TTM 4128 Exam May 19th 2012 (English)

Appendix 1: SNMPv2 PDU Definitions
Appendix 2: ASN.1 BER Encoding Summary

Task 1. SNMP and WEB-based management -General (35%)
Please provide short and precise answers.

1.1. Describe shortly the various models (viewpoints) that constitute the SNMP Network Management Framework.

Administrative model: Manager and Agents
Information Model: SMI (Structure of Mangement Information) and MIB (Managed Information Objects).
Communication Model: Communities, SNMP Message and PDUs. Transport protocol is UDP.

1.2. Explain shortly access control and security features in SNMPv1 and v2 which is part of the protocols

SNMPv1 and 2 has access control defined by communities

1.3. Explain shortly the security services provided in SNMPv3.

Avoidance of:
- Modification of information by unauthorized user
- Masquerade: change of originating address by unauthorized user
- Message stream modification
- Disclosure

1.4. What is the semantic Web?

The semantic Web is the Web that carries semantics or meanings enabling users to search not only for documents that contain data, but also for the desired data itself, through "semantic" identification and location techniques

It supports software agents that are able not only to locate data but to perform meaningful tasks with data automatically and on the fly that must be done manually on the first version of the World Wide Web and episodically by computer users.

1.5. List at least 3 key advantages of the web-based management systems.

Wide availability, cheap technology, applicability, etc.

1.6. What are the advantages of CIM compared to the information model used in the SNMP?

Object Oriented, classifications, encapsulation of data and behavior, dependencies, XML representation, etc.

1.7. Which management standard would you use to achieve configuration management of multiple network devices? Explain some of the functionalities of the selected standard with respect to configuration management.

NETCONF. Addresses configuration state as well as operational state, handles concurrency, distinguish multiple configurations, handled distribution as well as activation of configurations, handles configuration change events, support standard tools for comparison, versioning, etc.

Task 2. SNMPv2 protocol encoding (35%)
Please provide short and precise answers.

The definitions of SNMPv2 PDUS and ASN.1 BER Encoding Summary are given in Appendix 1 and 2, respectively.

2.1. Explain the TLV encoding principle.

TLV principle is used for the encoding of instances of ASN.1 types. For each ASN.1 instance there is a type field, length field and value field. The type field has the encoding tag of the ASN.1 type, length has the length of the ASN.1 instance value in number of octets and the value is the ASN.1 instance value. The value can be a TLV field of a nested ASN.1 instance.

2.2. What are the elements that constitute a managed object type?

■ ***object ID*** **and**	**unique ID for the object type**
name	**unique name for the object type**
■***syntax***	**the variable used to model the object type**
■ ***access***	**privilege to a managed object**
■ ***status***	**implementation requirements**
■ ***definition***	**textual description of the semantics of object type**

2.3. Explain shortly the role of the VarBindList in a SNMP PDU.

One VarBind element in the VarBindList is related to one managed object instance. The VarBindList is then related to a set of object managed instances.

2.4. Explain shortly the content of a VarBind element. How to interpret the CHOICE in a VarBind element?

There are two TLV fields per VarBind element, these are ObjectName and a CHOICE between ObjectSyntax and NULL. The ObjectName TLV field will carry the instance of the object identifier of the managed object instance. The choice field will either contain an encoded instance of NULL or an encoded instance of a SYNTAX attribute of a managed object.

2.5. What is the encoding of the T part of the TLV encoding of GetRequest-PDU?

GetRequest-PDU::= [0] IMPLICIT PDU.
Context specific tag is encoded by bit 8 and 7 is 10. The PDU is constructed and the value is 0. So the Octet that encodes T will be 1010 0000,which is 90 hex.

GetRequest-PDU is now going to be used to retrieve data from an instance of sysObjectId. The value of OBJECTIDENTIFIER sysObjectId is {internet 2 2 1 2}, and the value of OBJECT IDENTIFIER internet is { 1 3 6 1}.

