

# 중앙 집중형 서버 기반 망에서의 연결 관리 방식 비교 및 성능 분석

권태현<sup>†</sup> · 임영은<sup>††</sup> · 최인상<sup>†††</sup> · 김춘희<sup>††††</sup> · 차영욱<sup>†††††</sup>

## 요약

차세대 네트워크의 성공적인 추진을 위하여 서비스 품질과 트래픽 엔지니어링을 보장할 수 있는 중앙 집중형 통합망 제어 및 관리 기술이 요구된다. 본 논문에서는 전달 평면과 제어 평면이 분리되어 있는 중앙 집중형 서버 기반의 망에서 사용될 수 있는 연결 관리 방식인 SNMP, 웹 서비스 그리고 네트워크 개방형 인터페이스 기반의 연결 관리 방식을 기술하였다. 또한, 이들 각 연결 관리 방식에 대한 플랫폼을 구축하고 각 연결 관리 방식에 대한 연결 설정 지연을 측정하여 성능을 비교 분석하였다.

키워드: 연결 관리, SNMP, XML, GSMP, 웹 서비스

## Comparison and Performance Analysis of Connection Management Methods in the Centralized Server Based Network

TaeHyun Kwon<sup>†</sup> · YoungEun Yim<sup>††</sup> · InSang Choi<sup>†††</sup> · ChoonHee Kim<sup>††††</sup> · YoungWook Cha<sup>†††††</sup>

## ABSTRACT

For the successful promotion about the NGN (Next Generation Network), centralized network control and management technology is required to guarantee QoS and Traffic Engineering. This paper describes SNMP, Web Services and Network Open Interface based connection management scheme which can be used in centralized server based network in which the control plane is separated from transport plane. We also constructed platforms about each of these connection management schemes and performed comparing and analyzing the performance of connection setup delays.

Key Words : Connection Management, SNMP, XML, GSMP, Web Services

### 1. 서론

차세대 네트워크의 성공적인 추진을 위하여 서비스 품질과 트래픽 엔지니어링을 보장할 수 있는 중앙 집중형 통합망 제어 및 관리 기술이 요구된다[1, 2]. ITU-T의 NGN (Next Generation Network)에서 보듯이 차세대 망은 전달 평면과 제어 평면 사이에 네트워크 개방형 인터페이스가 도입되며, 이들 두 평면이 독립적으로 진화될 것이다[3-5]. 제어 평면과 전달 평면의 기능이 분리되어 있는 중앙 집중형 망에서 제어기는 스위치들을 중앙 집중화된 방식으로 제어

및 관리하기 위한 서버로 망의 구성 정보, 자원 예약, 경로 및 연결 설정과 같은 제어 평면의 기능을 수행하게 되며, 스위치는 인접 노드에 대한 정보 수집 및 서비스 품질과 트래픽 엔지니어링을 제공하는 전달 평면의 기능을 수행하게 된다.

중앙 집중형 서버 기반의 망에서 연결 관리 서버인 제어기와 스위치 사이의 연결 관리 방식으로는 전통적인 SNMP(Simple Network Management Protocol)[6] 기반의 관리 방식과 SNMP의 대체 방안으로 최근에 대두되고 XML(eXtensible Markup Language) 기반의 웹 서비스 관리 방식[7] 그리고 GSMP(General Switch Management Protocol)[8]를 사용하는 네트워크 개방형 인터페이스 기반의 관리 방식이 가능하다. XML 기반의 웹 서비스 관리는 IETF의 Netconf(Network Configuration) 워킹 그룹에서 망 관리 매니저와 에이전트 사이에 원격 오퍼레이션을 위한 오퍼레이션들과 전송 프로토콜들 그리고 이벤트 Notification 들에 대하여 표준화가 진행 중에 있다[9]. GSMP는 전달 평

※ 본 논문은 한국과학재단 우수연구센터(OIRC) 사업과 ETRI 정보통신 연구개발사업 위탁과제의 연구결과임  
† 준회원: 안동대학교 컴퓨터공학과 공학박사  
†† 준회원: 안동대학교 컴퓨터공학과 석사과정  
††† 정회원: 과학기술연합대학원대학교(UST) 광대역네트워크공학 전공 박사과정  
†††† 정회원: 대구사이버대학교 컴퓨터정보학과 조교수  
††††† 정회원: 안동대학교 컴퓨터공학과 부교수(교신일자)  
논문접수: 2006년 3월 31일, 심사완료: 2006년 8월 11일

면과 제어 평면의 분리를 실현하며, 데이터 전달 기술과 제어 기술 간의 독자적인 개발 자유도를 최대화할 수 있도록 IETF에서 표준화된 네트워크 개방형 인터페이스 기술이다.

네덜란드의 Twente 대학에서는 SNMP 기반 망 관리와 웹 서비스 기반 망 관리에 대한 대역폭 사용 및 왕복 지연 측면 등을 비교한 연구[10]를 수행한 바가 있다. 대역폭 사용 측면에서는 하나 또는 소수의 관리 객체를 가져오는 경우에는 SNMP가 웹 서비스 기반 망 관리보다 대역폭을 효율적으로 사용하나, 인터페이스 구성 테이블과 같은 많은 관리 객체들을 가져오는 경우에는 압축 기법을 적용한 웹 서비스가 SNMP보다 대역폭을 효율적으로 사용함을 보이고 있다. 일반적으로 웹 서비스의 XML 코딩이 SNMP의 BER(Basic Encoding Rules) 코딩보다 CPU의 시간이 많이 요구되나, 많은 관리 객체들을 에이전트에서 가져오는 경우에 코딩과 압축을 위한 시간은 관리 객체를 가져오는 시간에 비해 무시할 수 있다고 제시하고 있다. 이 연구에서는 MIB-II에 정의된 인터페이스 구성 테이블을 가지고 오는 경우에 대하여 성능을 비교하고 있다.

전달 평면과 제어 평면이 분리되어 있는 중앙 집중형 서버 기반 망에서 연결 관리 플랫폼의 성능 비교 및 분석은 효과적인 연결 관리를 위하여 필수적이다. 특히, 연결 설정 지연은 서비스 품질을 보장하기 위한 중요한 성능 파라미터이다. 본 논문에서는 중앙 집중형 서버 기반 망에서 연결 관리 방식에 대한 연결 설정 지연의 측정 및 성능을 비교 분석하기 위하여 표준 MIB에 대응되는 사설 MIB과 XML 기반 웹 서비스 연결 관리를 위한 정보 모델링과 절차를 기술한다. 또한, SNMP 기반 연결 관리 방식과 XML 기반의 웹 서비스 연결 관리 방식 그리고 GSMP를 사용하는 네트워크 개방형 인터페이스 기반의 연결 관리 방식에 대한 연결 관리 플랫폼을 설계 및 구현한다. 구현한 각각의 연결 관리 방식에 대하여 실험실 환경에서 연결 설정 지연을 측정하여 전달 평면과 제어 평면이 분리되어 있는 중앙 집중형 서버 기반 망에서의 연결 관리 방식들에 대한 연결 설정 지연을 비교 및 분석한다.

