

# 테라 라우터 시스템의 SNMP 망 관리 에이전트 정합 기능 구현

홍선미<sup>o</sup>, 김태일

한국전자통신연구원 라우팅프로토콜팀

Tel : 042-860-4824, Fax : 042-860-5440, E-mail : [smhong@etri.re.kr](mailto:smhong@etri.re.kr)

## Implementation of SNMP Agent Interface Function for Terabit Router

Seon-Mi Hong<sup>o</sup> and Tae-Il Kim

Routing Protocol Team, ETRI

Tel : 042-860-4824, Fax : 042-860-5440, E-mail : [smhong@etri.re.kr](mailto:smhong@etri.re.kr)

### 요 약

SNMP(Simple Network Management Protocol) 기반 망 관리 에이전트 기능을 구현하기 위해서는 에이전트 시스템 특성에 따른 표준 MIB(Management Information Base)을 구현하여 SNMP가 사용할 수 있도록 하여야 한다. 이때, SNMP 망 관리 에이전트 기능에 구현된 MIB은 망 관리를 위해 해당 객체(object)에 접근할 수 있는 가상의 정보일 뿐 MIB에 정의된 각 객체에 대한 값은 해당 MIB 정보를 제공하는 시스템 기능(이하 시스템 기능이라 함.)에서 제공된다. 이때 SNMP 망 관리 에이전트 기능에서는 시스템 기능과 어떠한 정합 방법으로 통신하는가 하는 문제는 시스템 구현 사항이다. 일반적으로 시스템 기능에서 각 객체별로 접근 가능한 API를 제공하여 SNMP 망 관리 에이전트 기능에서 이를 이용하도록 구현되어 있다.

본 논문에서는 이러한 방식의 문제점을 살펴보고, 테라 라우터에 구현하는 SMUX(SNMP Multiplexing) 프로토콜 정합 방법 및 IPC 정합 방법을 설명한다. 또한 IPC 통신시 재사용 가능한 메시지 구조를 제안한다.

### 1. 서론

망 관리 기능은 통신망과 서비스에 대한 관리 표준으로, 독자적인 개발에 근거를 둔 망 요소(network element)들과 서비스에 대하여 운용 관리 및 유지보수 등을 일관적이고 표준화된 행위로 체계적으로 수행하기 위한 인터페이스이다. 이러한 망 관리 표준은 망 관리 매니저 시스템과 망 관리 에이전트 시스템간 정의된 규격으로, 일반적으로 IP 망에서는 SNMP와 같은 표준 프로토콜을 이용하여 망 관리 기능을 수행한다[1].

망 관리 매니저의 요구에 따라 SNMP 망 관리 에이전트 기능에 의해 read 또는 write 되는 관리 정보(managed object)들은 시스템 기능에 구현되며, kernel 영역과 user 영역에 분산되어 존재할 수 있다. IP 망의 관리를 위해 기본적으로 구현되는 MIB-II의 네트워크 프로토콜 MIB(예: TCP Group, IP Group 등)과 같이 kernel 영역에 존재하는 관리 정보는 SNMP 망 관리 에이전트 기능에서 kernel 변수를 read 할 수 있도록 구현된다. 그러나, user 영역에 존재하는 관리 정보는 시스템에서 정한 독자적인 방법으로 접근하여 read 또는 write하여야 한다. 일반적으로 소형 시스템의 SNMP 망 관리 에이전트 기능에서는 시스템 기능에서 정의한 SNMP-API를 이용하여 user 영역의 관리 정보

를 read하거나 write 하도록 구현되었다.

