Chapter 21

Network Management: SNMP

Objectives

Upon completion you will be able to:

- *Understand the SNMP manager and the SNMP agent*
- *Understand the roles of SMI and MIB in network management*
- *Be familiar with SMI object attributes and encoding methods*
- *Know how an MIB variable is accessed*
- *Be familiar with the SNMP PDU and format*

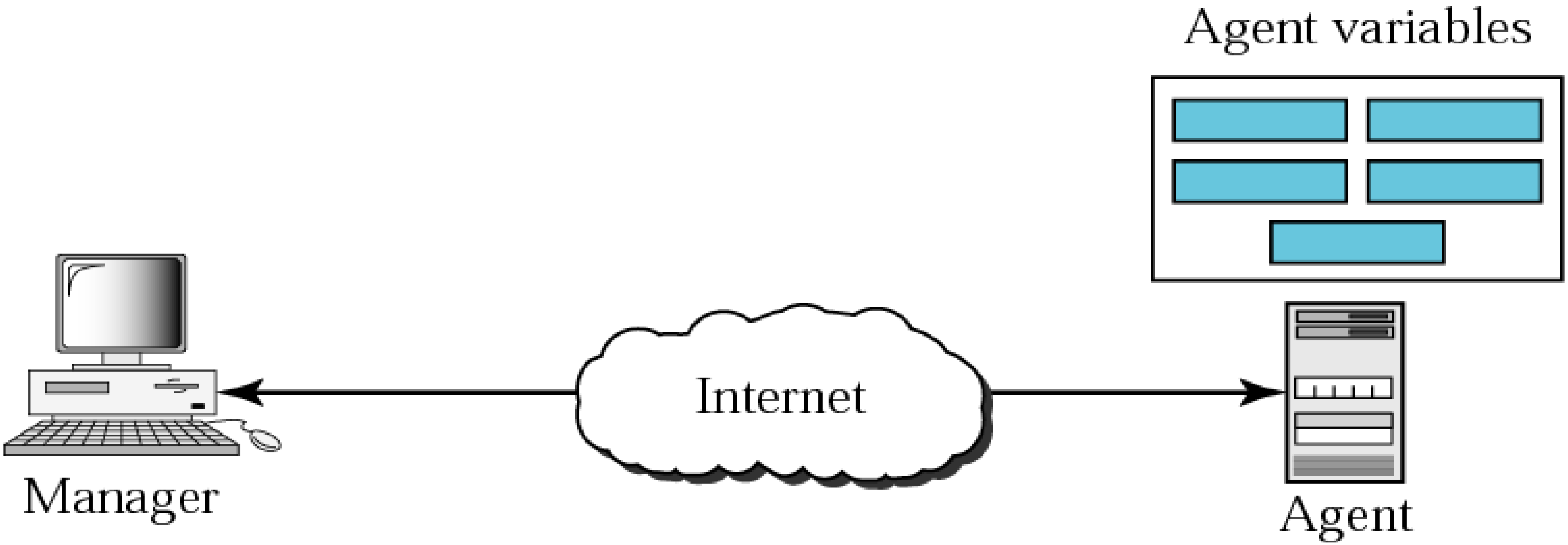
21.1 CONCEPT

SNMP defines a manager, usually a host, that controls and monitors a set of agents, usually routers.

The topics discussed in this section include:

Managers and Agents

Figure 21.1 *SNMP concept*



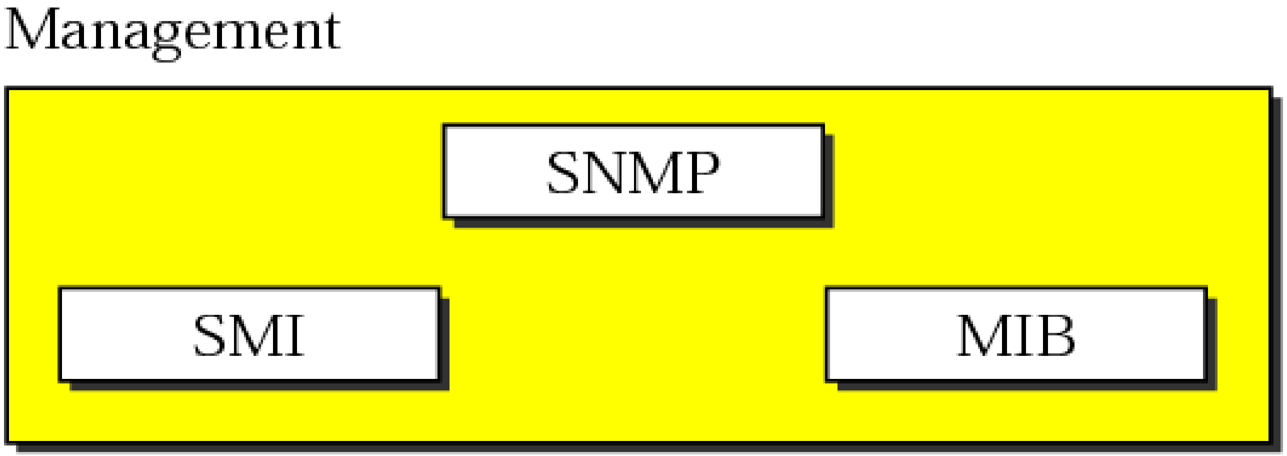
21.2 MANAGEMENT COMPONENTS

SNMP requires the use of two other protocols: Structure of Management Information (SMI) and Management Information Base (MIB). Network management on the Internet is done through the cooperation of SNMP, SMI, and MIB.

The topics discussed in this section include:

Role of SNMP
Role of SMI
Role of MIB
An Analogy
An Overview

Figure 21.2 *Components of network management on the Internet*



Note:

SNMP defines the format of packets exchanged between a manager and an agent. It reads and changes the status (values) of objects (variables) in SNMP packets.

Note:

SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values. SMI defines neither the number of objects an entity should manage, nor names the objects to be managed nor defines the association between the objects and their values.

Note:

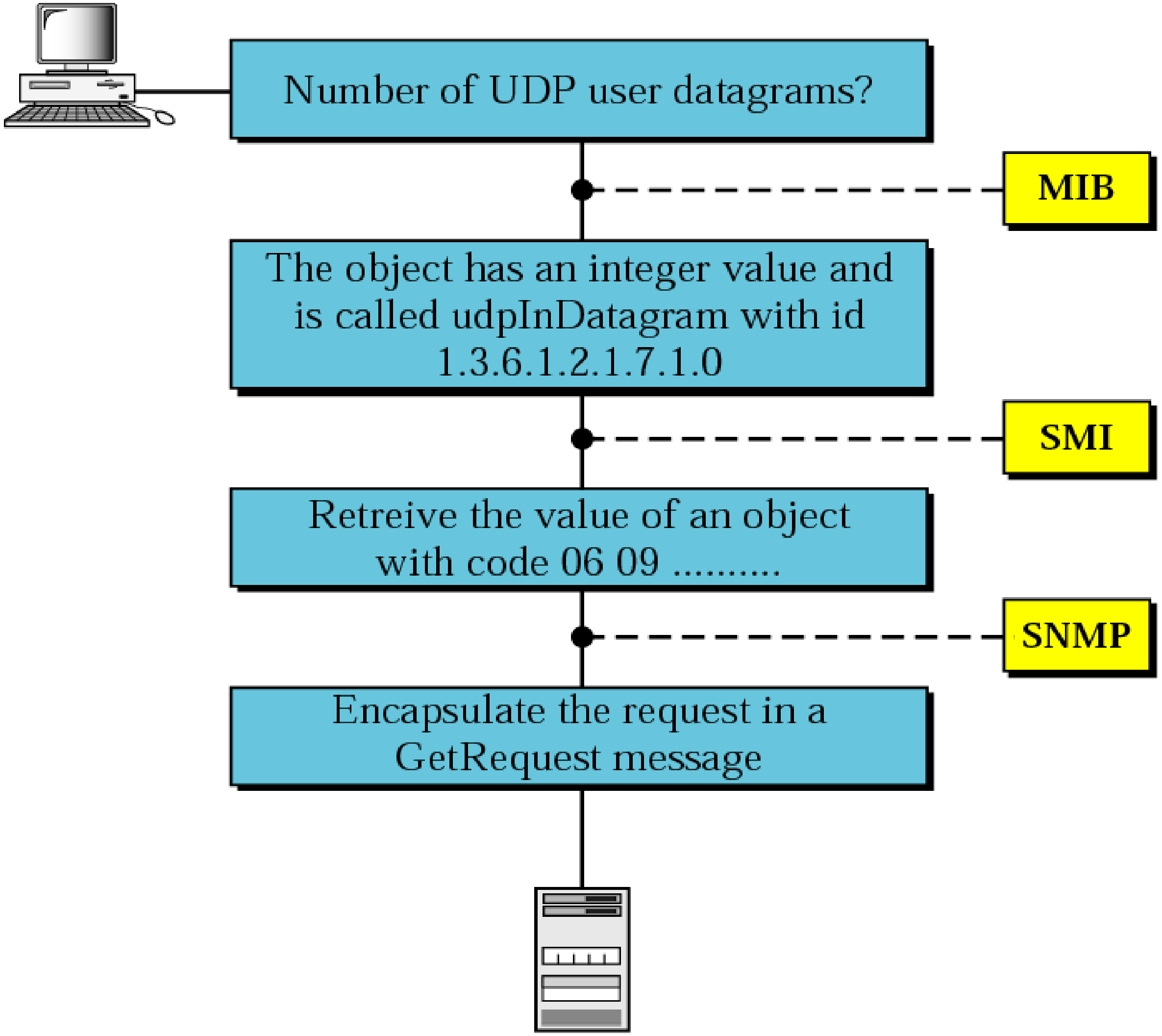
MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.

Note:

We can compare the task of network management to the task of writing a program.

- ***Both tasks need rules. In network management this is handled by SMI.***
- ***Both tasks need variable declarations. In network management this is handled by MIB.***
- ***Both tasks have actions performed by statements. In network management this is handled by SNMP.***

Figure 21.3 *Management overview*



Number of UDP user datagrams?

MIB

The object has an integer value and is called udpInDatagram with id 1.3.6.1.2.1.7.1.0

SMI

Retreive the value of an object with code 06 09 ....

