

차별화 서비스 망에서 COPS 기반 대역 브로커 설계 및 구현

한태만[†], 김동원^{**}, 정유현^{***}, 이준화^{****}, 김상하^{*****}

요 약

본 논문에서는 차별화 서비스를 지원할 수 있는 테스트 베드 개발 연구를 수행한다. 먼저 인터넷 상의 QoS 보장 모델인 차별화 서비스에 대한 기술을 소개한다. 그리고 FreeBSD 상에서 ALTQ 시스템을 사용한 차별화 서비스 지원 라우터와 대역 브로커를 설계하고 테스트베드 상에서 검증을 수행한다. 차별화 서비스 망에서 QoS를 보장받기 위해서 단말은 항상 대역 브로커와 사전에 SLA를 맺어 놓고, 서비스를 시작할 때 대역 브로커에게 RAR 메시지를 전송한다. 그러면, 정책 서버인 대역 브로커는 자신이 가지고 있는 PIB 정보를 기반으로 각 라우터에게 명령을 내린다. 물론 이때 대역 브로커와 각 라우터 사이의 모든 시그널링 메시지는 IETF에서 제안하고 있는 COPS 프로토콜을 이용해서 주고 받게 된다. 차별화 서비스가 지원되는 테스트베드에서 두 클라이언트가 VOD 서버에게 접속해서 비디오 서비스를 받을 때 한 사용자는 EF 클래스로 서비스를 받고, 다른 하나는 BE 클래스로 서비스를 받고, 망은 적절한 백그라운드 트래픽을 통해 일정한 혼잡 상태이다. 이 때 실험 결과에서는 각 수신 단말에서의 성능과 중간 라우터에서의 드롭된 패킷 수를 비교하였다. 그 결과 EF 클래스로 서비스 받는 수신 단말은 처음 RAR에 의해서 할당 받은 대역 내에서 항상 QoS를 보장받고, BE 클래스는 항상 망 상황에 의존해서 망에 혼잡이 발생하지 않을 때에는 성능이 좋지만 조그만 망에 혼잡이 발생해도 성능이 급격히 떨어짐을 볼 수 있었다.

An Implementation of Bandwidth Broker Based on COPS for Resource Management in Diffserv Network

Tae-Man Han[†], Dong-Won Kim^{**}, You-Hyeon Jeong^{***},
Jun-Hwa Lee^{****}, Sang-Ha Kim^{*****}

ABSTRACT

This paper discusses a testbed architecture for implementing scalable service differentiation in the Internet. The differentiated services (DiffServ) testbed architecture is based on a model in which a bandwidth broker (BB) can control network resources, and the ALTQ can reserve resources in a router to guarantee a Quality of Service (QoS) for incoming traffic to the testbed. The reservation and releasemessage for the ALTQ is contingent upon a decision message in the BB. The BB has all the information in advance, which is required for a decision message, in the form of PIB. A signaling protocol between the BB and the routers is the COPS protocol proposed at the IETF. In terms of service differentiation, a user should make an SLA in advance, and reserve required bandwidth through an RAR procedure. The SLA and RAR message between a user and the BB has implemented with the COPS extension which was used between a router and the BB. We evaluates the service differentiation for the video streaming in that the EF class traffic shows superb performance than the BE class traffic where is a network congestion. We also present the differentiated service showing a better packet receiving rate, low packet loss, and low delay for the EF class video service.

Key words: Quality of Service(서비스의 질), Bandwidth Broker(대역 브로커), Resource Allocation request(자원 할당 예약), Service Level Agreement(서비스 레벨 정의)

※ 교신저자(Corresponding Author): 한태만, 주소: 대전광역시 유성구 가정동 161(305-350), 전화: 042)860-5245, FAX: 042)860-5440, E-mail: tmhan@etri.re.kr

접수일: 2003년 7월 29일, 완료일: 2003년 10월 9일
* 한국전자통신연구원 네트워크연구소 스트리밍기술팀 선임연구원

** 충북과학대학 정보통신학과 교수

(E-mail: dwkim@citech.ac.kr)

*** 한국전자통신연구원 네트워크연구소 스트리밍기술팀 팀장, 책임연구원

(E-mail: yhjeong@etri.re.kr)

**** 충남대학교 대학원 컴퓨터학과 박사과정

(E-mail: junalee@cclab.cnu.ac.kr)

***** 충남대학교 컴퓨터학과 교수

(E-mail: shkim@cclab.cnu.ac.kr)

1. 서 론

현재까지 QoS(Quality of Service) 보장을 위한 연구는 서비스 품질 보장을 위한 모델의 정의 및 주요 모듈의 알고리즘 개발에 치중되었다. 그러나 최근 중단간 QoS 보장을 위한 이질적 QoS 제공 기술을 사용하는 도메인간의 연동 기술에 관한 연구가 대두되고 있다. 이러한 도메인간 연동 기술에서 핵심적 요소 기술 중 하나인 대역 브로커는 QBone, TF-NGN 등의 프로젝트를 중심으로 연구가 진행 중이다. 하지만 이제까지의 대역 브로커 연구 개발은 각각의 프로젝트에서 목표로 하는 서비스 모델, 관련 요소 기술 및 대역 브로커에서 사용하는 신호 프로토콜 등이 서로 상이했으며, 이에 따른 표준화나 연동 기법에 관한 연구가 미흡한 실정이다.

현재 QBone, AQUILA 등의 프로젝트에서는 망에서 제공하고자 하는 서비스 규격 및 관련 요소 기술의 상세화 작업만이 진행되어 있을 뿐, 차별화 서비스 지원 라우터의 구현과 이 라우터를 망에 적용한 연동 실험 등에 관한 연구는 미비하다. 또한, 위의 프로젝트들에서는 테스트베드 구축을 위하여 대부분 시스코나 주니퍼와 같은 벤더들의 라우터를 사용하여 망을 구성했는데, 이들 라우터는 CLI(Command Line Interface), SNMP 등의 서로 다른 신호 프로토콜을 사용한다. 따라서, 다양한 장비들을 위한 대역 브로커 설계 및 구현에 관한 연구가 진행되어야 한다.

MIB(Management Information Base)는 망 관리에 관련되는 모든 정보를 데이터베이스 형식으로 제공하는 반면, COPS의 PIB(Policy Information Base) 구조 형식으로 사용되는 SPPI(Structure of Policy Provisioning Information)는 QoS 제공에 필요한 정보만을 관리함으로써, 망의 효율적 관리가 가능하다. 또한, SNMP(Simple Network Management Protocol) 프로토콜을 기반으로 하는 QoS 제공 방안들은 상당히 연구가 진행되었지만, SNMP의 MIB의 구조 형식으로 사용되는 SMI(Structure of Management Information)는 1988년 처음 개발된 이후 SMIv2까지만 연구가 진행되었기 때문에, QoS 제공에 한계를 가진다. 따라서 향후 대역 브로커의 신호 프로토콜로서 COPS가 주로 사용될 것으로 보여지나, 아직까지 COPS를 이용한 동적 자원 관리, 도메인 간의 SLS 협상을 지원하는 대역 브로커 및 COPS를 지원하는

차별화 서비스 지원 라우터에 관한 개발 및 연구가 이루어지지 않고 있다.

본 논문에서는 이제까지의 차별화 서비스 요소 기술, 망 관리 기법을 분석하고 이를 바탕으로 차별화 서비스가 가능한 대역 브로커 구현을 목표로 한다. 대역 브로커가 각 라우터에게 정확한 정책을 내리기 위해서 초기 라우터 설정 정보를 받고 각 라우터에 적합한 정책을 내리기 위해서 IETF에서 제안하고 있는 COPS라는 표준 프로토콜을 사용한다. 그리고 각 라우터에게 QoS 정책을 내릴 수 있도록 기반이 되는 정책 서버를 PIB을 기반으로 구현하고 있다. 그리고 테스트베드를 구축하여 간단한 실험을 통해서 차별화 서비스가 수행됨을 보인다.