그러나, 테라 라우터에 구현되는 라우팅 프로토콜 기능, 시스템 운용 관리를 위한 기능과 같이 SNMP 망 관리 에이전트 기능과 서로 다른 프로세스로 구현될 경우, 이와 같은 방법으로 MIB 모듈을 정합하면 통신에 문제가 있을 뿐만 아니라, 상기 소프트웨어들을 다른 시스템에서 재사용시 기능 정합에 많은 어려움이 있다. 본 논문은 테라 라우터에서 이러한 문제를 해결하기 위해 제안된 MIB 정합 방법을 소개한다. 테라 라우터는 소프트웨어 기능의 특성에 따라 SMUX 프로토콜 및 IPC 메시지를 이용하여 정합한다.

본 논문의 구성은 서론에 이어 2장에서 테라 라우터에 구현되는 SNMP 망 관리 에이전트 기능 구조에 대해 살펴보고, 3장과 4장에서 SMUX 프로토콜을 이용한 MIB 정합 방법과 IPC 메커니즘을 이용한 MIB 정합 방법을 설명하고, 재사용이 가능한 메시지 구조를 제안하며, 5장에서 결론을 맺었다.

### 2. SNMP 망 관리 에이전트 기능

#### 2.1 구현 MIB

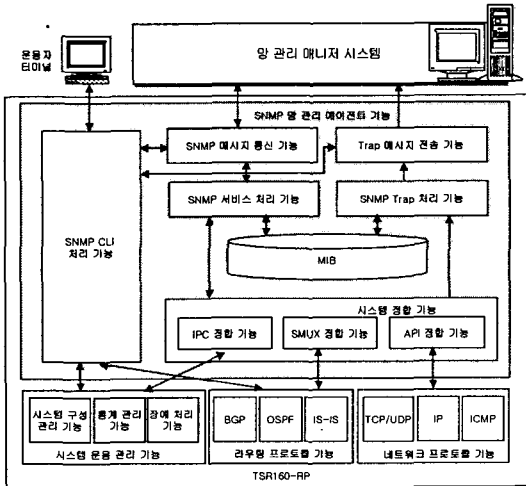
SNMP에서 관리 정보는 가상 정보 공간에 존재하

는 관리 객체(managed object)들의 집합이다. 이 집합을 MIB이라고 정의하며, 시스템의 특성에 따라 망 관리에 필요한 표준 MIB이 구현된다.

테라 라우터 시스템은 SNMPv2 기반의 망 관리 에이전트 기능을 제공한다. IP 망의 기본 MIB인 MIB-II, 라우팅 프로토콜 관리를 위한 BGPv4 MIB, OSPFv2 MIB, OSPFv3 MIB, IS-IS MIB, IPv6 기능을 위한 ICMPv6 group MIB, IPng MIB, 시스템 운용 관리를 위한 시스템 및 인터페이스 그룹 MIB, SONET/SDH 인터페이스 MIB, Entity MIB, 그리고, 시스템 상태 및 장애 관리를 위해 Private MIB이 추가된다.

## 2.2 기능 구조

[그림 1]은 테라 라우터의 SNMP 망 관리 에이전트 기능 구조를 나타낸 것이다. 각 기능별 세부 내용은 다음과 같다.



[그림 1] 테라 라우터 시스템의 망 관리 에이전트 기능 구조

SNMP 메시지 통신 기능과 Trap 메시지 전송 기능은 망 관리 매니저 시스템으로부터 SNMP 메시지를 수신하고, 이의 처리 결과 및 이벤트 정보를 송신하기 위한 통신 기능이다. SNMP는 UDP Socket 통신을 하며, 161번과 162번 포트를 사용한다.

SNMP 서비스 처리 기능은 SNMP 메시지 통신 기능에서 수신한 SNMP 패킷을 분석(Decoding)하여 GetRequest/GetNextRequest/SetRequest PDU에 대해 처리한다. SNMP version 정보와 community name이 확인되면 PDU의 종류에 따라 서비스를 제공하고, 확인되지 않으면 폐기한다. PDU 종류별로 VarBind 정보를 분석하여 각 객체에 대한 값을 read 또는 write하는 시스템 정합 기능으로 객체 정보를 전달한다. 한편, 망 관리 매니저 시스템으로 수행 결과를 송신하기 위한 Response PDU를 SNMP 패킷으로 변환(Encoding)하여 SNMP 메시지 통신 기능으로 보낸다.