SNMP

Encapsulate the request in a GetRequest message

21.3 SMI

SMI is a component used in network management. It names objects, defines the type of data that can be stored in an object, and shows how data can be encoded for transmission over the network

The topics discussed in this section include:

Name
Type
Encoding Method

Figure 21.4 *Object attributes*

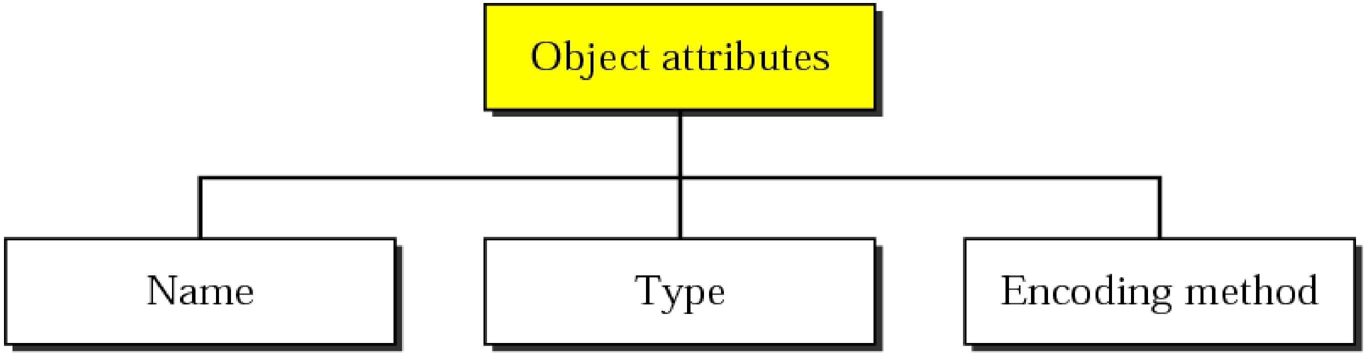
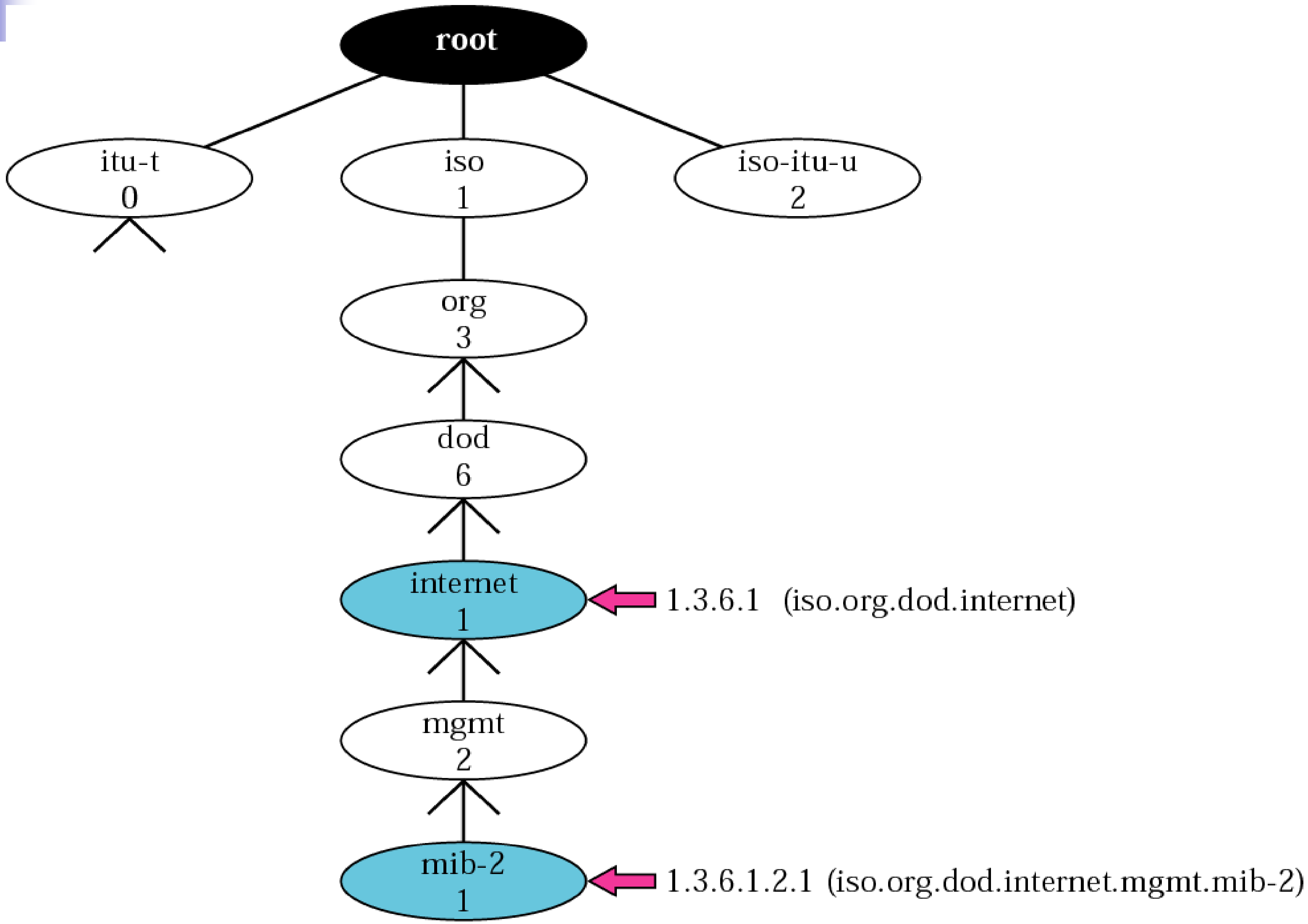


Figure 21.5 *Object identifier*


root
itu-t
0
iso
1
iso-itu-u
2
org
3
dod
6
internet
1
1.3.6.1 (iso.org.dod.internet)
mgmt
2
mib-2
1
1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2)

Note:

All objects managed by SNMP are given an object identifier.

The object identifier always starts with 1.3.6.1.2.1.

Figure 21.6 *Data type*

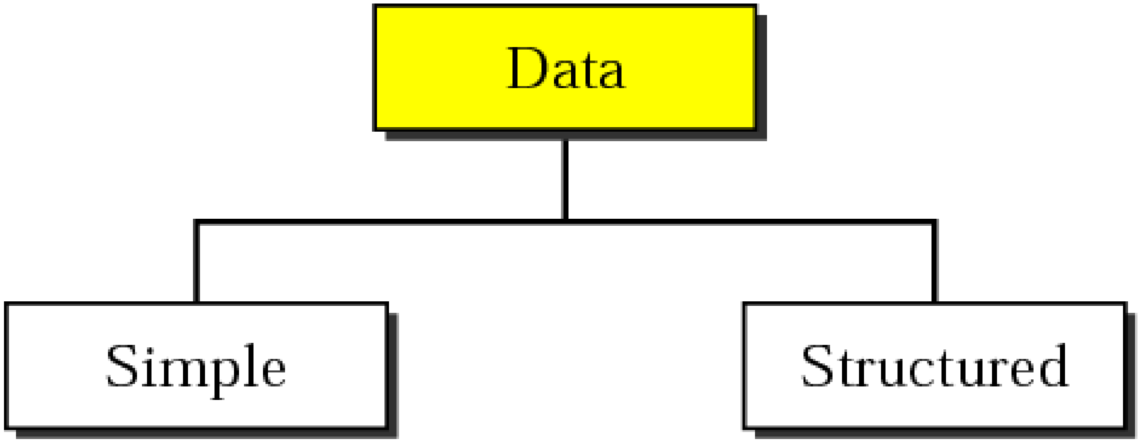
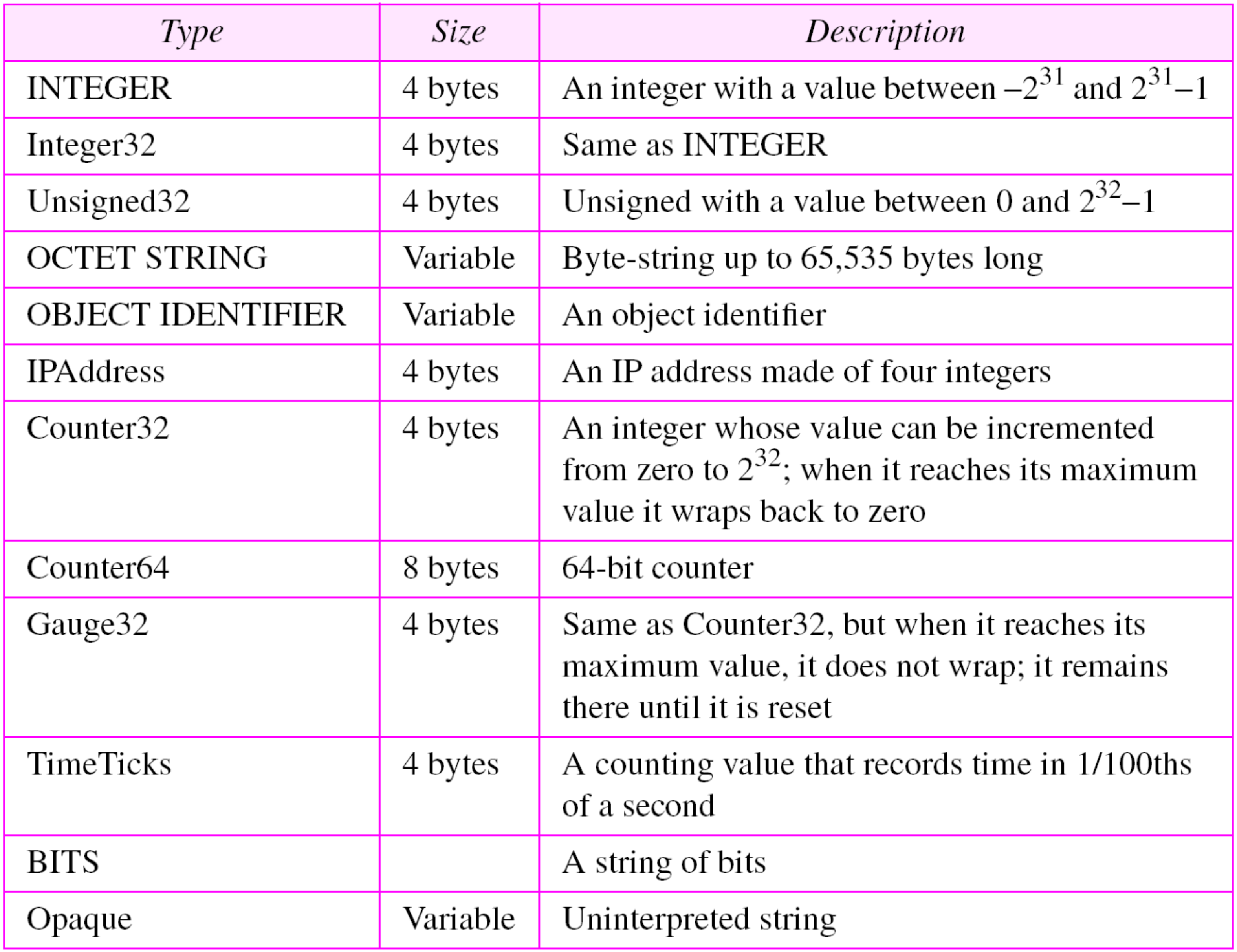