본 논문의 구성은 다음과 같다. 서론에 이어 제 2장과 3장에서는 차별화 서비스와 COPS 프로토콜에 대해 간단히 알아보고, 제4장에서는 대역 브로커와 라우터의 설계 및 구현과 실험망에서 테스트한 실험 결과에 대한 성능분석에 대해서 기술한다. 마지막으로 제 5 장은 결론과 향후 연구 방향에 대해서 설명한다.

2. 차별화 서비스

최근에 인터넷 서비스는 큰 대역폭과 비교적 낮은 지연을 요구하는 등 인터넷 서비스 품질에 대한 관심이 높아지고 있다. 특히 인터넷 서비스들은 IP 기반이므로 현재 인터넷에서 지원하는 것보다 더 확실한 서비스 보장을 전제로 한다. 하지만, 현재의 인터넷을 통해 패킷을 전송하면, 목적지까지 확실하게 도달한다는 보장이 없다. 그래서, 보다 나은 인터넷 서비스를 보장하기 위한 방안들이 많이 제시되었다. RSVP(Resource Reservation Protocol)를 사용한 통합 서비스(Integrated Service)가 제안되었지만, 각각의 플로우에 따라서 자원을 예약하므로, 확장성과 정보의 유지가 어렵고, 현재의 응용 프로그램에서도 RSVP 프로토콜을 지원해야 되므로 확장성과 호환성에 문제가 있다. 그래서 단순한 최선형 서비스와는 차별화된 서비스들을 제공함과 동시에 확장성과 호환성 문제를 해결하기 위한 방안으로 차별화 서비스(Differentiated Service)가 제안되었다. 차별화 서비스는 RFC 791에서 정의한 IPv4의 우선순위 표시 서비스를 개선한 것으로 인터넷에서 보다 다양한 서비스를

품질을 제공할 수 있도록 하는 기본 패러다임을 사용하고 있다. 차별화 서비스에서는 PHB(Per-Hop Behavior)라고 정의하는 종단간 우선순위를 갖는 서비스 집합을 미리 정의해서 처리함으로써, 서비스에 대한 정보를 유지할 필요가 없고 현재의 망에서 최소한의 수정을 통해 구현이 가능하다[1-5].

2.1 차별화 서비스 망 구조

그림 1은 차별화 서비스 망의 일반적인 구조를 나타낸 것이다. 차별화 서비스 망은 RSVP 사용에 따른 복잡한 자원 예약 절차를 피하기 위해 홉간의 패킷 처리를 PHB로 정의하고 있다. 그림 1에서 호스트 A와 호스트 B간에 서비스 품질을 보장 하기 위해서 호스트 A는 인접한 서비스 제공자와 계약을 맺게 된다. 이 계약에 의해서 호스트A로부터 전송되는 트래픽에 대한 서비스 품질의 정도를 규정하게 된다. 계약에 따라서, 호스트A는 에지 라우터(Edge Router)로 패킷을 보내게 되고, 에지 라우터는 서비스의 품질을 보장하기 위한 동작을 수행한다.

에지 라우터는 유입되는 트래픽의 호스트를 분류하여 계약 위반 여부를 확인하고 정책에 따라 DSCP(Differentiated Service Code Point)의 PHB(Per Hop Behavior)에 따른 마킹을 수행한다. 마킹된 패킷들은 DSCP 값을 통해서 코어 라우터(Core Router)로 보내진다.

코어 라우터에서는 패킷의 DSCP 값을 통해 PHB를 확인하고, 사전에 정해진 PHB에 따라 패킷을 처리한다. 이와 같이 코어 라우터는 단순히 패킷의 DSCP 값을 기준으로 패킷을 처리 하므로 각 플로우의 상태를 관리할 필요가 없어 구현이 용이하다.

대역브로커는 차별화 서비스 모델의 QoS 보장을 위해서는 망의 자원을 관리/감독하고 사용자의 자원

사용 요구에 따라 자원을 파악하고 승인/거부를 결정하는 장치가 필요하게 되었고 이러한 장치를 Van Jacobson가 대역 브로커(Bandwidth Broker)라 명명하였다[6-8].

2.2 대역 브로커의 기능

하나의 도메인 내에서는 하나의 논리적인 대역 브로커가 존재한다. 실제 물리적으로 봤을 때 대역 브로커는 다양한 부수 장치들로 구성이 가능하지만 그림 1에서와 같이 논리적으로는 하나의 대역 브로커로 보여져야 한다. 사용자가 자원의 사용을 대역 브로커에게 요구하면 사용자 요구에 대해 종합적인 플로우 정보를 담고 있는 PHB들을 참고해서 승인여부를 결정한다. 이 요구가 승인이 되면 PHB에 부합되는 DSCP 값이 에지 라우터에서 사용자의 패킷에 마킹된다. 이 때 RAR(Resource Allocation Reservation)은 플로우가 요구하는 자원의 양과 자원 예약이 가능한 기간을 표시하게 된다. 이런 사용자의 요구를 처리하기 위해 대역 브로커는 다양한 SLA와 SLA 별로 할당된 대역폭을 저장하고 향후 할당량을 결정하기 위한 기초 자료로 데이터베이스화 해서 가지고 있다. 즉 대역 브로커는 사용자의 요청이 있을 때 이런 기초 자료를 바탕으로 사용자에게 우선적인 서비스를 할당하고 정의된 서비스를 위하여 올바른 전달 행동으로 통신망 라우터를 구성하도록 하는 기능을 수행하게 된다.

대역 브로커의 통신은 인트라/인터 도메인 통신으로 구분된다. 도메인들은 서로 다른 장비 및 메커니즘을 사용할 수 있기 때문에 인터 도메인 통신은 표준이 요구된다. 인트라 도메인 통신은 특별히 외부 정책 서버를 사용하는 경우 RAR의 수용 여부 결정 기능을 수행할 때도 있다. 인트라 도메인에 대한 표준화가 부족한 관계로 지금까지의 대역 브로커 설계는 통신과 설정 위주보다는 PHB 설정에 비중을 두고 있다. 또한 대다수의 라우터 생산 업체들은 라우터를 설정하기 위해 각각의 고유한 방법을 사용한다. IETF의 차별화 서비스 워킹 그룹은 DiffServ SNMP MIB과 DiffServ COPS PIB을 명시하고 있다. 두 가지 모두 인터넷 드래프트의 형태로 발전하고 표준화가 진행되고 있는 상황이다. 대역 브로커 소프트웨어가 라우터의 종류에 따라 망의 관리에 제약을 받지 않아야 하므로, 대역 브로커는 차별화 서비스를 지원하기

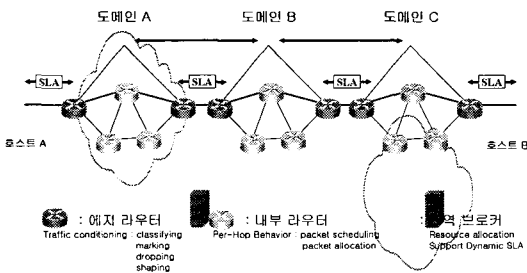


그림 1. 차별화 서비스 망의 기본 구조

위해서는 MIB나 PIB의 형태뿐만 아니라 다른 형태로 설치된 라우터에도 설치가 가능해야 한다.

대역 브로커는 도메인 내부의 자원만을 제어할 수 있기 때문에 다른 도메인에 속한 라우터는 설정할 수 없다. 그래서 인접 도메인 간에는 사전에 합의된 계약이 필요하다. 인접 도메인에 대해 대역 브로커가 대역 할당 요구를 승인하면 계약에 의해서 특정 클래스를 지정해서 대역폭을 할당한다. 그 후 대역 브로커는 도메인 내의 에지 라우터를 설정하고, 대역폭 예약에 관해 인접 도메인과 협조한다.

