

차등 서비스를 위한 EJB 기반 QoS 관리 시스템 프레임워크의 설계 및 구현

정회원 차 시 호*, 이 종 언**, 안 병 호***, 조 국 현**

Design and Implementation of EJB based QoS Management System Framework for Differentiated Services

Si ho Cha*, Jong eon Lee**, Byung ho Ahn***, Kuk hyun Cho** *Regular Members*

요 약

차등 서비스 모델은 개별적인 플로우들 보다는 집합적인 트래픽 클래스들을 지원하므로 IP 망 내의 다양한 패킷 클래스들에 대하여 다양한 QoS를 제공할 수 있다. 그럼에도 불구하고, 차등 서비스 모델은 완전한 QoS 관리 프레임워크와 그의 구현 모델을 제공하지는 못한다. 차등 서비스 망에서의 QoS 보장 서비스를 제공하기 위해서는 망 및 서비스 관리의 지원이 필수적으로 필요하다. 따라서, 본 논문은 차등 서비스를 지원하기 위하여 EJB 기반 QoS 관리 시스템을 제안하고 구현한다. 구현되는 QoS 관리 시스템은 상위 레벨 QoS 정책들은 유효한 XML 문서로 표현되며 하위 레벨 QoS 정책들인 EJB 기반 정책 서버 내의 EJB 빈들로 사상된다. 하위 레벨 QoS 정책들은 SNMP 차등 서비스 MIB를 통해 차등 서비스 라우터들로 분배되고 감시된다.

Key Words : DiffServ, QoS, Policy-Based Management, EJB, QoS Policy.

ABSTRACT

DiffServ supports aggregated traffic classes to provide various QoS to different classes of traffics. However, current DiffServ specifications do not have a complete QoS management framework. It is possible to lead to serious QoS violations without a QoS management support. From this reasoning, a QoS management system that can manage differentiated QoS provisioning is required. This paper proposes and implements a policy-based QoS management platform for differentiated services networks, which specifies QoS policies to guarantee dynamic QoS requirements. The implementation of the proposed platform is built on EJB framework and uses XML to represent and validate high-level QoS policies. High-level QoS policies are represented as valid XML documents and are mapped into EJB beans of the EJB-based policy server of the platform. The policy distribution and the QoS monitoring are processed using SNMP.

I. 서 론

최선형(best-effort) 서비스 모델은 트래픽을 가능한 한 빠르게 처리하지만, QoS를 보장하기 위한 어

떠한 처리도 제공하지 않는다. 이러한 문제를 해결하기 위하여 IETF(Internet Engineering Task Force)는 통합 서비스(IntServ) 모델[1]과 차등 서비스(DiffServ) 모델[2]을 제시하였다. IntServ의 단점

* 세종대학교 컴퓨터공학과(sihoc@sejong.ac.kr), **광운대학교 컴퓨터과학과(ljelee, khcho)@cs.kw.ac.kr

*** 충청대학 컴퓨터학부(bhahn@ok.ac.kr)

논문번호 : KICS2004-12-339, 접수일자 : 2005년 3월 11일

* 이 논문은 2004년도 광운대학교 연구년에 의하여 연구되었음.

은 플로우마다 상태 정보를 필요로 하며 코어 IP 망 내에서의 QoS 처리로 인하여 심각한 규모성 (scalability)의 문제들이 야기된다는 것이다[3]. 그러나, 차등 서비스 모델은 개별적인 플로우들 보다는 집합적인 트래픽 클래스들을 지원하므로 IP 망 내의 다양한 패킷 클래스들에 대하여 다양한 QoS를 제공할 수 있다. 그럼에도 불구하고, 차등 서비스 모델은 완전한 QoS 관리 프레임워크와 구현 모델을 제공하지는 못한다. 차등 서비스 망에서의 QoS 보장 서비스를 제공하기 위해서는 망 및 서비스 관리의 지원이 필수적으로 필요하다. 이런 지원이 없는 QoS를 제공하는 것이 불가능할 것이다. 최근에 들어 정책 기반 관리(Policy-Based Management) 기법이 차등 서비스망의 QoS를 제공할 수 있는 기술로 인식되고 있다. 정책 기반 QoS 관리의 목적은 망 장비들과 망 형상에 가능한 독립적인 방법으로 망의 행위를 기술하는 상위 레벨 QoS 정책들을 사용하여 QoS를 관리하는 것이다[4]. 하나의 QoS 정책이 특정 망 장비에만 국한되는 것이 아니라, 다양한 망 장비들에 사용될 수 있고, 단지 상위 레벨의 QoS 정책만을 수정함으로써 고객의 동적인 SLA (Service Level Agreement) 변경을 망에 동적으로 적용할 수 있기 때문에, QoS 관리를 위해 상위 레벨의 정책을 사용하는 방법은 효율적이고 동적인 관리를 가능하게 해준다. 현재 정책 기반 관리 시스템을 사용하여 QoS를 보장하기 위한 여러 연구들이 진행 중에 있지만, 정책 기반 관리 개념을 갖는 QoS 관리 플랫폼의 설계와 구현에 대한 연구와 정책 변환 기법에 대한 연구는 아직 부족한 실정이다.

본 논문은 차등 서비스를 지원하는 IP 기반 망에서 QoS를 효율적으로 관리하기 위한 정책 기반 QoS 관리 모델을 제안하고, 이 모델을 따르는 EJB(Enterprise JavaBeans) 기반 QoS 관리 시스템(PMQoSDS)을 설계하고 구현한다. 구현되는 PMQoSDS는 EJB 프레임워크를 사용하여 설계되고 구현되며, XML(Extensible Markup Language)을 사용하여 상위 레벨 QoS 정책들을 정의한다. PMQoSDS는 분산 컴포넌트 기반 기술인 EJB 프레임워크를 사용함으로써 개발 비용과 개발 시간을 줄일 수 있고 플랫폼의 유지보수, 확장성, 분산 트랜잭션, 안정성 등을 향상시킬 수 있다. 또한 상위 레벨 QoS 정책 정의에 XML을 사용함으로써 XML의 많은 유용한 파서와 유효성 분석기들을 용이하게 사용할 수 있으며, XML의 정보 표현의 융통성으로 인해 새로운 유형의 정보를 쉽게 추가하고 변경할 수 있다[5].

