

대역 브로커를 통한 동적 자원 관리를 지원하는 차별화 서비스 라우터 구현

최영수¹⁰, 박기현^{*}, 이성협^{*}, 조유제^{*}, 채희성^{**}, 한태만^{**}

¹⁰경북대학교, ^{*}한국전자통신연구원

{yschoi¹⁰, khpark, shlee}@m80.knu.ac.kr, {yzcho}@ee.knu.ac.kr, {hschae, tmhan}@etri.re.kr

Implementation of a DiffServ Router for Dynamic Resource Management using Bandwidth Broker

Young-Soo Choi¹⁰, Ki-Hyun Park^{*}, Sung-Hyup Lee^{*}, You-Ze Cho^{*}, Hee-Sung Chae^{**}, Tae-Man Han^{**}

^{*}School of Electronic and Electrical Engineering, Kyungpook National University

^{**}ETRI

요 약

본 논문에서는 PC 기반 FreeBSD 라우터에 대역 브로커를 통한 동적 자원 관리를 지원할 수 있도록 COPS를 신호 프로토콜로 사용하는 차별화 서비스 지원 라우터를 구현 하였다. 구현을 위해 ASN.1 표기법으로 PIB를 작성하고 이를 위한 인코딩/디코딩 모듈을 구현하였다. 또한, 라우터의 자원 제어를 수행하는 QoS(QoS Daemon)와 COPS(Common Open Policy Service) 클라이언트를 구현하였다. 그리고 QoS와 COPS 클라이언트 간의 통신 프로토콜을 정의하고 구현하였다. 또한, COPS 메시지 단위의 룰백을 위하여 라우터에 룰백 모듈을 구현하였고, 자원 관리 및 룰백 시 필요로 하는 인터페이스, 클래스, 필터, 컨디셔너의 추가 및 삭제 기능을 구현 하였다. 그리고 테스트 베드를 구축하여 구현된 라우터의 동작 검증을 수행하였다.

1. 서론

차별화 서비스는 인터넷 상에서 QoS 보장을 위해 제안된 서비스 모델이다. 이제까지 미터링, 마킹, 스케줄링과 같은 QoS 보장 및 차별화 서비스를 위한 요소 기술에 관한 많은 연구가 이루어 졌다. 하지만 사용자나 인터넷 서비스 제공자에게 실질적으로 의미 있는 종단간 QoS 보장에 관한 연구는 아직 미비한 실정이다. 종단간 QoS 보장을 위해서는 다양한 QoS 보장 기술을 제공하는 기기들로 구성된 이질적인 도메인간의 연동 기술이 요구되며, 망 상황에 유연하게 대처할 수 있는 대역 브로커 및 대역 브로커와의 통신을 통한 동적 자원 제어를 지원하는 차별화 서비스 라우터 개발이 중요하다. 본 논문에서는 COPS를 신호 프로토콜로 이용하는 차별화 서비스 지원을 위한 라우터를 구현하였으며, 이를 위해 PIB 작성 및 인코딩/디코딩 모듈 구현, COPS 클라이언트 구현, 라우터 자원 제어 및 룰백 루틴을 수행하는 QoS를 구현하였다.

본 논문의 구성은 2장에서는 차별화 서비스 모델 및 대역 브로커 개념에 대해서 기술하고 3장에서는 차별화 서비스 지원을 위해 구현된 라우터 주요 모듈 구조 및 기능을 설명한다. 4장에서는 라우터 동작 검증을 위한 테스트 베드 구조와 실험 환경 및 결과에 대해서 분석하고 5장에서는 결론을 맺는다.