Table 21.1 Data types



Type	*Size*	*Description*
INTEGER	4 bytes	An integer with a value between -2^{31} and $2^{31}-1$
Integer32	4 bytes	Same as INTEGER
Unsigned32	4 bytes	Unsigned with a value between 0 and $2^{32}-1$
OCTET STRING	Variable	Byte-string up to 65,535 bytes long
OBJECT IDENTIFIER	Variable	An object identifier
IPAddress	4 bytes	An IP address made of four integers
Counter32	4 bytes	An integer whose value can be incremented from zero to 2^{32}; when it reaches its maximum value it wraps back to zero
Counter64	8 bytes	64-bit counter
Gauge32	4 bytes	Same as Counter32, but when it reaches its maximum value, it does not wrap; it remains there until it is reset
TimeTicks	4 bytes	A counting value that records time in 1/100ths of a second
BITS		A string of bits
Opaque	Variable	Uninterpreted string

Figure 21.7 *Conceptual data types*

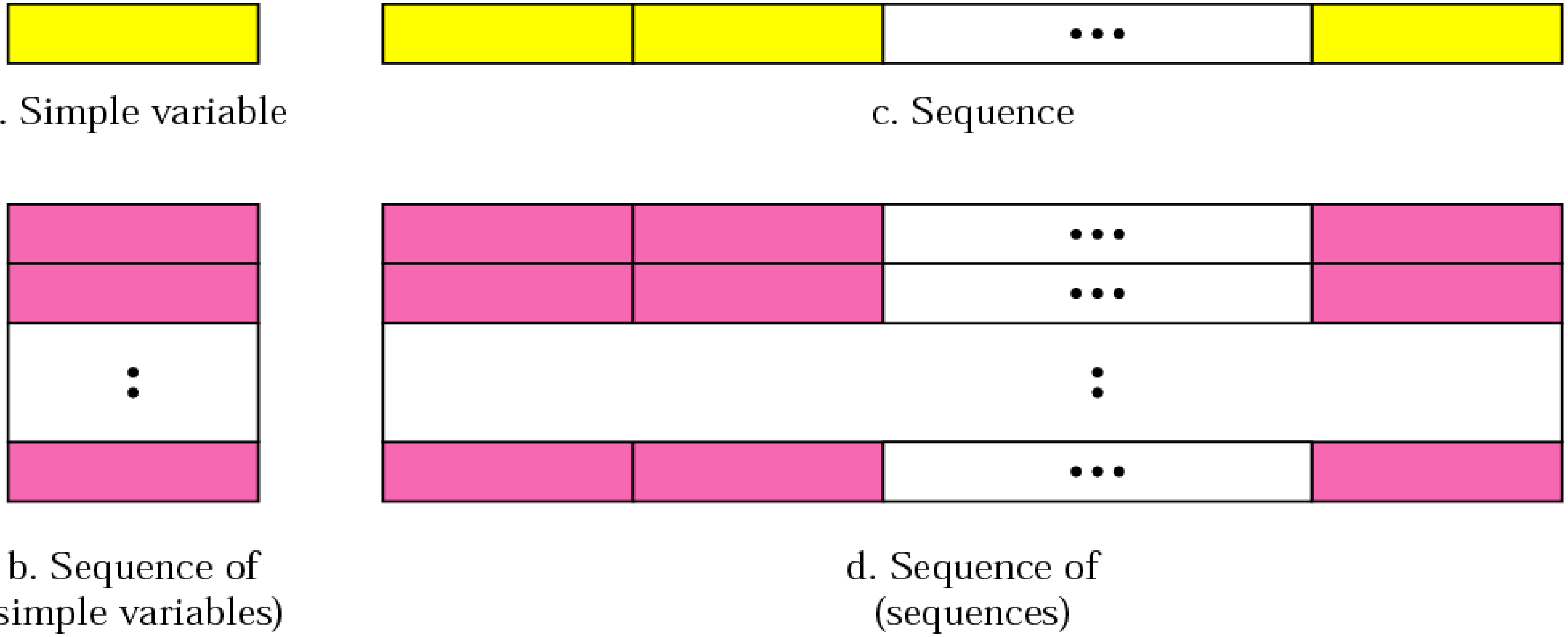


Figure 21.8 *Encoding format*

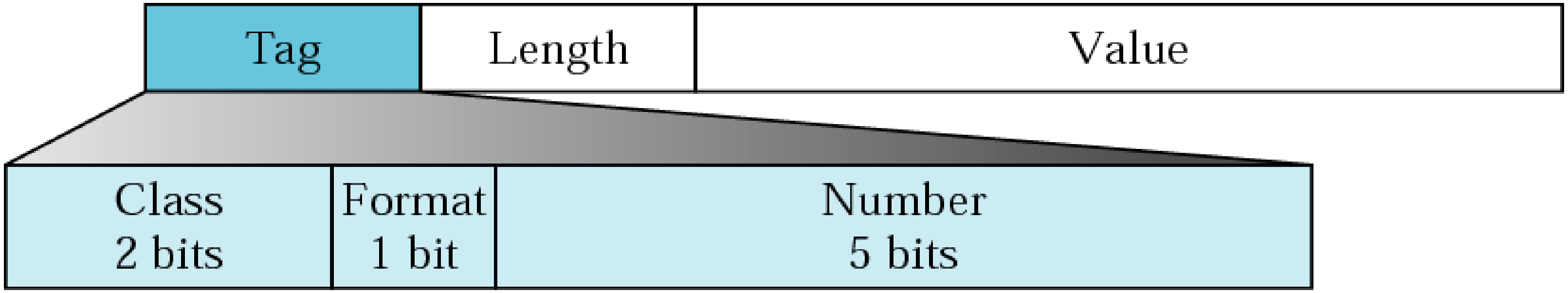
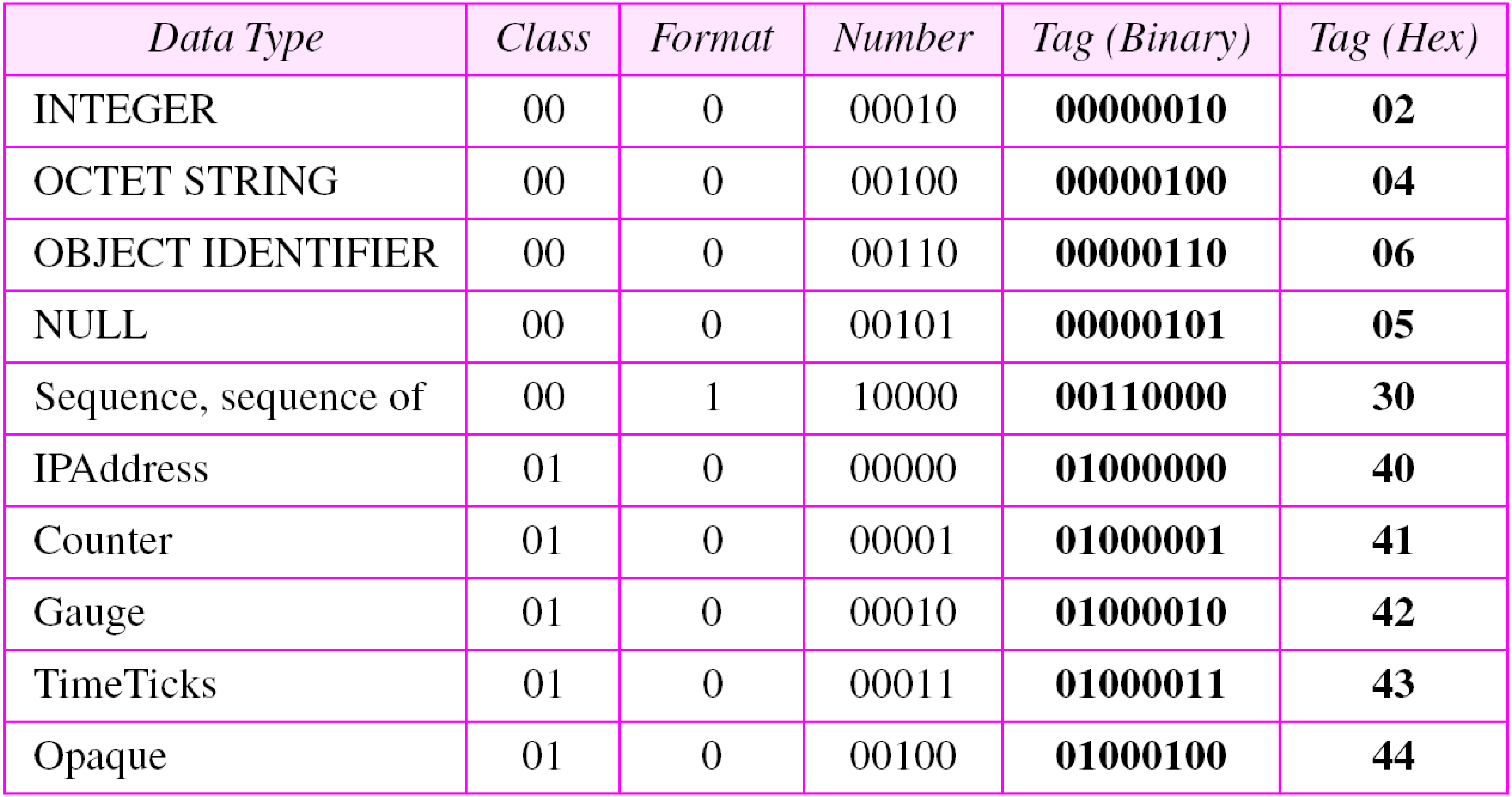
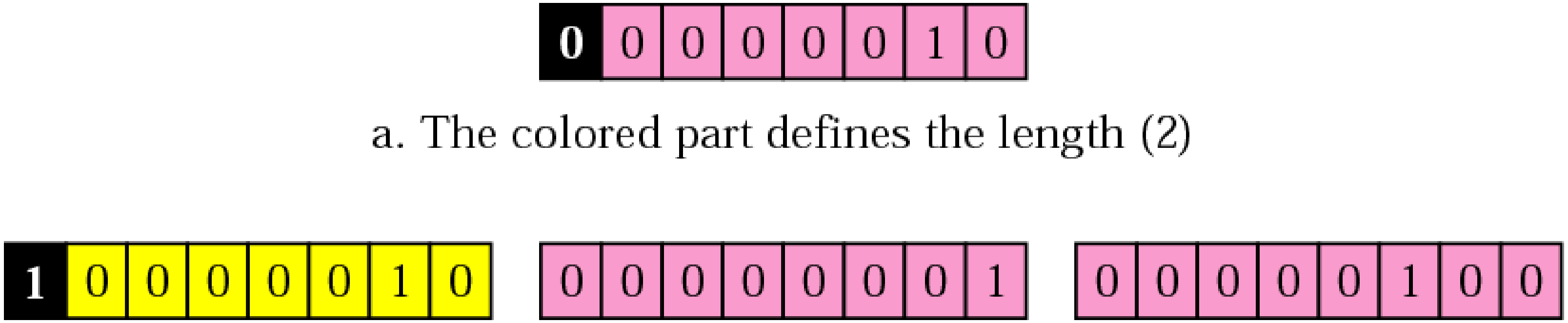


Table 21.2 Codes for data types



Data Type	*Class*	*Format*	*Number*	*Tag (Binary)*	*Tag (Hex)*
INTEGER	00	0	00010	**00000010**	**02**
OCTET STRING	00	0	00100	**00000100**	**04**
OBJECT IDENTIFIER	00	0	00110	**00000110**	**06**
NULL	00	0	00101	**00000101**	**05**
Sequence, sequence of	00	1	10000	**00110000**	**30**
IPAddress	01	0	00000	**01000000**	**40**
Counter	01	0	00001	**01000001**	**41**
Gauge	01	0	00010	**01000010**	**42**
TimeTicks	01	0	00011	**01000011**	**43**
Opaque	01	0	00100	**01000100**	**44**

Figure 21.9 *Length format*



a. The colored part defines the length (2)

b. The yellow part defines the length of the length (2 bytes);
the pink bytes define the length (260 bytes)

EXAMPLE 1

Figure 21.10 shows how to define INTEGER 14.