IETF 정책 기반 관리 구조에서 정책 분배를 위한 표준 프로토콜은 COPS(Common Open Policy Service)이다[6]. 하지만, 일반적인 차등 서비스 라우터들은 SNMP는 기본적으로 서비스 할 것이지만, COPS는 지원하지 않을 것이기 때문에, 현재의 PMQoSDS의 구현은 정책의 분배와 모니터링을 위해 SNMP를 사용함으로써 COPS를 지원하지 않는 차등 서비스 라우터들도 정책 기반으로 관리할 수 있다. 하지만 PMQoSDS는 컴포넌트 기반 플랫폼을 채택하고 있기 때문에 COPS를 지원하는 라우터들을 위해 COPS도 쉽게 추가될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 기술에 대하여 살펴보고, 3장에서는 제안된 PMQoSDS의 구조와 컴포넌트 그리고 QoS 관리절차에 대하여 자세히 기술한다. 4장에서는 리눅스 기반 차등 서비스 라우터와 PMQoSDS의 구현을 설명하고 비디오 스트리밍 시스템에서의 실험 결과에 대하여 살펴본다. 마지막으로 5장에서는 본 논문의 요약 및 향후의 연구 과제에 대하여 기술한다.

II. 관련 연구

2.1 차등 서비스

차등 서비스는 효율적이고 규모성있게 QoS를 제공할 수 있는 기술이다. 차등 서비스는 개별적인 플로우들 보다는 집합적인 트래픽 클래스들을 지원하므로 IP 망 내의 다양한 패킷 클래스들에 대하여 다양한 QoS를 제공할 수 있다. 차등 서비스는 차등 서비스 코드 포인트(DSCP) 값에 따라 다양한 PHB (Per Hop Behavior)들의 수행에 의해 집합적인 IP 패킷 스트림들에 대해 차등화된 서비스를 제공한다. 즉, 에지 라우터는 IP 패킷들에 차등 서비스 코드 포인트인 DSCP 값을 설정하고 코어 라우터는 에지 라우터가 설정한 DSCP 값에 따라 해당 패킷들에 적절한 PHB를 수행함으로써 집합적인 IP 패킷 스트림들에 서비스 차등화를 제공한다.

에지 라우터의 주요 구성 요소는 MF(multi-field) 분류기(traffic classifier)와 트래픽 조건 블록(TCB: traffic conditioner block)로 구성된다. MF 분류기는 발신지 주소, 목적지 주소, DS 필드, 프로토콜 ID, 발신지 포트 번호, 목적지 포트 번호 등 하나 혹은 둘 이상의 헤더 필드를 조합한 값으로 패킷을 분류한다. TCB는 트래픽 측정기(meter), 마커(marker)와 셰이퍼(shaper)등의 기능 블록으로 구성된다. MF 분류기에 의해 분류된 트래픽은 TCB에서 분류에

따른 정책 요구 사항들을 만족시키기 위하여 셰이핑을 수행하기 위해 특정 형태의 접속 수락(admission) 필드를 통과하게 된다. 그 다음 셰이핑된 트래픽 스트림은 IP 패킷에 DSCP 값을 설정함으로써 특정한 행위 집합에 할당된다. 예지 라우터를 통과한 스트림은 마킹된 DSCP 값에 따라 코어 라우터의 BA(behavior aggregate) 분류기에 의해서 분류되고 PHB가 적용된다. DSCP는 단순히 IP 헤더 내의 차등 서비스(DS) 필드의 6 비트 값을 의미한다. 이 값은 해당 패킷에 제공되어야 할 특정한 PHB를 나타낸다. 현재의 차등 서비스 망에는 디폴트 서비스(best-effort), 클래스 선택자(CLS), 신속 전송(EF), 보장 전송(AF)의 4가지의 PHB가 사용되고 있다.

2.2 정책 기반 관리

정책 기반 QoS 관리의 목적은 망 장비들과 망 형상에 대해 가능한 독립적인 방법으로 망의 행위를 기술하는 상위 레벨 QoS 정책들을 사용하여 QoS를 관리하는 것이다. 하나의 QoS 정책이 특정 망 장비에만 국한되는 것이 아니라, 다양한 망 장비들에 사용될 수 있고, 단지 상위 레벨의 QoS 정책만을 수정함으로써 변경 사항을 망에 동적으로 적용할 수 있기 때문에, QoS 관리를 위해 정책을 사용하는 방법은 효율적이고 동적인 관리를 가능하게 해준다. 정책 기반 관리의 주요 기능은 그래픽 사용자 인터페이스, 자원 검출, 변환 로직과 정책 배치자의 네 가지 요소로 구성된다. 그래픽 사용자 인터페이스는 망 및 서비스 관리자가 차등 서비스 망으로 QoS를 배치하기 위한 비즈니스 SLA들을 입력할 수 있다. 고객들을 위해 차등 서비스 망이 제공해야 할 각 서비스에 대한 성능이 SLA로 정의되고 정의된 SLA와 자원 검출 로직이 검출한 차등 서비스 망의 형상과 라우터 정보들에 따라 상위 레벨 QoS 정책이 정의된다. 변환 로직은 상위 레벨 QoS 정책들을 다양한 망 장비들에 배치될 수 있는 장비 구성 정보인 하위 레벨 QoS 정책들로 변환시키는 역할을 수행한다. 정책 배치자는 장비 구성 정보인 하위 레벨 QoS 정책들이 차등 서비스 망 내의 다양한 장비들로 분산되도록 보장하는 역할을 수행한다[7].