2. 인터넷 차별화 서비스 및 대역 브로커

차별화 서비스는 복잡한 트래픽 조절 기능 대부분을 망의 경계에 두고, 망 코어에서는 간단한 패킷 전달 기능만을 수행하여, PHB(Per Hop Behavior)로 대변되는 차별화 체계를 구성하여 대규모 백본 망에서도 적용될 수 있는 확장성을 가진다[1]. 차별화 서비스 상에서 종단간 QoS 보장을 위해서는 망 상황에 따라 자원을 동적으로 관리하는 방식이 요구되며 이에 대한 연구가 최근 이루어 지고 있다. 가입자 혹은 가입자 망과 인터넷 서비스 제공자 간에는 SLA(Service Level Agreement)를 체결하고 망 관리자는 SLA에 명시된 내용에 따라 사용자의 트래픽의 수락 제어 및 자원 할당 과정을 수행한다. 대역 브로커는 정책 서버를 포함하는 개념으로, 위와 같은 사용자 자원 예약 요청에 대한 수락 제어 및 관리하는 망 내의 자원 관리를 담당한다[2]. 또한 대역 브로커는 다른 도메인을 담당하는 대역 브로커들과의 통신을 통해 이질적 도메인 상에서 종단간 QoS 보장을 위한 중요한

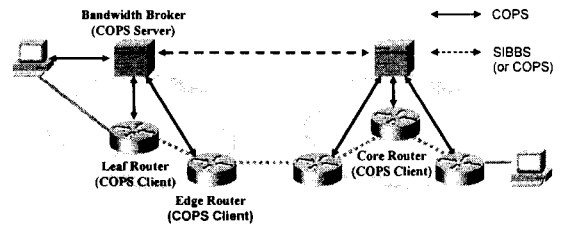


그림 1. 차별화 서비스 및 대역 브로커 모델

역할을 한다. 그림 1은 차별화 서비스 상에서 대역 브로커를 사용한 종단간 QoS 보장 모델의 예를 보여준다. 차별화 서비스 라우터와 대역 브로커 사이에는 다양한 신호 프로토콜이 사용될 수 있으나, COPS를 고려하는 추세이며, 대역 브로커간 신호 프로토콜로는 일반적으로 QBone의 SIBBS(Simple Interdomain Bandwidth Broker Signaling)나 COPS가 고려되고 있다[2].

3. 차별화 서비스 지원 라우터 구현

현재 차별화 서비스 라우터는 Linux, FreeBSD 등의 다양한 플랫폼에서 구현되어 있다. 하지만 대부분의 구현물은 미리 작성된 설정 파일, 텔넷 혹은 커맨드 라인을 통한 사용자의 명령 입력과 같이 정적인 자원 제어만을 지원하거나 서로 다른 신호 프로토콜을 사용하는 한계를 가진다.

본 논문에서는 신호 프로토콜로 COPS를 사용하여 라우터의 동적 자원 제어가 가능하도록 기존 FreeBSD의 구현물 수정을 통한 차별화 서비스 라우터를 개발하고, 정책 서버 혹은 대역 브로커와 정책 정보를 교환하는 COPS-PR(COPS Usage for Policy Provisioning) 클라이언트를 구현하였다. COPS-PR 클라이언트는 Vovida에서 공개 소스로 제공되는 COPS 스택을 사용하여 구현하였고, 라우터 자원 제어와 룰백을 수행하는 QoS는 ALTQ를 사용하여 구현하였다. QoS와 COPS 클라이언트 간에는 유닉스 도메인 소켓을 이용한 간단한 프로토콜을 설계하고 구현하여 대역 브로커를 통한 자원 관리가 가능하도록 하였다. 구현한 라우터의 구현 규칙과 개발 도구는 표 1과 같다.

