

SNMP 기반 NG-SDH 관리 기능 구현

조은영, 남현순, 예병호, 박용석(*), 이동훈(*)

ETRI 광대역통합망연구단 광전달망제어팀, (주)아이티(*)

Implementation of SNMP Agent on NG-SDH System

Eun-Young Cho, Hyun-Soon Nam, Byung-Ho Ye, Yong-Suk Park, Dong-choon Lee

ETRI Broadband Convergence Network Lab., OTN S/W Team

{eycho, namhs, bhye}@etri.re.kr, {piao1234, dclee}@it.co.kr

Abstract

This paper describes the design and implementation of a SNMP Agent for NG-SDH transmission system. One of the most important items in SDH network management is the information model that implements every function of SDH equipment at network level. Although the TMN framework of SDH information model has been defined quite well, CMIP constraints of a real implementation urge vendors to implement proprietary features, and make it more difficult to maintain an integrated view of multivendor SDH networks management. To solve these aspects, this work proposes SNMP information model that allows a simple view of NG-SDH network management.

1. 서론

동기식 전송 시스템 관리는 ITU-T SG15 에서 추진되어 왔으며 표준 관리는 CMIP 을 이용한 정보 모델로 개발되었다. 그러나 실제 개발 제품은 하나의 모델과 프로토콜로 추진되지 못함에 따라 CORBA 및 SNMP, XML 기반의 관리 체계를 모두 수용하고 있다. 따라서 최근에는 UML 표기 방식을 이용하여 프로토콜 중립적인 관리 정보 모델을 정의하고 있는데 OTN 시스템부터 적용되고 있다. 현재까지 한국에서 SDH 시스템은 주로 CMIP 기반으로 개발되었지만 프로토콜의 부담 등으로 상호 운용이 용이한 환경으로 변하고 있다. 이에 본 논문에서는 IETF 의 SNMP MIB 과 자체 정의한 NG-SDH MIB 을 이용하여 통합된 시스템 사례를 소개하고 향후 통신망 사업자 관리 시스템과의 연계를 위해 제공한 다양한 기능에 대해 소개하고자 한다.

2. SONET SNMP 표준 관리 정보

현재 IETF RFC2558 에서는 SONET/SDH 관리를 위한 정보가 정의되어 있다. 그러나 이것은 물리층의 간단한 표현 및 전송 계층의 성능 관리 정보 수준에서 정의되어 다양한 전송 관리 기능의 수용이 어려운 상황이다. 따라서 SNMP 에 기반한 전송 시스템 관리를

위해서는 자체 관리 정보의 정의가 불가피하다. 본 장에서는 먼저 장치 구성 등 표준 관리 정보의 구현 사례에 대해서 소개하기로 한다.

2.1 구성 관리

전송 시스템의 구성 관리에는 대표적으로 장치와 신호 구성이 포함된다. SNMP MIB 에서는 entityPhysicalTable 과 인터페이스로 표현되는 ifTable 이 사용된다. 물리적 구성인 랙, 셸프, 유니트, FAN 등에 대해서 stack(11), chassis(3), container(5), fan(7) 등의 PhysicalClass 값을 가지는 객체 인스턴스를 유지하게 되고 SDH 다중화 구간이나 중계 구간, AU(Administrative Unit) 등의 계위에 대해서 ifTable 요소를 유지하게 된다.

그런데 여기에서 SNMP MIB 에 속한 각 테이블의 Index 는 해당 정보를 효율적으로 유지하는 중요한 역할을 한다. 이것은 CMIP 기반의 agent 에서도 동일하게 적용되는 사안으로 객체 인스턴스들은 논리적 관계성이 존재하고 이를 유지해야 한다. 이것은 대표적으로 장치 구성 정보와 신호 구성 정보간, 신호 구성 정보와 연결 관리 정보 혹은 절체 관리 정보간에 존재하게 된다. 또한 이렇게 상호 관련된 정보들이 신속하게 접근되고 상태도 일관성 있게 유지되어야 한다. 따라서 객체 접근의 열쇠가 되는 Index 값 정의 규칙이 중요한데 물리 객체와 신호 객체 Index 의 연관성이 존재하는 방식이 되면 효율적으로 사용될 수 있다. 가령 다음과 같은 예제에서 entityPhysicalTable index 는 랙, 셸프, 유니트까지의 조합을 키로 사용하게 되고 ifTable 에서는 포트번호까지 조합된 값을 사용된다.

