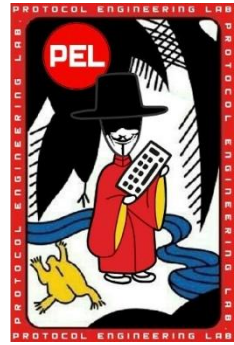




2026/10/16, 2026 보안 기초 세미나



TCP/IP 프로토콜

-24장 SNMP-

민 윤 홍(yunhong@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 개요
- SMI (Simple Management Information)
- MIB (Management Information Base)
- SNMP (Simple Network Management Protocol)

목 차

- 개요
- SMI (Simple Management Information)
- MIB (Management Information Base)
- SNMP (Simple Network Management Protocol)

개요

- 등장 배경

- 1980년대 후반 인터넷이 확장되면서 여러 제조사의 라우터, 호스트 등 여러 기종의 장비가 한 네트워크에 공존함

- 문제점

- 장비마다 전용 관리 도구와 방식이 달라서, 상태 확인이나 설정 변경을 장비별로 따로 관리해야함
- 한 장비에 장애가 발생하면 네트워크 전체 상태를 한 곳에서 파악하기 어려움

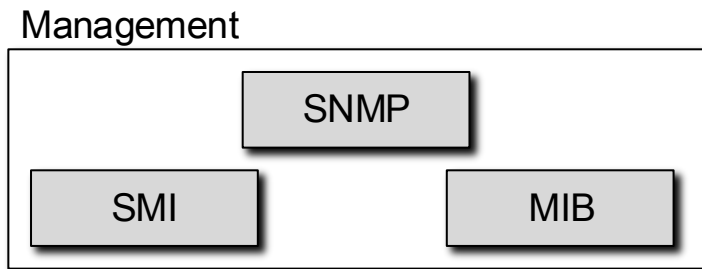
- 요구 사항

- 제조사와 무관한 공통된 관리 방식이 필요함
- 장비의 상태를 통일된 형식으로 표현하는 방법이 필요함
- 원격에서 조회, 설정 변경, 장애 통보를 할 수 있는 수단이 필요함

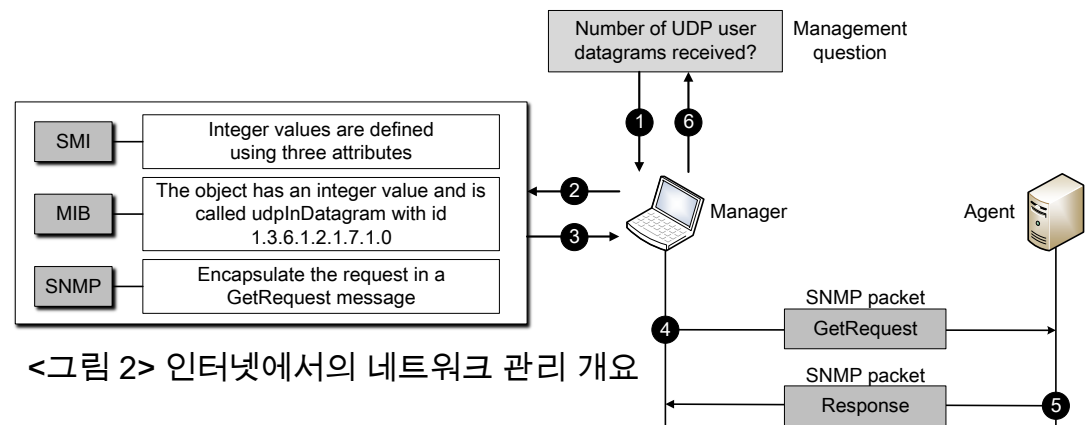
개요

• 해결 방안 (1/3)

- SNMP (Simple Network Management Protocol)
 - 관리자와 에이전트 사이에 교환되는 패킷의 형식을 정의함
- SMI (Structure of Management Information)
 - 객체의 이름 체계를 부여하고 유형을 정의함
 - 객체와 값들을 부호화하는 방법을 나타내기 위한 규칙들을 정의함
- MIB (Management Information Base)
 - 관리될 집합에서 이름이 부여된 객체와 유형, 서로에 대한 관계 등을 정의함



<그림 1> 인터넷에서의 네트워크 관리 구성요소



<그림 2> 인터넷에서의 네트워크 관리 개요

개요

- 해결 방안 (2/3)

- SNMP (Simple Network Management Protocol)

- 정의

- 관리자와 에이전트 사이에 교환되는 패킷의 형식을 정의하는 프로토콜

- 특징

- 제조회사가 다른 장치들을 감시하기 위해 만들어짐
 - 관리 작업을 수행하기 위해 SMI와 MIB를 사용함
 - 응용 계층에서 작동함

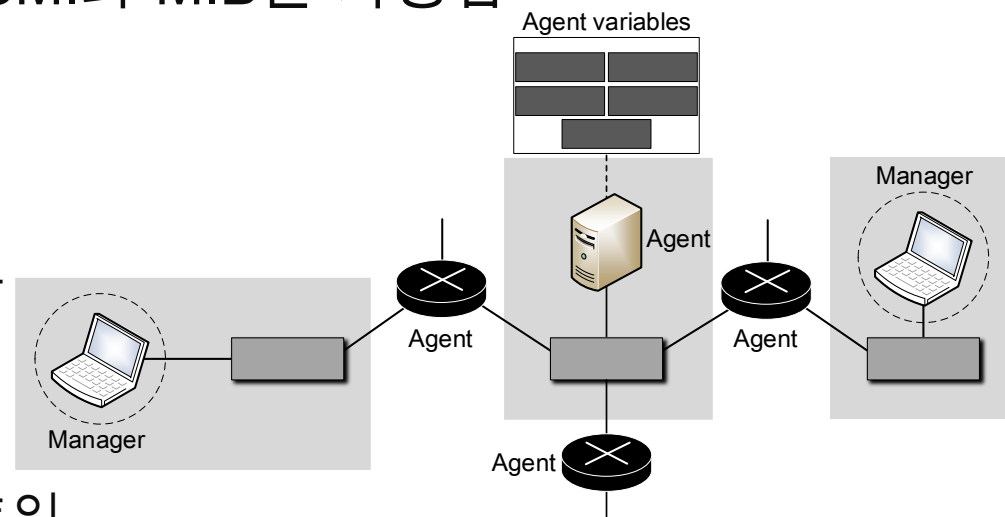
- 구성 요소

- 관리자(manager)

- SNMP 클라이언트 프로그램을 수행하는 호스트임

- 에이전트(agent)

- SNMP 서버 프로그램을 수행하는 라우터이거나 호스트임



<그림 3> SNMP 개념

개요

- 해결 방안 (3/3)
 - SMI (Structure of Management Information)
 - 정의
 - 객체의 이름 체계, 데이터 유형, 인코딩 방법을 정의하는 프로토콜
 - 특징
 - 데이터를 주고받는 송,수신자의 데이터 타입을 정의함
 - 객체의 값들을 부호화하는 방법을 나타냄
 - MIB (Management Information Base)
 - 정의
 - 관리 대상 장치가 관리 객체들의 이름, 유형, 접근 권한 등을 정의한 집합
 - 특징
 - SMI가 정의한 유형에 따라 객체에 이름을 붙임

개요

- 작동 예시

- SMI

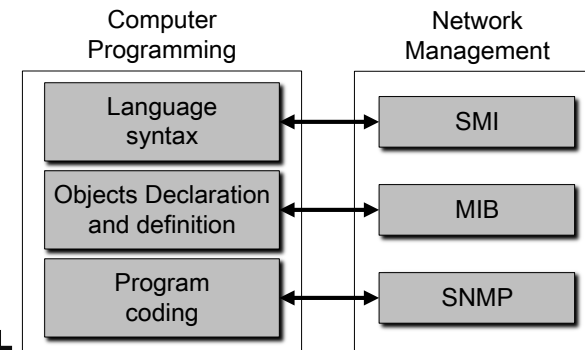
- 코딩을 할 때 언어의 형식을 정하는 것과 같음
 - e.g., C, Java, Python 등
- 변수의 유형과 이름을 정함
 - e.g., int, char 등