2.3 EJB 기반 망 관리

EJB 기술에 기반한 애플리케이션 구조는 다 계층 애플리케이션들의 개발을 단순화시켜줄 뿐만 아니라, 트랜잭션 관리, 생명 주기 관리, 자원 풀링

(pooling)과 같은 다 계층 엔터프라이즈 애플리케이션들에 내재되어 있는 다양한 복잡 요소들을 사용자 인터페이스와 비즈니스 로직 간에 분리시킴으로써 복잡성을 감추고 이식성을 향상시킨다[8]. EJB 기술의 주요 컴포넌트는 EJB 컴포넌트, EJB 컨테이너 그리고 EJB 커넥터로 구성된다. EJB 컴포넌트는 많은 애플리케이션들에 의해 재사용될 수 있는 기능을 캡슐화한 자바 클래스이다. EJB 컨테이너는 EJB 컴포넌트들의 수행 환경을 제공한다. EJB 커넥터는 EJB 컴포넌트들이 기존의 시스템들로부터 데이터를 액세스 할 수 있도록 해주는 메카니즘을 제공한다. EJB 기술은 세션 빈(Session Beans), 엔티티 빈(Entity Beans) 그리고 메시지 방식 빈(Message-driven Beans)과 같은 별개의 EJB 컴포넌트 유형들을 정의함으로써 망 및 서비스 관리에 필요한 모든 객체들을 생성할 수 있는 능력을 제공한다. 세션 빈들은 클라이언트 세션들과 관련된 행위들을 나타내며 비영속적인 객체를 의미한다. 엔티티 빈들은 데이터 저장소에 유지되는 데이터나 공유 자원을 나타내는 영속적인 객체를 의미한다. 메시지 방식 빈들은 서버 상에서 비즈니스 로직을 구현하는 JMS(Java Message Service)의 소비자들을 나타낸다[8]. 이러한 특성으로 인하여, EJB 기반 관리 프레임워크는 비즈니스 로직의 캡슐화, 애플리케이션 개발의 단순화, 쉬운 분산 객체(데이터베이스, 디렉토리) 접근, 규모성, 일관성, 영속성, 트랜잭션 관리, 프로토콜과 정보 모델의 변환, 그리고 발행-구독 메시지 전달등과 같은 장점들을 제공할 수 있다.

III. PMQoSDS 프레임워크 설계

본 장에서는 EJB 기반 QoS 관리 시스템인 PMQoSDS를 제안하고 설계한다. 이를 위하여 PMQoSDS의 프레임워크 구조와 기능 구조, 그리고 QoS 정책에 대하여 기술한다.

3.1 프레임워크 구조

차등 서비스 망을 위한 정책 기반 QoS 관리 프레임워크인 PMQoSDS의 전체적인 구조는 그림 1과 같다. PMQoSDS 프레임워크는 모델-뷰-컨트롤러(MVC) 구조를 사용하여 설계되었다. PMQoSDS 프레임워크는 크게 웹 서버와 EJB 기반 정책 서버로 구성된다. 웹 서버는 PMQoSDS 플랫폼의 표현을 담당하고, EJB 기반 정책 서버는 비즈니스 로직을 담당한다. PMQoSDS 프레임워크에서는 차등 서

비스 망 상에서 수행될 상위 레벨의 QoS 정책들을 생성하고 관리하기 위한 사용자 인터페이스로 웹 브라우저를 사용한다.

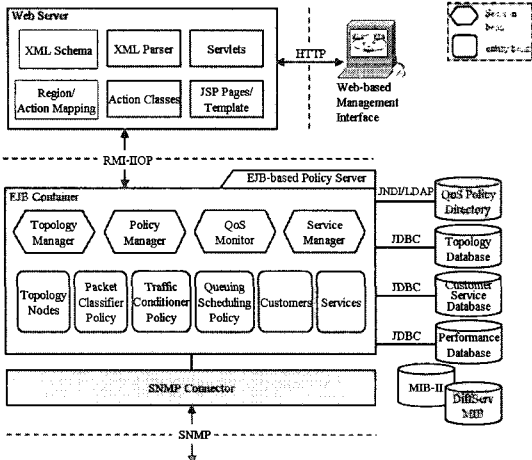


그림 1. PMQoS의 QoS 관리 구현 구조
Fig 1. Implemental QoS management architecture of PMQoS

EJB 기반 정책 서버를 구성하는 관리 요소들은 세션 빈으로 구현되며, 크게 네트워크의 형상과 라우터의 역할을 검출하는 형상 관리자(TM: Topology Manager), 하위 레벨 QoS 정책을 설정하고 이 정책들을 라우터들로의 배치하는 정책 관리자(PM: Policy Manager), 라우터에 적용된 QoS 정책의 준수 여부를 감시하는 QoS 관리자(QM: QoS Monitor), 고객 서비스 관리를 수행하는 서비스 관리자(SM: Service Manager)로 구성된다.

또한 형상 노드(TN : Topology Node)는 망의 라우팅 형상과 라우터의 역할 정보를 포함하는 엔티티 빈으로, 이들에 대한 정보는 SNMP MIB-II를 사용하여 검출된다. 패킷 분류자(PC : Packet Classifier) 정책 빈은 패킷 흐름들 내의 패킷들을 분류하고 클래스 식별자(CID)라 불리는 레이블을 패킷들에 할당하는 엔티티 빈이다. PC 빈의 행위는 에지 라우터들로 분배되어 차분화 서비스 망으로 입력되는 트래픽을 제어하게 된다. 트래픽 조절자(TC : Traffic Conditioner) 정책 빈은 패킷 분류자에서 선별된 패킷들이 트래픽 프로파일에 적합한지 측정하고 이 결과에 따라 해당 패킷을 마킹하고, 폐기하거나 웨이핑하는 엔티티 빈이다. TC 빈의 행위는 에지 라우터들로 분배되어 에지 라우터로부터 출력되는 트래픽을 제어하게 된다. 큐잉과 스케줄링(QS : Queuing & Scheduling) 정책 빈은 패킷들을 큐잉

하고, 스케줄링하거나 폐기하는 행위를 수행하는 엔티티 빈이다. QS 빈의 행위는 코어 라우터들로 분배되어 코어 라우터로부터 출력되는 트래픽을 제어하게 된다.