본 논문의 2장에서는 SNMP 기반 연결 관리 및 네트워크 개방형 인터페이스 기반 연결 관리와 XML 기반 웹 서비스 관리 기술의 연구 동향에 대하여 기술한다. 3장에서는 표준 MIB에 대응되는 사설 MIB과 XML 기반 웹 서비스 방식의 연결 관리를 위한 정보 모델링 및 절차를 정의한 후, 각 연결 관리 방식에 대한 연결 관리 플랫폼의 설계 및 구현에 대하여 기술한다. 4장에서는 실험실 환경에서 본 논문에서 제시된 각 연결 관리 방식에 대한 연결 설정 지연을 측정하여 비교 분석하며, 마지막으로 5장에서는 본 논문의 결론 및 향후 과제에 대하여 기술한다.

## 2. 연결 관리 방식과 XML 기반 웹 서비스 관리의 기술 동향

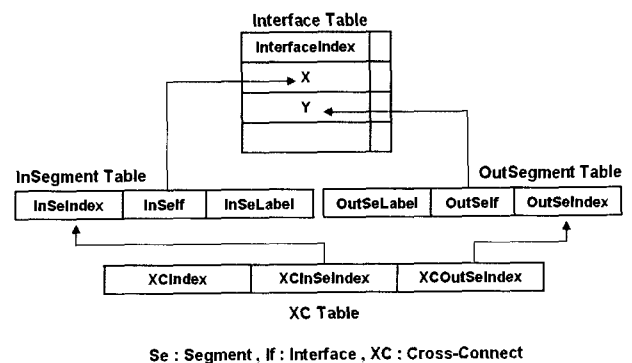
본 장에서는 SNMP 및 네트워크 개방형 인터페이스 기반

연결 관리 방식과 XML 기반의 웹 서비스 관리 기술 동향에 대하여 기술한다.

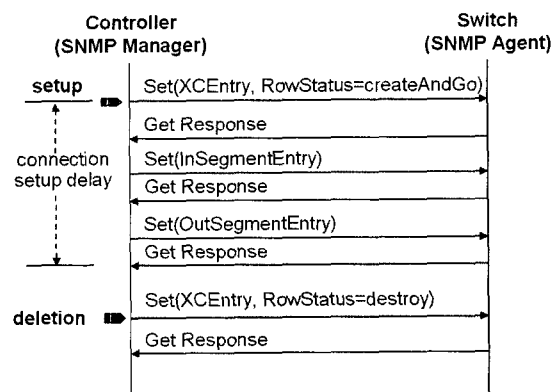
### 2.1. SNMP 기반 연결 관리

MPLS(Multiprotocol Label Switching)/GMPLS(Generalized MPLS)의 망 관리를 위하여 IETF에서 MPLS-FTN-STD-MIB, MPLS-LSR-STD-MIB, MPLS-LDP-STD-MIB, MPLS-TE-STD-MIB 그리고 GMPLS-LSR-STD-MIB, GMPLS-TE-STD-MIB 등을 정의하고 있다[11,12]. MPLS-FTN-STD-MIB을 제외한 MIB들은 모두 LER(Label Edge Router) 및 LSR(Label Switching Router)에 적용될 수 있으나, MPLS-FTN-STD-MIB은 입구 LER에만 적용된다. (그림 1)은 망 관리를 통하여 제어기가 LSR에 연결을 설정하고 해제하기 위한 표준 MPLS-LSR-STD-MIB[13]에 정의된 연결 관리 테이블 엔트리들의 상호 연관 관계를 나타낸다. 입력 및 출력 세그먼트 테이블 엔트리들은 인터페이스 테이블 엔트리와 연관되며, 크로스-커넥트 테이블 엔트리는 입력 및 출력 세그먼트 테이블 엔트리와 연관 관계를 구성한다[14].

(그림 2)는 표준 MPLS-LSR-STD-MIB을 사용하여 망 관리 매니저가 연결 관리 기능을 통하여 연결을 설정하고 해제하는 절차를 나타낸다[13].



(그림 1) MPLS-LSR-STD-MIB에 정의된 연결 관리 테이블 엔트리들의 연관 관계



(그림 2) MPLS-LSR-STD-MIB을 사용한 연결 설정 및 해제 절차

제어기의 망 관리 매니저는 크로스-커넥트 테이블 엔트리에 해당 값을 설정한 후, 스위치에 탑재된 에이전트에게 크로스-커넥트 테이블 엔트리의 설정을 요구하게 된다. 설정 요구를 받은 에이전트는 해당 명령에 대한 응답을 매니저에게 보낸다. 응답을 받은 매니저는 설정한 크로스-커넥트 테이블 엔트리의 입력 및 출력 세그먼트 인덱스와 연관되는 입력 및 출력 세그먼트 테이블 엔트리의 설정을 에이전트에게 요구한다. 요구를 받은 에이전트가 응답을 매니저에게 보내면 연결 설정이 완료된다. 연결의 해제 과정은 크로스-커넥트 테이블 엔트리의 상태 칼럼에 'destroy' 값을 설정하여 에이전트에게 연결 해제를 요구하게 된다.

2.2 네트워크 개방형 인터페이스 기반 연결 관리

네트워크 개방형 인터페이스 기반의 연결 관리 방식은 GSMP를 사용한다. GSMP는 전달 평면과 제어 평면의 분리를 실현하며, 데이터 전달 기술과 제어 기술 간의 독자적인 개발 자유도를 최대화할 수 있도록 IETF에서 표준화된 네트워크 개방형 인터페이스 기술로 제어기와 스위치 사이에서 연결, 구성, 상태 및 통계, 이벤트 관리 및 동기화 기능 등을 제공하는 프로토콜이다.

이벤트 관리 기능을 제외한 GSMP 프로토콜의 관리 기능 메시지들은 제어기에 의해 생성되며, 스위치는 제어기가 보낸 메시지에 대한 응답을 수행한다. 이벤트 관리를 위한 이벤트 메시지들은 스위치에 의해 생성되며, 제어기는 이벤트 메시지에 대하여 응답 메시지를 보내지 않는다. 프로토콜 버전의 합의, 상태 동기화 등의 정보를 교환하는 GSMP 프로토콜의 Adjacency 기능은 제어기나 스위치 어느 곳에서든 먼저 활성화될 수 있다[8].

GSMP는 제어기에 마스터 기능이 탑재되며 스위치에는 슬레이브 기능이 탑재되는 비대칭적인 구조를 가진다. 제어기에 탑재된 GSMP 마스터와 스위치에 탑재된 GSMP 슬레이브 사이의 연결 설정 및 해제를 위한 절차는 (그림 3)과 같다.

연결 설정을 위하여 Add Branch 메시지를 사용하며, 해제를 위해서는 Delete Branches 또는 Delete Tree 메시지를 사용한다. Delete Branches 메시지는 Add Branch 메시지로 설정한 연결 중에서 하나 또는 그 이상의 브랜치를 해제하기 위한 메시지이고, Delete Tree는 설정된 연결의 모든 브

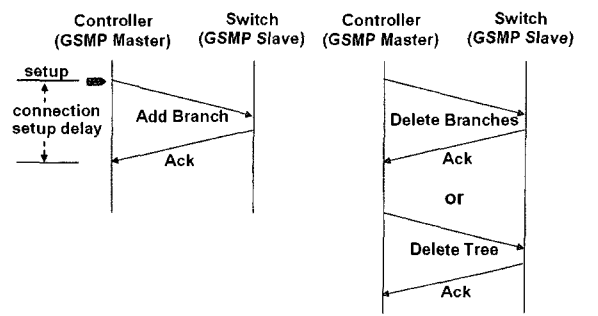
랜치, 즉 연결을 완전히 해제하기 위하여 사용하는 메시지이다.