- MIB

- SMI에서 정한 형식을 따라 변수를 선언함
 - e.g., int a, char b 등

- SNMP

- SMI와 MIB에서 문법과 변수를 사용하여 프로그래밍을 하는 것과 같음



<그림 4> 컴퓨터 프로그래밍과 네트워크 관리의 비교

목 차

- 개요
- SMI (Simple Management Information)
- MIB (Management Information Base)
- SNMP (Simple Network Management Protocol)

SMI

- 개요

- 서로 다른 제조사의 장비를 관리하기 위해 등장함

- 특징

- 규칙을 규정하지만 실제로 정의하지는 않음

- 규정 요소

- 이름
- 데이터 유형
- 인코딩 방식

SMI

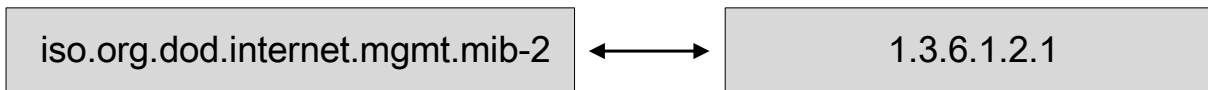
- 이름

- 개요

- 각 관리 대상 객체에게 고유한 이름을 가지도록 요구함

- 특징

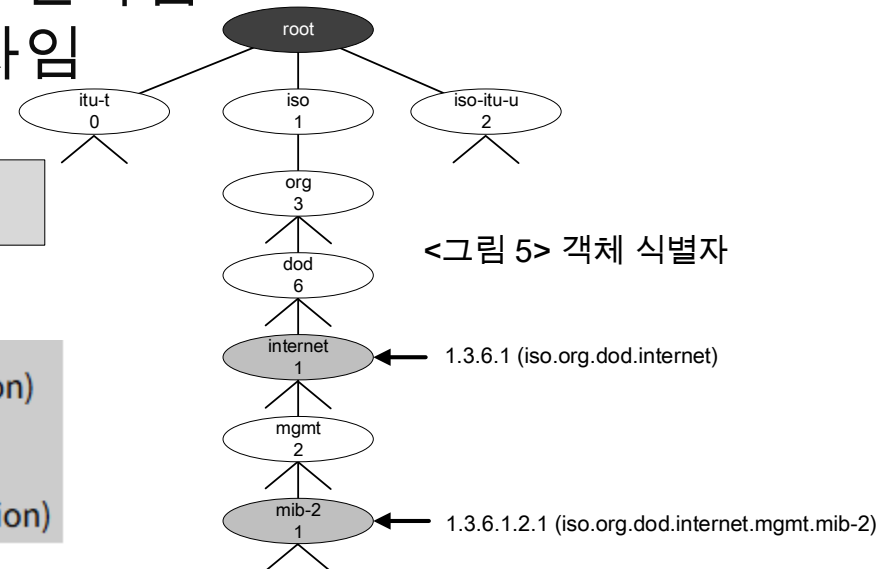
- 객체 식별자(OID, Object Identifier)를 사용하여 구현함
 - 객체를 고유하게 식별하는 십진수 문자임
 - 트리 구조에 기반한 계층적 식별자임



<그림 6> 객체 식별자 예시

	{iso(1) identified-organization(3) dod(6) internet(1) mgmt(2) mib-2(1)}	(ASN.1 notation)
OID:	1.3.6.1.2.1	(dot notation)
	/ISO/Identified-Organization/6/1/2/1	(OID-IRI notation)

<그림 7> 실제 인터넷에서의 OID 조회



<그림 5> 객체 식별자

SMI

- 데이터 유형

- 개요

- 관리 대상 객체가 가질 수 있는 값의 유형을 규정함

- 특징

- ASN.1 (Abstract Syntax Notation One) 규약을 사용함
 - 데이터의 구조를 기계나 프로그래밍 언어와 무관하게 기술하는 언어

- 종류

- 단순 유형
 - 구조적 유형

SMI

- 데이터 유형

- 단순 유형

- 개요

- ASN.1 표준에서 가져온 기본 데이터 형식

- 종류

[표 1] 데이터 유형 종류

Type	Size	Description
INTEGER	4 bytes	-2^{31} 와 $2^{31} - 1$ 사이의 일반 정수, 포트가 살아있는지 확인할 때 사용
Integer32	4 bytes	32비트 이하로 범위를 가지는 일반 정수, 인터페이스가 보낼 수 있는 최대 크기를 알아볼 때 사용
Unsigned32	4 bytes	0 과 $2^{32} - 1$ 사이의 양의 정수, 음수가 될 수 없는 수에 사용
OCTET STRING	Variable	최대 65,535바이트 길이의 바이트열, 장비 이름 및 MAC 주소를 확인할 때 사용
OBJECT IDENTIFIER	Variable	객체 식별자(값 자체가 OID인 경우), 장비의 제조사와 모델을 확인할 때 사용
IPAddress	4 bytes	정수 4개로 이루어진 IP 주소, 장비에 설정된 IP 주소를 확인할 때 사용
Counter32	4 bytes	0에서 $2^{32}-1$ 까지 증가하는 정수, 최댓값에 도달하면 다시 0으로 돌아감, 지금까지 받은 데이터 양을 확인할 때 사용
Counter64	8 bytes	0에서 2^{64} 까지 증가하는 정수, 최댓값에 도달하면 다시 0으로 돌아감
Gauge32	4 bytes	0에서 2^{32} 까지 증가하는 정수, 최댓값에 도달하면 최댓값에 머무름, 포트 속도를 확인할 때 사용
TimeTicks	4 bytes	시간을 1/100초 단위로 기록하는 값, 장비가 켜지고 난 후 경과시간을 확인할 때 사용
BITS	Variable	비트들의 열, 여러 개의 상태를 한 값에 담을 때 사용 (10101011101...)
Opaque	Variable	해석하지 않은 문자열. 다른 데이터를 포장할 때 사용, 지금은 사용하지 않음

SMI

- 데이터 유형

- 구조적 유형

- 개요

- 여러 데이터 유형을 조합하여 복합 구조를 표현하는 데이터 형식

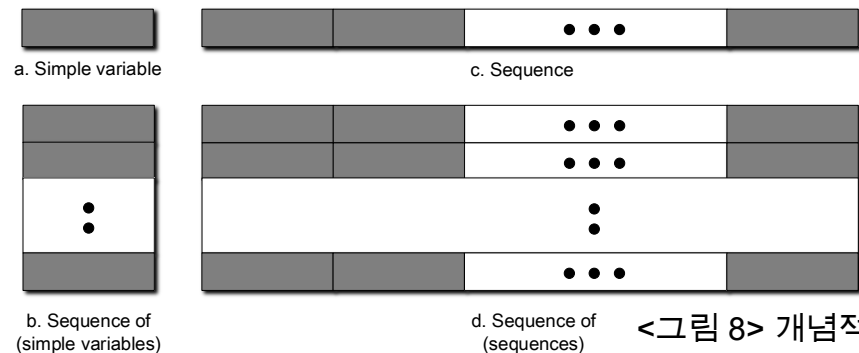
- 종류

- Sequence

- 데이터 유형이 다른 데이터들의 결합임
 - C언어에서 사용하는 구조체의 개념과 비슷함

- Sequence of

- 데이터 유형이 모두 같은 데이터들이나 Sequence의 결합임
 - C언어에서 사용하는 배열의 개념과 비슷함



<그림 8> 개념적 데이터 유형

SMI

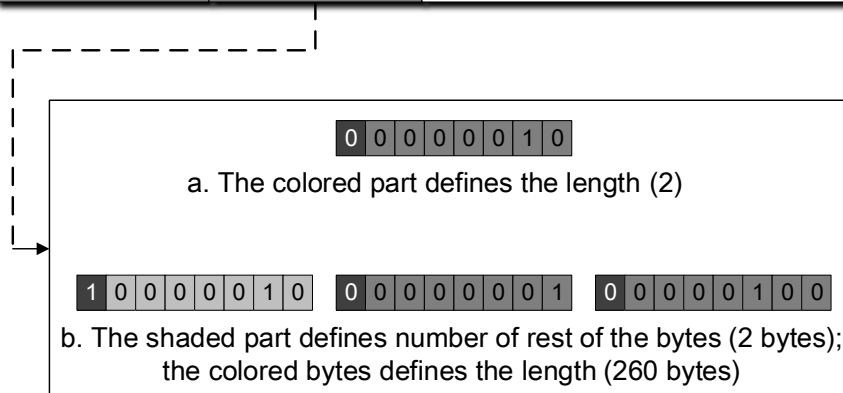
• 인코딩 방식

• 개요

- 기종이 다른 기계들 간 비트 수준의 호환성있는 통신을 위해 BER (Basic Encoding Rules) 표준을 이용함

• 특징 (1/2)

