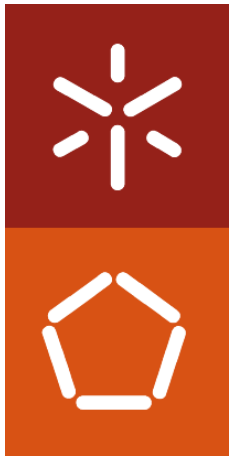
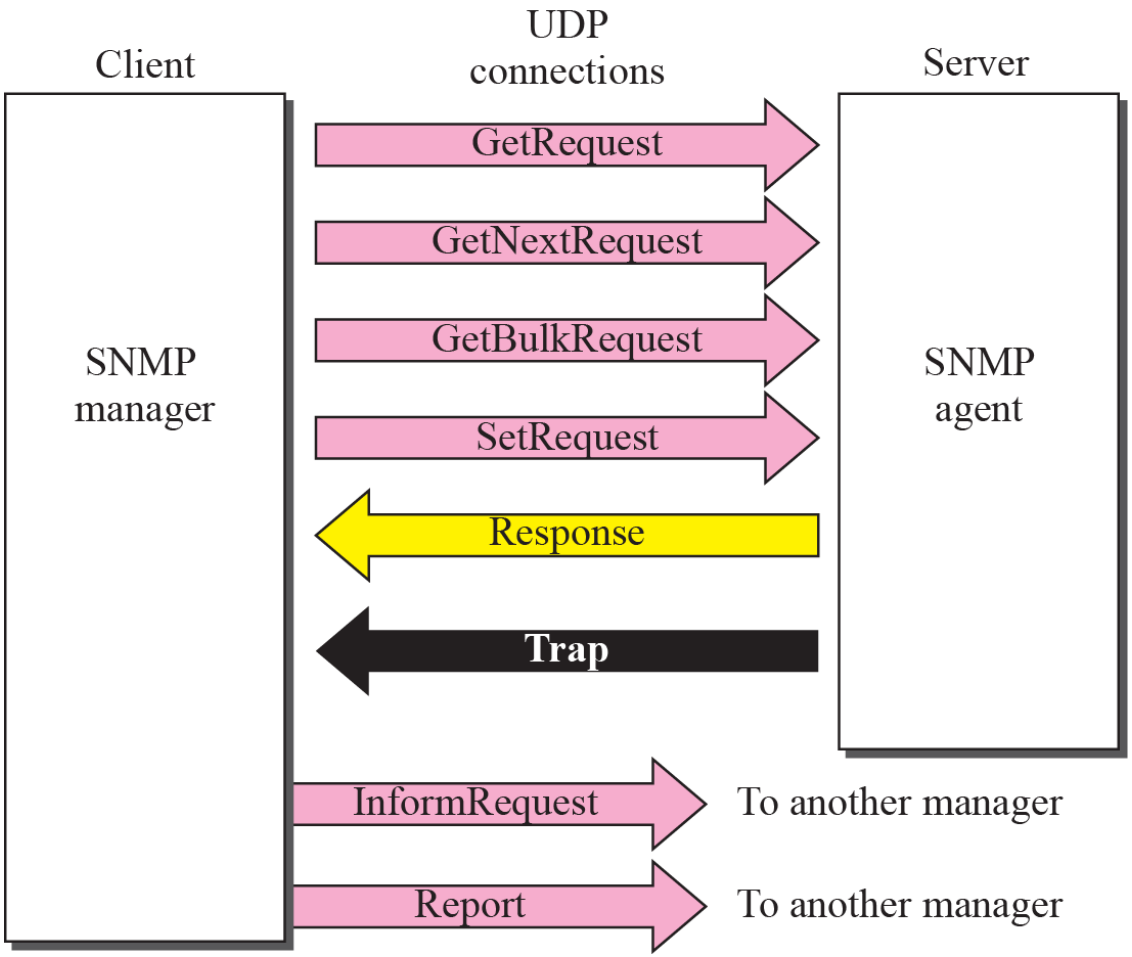
GESTÃO DE REDES / NETWORK MANAGEMENT