3. COPS

COPS(Common Open Policy Service)[9] 프로토콜은 정책기반 네트워크 관리를 위해서 정책정보를 표현하고 이를 실제 장비에 반영할 수 있도록 IETF에서 정의한 프로토콜로서 메시지들을 정책 개체들로 설계하며, 정책 요소들로 구성된다. 정책 요소들은 규칙을 평가하기 위해 필요한 정보들의 단위가 된다. 또한, 정책 요소는 사용자 정의 또는 어플리케이션 정의를 운반하고, 다른 정책 요소는 사용자 인증과 신용 정보를 운반할 수도 있다. 정책 요소들은 QoS 제공을 위한 신호 프로토콜과는 별개다.

COPS는 질의/응답(Query/Response) 프로토콜로서 Outsourcing 모델과 Provisioning 모델을 제공한다. Outsourcing 모델은 PEP에서 일어나는 이벤트를 처리하기 위한 모델로서 이 시나리오에서는 PEP(Policy Enforcement Point)는 PDP(Policy Decision Point)에게 결정을 위임한다. 그리고 COPS-RSVP 개념을 사용한다. 예를 들어, RSVP메시지가 PEP에 도착했을 때 PEP는 이 요청의 수용 여부를 결정해야 하는데, PEP는PDP에게 질의를 보냄으로써 결정을 PDP에게 위임한다. COPS-PR(COPS-Provisioning) 모델[10]은 PEP와 PDP의 1:1관계를 가정하지 않는다. PDP는 외부 이벤트(user input)나 PEP 이벤트에 대한 반응으로서 PEP를 설정한다. Provisioning은 전체 라우터에 QoS관련 설정을 하는 것이 될 수도 있고, 단순한 DiffServ 마킹, 필터만 업데이트하는 것일 수도 있다.

Decision은 대부분 외부 이벤트나 PDP내에서의 이벤트로 인해서 PEP로 발생되며, PEP로부터의 요청에 대한 응답으로 Decision이 발생하는 경우는 드

물다. 데이터 모델은 정책 데이터를 정의하는 PIB을 기반으로 한다. 정책 제어 범위 내에서 PEP는 하나의 PDP와 연결을 가질 수 있기 때문에 한 서버만이 특정 정책 설정을 변경할 수 있다. 그래서 정책 설정이 효과적으로 잠글 수 있다. 신뢰성 있는 TCP 연결을 사용하고 상태 정보의 공유나 동기화 방법을 사용한다. PEP와 PDP사이에 폴링을 요구하지 않는 실시간 이벤트 처리 통신 메커니즘을 사용한다.

PEP와 PDP 간의 상호 작용은 부팅 시 주 PDP로 COPS를 통해 연결되고 연결이 이루어지면 PEP는 자신의 정보를 환경 설정 요청 형태로 PDP에 전달한다. 이때 전달되는 정보는 하드웨어 타입, 소프트웨어 버전, 환경 설정 정보, 최대 COPS-PR 메시지 크기 등이다. 응답으로 PDP는 그 장치와 관련된 모든 정책 데이터를 전송한다.

PEP는 정책 데이터를 받았을 때 이를 그 장비에 맞는 QoS 파라미터로 대응시킨다. PDP에서 정책 데이터가 변했을 때 이를 PEP에게 전달하면 PEP는 이를 자신의 파라미터로 변경하여 설정한다. 정책에 포함될 수 없는 변경 사항(board removed, board added, new software installed)이 PEP에서 발생했을 때, PEP는 이 새로운 정보를 PDP에게 보낸다. PDP는 이 정보를 받았을 때, PEP에게 새로 필요한 정책을 추가하거나 필요 없어진 정책을 변경시킨다.

4. 설계 및 구현

현재 차별화 서비스를 제공할 수 있는 테스트베드는 Linux, FreeBSD 등의 다양한 플랫폼에서 구현되어 있다. 하지만, 대부분의 구현물은 미리 작성된 설정 파일이나 혹은 커맨드 라인을 통한 사용자의 명령 입력을 이용한 정적 자원 제어만을 지원하고 있는 상태이다. 따라서 본 논문에서는 COPS 프로토콜을 이용하여 라우터를 동적으로 제어하기 위해서 기존 구현물을 수정하고 라우터의 동적 자원 제어가 가능하도록 기존의 구현물 수정을 통한 차별화 서비스 라우터를 개발하고 정책 서버와 정책 정보를 교환하는 COPS-PR 클라이언트를 구현하였다.

COPS-PR 클라이언트는 Vovida[11]에서 공개 소스로 제공되는 COPS 스택을 사용하여 구현하였고, QoS 데몬은 UNIX 도메인 소켓을 이용해 COPS 클라이언트와 통신하며 라우터의 자원 제어를 수행한다.

Vovida에서 구현한 라이브러리는 COPS-PR의 표준을 따르지 않고 VOCAL(Vovida Open Communication Application Library)을 지원하기 위한 클라이언트 타입을 위해 구현되어 IETF에서 정의한 프로토콜 스펙을 그대로 따르지 않는 문제가 있어서 이들 라이브러리를 그대로 사용하기는 어렵고 수정을 하였다.

4.1 대역 브로커

본 논문에서 제안하고 있는 대역 브로커의 구조는 그림 2와 같다. 대역 브로커의 인터페이스는 VOD 서버에서 오는 SLA와 RAR을 처리하고, 망에 있는 각 라우터에서 올라오는 초기 정보 및 각 정책에 맞는 명령이 내려가는 부분이 있다. 그리고 대역 브로커가 정책을 결정하기 위해서 사용자 또는 각 라우터에 올라오는 모든 정보를 DB에 저장하고 있다. 그리고 필요한 정책을 결정하기 위해서 기존에 DB에 있는 정보를 기반으로 만들 수 있다. 또한 대역 브로커의 정확한 정책 결정을 위해서 SNMP 프로토콜을 이용한 망 관리 정보 및 라우팅 프로토콜을 이용한 라우팅 테이블의 정보도 필요하게 된다.

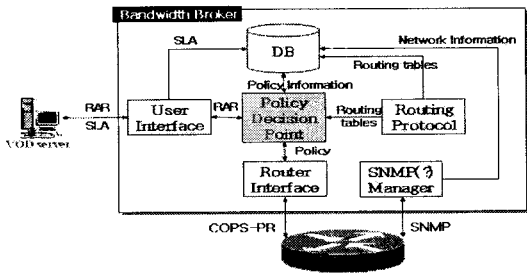


그림 2. 대역 브로커의 구조

4.1.1 대역 브로커와 VOD 서버

VOD 서버는 서버에 접속한 클라이언트에 전송하는 트래픽에 대해서 QoS를 보장하기 위해서 사전에 대역 브로커와 SLA(Service Level Agreement)를 맺는다. SLA는 사전에 VOD 서버와 대역 브로커 사이의 계약으로 망을 사용할 유효기간과 보장 받을 대역폭 양에 대해서 명시한다. 본 논문에서는 VOD 서버가 QoS 보장을 원하는 클래스에 대해서 서비스하기 전에 꼭 SLA부터 맺도록 하고 있다. 그래서, 본 논문에서는 시스템 시작과 동시에 SLA를 맺고

있다.

사용자들이 VOD 서버에 접속 후 서비스를 요청할 때, 본인은 QoS 서비스를 받을 것인지 아닌지에 대한 명시도 같이 이루어진다. 만약에 클라이언트가 QoS 보장을 원하는 경우, VOD 서버는 대역 브로커에게 대역폭에 대해서 미리 할당 받아야만 한다. 이때 VOD 서버는 대역 브로커와 RAR(Resource Allocation Request)를 통해서 대역폭을 할당 받는다.

그림 3은 VOD 서버와 대역 브로커 사이에서 SLA와 RAR을 맺을 때 시나리오 절차를 보여주고 있다. 첫 번째, SLA 인 경우에는 VOD 서버는 대역 브로커에게 SLA Request 메시지를 전송하고, 대역 브로커는 수신한 SLA Request 정보를 DB에 저장하고 제대로 처리가 되었으면 다시 VOD 서버에게 SLA Decision 메시지를 통해서 OK인지 아닌지를 알려준다. 두 번째, RAR인 경우에는 VOD 서버는 RAR Request 메시지를 대역 브로커에게 전송하고, 대역 브로커는 DB의 정보를 이용해서 VOD 서버가 요구하는 대역 양만큼 각 라우터를 셋팅한다. 그런 후에 모든 절차가 제대로 이루어 졌으면 다시 VOD 서버에게 RAR Decision 메시지를 통해서 OK인지 아닌지를 알려준다.