- 데이터의 각 조각을 태그, 길이, 값으로 된 형식으로 인코딩함



Length field
<그림 9> 인코딩 형식

[표 2] 데이터 유형 코드

Data Type	Tag (Binary)	Tag (Hex)
INTEGER	00000010	02
OCTEC STRING	00000100	06
OBJECT IDENTIFIER	00000110	06
NULL	00000101	05
Sequence, sequence of	00110000	30
IPAddress	01000000	40
Counter	01000001	41
Gauge	01000010	42
TimeTicks	01000011	43
Opaque	01000100	44

SMI

• 인코딩 방식

• 특징 (2/2)

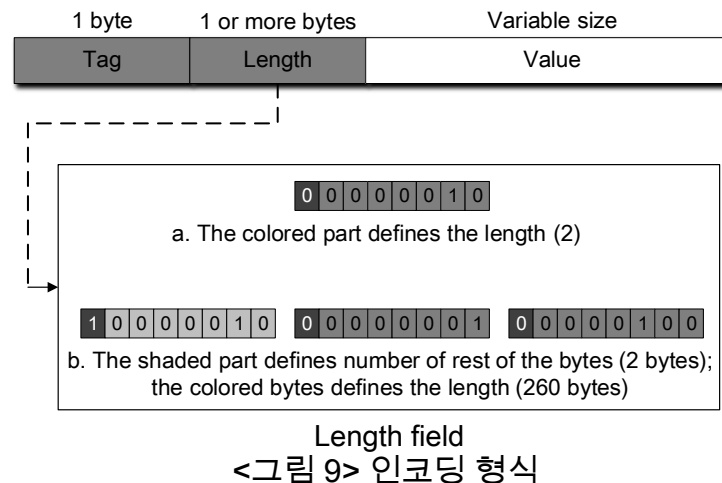
• 데이터의 각 조각을 태그, 길이, 값으로 된 형식으로 인코딩함

• 태그

- 데이터 유형을 정하는 1바이트 필드로 2진수와 16진수로 이루어짐

• 길이

- 데이터의 길이를 나타내는 1바이트 이상의 가변길이필드임
- 길이가 1바이트인 경우 최상위 비트는 0이 되고 나머지 7비트로 길이를 규정함
- 길이가 1바이트 초과면 최상위 비트는 1이 되고 안내 바이트를 표기함



- 메시지 압축

- 예제 24.1

길이가 500인 메시지의 길이 필드에 들어갈 값을 구하여라.

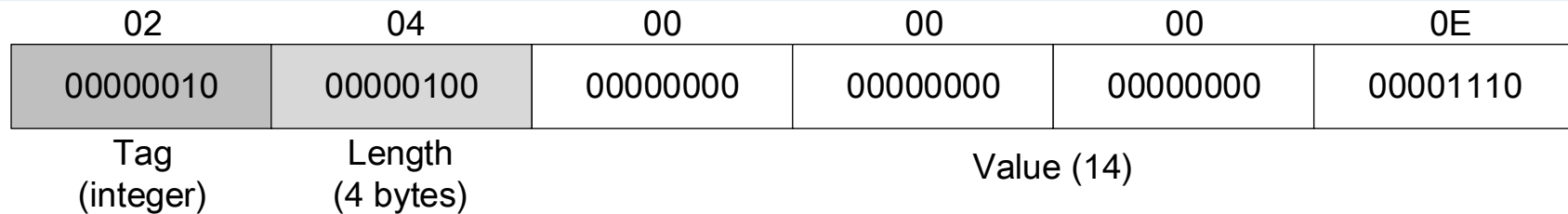
- 500을 길이 필드에 기입해야 하는 경우 길이가 255를 넘어서 2바이트가 필요함
- 500을 16진수로 표기하면 01F4이고, 이를 2바이트 01 F4로 작성함
 - $01 \times 256 = 256$
 - $F4 = 244$
 - $256 + 244 = 500$
- 안내 바이트는 1000 0010임
 - 앞에 1은 길이가 1바이트를 초과함을 나타냄
 - 뒤에 1은 길이가 2바이트임을 알림
- 10000010을 16진수로 표현하면 82임
- 최종 길이 필드는 82 01 F4로 3바이트가 됨

SMI

- 메시지 압축

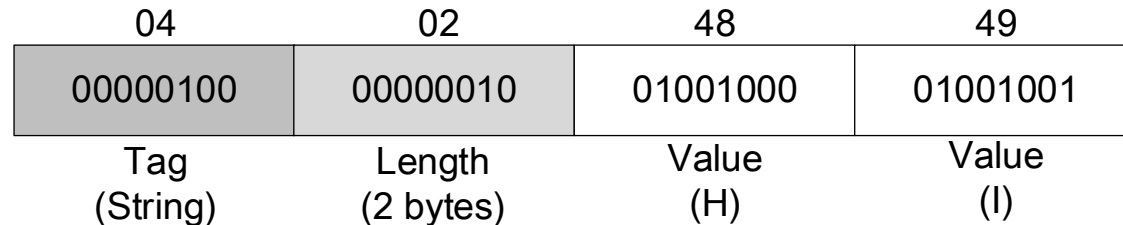
- 예제 24.2

SMI를 이용해 INTERGER 14를 정의하는 방식은 다음과 같음



- 예제 24.3

SMI를 이용해 OCTET STRING인 'HI' 를 정의하는 방식은 다음과 같음



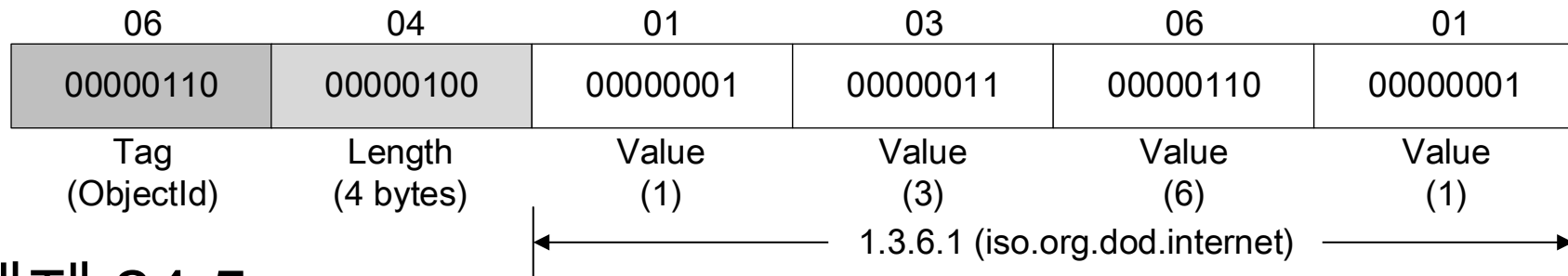
- 문자는 ASCII 코드를 사용해 표기함 $H = 73 = 48_{(16)}$, $I = 73 = 49_{(16)}$

SMI

- 메시지 압축

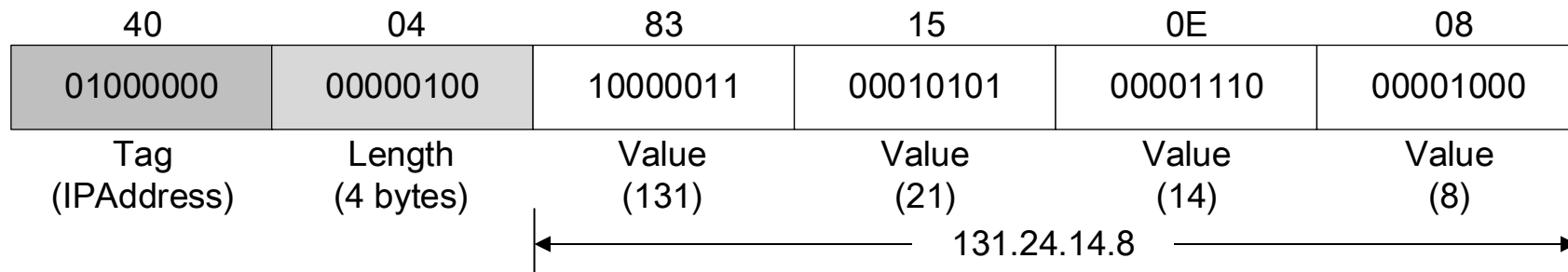
- 예제 24.4

SMI를 이용해 객체 식별자 1.3.6.1 (iso.org.dod.internet)을 정의하는 방법은 다음과 같음



- 예제 24.5

SMI를 이용해 IP 주소 131.21.14.8을 정의하는 방법은 다음과 같음



목 차

- 개요
- SMI (Simple Management Information)
- MIB (Management Information Base)
- SNMP (Simple Network Management Protocol)

