니다.

`tokenmt`의 구성 매개변수는 다음과 같습니다.

- `committed_rate` - 플로우의 약정된 속도 bps(초당 비트)로 지정합니다.
- `committed_burst` - 약정된 버스트 크기를 비트로 지정합니다. `committed_burst` 매개변수는 특정 클래스의 나가는 패킷이 약정된 속도로 네트워크에 전달될 수 있는 크기를 정의합니다.
- `peak_rate` - 최대 속도를 bps(초당 비트)로 지정합니다.

- `peak_burst` - 최대 또는 초과 버스트 크기를 비트로 지정합니다. `peak_burst` 매개변수는 약정된 속도를 초과하는 최대 버스트 크기를 트래픽 클래스에 부여합니다.
- `color_aware` - `tokenmt`에 대한 인식 모드를 설정합니다.
- `color_map` - DSCP 값을 녹색, 노란색 또는 빨간색으로 매핑하는 정수 배열을 정의합니다.

단일 속도 측정기로 `tokenmt` 구성

`tokenmt`를 단일 속도 측정기로 구성하려면 IPQoS 구성 파일에서 `tokenmt`에 대해 `peak_rate` 매개변수를 지정하지 마십시오. 단일 속도 `tokenmt` 인스턴스가 빨간색, 녹색 또는 노란색 결과를 가지도록 구성하려면 `peak_burst` 매개변수를 지정해야 합니다. `peak_burst` 매개변수를 사용하지 않을 경우 `tokenmt`가 빨간색 결과 또는 녹색 결과만 가지도록 구성할 수 있습니다. 두 결과를 가지는 단일 속도 `tokenmt`의 예는 예 3-3을 참조하십시오.

`tokenmt`가 단일 속도 측정기로 작동하는 경우 `peak_burst` 매개변수는 실제로 초과 버스트 크기입니다. `committed_rate` 및 `committed_burst` 또는 `peak_burst`는 0이 아닌 양의 정수여야 합니다.

두 속도 측정기로 `tokenmt` 구성

`tokenmt`를 두 속도 측정기로 구성하려면 IPQoS 구성 파일에서 `tokenmt` 작업에 대해 `peak_rate` 매개변수를 지정합니다. 두 속도 `tokenmt`는 항상 빨간색, 노란색 및 녹색의 세 가지 결과를 가집니다. `committed_rate`, `committed_burst` 및 `peak_burst` 매개변수는 0이 아닌 양의 정수여야 합니다.

색상을 인식하도록 `tokenmt` 구성

두 속도 `tokenmt`가 색상을 인식하도록 구성하려면 “색상 인식”을 구체적으로 추가하는 매개변수를 추가해야 합니다. 다음은 색상을 인식하도록 `tokenmt`를 구성하는 예제 작업 명령문입니다.

예 6-1 IPQoS 구성 파일에 대한 색상 인식 `tokenmt` 작업

```
action {
  module tokenmt
  name meter1
  params {
    committed_rate 4000000
    peak_rate 8000000
    committed_burst 4000000
    peak_burst 8000000
    global_stats true
    red_action_name continue
    yellow_action_name continue
    green_action_name continue
  }
}
```

예 6-1 IPQoS 구성 파일에 대한 색상 인식 tokenmt 작업 (계속)

```

        color_aware true
        color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
    }
}

```

color_aware 매개변수를 true로 설정하여 색상 인식을 사용으로 설정해야 합니다. 색상 인식 측정기로서 tokenmt는 패킷이 이전 tokenmt 작업에 의해 이미 빨간색, 노란색 또는 녹색으로 표시되었다고 간주합니다. 색상 인식 tokenmt는 두 속도 측정기에 대한 매개변수와 함께 패킷 헤더의 DSCP를 사용하여 패킷을 평가합니다.

color_map 매개변수에는 패킷 헤더의 DSCP가 매핑되는 배열이 포함됩니다. 다음 color_map 배열을 고려하십시오.

```
color_map {0-20,22:GREEN;21,23-42:RED;43-63:YELLOW}
```

DSCP 0-20 및 22의 패킷은 녹색으로 매핑됩니다. DSCP 21 및 23-42의 패킷은 빨간색으로 매핑됩니다. DSCP 43-63의 패킷은 노란색으로 매핑됩니다. tokenmt는 기본 색상 맵을 유지 관리합니다. 하지만 color_map 매개변수를 사용하여 필요에 따라 기본값을 변경할 수 있습니다.

color_action_name 매개변수에서 continue를 지정하여 패킷 처리를 완료할 수 있습니다. 또는 패킷을 표시기 작업에 보내는 인수를 추가할 수 있습니다(예:yellow_action_name mark22).

tswtclmt 측정 모듈

tswtclmt 측정 모듈은 시간 기반 **속도 추정기**를 사용하여 트래픽 클래스에 대한 평균 대역폭을 추정합니다. tswtclmt는 항상 세 가지 결과 측정기로 작동합니다. 속도 추정기는 플로우의 도착 추정 속도를 제공합니다. 이 속도는 지정된 기간(**시간 창**) 동안 트래픽 스트림의 실행 평균 대역폭에 근접해야 합니다. 속도 추정 알고리즘은 RFC 2859, *A Time Sliding Window Three Colour Marker*에서 가져옵니다.