3.2 기능 구조

PMQoS는 정책 기반 QoS 관리를 효율적으로 처리하기 위하여 QoS 관리 절차를 정의하고 있다. 그림 2는 이러한 PMQoS의 개념적인 QoS 관리 구조를 그림으로 보인 것이다.

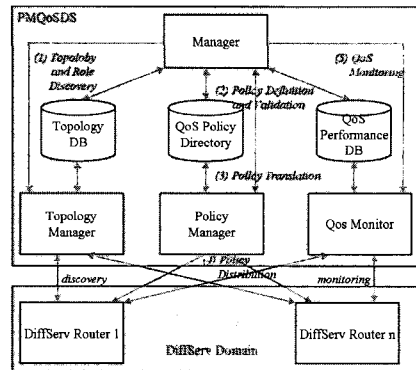


그림 2. PMQoS의 개념적 관리 구조
Fig 2. Conceptual QoS management architecture of PMQoS

이러한 QoS 관리 절차는 아래와 같이 형상 검출 단계, 정책 정의와 검증 단계, 정책 변환 단계, 정책 배치 및 QoS 감시 단계로 구성된다.

1. 형상 검출. 망의 QoS를 관리하기 위해서는 우선적으로 차등 서비스 망의 라우팅 형상 정보와 망 내의 모든 라우터들의 역할 정보가 필요하다. 형상 관리자(TM) 세션 빈은 SNMP를 사용하여 라우팅 형상 정보와 라우터의 역할 정보를 검출한다. 검출된 라우팅 형상 정보와 라우터의 역할 정보는 형상 데이터베이스(TD)에 저장되며 EJB 기반 정책 서버의 EJB 컨테이너 내에 형상 노드(TN) 엔티티 빈들로 표현된다. 이러한 일련의 작업들은 형상 관리자(TM) 세션 빈에 의해 관리된다. 라우팅 형상 정보와 라우터의 역할 정보는 두 개의 SNMP MIB-II 테이블(ipAddrTable과 ipRouteTable)을 사용함으로써 검출된다. 라우팅 형상 정보는 ipRouteTable을 사용함으로써 얻을 수 있고, 라우터의 역할 정보는 ipAddrTable을 검색함으로써 얻을 수 있다. ipAddrTable은 라우터 내의 모든 망 인터페이스의 IP 주소를 포함하

며, ipRouteTable은 목적지 IP 주소들을 위한 다음으로 연결된 호스트와 망 인터페이스를 갖는 IP 라우팅 테이블을 포함한다.

2. 정책 정의와 검증: PMQoSs는 상위 레벨 QoS 정책들의 정보를 검증하여 차등 서비스 망의 장비들을 구성하기 위한 하위 레벨 QoS 정책들로 변환시킨다. 이를 위하여 PMQoSs는 관리자가 입력하는 비즈니스 SLA들을 XML 스키마를 갖는 XML QoS 정책 문서들로 정의한다. PMQoSs는 XML 문서로 표현된 상위 레벨 QoS 정책들의 유효성 검증을 수행하기 위하여 XML 스키마를 사용한다. PMQoSs의 웹 서버 내의 자바 서블릿은 관리자로부터 QoS 정책 데이터를 입력 받아 유효한 XML 문서를 생성한 후 XML 스키마를 이용하여 유효성 검증을 수행한다.
3. 정책 변환: 상위 레벨 QoS 정책을 하위 레벨 QoS 정책으로의 정책 변환은 PMQoSs의 EJB 기반 정책 서버 내에 있는 정책 관리(PM) 세션 빈이 수행한다. 자바 서블릿이 정책의 유효성 검증을 완료한 후에는 PM 세션 빈에게 하위 레벨 QoS 정책 엔터티 빈들인 패킷 분류(PC) 엔터티 빈, 트래픽 조절(TC) 엔터티 빈, 큐잉 및 스케줄링(QS) 엔터티 빈의 인스턴스를 생성하도록 요청한다. PM 세션 빈은 하위 레벨 QoS 정책 엔터티 빈들의 인스턴스를 생성하고 그 인스턴스들이 갖는 변수 속성 값들을 적절히 설정함으로써 정책 변환을 수행한다. 따라서 XML로 표현된 상위 레벨 QoS 정책들은 실제 차등 서비스 라우터들의 기능을 설정하기 위해 요구되는 장비 구성 파라미터들을 속성 값으로 갖는 EJB 컴포넌트들로 변환된다.
4. 정책 배치: 정책의 배치는 PM 세션 빈과 하위 레벨 QoS 정책 엔터티 빈들인 PC 빈, TC 빈, 그리고 QS 빈을 사용하여 이루어진다. PC 정책 빈은 패킷의 흐름들을 분류하여 그들에 분류 식별자를 할당하고, TC 정책 빈은 패킷들을 측정하여 그들에 마킹, 카운팅, 그리고 폐기 등의 행위를 수행한다. QS 정책 빈은 패킷들을 큐잉하고, 스케줄링하거나 폐기하는 행위를 수행한다. PM 세션 빈은 이러한 하위 레벨 QoS 정책 엔터티 빈들을 사용함으로써 정책 변환을 수행하고, 정책 배치를 수행하게 된다. 하위 레벨 QoS 정책 빈들은 SNMP 오피

레이션을 사용하여 차등 서비스 MIB에 대한 적절한 값의 설정과 모니터링을 수행해야 한다. PM 세션 빈은 PC 정책 빈과 TC 정책 빈 그리고 QS 정책 빈으로 하여금 SNMP를 사용하여 정책 배치를 수행하도록 한다. PC 엔터티 빈은 차등 서비스 망으로 진입하는 트래픽을 제어하기 위해 에지 라우터들로 배치되고 TC 엔터티 빈은 에지 라우터로부터 출력되는 트래픽을 제어하기 위하여 에지 라우터들로 배치된다. 또한 QS 엔터티 빈은 코어 라우터로부터 출력되는 트래픽을 제어하기 위하여 코어 라우터들에 배치된다.

