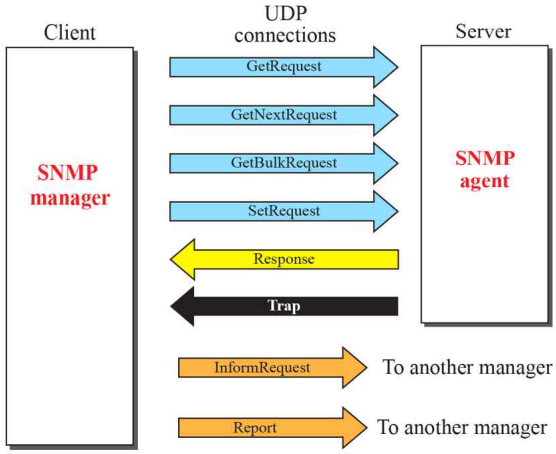
<인터넷 프로토콜>

작성자_2018320161_송대선
작성일_06_11

manager는 agent에 정의된 변수의 값을 알아낸다.
manager는 agent에 정의된 변수의 값을 변경한다.
agent는 이상 상황을 manager에게 알린다.

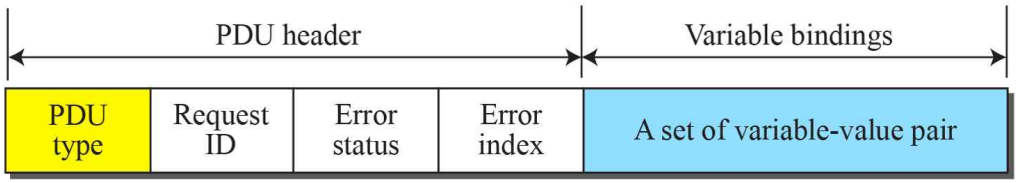


무조건 당하는 것이 Server다. agent가 당하는 중이라 server가 된다.

GetRequest: 하나의 변수정보를 알아온다.
GetNextRequest: 지정한 변수의 다음 변수정보를 알아온다.
GetBulkRequest: 지정한 변수의 다음 변수정보를 여러 개 알아온다.
SetRequest: 지정한 변수의 변수정보를 변경한다.

Response: Get~Request와 Set에 대하여 모두 response가 온다.
->SetRequest에 대한 Response는 SetRequest에 쓰여진 것을 반복한다.

Trap: agent내부에서 exception이 발생하면 바로 manager에게 알림



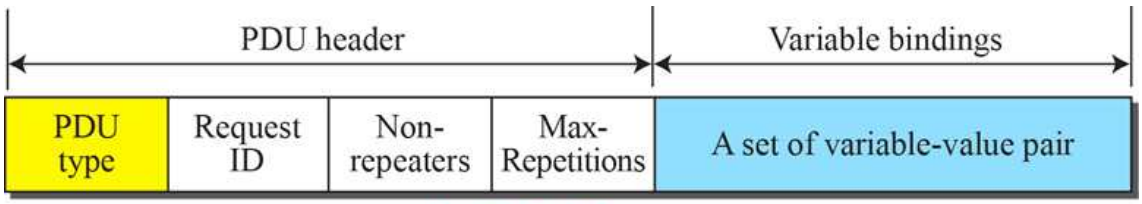
PDU type: GetRequest, GetNextRequest 등 정체가 뭐냐?

Type	*Tag (Hex)*	*Type*	*Tag (Hex)*
GetRequest	**A0**	GetBulkRequest	**A5**
GetNextRequest	**A1**	InformRequest	**A6**
Response	**A2**	Trap (SNMPv2)	**A7**
SetRequest	**A3**	Report	**A8**

Request ID: UDP를 사용하는 protocol은 전부 request와 response에 대한 매칭이 필요하다. -> 이 값으로 매칭을 시켜준다.
Error status: 어떤 종류의 error인가?
Error index: 몇 번째 변수에 error가 생겼나?
-> Request message들에서는 전부 0으로 가득차있다.

variable bindings:
변수명과 변수의 값이 같이 온다.
Get~Request message들은 변수값이 NULL이다.