Notas complementares / Complementary notes

SNMP Protocol Data Units
Basic Encoding Rules



SNMP PDUs

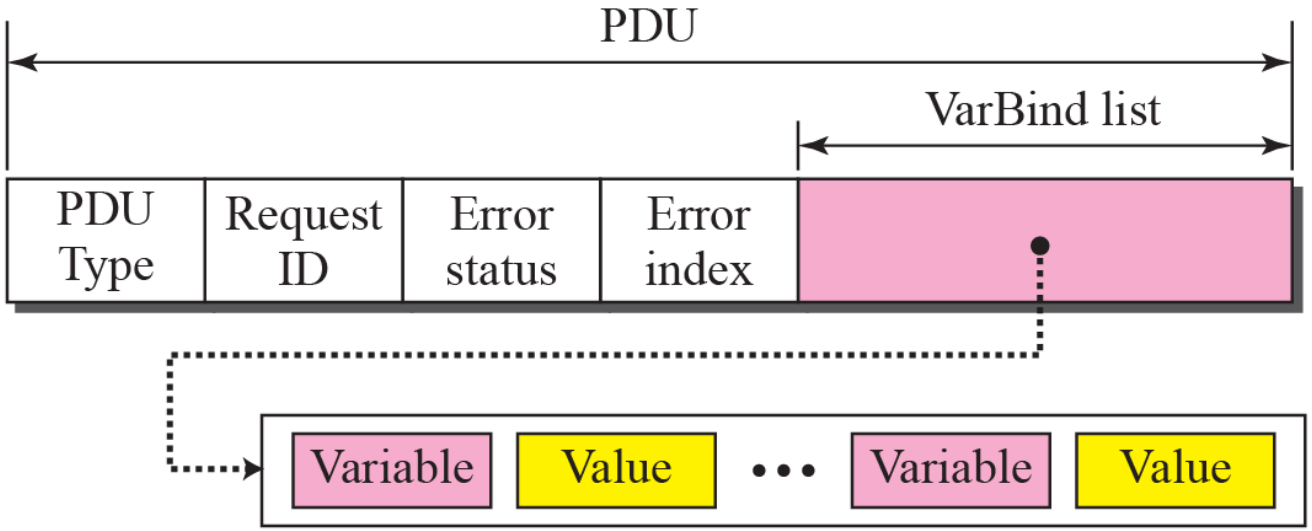


Livro "TCP/IP Protocol Suite", Behrouz A. Forouzan, McGraw-Hill (capítulo 24)

SNMP Messages (v1/v2c)

```
SNMPv1/SNMPv2c Message {
        version     snmpv1=0 snmpv2c=1
        community   STRING
        PDU (operation) {
                request-id          INTEGER
                error-status        INTEGER
                error-index         INTEGER
                variable-bindings {
                        OID, VALUE
                        …
                }
        }
}
```

SNMP PDUs



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.

Livro "TCP/IP Protocol Suite", Behrouz A. Forouzan, McGraw-Hill (capítulo 24)

SNMP PDUs

Table 24.3 *PDU Types*

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**

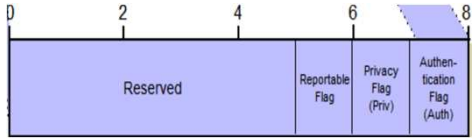
Table 24.4 *Types of Errors*

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

Livro "TCP/IP Protocol Suite", Behrouz A. Forouzan, McGraw-Hill (capítulo 24)

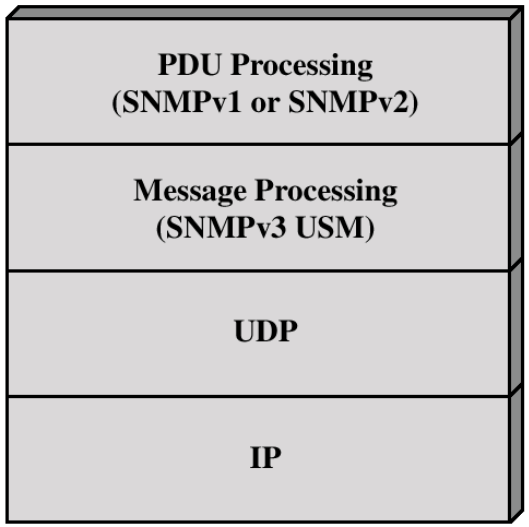
SNMP Messages (v3)