5. QoS 감시: 차등 서비스 라우터들로 배치된 QoS 정책들은 정책 정의에 기술된 내용에 맞게 동작하지 않을 수 있다. 즉 정책 배치로부터 발생된 QoS의 결과가 실제 정책과 다를 수 있다. 따라서 망 관리자는 QoS 정책 배치로부터 발생된 QoS가 정책에 맞게 동작하는지를 감시해야 한다. PMQoSs에서의 QoS 감시도 정책 배치와 같이 SNMP 차등 서비스 MIB를 사용한다. QoS 감시자(QM) 세션 빈은 PC 정책 빈과 TC 정책 빈 그리고 QS 정책 빈에 정의된 정책들에 접근하여 실제 망으로부터 모니터링된 결과를 비교하게 된다. 만일 정책에 정의된 만큼의 QoS를 제공하고 있지 못하면, QM 세션 빈은 망 관리자에게 해당 메시지를 발생시키고 성능 데이터베이스를 갱신시키게 된다.

3.3 QoS 정책

QoS 관리 정책은 상위 레벨 QoS 정책과 하위 QoS 레벨 정책의 두 관점으로 표현된다. 상위 레벨 QoS 정책은 망 관리자에 의해 조작되는 비즈니스 수준의 SLA와 대응되고, 하위 레벨 QoS 정책은 개개의 장비 구성에 대응된다. 상위 레벨 QoS 정책은 망 관리자에 의해 조작되지만, 하위 레벨 QoS 정책은 논리적 컴포넌트에 의하여 생성된다. 상위 레벨 QoS 정책은 발신지/목적지 그룹, 서비스 그룹, 라우터 그룹, 그리고 시간 그룹들로 그룹화될 수 있다. 발신지 요소는 IP 주소나 도메인 이름으로 기술되고 하나의 발신지 그룹은 IP 주소들에 대한 하나의 그룹을 의미한다. 서비스 요소는 서비스가 사용하는 TCP나 UDP 포트의 번호에 의해 특성화된다. 시간 요소는 정책의 활성화 시간을 정의한다. 서비스 요소는 Premium, Gold, Silver, Bronze와 같은 서비

스 클래스의 집합이다. Premium 서비스는 차등 서비스의 EF PHB로 사상되고 Gold, Silver는 AF PHB 그룹으로 사상된다. Bronze 서비스는 BE 전승으로 사상된다. PMQoSDS 시스템에서의 하위 QoS 레벨 정책은 PC 정책 빈, TC 정책 빈, QS 정책 빈들로 구성된다. 상위레벨에서 하위 레벨의 정책 변환은 PM 세션 빈에 의하여 수행된다. 자바 서블릿은 관리자로부터 정책 데이터를 입력으로 받아들이며 XML 스키마를 따르는 XML 문서를 생성한 후 그 XML 스키마를 이용하여 유효성 검증을 수행한다. 자바 서블릿이 정책의 유효성 검증을 완료한 후에는 PM 세션 빈 인스턴스에 하위 레벨 QoS 정책 엔터티 빈들의 인스턴스를 생성할 것을 요청하게 된다.

다음은 발신지 주소가 192.168.1.2인 고객에게 Premium 서비스를 제공하는 가장 단순한 정책의 예이다.

```
policy1: IF (Source-IP-Address is 192.168.1.2)
    THEN serviceLevel is Premium;
```

앞에서 보인 정책의 예는 다음과 같은 유효한 XML 정책 문서로 표현될 수 있다.

```
<policy>
<id>1</id>
<sourceGroup>
<source>192.168.1.2</source>
</sourceGroup>
<destinationGroup>
    <destination>any</destination>
</destinationGroup>
<routerGroup>
    <router>any</router>
</routerGroup>
<applicationGroup>
    <application>any</application>
</applicationGroup>
<timeGroup>
    <time>
        <startTime>any</startTime>
        <endTime>any</endTime>
    </time>
</timeGroup>
<serviceLevel>Premium</serviceLevel>
</policy>
```

IV. 구현

본 장에서는 리눅스 기반 차등 서비스 라우터의 구축과 차등 서비스 MIB를 갖는 SNMP 관리 대행자의 구현에 대하여 기술한다. 또한 PMQoSDS의 프로토타입 시스템을 구현하고, 구체적인 적용 타당성을 검증하기 위하여 리눅스 기반 차등 서비스 망에 적용해 봄으로써 시스템의 실행 결과 등을 분석한다.

4.1 리눅스 기반 차등 서비스 라우터 구축

차등 서비스 라우터는 리눅스 시스템을 사용하여 설정하였으며 MIB-II와 차등 서비스 MIB를 처리하기 위한 SNMP 에이전트를 각각의 라우터에 추가하였다. 리눅스 시스템에서 차등 서비스를 위한 지원은 이미 커널 버전 2.3에서 통합되었기 때문에 커널을 재구성하고 재컴파일 함으로써 지원 가능한 라우터를 생성할 수 있다[9]. 차등 서비스 MIB를 지원하는 SNMP 에이전트는 에이전트 확장성을 제공하는 UCD-SNMP 4.2.5 [10]를 이용하여 구현하였다. 이 SNMP 에이전트와 리눅스 트래픽 제어 커널 간의 통신은 netlink 소켓을 사용하였다[11]. Netlink는 커널 레벨과 사용자 공간 프로세서 간의 정보 전달을 위해 사용되는 양방향 통신 링크이다. 이렇게 구현된 SNMP 에이전트는 EJB 기반 정책 서버 내의 하위 레벨 정책 빈들로부터 관리 오퍼레이션을 수신하여 리눅스 트래픽 제어 커널 내의 적절한 파라미터 변경을 수행한다. 이런 오퍼레이션의 결과로써 QoS 정책들이 PEP인 리눅스 차등 서비스 라우터들로 분배되고 이들의 결과에 대한 모니터링이 가능하게 된다.