다음 매개변수를 사용하여 tswtclmt를 구성합니다.

- committed_rate - 약정된 속도를 bps(초당 비트)로 지정합니다.
- peak_rate - 최대 속도를 bps(초당 비트)로 지정합니다.
- window - 평균 대역폭 내역이 보관되는 시간 창을 밀리초로 정의합니다.

tswtclmt에 대한 기술적인 세부 정보는 [tswtclmt\(7ipp\)](#) 매뉴얼 페이지를 참조하십시오. tswtclmt와 유사한 속도 샤퍼(Shaper)에 대한 일반적인 정보는 [RFC 2963, A Rate Adaptive Shaper for Differentiated Services](#) (<http://www.ietf.org/rfc/rfc2963.txt?number=2963>)를 참조하십시오.

표시기 모듈

IPQoS에는 두 가지 표시기 모듈인 `dscpmk` 및 `dlcosmk`가 포함됩니다. 이 절에서는 두 표시기 사용에 대한 정보가 포함되어 있습니다. `dlcosmk`만 VLAN 장치가 있는 IPQoS 시스템에 대해 사용할 수 있으므로 일반적으로는 `dscpmk`를 사용해야 합니다.

`dscpmk`에 대한 기술적인 정보는 `dscpmk(7ipp)` 매뉴얼 페이지를 참조하십시오. `dlcosmk`에 대한 기술적인 정보는 `dlcosmk(7ipp)` 매뉴얼 페이지를 참조하십시오.

패킷 전달을 위해 `dscpmk` 표시기 사용

표시기는 분류기 또는 측정 모듈이 플로우를 처리한 후 트래픽 플로우를 수신합니다. 표시기는 트래픽을 전달 동작으로 표시합니다. 이 전달 동작은 플로우가 IPQoS 시스템을 떠난 후 수행할 작업입니다. 트래픽 클래스에 대해 수행할 전달 동작은 **PHB(흡별 동작)**에서 정의됩니다. PHB는 다른 트래픽 클래스와 관련하여 해당 클래스의 우선권 플로우를 나타내는 우선 순위를 트래픽 클래스에 지정합니다. PHB는 IPQoS 시스템의 인접 네트워크에 대한 전달 동작만 제어합니다. PHB에 대한 자세한 내용은 **21 페이지 “흡별 동작”**을 참조하십시오.

패킷 전달은 특정 클래스의 트래픽을 네트워크의 다음 대상으로 보내는 프로세스입니다. IPQoS 시스템과 같은 호스트의 경우, 패킷은 호스트에서 로컬 네트워크 스트림으로 전달됩니다. Diffserv 라우터의 경우, 패킷은 로컬 네트워크에서 라우터의 다음 흡으로 전달됩니다.

표시기는 패킷 헤더의 DS 필드를 IPQoS 구성 파일에서 정의된 잘 알려진 전달 동작으로 표시합니다. 그러면 IPQoS 시스템 및 후속 Diffserv 인식 시스템은 표시가 바뀔 때까지 DS 필드에 나타난 대로 트래픽을 전달합니다. PHB를 지정하기 위해 IPQoS 시스템은 패킷 헤더의 DS 필드에 값을 표시합니다. 이 값을 DSCP(차별화 서비스 코드 포인트)라고 합니다. Diffserv 아키텍처는 서로 다른 DSCP를 사용하는 두 가지 유형의 전달 동작인 EF 및 AF를 정의합니다. DSCP에 대한 개요는 **21 페이지 “DS 코드 포인트”**를 참조하십시오.

IPQoS 시스템은 트래픽 플로우에 대해 DSCP를 읽고 다른 송신 트래픽 플로우와 관련하여 플로우의 우선권을 평가합니다. 그런 다음 IPQoS 시스템은 모든 동시 트래픽 플로우에 우선 순위를 지정하고 각 플로우를 우선 순위에 따라 네트워크로 보냅니다.

Diffserv 라우터는 송신 트래픽 플로우를 수신하고 패킷 헤더의 DS 필드를 읽습니다. DSCP는 라우터가 동시 트래픽 플로우에 우선 순위를 지정하고 일정을 예약하도록 합니다. 라우터는 PHB로 지정된 우선 순위에 따라 각 플로우를 전달합니다. 후속 흡의 Diffserv 인식 시스템도 동일한 PHB를 인식하지 못하면 PHB는 네트워크의 경계 라우터를 벗어나서 적용할 수 없습니다.