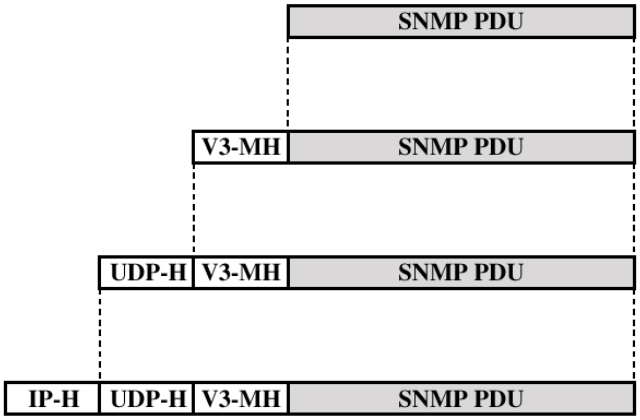
```
SNMPv3Message {
  Version                                INTEGER
                                         (snmpv3 = 3)

  HeaderData {
         msgID                           INTEGER
         MaxSize                         INTEGER
         Flags                           STRING
         SecurityModel                   INTEGER
                                         (USM = 3)
  }
  UsmSecurityParameters STRING {
         AuthoritativeEngineID           STRING,
         AuthoritativeEngineBoots        INTEGER
         AuthoritativeEngineTime         INTEGER
         UserName                        STRING
         AuthenticationParameters        STRING
         PrivacyParameters               STRING
  }
  ScopedPduData {
         contextEngineID                 STRING,
         contextName                     STRING,
         PDU                             SNMPv2 PDUs
  }
}
```



SNMP Messages (v3)

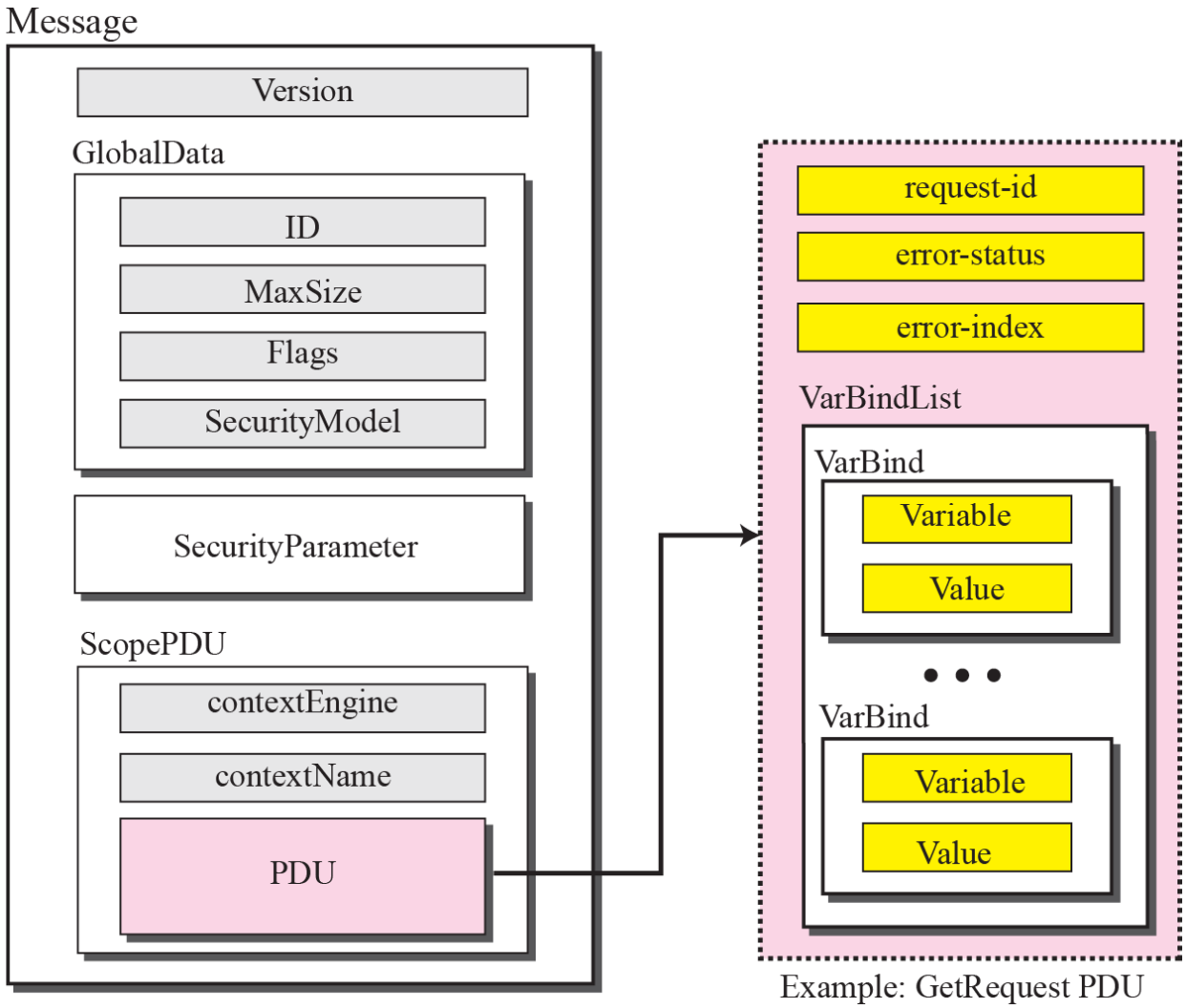
- **SNMPv3 defines a new message format that includes security and encapsulates v1/v2c messages**