4.2 PMQoSDS 프레임워크

PMQoSDS 프레임워크는 Windows 2000 서버 플랫폼 상에서 구현되었다. PMQoSDS는 웹 서버와 EJB 기반의 정책 서버로 구성된다. 웹 서버는 PMQoSDS의 표현 계층의 로직을 수행하고, EJB 기반의 정책 서버는 PMQoSDS의 비즈니스 계층의 로직을 수행한다. MVC 구조 중 뷰에 대한 구현은 JSP 템플릿 메커니즘과 컴포지트 뷰 패턴을 적용하여 구현하고, 제어는 프론트 컨트롤러 패턴(액션 서블릿)과 세션 패사드 패턴(EJB 세션 빈)을 사용하여 구현된다. 모델은 EJB 엔터티 빈과 서비스 로케이터 패턴을 이용하여 구현된다[12].

서블릿과 JSP 컨테이너로 Apache Tomcat 4.0.1

[13]을 사용하였다. 각 QoS 정책 빈들을 관리하기 위한 EJB 컨테이너/서버로는 JBoss 2.4.10[14]을 사용하고 모든 관리 빈들은 EJB 버전 1.1 표준을 사용하여 구현되었다. 또한, EJB 기반 정책 서버의 SNMP 오퍼레이션을 처리하기 위하여 자바로 작성된 AdventNet SNMP API [15]를 사용하였다. MIB 테이블로부터 얻어진 성능 정보 및 형상 정보 등을 저장하기 위한 데이터 베이스로는 Oracle 8i Enterprise Edition 8.1.6버전[16]을 사용하여 구현하였다.

PMQoSDS의 웹 계층 ActionServlet에 의해 매핑된 액션들은 EJB 엔터티 빈들 각각에 대한 참조가 아닌 세션 패사드에 대한 하나의 원격 참조를 가지고 있다. 그림 3은 PMQoSDS 플랫폼의 비즈니스 계층에서의 주요 정책 관리 절차를 보여주는 UML 순차 다이어그램이다.

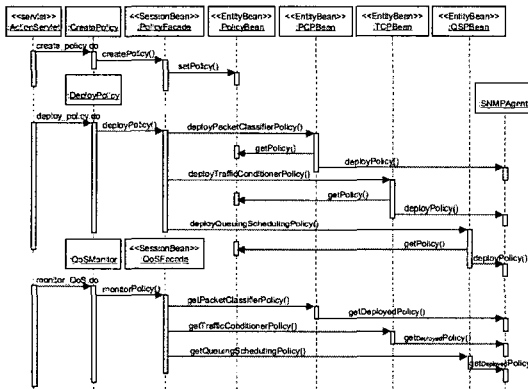


그림 3. UML 순차 다이어그램
Fig 3. UML sequence diagram

그림 3에서 보인 것과 같이 유스케이스에 따라 정책을 생성하고 배치하기 위한 PolicyFacade와 QoS를 감시하기 위한 QoSFacade를 구현하고 있다. PolicyFacade는 새로운 정책을 생성하기 위한 유스케이스와 생성된 정책을 차등 서비스 망으로 배치시키기 위한 유스케이스를 구현한다. QoSFacade는 배치된 정책이 정확히 망 장비에 설정되었는지를 검사하기 위한 유스케이스를 구현한다. 그러나 세션 패사드와 상호 작용하는 웹 계층에서는 세션 빈 (PolicyFacade와 QoSFacade, TopologyFacade)을 찾아내는 작업이 필요하다. 이를 위해 비즈니스 델리 게이트(Business Delegate) 패턴을 이용하여 웹 계층의 결합도를 더욱 낮추기 위해 웹 계층과 세션 패사드 사이에서 위임 객체처럼 행동하는 세 개의 일반 자바 클래스를 구현하였다. 따라서 PMQoSDS

시스템은 PolicyFacade를 위한 PolicyDelegate와 QoSFacade를 위한 QoSDelegate, 그리고 TopologyFacade를 위한 TopologyDelegate를 구현하고 있다. 이러한 위임 클래스들의 역할은 단지 패사드에 전송해야 할 인터페이스를 찾아내는 기능을 수행한다.

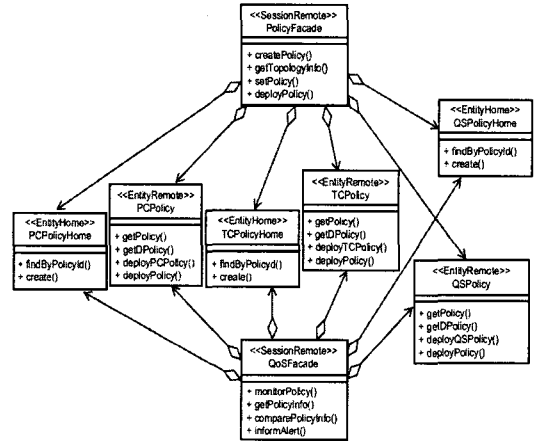


그림 4. UML 클래스 다이어그램
Fig 4. UML class diagram

그림 4는 정책의 생성, 배치, 감시와 관련된 모델 구현에 필요한 주요 EJB 빈들에 대한 UML 클래스 다이어그램을 보인 것이다. PMQoSDS의 모델은 엔터티 빈들로 구현된다. 현재 모델을 구현하기 위해 사용되는 엔터티 빈들은 EJB 컨테이너에 의해 자동적으로 영속성이 관리되는 CMP(Container Managed Persistence) 엔터티 빈들로 구현되었다. PolicyFacade 세션 빈에 의한 새로운 정책의 정의 및 배치를 위한 유스케이스에 대한 모델을 구현하기 위해 PacketClassifierPolicy 엔터티 빈과 TrafficConditionerPolicy 엔터티 빈, 그리고 QueuingSchedulingPolicy 엔터티 빈이 구현되었다. 이들 엔터티 빈들의 정책의 배치와 검사는 DiffServ MIB에 대한 SNMP Set 오퍼레이션과 Get 오퍼레이션을 통해 이루어진다. 이들 엔터티 빈들은 정책을 배치하기 위해 NodeList 엔터티 빈과 Policy 엔터티 빈을 참조한다. Policy 엔터티 빈은 배치할 정책에 대한 정보를 제공한다. 현재 구현된 Policy 엔터티 빈은 정책의 ID, 발신지 IP 주소, 목적지 IP 주소, 서비스 유형, 정책의 시작 시각과 종료 시각, 서비스 등급 등과 같은 정책에 대한 여섯 개의 필드를 가지고 구현되었다.