2.3 XML 기반의 웹 서비스 관리 기술

최근에는 SNMP의 효율성과 확장성에 대한 문제를 해결하기 위해 XML를 사용하는 웹 서비스 기반 네트워크 관리가 대두 되고 있다. 웹 서비스 기반 네트워크 관리와 관련해서는 IETF의 Netconf 워킹 그룹에서 XML 기반 네트워크 관리에 대한 표준화를 진행 중에 있으며, 학계 및 업계에서도 많은 연구들이 진행 중에 있다[14-20].

Netconf 워킹 그룹은 구성 관리를 위한 RPC(Remote Procedure Call) 기반 오퍼레이션을 정의하고 있다. 또한, 매니저와 에이전트 간에 전송되는 관리 프로토콜과 전송 프로토콜과의 매핑에 대해서도 표준화를 진행하고 있다. Netconf 프로토콜은 매니저가 네트워크 장비들의 구성 정보를 쉽게 관리하고, 에이전트간의 상호 운용성을 보장하기 위해서 매니저와 에이전트 사이에 전송되는 메시지를 구조화된 XML 형식으로 정의하고 있다. 구성 관리의 요청을 위하여 매니저는 오퍼레이션을 포함하는 XML 메시지를 RPC를 이용하여 에이전트에게 전달한다. 즉, 요청 메시지에 수행하고자 하는 오퍼레이션을 XML 태그에 명시하여 보내면, 에이전트는 자신의 XML 파서를 이용하여 오퍼레이션을 추출하고 관리 작업을 수행하여 수행 결과를 매니저에게 응답하게 된다. Netconf 프로토콜은 다양한 환경의 네트워크 장비에 대한 관리 요구 사항을 충족시키기 위해 전송되는 메시지를 <표 1>과 같이 4가지 계층으로 나누어 정의하였다. 'Content' 계층은 관리되어야 할 장비 및 장비 업체에 의존적인 관계로 아직 표준화가 이루어지지 않은 상태이다. 현재 Netconf 워킹 그룹에서는 'Content' 계층 아래 부분에 대한 표준화 작업을 마무리하고 다음 단계로 'Content' 계층에 대한 표준화 작업의 가능성에 대하여 검토 중이다.

업계의 대표적인 XML 기반 구성 관리 적용 사례들은 시스코의 Configuration Registrar[16], 휴니퍼의 JUNOScript [17], Bell Laboratories의 TEAMS(Traffic Engineering and Monitoring Server)[18] 등이 있으며, 학계의 대표적인 XML 기반 망 관리에 대한 연구 결과로는 독일 Braunschweig의 Technical 대학의 XML-to-SNMP 게이트웨이에 대한 연구 [19], 네덜란드 Twente 대학의 SNMP 기반 망 관리와 웹 서비스 기반 망 관리 비교 연구[10] 그리고 포항 공대의



(그림 3) GSMP의 연결 설정 및 해제 절차

<표 1> Netconf 프로토콜의 계층적 구조

Layer	Example
Content	Configuration data (IP address, Admin ID/Password, etc.)
Operations	<get-config>, <edit-config>, etc.
RPC	<rpc>, <rpc-reply>, etc.
Transport	SSH, BEEP, SOAP over HTTP, etc.

XCMS(XML-based Configuration Management System) [20] 등이 있다.

### 3. 연결 관리 플랫폼의 설계 및 구현

본 장에서는 중앙 집중형 서버 기반 망에서의 연결 관리 방식에 대한 연결 설정 지연의 측정 및 성능을 비교 분석하기 위하여 표준 MPLS-LSR-STD-MIB[13]에 대응되는 사설 SIMPLE-MPLS-MIB을 정의하고, 이를 기반으로 XML 기반 웹 서비스 방식의 연결 관리를 위한 정보 모델링 및 절차를 정의한다. 또한, 각 연결 관리 방식에 대한 연결 관리 플랫폼의 설계 및 구현에 대하여 기술한다.

#### 3.1 SIMPLE-MPLS-MIB 정의 및 연결 관리 절차

SNMP 기반 연결 관리 방식은 표준 MIB 기반 방식과 사설 MIB 기반 방식이 있다. 표준 MPLS-LSR-STD-MIB 기반의 연결 관리에서는 입력 및 출력 세그먼트 테이블, 크로스-커넥트 테이블 엔트리를 각각 별도의 SNMP 요청 및 응답 메시지를 이용하여 설정하므로 연결 설정 지연이 발생할 수 있다. 본 논문에서는 이러한 연결 설정 지연을 최소화하기 위하여 하나의 SNMP 요청 및 응답 메시지로 연결 설정 및 해제를 할 수 있도록 표준 MPLS-LSR-STD-MIB을 단순화하여 사설 SIMPLE-MPLS-MIB을 정의하였다. (그림 4)는 연결 설정 및 해제를 위하여 사설 SIMPLE-MPLS-MIB에 정의된 연결 관리 테이블 및 엔트리에 포함되는 관리 정보들을 보여준다.

(그림 5)는 사설 SIMPLE-MPLS-MIB을 사용하여 망 관리 매니저가 스위치에게 연결을 설정하고 해제하는 절차를 나타낸다. 사설 SIMPLE-MPLS-MIB을 사용한 연결 설정

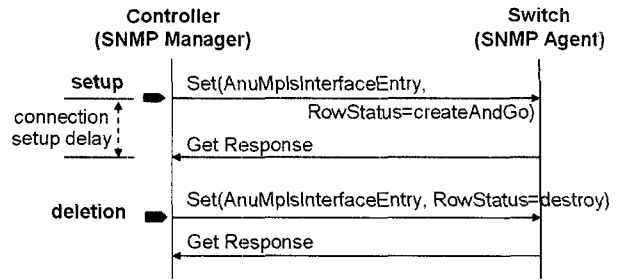
```

anuMPLSInterfaceTable OBJECT-TYPE
SYNTAX SEQUENCE OF AnuMPLSInterfaceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Transit Node Configuration Info"
 ::= { anSimpleMplsMIB 1 }

anuMPLSInterfaceEntry OBJECT-TYPE
SYNTAX AnuMPLSInterfaceEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Row Description"
INDEX { anuIfName, anuInLabel }
 ::= { anuMplsInterfaceTable 1 }

AnuMPLSInterfaceEntry ::= SEQUENCE (
    anuIfName DisplayString,
    anuInLabel Integer32,
    anuNextHopIPAddr DisplayString,
    anuLabelAction Integer32,
    anuSwapOutLabel Integer32,
    anuMPLSInterfaceRowStatus RowStatus
)
    
```

(그림 4) SIMPLE-MPLS-MIB의 연결 관리 테이블 및 엔트리



(그림 5) SIMPLE-MPLS-MIB을 사용한 연결 설정 및 해제 절차

및 해제 절차는 (그림 2)의 표준 MPLS-LSR-STD-MIB을 사용한 연결 설정 및 해제 절차보다 단순하다.