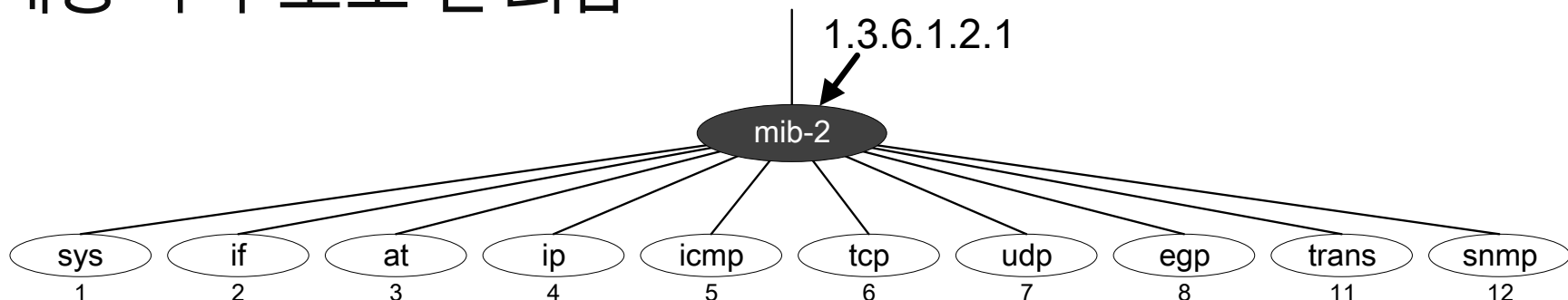
MIB

- 개요

- 관리 대상 장치가 관리 객체들의 이름, 유형, 접근 권한 등을 정의한 집합

- 특징

- 각 에이전트는 관리자가 관리할 수 있는 모든 객체를 모아놓은 자신의 MIB를 가짐
 - 객체들은 10개의 그룹으로 분류됨
- 계층적 구조로 관리함



<그림 10> mib-2

MIB

- 그룹 (1/2)
 - sys (system)
 - 노드의 이름, 위치, 수명 등 일반적인 정보를 규정함
 - if (interface)
 - 노드의 인터페이스 번호, 물리 주소, IP 주소 등 모든 인터페이스에 관한 정보를 규정함
 - at (address translation)
 - ARP 테이블에 관한 정보를 규정함
 - 네트워크에서 IP 주소와 물리 주소를 1대1로 매칭하여 저장해 둔 목록임
 - icmp
 - 송, 수신된 패킷 수와 생성된 전체 오류 등 ICMP에 관한 정보를 규정함

MIB

- 그룹 (2/2)

- tcp

- 연결 테이블, 타임-아웃 값, 포트 수, 송, 수신된 패킷 수 등 TCP에 관한 일반적인 정보를 규정함

- udp

- 포트 수, 송, 수신된 패킷 수 등 UDP에 관한 일반적인 정보를 규정함

- egp

- EGP (Exterior Gateway Protocol)에 관한 정보를 규정함
 - BGP (Border Gateway Protocol)의 전신으로 현재는 사용하지 않음

- trans

- 네트워크의 송, 수신을 하는 모듈에 관한 정보를 규정함

- snmp

- SNMP 자체에 관련된 일반적인 정보를 규정함

MIB

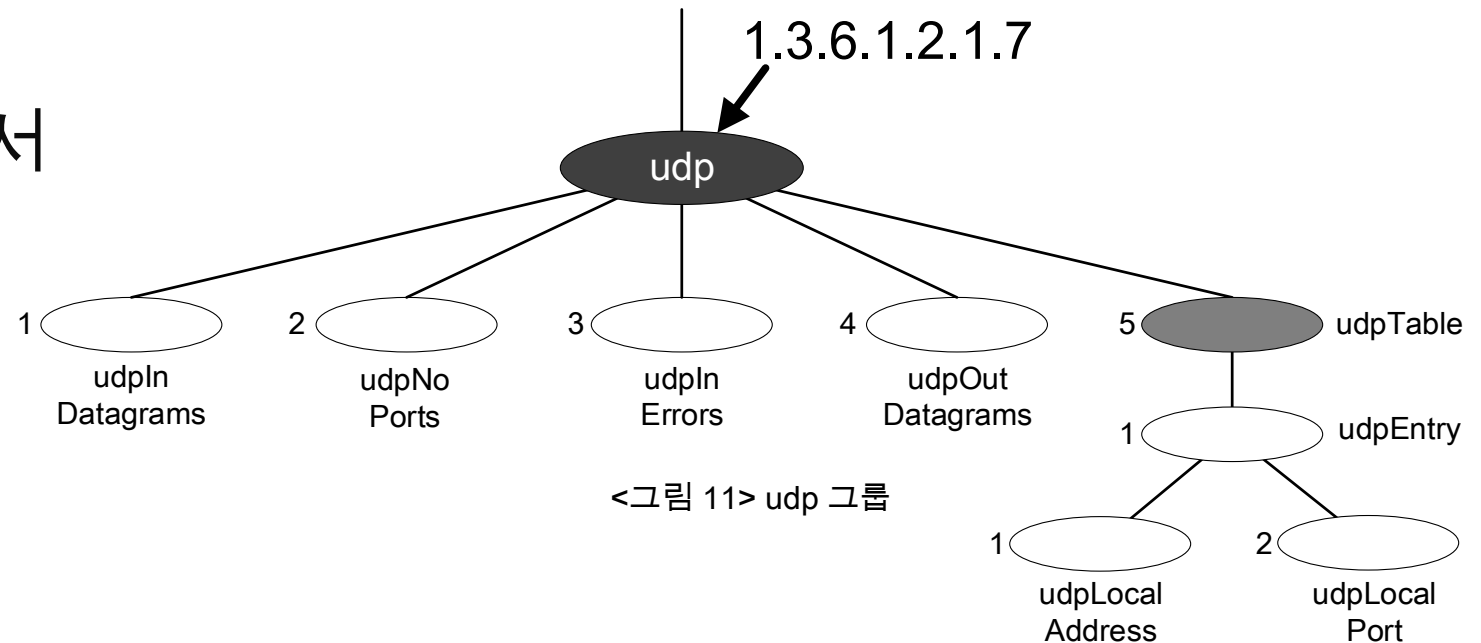
- 변수 액세스

- 개요

- SNMP 매니저가 MIB에 있는 특정 값을 가리켜 읽어 오는 방법

- 종류

- 단순 변수
 - 테이블
 - 사전적 순서



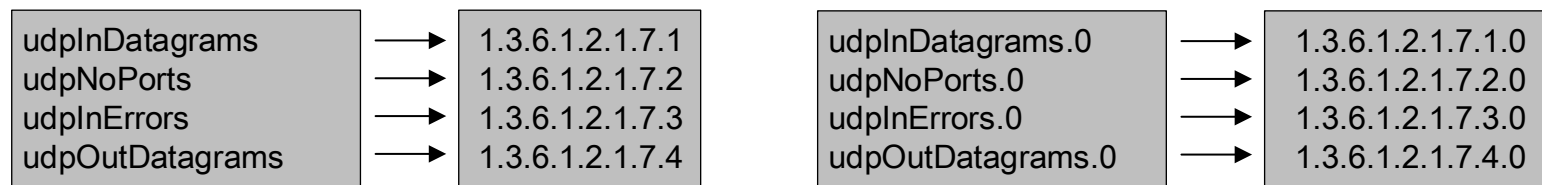
MIB

- 변수 액세스

- 단순 변수

- 개요

- 객체 식별자들의 뒤에 0을 붙여 실제 값을 보는 방법



<그림 12> 객체 식별자(좌)와 단순 변수(우)