See Next Slide

Figure 21.10 *Example 1, INTEGER 14*

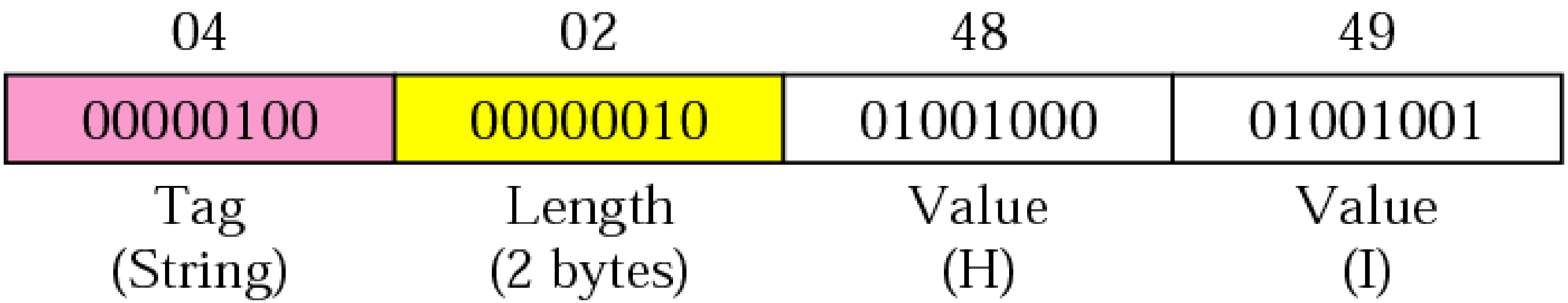
02	04	00	00	00	0E
00000010	00000100	00000000	00000000	00000000	00001110
Tag (integer)	Length (4 bytes)	Value (14)			

EXAMPLE 2

Figure 21.11 shows how to define the OCTET STRING "HI."

See Next Slide

Figure 21.11 *Example 2, OCTET STRING “HI”*



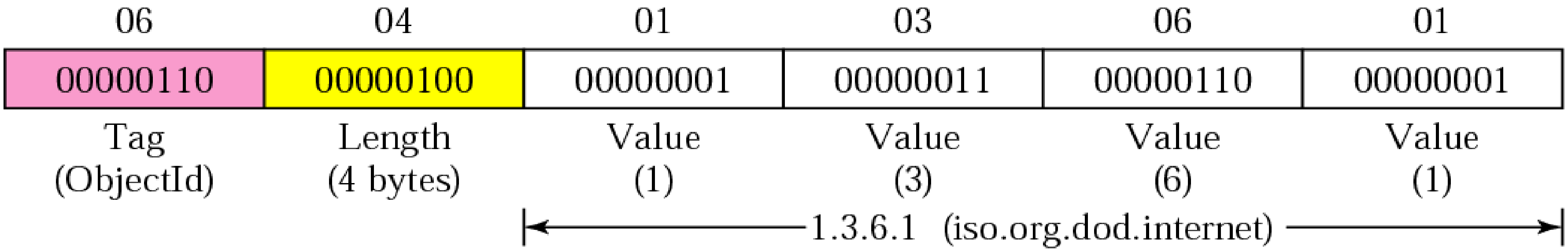
04	02	48	49
00000100	00000010	01001000	01001001
Tag (String)	Length (2 bytes)	Value (H)	Value (I)

EXAMPLE 3

Figure 21.12 shows how to define ObjectIdentifier 1.3.6.1 (iso.org.dod.internet).

See Next Slide

Figure 21.12 *Example 3, ObjectIdentifier 1.3.6.1*



06	04	01	03	06	01
00000110	00000100	00000001	00000011	00000110	00000001
Tag (ObjectId)	Length (4 bytes)	Value (1)	Value (3)	Value (6)	Value (1)

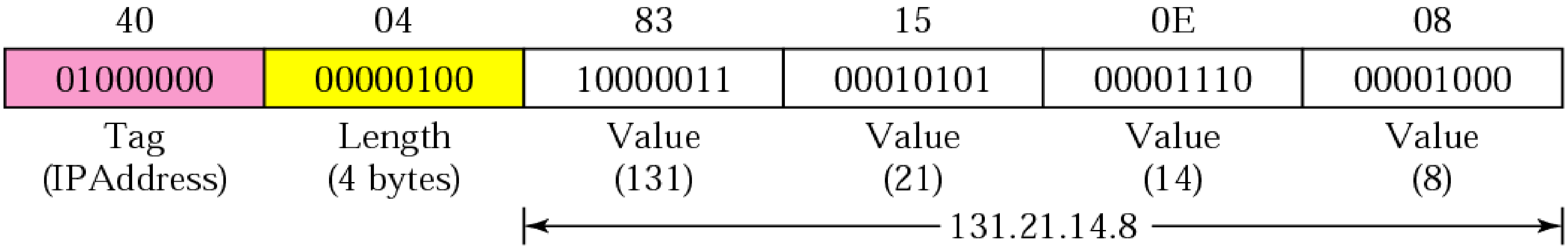
1.3.6.1 (iso.org.dod.internet)

Example 4

Figure 21.13 shows how to define IPAddress 131.21.14.8.

See Next Slide

Figure 21.13 *Example 4, IPAddress 131.21.14.8*



40	04	83	15	0E	08
01000000	00000100	10000011	00010101	00001110	00001000
Tag (IPAddress)	Length (4 bytes)	Value (131)	Value (21)	Value (14)	Value (8)
		131.21.14.8			

21.4 MIB

MIB is a component used in network management. Each agent has its own MIB, a collection of all the objects that the manager can manage.

The topics discussed in this section include:

Accessing MIB Variables
Lexicographic Ordering

Figure 21.14 *mib-2*

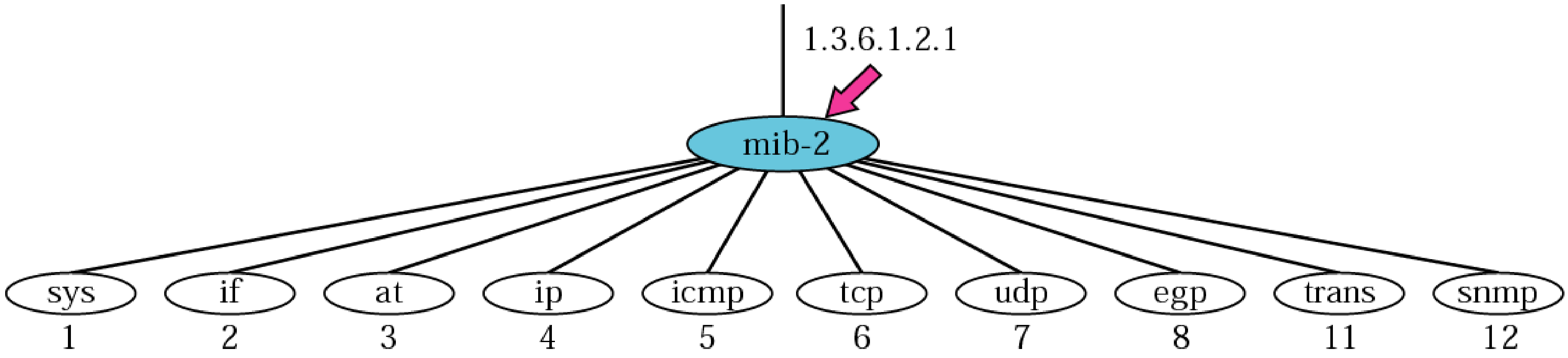


Figure 21.15 *udp group*

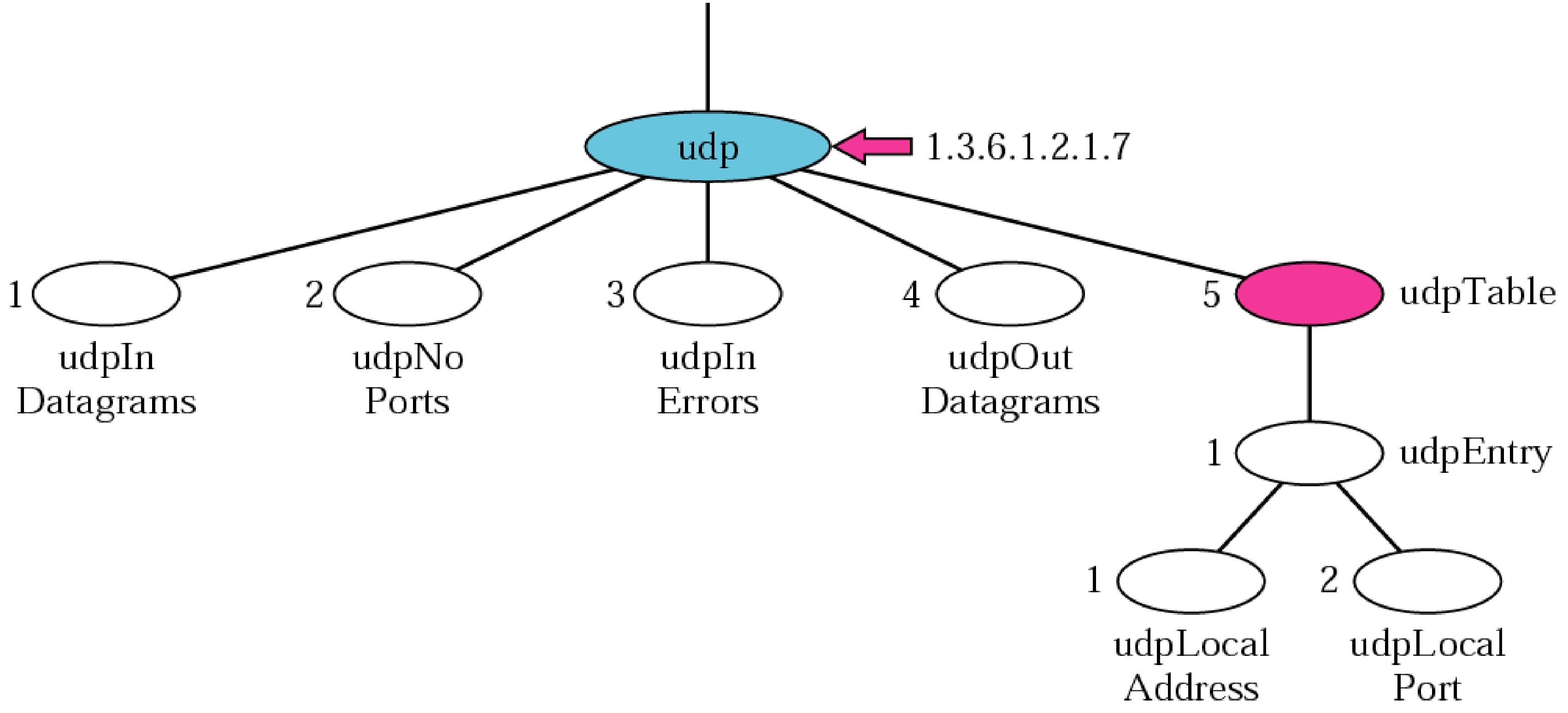


Figure 21.16 *udp variables and tables*

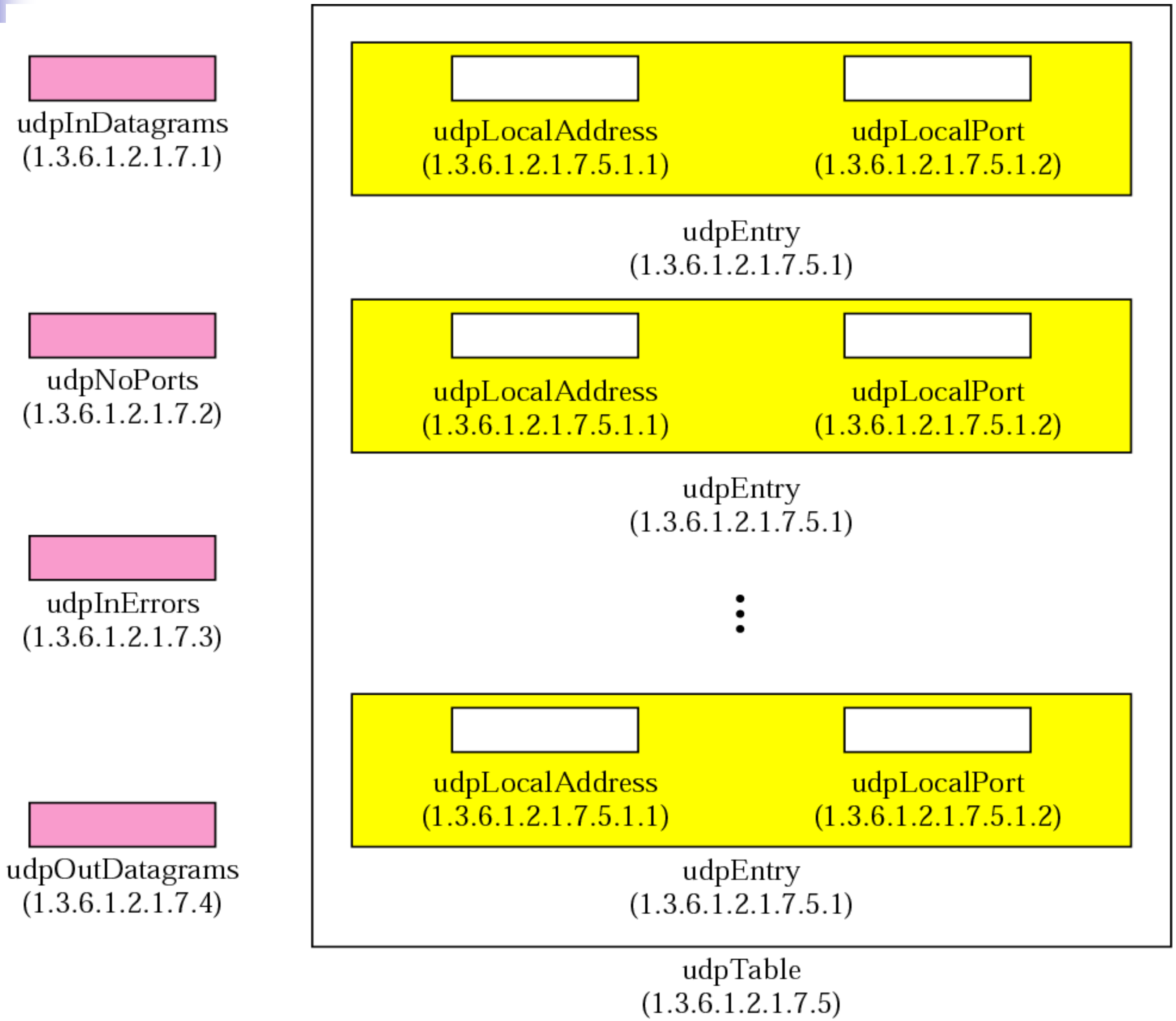
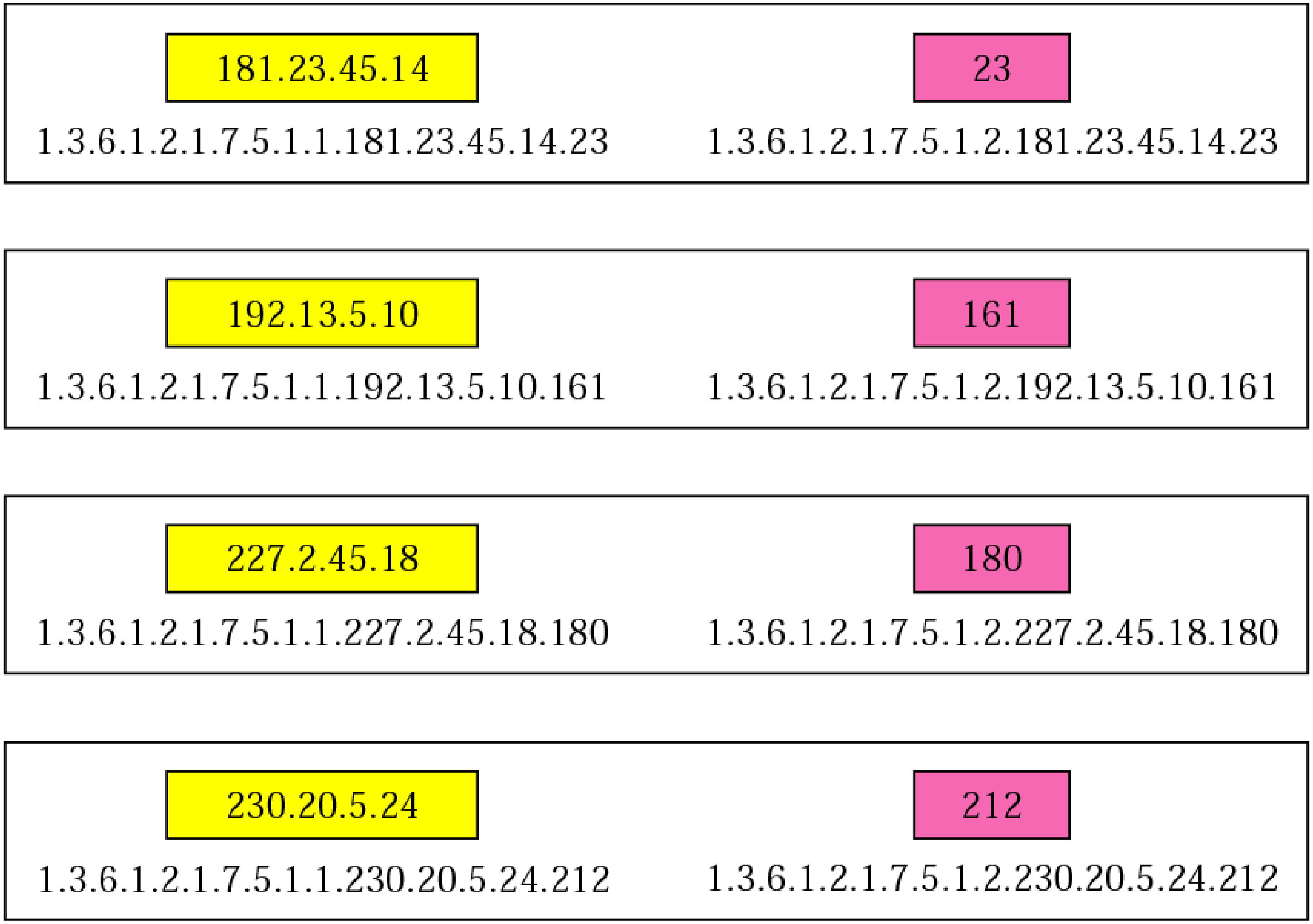