그림 4는 VOD 서버와 대역 브로커 사이에 RAR

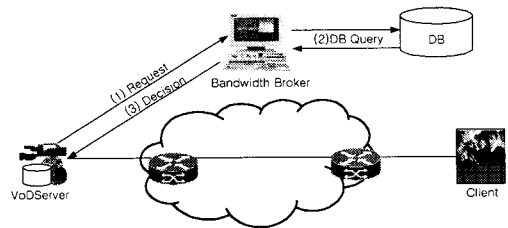


그림 3. 대역 브로커와 VOD 서버 사이에 SLA와 RAR

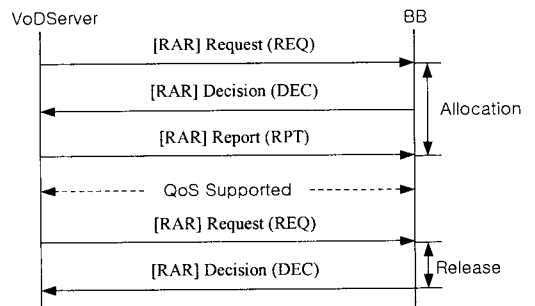


그림 4. RAR 메시지 처리 과정

메시지를 주고 받는 절차를 보여주고 있다. 본 논문에서는 VOD 서버와 대역 브로커 사이에 RAR 메시지와 SLA 메시지를 주고 받기 위해서 새로운 프로토콜을 사용하지 않고, 대역 브로커와 라우터 사이에서 시그널링을 주고 받을 때 사용하는 COPS 프로토콜을 확장해서 사용하고 있다. COPS 프로토콜의 Common 헤더의 Client Type 필드에 0x8888의 값을 셋팅함으로 대역 브로커에게 VOD 서버에서 받은 메시지라는 것을 알려줄 수 있다. Client Handle은 SLA와 RAR에 필요한 모든 파라미터에 대해서 정의하고 있다. 이 이유는 SLA와 RAR은 성격이 비슷하므로 동일한 메시지 포맷을 사용하기 위해서이다. 단 메시지를 받아서 처리하는 대역 브로커에게 SLA인지 RAR인지를 알려주기 위해서 Client Handle의 SLS ID 필드에서 구별해 주고 있다. Client Handle은 대표적으로 송신단말과 수신단말의 IP 주소와 Port 번호, 트래픽을 전송하기 위해서 사용된 프로토콜, 유효날짜, 요구 대역 양 등의 필드가 있다.

4.1.2 대역 브로커와 라우터

그림 5는 대역 브로커와 각 라우터 사이에서 COPS 프로토콜을 이용해서 주고 받는 시그널링 절차를 나타내고 있다. 첫 번째 각 라우터는 대역 브로커와 시그널링을 주고 받기 위해서 소켓을 열어야만 한다. 그래서 모든 라우터는 대역 브로커에게 OPN(COPS Client Open) 메시지를 전송하고, 대역 브로커는 CAT(COPS Client Accept) 메시지를 전송한다. 이때는 각 라우터와 대역 브로커 사이에 소켓을 형성된 상태로 어떤 메시지도 주고 받을 수 있도록 준비된 상태이다. 두 번째는 현재 본 논문은 COPS-PR 프로토콜을 사용하므로, 처음 각 라우터의 초기 정보를 알아야만 한다. 그래서 각 라우터는 라우터가 가지고 있는 각각의 인터페이스의 정보를 모두 대역 브로커에게 REQ(COSP Request) 메시지를 전송하고, 대역 브로커는 수신한 정보를 PIB에 저장한 후 각 라우터에게 초기 셋팅할 정보(모든 라우터에 ALTQ[12]에서 EF 클래스에 대역 할당을 하는 Class 명령과 Filter 명령)에 대해서 DEC(COPS Decision) 메시지를 전송한다. 세 번째는 대역 브로커가 VOD 서버에서 온 RAR 메시지 수신 후 입구 에지 라우터에 ALTQ에서 패킷을 마킹할 수 있도록 CDNR(Conditioner) 명령과 Filter 명령을 DEC에 실어서 전송한다. 네 번

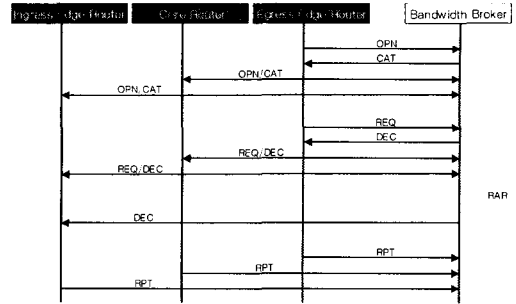


그림 5. 대역 브로커와 라우터 사이의 시그널링 절차

째는 주기적으로 모든 라우터가 대역 브로커에게 RPT(COPS Report State) 메시지를 보낸다. 이 메시지는 각 라우터가 주기적으로 대역 브로커에게 이 메시지를 전송하므로 대역 브로커가 자신이 관리하고 있는 라우터가 죽었는지 살았는지를 판단할 수 있도록 해 준다.

이 절에서 설명하고 있는 대역 브로커와 각 라우터는 시그널링인 COPS 프로토콜을 이용해서 각 라우터는 대역 브로커에게 자신의 정보(REQ)를 전송하고, 대역 브로커는 각 라우터에게 실제 망에서 QoS를 보장하기 위해서 ALTQ 명령어(DEC)를 실어서 보낸다.

대역 브로커와 각 라우터 사이에서 주고 받는 메시지 중 REQ는 다음 절인 라우터 부분에서 자세히 설명하고, 본 절에서는 대역 브로커 측면에서 메시지를 생성하는 DEC 메시지에 대해서 자세히 살펴보겠다. DEC 메시지에 들어가는 내용은 앞에서 설명했듯이 각 라우터에서 유일되는 트래픽 중 EF 클래스에 해당하는 트래픽에 대해서 QoS를 보장할 수 있도록 ALTQ라는 큐잉 부분의 명령어가 들어간다. 본 논문에서는 ALTQ의 여러 가지 명령 중 차별화 서비스에서 사용하는 명령어는 단 세 개뿐이다. 하나는 클래스 명령으로 차별화 서비스에서 제공할 수 있는 클래스 종류대로 셋팅할 수 있다. 두 번째는 필터 명령으로 입력된 패킷을 분류하는 역할을 한다. 세 번째 CDNR 명령은 입력된 패킷을 in-of-profile일 경우의 action과 out-of-profile일 경우의 action에 대해서 정의할 수 있다.

이 세가지 명령 중 클래스 명령과 필터 명령은 초기 각 라우터에서 올라오는 REQ 메시지에 대한 응답으로 DEC 메시지를 보낼 때 그 때 실어지는 정보이다. CDNR 명령과 필터 명령은 VOD 서버로부터

RAR 메시지가 왔을 때 대역 브로커가 입구 에지 라우터에게 전송하는 DEC 메시지에 실어서 보내진다. 본 논문에서 구현한 클래스 명령의 구조체는 그림 6에서 보여주고, 그림 7은 필터 명령, 그림 8은 CDNR 명령의 구조체를 보여주고 있다.

```

struct class_command{
    int classId;        // class ID
    int schedType;     // scheduler type
    char className[20]; // class Name
    char parentname[50];
                        // parent class name
    struct BasicOption basicOpt
                        // RED or RIO
    struct CommandChoice dscpSpOpt
                        // queue 종류 (prioq)
}
    
```

그림 6. 클래스 명령 구조체

```

struct Filter_command{
    int classId;        // filter ID
    char ifName[20];   // interface Name
    char className[20]; // class Name
    char filtername[50]; // filter Name
    int ruleNo;        // Rule Number
    struct FilterOption filterOptions
                        // filter parameter :
                        // source & destination의 IP,
                        // Port, subnet mask
                        // Protocol(UDP or TCP),
                        // DSCP, flow Label, gpi
}
    
```

그림 7. 필터 명령 구조체

```

struct CDNR_command{
    int cdnrId;        // conditioner ID
    char ifName[20];  // interface Name
    char cdnrName[20];
                        // conditioner Name
    struct Action action
                        // Action parameter :
                        // pass, drop, mark
}
    
```

그림 8. CDNR 명령 구조체

대역 브로커는 각 라우터에게 정책 기반에 Decision을 내리기 위해서, VOD 서버나 또는 각 라우터에서 오는 정보, 대역 브로커가 전송하는 정보 모두를 DB에 저장해서 PIB을 형성한다. 그런 후 항상 DEC 메시지를 보낼 때 이 PIB에 있는 정보를 기반으로 Decision을 내린다.