EF(빠른 전달) PHB

빠른 전달(EF)은 권장 EF 코드 포인트 46(101110)의 패킷이 네트워크로 전송 시 사용 가능한 가장 좋은 취급을 받도록 보장합니다. 빠른 전달은 임대 회선과 비교되기도

합니다. 46(101110) 코드 포인트의 패킷은 패킷의 대상으로 향하는 모든 Diffserv 경로에서 선호 취급이 보장됩니다. EF에 대한 기술적인 정보는 RFC 2598, *An Expedited Forwarding PHB*를 참조하십시오.

AF(보장 전달) PHB

보장 전달(AF)은 표시기에 지정할 수 있는 네 가지 서로 다른 클래스의 전달 동작을 제공합니다. 다음 표는 클래스, 각 클래스에 제공되는 세 가지 삭제 우선권 및 각 우선권과 연결된 권장 DSCP를 보여줍니다. 각 DSCP는 해당 AF 값, 십진수 값 및 이진수 값으로 표시됩니다.

표 6-2 보장 전달 코드 포인트

	클래스 1	클래스 2	클래스 3	클래스 4
낮은 삭제 우선권	AF11 = 10 (001010)	AF21 = 18 (010010)	AF31 = 26 (011010)	AF41 = 34 (100010)
중간 삭제 우선권	AF12 = 12 (001100)	AF22 = 20 (010100)	AF32 = 28 (011100)	AF42 = 36 (100100)
높은 삭제 우선권	AF13 = 14 (001110)	AF23 = 22 (010110)	AF33 = 30 (011110)	AF43 = 38 (100110)

모든 Diffserv 인식 시스템에서는 AF 코드 포인트를 기준으로 사용하여 서로 다른 클래스의 트래픽에 차별화된 전달 동작을 제공할 수 있습니다.

이러한 패킷이 Diffserv 라우터에 도달하면 라우터는 대기열에 있는 다른 트래픽의 DSCP와 함께 패킷의 코드 포인트를 평가합니다. 그런 다음 라우터는 사용 가능한 대역폭 및 패킷의 DSCP로 지정된 우선 순위에 따라 패킷을 전달하거나 삭제합니다. EF PHB로 표시된 패킷은 다양한 AF PHB로 표시된 패킷에 비해 대역폭이 보장됩니다.

패킷이 예상한 대로 전달되도록 하려면 네트워크의 IPQoS 시스템과 Diffserv 라우터 사이에 패킷 표시를 조정하십시오. 예를 들어, 네트워크의 IPQoS 시스템이 AF21(010010), AF13(001110), AF43(100110) 및 EF(101110) 코드 포인트로 패킷을 표시한다고 가정해 보겠습니다. 그러면 AF21, AF13, AF43 및 EF DSCP를 Diffserv 라우터의 해당 파일에 추가해야 합니다.

AF 코드 포인트 표시의 기술적인 설명은 RFC 2597을 참조하십시오. 라우터 제조업체 Cisco Systems 및 Juniper Networks의 회사 웹 사이트에 가면 자세한 AF PHB 설정 정보가 제공됩니다. 이러한 정보를 활용하여 IPQoS 시스템 및 라우터에 대한 AF PHB를 정의할 수 있습니다. 또한 라우터 제조업체의 설명서에는 자사 장비에서 DS 코드 포인트 설정에 대한 지침이 포함되어 있습니다.

표시기에 DSCP 제공

DSCP는 6비트 길이입니다. DS 필드는 1바이트 길이입니다. DSCP를 정의할 때 표시기는 패킷 헤더의 처음 중요 6비트를 DS 코드 포인트로 표시합니다. 나머지 덜 중요한 2비트는 사용되지 않습니다.

DSCP를 정의하려면 표시기 작업 매개변수 내에서 다음 매개변수를 사용합니다.

```
dscp_map{0-63:DS_codepoint}
```

dscp_map 매개변수는 (DSCP) 값으로 채우는 64 요소 배열입니다. dscp_map은 들어오는 DSCP를 dscpmk 표시기에 의해 적용된 나가는 DSCP로 매핑하는 데 사용됩니다.

DSCP 값은 십진수 형식의 dscp_map으로 지정해야 합니다. 예를 들어, EF 코드 포인트 101110은 십진수 값 46으로 변환해야 하며, 결과적으로 dscp_map{0-63:46}이 됩니다. AF 코드 포인트의 경우, 표 6-2에 나온 다양한 코드 포인트를 dscp_map에서 사용할 십진수 형식으로 변환해야 합니다.

VLAN 장치에서 dlcosmk 표시기 사용

dlcosmk 표시기 모듈은 데이터그램의 MAC 헤더에서 전달 동작을 표시합니다. VLAN 인터페이스가 있는 IPQoS 시스템에서만 dlcosmk를 사용할 수 있습니다.

dlcosmk는 VLAN 태그로 알려진 4바이트를 MAC 헤더에 추가합니다. VLAN 태그에는 IEEE 801.D 표준에서 정의된 3비트 사용자 우선 순위 값이 포함됩니다. VLAN을 이해하는 Diffserv 인식 스위치는 데이터그램의 사용자 우선 순위 필드를 읽을 수 있습니다. 801.D 사용자 우선 순위 값은 잘 알려지고 상용 스위치에서 이해할 수 있는 CoS(서비스 클래스) 표시를 구현합니다.