Figure 21.17 *Indexes for udpTable*



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

1.3.6.1.2.1.7.5.1.2.192.13.5.10.161

227.2.45.18

180

1.3.6.1.2.1.7.5.1.1.227.2.45.18.180

1.3.6.1.2.1.7.5.1.2.227.2.45.18.180

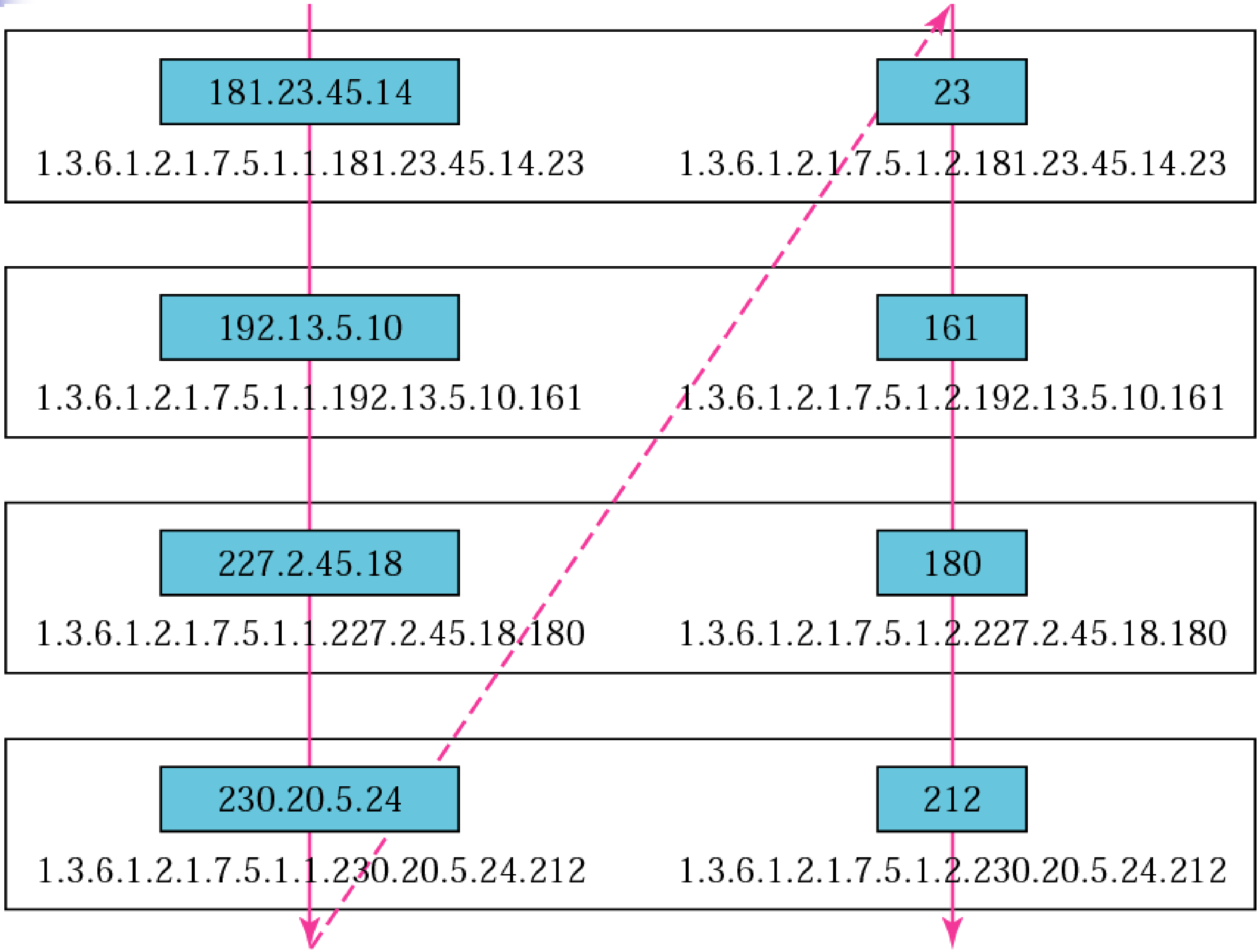
230.20.5.24

212

1.3.6.1.2.1.7.5.1.1.230.20.5.24.212

1.3.6.1.2.1.7.5.1.2.230.20.5.24.212

Figure 21.18 *Lexicographic ordering*



21.5 SNMP

SNMP is an application program that allows 1) a manager to retrieve the value of an object defined in an agent; 2) a manager to store a value in an object defined in an agent; and 3) an agent to send an alarm message about an abnormal situation to the manager

The topics discussed in this section include:

PDUs
Format

Figure 21.19 *SNMP PDUs*

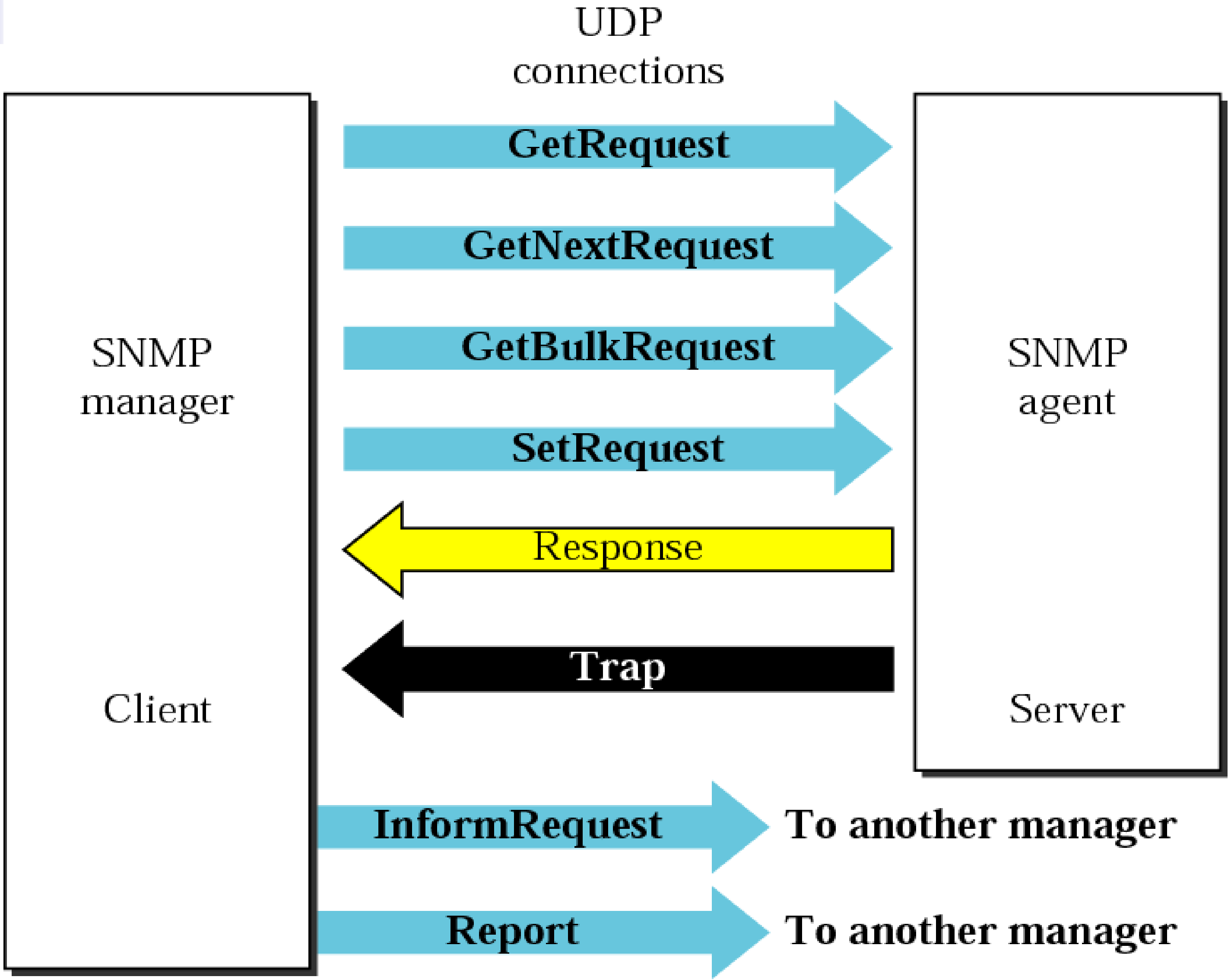
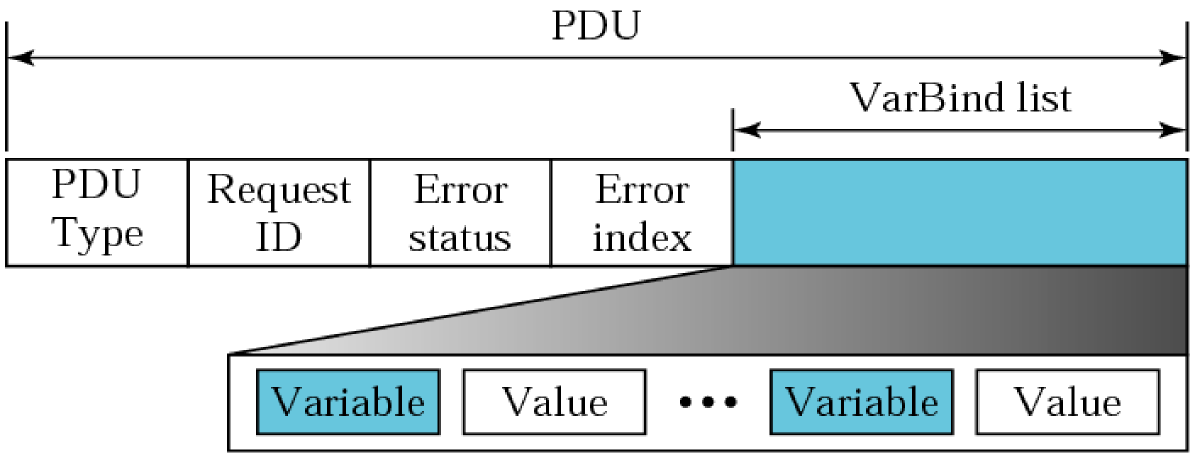


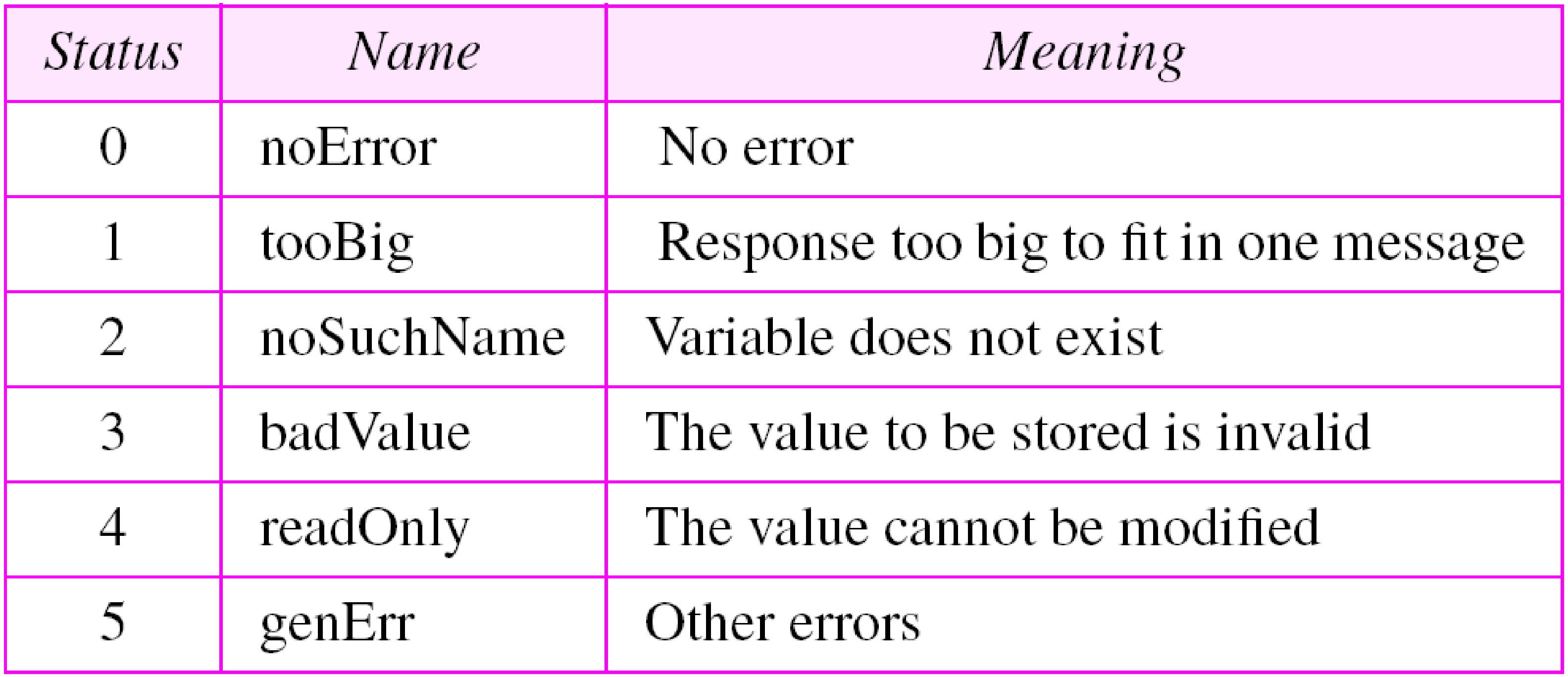
Figure 21.20 *SNMP PDU format*



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.