Rack	Shelf	Unit	Port
<i>l..r</i>	<i>l..s</i>	<i>l..u</i>	<i>l..p</i>
ifIndex			
entPhysicalIndex			

그림 1. 구성 정보 인덱스 사용 예

또한 전송시스템에서는 Cross Connection 정보가 유지되어야 하는데 이는 ifStackTable 을 응용할 수 있다. ifStackTable 은 망 인터페이스의 다양한 계층간의

관계성 정보를 포함하는데 상호 연결되는 신호 객체를 지정하여 연결 설정 및 해제에 이용할 수 있다. ifStackStatus.from.to 를 active 혹은 destroy 로 지정하여 원하는 동작을 실행할 수 있다. 다음은 각 구성테이블의 예이다.

```
SNMPv2-SMI::mib-2.47.1.1.1.1.2.1200 = STRING:
"NGTS-COM"
SNMPv2-SMI::mib-2.47.1.1.1.1.2.1201 = STRING: "SCCA"
SNMPv2-SMI::mib-2.47.1.1.1.1.2.1202 = STRING: "SCCA"
SNMPv2-SMI::mib-2.47.1.1.1.1.2.1203 = STRING: "BSIA"
SNMPv2-SMI::mib-2.47.1.1.1.1.2.1204 = STRING: "NONE"
SNMPv2-SMI::mib-2.47.1.1.1.1.2.1205 = STRING: "BSIA"
IF-MIB::ifDescr.12051050 = STRING: STM-16
IF-MIB::ifDescr.12051051 = STRING: AU3
IF-MIB::ifDescr.12051052 = STRING: AU3
IF-MIB::ifDescr.12051053 = STRING: AU3
IF-MIB::ifOperStatus.12051050 = INTEGER: up(1)
IF-MIB::ifOperStatus.12051051 = INTEGER: up(1)
IF-MIB::ifOperStatus.12051052 = INTEGER: up(1)
IF-MIB::ifOperStatus.12051053 = INTEGER: up(1)
IF-MIB::ifType.12051050 = INTEGER: sonet(39)
IF-MIB::ifType.12051051 = INTEGER: sonet(39)
IF-MIB::ifType.12051052 = INTEGER: sonet(39)
IF-MIB::ifType.12051053 = INTEGER: sonet(39)
IF-MIB::ifSpeed.12051050 = Gauge32: 2500000
IF-MIB::ifSpeed.12051051 = Gauge32: 2500000
IF-MIB::ifSpeed.12051052 = Gauge32: 2500000
IF-MIB::ifSpeed.12051053 = Gauge32: 2500000
IF-MIB::ifStackStatus.12031051.12051051 = INTEGER:
active(1)
IF-MIB::ifStackStatus.12031052.12051052 = INTEGER:
active(1)
IF-MIB::ifStackStatus.12031053.12051053 = INTEGER:
active(1)
```

그림 2. 구성 정보 인스턴스 예

2.2 장애 관리

장애 관리는 전송 망에서 발생할 수 있는 장애에 대해서 신속하게 인지되고 고장 개소 추적이 용이하며 효과적인 복구가 이루어지도록 고려되어야 한다. SNMP에서는 대표적으로 cold start, warm start 가 정의되어 있고 link up/down trap 을 이용하여 manager 가 장애를 인지하여 조치하도록 제공한다. 그런데 장애 정보뿐만 아니라 모든 장치의 상태를 관리자에게 통지하는 수단이 trap 을 이용하는 것이므로 구성 변경, 주기적 성능 보고도 동일한 과정으로 전달된다. 따라서 표준 정보 외에도 장치 고유의 모든 구성, 장애, 성능에 대한 통지 사항들은 private MIB 에서 정의하여 사용하게 된다.

2.3 성능 관리

SONET-MIB 은 회선 provisioning 정보를 유지하는데 대부분은 성능 감시 정보에 대한 정의를 포함한다. PerfHist-TC-MIB 에 정의된 PerfCurrentCount 와 PerfIntervalCount 가 Gauge32 타입으로 bBE, eS, sES