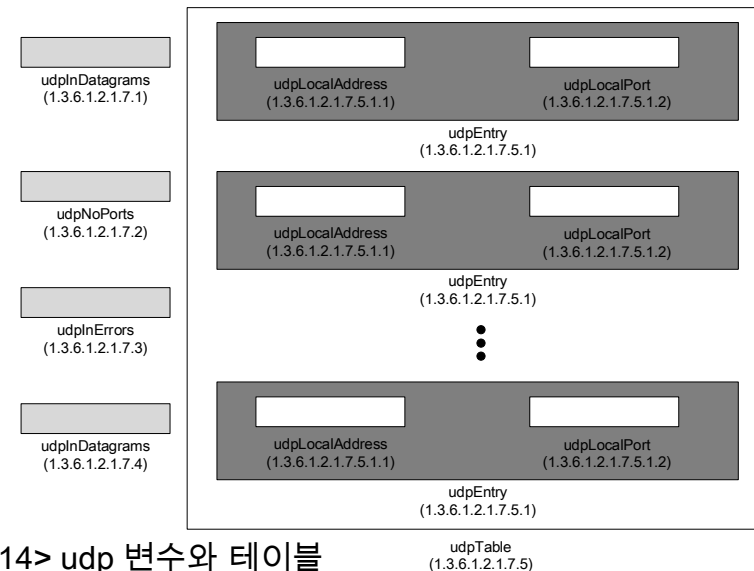
- 테이블 (1/2)

- 개요

- 여러 행으로 이루어진 테이블 형태의 MIB 객체에서 열의 OID 뒤에 행을 식별하는 인덱스를 붙여 접근하는 방법



<그림 13> udp테이블의 OID



<그림 14> udp 변수와 테이블

MIB

- 변수 액세스

- 테이블 (2/2)

- 개요

- 여러 행으로 이루어진 테이블 형태의 MIB 객체에서 열의 OID 뒤에 행을 식별하는 인덱스를 붙여 접근하는 방법

- 특징

- Sequence of 와 Sequence 는 값을 갖지 않으므로 직접 접근할 수 없어 값을 가진 말단(leaf) 노드인 열 객체에만 접근할 수 있음
 - 접근할 행의 식별자는 열 OID에 인덱스를 붙여 만듦
 - 인덱스로 사용하는 필드는 테이블마다 다름

181.23.45.14	23
1.3.6.1.2.1.7.5.1.1.181.23.45.14.23	1.3.6.1.2.1.7.5.1.2.181.23.45.14.23
192.13.5.10	161
1.3.6.1.2.1.7.5.1.1.192.13.5.10.161	1.3.6.1.2.1.7.5.1.1.192.13.5.10.161
227.2.45.18	180
1.3.6.1.2.1.7.5.1.1.227.2.45.18.180	1.3.6.1.2.1.7.5.1.1.227.2.45.18.180
230.20.5.24	212
1.3.6.1.2.1.7.5.1.1.230.20.5.24.212	1.3.6.1.2.1.7.5.1.1.230.20.5.24.212

(좌) <그림 15> udpTable의 Index

udpLocalAddress.181.23.45.14.23 → 1.3.6.1.2.1.7.5.1.1.181.23.45.14.23

<그림 16> udpTable의 첫 번째 행 로컬 주소

MIB

- 변수 액세스

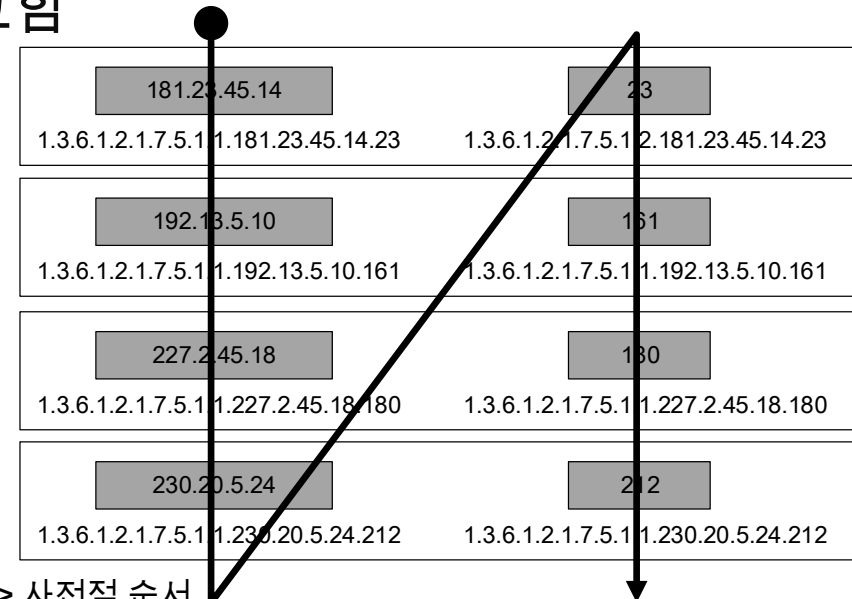
- 사전적 순서

- 개요

- MIB의 모든 객체 인스턴스를 OID 기준으로 사전순으로 나열한 규칙

- 특징

- OID를 앞 자리부터 한 자리씩 비교하여 앞 자리가 같으면 다음 자리를 비교함
 - 각 문자는 문자가 아닌 정수로 비교함
 - e.g., 1.3.6.1.9가 1.,3.6.1.10보다 앞임
 - 부모 노드가 자식 노드보다 항상 앞에 옴
 - 테이블은 열 우선 순서로 정렬됨
 - 첫 번째 열의 모든 행을 정렬 후 두 번째 열로 넘어감



<그림 17> 사전적 순서

목 차

- 개요
- SMI (Simple Management Information)
- MIB (Management Information Base)
- SNMP (Simple Network Management Protocol)

SNMP

- 개요
 - 인터넷 네트워크 관리에 SMI와 MIB를 모두 사용하는 프로토콜
- 특징
 - 관리자가 에이전트에서 정의된 객체의 값을 읽고 저장함
 - 에이전트가 비정상적 상황에 대한 경고 메시지를 관리자에게 보냄

SNMP

- PDU (Protocol Data Unit)

- 개요

- SNMP 메시지 안에 들어가는 실제 요청 및 응답 단위

- 종류 (1/3)

- GetRequest

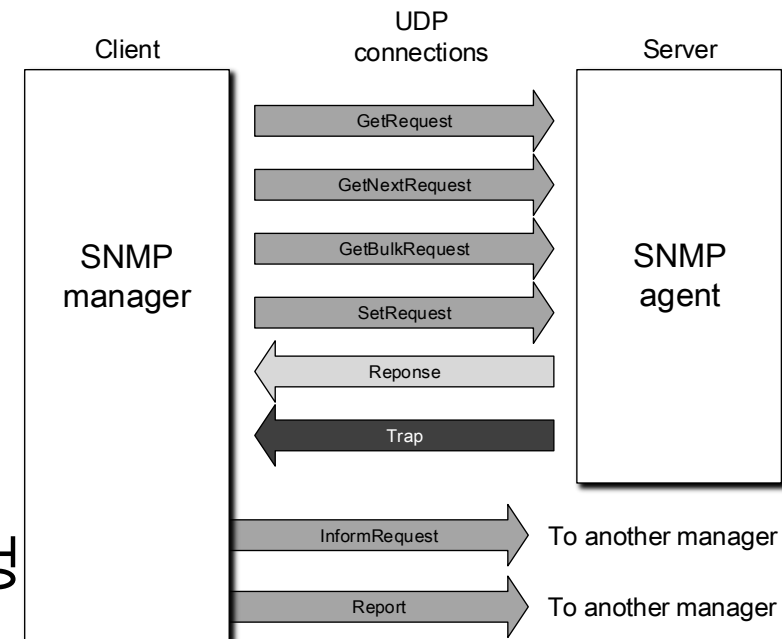
- 변수나 변수들의 값을 읽기 위해 관리자가 에이전트로 보내는 요청

- GetNextRequest

- 지정한 OID의 사전식 순서상 다음 변수의 OID와 값을 요청
- 반복 요청하여 테이블 전체를 확인할 수 있음

- GetBulkRequest

- 지정한 OID의 사전식 순서상 다음 변수들을 여러 개 가져오라는 요청



<그림 18> SNMP PDU

SNMP

- PDU (Protocol Data Unit)