그림 5는 상위 레벨 정책 정보의 입력 폼과 정책 생성 명령의 결과 화면을 보인 것이다.

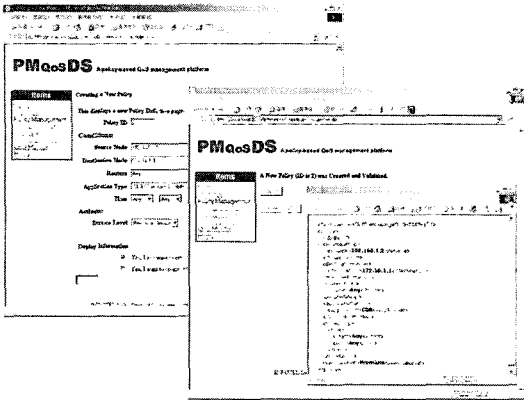


그림 5. 상위 레벨 QoS 정책 생성 화면
Fig 5. Screenshot of Creation High Level QoS Policy

4.3 실험 및 평가

PMQoS DS의 유효성을 입증하기 위하여 H.263 기반의 비디오 스트리밍 시스템[17]을 리눅스 기반 차등 서비스 망에 적용하였다. 실험하기 위해 테스트베드로 구축된 리눅스 기반 차등 서비스 망의 구성은 그림 6과 같다.

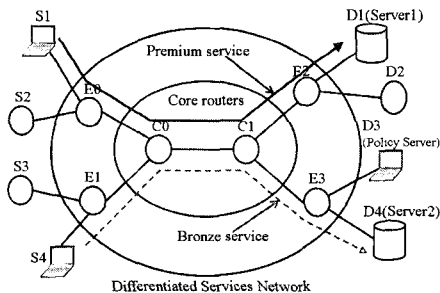


그림 6. 실험 환경
Fig 6. Experiment Environment

두 개의 VOD 서버가(D1, D4)가 테스트베드망의 에지 라우터(E2, E3)에 연결되어 있다. 두 개의 VOD 서버는 같은 하드웨어와 소프트웨어 환경으로 구성되어있다. 정책 서버는 E3를 통하여 망에 연결되어있다. 실험 테스트베드의 하드웨어 구성은 다음과 같다. 코어 라우터들은 펜티엄 IV 1.8GHz의 CPU와 512MB 메인 메모리를 갖고 에지 라우터들은 펜티엄 IV 1.5GHz의 CPU와 512MB 메인 메모리를 갖는다. 두 개의 VOD 서버는 펜티엄 IV 2.0GHz CPU와 512MB 메인 메모리를 갖는다. 그 외의 나머지 시스템은 펜티엄 III 1.0GHz CPU와 256MB 메인 메모리를 갖는다. 또한, 테스트베드에서 사용된 링크들은 100 Mbps 패스트 이더넷 네트워크 카드를 사용하여 연결된다.

테스트베드 구성에서 두 개의 멀티미디어 연결과 하나의 크로스 트래픽 연결이 이루어진다. 두 개의 멀티미디어 연결을 위하여 S1, S4 클라이언트는 각각 D1, D4 스트리밍 서버와 멀티미디어 연결을 설정하게 되며 이들 연결은 C0에서 C1까지 공유한다. S1과 S4에 연결되는 스트리밍 서비스의 QoS를 차등화하기 위하여 S1과D1간의 연결은 프리미엄 서비스 등급으로 설정하고 S4와 D4간의 연결은 브론즈 서비스 등급으로 설정한다. S1과 S4가 스트리밍 서비스를 제공받은 공유링크인 C0과 C1간에 트래픽 혼잡을 발생시키기 위해 MGEN 툴[18]을 사용하여 CBR 트래픽을 발생시킨다. 이렇게 함으로써, PMQoS DS에 의해 설정된 서비스 등급과 QoS 결과가 더욱 명확하게 증명될 수 있다.

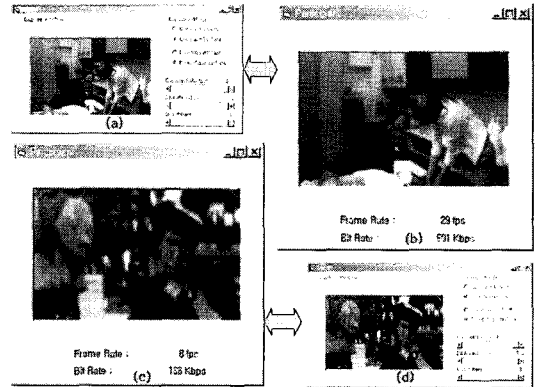


그림 7. H.263 스트리밍 시스템의 결과 화면
Fig 7. Snapshots of H.263 streaming system

그림 7은 두 개의 H.263 스트리밍 서버와 클라이언트의 결과 화면을 보여준다. 그림 (a)와 (b)는 D1과 S1에서의 실행화면이고 그림 (c)와 (d)는 D4와 S4에서의 실행 화면이다. 그림 7에서 볼 수 있듯이 프리미엄 서비스를 받는 D1과 S1간의 연결의 경우 대역폭은 501Kbps의 비트율과 29fps의 프레임율의 비디오 스트림을 수신하는 반면, D4와 S4의 연결은 대역폭은 136 Kbps의 비트율과 8fps의 비디오 스트림을 갖는다. 이러한 실험의 결과의 PMQoS DS 시스템은 상위 레벨의 QoS 정책을 통해 서로 경쟁하는 공유 연결에 차등적인 QoS를 제공할 수 있음을 입증할 수 있다.