표 1. 구현된 라우터 규격

플랫폼 타입	PC 기반 라우터
OS	FreeBSD 4.6 Release
혼잡관리방식	RED, RIO
트래픽 분류 기준	IP 주소, Port 번호, 프로토콜 필드 값
스케줄링 기법	FIFO, PRIQ, CBQ, etc
구현 상 참고 문서	QBone Bandwidth Broke[2], ALTQ 3.1[3], Vovida COPS 1.4.0[4], Snacc 1.3[5]
대역 통신 프로토콜	COPS

구현한 차별화 서비스 지원 라우터의 프로토콜 스택은 그림 2와 같다. 그림에서 음영으로 표시된 COPS 클라이언트, QoSD, parser/interpreter, qcmd, qop 계층은 본 논문에서 구현한 부분을 나타낸다. 각 구현된 부분의 주요 기능은 다음과 같다. 먼저, QoSD는 기존의 ALTQ를 확장하여 COPS 클라이언트가 유닉스 도메인 소켓을 통하여 자원 제어가 가능하도록 하였다. 또한 QoSD는 COPS 메시지 단위의 롤백을 지원하기 위한 롤백 관리 및 이에 필요한 명령 생성 및 파싱 루틴을 구현하였다. 그리고 기존의 ALTQ에서는 지원되지 않던 인터페이스 삭제, 클래스, 필터의 추가 삭제 등을 지원하도록 ALTQ의 파싱 계층, qcmd 계층, qop 계층의 API 및 관련 함수를 구현하였다.

3.1 QoSD의 설계 및 구현

QoSD는 그림 3과 같이 COPS 클라이언트 및 QUIP 연결 관리 루틴, COPS 메시지 파싱 및 관리 루틴, 롤백을 위한 명령어 파싱 및 생성 루틴 등으로 구성되어 있다. 먼저 초기화 루틴은 qosd 구조 시 사용되는 옵션 등을 분석하여 디버깅 레벨, COPS 및 QUIP를 위한 소켓 생성, 명령행 테이블 및 롤백 테이블을 초기화한다. COPS 클라이언트 연결 관리 루틴은 유닉스 도메인 소켓의 생성, 제거를 담당하며, COPS 메시지 관리 루틴은 COPS에서 받은 전달된 메시지를 파싱하여 QoSD 제어 루틴으로 넘겨주고 롤백을 위해 이를 명령행 테이블 저장한다. QoSD 제어 루틴은 COPS 메시지 관리 루틴으로부터 전달된 메시지를 파싱 계층으로 넘겨주고 성공적 처리 여부를 판단한다. 이때, 잘못된 명령이나 등이 수행된 경우 롤백 처리 루틴을 호출한다. 롤백 처리 루틴은 명령행 테이블에 저장된 명령을 파싱하고 롤백을 위한 명령어를 생성하여 이를 롤백 테이블에 저장하고 QoSD 제어 루틴으로 롤백 명령 테이블 값을 넘겨준다. QoSD 제어 루틴은 롤백 테이블의 명령을 읽어 롤백을 수행한다.

3.2 COPS-PR 클라이언트 설계 및 구현

Vovida에서 제공하는 PEP 라이브러리는 VOCAL의 지원을 위해 새롭게 정의된 클라이언트 타입으로 구현되었다. 즉, Vovida에서는 SIP 클라이언트의 호 요청이 있을 때마다 COPS REQ 메시지를 생성하여 PDP에게 전송하여 DEC 메시지를