연결 설정을 위하여 표준 MPLS-LSR-STD-MIB을 사용할 경우, 망 관리 매니저는 크로스-커넥트, 입력 및 출력 세그먼트 테이블 엔트리의 설정을 에이전트에게 요구하여야 한다. 하지만, 사설 SIMPLE-MPLS-MIB을 사용할 경우는 AnuMplsInterfaceEntry의 설정만을 에이전트에게 요구하면 된다. 또한, 연결 해제를 위하여 표준 MPLS-LSR-STD-MIB을 사용할 경우, 망 관리 매니저가 크로스-커넥트 테이블 엔트리의 상태 칼럼에 'destroy' 값을 설정하여 에이전트에게 요구하게 되면, 에이전트는 내부적으로 크로스-커넥트 테이블 엔트리와 연관된 입력 및 출력 세그먼트 테이블 엔트리를 삭제하게 된다. 반면, 사설 SIMPLE-MPLS-MIB을 사용하여 연결 해제를 할 경우에는 AnuMplsInterfaceEntry의 상태 칼럼에 'destroy' 값을 설정하여 에이전트에게 요구하게 되면, 요구를 수신한 에이전트는 해제를 요청한 해당 AnuMplsInterfaceEntry를 삭제하게 된다.

#### 3.2 XML 기반 웹 서비스 방식의 연결 관리를 위한 정보 모델링 및 절차

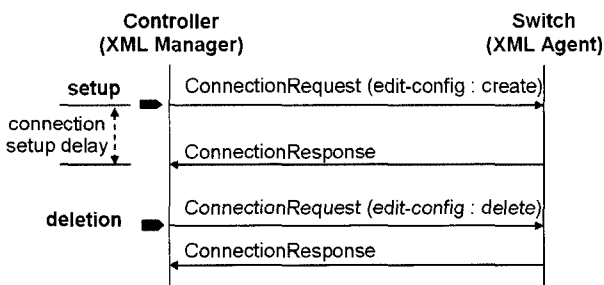
IETF의 Netconf 워킹 그룹에서는 실제 관리 정보들에 대한 모델링은 정의하지 않고 있다. XML를 사용하는 웹 서비스 기반의 연결 관리를 위해서는 정보 모델링과 절차가 정의되어야 한다. 정보 모델링은 표준 MIB을 기반으로 정의할 수도 있고, 사설 MIB과 같이 망의 구성 및 용도에 따라 필요한 관리 정보 및 요구 사항을 기반으로 정의할 수 있다. 본 논문에서는 표준 MPLS-LSR-STD-MIB을 단순화하여 정의한 사설 SIMPLE-MPLS-MIB을 기반으로 (그림 6)과 같이 XML 스키마를 이용하여 정보를 모델링 하였다.

anuMplsInterfaceTable은 데이터 타입으로 anuMplsInterfaceTableType을 사용하며, 자식 엘리먼트로 anuMplsInterfaceTableEntry를 가진다. anuMplsInterfaceTableEntry 엘리먼트는 anuMplsInterfaceTableEntryType을 데이터 타입으로 사용하며, 5개의 자식 엘리먼트로 anuIfName, anuInLabelName, anuNextHopIpaddr, anuLabelAction 그리고 anuSwapOutLabel를 가진다. anuLabelAction 엘리먼트는 데이터 타입으로 anuLabelActionType을 사용하며, 1 또는 2의 값(pop 또는 swap)을 가질 수 있는 엘리먼트이다.

(그림 7)은 웹 서비스 연결 관리 방식에서의 연결 설정

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="anumplsInterfaceTable"
    type="anumplsInterfaceTableType">
    <xs:annotation>
      <xs:documentation> Connent describing your root element
    </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="anumplsInterfaceTableType">
    <xs:sequence>
      <xs:element name="anumplsInterfaceTableEntry"
        type="anumplsInterfaceTableEntryType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="anumplsInterfaceTableEntryType">
    <xs:sequence>
      <xs:element name="anuIfName" type="xs:string"/>
      <xs:element name="anuInLabelName" type="xs:integer"/>
      <xs:element name="anuNextHopIpaddr" type="xs:string"/>
      <xs:element name="anuLabelAction" type="anuLabelActionType"/>
      <xs:element name="anuSwapOutLabel" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="anuLabelActionType">
    <xs:restriction base="xs:integer">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

(그림 6) XML 스키마를 이용한 웹 서비스 연결 관리에 대한 정보 모델링



(그림 7) XML 기반의 웹 서비스 연결 관리 절차

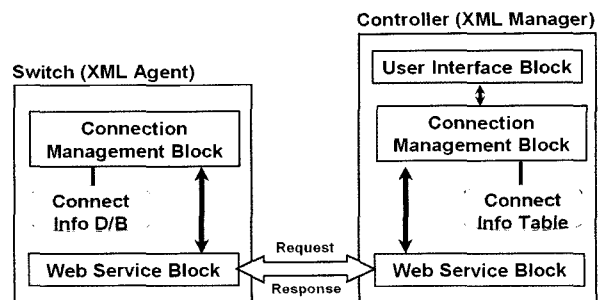
및 해제를 위한 절차를 나타낸다. 연결 설정 및 해제를 위하여 ConnectionRequest 메시지와 ConnectionResponse 메시지를 정의하여 사용하였으며, 연결 설정 및 해제를 위한 오퍼레이션은 Netconf 워킹 그룹의 관리 프로토콜의 규격을 채택하였다. 즉, ConnectionRequest 메시지와 ConnectionResponse 메시지는 Netconf 프로토콜의 'Operation' 계층에 정의되어 있는 <edit-config> 오퍼레이션과 'create' 및 'delete' 속성을 이용하여 연결 설정 및 해제 기능을 수행하도록 하였다. 본 논문에서는 추가적인 XML 파싱 작업을 없애기 위하여 포함 공대에서 제안된 바가 있는 <rpc> 및 <rpc-reply> 엘리먼트를 생략하고, <edit-config>를 오퍼레

이션으로 사용하는 방식[20]을 채택하였다.

### 3.3 XML 기반 연결 관리 플랫폼의 설계 및 구현

(그림 8)은 XML 기반의 웹 서비스 연결 관리 플랫폼에 대한 소프트웨어의 구현 구조를 나타낸다. 제어기와 스위치에 SOAP(Simple Object Access Protocol)[21] 엔진을 포함하고 있는 웹 서비스 블록과 연결 관리를 담당하는 연결 관리 블록이 구현되어지며, 사용자 인터페이스를 제공해 주는 UI(User Interface) 블록은 제어기에 구현되어진다.

제어기의 웹 서비스 블록은 SOAP 클라이언트 기능을 하며, 스위치의 웹 서비스 블록은 SOAP 서버 기능을 한다.