다음 표에 나열된 서비스 클래스 표시를 정의하여 dlcosmk 표시기 작업에서 사용자 우선 순위 값을 사용할 수 있습니다.

표 6-3 801.D 사용자 우선 순위 값

서비스 클래스	정의
0	최선 조건
1	백그라운드
2	여분
3	최우선 조건
4	제어 로드
5	100ms 대기 시간 미만의 비디오
6	10ms 대기 시간 미만의 비디오
7	네트워크 제어

dlcosmk에 대한 자세한 내용은 [dlcosmk\(7ipp\)](#) 매뉴얼 페이지를 참조하십시오.

VLAN 장치가 있는 시스템에 대한 IPQoS 구성

이 절에서는 VLAN 장치가 있는 시스템에서 IPQoS를 구현하는 방법을 보여주는 단순한 네트워크 시나리오를 소개합니다. 시나리오에는 스위치로 연결된 machine1 및 machine2의 두 IPQoS 시스템이 포함됩니다. machine1의 VLAN 장치는 IP 주소 10.10.8.1을 가집니다. machine2의 VLAN 장치는 IP 주소 10.10.8.3을 가집니다.

machine1에 대한 다음 IPQoS 구성 파일은 스위치를 거쳐 machine2로 이동하는 트래픽을 표시하기 위한 간단한 솔루션을 보여줍니다.

예 6-2 VLAN 장치가 있는 시스템에 대한 IPQoS 구성 파일

```
fmt_version 1.0
action {
    module ipgpc
        name ipgpc.classify

    filter {
        name myfilter2
        daddr 10.10.8.3
        class myclass
    }

    class {
        name myclass
        next_action mark4
    }
}

action {
    name mark4
    module dlcsmk
    params {
        cos 4
        next_action continue
    }
    global_stats true
}
```

이 구성에서 machine2의 VLAN 장치를 대상으로 하는 machine1의 모든 트래픽은 dlcsmk 표시기로 전달됩니다. mark4 표시기 작업은 dlcsmk가 VLAN 표시를 CoS가 4인 myclass 클래스의 데이터그램에 추가하도록 지시합니다. 사용자 우선 순위 값 4는 두 시스템 사이에 있는 스위치가 machine1의 myclass 트래픽 플로우에 제어 로드 전달을 제공해야 한다는 것을 나타냅니다.

flowacct 모듈

IPQoS flowacct 모듈은 트래픽 플로우에 대한 정보를 기록하며, 이 프로세스를 **플로우 정산**이라고 합니다. 플로우 계산은 고객 청구 또는 특정 클래스에 대한 트래픽의 양 평가 목적으로 사용될 수 있는 데이터를 생성합니다.

플로우 계산은 선택 사항입니다. flowacct는 일반적으로 측정되거나 표시된 트래픽 플로우가 네트워크 스트림으로 보내지기 전에 만날 수 있는 마지막 모듈입니다. Diffserv 모델에서 flowacct의 위치에 대한 그림은 [그림 1-1](#)을 참조하십시오. flowacct에 대한 기술적인 세부 정보는 flowacct(7ipp) 매뉴얼 페이지를 참조하십시오.

플로우 정산을 사용으로 설정하려면 flowacct와 함께 Oracle Solaris exacct 정산 기능 및 acctadm 명령을 사용해야 합니다. 플로우 계산 설정에 대한 전체 단계는 [79 페이지 “플로우 계산 설정\(작업 맵\)”](#)을 참조하십시오.

flowacct 매개변수

flowacct 모듈은 **플로우 테이블**에서 **플로우 레코드**로 구성된 플로우에 대한 정보를 수집합니다. 테이블의 각 항목은 하나의 플로우 레코드를 포함합니다. 플로우 테이블을 표시할 수는 없습니다.

IPQoS 구성 파일에서 다음 flowacct 매개변수를 정의하여 플로우 레코드를 측정하고 레코드를 플로우 테이블에 기록합니다.

- **timer** - 시간 초과된 플로우가 플로우 테이블에서 제거되고 acctadm으로 만들어진 파일에 기록되는 간격을 밀리초로 정의합니다.
- **timeout** - 플로우가 시간 초과되기 전에 패킷 플로우가 비활성화되어야 하는 기간을 밀리초로 정의합니다.

주 - timer 및 timeout이 서로 다른 값을 가지도록 구성할 수 있습니다.

- **max_limit** - 플로우 테이블에 저장할 수 있는 플로우 레코드 수에 대한 상한 제한을 둡니다.

flowacct 매개변수가 IPQoS 구성 파일에서 사용되는 예는 [66 페이지 “IPQoS 구성 파일에서 플로우 제어를 구성하는 방법”](#)을 참조하십시오.

플로우 테이블

flowacct 모듈은 flowacct 인스턴스에서 확인되는 모든 패킷 플로우를 기록하는 플로우 테이블을 유지 관리합니다.