4.2 라우터

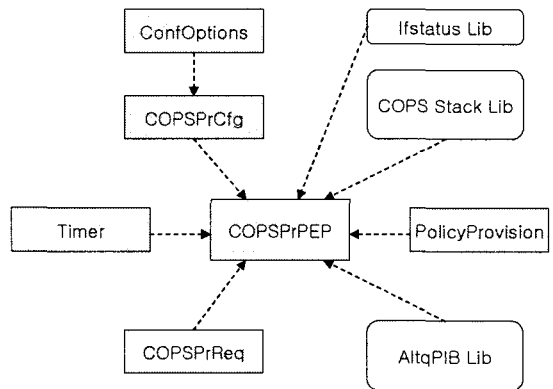


그림 9. PEP 라이브러리 구조

그림 9는 본 논문에서 구현한 PEP 라이브러리 구조를 나타내고 있다. COPSPrPEP는 가장 핵심이 되는 클래스로 PEP의 모든 동작은 COPSPrPEP를 통해 이루어진다. COPSPrCfg 클래스는 PEP에 관련된 정책서버 주소와 같은 환경 설정 정보를 관리한다. COPSPrReq 클래스는 요청 상태 정보를 관리하는 클래스이다. REQ 메시지는 PEP의 인터페이스 정보를 포함하는데, 이러한 정보는 ifstatus 라이브러리

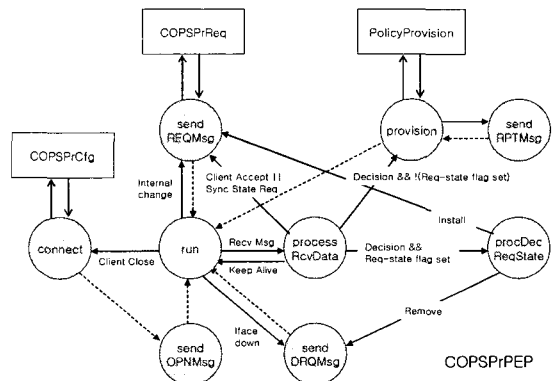


그림 10. COPS-PR 라우터 상태 천이도

를 통해 얻을 수 있다. PolicyProvision은 서버로부터 얻은 정책 정보를 분석하고 이를 이행하기 위해 QoS 데몬에 정보를 전달하는 역할을 한다. 이러한 정보의 전달은 UNIX 도메인 소켓을 이용하여 구현하였다. AltqPIB 라이브러리는 ALTQ 명령어를 바탕으로 작성한 PIB를 인코딩 및 디코딩하는 역할을 담당하는데, 이 라이브러리는 ASN.1 표기법으로 작성된 PIB를 ASN.1 컴파일러인 SNACC 사용하여 생성하였다. Timer 클래스는 Keep-Alive 메시지와 같이 주기적으로 발생하는 메시지를 지원하기 위해 스레드를 이용하여 구현하였다.

4.2.1 라우터 설계

그림 10은 본 논문에서 구현한 COPS-PR 라우터의 상태 천이도를 나타내는데, 각각의 상태는 COPSPrPEP 클래스에 함수로 구현하였다.

서버와 연결을 위해 필요한 정보는 COPSPrCfg 클래스를 이용하여 얻어 낸다. 여기서 COPSPrCfg 클래스는 ConfOptions 클래스를 이용하여 COPS-PR 클라이언트에 필요한 여러 가지 설정 정보를 얻어낸다.

소켓의 연결이 이루어지면 *sendOPNMsg* 함수를 호출하여 OPN 메시지를 서버에게 전송한다. 이에 대한 응답으로 서버가 CAT 메시지로 수락한다면 *sendREQMsg* 함수를 호출하여 REQ 메시지를 보내어 정책 정보를 요청한다. 이때 REQ 메시지에 대한 COPSPrReq 클래스의 인스턴스를 생성하도록 하여 요청 상태 정보를 관리하게 한다. 여기서 DEC 메시지를 수신했다면 요청 상태 플래그(Request-State Flag)가 설정되었는지 확인하여 명령 코드에 따라 *sendREQMsg* 함수를 호출하여 REQ 메시지를 또는 *sendDRQMsg* 함수를 호출하여 DRQ 메시지를 보낸다. DEC 메시지에 요청 상태 플래그가 설정되어 있지 않으면 PolicyProvision 클래스를 통해 정책 정보를 QoS 데몬에게 전송한다. 이 결과는 *sendRPTMsg* 함수의 호출을 통해 RPT 메시지를 전송하여 서버에게 보고하게 된다. 또한 내부적인 변화에 따라 REQ 메시지를 전송하거나 DRQ 메시지를 전송하기도 한다.

4.2.2 COPS-PR 메시지 처리

그림 11은 라우터에서 COPS-PR 메시지를 처리하는 과정을 나타낸다. *processRcvData* 함수를 통

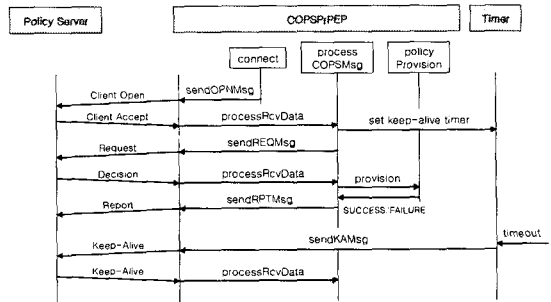


그림 11. 라우터의 COPS-PR 메시지 처리 과정

해 수신된 모든 메시지는 *processCOPSMsg* 함수에서 처리되는데, 헤더 정보를 이용하여 해당하는 함수를 호출한다.

클라이언트는 서버와의 연결을 설립하기 위해 *sendOPNMsg* 함수를 호출하여 OPN 메시지를 생성하여 보낸다. 서버는 클라이언트 타임을 확인한 후 CAT 메시지를 클라이언트에 보내 수락한다. *processCOPSMsg* 함수에서는 Account 및 Keep-Alive 타이머 값을 얻어내고 이를 설정한 다음, *sendREQMsg* 함수를 호출하여 REQ 메시지를 서버에 보낸다.

REQ 메시지는 요청 상태 정보를 식별하기 위한 클라이언트 핸들과 라우터의 인터페이스 정보를 포함한다. 여기서 핸들은 *createClientHandle* 함수를 통해 얻으며, 인터페이스 정보는 *getIfInfo* 함수를 통해 얻는다. 인터페이스 정보는 *if_info* 구조체를 통해 저장되는데 이 구조체의 내용은 그림 12와 같다.