(그림 8) XML 기반 웹 서비스 연결 관리 플랫폼의 구현 구조

제어기는 웹 서비스 블록에서 SOAP 엔진으로 Axis[22]를 사용하는 Axis 기반의 제어기와 gSOAP[23]를 사용하는 gSOAP 기반의 제어기로 구분하여 구현하였으며, 스위치는 gSOAP를 사용하여 구현하였다. 자바로 구현된 Axis는 윈도우 또는 리눅스 환경에서 동작이 가능하며 아파치 프로젝트에서 제공하는 Tomcat 서버와 연동이 수월하여 웹 브라우저를 이용하는 웹 기반의 애플리케이션을 쉽게 개발할 수 있는 이점이 있다. 반면 C/C++ 기반의 웹 서비스 개발 도구인 gSOAP은 Axis에 비해 웹 서비스 개발과 배치가 쉽고 간단하며, 적은 양의 메모리를 차지하기 때문에 경량의 웹 서비스를 구축하는데 이점이 있다.

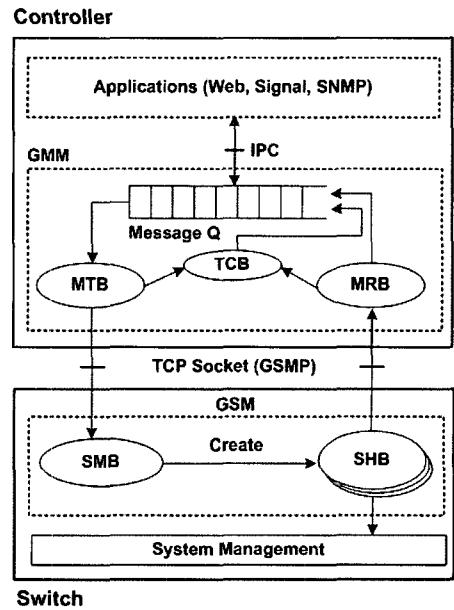
3.4 SNMP 및 GSMP 기반 연결 관리 플랫폼의 설계 및 구현

SNMP 기반 연결 관리 플랫폼의 매니저와 에이전트를 위하여 AdventNet[24]의 Management Builder와 Agent Toolkit을 이용하였으며 매니저는 자바로, 에이전트는 리눅스 환경에서 C로 구현하였다[25]. SNMP 기반 연결 관리 플랫폼은 표준 MIB인 MPLS-LSR-STD-MIB을 사용하는 플랫폼과 사설 MIB인 SIMPLE-MPLS-MIB을 사용하는 플랫폼으로 구분하여 구현하였다.

(그림 9)는 GSMP 기반 연결 관리 방식을 위한 소프트웨어의 구현 구조를 나타낸다. GSMP 마스터와 GSMP 슬레이브는 리눅스 환경에서 C로 구현하였으며, 제어기에 탑재되는 어플리케이션들과 GMM(GSMP Master Module)은 IPC 통신을 위하여 메시지-큐를 사용한다.

제어기의 GMM과 스위치의 GSM(GSMP Slave Module)은 GSMP 마스터와 GSMP 슬레이브 기능을 수행하는 모듈이다. 제어기의 GMM과 스위치의 GSM은 GSMP 규격에 정의된 TCP 소켓을 사용하여 통신한다. GMM은 MTB(Master-Tx Block), MRB(Master-Rx Block) 그리고 TCB(Time Check Block)으로 구성되며, 각각 Linux의 스레드로 동작한다. MTB는 GSMP 메시지를 생성하여 스위치로 전달하는 역할과 제어기와 스위치 사이에 동기를 설정하고 유지하는 기능을 수행한다. 스위치로부터 응답 받은 메시지를 처리하는 기능은 MRB에서 수행한다. TCB는 제어기가 스위치 사

이에 메시지를 교환하는 경우에 발생할 수 있는 메시지의 손실을 확인하기 위한 프로토콜의 타이머 기능을 지원한다. GSM은 제어기로부터 SMB(Slave Main Block)을 통해 Add Branch 메시지를 수신하면 SHB(Slave Handler Block)을 생성한다. SHB에서는 스위치의 시스템 관리 모듈과의 상호 작용을 통하여 연결 설정 및 해제와 관련된 일을 수행하며, 처리된 결과를 제어기에 응답한다.

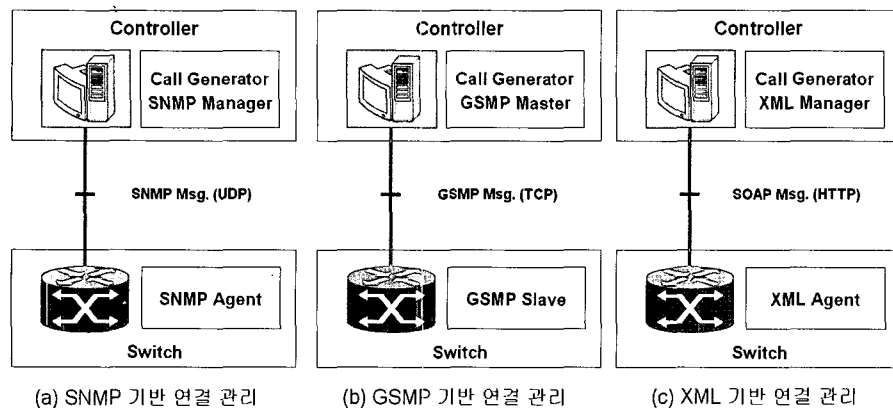


GMM : GSMP Master Module , GSM : GSMP Slave Module  
 MTB : Master Tx Block , TCB : Time Check Block , MRB : Master Rx Block  
 SMB : Slave Main Block , SHB : Slave Handler Block

(그림 9) 네트워크 개방형 인터페이스 기반 연결 관리 플랫폼의 구현 구조

4. 연결 관리 플랫폼의 연결 설정 지원

본 장에서는 본 논문에서 제시된 각 연결 관리 방식에 대한 실험 환경 및 연결 설정 지원을 정의한 후, 각 연결 관리 플랫폼에서 연결 설정 지원을 측정하여 비교 분석한다.



(그림 10) 연결 관리 방식에 대한 실험 환경

4.1 실험 환경 및 연결 설정 지연의 측정 방식

(그림 10)은 연결 설정 지연을 측정하기 위한 각 연결 관리 방식에 대한 실험 환경을 나타낸다. 제어기는 3.2GHz Xeon CPU에 메모리 2GB를 장착한 서버급 PC를 사용하였고, 스위치는 1.7GHz 펜티엄-4 CPU에 메모리 256MB를 장착한 일반 PC를 사용하였다. 제어기와 스위치의 운영체제는 커널 2.4 기반의 리눅스를 동일하게 사용하였다.

제어기에는 포아송 분포로 호를 생성하는 호 생성기(Call Generator)와 각 연결 관리 방식별로 SNMP 매니저, GSMP 마스터 또는 XML 매니저가 탑재된다. 스위치에는 각 연결 관리 방식별로 SNMP 에이전트, GSMP 슬레이브 또는 XML 에이전트가 탑재되며 디바이스 드라이버를 통하여 연결을 설정하는 시간으로 0.5ms의 지수 분포를 갖는 지연을 두어 인위적인 프로세싱 작업을 하도록 하였다.

실험 환경 및 각 연결 관리 방식의 구현 구조에 따라서 연결 설정 지연의 측정 방식을 본 논문에서는 (그림 11)과 같이 7가지 방식으로 구분하였다.