플로우는 flowacct 8-튜플을 포함하는 다음 매개변수로 식별됩니다.

- 소스 주소
- 대상 주소
- 소스 포트
- 대상 포트
- DSCP
- 사용자 ID
- 프로젝트 ID
- 프로토콜 번호

플로우에 대한 8-튜플의 모든 매개변수가 동일하게 유지될 경우 플로우 테이블은 하나의 항목만 포함합니다. max_limit 매개변수는 플로우 테이블에 포함될 수 있는 항목 수를 결정합니다.

플로우 테이블은 timer 매개변수에 대해 IPQoS 구성 파일에서 정의된 간격으로 검사됩니다. 기본값은 15초입니다. 플로우는 IPQoS 구성 파일의 timeout 간격 이상 동안 IPQoS 시스템에서 해당 패킷을 볼 수 없을 때 “시간 초과”됩니다. 기본 시간 초과 간격은 60초입니다. 그런 다음 시간 초과된 항목은 acctadm 명령으로 만들어진 계산 파일에 기록됩니다.

flowacct 레코드

flowacct 레코드에는 다음 표에 설명된 속성이 포함됩니다.

표 6-4 flowacct 레코드의 속성

속성 이름	속성 내용	유형
src-addr-address-type	발신자의 소스 주소. address-type은 IPQoS 구성 파일에 지정된 대로 IPv4의 경우 v4 또는 IPv6의 경우 v6입니다.	기본
dest-addr-address-type	패킷에 대한 대상 주소. address-type은 IPQoS 구성 파일에 지정된 대로 IPv4의 경우 v4 또는 IPv6의 경우 v6입니다.	기본
src-port	플로우가 발생한 소스 포트	기본
dest-port	이 플로우가 향하는 대상 포트 번호	기본
protocol	플로우에 대한 프로토콜 번호	기본
total-packets	플로우의 패킷 수	기본
total-bytes	플로우의 바이트 수	기본
action-name	이 플로우를 기록한 flowacct 작업의 이름	기본
creation-time	플로우에 대한 패킷이 flowacct에 의해 처음으로 목격된 시간	확장 전용
last-seen	플로우의 패킷이 마지막으로 목격된 시간	확장 전용

표 6-4 flowacct 레코드의 속성 (계속)

속성 이름	속성 내용	유형
diffserv-field	플로우의 나가는 패킷 헤더에 있는 DSCP.	확장 전용
user	응용 프로그램에서 가져온 UNIX 사용자 ID 또는 사용자 이름.	확장 전용
projid	응용 프로그램에서 가져온 프로젝트 ID.	확장 전용

flowacct 모듈에서 acctadm 사용

acctadm 명령을 사용하여 flowacct로 생성된 다양한 플로우 레코드를 저장할 파일을 만듭니다. acctadm은 확장 정산 기능과 함께 작동합니다. acctadm에 대한 기술적인 정보는 [acctadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

flowacct 모듈은 플로우를 관찰하고 플로우 테이블을 레코드로 채웁니다. 그런 다음 flowacct는 timer에서 지정된 간격으로 해당 매개변수 및 속성을 평가합니다. 패킷이 last_seen + timeout 값 이상 동안 보이지 않으면 패킷은 시간 초과됩니다. 모든 시간 초과된 항목은 플로우 테이블에서 삭제됩니다. 그런 다음 이러한 항목은 timer 매개변수에 지정된 간격이 경과할 때마다 계산 파일에 기록됩니다.

flowacct 모듈에서 사용할 acctadm을 호출하려면 다음 구문을 사용합니다.

```
acctadm -e file-type -f filename flow
```

acctadm -e -e 옵션과 함께 acctadm을 호출합니다. -e는 리소스 목록이 있음을 나타냅니다.

file-type 수집할 속성을 지정합니다. file-type은 basic 또는 extended로 바뀌어야 합니다. 각 파일 유형의 속성 목록은 [표 6-4](#)를 참조하십시오.

-f file-name 플로우 레코드를 보관할 file-name 파일을 만듭니다.

flow acctadm이 IPQoS에서 실행됨을 나타냅니다.

IPQoS 구성 파일

이 절에서는 IPQoS 구성 파일의 부분에 대한 전체 세부 정보가 포함되어 있습니다. IPQoS 부트 시 활성화되는 정책은 /etc/inet/ipqosinit.conf 파일에 저장됩니다. 이 파일을 편집할 수 있지만 새 IPQoS 시스템의 경우 가장 좋은 방법은 다른 이름으로 구성 파일을 만드는 것입니다. IPQoS 구성을 적용하고 디버깅하는 작업은 [3 장, “IPQoS 구성 파일 만들기\(작업\)”](#)를 참조하십시오.

IPQoS 구성 파일의 구문은 [예 6-3](#)을 참조하십시오.

예에서는 다음 규약을 사용합니다.