등의 성능 파라미터 감시에 사용되고 이것이 중계 구간, 다중화 구간, 종속 신호 구간의 현재 성능값 유지에 사용된다. SDH 시스템에서는 15 분과 1 일 주기의 성능 감시가 표준으로 정의되어 있는데 SONET-MIB 에 정의된 테이블은 15 분 단위의 성능 수집에 적용하고 1 일 및 임계치 정보는 별도의 테이블을 정의하여 사용하였다.

```
SonetSectionCurrentEntry ::= SEQUENCE {
sonetSectionCurrentStatus Integer32,
sonetSectionCurrentESs PerfCurrentCount,
sonetSectionCurrentSESs PerfCurrentCount,
sonetSectionCurrentSEFSs PerfCurrentCount,
sonetSectionCurrentCVs PerfCurrentCount
}
SonetLineIntervalEntry ::= SEQUENCE {
sonetLineIntervalNumber Integer32,
sonetLineIntervalESs PerfIntervalCount,
sonetLineIntervalSESs PerfIntervalCount,
sonetLineIntervalCVs PerfIntervalCount,
sonetLineIntervalUASS PerfIntervalCount,
sonetLineIntervalValidData TruthValue
}
```

그림 3. SONET-MIB 성능관리 정보 예

3. NG-SDH 비표준 관리 정보

IETF 의 SNMP 관련 RFC 에서 SONET-MIB 은 앞에서 보았듯이 전송 계층별 성능 수집 정보 유지가 주된 내용이다. 따라서 ITU-T 의 절체 관리는 비표준을 도입할 수밖에 없다. 본 장에서는 NG-SDH 관리를 위해 도입한 private MIB 과 그 효율성에 대하여 논의하기로 한다.

3.1 구성 관리

NG-SDH 시스템의 구성 요소로써 필요한 사항을 보면 교차연결 관리, 절체 관리와 동기클럭 관리 및 VCAT 관리 등이 대표적인 사항이다. 교차연결 관리는 2 장에서와 같이 ifStackTable 을 적용하였고 EMS GUI 를 보면 그림 5.와 같다. 모든 관리 기능은 SNMP manager 의 set 명령으로 실행된다. 절체 관리는 ITU-T SG15 G.774.03 을 참조하여 유사한 ngtsApsConfigTableEntry 를 정의하였는데 절체 명령 속성에 lockout, forced, manual 등 미리 정의된 타입 중의 하나를 지정하면 실행 결과를 상태 속성에 반영하게 된다. 동기클럭 관리는 가능한 클럭원을 선택하도록 제공하는 인터페이스로 다음과 같이 정의하여 사용하였다.

```
ngtsSyncClockSource OBJECT-TYPE
SYNTAX INTEGER {
DOTS(1),
DOTS(2),
Ref1(3),
Ref2(4),
Ref3(5),
Ref4(6),
HoldOver(7),
```

```
FreeRun(8) }
MAX-ACCESS read-write
STATUS current
DESCRIPTION "This object specifies
the Synchronization Clock Source"
::= { ngtsSynchSwitch 4 }
```

그림 4. 동기클럭관리 정보 예

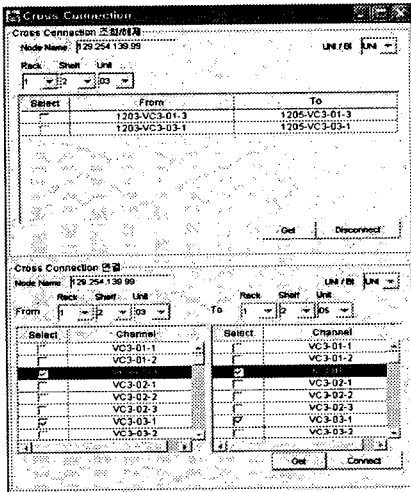


그림 5 교차연결 구성도

VCAT 관리는 허용/금지, 속도, 현재 상태 등의 정보가 필요하므로 자체 MIB 으로 정의하고 신호 mapping 관계는 ifStackTable 을 활용하였다. 표준 SONET-MIB 의 sonetMediumLoopbackConfig 객체는 루프백 구성의 허용/금지를 나타내는데 회선 루프백 시험에 이용하였다.