- 종류 (2/3)

- SetRequest

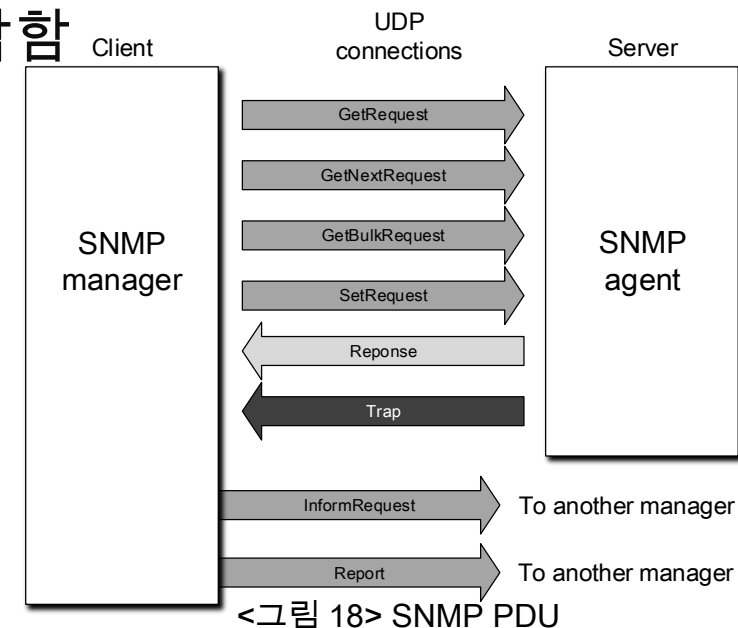
- 변수에 값을 저장하기 위해 관리자가 에이전트로 보내는 요청

- Response

- GetRequest나 GetNextRequest에 대한 응답으로 에이전트가 관리자에게 보내는 응답
 - 관리자에 의해 요청된 변수의 값을 포함함

- Trap

- 에이전트가 이벤트를 관리자에게 보고하는 응답



SNMP

- PDU (Protocol Data Unit)

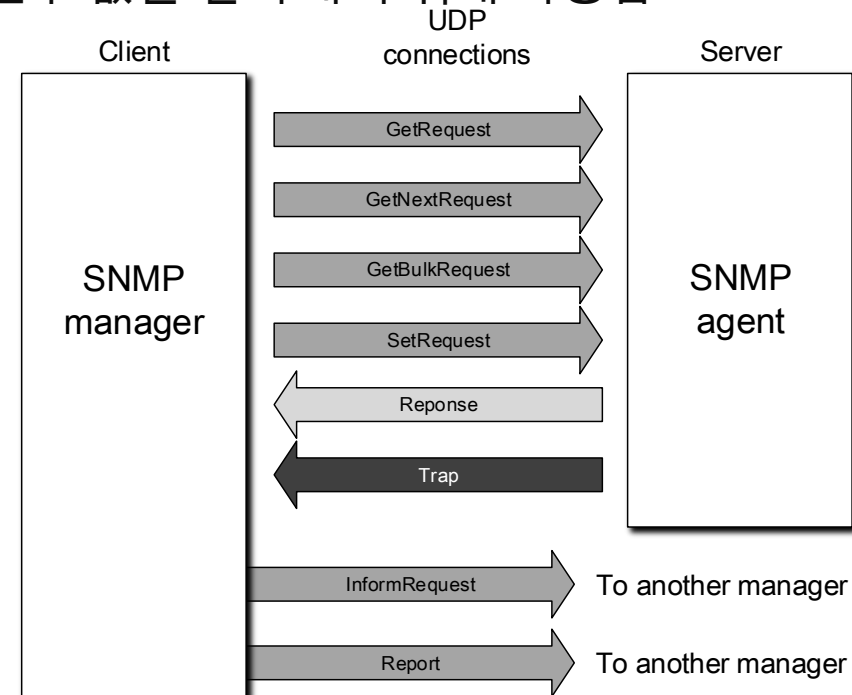
- 종류 (3/3)

- InformRequest

- 변수의 값을 알아내기 위해 관리자가 다른 원격 관리자에게 전송하는 요청
 - 원격 관리자가 제어하는 에이전트의 변수 값을 알아내기 위해 사용함

- Report

- 관리자들 사이에 오류 유형들을 보고하기 위해 설계된 응답



<그림 18> SNMP PDU

SNMP

• PDU (Protocol Data Unit)

• 형식 (1/3)

• PDU Type

- PDU 유형을 정의하는 필드

• Request ID

- 요청과 응답의 ID를 일치시키는데 사용되는 필드 [표 3] PDU 유형
- 관리자가 요청 PDU에서 사용함

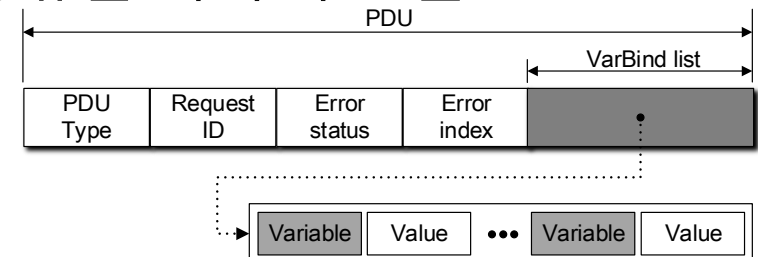
• Error status

- 에이전트에 의해 보고되는 오류의 종류를 나타내는 필드
- 요청 PDU에서는 0으로 채움

[표 4] 오류 종류

Status	Name	Meaning
0	noError	오류 없음
1	tooBig	응답이 최대 메시지 크기를 초과함
2	noSuchName	장비에 없는 OID를 요청함
3	badValue	저장하려는 값이 올바르지 않음
4	readOnly	값을 변경할 수 없음
5	genErr	그 외에 오류를 나타냄

Type	Tag (Binary)	Tag (Hex)
GetRequest	10100000	A0
GetNextRequest	10100001	A1
Response	10100010	A2
SetRequest	10100011	A3
GetBulkRequest	10100101	A5
InformRequest	10100110	A6
Trap(SNMPv2)	10100111	A7
Report	10101000	A8



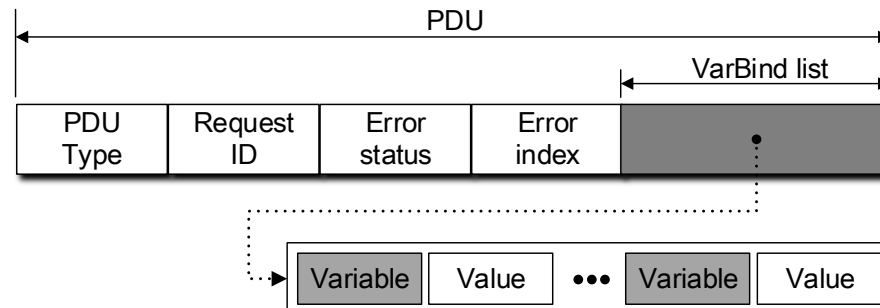
Differences:

1. Error status and error index values are zeros for all request messages except GetBulkRequest
2. Error status field is replaced by non-repeater field and error index field is replaced by max-repetitions field in GetBulkRequest

<그림 19> SNMP PDU 형식

SNMP

- PDU (Protocol Data Unit)
- 형식 (2/3)
 - Non-repeaters
 - 요청 PDU에서 비어 있는 Error status 필드를 대체하는 필드
 - GetBulkRequest에서만 사용됨
 - Error index
 - 관리자에게 오류를 일으킨 변수가 어느 것인지를 알려주는 필드



Differences:

1. Error status and error index values are zeros for all request messages except GetBulkRequest
2. Error status field is replaced by non-repeater field and error index field is replaced by max-repetitions field in GetBulkRequest

<그림 19> SNMP PDU 형식

SNMP

- PDU (Protocol Data Unit)

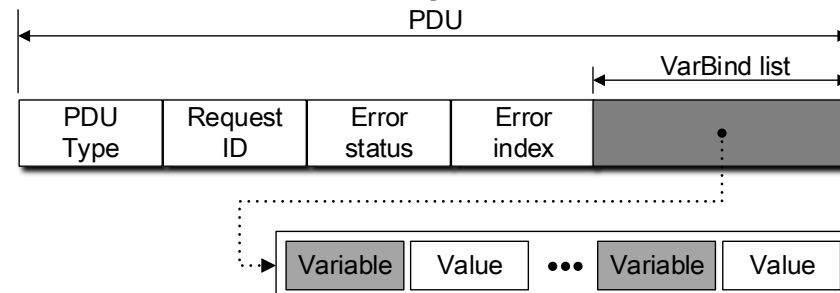
- 형식 (3/3)