SNMP 기반 연결 관리는 표준 MPLS-LSR-STD-MIB을 사용하는 방식과 사설 SIMPLE-MPLS-MIB을 사용하는 방식으로 구분하며, 각 방식은 다시 스레드 방식과 스레드 풀 방식으로 구분한다. 스레드 방식은 자바로 구현된 SNMP 매니저가 호 생성기로부터 연결 설정에 대한 요청을 수신하면 연결 설정을 담당하는 새로운 스레드 객체를 생성하여 시작시키는 방식이다. 스레드 풀 방식은 SNMP 매니저가 연결 설정을 담당하는 스레드 객체를 미리 생성시키고, 호 생성기로부터 연결 설정에 대한 요청을 수신하면 미리 생성한 스레드 객체를 시작시키는 방식이다.

SNMP 기반의 스레드 방식들에 대한 총 연결 설정 지연은 스레드 객체를 생성 및 시작하는데 걸리는 시간과 스레드에서 연결을 설정하는데 걸리는 순수 연결 설정 지연의 합으로 정의된다. 스레드 풀 방식들에 대한 총 연결 설정 지연은 스레드 방식에서 요구되는 스레드 생성 시간을 제외한 스레드 객체를 시작하는데 걸리는 시간과 스레드에서의 순수 연결 설정 지연의 합으로 정의된다. 표준 MPLS-LSR-STD-MIB 기반 연결 관리 방식에서의 순수 연결 설정 지연은 연결 설정을 담당하는 스레드 객체가 (그림 2)와 같이 크로스-키넥트 엔트리의 생성을 요구한 시점부터 출력 세그먼트 엔트리의 생성에 대한 응답을 받을 때까지로 규정

한다. 사설 SIMPLE-MPLS-MIB 기반 연결 관리 방식에서의 순수 연결 설정 지연은 연결 설정을 담당하는 스레드 객체가 (그림 5)와 같이 AnuMplsInterfaceTableEntry의 생성을 요구한 시점부터 응답을 받을 때까지로 규정한다.

XML 기반 연결 관리 방식은 XML 매니저의 구현 구조에 따라 Axis 기반 방식과 gSOAP 기반 방식으로 구분하며, XML 매니저가 호 생성기로부터 연결 설정에 대한 요청을 수신하면 연결 설정을 담당하는 새로운 스레드(gSOAP 기반인 경우) 또는 스레드 객체(Axis 기반인 경우)를 생성하여 시작시키는 스레드 방식으로 동작한다. 연결 설정 지연은 SNMP의 스레드 방식과 동일하게 규정하며, 순수 연결 설정 지연은 연결 설정을 담당하는 스레드 객체가 (그림 7)과 같이 ConnectionRequest 메시지를 보내는 시점부터 응답 메시지로 ConnectionResponse 메시지를 수신하는 시점까지로 규정한다.

GSMP 기반의 연결 관리 방식에 따른 연결 설정 지연은 호 생성기로부터 연결 설정을 요구 받은 GSMP 마스터가 (그림 3)과 같이 Add Branch 메시지를 보내는 시점부터 응답 메시지인 Ack 메시지를 수신하는 시점까지로 규정한다.

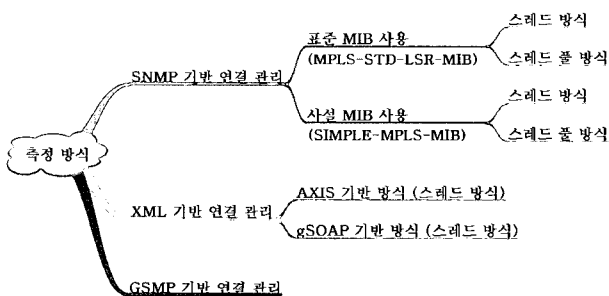
4.2 연결 설정 지연 측정 및 비교 분석

<표 2>는 포아송 분포를 갖는 초당 호의 평균 도착 개수가 20에서 1,000까지 증가하는 경우에 대하여 (그림 11)의 7가지 측정 방식들에 대한 연결 설정 지연을 측정된 결과이다.

4.2.1 SNMP 연결 관리 방식

표준 MPLS-LSR-STD-MIB 기반의 스레드 방식에서 연결 설정 지연은 20에서 60호까지 24.25에서 25.42ms 사이의 값이 측정되었고, 스레드 풀 방식에서의 연결 설정 지연은 20에서 1,000호까지 2.23에서 55.43ms 사이의 연결 설정 지연 값이 측정되었다. 스레드 방식에 대한 측정 결과를 60호까지 나타낸 이유는 SNMP 매니저가 연결 설정을 위한 스레드 객체를 생성하고 시작시키는 지연 시간 때문에 60호 이상부터는 의미가 없기 때문이다. 초당 20호일 경우에는 평균적으로 약 50ms마다 호가 생성되어진다. <표 2>의 20호일 경우에 측정된 결과에 따르면 순수 연결 설정 지연은 3.29ms, 스레드 객체의 생성 및 시작에 대한 지연은 20.95ms로 총 연결 설정 지연은 24.25ms로 측정되었다. 이는 호가 생성되는 시간보다 더 빨리 SNMP 매니저가 스레드 객체를 생성하여 시작시키는 의미로 해석된다. 그러나 60호의 경우는 약 16ms마다 호가 생성되는 반면에 스레드 객체의 생성 및 시작에 대한 지연은 21.63ms로 측정되었다. 이 결과는 호가 생성되는 시간보다 더 많은 시간이 소요된다는 의미로 해석된다. 초당 호의 수가 60부터는 생성되는 초당 호의 수와 관계없이 스레드 객체를 생성하고 시작시키는 지연 시간에 영향을 받게 되므로 60호 이상에서는 60호와 유사한 연결 설정 지연 값이 측정되어진다.

사설 SIMPLE-MPLS-MIB 기반의 스레드 방식에서 연결



(그림 11) 연결 설정 지연에 대한 측정 방식

〈표 2〉 연결 관리 방식에 따른 연결 설정 지연

단위 (ms)