- **컴퓨터 스타일 유형** - 구성 파일의 부분을 설명하기 위해 제공되는 구문 정보입니다. 컴퓨터 스타일 유형으로 나타나는 텍스트는 입력하지 않습니다.
- **굵은체 유형** - IPQoS 구성 파일에 입력해야 하는 리터럴 텍스트입니다. 예를 들어, IPQoS 구성 파일은 항상 **fmt_version**으로 시작해야 합니다.
- **기울임꼴 유형** - 구성에 대한 설명 정보로 바꾸는 변수 텍스트입니다. 예를 들어, *action-name* 또는 *module-name*은 항상 조직에 해당하는 정보로 바꾸어야 합니다.

예 6-3 IPQoS 구성 파일의 구문

```
file_format_version ::= fmt_version version

action_clause ::= action {
    name action-name
    module module-name
    params-clause | ""
    cf-clauses
}
action_name ::= string
module_name ::= ipgpc | dlcosmk | dscpmk | tswtclmt | tokenmt | flowacct

params_clause ::= params {
    parameters
    params-stats | ""
}
parameters ::= prm-name-value parameters | ""
prm_name_value ::= param-name param-value

params_stats ::= global-stats boolean

cf_clauses ::= class-clause cf-clauses |
             filter-clause cf-clauses | ""

class_clause ::= class {
    name class-name
    next_action next-action-name
    class-stats | ""
}
class_name ::= string
next_action_name ::= string
class_stats ::= enable stats boolean
boolean ::= TRUE | FALSE

filter_clause ::= filter {
    name filter-name
    class class-name
    parameters
}
filter_name ::= string
```

IPQoS 구성 파일의 각 주요 부분을 설명하는 나머지 텍스트입니다.

action 명령문

action 명령문을 사용하여 85 페이지 “IPQoS 아키텍처 및 Diffserv 모델”에 설명된 다양한 IPQoS 모듈을 호출합니다.

IPQoS 구성 파일을 만들 때는 항상 버전 번호로 시작해야 합니다. 그리고 다음 action 명령문을 추가하여 분류기를 호출합니다.

```
fmt_version 1.0

action {
    module ipgpc
    name ipgpc.classify
}
```

분류기 action 명령문 다음에는 params 절 또는 class 절이 옵니다.

기타 모든 action 명령문에 대해 다음 구문을 사용합니다.

```
action {
    name action-name
    module module-name
    params-clause | ""
    cf-clauses
}
```

name action_name 작업에 이름을 지정합니다.

module module_name 호출할 IPQoS 모듈을 식별합니다. 표 6-5의 모듈 중 하나이어야 합니다.

params_clause 분류기가 처리할 매개변수가 될 수 있습니다(예: 전역 통계 또는 처리할 다음 작업).

cf_clauses 0개 이상의 class 절 또는 filter 절 세트입니다.

모듈 정의

모듈 정의는 action 명령문에서 매개변수를 처리할 모듈을 나타냅니다. IPQoS 구성 파일에는 다음 모듈이 포함될 수 있습니다.

표 6-5 IPQoS 모듈

모듈 이름	정의
ipgpc	IP 분류기
dscpmk	IP 패킷에서 DSCP를 만드는 데 사용할 표시기

표 6-5 IPQoS 모듈 (계속)

모듈 이름	정의
dlcosmk	VLAN 장치에서 사용할 표시기
tokenmt	토큰 버킷 측정기
tswtclmt	시간별 창 측정기
flowacct	플로우 계산 모듈

class 절

각 트래픽 클래스에 대해 class 절을 정의합니다.

이 구문을 사용하여 IPQoS 구성의 나머지 절을 정의합니다.

```
class {
    name class-name
    next_action next-action-name
}
```

특정 클래스에 대한 통계 수집을 사용으로 설정하려면 먼저 `ipgpc.classify action` 명령문에서 전역 통계를 사용으로 설정해야 합니다. 자세한 내용은 99 페이지 “[action 명령문](#)”을 참조하십시오.

클래스에 대한 통계 수집을 설정할 때는 항상 `enable_stats TRUE` 명령문을 사용합니다. 클래스에 대한 통계를 수집할 필요가 없는 경우 `enable_stats FALSE`를 지정할 수 있습니다. 또는 `enable_stats` 명령문을 제거할 수 있습니다.

명시적으로 정의하지 않은 IPQoS 사용 네트워크에 대한 트래픽은 **기본 클래스**로 들어갑니다.

filter 절

필터는 트래픽 플로우를 클래스로 그룹화하는 선택기로 구성됩니다. 이러한 선택기는 class 절에서 만들어진 클래스의 트래픽에 적용될 조건을 구체적으로 정의합니다. 패킷이 가장 높은 우선 순위 필터의 모든 선택기와 일치할 경우 해당 패킷은 필터 클래스의 멤버로 간주됩니다. ipgpc 분류기에서 사용할 수 있는 전체 선택기 목록은 [표 6-1](#)을 참조하십시오.