3.2 장애 관리

SNMP 기반 관리 기능에서는 trap 에 의한 장애 보고를 수행한다. 그런데 CMIP 기반 관리와 비교하면 장애 정보의 분류나 등급 설정, 로그 등의 기능이 덜 세분화 되어있다. 따라서 관리자 측에 섬세한 제어 기능을 제공하기 위해서 경보의 종류를 ITU-T SG4 표준 체계와 같이 장치 경보, 신호 경보 등으로 분류하고 등급도 조절할 수 있도록 제공하며 모든 trap 을 기록하는 테이블을 정의하여 사용하였다. 다음은 EMS 로 전달된 trap 의 예이다.

```
rept_processFM: ptr->aid.rackNo = 1
rept_processFM: ptr->aid.shelfNo = 2
rept_processFM: ptr->aid.unitNo = 8
rept_processFM: ptr->aid.portNo = 7
rept_processFM: ptr->alarmType(Cond) = 17
alarmType: PORT_TX_FAIL
rept_processFM: ptr->serviceEffect = 1
rept_processFM: ptr->severityType = 1
unitType: BLK
```

```
rept_processFM: ptr->rtxType = 3
rept_processFM: ptr->pathNo = 0
rept_processFM: ptr->date = 2004-04-20
rept_processFM: ptr->time = 14:41:32
rept_processFM: communications Alarm ...
rept_processFM: EVENT date&time =
2004-04-20,14:41:32.3,+9:30
rept_processFM: (INT32)strlen(ocrtime) = 27
rept_processFM
CreateAndAddNewNgtsAlarmTableEntry...
SendV2Trap: ngtsAlarmIndexType = 1
SendV2Trap: ngtsAlarmIndex = 12080807
SendV2Trap: ngtsAlarmCond = 217
SendV2Trap: ngtsAlarmSeverity = 1
SendV2Trap: ngtsAlarmSrveff = 1
... End of rept_processFM ...
```

그림 6. 장애 보고 예

EMS 로 보고된 내용에는 구성, 장애, 성능 정보 등 모든 정보가 기록되는데 가령 절체 보고가 포함된 로그를 조회하면 다음과 같이 나타난다.

```
$snmpwalk -v2c -c public 129.254.139.99 private
```

```
SNMPv2-SMI::enterprises.2542.4.7.2.1.2.100.3 = STRING:
"APS 1203:FORCED(P), 2004-04-23 18:02:57"
SNMPv2-SMI::enterprises.2542.4.7.2.1.2.100.4 = STRING:
"APS 1203:CLEAR(W), 2004-04-23 18:03:05"
SNMPv2-SMI::enterprises.2542.4.7.2.1.2.100.5 = STRING:
"APS 1203:FORCED(P), 2004-04-23 18:16:20"
SNMPv2-SMI::enterprises.2542.4.7.2.1.2.100.6 = STRING:
"APS 1203:FORCED(W), 2004-04-23 18:16:24"
```

3.3 성능 관리

먼저 표준 성능 정보는 다음을 사용할 수 있다.

Interface	IETF RFC2558
VT (VC11/12/2, TU11/12/2, TUG)	SonetVTCurrentEntry SonetVTIntervalEntry SonetFarEndVTCurrentEntry SonetFarEndVTIntervalEntry
Path (VC3/4, AU3/4, AUG)	SonetPathCurrentEntry SonetPathIntervalEntry SonetFarEndPathCurrentEntry SonetFarEndPathIntervalEntry
Line (Regeneration Section)	SonetLineCurrentEntry SonetLineIntervalEntry SonetFarEndLineCurrentEntry SonetFarEndLineIntervalEntry
Section (Multiplex Section)	SonetSectionCurrentEntry SonetSectionIntervalEntry

그림 7. SONET/SDH 계층별 성능관리

이와 같은 표준 테이블을 사용하고 1일 성능 자료를 위해 필요한 부분은 enterprise MIB 을 정의하여 SonetSectionTotalEntry, SonetLineTotalEntry, Sonet-FarEndLineTotalEntry 등으로 사용하였다. 또한 임계치 정보 유지가 필요한데 예로써 다음과 같이

정의하여 구성되는 신호 정보에 따라 생성하였다. 이와 같은 정보를 15 분과 1 일 각각에 대해서 유지하여 ITU-T 와 같은 표준 SDH 성능 임계치 관리 기능을 제공하였다.

```

NgtsSonetThEntry ::= SEQUENCE {
    ngtsSonetThIndex      Unsigned32,
    ngtsSonetThVariable  OBJECT IDENTIFIER,
    ngtsSonetThValue     Gauge32 }
    
```

그림 8. 임계치 관리정보 예

4. EMS-NE 운용 사례

EMS는 NE와 SNMP연동에 의해 장치 구성 상태 및 경보 상태를 실제와 동일하게 유지하면서 GUI를 통한 운용자의 관리를 가능하게 한다. 가장 기본적인 화면은 그림 9와 같이 랙 형상과 유니트의 실장 상태 및 경보 상태 표시이다. 그 외에 절체 관리, 교차연결 관리, 동기 클럭 관리 등의 기능을 수행하게 된다.