초당 호수	표준 MIB (스레드)	사실 MIB (스레드)	표준 MIB (스레드 풀)	사실 MIB (스레드 풀)	GSMP	XML, Axis (스레드)	XML, gSOAP (스레드)
20	24.25 (3.29 + 20.95)	9.05 (1.97 + 7.08)	2.23 (2.17 + 0.06)	1.58 (1.52 + 0.06)	1.03	9.21 (5.68 + 3.53)	1.82 (1.75 + 0.07)
40	25.08 (3.63 + 21.45)	9.04 (2.04 + 7.00)	2.29 (2.22 + 0.07)	1.59 (1.52 + 0.07)	1.03	10.00 (6.07 + 3.93)	1.82 (1.75 + 0.07)
60	25.42 (3.80 + 21.63)	9.22 (2.20 + 7.03)	2.30 (2.23 + 0.07)	1.59 (1.52 + 0.07)	1.04	11.02 (6.82 + 4.21)	1.83 (1.75 + 0.08)
80		9.44 (2.37 + 7.07)	2.35 (2.27 + 0.08)	1.61 (1.52 + 0.08)	1.04	11.83 (7.28 + 4.55)	1.83 (1.74 + 0.08)
100		9.60 (2.46 + 7.13)	2.42 (2.34 + 0.08)	1.67 (1.54 + 0.13)	1.04	13.35 (8.57 + 4.77)	1.84 (1.76 + 0.08)
200		9.74 (2.46 + 7.28)	2.74 (2.61 + 0.12)	1.80 (1.65 + 0.15)	1.06	17.57 (11.45 + 6.12)	1.90 (1.81 + 0.09)
300			3.28 (3.11 + 0.17)	1.98 (1.79 + 0.19)	1.10	17.94 (11.83 + 6.11)	1.98 (1.90 + 0.09)
400			4.31 (4.11 + 0.20)	2.19 (1.99 + 0.20)	1.14		2.14 (2.04 + 0.09)
500			4.54 (4.30 + 0.24)	2.71 (2.47 + 0.25)	1.18		2.24 (2.15 + 0.09)
600			5.89 (5.61 + 0.28)	2.82 (2.56 + 0.26)	1.24		2.44 (2.35 + 0.09)
700			7.96 (7.61 + 0.35)	2.97 (2.67 + 0.30)	1.26		2.65 (2.55 + 0.10)
800			13.51 (13.02 + 0.49)	3.51 (3.20 + 0.31)	1.31		3.05 (2.96 + 0.10)
900			52.86 (51.50 + 1.36)	4.50 (4.18 + 0.32)	1.46		3.64 (3.54 + 0.10)
1000			55.43 (54.07 + 1.36)	5.27 (4.90 + 0.37)	1.49		4.69 (4.59 + 0.10)

\* 스레드 방식 : [총 연결 설정 지연] = (순수 연결 설정 지연 + 스레드 생성/시작 지연)  
 \* 스레드 풀 방식 : [총 연결 설정 지연] = (순수 연결 설정 지연 + 스레드 시작 지연)

설정 지연은 20에서 200호까지 9.05에서 9.74ms 사이의 값이 측정되었다. 표준 MIB 기반의 스레드 방식과 유사하게 300호부터는 생성되는 초당 호의 수와 관계없이 스레드 객체를 생성하고 시작시키는 지연 시간에 영향을 받게 되므로 300호부터는 200호와 유사한 연결 설정 지연 값이 측정되어진다. 스레드 풀 방식에서의 연결 설정 지연은 20에서 1,000호까지 1.58에서 5.27ms 사이의 연결 설정 지연 값이 측정되었다.

#### 4.2.2 XML 및 GSMP 기반 연결 관리 방식

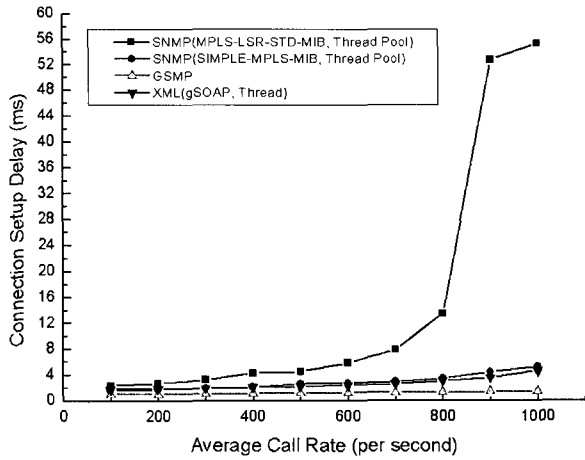
XML 기반의 Axis 방식은 20에서 300호까지 9.21에서 17.94ms 사이의 값이 측정되었으며, SNMP 기반의 스레드 방식과 동일하게 400호부터는 300호와 유사하게 연결 설정 지연 값이 측정되어진다. XML 기반의 gSOAP 방식에 대한 연결 설정 지연은 20에서 1,000호까지 1.82에서 4.69ms 사이의 값이 측정되었다. GSMP 기반의 연결 관리 방식에서의 연결 설정 지연은 20에서 1,000호까지 1.03에서 1.49ms 사이의 값이 측정되었다.

#### 4.2.3 연결 설정 지연 비교 및 분석

(그림 12)는 <표 2>에서 제시된 각 연결 관리 방식 중에서 초당 100호 이상에서 의미 있는 값이 측정되는 SNMP 기반의 스레드 풀, XML의 gSOAP 그리고 GSMP 기반의 연결 관리 방식에 대한 연결 설정 지연을 그래프로 나타낸 것이다. GSMP 기반의 연결 관리 방식에서 측정된 연결 설정 지연은 100에서 1,000호까지 1.04에서 1.49ms 사이의 값이 측정되었으며, XML의 gSOAP 기반 연결 관리 방식에서는 100에서 1,000호까지 1.84에서 4.69ms 사이의 값이 측정되었다. SNMP 기반의 사실 SIMPLE-MPLS-MIB을 사용하는 스레드 풀 방식에서 측정된 연결 설정 지연은 100에서 1,000호까지 1.67에서 5.27ms 사이의 값이 측정되었다. 그러나 SNMP 기반의 표준 MPLS-LSR-STD-MIB을 사용하는 스레드 풀 방식의 경우는 500호까지 연결 설정 지연이 4.54ms이었으나 800호부터 급격히 증가하여 1,000호에서 55.43ms로 측정되었다.

SNMP 기반의 표준 MPLS-LSR-STD-MIB을 사용하는 스레드 풀 방식에서의 연결 설정이 증가한 이유는 순수 연





(그림 12) 연결 관리 방식에 따른 연결 설정 지연

결 설정 지연의 증가 때문이다. <표 2>의 SNMP 기반의 표준 MPLS-LSR-STD-MIB을 사용하는 스레드 풀 방식에서 순수 연결 설정 지연은 500호부터 점차적으로 증가하다가 800호부터는 급격히 증가하고 있다. 이처럼 순수 연결 설정 지연이 급격히 증가하는 이유는 스레드 객체의 시간 지연에 비하여 더 짧은 주기로 호가 생성되므로 호의 생성 시간에 비례하여 짧은 시간에 스레드의 수가 많아지게 되어 스레드 간 문맥교환(context switching)이 빈번히 이루어지기 때문으로 사료된다.

<표 2>와 (그림 12)의 측정 결과를 기반으로 각 연결 관리 방식을 비교 및 분석한 결과, 초당 호의 평균 도착 개수가 100에서 1,000까지 증가하는 경우에 GSMP 기반 연결 관리 방식이 다른 연결 관리 방식보다 연결 설정 지연이 가장 적었다. 다음으로는 XML 기반의 gSOAP 방식, SNMP 기반의 사실 MIB을 사용하는 스레드 풀 방식 그리고 SNMP 기반의 표준 MIB을 사용하는 스레드 풀 방식 순임을 알 수 있었다.