다음 구문을 가지는 **filter 절**을 사용하여 IPQoS 구성 파일에서 필터를 정의합니다.

```
filter {
    name filter-name
    class class-name
    parameters (selectors)
}
```

params 절

params 절에는 작업 명령문에서 정의된 모듈에 대한 처리 지침이 포함됩니다. params 절에 대해 다음 구문을 사용합니다.

```
params {
    parameters
    params-stats | ""
}
```

params 절에서 모듈에 적용 가능한 매개변수를 사용합니다.

params 절의 *params-stats* 값은 `global_stats TRUE` 또는 `global_stats FALSE`입니다. `global_stats TRUE` 지침은 전역 통계가 호출되는 action 명령문에 대해 UNIX 스타일 통계를 설정합니다. 통계는 `kstat` 명령을 사용하여 볼 수 있습니다. 클래스별 통계를 사용으로 설정하려면 먼저 action 명령문 통계를 사용으로 설정해야 합니다.

ipqosconf 구성 유틸리티

ipqosconf 유틸리티를 사용하여 IPQoS 구성 파일을 읽고 UNIX 커널에서 IPQoS 모듈을 구성합니다. ipqosconf는 다음 작업을 수행합니다.

- 구성 파일을 IPQoS 커널 모듈에 적용합니다(ipqosconf -a filename).
- 커널에 현재 상주하는 IPQoS 구성 파일을 나열합니다(ipqosconf -l).
- 시스템이 재부트될 때마다 현재 IPQoS 구성을 읽고 적용되도록 합니다(ipqosconf -c).
- 현재 IPQoS 커널 모듈을 비웁니다(ipqosconf -f).

기술적인 정보는 [ipqosconf\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

색인

A

acctadm 명령, 플로우 정산, 97
acctadm 명령, 플로우 정산용, 81
action 명령문, 99
AF(보장 전달), 22, 92
 AF 코드 포인트 표, 92
 표시기 action 명령문, 54

C

class 절, IPQoS 구성 파일, 51
class 절, IPQoS 구성 파일, 100
CoS(서비스 클래스) 표시, 19

D

Diffserv 모델
 IPQoS 구현, 16-20, 19
 분류기 모듈, 17-18
 측정기 모듈, 18
 표시기 모듈, 18-19
 플로우 예, 19
Diffserv 인식 라우터
 DS 코드 포인트 평가, 92
 계획, 31
dlcosmk 표시기, 19
 VLAN 태그, 93
 사용자 우선 순위 값, 표, 93
dlcosmk 표시자, 데이터그램 전달 계획, 38
DS 코드점(DSCP), 계획, QoS 정책에서, 38

DSCP(DS 코드 포인트), 19, 21
 AF 전달 코드 포인트, 22, 92
 dscp_map 매개변수, 93
 EF 전달 코드 포인트, 22, 91
 PHB 및 DSCP, 21
 구성, diffserv 라우터, 69, 91
 색상 인식 구성, 90
 정의, IPQoS 구성 파일, 54
dscpmk 표시기, 19
 패킷 전달 계획, 38
 패킷 전달을 위한 PHB, 91-93
 호출, 표시기 action 명령문, 54, 59, 65, 67

E

EF(빠른 전달), 22, 91
 정의, IPQoS 구성 파일, 55

F

filter 절, IPQoS 구성 파일, 52, 100
flowacct 모듈, 19, 95
 acctadm 명령, 플로우 정산 파일 만들기, 97
 flowacct에 대한 action 명령문, 56
 매개변수, 95
 플로우 레코드, 80
 플로우 레코드 테이블, 95-96
 플로우 레코드의 속성, 96

I

ipgpc 분류기, “분류기 모듈” 참조
 IPQoS, 11

- Diffserv 모델 구현, 16–20
- IPQoS 네트워크의 라우터, 69
- QoS 정책 계획, 29
- VLAN 장치 지원, 93–94
- 관련 RFC, 13
- 구성 계획, 25
- 구성 예, 41–43
- 구성 파일, 47, 97
 - action 명령문 구문, 99
 - class 절, 51
 - filter 절, 52
 - IPQoS 모듈 목록, 99
 - 구문, 98
 - 초기 action 명령문, 99
 - 초기 작업 명령문, 50
 - 표시기 action 명령문, 54
- 기능, 12
- 네트워크 예, 47
- 매뉴얼 페이지, 13
- 메시지 로깅, 73
- 오류 메시지, 74
- 지원되는 네트워크 토폴로지, 26, 27, 28
- 통계 생성, 82
- 트래픽 관리 기능, 15, 16

 IPQoS 네트워크의 가상 LAN(VLAN) 장치, 93–94
 IPQoS 지원 네트워크용 하드웨어, 26
 ipqosconf, 47
 ipqosconf 명령