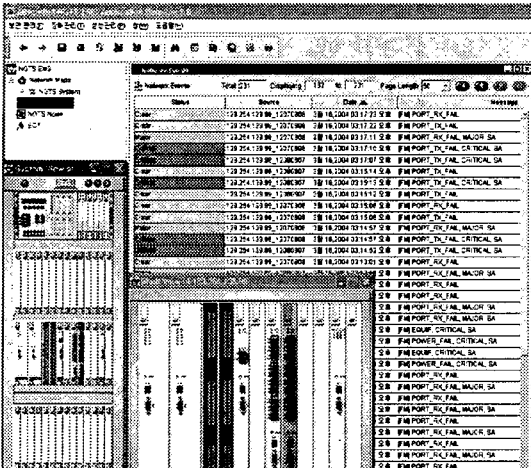


그림 9. EMS 운용자 화면

위 그림은 SNMP agent로부터 수집된 구성 정보와 장애, 성능 정보 수집 결과를 보여준다. 랙 및 셀프 단위의 유니트별 실태장, 장애 정보를 나타내고 있다. SNMP MIB에서 관련 정보의 일관성을 위해 각 객체 생성 및 Operational-Status 유지는 다음과 같이 실행하였다.

- (1) 유니트 실장: 유니트 타입에 따라 TDM 혹은 GbE if 객체 생성 및 opState 설정
- (2) 유니트 실장: 유니트 타입에 따라 TDM 혹은 GbE if 객체 삭제
- (3) 장치 경보: 해당 유니트에 대한 entity 테이블 객체와 if 인스턴스의 opState 설정
- (4) 신호 경보: 해당 신호에 대한 if 테이블 객체 인스턴스의 opState 설정

이와 같이 운용하게 되면 임의 유니트가 임의의 슬롯에 실장 되는 동적 관리가 이루어져 신호구성 정보의 무결성이 제공된다.

5. 결론

동기식 전송 기술에 속하는 MSPP 시스템의 SNMP 기반 관리 기능을 살펴보았다. IETF의 기본 구성과 SONET-MIB을 활용하고 장치 고유의 기능을 위해 필요한 많은 정보에 대해서는 private MIB으로 정의하여 사용하였다. 이는 구성, 장애, 성능 관리 기능 모두에서 도입되었는데 EMS와 연동한 결과 주요한 관리 기능들이 시스템 OAM 모듈과 연동하여 수행되었다. 간단한 테이블 형식으로 이루어진 감시 위주의 SNMP 관리 기반에 적극적인 전송 시스템 관리 기능을 구현하기 위해서는 비표준 객체의 정의가 많이 필요하고 관리 동작의 적용도 용이하지 않다. 그러나 리눅스 기반의 embedded agent 개발을 하기 위한 간단하고도 경제적인 환경이기도 하다.

본 논문에서는 이러한 간단한 관리 이념의 실현 환경에서 다양한 NG-SDH 관리 기능의 실현 사례와 통합 결과를 제시하였으며 운용에 있어 타 프로토콜 기반의 주요한 관리 기능이 모두 수용되어 SNMP 기반의 표준 NG-SDH 관리 모델로의 추진이 가능할 것으로 기대된다. 표준 망 관리 기능은 ITU-T에서 주도한 TMN 구조하에서 CMIP을 기반으로 하였다가 CORBA 수용 모델이 되었으며 나아가 SNMP 및 XML 기반 관리를 위해 프로토콜 중립 모델의 개발이 추진되고 있다. 또한 2003년도 말부터 TMF 표준이 본격적으로 ITU-T에서 수용되고 있는데 이와 같은 표준 동향을 고려할 때 KT의 NeOSS-TN(Network) 통합은 EMS를 통해서 TMF 표준 인터페이스를 이용하는 것이 바람직한 연동으로 예상된다.

References

- [1] IETF, RFC2558, Definitions of Managed Objects for the SONET/SDH Interface Type, 1999
- [2] IETF, RFC2737, Entity MIB (Version 2), 1999 .
- [3] IETF, RFC2863, The Interfaces Group MIB, 2000
- [4] ITU-T Recommendation M.3100, "Generic Network Information Model", 1992
- [5] ITU-T Recommendation G.774, "Synchronous Digital Hierarchy(SDH) Management Information Model for the Network Element View", 1992
- [6] ITU-T Recommendation G.774.01, "Synchronous Digital Hierarchy(SDH) Performance Monitoring for the Network Element View", 1994
- [7] ITU-T Recommendation G.774.03, "Synchronous Digital Hierarchy(SDH) Management of Multiplex Section Protection for the Network Element View", 1994