SNMP Trap 처리 기능은 테라 라우터 시스템에서

장애나 상태 등의 변경이 발생할 경우, Trap 메시지를 수신하여 처리하는 기능을 수행한다.

MIB은 망 관리 매니저 시스템과 통신할 수 있는 논리적 객체 정보와 물리적인 정보와의 관계를 Tree 및 테이블 형태로 관리하며, 기본적인 Binary Search를 이용해 MIB을 검색한다.

시스템 정합 기능은 시스템 운용 관리 기능, 라우팅 프로토콜 기능, 네트워크 프로토콜 기능과 통신하여 망 관리 매니저가 요구한 각 MIB에 대한 객체의 실제 데이터를 검색 및 변경한다. 테라 라우터에서는 실제 MIB 정보를 처리하는 기능의 특성에 따라 IPC 정합, SMUX 정합, API 정합 기능을 제공한다.

SNMP CLI 처리 기능은 망 관리 매니저 시스템과 통신하기 위한 정보 및 망 관리 에이전트 기능 동작에 필요한 정보를 CLI 명령으로 수신하여 설정 및 출력하는 기능이다.

## 2.3 MIB 정합 방법

망 관리 매니저의 요구에 따라 SNMP 망 관리 에이전트 기능에 의해 read 또는 write 되는 관리 정보들은 kernel과 user 영역 사이에 분산되어 존재할 수 있다. Kernel 영역에 존재하는 관리 정보들은 SNMP 망 관리 에이전트가 kernel memory에 접근하여 해당 관리 정보를 read 하며, user 영역에 존재하는 관리 정보는 시스템에서 정한 독자적인 방법으로 접근하여 read 또는 write 한다. 일반적으로 소형 시스템에 구현된 SNMP 망 관리 에이전트 기능은 관리 정보를 처리하는 시스템 기능에서 제공하는 API를 이용하여 user 영역의 관리 정보를 read하거나 write 한다.

그러나, 테라 라우터에 구현되는 SNMP 망 관리 에이전트 기능은 BGP, OSPF, IS-IS와 같은 라우팅 프로토콜과 시스템 운용 관리 기능과 서로 프로세스로 구현되며, 이들간의 MIB 정합 기능을 구현하기 위하여 SNMP 망 관리 에이전트와 이들 프로세스간 통신이 문제가 된다[3]. 이러한 문제를 해결하기 위하여 SMUX 프로토콜을 이용하거나, IPC 메커니즘을 이용하여야 한다.

테라 라우터에서는 이들 두 가지 방법을 이용하여 SNMP 망 관리 에이전트 기능 정합을 구현한다. 먼저, 라우팅 프로토콜과 같이 여러 시스템에 적용 가능한 프로토콜의 MIB은 SMUX 인터페이스를 통해 제공되며, 시스템 운용 관리 기능과 같이 시스템 종속적인 기능에서 시스템 관리를 위해 제공하는 관리 정보는 IPC 메커니즘을 이용하여 통신한다.

## 3. SMUX 정합 방법

### 3.1 라우팅 프로토콜 MIB과 SMUX 정합 구현

테라 라우터는 BGP, OSPF, IS-IS와 같은 라우팅 프로토콜이 수행되며, SNMP 망 관리 에이전트 기능은 라우팅 프로토콜 표준 MIB과 SMUX 프로토콜을 이용하여 정합한다.