V. 결론

본 논문은 차등 서비스를 지원하는 차세대 IP 기반

망을 위한 정책 기반 QoS 관리 시스템인 PMQoS를 제안하고 구현하였다. 제안된 PMQoS 시스템은 IETF 정책 프레임워크를 확장시킴으로써 정책 관리와 QoS 모니터링 기능을 통합하였다. PMQoS 시스템의 구현은 EJB 프레임워크를 사용하여 구현함으로써 시스템의 개발비용과 개발시간을 줄일 수 있고, 시스템의 유지보수, 확장성, 분산 트랜잭션, 안정성 등의 이점을 얻을 수 있다. 또한 XML을 사용하여 상위 레벨 QoS 정책들을 정의함으로써 XML이 제공하는 많은 유용한 파서와 유효성 분석기들을 용이하게 이용할 수 있으며, XML의 정보 표현에 대한 유연성으로 인해 새로운 유형의 정보를 쉽게 추가하고 변경할 수 있다.

본 논문에서는 또한 구현된 PMQoS 시스템의 유효성을 입증하기 위하여 테스트베드로 구축한 리눅스 기반 차등 서비스 망에서 H.263 기반의 비디오 스트리밍 시스템을 사용하여 실험하였다. 이 실험의 결과 실제 PMQoS 시스템이 다양한 시스템들에게 차등화된 QoS를 제공할 수 있으며, 동적인 QoS 변경을 지원할 수 있음을 입증하였다. 따라서 PMQoS 시스템은 ISP들에 의해 사용되는 망 관리 시스템으로 통합됨으로써 고객들의 다양한 QoS 요구사항을 동적으로 수용하고, 동적으로 수정하고, 동적으로 반영할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", IETF RFC 1633, June 1994.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.
- [3] D. Goderis, S. V. Dosch, Y. T'Joens, et al, "A Service-Centric IP Quality of Service Architecture for Next Generation Networks", Proceedings of 2002 IEEE/IFIP Network Operations & Management Symposium, 2002.
- [4] Y. Kanada, B. J. O'Keefe, "DiffServ Policies and Their Combinations in a Policy Server", Proceedings of the 5th Asia-Pacific Network Operations and Management Symposium, Sydney, Australia, September 2001.
- [5] D. Verma, M. Beigi, R. Jennings, "Policy based SLA Management in Enterprise Networks", Lecture Notes in Computer Science (LNCS) 1995, January 2001.
- [6] J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", IETF RFC 2748, January 2000.
- [7] R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy-based Admission Control", IETF RFC 2753, January 2000.
- [8] Sun Microsystems, Inc., "Telecom Network Management With Enterprise JavaBeans™ Technology", Technical White Paper, May 2001.
- [9] Differentiated Services on Linux, <http://diffserv.sourceforge.net>.
- [10] UCD-SNMP Package, University of California at Davis, <http://www.net-snmp.org/>.
- [11] Linux-Netlink, <http://cam-lough.lemonology.com/linux-netlink.html>
- [12] F. Marinescu, "EJB Design Patterns - Advanced Patterns, Processes, and Idioms", John Wiley & Sons, Inc., 2002.
- [13] Jakarta Tomcat 4.0.1, <http://jakarta.apache.org/tomcat/>.
- [14] JBoss Application Server, <http://www.jboss.org/>.
- [15] AdventNet SNMP API, <http://www.adventnet.com/products/>.
- [16] Oracle 8i Enterprise Edition, <http://otn.oracle.com/software/content.html>.
- [17] H. Cha, B. Ahn, K. Cho, "A QoS-providing multicast network management system", Computer Communications 24, 2001.
- [18] The Multi-Generator Toolset, <http://mgen.pf.itd.nrl.navy.mil/>.

차 시 호 (Si ho Cha)

정회원



1995년 순천대학교 전자계산
학과 졸업(이학사)
1997년 광운대학교 대학원 전
자계산학과 졸업(이학석사)
1997년~2000년 대우통신 종합
연구소 선임연구원
2004년 광운대학교 대학원 컴
퓨터과학과(공학박사)

2004년~2005년 (주)웨어플러스 책임연구원

2005년~현재 세종대학교 컴퓨터공학과 초빙교수

2005년~현재 (사)산업정보보호학회 이사

<관심분야> 네트워크 관리, 무선 Ad-Hoc 네트워크,
유비쿼터스 센서 네트워크

안 병 호 (Byung ho Ahn)

정회원



1988년~1994년 한국전자통신
연구소 정보통신 표준연구센
터 선임연구원

1999년 광운대학교 컴퓨터과
학과 대학원 이학박사

1994년~현재 충청대학 컴퓨터
학부, 의료산업학부 교수

2002년~현재 ISO/IEC JTC1/SC6 산자부 표준
화위원

2002년~현재 한국정보처리학회 IT인증본부 이사

2004년 미국 오클라호마대학교 컴퓨터과학과 교환교수

<관심분야> 네트워크 관리, 통신 프로토콜 표준화,
QoS, Mobile Internet, 멀티캐스트 통신, ASN.1 등

이 종 언 (Jong eon Lee)

정회원



2001년 광운대학교 전자계산
학과 졸업(이학사)

2003년 광운대학교 대학원 컴
퓨터과학과 졸업(공학석사)

2003년~현재 광운대학교 대학
원 컴퓨터과학과 박사수료

<관심분야> 네트워크 관리, 애
드후 네트워크, 차세대 네트워크, 소프트웨어 컴
포넌트 기술

조 국 현 (Kuk hyun Cho)

정회원



1977년 한양대학교 전자공학
과 졸업(공학사)

1981년 일본 동북대학교 대학
원 졸업(공학석사)

1984년 일본 동북대학교 대학
원 졸업(공학박사)

1984년~현재 광운대학교 전자
정보대학 컴퓨터공학부 교수

개방형컴퓨터통신연구회(OSIA) 회장 역임

<관심분야> 네트워크 관리, 분산처리, 정보통신 분
야의 표준화 등