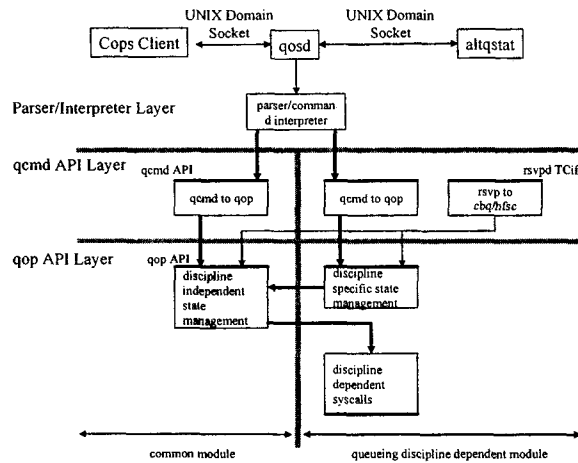


그림 2. 구현 라우터 프로토콜 스택 구조

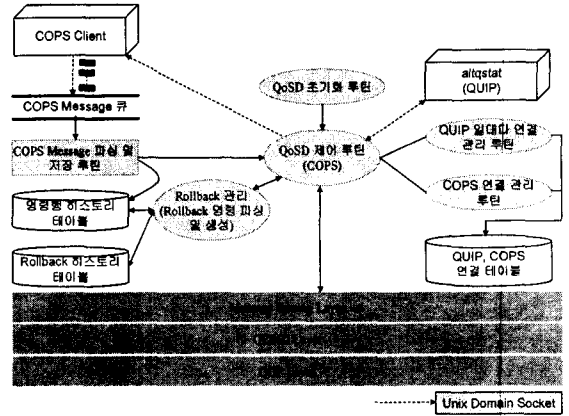


그림 3. QoSD 구현 구조

PDP로부터 수신하고, 호가 끝나면 비로소 RPT 메시지를 PDP에게 전송하도록 구현하였다. 이러한 구조는 IETF에서 정의한 COPS-PR과 차이점을 보인다. 따라서, 본 논문에서는 새롭게 COPS-PR 클라이언트 타입에 대한 PEP 라이브러리를 구현하였다.

그림 4는 본 논문에서 구현한 PEP 라이브러리 구조를 나타낸다. COPSPrPEP는 가장 핵심이 되는 클래스로, PEP의 모든 동작은 COPSPrPEP를 통해 이루어 진다. COPSPrCfg 클래스는 정책서버 주소와 같은 환경 설정 정보를 관리하며, COPSPrReq 클래스는 요청 상태 정보를 관리한다. REQ 메시지의 경우 라우터의 인터페이스 정보를 필요로 하는데, 이러한 정보는 ifstatus 라이브러리를 통해 얻는다. 서버로부터 얻은 정책 정보는 GenCmdString 클래스를 통해 ALTQ 명령어로 변환되고, PolicyProvision은 유닉스 도메인 소켓을 이용하여 QoSD에 정보 전달을 한다. AltqPIB 라이브러리는 ALTQ 명령어를 바탕으로 작성한 PIB를 인코딩 및 디코딩하는 역할을 담당하는데, 이 라이브러리는 ASN.1 표기법으로 작성된 PIB 바탕으로 ASN.1 컴파일러인 snacc을 사용하여 생성하였다. Timer 클래스는 KA와 같이 주기적으로 발생하는 메시지를 지원하기 위해 스레드를 이용하여 구현하였다.

그림 5는 COPS-PR 클라이언트의 상태 천이도를 보여주며, 각 상태는 COPSPrPEP 클래스의 함수를 나타낸다. 서버와 연결을 위해 필요한 정보는 COPSPrCfg 클래스를 이용하여 얻어 내고, COPSPrCfg 클래스는 ConOptions 클래스를 이용하여 COPS-PR 클라이언트에 설정 정보를 얻는다. 소켓의 연결이 이루어 지면 sendOPNMsg 함수를 호출하여 OPN 메시지를 COPS 서버로 전송한다. 서버가 CAT 메시지를 클라이언트로 전송하여 수락을 통지하면, sendREQMsg 함수를 호출하여 REQ 메시지를 보내어 정책 정보를 요청한다. 이때 REQ 메시지에 대한 COPSPrReq 클래스의 인스턴스를 생성하고 요청 상태 정보를 관리한다. 이후, DEC 메시지를 수신한 경우 Request State Flag가 설정되었는지 여부를