-GetBulkRequest-

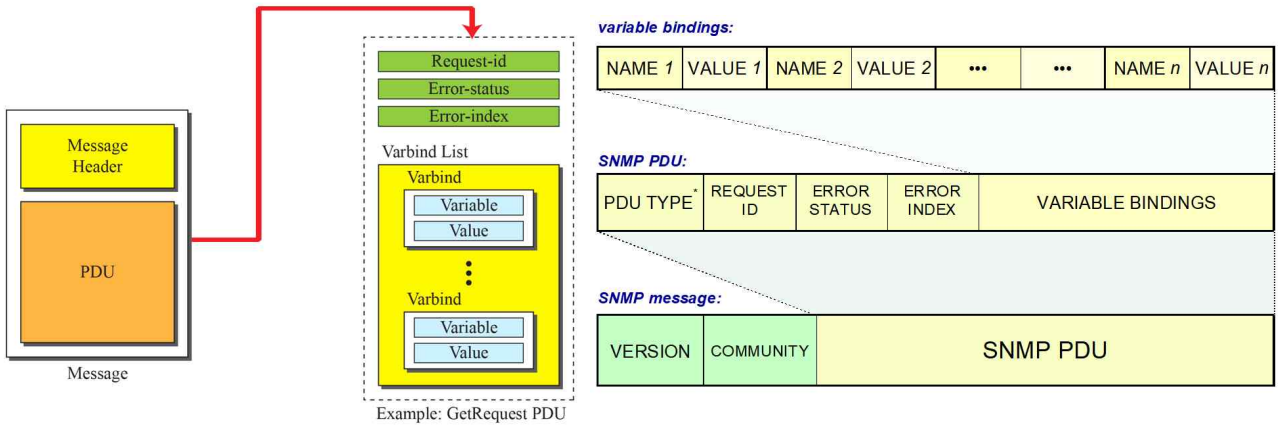


Non-repeaters와 Max-Repetitions는 어차피 error가 들어갈 자리라서 넣어도 상관이 없다. (Request message는 error에 쓰레기 값을 넣기 때문!)

Non-repeaters: repeat을 안할 변수의 개수. 시작변수가 몇 개인가?
Max-Repetitions: 시작변수로부터 몇 개를 더 읽을 것인가? (몇개를 더 GetNextRequest할 것인가?)

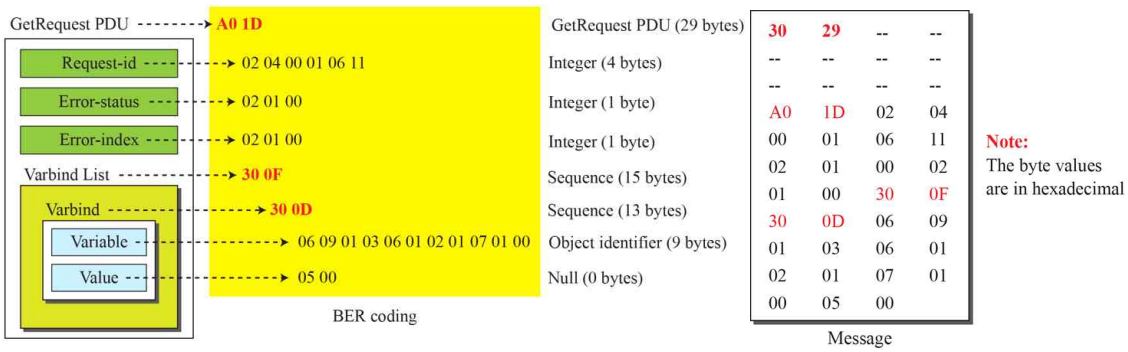
Status	*Name*	*Meaning*
0	noError	No error
1	tooBig	Response too big to fit in one message
2	noSuchName	Variable does not exist
3	badValue	The value to be stored is invalid
4	readOnly	The value cannot be modified
5	genErr	Other errors

Error table을 외울 필요는 없다.
too big: UDP를 사용하는데, packet이 512B를 넘었다.
badValue: 쓰려는 값 자체가 좀 이상하다.
readOnly: 읽기 기능만 있는 변수이다.
noSuchName: 그런 이름가진 변수는 없다.



Message Header:
Version: 1, 2, 3 중에서 SNMP가 버전 몇인가?
Community string: 일종의 password이다.

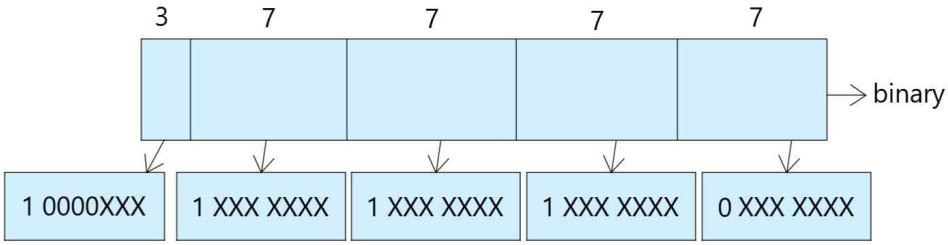
-BER-



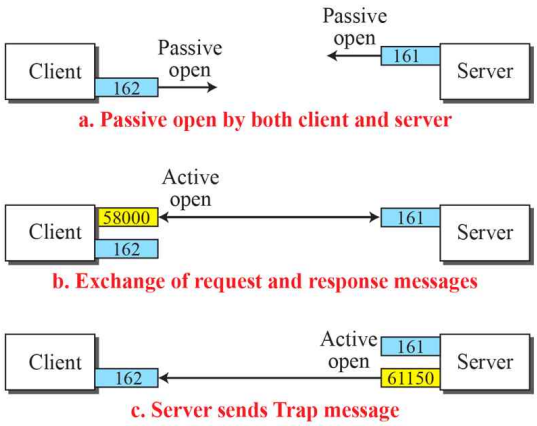
varbind List도 sequence of sequence 30이며,
varbind도 sequence 30이다.