- Max-repetition

- 요청 PDU에서 비어 있는 Error index 필드를 대체하는 필드
 - GetBulkRequest에서만 사용됨

- VarBind list

- PDU에서 다른 변수들의 OID와 그 값을 한 쌍으로 묶어 나열한 필드
 - GetRequest와 GetNextRequest는 0을 가짐



Differences:

1. Error status and error index values are zeros for all request messages except GetBulkRequest
2. Error status field is replaced by non-repeater field and error index field is replaced by max-repetitions field in GetBulkRequest

<그림 19> SNMP PDU 형식

SNMP

- 메시지

- 구성 요소

- Version

- SNMP 버전을 정의하는 INTEGER 데이터 유형 필드

- GlobalData

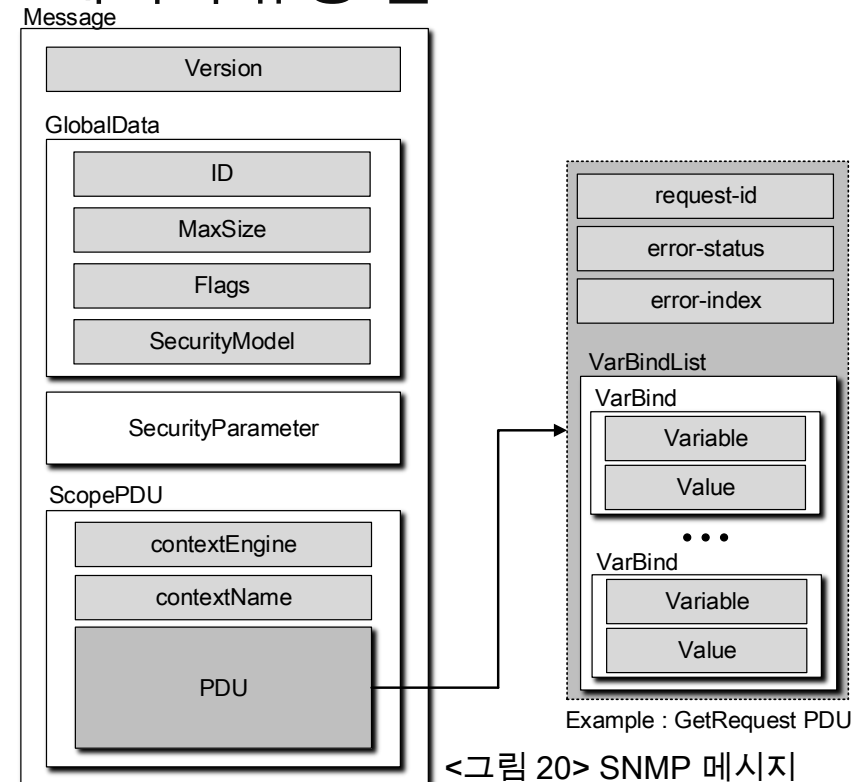
- 단순 데이터인 ID, MaxSize, Flags, Security-Model를 나열한 필드

- Security Parameter

- 메시지의 인증, 암호화, 재전송 공격 방지에 필요한 정보를 담는 필드

- ScopePDU

- 요청에 대한 범위와 PDU를 합친 필드
 - 요청을 처리할 SNMP의 ID와 MIB 집합 정보가 들어있음

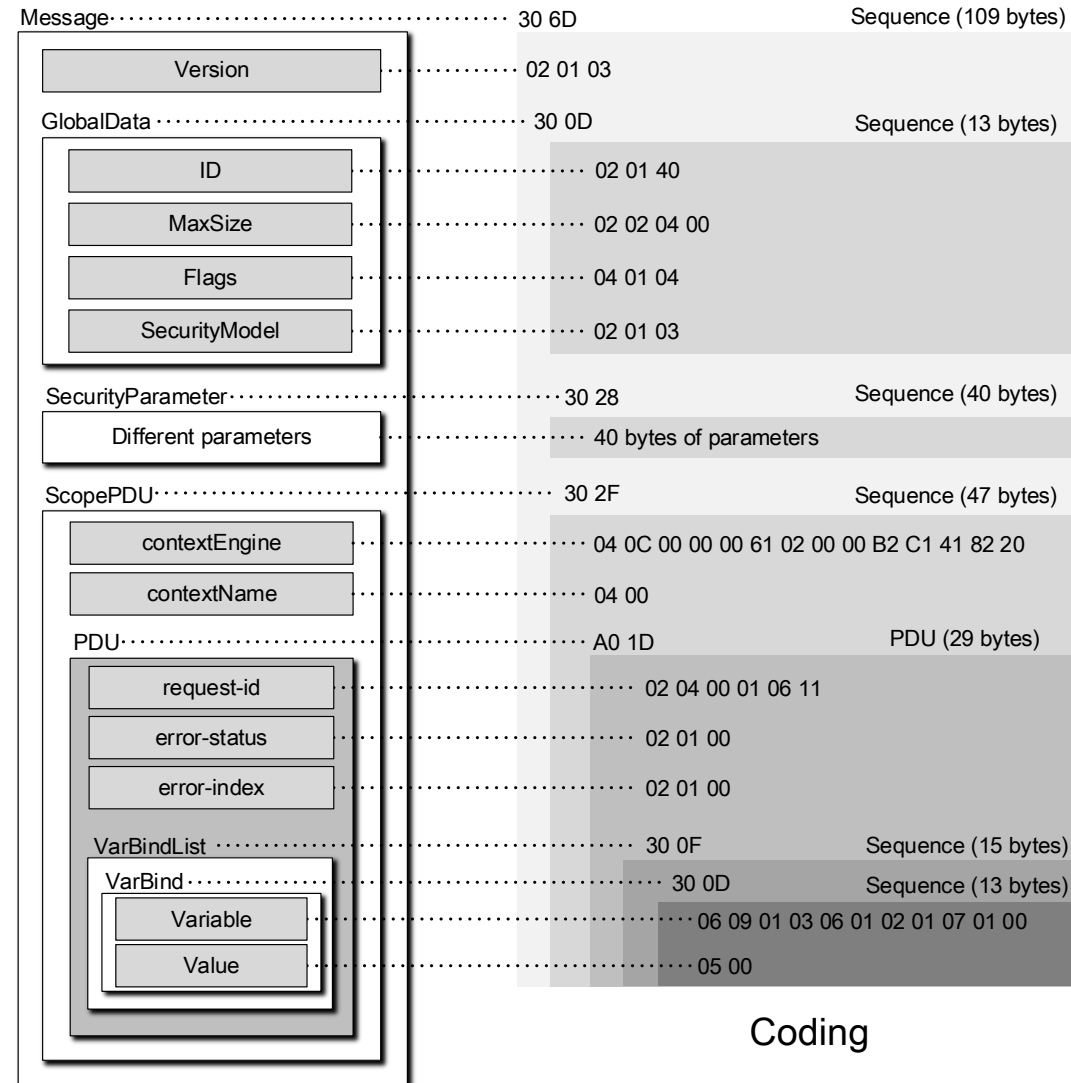


SNMP

• 메시지

• 예제 24.6

- 우측은 udpInDatagrams값을 알려달라고 보내는 GetRequest 메시지를 도식화 한 것임
- 인코딩 형식을 참고하면 두 번째 값이 길이임을 알 수 있음
- 각 캡슐화가 진행이 될 때 2 바이트씩을 더해야함
- VarBind의 길이를 구하면
 - Variable 값 = 9 bytes
 - Variable 태그 + 길이 + 값 = 11 bytes
 - Value 값 = 0 bytes
 - Value 태그 + 길이 + 값 = 2 bytes
- 위와 같이 모든 시퀀스를 계산하면 메시지의 값은 109, 총 길이는 111임
 - 자세한 계산 과정은 부록 #1 참조