Table 21.3 Types of errors



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

21.6 MESSAGES

A message in SNMP is made of four elements: version, header, security parameters, and data (which includes the encoded PDU).

Figure 21.21 *SNMP message*

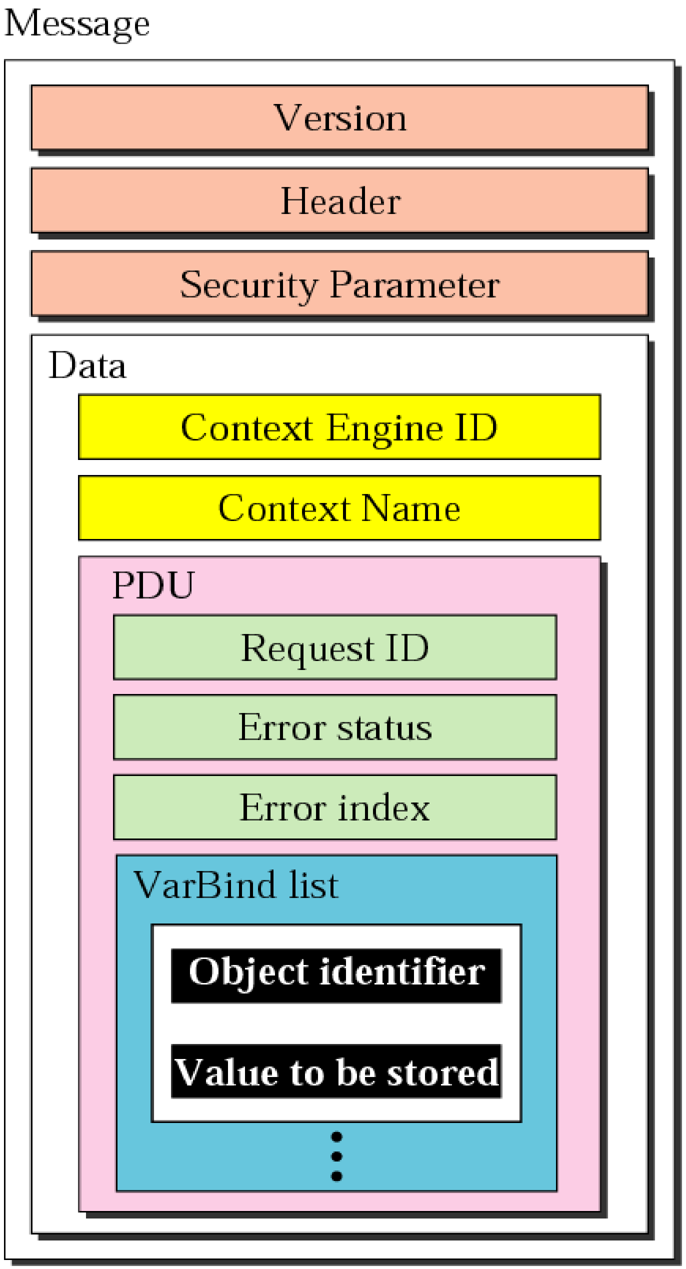
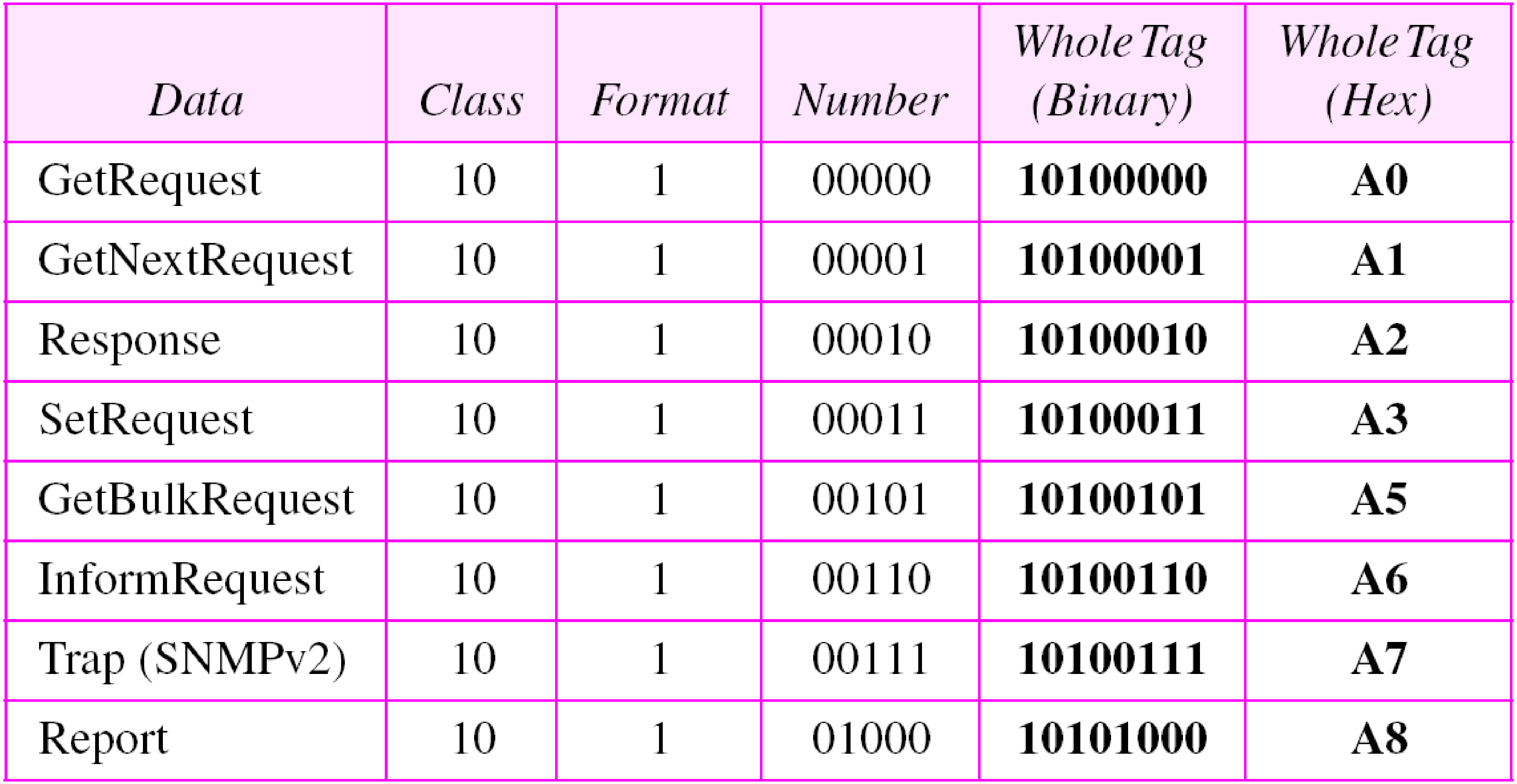


Table 21.4 Codes for SNMP messages



Data	*Class*	*Format*	*Number*	*Whole Tag (Binary)*	*Whole Tag (Hex)*
GetRequest	10	1	00000	**10100000**	**A0**
GetNextRequest	10	1	00001	**10100001**	**A1**
Response	10	1	00010	**10100010**	**A2**
SetRequest	10	1	00011	**10100011**	**A3**
GetBulkRequest	10	1	00101	**10100101**	**A5**
InformRequest	10	1	00110	**10100110**	**A6**
Trap (SNMPv2)	10	1	00111	**10100111**	**A7**
Report	10	1	01000	**10101000**	**A8**

EXAMPLE 5

In this example, a manager station (SNMP client) uses the GetRequest message to retrieve the number of UDP datagrams that a router has received. There is only one VarBind entity. The corresponding MIB variable related to this information is udpInDatagrams with the object identifier 1.3.6.1.2.1.7.1.0. The manager wants to retrieve a value (not to store a value), so the value defines a null entity. Figure 21.22 shows the conceptual view of the packet showing the hierarchical nature of sequences. We have used white and color boxes for the sequence and a gray one for the PDU.