2.6. What is the TLV encoding of the VarBind element ObjectName for an instance of sysObjectId.

Type is OBJECT IDENTIFIER. The value is {1 3 6 1 2 2 1 2}.
1 3 is encoded in one octet as 43hex,the rest with one octet per value.
So there will be 7 octets in the L-field, i.e. 43 06 01 02 02 01 02 hex.
OBJECT IDENTIFIER has encoding tag = Universal 6.
The T-field will be 00 0 00110 = 06hex.
The length field will be 00000111 = 07hex.
So concatenated to: 06 07 43 06 01 02 02 01 02 hex.

2.7 What is the TLV encoding of GetRequest-PDU when the CHOICE in the VarBind element is encoded as unspecified, i.e. as a NULL type.
(An Instance of a NULL type is encoded with 00 in the L-field and empty V-field)

A GetRequest-PDU will be encoded according to the general definition of PDU, but as GetRequest-PDU is defined as IMPLICIT PDU the SEQUENCE type is not explicitly encoded. So a GetRequest-PDU will be the concatenation of

- TLV encoding of request-ID
- TLV encoding of error-status
- TLV encoding of error-index
- TLV encoding of the Varbind element

Both request-ID, error-status, and error-index has SYNTAX Integer. Assume for simplicity the values zero for all of these. This gives <02 01 00> hex for each of these fields. Concerning the TLV encoding of the Varbind element, we have the concatenation of the TLV value of the ObjectName field and the CHOICE which in this case is the TLV encoding of NULL.

The encoding of type field of NULL is 0000 0101 = 05 hex, the length field is 00 and there is no value, i.e. <05 00>

The Varbind element content is then : <06 07 34 06 01 02 02 01 02> <05 00>
Sequence is 0011 0000 = 30hex and length is 11 decimal 0B hex.
Total Varbind encoding of the SEQUENCE is then:

<30 0B 06 07 34 06 01 02 02 01 02 05 00>

GetRequest-PDU length is then 13 + 3*3 = 22 decimal = C0 hex so we get

GetRequest total encoding is then:

90 C0 <02 01 00> <02 01 00> <02 01 00> <30 0B 06 07 34 06 01 02 02 12 02 05 00>

Task 3. A Network Management Application (30%)
Please provide short and precise answers.

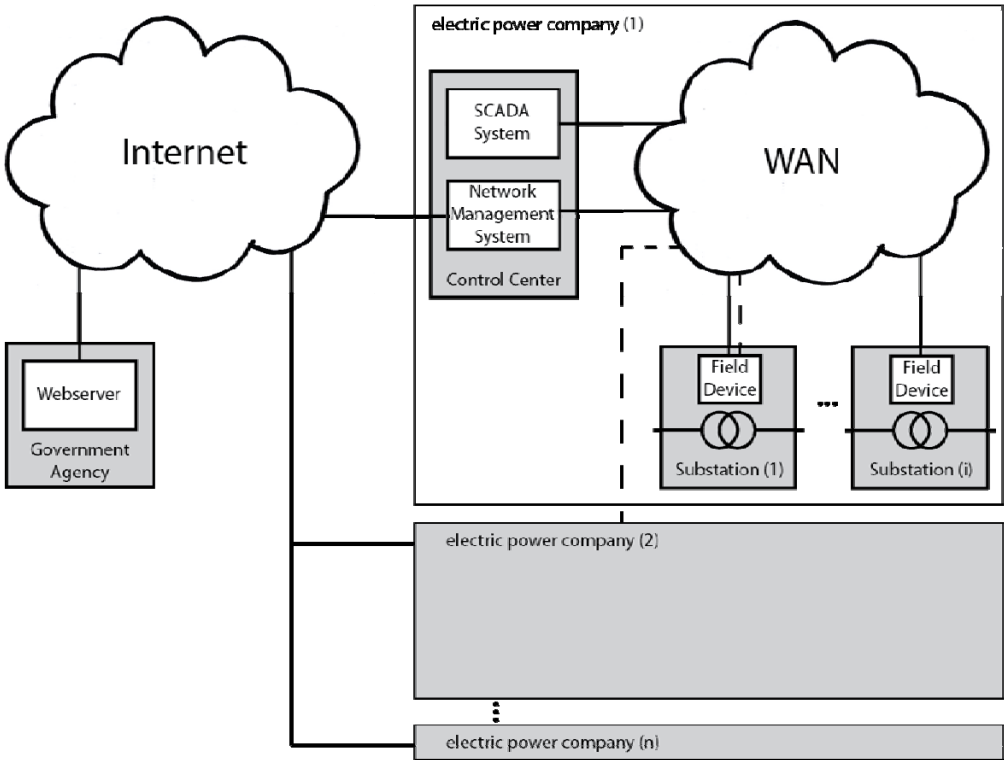
Today's power grid will be replaced by what is denoted as Smart Grid. Smart Grid improves the current grid by increased use of ICT devices and new communication infrastructure. In this task we set the focus on the management in Smart Grid.