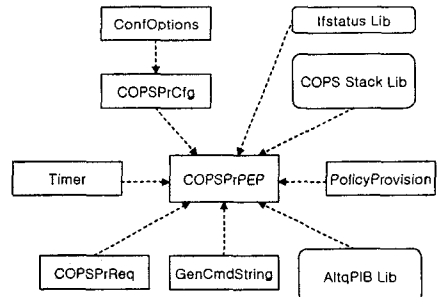


그림 4. 구현한 PEP 라이브러리 구조

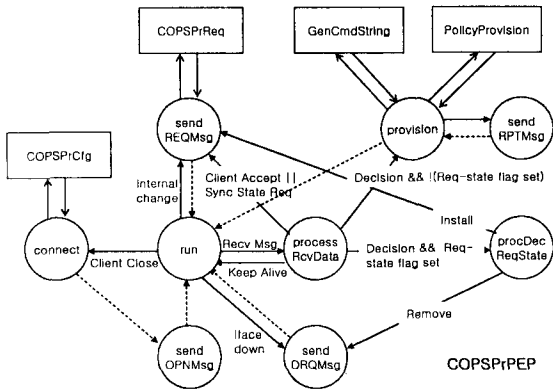


그림 5. COPS-PR 클라이언트 상태 천이도

확인한다. 만약 설정된 경우 명령 코드에 따라 sendREQMsg 함수를 호출하여 REQ 메시지 보내거나 sendDRQMsg 함수를 호출하여 DRQ 메시지를 보낸다. DEC 메시지에 Request State Flag가 설정되지 않았다면, PolicyProvision 클래스를 이용하여 정책 정보를 QoS에 전송하고 이에 대한 결과는 sendRPTMsg 함수를 호출하여 서버에 RPT 메시지를 전송하여 알린다. 또한, 라우터 내부의 상태 변화에 따라 REQ 메시지를 전송하거나 DRQ 메시지를 전송할 수 있다.

3.3 COPS 클라이언트와 QoS간의 프로토콜 설계 및 구현

QoS와의 연결은 유닉스 도메인 소켓을 사용하여 이루어지며, 주소는 설정파일의 ALTQ_UADDR 필드 값을 통해 얻는다. 기본적으로 QoS와의 정보교환은 DEC 메시지를 수신할 때마다 이루어지며, 이러한 정보는 ALTQ 명령어들로 이루어진다. COPS 클라이언트와 QoS와의 통신 프로토콜은 그림 6과 같이 Stop and Wait 방식과 유사하게 설계하였다. 또한, DEC 메시지의 끝을 명시하기 위해 EOM(End Of Message)을 사용한다. EOM 전까지 전송되는 명령어들의 성공적 처리 여부는 각 명령에 대한 OK/NOK 메시지를 통해 알아낸다. 설계된 프로토콜은 간단하면서도 QoS에서 요구되는 톨백에 충분한 정보를 제공할 수 있으며, 유닉스 도메인 소켓을 사용하므로 명령 혹은 OK/NOK 메시지의 전송 시 에러 및 손실을 고려하지 않았다.

COPS 클라이언트에서 위의 기능들은 PolicyProvision 클래스가 담당하는데, write 함수로 QoS에 명령 송신을, getResponse 함수로 그에 대한 응답을 수신한다. Decision 메시지에 해당하는 모든 명령이 성공적이면 provision 함수는 참을, 그렇지 않으면 거짓을 반환한다. 잘못된 명령이 전송된 경우 QoS는 즉시 톨백을 수행하며, EOM이 수신될 때까지 COPS 클라이언트로 온 명령 모두를 폐기한다.