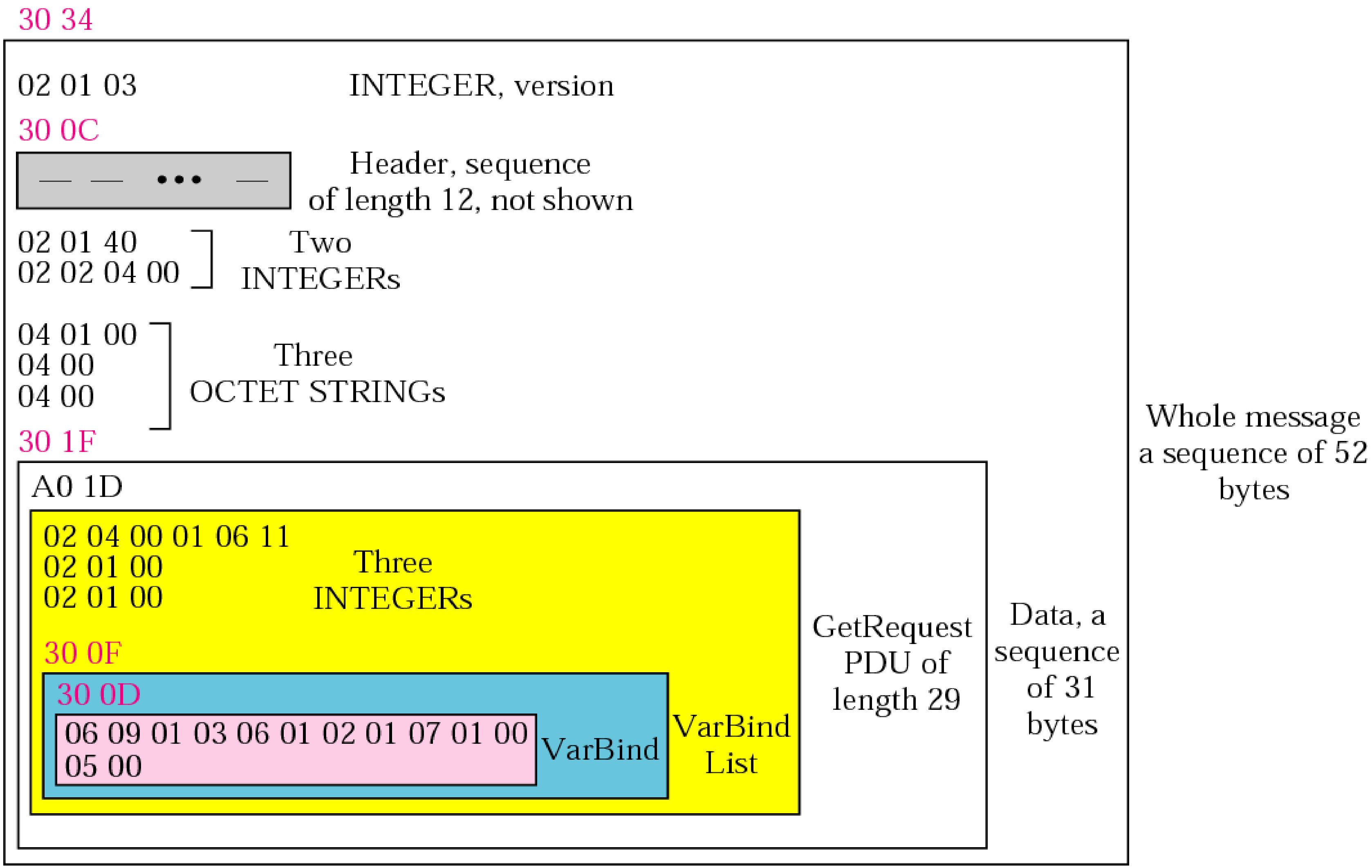
See Next Slide

EXAMPLE 5

The VarBind list has only one VarBind. The variable is of type 06 and length 09. The value is of type 05 and length 00. The whole is a sequence of length 0D (13). The VarBind list is also a sequence of length 0F (15). The GetRequest PDU is of length 1D (29). Now we have three OCTET STRINGs related to security parameter, security model, and flags. Then we have two integers defining maximum size (1024) and message ID (64). The header is a sequence of length 12, which we left blank for simplicity. There is one integer, version (version 3). The whole message is a sequence of 52 bytes. Figure 21.23 shows the actual message sent by the manager station (client) to the agent (server).

See Next Slide

Figure 21.22 *Example 5*



30 34

02 01 03 INTEGER, version

30 0C

— — ••• —

Header, sequence of length 12, not shown

02 01 40
02 02 04 00

Two INTEGERs

04 01 00
04 00
04 00

Three OCTET STRINGs

30 1F

A0 1D

02 04 00 01 06 11
02 01 00
02 01 00

Three INTEGERs

30 0F

30 0D

06 09 01 03 06 01 02 01 07 01 00
05 00

VarBind

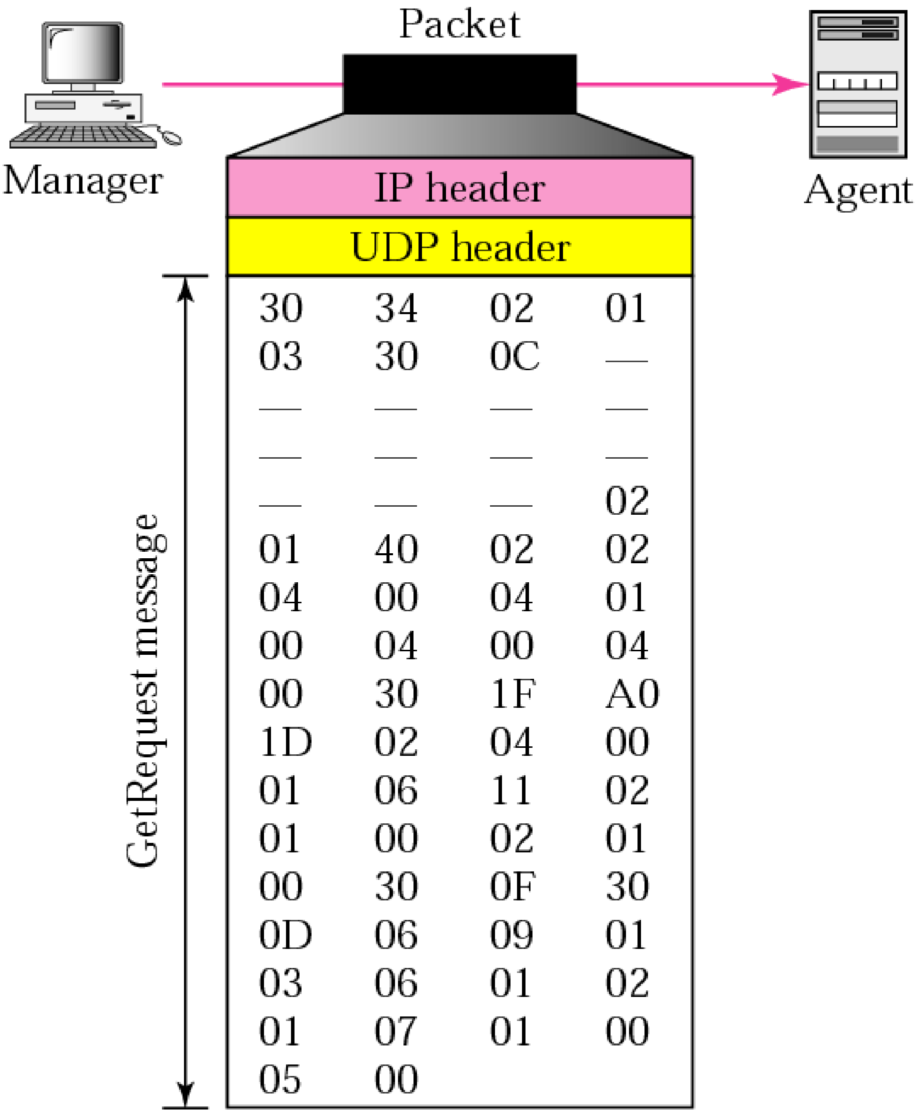
VarBind List

GetRequest PDU of length 29

Data, a sequence of 31 bytes

Whole message a sequence of 52 bytes

Figure 21.23 *GetRequest message*

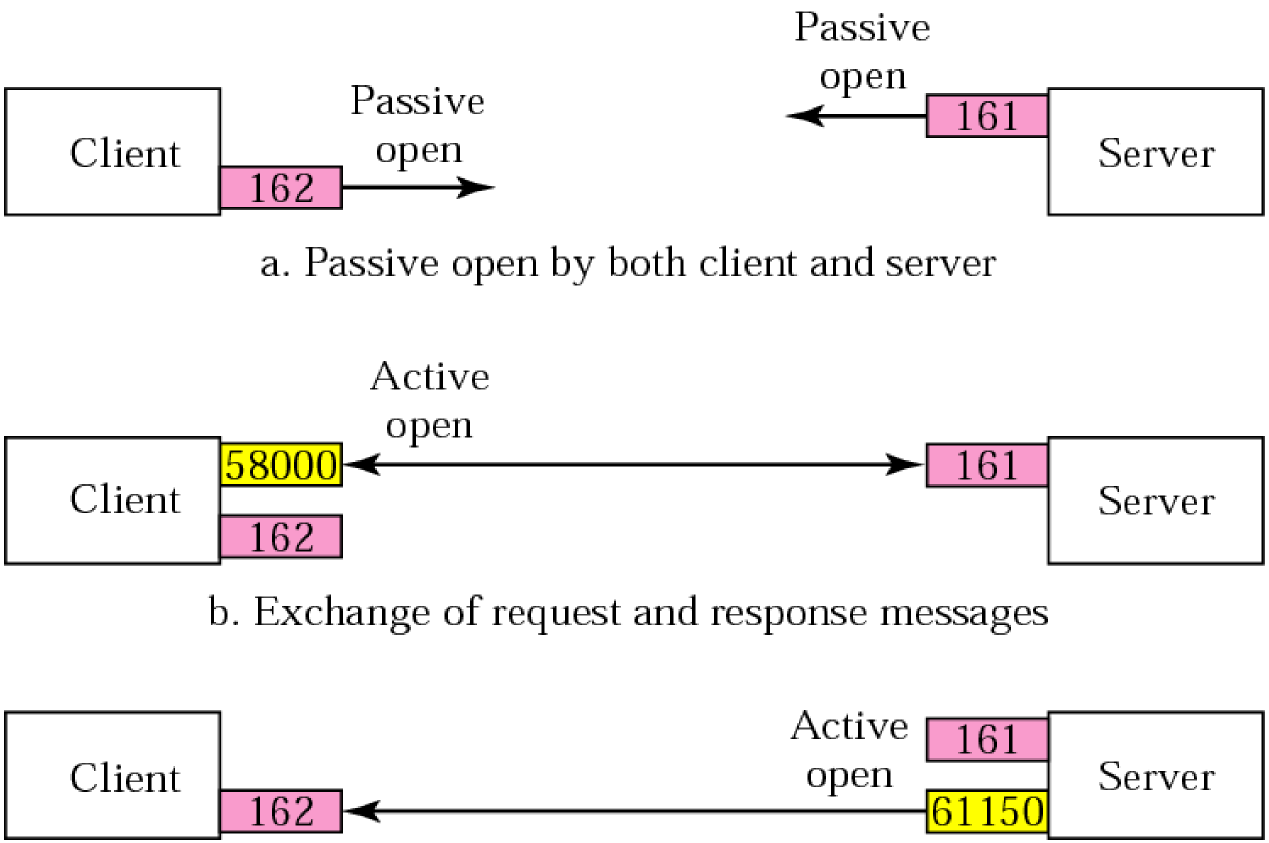


30	34	02	01
03	30	0C	—
—	—	—	—
—	—	—	—
—	—	—	02
01	40	02	02
04	00	04	01
00	04	00	04
00	30	1F	A0
1D	02	04	00
01	06	11	02
01	00	02	01
00	30	0F	30
0D	06	09	01
03	06	01	02
01	07	01	00
05	00		

21.7 UDP PORTS

SNMP uses the services of UDP on two well-known ports, 161 and 162. The well-known port 161 is used by the server (agent), and the well-known port 162 is used by the client (manager).

Figure 21.24 *Port numbers for SNMP*



a. Passive open by both client and server

b. Exchange of request and response messages

c. Server sends trap message

21.8 SECURITY

The main difference between SNMPv3 and SNMPv2 is the enhanced security. SNMPv3 provides two types of security: general and specific. SNMPv3 provides message authentication, privacy, and manager authorization.