REQ 메시지를 수신한 서버는 정책정보를 포함하는 Decision 메시지를 생성하여 클라이언트에게 보내는데, DEC 메시지의 요청 상태 플래그가 설정되어 있으면 클라이언트의 *processCOPSMsg* 함수에서는 *procDecReqState* 함수를 호출한다. 그리고 *procDecReqState* 함수는 명령코드를 검사하여 명령코드가 "Install"이면 *sendREQMsg* 함수를 호출하여 REQ 메시지를 보내도록 하며, "Remove"이면 *sendDRQMsg* 함수를 호출하여 DRQ 메시지를 보내도록 한다. 요청 상태 플래그가 설정되지 않은 경우는 *provision* 함수를 호출하는데 Named Decision Data 오브젝트로부터 정책 정보를 얻어 이를 QoS 데몬에게 전송하도록 한다. *provision* 함수는 명령코드를 검사하여 "Install"이면 *decodeInstallDec* 함수를 호출하고, "Remove"이면 *decodeRemoveDec* 함


```

struct if_info{
    char name[50]; // interface name
    int ifid;      // interface id
    int mtu;      // mtu size
    int addrfmly; // address family:

                //IPv6, IPv6, all
    struct in_addr inaddr; // IPv4 address
    int prefix;    // IPv4 address prefix
    struct in6_addr in6addr;
                // IPv6 address
    int prefix6;  // IPv6 address prefix
    int linktype;
                // link type: ethernet, etc.
    unsigned long linkspeed; // link speed
    char qdisc[16]; // queueing discipline
}
    
```

그림 12. if_info 구조

수를 호출한다. decodeInstallDec 함수는 Named Decision Data 오브젝트를 디코딩하여 ALTQ 명령어를 생성한다.

디코딩은 snacc으로부터 생성된 클래스 라이브러리를 이용하는데, 이 클래스들은 각각 ALTQ의 interface, class, filter, 그리고 conditioner 명령에 해당된다. 따라서 디코딩을 위해서는 각각의 오브젝트에 해당되는 클래스의 식별 정보가 필요하다. 이 정보는 PRID 값을 이용하는데, 각각의 PRID 값은 표 1과 같다. 클래스로 디코딩된 정책정보는 다시 GenCmdString 클래스를 이용하여 ALTQ 명령어로 변

표 1. ALTQ PIB의 PRID 값

구분		PRID 값
DEC 메시지	ALTQ interface 명령	1.3.6.1.2.2.1.interfaceid
	ALTQ class 명령	1.3.6.1.2.2.2.counting
	ALTQ filter 명령	1.3.6.1.2.2.3.counting
	ALTQ conditioner 명령	1.3.6.1.2.2.4.counting
REQ 메시지	인터페이스 정보	1.3.6.1.2.2.5.interfaceid

환된다. 여기서 생성된 명령어는 PRID 값과 짝을 이루어 저장된다. 이는 명령 코드가 "Remove"인 DEC 메시지를 처리할 때 사용된다. 이렇게 생성된 명령어는 PolicyProvision 클래스를 이용하여 QoS 데몬에게 전송된다.

Provision 함수의 리턴값이 참이면 정상적으로 ALTQ 명령이 수행되었음을 의미하며, 이러한 정보는 sendRPTMsg 함수를 호출하여 RPT 메시지를 통해 서버에게 전달된다.

4.2.3 QoS 데몬과의 연동

QoS 데몬과의 연결은 UNIX 도메인 소켓을 사용하여 이루어진다. 이 방식은 특정 파일이 주소역할을 한다.

기본적으로 QoS 데몬과의 정보교환은 DEC 메시지를 수신할 때마다 이루어지며, 이러한 정보는 ALTQ 명령어들로 이루어진다. 연동은 그림 6과 같이 간단한 프로토콜로 이루어지는데, DEC 메시지의 끝은 알리기 위해 EOM(End Of Message)을 사용한다. 그리고 이 EOM 전까지 전송되는 명령어들의 성공여부는 각각 명령에 대하여 OK, NOK 메시지를 QoS 데몬으로부터 수신함으로써 알 수 있다. 이러한 프로토콜 설계는 간단하면서도 QoS 데몬에서 DEC 메시지 단위로 rollback이 가능한 충분한 정보를 제공한다. 그리고 QoS 데몬과의 통신은 UNIX 도메인 소켓을 사용하므로 프로토콜 설계 시에 메시지 전송시의 에러나 손실은 고려하지 않아도 된다.

QoS 데몬과의 연동과정은 그림 13과 같다. 이 데

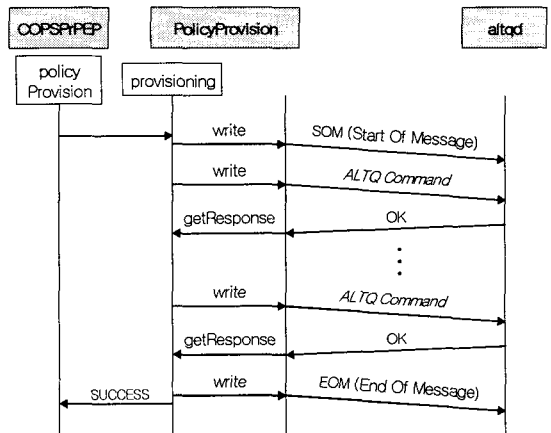


그림 13. 라우터에서 QoS 데몬과 연동 처리 과정

문과의 메시지 전송 및 처리과정은 PolicyProvision 클래스에서 이루어진다. write 함수를 통해 QoS 데몬에게 명령을 전송하며, getResponse 함수를 통해 응답을 수신하여 명령의 성공 여부를 확인한다. Decision 메시지에 해당하는 모든 명령이 성공적으로 이루어 지면 provisioning 함수는 참을 리턴하고 그렇지 않으면 거짓을 리턴한다.

5. 성능분석

5.1 실험환경

그림 14에서는 본 논문에서 구현한 차별화 서비스 테스트베드의 망 구성도이다. tbr1에 물려있는 단말들은 송신단말이고, tbr2에 물려있는 단말들은 수신단말이다. 클라이언트 A와 B는 VOD라는 VOD 서버에 접속해서 비디오 서비스를 받고, traffic1과 traffic2는 백그라운드 트래픽을 송수신하는 단말이다. 사용된 각 장비의 제원은 표 2와 같다.

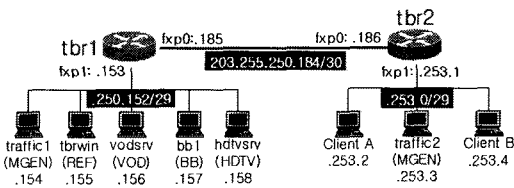


그림 14. 실험 환경

5.2 실험 절차

본 논문에서 구현한 환경에서 실험 절차는 그림 15에서 보여주고 있다.

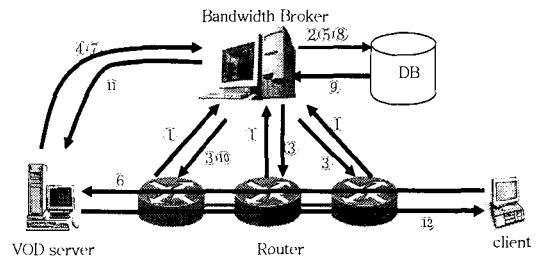


그림 15. 실험 절차

실험을 하기 위해서는 세 가지 부분으로 나누어서 동작시킬 수 있다. 첫 번째 각 라우터와 대역 브로커 사이에 초기 정보를 주고 받는 부분이다. 단계는 ①에서부터 ③까지이다.

① 망에 있는 각 라우터의 초기 정보를 대역 브로커에게 전송한다.

② 대역 브로커는 라우터의 초기 정보를 DB에 저장한다.

③ 각 라우터에게 기본 QoS에 필요한 셋팅 정보를 내린다. 본 논문에서는 Class 정보 및 Filter 정보를 내려주고 있다.

두 번째 VOD 서버와 대역 브로커 사이에 처음 SLA를 맺는 부분이다.

④ VOD 서버와 대역 브로커 사이에 SLA 계약 내용을 대역 브로커에게 전달한다.

⑤ 대역 브로커는 계약 맺은 SLA 정보를 DB에 저장한다.

마지막으로 실제로 클라이언트가 VOD 서버에게 서비스를 요청한 경우에, VOD 서버와 클라이언트 사이에 QoS를 보장하기 위해서 필요한 절차이다.