4. 테스트 베드 구축 및 동작 검증

COPS-PR 클라이언트와 QoS 데몬과의 동작 검증을 위해 COPS 서버로부터 클라이언트에게 PIB로 설정 정보가 제대로 전달되어 디코딩 되는지, 그리고 디코딩 된 설정정보가 QoS에 전달되어 정상적으로 설정이 이루어 지는지를 살펴본다. 테스트를 위해 사용된 명령은 다음과 같다.

```
class priq fxp0 fxp0_ef_class NULL priority 2
filter fxp0 fxp0_ef_class name _Ef Filter 0 0 0 1
conditioner fxp0 fxp0_ef_cdnr <tbmeter 6000 64K <mark 0xb8> <drop>>
filter fxp0 fxp0_ef_cdnr name cdnr_Filter 0 0 155.230.14.213 0 0
delete fxp0 fxp0_ef_class
```

이들 명령은 DEC 메시지를 통해 전송되는데, EF 서비스를 위한 클래스/필터 명령, 그리고 컨디셔너 및 해당 필터 설정은 하나의 "Install" DEC 메시지로 전송하고, 일정 시간이 지난 뒤에 클래스를 삭제하도록 "Remove" DEC 메시지를 전송하였다. 그림 7-(a)는 PDP로부터 수신한 DEC 메시지 중 class

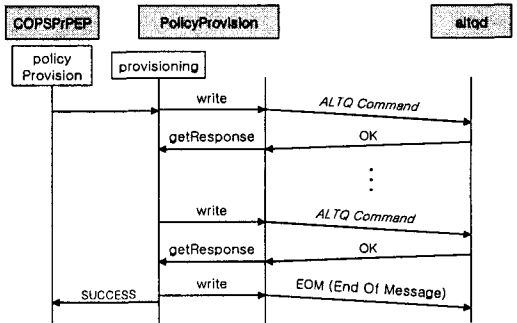
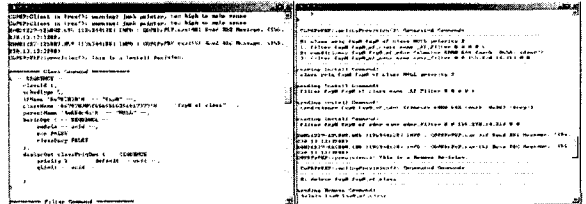
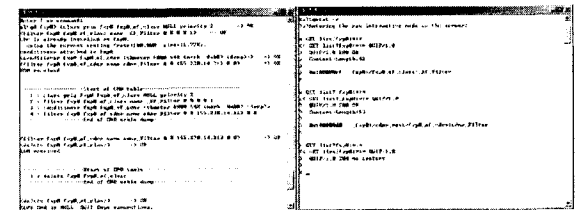


그림 6. QoS와 COPS 클라이언트 간 통신

명령의 디코딩 결과를, (b)는 디코딩된 정보를 ALTQ 명령어로 변환하여 QoS에게 전송하는 과정을, 7-(c)는 QoS에서 수신한 명령의 처리 결과를, 7-(d)는 altqstat 응용을 이용한 라우터 자원 예약 상태 정보를 출력한 결과를 보여준다. 그림과 같이 구현된 라우터는 COPS를 이용하여 동적 자원 제어가 가능함을 보이고 있다.



(a) 명령의 디코딩 테스트 화면 (b) QoS로 메시지 전송 과정



(c) QoS 출력 화면 (d) 자원 관리 확인
그림 7. 구현된 라우터의 검증 결과

5. 결론

본 논문에서는 대역 브로커를 통한 동적 자원 관리를 지원하는 PC 기반의 차별화 서비스 지원 라우터를 구현하였다. 이를 위해 COPS 클라이언트, 라우터 자원 제어를 위한 QoS, 톨백 모듈, 인터페이스/클래스/필터 추가 삭제 등의 기능, 그리고 QoS와 COPS 클라이언트간 프로토콜을 설계 및 구현하였다. 또한 테스트 베드 상에서 구현한 라우터의 동작 검증을 수행하였다.

참고문헌

[1] S. Blake, et. al., "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.
 [2] http://qbone.internet2.edu/bb/
 [3] http://www.csl.sony.co.jp/person/kjc/
 [4] http://www.voida.org/
 [5] http://www.fokus.gmd.de/ovma/freeware/snacc/