IP-H = IP header
UDP-H = UDP header
V3-MH = SNMPv3 message header
PDU = Protocol data unit

SNMP Messages (v3)



Livro "TCP/IP Protocol Suite", Behrouz A. Forouzan, McGraw-Hill (capítulo 24)

SNMP Messages (v3)

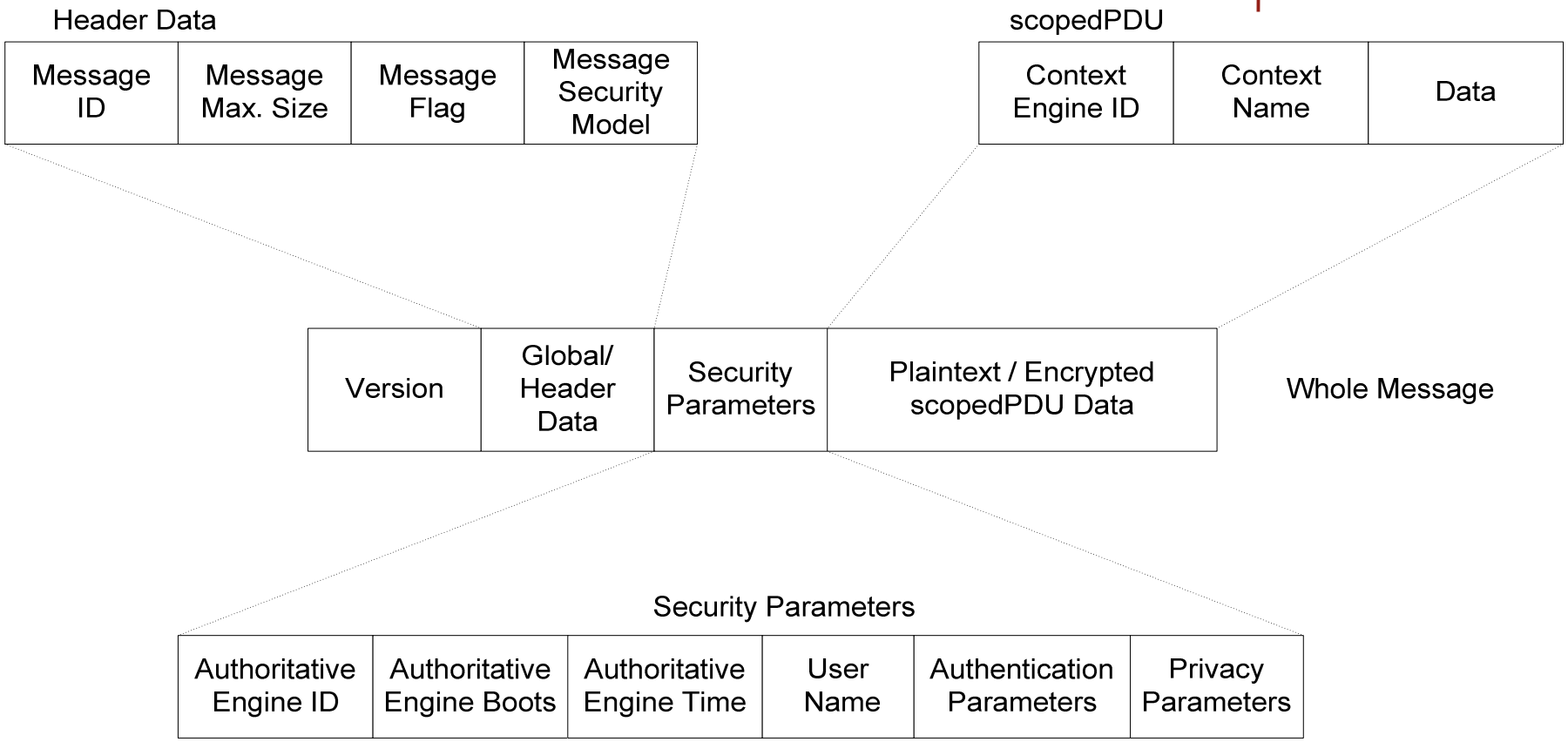
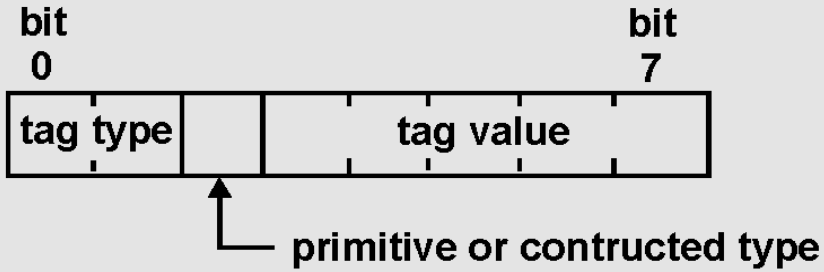