### 5. 결론 및 향후 연구 과제

망의 구성과 용도에 따른 적절한 연결 관리 메커니즘의 도입은 망의 성능을 좌우하게 된다. 본 논문에서는 전달 평면과 제어 평면이 분리되어 있는 중앙 집중형 서버 기반의 망에서 사용될 수 있는 연결 관리 방식들인 SNMP, XML 기반의 웹 서비스 그리고 GSMP를 사용하는 네트워크 개방형 인터페이스 기반의 연결 관리 플랫폼들을 구축하였다. 구축된 연결 관리 플랫폼들을 이용하여 제어기와 스위치 사이에서 각 연결 관리 방식들에 대한 연결 설정 지연을 측정하여 성능을 비교 분석하였다. 측정 결과, 네트워크 개방형 인터페이스인 GSMP를 사용하는 연결 관리 방식이 SNMP 나 XML 기반 연결 관리 방식에 비하여 최소의 연결 설정 지연을 요구함을 확인할 수 있었다.

향후 연구 과제로는 본 연구에서 제시된 연결 관리 방식

들에 대하여 다중 CPU 환경에서 다양한 스레드 풀 방식의 알고리즘을 도입하여 성능을 측정 및 비교 분석하는 것이다.

### 참고 문헌

- [1] 이종현, 예병호, "BcN QoS 스위치 및 라우터 개발방향," *TTA저널*, 제 96호, pp. 69-77, 2004년 12월.
- [2] 이순석, 김영선, 전경표, "BcN의 핵심 인프라: Flow 기반 QoS 보장 네트워크," *Telecommunications Review*, 제 15권 제6호, pp.866-877, 2005년 12월.
- [3] Keith Knightson, Naotaka Morita and Thomas Towle, "NGN Architecture: Generic Principles, Functional Architecture, and Implementation," *IEEE Communications Magazine*, Vol.43, No.10, pp.49-56, October, 2005.
- [4] ITU-T, "Revision 5 of TR-RACF," NGN Focus Group, 7th FGNGN Meeting, June, 2005.
- [5] ITU-T, "Carrier Grade Open Environment Reference Model," OCAF Focus Group, May, 2005.
- [6] J. Case, M. Fedor, M. Schoffstall and J. Davin, "A Simple Network Management Protocol (SNMP)," IETF RFC 1157, May, 1990.
- [7] R. Enns, "NETCONF Configuration Protocol," IETF Internet-Draft, draft-ietf-netconf-prot-11.txt, February, 2006.
- [8] A. Doria, F. Hellstrand, K. Sundell and T. Worster, "General Switch Management Protocol v3," IETF RFC 3292, June, 2002.
- [9] IETF, Network Configuration (netconf) Working Group, <http://www.ietf.org/html.charters/netconf-charter.html>
- [10] Aiko Pras, Thomas Drevers, Remco van de Meent and Dick Quartel, "Comparing the Performance of SNMP and Web Services-Based Management," *eTransactions on Network and Service Management*, Vol.1, No.2, December, 2004.
- [11] IETF, Multiprotocol Label Switching (mpls) Working Group, <http://www.ietf.org/html.charters/mpls-charter.html/>.
- [12] IETF, Common Control and Measurement Plane (ccamp) Working Group, <http://www.ietf.org/html.charters/ccamp-charter.html/>.
- [13] C. Strinivasn, A. Viswanathan and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switch Router (LSR) Management Information Base (MIB)," IETF RFC 3813, June, 2004.
- [14] T. Nadeau, C. Strinivasan and A. Farrel, "Multiprotocol Label Switching (MPLS) Management Overview," IETF RFC 4221, November, 2003.
- [15] Aiko Pras, Jurgen Schonwalder and Olivier Festor,

"XML-Based Management of Networks and Services," *IEEE Communications Magazine*, Vol.42, No.7, pp. 56-57, July, 2004,

- [16] Cisco Systems, Cisco Configuration Registrar, <http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/intro.pdf/>.
- [17] Juniper Networks, JUNOScript, <http://www.juniper.net/techpubs/software/junos/junos73/junoscript73-guide/frameaset.htm/>.
- [18] Lawrence E. Menten, "Experiences in the Application of XML for Device Management," *IEEE Communications Magazine*, Vol.42, No.7, pp.92-100, July, 2004.
- [19] Torsten Klie and Frank Strauss, "Integrating SNMP Agents with XML-Based Management Systems," *IEEE Communications Magazine*, Vol.42, No.7, pp.76-83, July, 2004.
- [20] Mi-Jung Choi, Hyoun-Mi Choi, James W. Hong and Hong-Taek Ju, "XML-Based Configuration Management for IP Network Devices," *IEEE Communications Magazine*, Vol.42, No.7, pp.84-91, July, 2004.
- [21] W3C, XML Protocol Working Group Recommendation, "SOAP Version 1.2 Part 0," <http://www.w3.org/TR/2003/REC-soap12-part0-20030624>, June, 2003.
- [22] Apache Web Services Project, Axis (Java), <http://ws.apache.org/axis/>.
- [23] SourceForge gSOAP Project, <http://sourceforge.net/projects/gsoap2/>.
- [24] AdventNet. Management Builder and Agent Toolkit, <http://www.adventnet.com/>.
- [25] 권태현, 김춘희, 차영욱, "개방형 인터페이스가 적용된 OBS 망의 관리객체 및 분산 망 관리 모델", 정보처리논문지C, 제12-C권 제3호, pp. 449-456, 2005년 6월.



**권 태 현**

e-mail : taehyun@andong.ac.kr  
 2001년 안동대학교 컴퓨터공학과(학사)  
 2003년 안동대학교 컴퓨터공학과  
 (공학석사)  
 2006년 8월 안동대학교 컴퓨터공학과  
 (공학박사)

관심분야: 광 인터넷, 개방형 인터페이스, 망 제어 및 관리 등

**임 영 은**



e-mail : yeyim@comeng.andong.ac.kr  
 2005년 안동대학교 컴퓨터공학과(학사)  
 2005년~현재 안동대학교 컴퓨터공학과  
 (석사과정)  
 관심분야: 웹 서비스, XML, NGN 등

**최 인 상**



e-mail : ischoi@etri.re.kr  
 2003년 안동대학교 컴퓨터공학과(학사)  
 2006년 안동대학교 컴퓨터공학과  
 (공학석사)  
 2006년~현재 과학기술연합대학원  
 대학교(UST) 광대역네트워크공학  
 전공(박사과정)

관심분야: 광 인터넷, 개방형 인터페이스, 망 관리 등

**김 춘 희**



e-mail : chkim@dcu.ac.kr  
 1988년 전남대학교 전산통계학과(학사)  
 1992년 충남대학교 전자계산학과  
 (이학석사)  
 2000년 8월 경북대학교 컴퓨터공학과  
 (공학박사)

1988년~1995년 한국전자통신연구원 연구원  
 2002년~현재 대구사이버대학교 컴퓨터정보학과 조교수  
 관심분야: 고속통신망, 트래픽 제어, 망 관리 등

**차 영 욱**



e-mail : ywcha@andong.ac.kr  
 1987년 경북대학교 전자공학과(학사)  
 1992년 충남대학교 계산통계학과  
 (이학석사)  
 1998년 경북대학교 컴퓨터공학과  
 (공학박사)

1987년~1999년 한국전자통신연구원 선임연구원  
 2003년~2004년 매사추세츠 주립대학 방문교수  
 1999년~현재 안동대학교 컴퓨터공학과 부교수  
 관심분야: 광 인터넷, 개방형 통신망, 망 관리 등