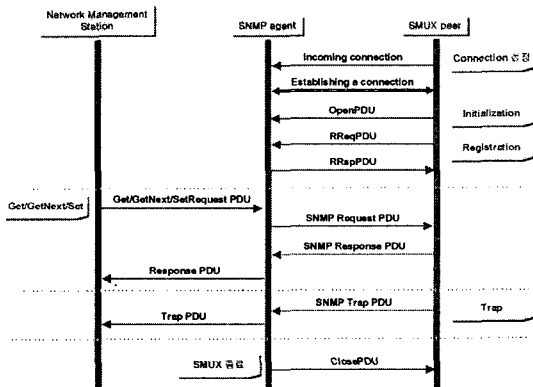
SMUX는 SMUX peer라고 하는 user 프로세스에서 라우팅 프로토콜 MIB에 대한 tree를 구성하여 망 관리 에이전트 기능으로 연결 및 관리를 위한 초기화 및 등록 절차를 거쳐 관리 정보 교환의 역할을 수행한다. 이때 SNMP 망 관리 에이전트 기능은 망 관리 매니저와 161번 포트를 사용하여 UDP 통신으로 SNMP 패킷을 송수신 하며, SMUX peer와는 TCP 통신으로 199번 포트를 사용하여 SNMP PDU를 처리한다.

### 3.2 SMUX Architecture

SNMP 망 관리 에이전트 기능과 SMUX peer간 MIB module 정합 방법으로 request-response model과 cache-ahead model이 있다. Request-response model은 SNMP 망 관리 에이전트가 망 관리 매니저로부터 수신한 SNMP request를 SMUX peer로 전달하여 SMUX peer에서 SNMP 오퍼레이션을 수행한 후 SNMP response를 전달하면, 이를 SNMP 망 관리 에이전트 기능에서 망 관리 매니저 시스템으로 전달하는 구조로 수행된다. Cache-ahead model은 SMUX peer가 SNMP 망 관리 에이전트에 등록된 후, 라우팅 프로토콜 MIB module을 위한 subtree를 갖고 있는 SNMP 망 관리 에이전트 기능으로 주기적인 update 명령을 보내어 SNMP 망 관리 에이전트에서 MIB module을 처리하도록 하는 구조로 수행된다. 테라 라우터는 request-response model을 이용하여 구현한다.

### 3.3 SMUX Protocol

[그림 2]는 request-response model을 적용한 SMUX protocol 동작 흐름을 나타낸 것이다.



[그림 2] SMUX protocol

### 3.4 SMUX Protocol Data Units

SMUX PDU는 SMUX 인증 및 초기화 기능을 수행하는 OpenPDU, SMUX 종료를 위한 ClosePDU, SMUX peer의 등록을 수행하는 RRReqPDU, 등록 응답을 수행하는 RRspPDU, SMUX peer에서 set operation에 대한 commit/rollback을 위한 SOutPDU로 구성된다. 이들

PDU는 ASN.1(Abstract Syntax Notation One)을 사용하여 정의하며, tags는 SNMP-generic PDU와 구분하기 위하여 application-wide로 정의한다. SNMP 망 관리 에이전트 기능에 SNMP GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU를 SMUX peer로 보낼 때, 망 관리 매니저로부터 수신한 request-id를 사용한다.

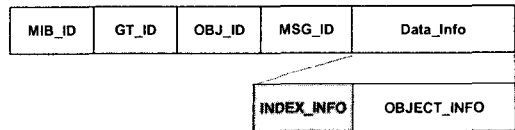
## 4. IPC 정합 방법

### 4.1 시스템 운용 관리 MIB과 IPC 정합 구현

테라 라우터의 구성 관리, 통계 관리, 장애 관리를 위한 MIB은 MIB-II, Entity MIB, SONET/SDH interface MIB이 기본으로 구현되고, 테라 라우터 각 장치별 세부 상태 관리 및 장애 관리를 위해 private MIB을 추가 정의하여 망 관리 기능을 수행한다. 테라 라우터에서는 시스템 운용 관리 기능을 위한 MIB은 IPC 메시지를 이용하여 통신한다. IPC 통신을 위한 메시지 구조는 MIB을 구성하는 객체의 syntax, index의 수, 객체의 성격 등에 무관하게 메시지 처리가 가능하며, MIB에 정의된 객체의 id를 이용해 header의 정보를 분석하므로 시스템 운용 관리 기능과 독립적으로 구현이 가능하므로 망 관리 에이전트 기능을 타 시스템에 이식할 경우 재사용이 가능한 구조이다.