⑥ 클라이언트가 VOD 서버에게 서비스를 요청한다.

표 2. 실험 장비의 제원

	VOD server	백그라운드 트래픽 발생기	EF/BE class traffic 수신용 client	core Router	Edge Router
CPU	P-III 400 MHz	P-IV 500 MHz	P-IV 1.5 GHz	P-IV 500 MHz	P-IV 500 MHz
NIC	Realtek 8139c	3C905TX	3C905TX	3C905TX	3C905TX
OS	Solaris 2.8	Linux Red Hat 7.3	Window 2000	FreeBSD 3.4 Release	FreeBSD 3.4 Release
Application	Content:주만지 (MPEG2)	Traffic generator:MGEN	Player:Window Media Player	DS Demon: ALTQ-2.2, COPS	DS Demon: ALTQ-2.2, COPS

⑦ VOD 서버는 서비스에 요구되는 필요한 파라메터 정보를 RAR에 실어서 대역 브로커에게 보낸다.

⑧ 대역 브로커는 DB에 현재 서비스의 QoS를 만족시킬 수 있는 정책을 가지고 오도록 요청을 한다.

⑨ DB에서 현재 가지고 있는 정보를 기반으로 필요한 decision을 만든다.

⑩ DB로부터 만들어진 decision 명령을 라우터에 전송한다. 본 논문에서는 입구 에지 라우터에 Conditioner 명령과 Filter 명령만 내리면 된다.

⑪ VOD 서버에게 OK/FAIL에 대한 응답을 해 준다.

⑫ 만약 대역 브로커로부터 OK 메시지가 오면, 클라이언트가 요구하는 서비스를 제공해 준다.

5.3 실험 결과 분석

5.3.1 멀티미디어 응용 서비스 실험

클라이언트 A와 클라이언트 B 모두 VOD 서버에 접속해서 비디오 서비스를 받는다. 두 클라이언트가 같은 종류의 비디오 서비스를 받지만, 클라이언트 A는 EF 클래스로 서비스를 받고 클라이언트 B는 BE 클래스로 서비스를 받는다. 이 때, 비디오 서비스를 받는 각 수신단에서는 다양한 백그라운드 트래픽에 의한 망의 혼잡 환경 하에서 차별화된 서비스에 대한 품질을 살펴보고, BE 트래픽에서의 패킷 드롭과 실시간 서비스를 보장하는 EF 서비스를 실험한다.

실험에서는 차별화 서비스를 제공하기 위해서 Priority Queue를 지원하는 ALTQ를 통해, 각 라우터에서 EF 클래스의 서비스에 대해서 7M의 대역폭을 보장하고 전체 대역폭 중 EF 클래스가 사용하고 남은 대역에 대해서 default 클래스인 BE 서비스가 서비스 되는 PHB를 수행한다. 실제 실험에서 사용된 NIC 카드가 100Mbps일지라도 ALTQ에서는 전체 대역폭에 대해서 10Mbps 만 설정하고 실험을 한다.

클라이언트 A와 B는 VOD 서버에 접속해서 비디오 서비스를 수신하여 수신률을 측정한다. 이때 백그라운드 트래픽을 1M, 1.5M, 2M, 2.5M를 생성하여 임의의 단말에 보내어 네트워크의 혼잡을 발생시키면서, VOD 서버에서 전송되는 트래픽의 실시간 서비스를 보장하는지와 BE 서비스에 대한 드롭된 패킷에 대해서 알아본다.

실험 망에서 입구 에지 라우터에서 altqstat에 의하여 측정된 결과, EF 클래스를 서비스하는 트래픽에 대해서는 백그라운드 트래픽이 증가하여도 패킷

표 3. 백그라운드 트래픽에 따라 라우터에서 드롭되는 패킷 수

background traffic	Best Effort	EF Service
0.5 M	26,787	0
1M	819,474	0
1.5M	829,723	0
2M	861,759	0
2.5M	879,736	0

의 손실이 없는 반면에 BE 클래스 트래픽은 패킷의 손실이 증가한다. 표 3에서 잘 보여주고 있다.

표 4는 1분 동안 비디오 트래픽을 수신시에 각 수신단에서 측정된 시간을 나타내고 있다. 결과는 EF 클래스로 서비스되고 있는 비디오 서비스는 실시간 서비스의 QoS 만족 파라메터 중 하나인 지연 없이 서비스를 제공하고 있음을 볼 수 있다.

표 4. 1분 동안 비디오 파일을 수신시 측정된 시간

	EF class Service	Best Effort Service
3.5M background traffic	1분	2분 23초
4M background traffic	1분	3분 11초

표 3, 표 4에서 보면, 7M의 대역폭을 예약한 EF 클래스 서비스를 통한 비디오 서비스는 어떠한 패킷 드롭도 발생하지 않으며 지연이나 끊김 없이 서비스가 되고 있다. 반면에 BE 클래스 트래픽을 수신하는 수신단은 비디오 트래픽이 백그라운드의 트래픽에 따라서 많은 지연과 끊김 현상을 볼 수 있다.

5.3.2 UDP_generator tool을 이용한 실험

본 실험에서도 EF 클래스의 대역폭에 7M를 할당하고, EF 클래스와 BE 클래스 두 부분으로 나누어서 트래픽을 발생시킨다. 그리고, 다양한 백그라운드 트래픽을 통해 여러 종류의 망에 혼잡을 발생시킨다. 백그라운드 트래픽은 3M, 4M, 5M, 6M, 7M, 8M에 대해서 실험을 한다.

그림 16과 그림 17은 송신단말에서 EF 클래스와 BE 클래스 트래픽에 대해서 각각 3M의 트래픽을 생성시켰을 때, 각각의 수신자에서의 throughput과 중간 라우터에서 드롭되는 패킷 수이다.

그림 18은 EF 클래스와 BE 클래스의 트래픽을 동

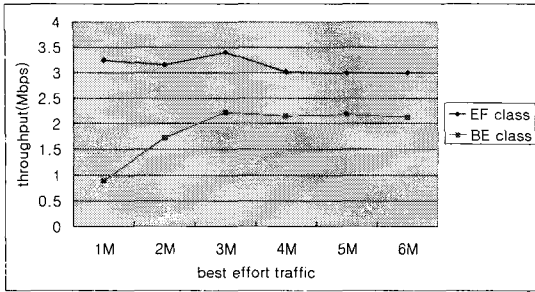


그림 16. 3M EF 클래스의 throughput

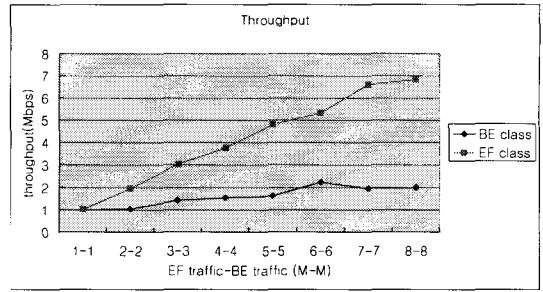


그림 18. throughput

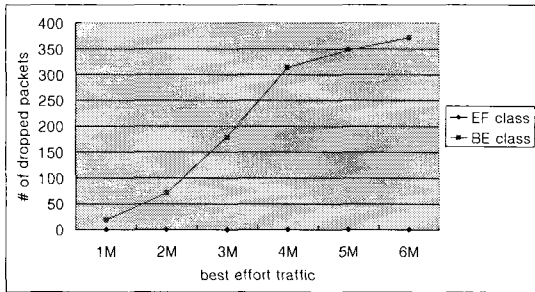


그림 17. 3M EF 클래스의 드롭된 패킷 수

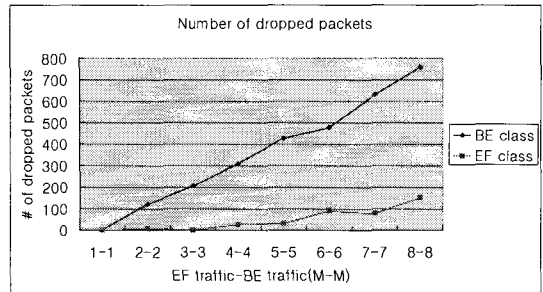


그림 19. 드롭된 패킷 수

일하게 1M부터 8M까지 증가시켰을 때의 throughput을 보여주고 있다. EF 클래스는 유입된 트래픽 양이 증가함에 따라서 수신된 트래픽의 양도 같이 증가한다. 단, 보장된 대역폭을 넘지 않는 범위에서는 송신단말에서 전송한 트래픽의 대부분을 보장 받는다. 그러나, 보장된 대역폭을 넘는 트래픽(실험에서는 8M인 경우)에 대해서는 보장된 대역폭 양인 7M까지만 보장하고 있음을 보여주고 있다. BE 클래스의 트래픽은 송신단말에서 전송한 트래픽이 2M를 넘는 경우부터는 전체적인 성능을 볼 때 수신단말에서는 약 2M 정도의 트래픽만 수신되고 있다.