Object identifier에서 1.3으로 시작하는 것은 2b로 표현한다.
-> 그래서 length가 하나 작아져서 나오게 된다.

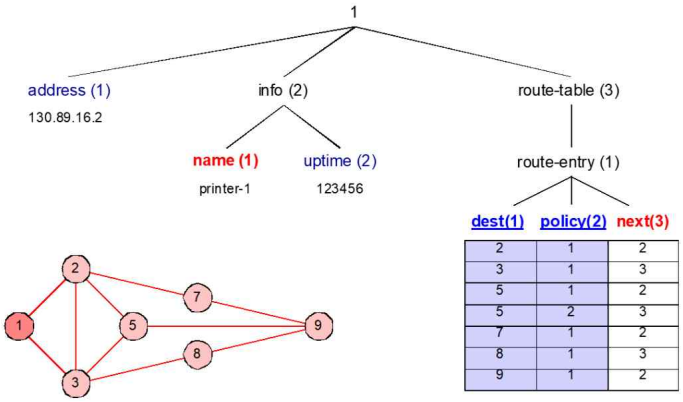
Object identifier에서 127을 초과하게 되면 코딩 자체가 바뀐다.
-> 7-bit ASCII Code로 표시하게 된다.



-port-



Client(manger)가 Server(agent)에게 요청을 보낼 때는 161으로 보내고,
Server(agent)가 Client(manger)에게 응답을 보낼 때는 162로 보낸다.



get(1.1.0) -> response(1.1.0 => 130.89.16.2)
table이 아닌 다른 instance들을 찾을 때는 ".0"을 붙여줘야 한다.

get(1.2) -> response(error-status = noSuchName)
set(1.2.1.0 => my-printer) -> response(noError; 1.2.1.0 => my-printer)

1.3.1.3.5.1
1.3.1: class 주소
3: 알고 싶은 column number
5.1: index column의 값

INSTANCE ID	INSTANCE VALUE
1.1.0	130.89.16.2
1.2.1.0	printer-1
1.2.2.0	123456
1.3.1.1.2.1	2
1.3.1.1.3.1	3
1.3.1.1.5.1	5
...	...
1.3.1.1.9.1	9
1.3.1.2.2.1	1
1.3.1.2.3.1	1
...	...
1.3.1.2.9.1	1
1.3.1.3.2.1	2
...	...

: next를 찾는 것은 사전 순서이다.

dest(1)	policy(2)	next(3)
2	1	2
3	1	3
5	1	2
5	2	3
7	1	2
8	1	3
9	1	2

1.3.1.2.2.1—Next->1.3.1.2.3.1
1.3.1.2.9.1.-Next->1.3.1.3.2.1

-tag field-

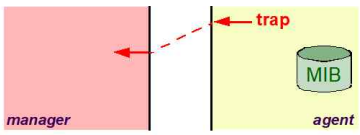
Application-wide tags

BIT PATTERN	APPLICATION TYPE
01 0 0 0000	IpAddress
01 0 0 0001	Counter32
01 0 0 0010	Gauge32
01 0 0 0010	Unsigned32
01 0 0 0011	TimeTicks
01 0 0 0100	Opaque
01 0 0 0110	Counter64

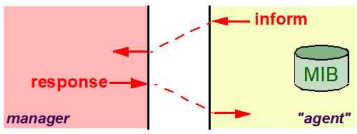
Universal tags

BIT PATTERN	ASN.1 TYPE
00 0 0 0010	INTEGER
00 0 0 0100	OCTET STRING
00 0 0 0110	OBJECT IDENTIFIER

: 빌려온 tag는 모두 대문자, 그리고 “0000”으로 시작한다.
직접 만든 tag는 첫 글자만 대문자, 그리고 “0100”으로 시작한다.
-> 별로 안 중요하다.



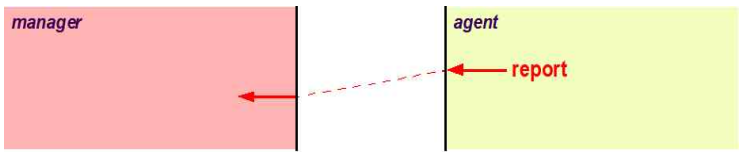
: trap은 manager에게 보내고 끝이다.



: inform는 manager에게 보내고 받았다고 response까지 온다.

getBulk(max-repetitions = 4; 1.1) ->
response(
1.1.0 => 130.89.16.2
1.2.1.0 => printer-1
1.2.2.0 => 123456
1.3.1.1.2.1 => 2)

getBulk(max-repetitions = 3; 1.3.1.1; 1.3.1.2; 1.3.1.3) ->
response(
1.3.1.1.2.1 => 2; 1.3.1.2.2.1 => 1; 1.3.1.3.2.1 => 2
1.3.1.1.3.1 => 3; 1.3.1.2.3.1 => 1; 1.3.1.3.3.1 => 3
1.3.1.1.5.1 => 5; 1.3.1.2.5.1 => 1; 1.3.1.3.5.1 => 2)



report는 SNMPv3부터 나왔다.