### 4.2 메시지 구조

테라 라우터 시스템의 SNMP 망 관리 에이전트 기능에서 시스템 운용 관리 기능과 정합을 위해 사용한 메시지 구조는 [그림 3]과 같다. 이 메시지 구조는 SNMP PDU의 Variable-Binding에 포함된 객체 정보 및 PDU type에 대한 정보를 포함하며, 송수신시 동일 메시지 구조를 사용하지만, 객체의 특성에 따라 Data\_Info의 구조는 달라진다. 각 필드별 상세한 내용은 다음과 같다.



[그림 3] IPC 메시지 구조

MIB\_ID는 구현되는 MIB을 식별하기 위한 정보이며, GT\_ID는 MIB의 하위 객체인 Group과 Table의 Object ID 조합으로 정의한다. OBJ\_ID는 해당 MIB의 최하위 객체 ID로, MIB에 정의된 OID와 동일하게 사용한다. MSG\_ID는 SNMP PDU type 및 시스템 운용 관리 기능에서의 처리 결과를 나타낸다.

Data\_Info는 OBJ\_ID에 대한 데이터 구조를 나타낸다. Scalar Object인 경우 OBJECT\_INFO로만 구성되며, scalar object가 속한 group내의 모든 객체에 대해 union으로 선언된 멤버 중 하나를 나타낸다. Columnar Object인 경우 INDEX\_INFO와 OBJECT\_INFO로 구성된다. INDEX\_INFO는 테이블의 엔트리 중 index 정보

로 구성된 구조체이며, OBJECT\_INFO는 테이블의 엔트리중 index 정보를 뺀 나머지 엔트리에 대해 union으로 선언된 멤버 중 하나를 나타낸다.

본 메시지 구조의 특징은 MIB\_ID, GT\_ID, OBJ\_ID는 MIB에 정의된 OID를 그대로 이용하므로 새로운 MIB이 추가되거나 삭제 또는 변경이 되어도 메시지 구조를 변경할 필요는 없다.

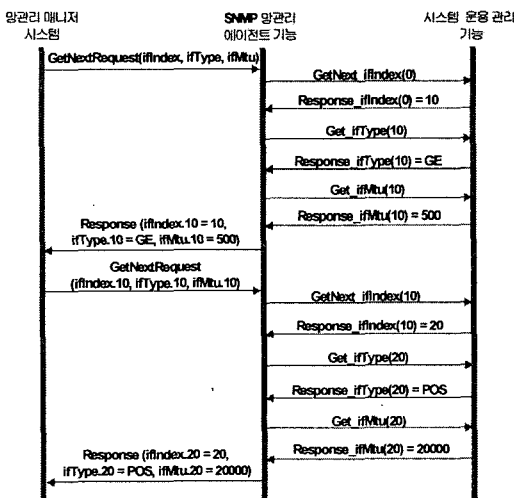
### 4.3 GetNextRequest의 처리

SNMP PDU중 GetNextRequest는 Columnar Object의 테이블 정보를 검색하기 위해 망 관리에서 가장 많이 사용하는 오퍼레이션 중의 하나이다. 따라서 GetNextRequest PDU의 처리 시간을 단축함으로써 망 관리 성능을 개선할 수 있다. 기존 API 방식에서의 GetNextRequest PDU의 처리는 Next Index 정보를 검색하기 위한 API를 호출한 후, 검색된 Index 값을 Key로 하여 해당 객체(object)를 검색하기 위한 API를 다시 호출하였다. 따라서, GetNextRequest PDU를 처리하기 위해 Index 및 객체에 대한 API를 각각 호출한 것이다.