그림 19는 EF 클래스와 BE 클래스의 트래픽을 동일하게 1M부터 8M까지 증가시켰을 때의 드롭된 패킷 수를 보여주고 있다. EF 클래스에서는 7M의 대역폭을 보장하고 있기 때문에, EF 클래스의 트래픽은 7M보다 작은 대역폭에 대해서는 큰 손실 없이 전송하고 있다. 그러나, BE 클래스의 트래픽은 트래픽 양이 증가함에 따라 드롭되는 패킷 수는 선형적으로 증가하고 있음을 보여주고 있다.

6. 결 론

본 논문에서는 PC 기반의 차별화 서비스 지원을

할 수 있는 테스트 베드 개발 연구를 수행하였다. 먼저 인터넷 상의 QoS 보장 모델인 차별화 서비스에 대한 기술을 소개하였다. 그리고 FreeBSD 상에서 ALTQ 시스템을 사용한 차별화 서비스 지원 라우터를 구현하고 테스트베드 상에서 검증을 수행하였다.

차별화 서비스 망에서 QoS를 보장받기 위해서 단말은 항상 대역 브로커와 사전에 SLA를 맺어 놓고, 서비스 시작할 때 대역 브로커에게 RAR 메시지를 전송한다. 그러면, 정책 서버인 대역 브로커는 자신이 가지고 있는 PIB 정보를 기반으로 각 라우터에게 Decision을 내린다. 물론 이때 대역 브로커와 각 라우터 사이의 모든 시그널링 메시지는 COSP 프로토콜을 이용해서 주고 받게 된다.

차별화 서비스가 지원되는 테스트베드에서 두 클라이언트가 VOD 서버에게 접속해서 비디오 서비스를 받을 때 한 사용자는 EF 클래스로 서비스를 받고, 다른 하나는 BE 클래스로 서비스를 받고, 망에는 적절한 백그라운드 트래픽을 통해 일정한 혼잡 상태이다. 이 때 실험 결과에서는 각 수신 단말에서의 성능과 중간 라우터에서의 드롭된 패킷 수를 비교하였다. 그 결과 EF 클래스로 서비스 받는 수신 단말은 처음 RAR에 의해서 할당 받은 대역 내에서 항상 QoS를 보장받고, BE 클래스는 항상 망 상황에 의존해서 망

이 혼잡이 발생하지 않을 때에는 성능이 좋지만 조그만 망에 혼잡이 발생해도 성능이 급격히 떨어짐을 볼 수 있었다.

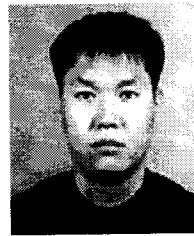
앞으로의 연구 방향은 다음과 같다. 첫 번째, 본 논문에서는 차별화 서비스에서 클래스를 EF 클래스와 BE 클래스, 이렇게 두 클래스로 나누어서 구현을 하였다. 앞으로는 이렇게 두 클래스가 아닌 중간에 AF 클래스에 대한 테스트가 더욱 필요하다. 특히, AF 클래스 같이 차별화 서비스의 미터에 의해서 패킷의 프로파일에 의해서 DSCP 값을 변화시키는 작업이 더욱 중요하다. 그러기 위해서 실제로 트래픽의 QoS를 보장해 주는 ALTQ에 대한 연구가 더욱 더 필요하다. 두 번째, 본 논문에서는 인트라 도메인 내에서 차별화 서비스에 대해서 구현했지만, 실제 망에 적용하기 위해서는 다른 사업자 망을 여러 개 넘어가는 환경 하에서 QoS를 보장하기 위해서 대역 브로커끼리의 시그널링 프로토콜에 대한 구현이 필요하다. 특히 다른 사업자 망 사이에서 SLA와 각 클래스에 대한 매핑이 더욱 중요하다.

참 고 문 헌

[1] S. Blake et al., "An Architecture for Differentiated Service," RFC 2475, December 1998.
 [2] K. Nichols et al., "Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers," RFC 2474, December 1998.
 [3] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999.
 [4] J. Heinane et al., "Assured Forwarding PHB Group," RFC 2598, June 1999.
 [5] S. Brim, B. Carpenter, F. L. Faucher, "Per Hop Behavior Identification Codes," RFC 2836, May 2000.
 [6] <http://irl.cs.ucla.edu/twotier/>
 [7] <http://www.ittc.ku.edu/~kdroa/845/>
 [8] <http://qbone.internet2.edu/bb/bboutline2.html>
 [9] D. Durham et al., "The COPS (Common Open

Policy Service) Protocol," RFC 2748, January 2000.

[10] K. Chan et al., "COPS Usage for Policy Provisioning (COPS-PR)," RFC 3084, March 2001.
 [11] <http://www.vovida.org/>
 [12] <http://www.csl.sony.co.jp/~kjc/altq-rel-projects.html>



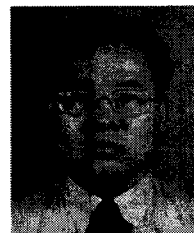
한 태 만

1985년 경북대 전자공학과(학사)
 2003년 충남대 컴퓨터학과(석사)
 1986년~1987년 삼성전자 통신 연구소
 1987년~1995년 LG정보통신 중앙연구소
 1995년~현재 한국전자통신연구원 네트워크연구소 스트리밍기술팀 선임연구원
 관심분야 : 인터넷 QoS, 멀티캐스트 기술, 트랜스코딩 기술, 스트리밍 프로토콜 기술, HDTV 콘텐츠 제작 기술



김 동 원

1983년 경북대학교 전자공학과(공학사)
 1990년 경북대학교 전자공학과(공학석사)
 1998년 충북대학교 전자공학과(공학박사)
 1983년~1998년 한국전자통신연구원 선임연구원
 1998년~현재 충북과학대학 정보통신과학과 교수
 관심분야 : 스트리밍 기술, 무선 인터넷, NGcN



정 유 현

1980년 광운대 전자계산학과(학사)
 1989년 광운대 대학원 컴퓨터학과(석사)
 1998년 광운대 대학원 컴퓨터공학과(박사)
 1980년~현재 한국전자통신연구원, 네트워크연구소 스트리밍기술팀 팀장, 책임연구원
 관심분야 : 인터넷 QoS, 광인터넷, 이미지 트랜스코딩 기술, 음성정보처리(음성인식 & 합성) 기술



이 준 화

2000년 충남대 컴퓨터과학과(학사)
2002년 충남대 대학원 컴퓨터과
학과(석사)
2002년~현재 충남대 대학원 컴
퓨터학과 박사과정

관심분야 : 인터넷 QoS, 이동 통신



김 상 하

1980년 서울대학교 화학과(학사)
1984년 U. of Houston 화학과(석사)
1989년 U. of Houston 전산학과
(박사)
1989년 HNSX Supercomputers
Inc. 자문위원
1990년~1991년 시스템공학 연구

소 선임연구원

1992년~현재 충남대학교 컴퓨터 과학과 교수
관심분야 : 컴퓨터 네트워크, 이동통신, 분산 컴퓨팅