- 구성 적용, 72, 73
- 명령 옵션, 101
- 현재 구성 나열, 73

 IPQoS에 대한 syslog.conf 파일 로깅, 73
 IPQoS에 대한 네트워크 예, 47
 IPQoS에 대한 네트워크 토폴로지

- IPQoS 사용 방화벽의 LAN, 28
- IPQoS 사용 호스트의 LAN, 27
- 구성 예, 41

 IPQoS에 대한 오류 메시지, 74
 IPQoS에 대한 통계

- 수집, kstat 명령 사용, 82
- 전역 통계 사용, 50, 100

IPQoS에 대한 통계 (계속)

클래스 기반 통계 사용, 100
 IPQoS용 네트워크 토폴로지, 26
 IPQoS 지원 서버 팜 LAN, 27

K

kstat 명령, IPQoS에서 사용, 82

P

params 절

- action 측정, 67
- flowacct action, 56
- 구문, 101
- 전역 통계 정의, 50, 101
- 표시기 action, 54

 PHB(홉별 동작), 21

- AF 전달, 22
- EF 전달, 22
- 사용, dscpmk 표시기 사용, 91–93
- 정의, IPQoS 구성 파일, 68

Q

QoS(서비스 품질)

- QoS 정책, 14–15
- 작업, 12

 QoS 정책, 14

- 계획 작업 맵, 30
- 구현, IPQoS 구성 파일, 45
- 정책 구성 템플릿, 29
- 필터 만들기, 33

R

RFC(Requests for Comments), IPQoS, 13

S

- SLA(서비스 단계 계약), 14
 - 서로 다른 서비스 클래스 우선 순위 지정, 16
 - 서비스 클래스, 17
 - 청구 클라이언트, 플로우 정산 기반, 80

T

- tokenmt 측정기, 18
 - 단일 속도 측정기, 89
 - 두 속도 측정기, 89
 - 색상 인식 구성, 18, 89
 - 속도 매개변수, 88
 - 측정 속도, 88-90
- tswtclmt 측정기, 18, 90
 - 측정 속도, 90

대

- 대역폭 규제, 15
 - 계획, QoS 정책에서, 32

로

- 로드 균형 조정, IPQoS 사용 네트워크에서, 27

분

- 분류기 모듈, 17-18
 - action 명령문, 50
 - 분류기의 기능, 86

사

- 사용자 우선 순위 값, 19

색

- 색상 인식, 18, 89

서

- 서비스 클래스, “클래스”참조

선

- 선택기, 17-18
 - IPQoS 5-튜플, 17
 - 계획, QoS 정책에서, 33
 - 선택기, 목록, 86

애

- 애플리케이션 서버, IPQoS에 대한 구성, 60

예

- 예제 IPQoS 구성 파일
 - VLAN 장치 구성, 94
 - 색상 인식 세그먼트, 89
 - 애플리케이션 서버, 60
 - 최선 조건 웹 서버, 49
 - 프리미엄 웹 서버, 47

웹

- 웹 서버
 - IPQoS에 대한 구성, 47, 49, 57, 58

작

- 작업 맵
 - IPQoS
 - QoS 정책 계획, 30
 - 구성 계획, 25
 - 구성 파일 만들기, 45
 - 플로우 계산 설정, 79

- 차**
 차등화 서비스, 11-12
 차등화 서비스 모델, 16-20
 차별화 서비스, 서로 다른 서비스 클래스 제공, 16
 차별화된 서비스, 네트워크 토폴로지, 26
- 측**
 측정 모듈
 “tokenmt 측정기”참조
 “tswtclmt 측정기”참조
 소개, 18
 측정 결과, 18, 88
 호출, IPQoS 구성 파일, 67
- 클**
 클래스, 17
 class 절의 구문, 100
 선택기, 목록, 86
 정의, IPQoS 구성 파일, 58, 62
- 트**
 트래픽 관리
 네트워크 토폴로지 계획, 27
 대역폭 규제, 15
 제어 플로우, 18
 트래픽 전달, 21, 22, 23
 트래픽 플로우 우선순위 지정, 15-16
 트래픽 전달
 Diffserv 네트워크를 통한 트래픽 플로우, 22
 IP 패킷 전달, DSCP 사용, 21
 계획, QoS 정책에서, 32
 데이터그램 전달, 93-94
 패킷 전달에 대한 PHB의 효과, 91-93
 트래픽 준수
 결과, 18, 88
 계획
 QoS 정책 결과, 36
 QoS 정책의 속도, 36
 속도 매개변수, 88
- 트래픽 준수 (계속)
 정의, 67
- 표**
 표시기 모듈, 18-19
 “dlcosmk 표시기”참조
 “dscpmk 표시기”참조
 DS 코드 포인트 지정, 93
 PHB, IP 패킷 전달, 21
 VLAN 장치 지원, 93-94
- 플**
 플로우 정산, 80-82, 95
 플로우 레코드 테이블, 95-96
 플로우 정산에 대한 acctadm 명령, 19
 플로우 제어, 측정 모듈을 통해, 18
- 필**
 필터, 17-18
 filter 절 구문, 100
 계획, QoS 정책, 33
 만들기, IPQoS 구성 파일, 58, 63
 선택기, 목록, 86