본 논문에서 제안한 IPC 메시지 구조를 이용하여 통신할 경우, 한번의 메시지 전송으로 Index와 객체 정보를 동시에 처리할 수 있다. [그림 4]는 ifTable의 객체를 예로 사용하여 GetNextRequest PDU 처리 흐름도를 기존의 API 방식과 IPC 방식을 비교하여 나타내었다.

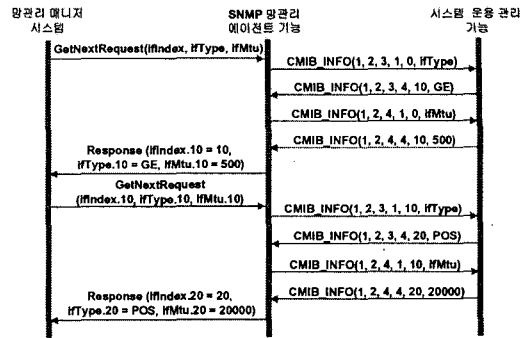
(a) ifTable의 예

ifIndex	ifType	ifMtu
10	GE	500
20	POS	20000



(b) 기존 API를 이용한 기능 흐름도(예)

MIB\_ID = RFC1213(1)  
 GT\_ID = ifTable(2)  
 MSG\_ID = SNMP\_Get(0), SNMP\_GetNext(1), SNMP\_Set(2),  
 SNMP\_Noti(3), SNMP\_ResOk(4), SNMP\_ResNok(5)



(c) 제안된 메시지 구조를 이용한 기능 흐름도(예)

[그림 4] GetNextRequest 처리 흐름도(예)

## 5. 결론

본 논문은 테라 라우터 시스템의 SNMP 망 관리 에이전트 기능의 MIB 정합 방법에 대한 논하였다. 관리 대상 시스템에서 SNMP 망 관리 에이전트 기능을 구현할 때 시스템 기능으로부터 MIB 객체에 대한 실제 값을 가져오는 시스템 정합 기능 부분이 큰 문제로 대두된다. 이 부분은 표준화되지 않아 시스템 환경에 따라 시스템 종속적으로 구현되기 때문에 망 관리 에이전트 기능과 정합을 위해서는 MIB에 정의된 수 없이 많은 객체에 대한 정보를 제공하기 위해 일일이 API를 정의하여 망 관리 정보를 제공하였다.

본 논문에서는 기존의 API 정합시 발생한 문제를 해결하기 위해 SMUX 프로토콜 및 IPC 메시지를 이용한 정합 방법 및 메시지 구조를 제안하였다. 여러 시스템에 동작되는 라우팅 프로토콜은 SMUX 프로토콜을 사용하여 라우팅 프로토콜 기능의 재사용에 중점을 두었으며, 시스템 종속적인 시스템 운용 관리 기능은 망 관리 에이전트 기능의 재사용에 중점을 두어 설계를 하였다. 따라서 본 논문의 SNMP 망 관리 에이전트 정합 방법은 소프트웨어 재사용면에서 효과적이며, 특히 IPC 메시지를 이용한 시스템 운용 관리 MIB의 정합 방법은 GetNextRequest PDU의 처리 시 송수신 메시지 수를 줄임으로서 성능 개선의 효과를 기대할 수 있다. IPC 통신 방법은 시스템의 자원을 효율적으로 관리하면 소프트웨어 재사용 및 성능 개선의 효과를 기대할 수 있을 것이다.

### [ 참고문헌 ]

- [1] 권영희, 김영집, 박애순, 조은경, 구연설, "SNMP 에이전트 소프트웨어의 재사용을 위한 구조 설계," 정보처리논문집, v.7 no.6, pp 1803 ~ 1809, 2000.6.
- [2] William Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Third Edition.
- [3] RFC 1227, SNMP MUX Protocol and MIB