Figure 7.12 SNMPv3 Message Format

BER – Basic Encoding Rules

- **The data types defined in ASN.1 must be sent between heterogeneous systems**
  - We need a universal encoding scheme, independent of the hardware/operating system/software (encoding rules)
- **BER: Basic Encoding Rules**
  - TLV Triplets (Tag, Lenght, Value) for each type:

  - 8 bit Tags:

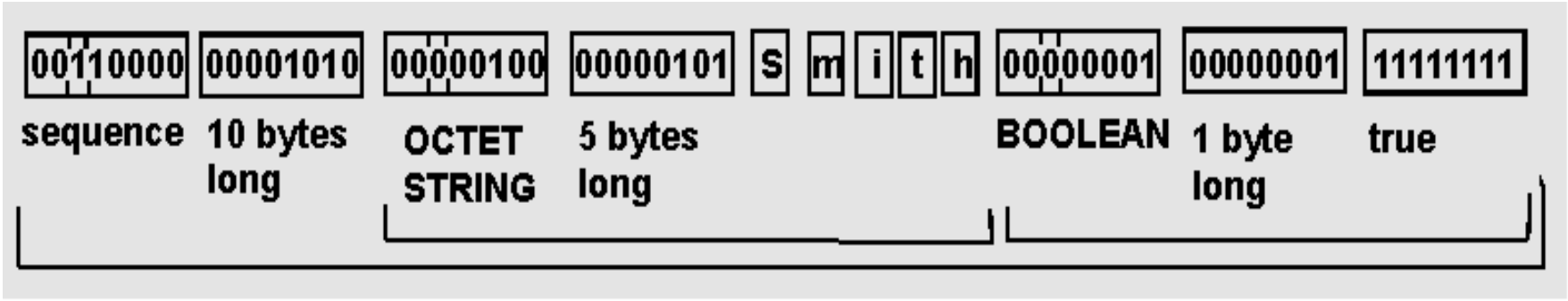


BER – Basic Encoding Rules

- **Example for the following ASN.1 Sequence:**

```
Attendee ::= SEQUENCE {
  name   OCTET STRING,
  paid    BOOLEAN
}
```

- The BER byte stream for the record data **{"Smith", T}** is:



BER – Basic Encoding Rules

- **BER is defined in ITU-T standard X.690**
- **X.690 defines subsets of BER (simplified versions, with pragmatic decisions, no choices):**
  - Canonical Encoding Rules (CER)
  - Distinguished Encoding Rules (DER)
- **But there are others available:**
  - XML Encoding Rules (XEL)
  - JSON Encoding Rules (JER)
  - ...
- **Question/discussion: How efficient is BER encoding?**
  - ... ? ....

BER – Basic Encoding Rules

- **The ASN.1 "compilers" generate definitions for:**
  - Definitions of data structures (in the programming language)
  - Functions for Encoding/Decoding data structures