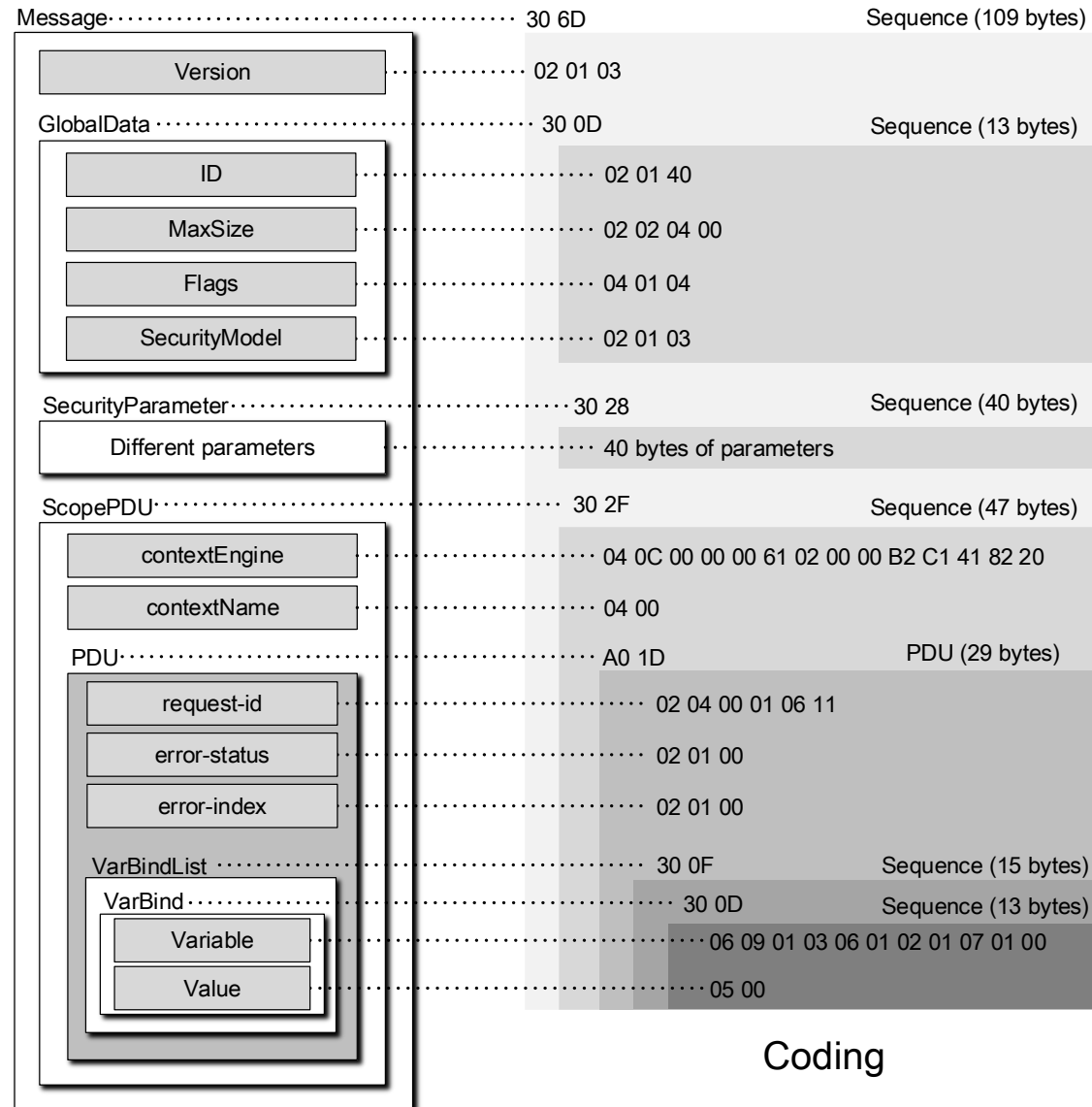
SNMP

• 메시지

• 예제 24.6

- 아래는 실제로 전송되는
GetRequest 메시지임

30	6D	02	01	03	30	0D	02	01	40	02	02	04	00	04	01
04	02	01	03	30	28	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	30	2F
04	0C	00	00	00	61	02	00	00	B2	C1	41	82	20	04	00
A0	1D	02	04	00	01	06	11	02	01	00	02	01	00	30	0F
30	0D	06	09	01	03	06	01	02	01	07	01	00	05	00	



SNMP

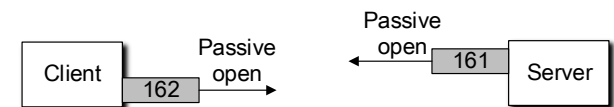
- UDP 포트

- 개요

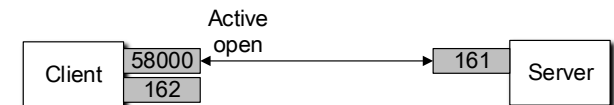
- SNMP는 UDP의 161번과 162번 포트를 사용함

- 특징

- 161번은 에이전트(서버)가 사용함
 - 162번은 관리자(클라이언트)가 사용함
 - 클라이언트와 서버 모두 잘 알려진 포트를 사용함
 - 클라이언트와 서버 둘 다 메시지를 만들어야 하기 때문에 무한히 수행되어야 함



a. Passive open by both client and server



b. Exchange of request and response messages



c. Server sends trap message

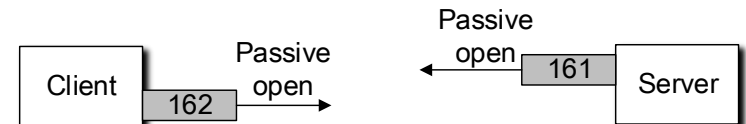
<그림 21> SNMP에 대한 포트 번호

SNMP

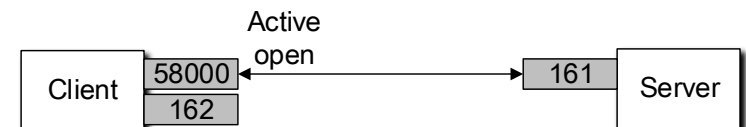
- UDP 포트

- 동작 과정 (1/2)

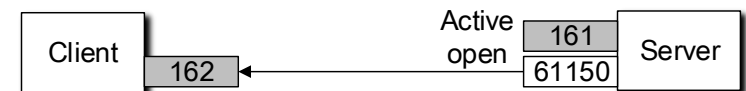
1. 에이전트는 포트 161로 Passive open을 시도하고 관리자로부터 연결을 기다림
2. 관리자는 임시 포트를 사용하여 Active open을 시도함
3. 요청 메시지는 관리자로부터 에이전트에게 보내짐
 - 발신지 포트로는 임시 포트를 사용하고 목적지 포트로는 161번을 사용함



a. Passive open by both client and server



b. Exchange of request and response messages



c. Server sends trap message

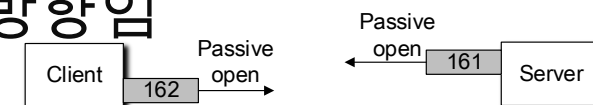
<그림 21> SNMP에 대한 포트 번호

SNMP

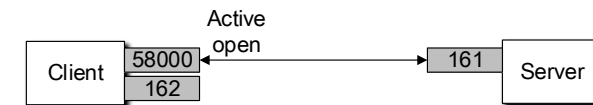
- UDP 포트

- 동작 과정 (2/2)

4. 응답 메시지는 에이전트로부터 관리자에게 보내짐
 - 발신지 포트로는 161번을 사용하고 목적지 포트로는 임시포트를 사용함
5. 관리자는 포트 162번으로 Passive open을 시도하고 에이전트로부터의 연결을 기다림
6. 에이전트는 Trap 메시지가 있을 때마다 임시 포트를 사용하여 Active open을 시도함
 - 해당 연결은 에이전트로부터 관리자로의 단방향임



a. Passive open by both client and server



b. Exchange of request and response messages

<그림 21> SNMP에 대한
포트 번호



c. Server sends trap message

SNMP

- 보안

- SNMPv1, v2

- 관리자와 에이전트가 미리 정한 문자열을 메시지에 넣고 일치하는지 확인함

- SNMPv3

- 사용자의 이름과 사용자별 키를 같은 키로 공유하고 암호화 및 복호화하여 사용함
- Security Parameters 필드에 정보를 담아서 전송함

Thanks!

민 윤 홍 (yunhong@pel.sejong.ac.kr)

부록 #1

• 예제 24.5 메시지 크기 계산 과정