The following figure gives an overview of the communication and management system in a solution for Smart Grid. There are a number of electric power companies. Each of them has substations with field devices which are connected to a control center via WAN. All of the electric power companies are connected to each other and a government agency via Internet.

First, consider only one electric power company. Each field device monitors and controls a voltage transformer. The company wants now to install a network management system to manage all these field devices.

3.1 The company wants to use SNMP. Considering the various models (viewpoints) of the SNMP framework, which SNMP components will be assigned to the network management system and field devices?

NMS: Network Manager, Network Management Base, MIB, SNMP protocol
Field Device: Network Agent, MIB, Managed Object Variables, SNMP protocol



3.2 The company would like to use two different methods for data collection, one method with the network management system in an active role, and a second method with the network management system in a passive role. Considering the SNMP protocol, a short description shall be given of the two alternative ways for management. What are advantages and disadvantages of the two methods?

Active manager: Queries to agents for data by Get: information is only sent if requested from manager
Active devices: Send trap if something happens, i.e. information sent when certain events occurs (pre-configured)

Advantages and disadvantages of the alternatives:

Get:
+ get information on demand
+ good to monitor the state of the system
- does not scale well
- NMS gets data only on request, could miss important events

Trap:
+ good for alarms and other event based information
+ information is sent only when needed
+ scales well (if trap condition is set well)
- less control for NMS (info only if a condition is true in field device)
- device could die unnoticed (no info = good OR dead)

3.3. The company wants to use net-snmp. Which components of this tool must be installed for the two alternative methods in 3.2, and where to install the components?

Snmpd on Field Devices, snmpget commands on NMS
Snmptrapd on NMS, snmptrap on field device

We do now consider two network management systems of two companies

3.4 The field device in Substation 1 shall both be accessed by Company 1 and 2. How to realize access control with net-snmp so that company 1 gets read and write access, while company 2 only gets read access? How to configure and where?

Use the community string. Add following lines to snmpd.conf in field device 1:
rwcommunity company_1; rcommunity company_2

The government agency now introduces a new law. All electric power companies have to register all changes of instances of the MIB Object healthStatus in the field devices. Every day at midnight, the companies have to send a report about all the changes to the government agency.

3.5. Which alternative from 3.2 would you recommend for getting the necessary information from the field devices? Explain your choice?

Alternative with traps. Only with *get* you could miss a status change.

3.6. The government agency would like to collect the information based on Web-services. In addition to the Web-server in the Government agency, each Control center will have a Web-server. Which information do the parties have to exchange to realize the Web-service.

Government agency: webserver needs a web service engine like AXIS2. Write a web service that accepts the data from the companies. Create a description of the service with WSDL. Provide companies with WSDL (or client stub).

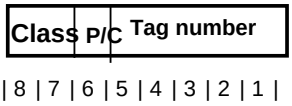
Companies: create client stub with WSDL, implement class based on client stub.

Appendix 1. ASN.1 BER Encoding Summary - 1

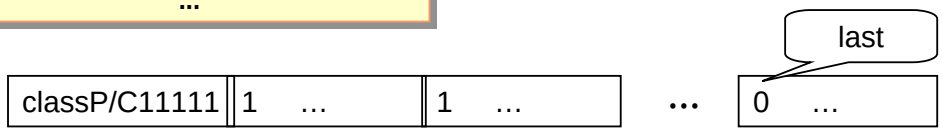
Universal Class Tag No	ASN.1 Type
1	BOOLEAN
2	INTEGER
3	BITSTRING
4	OCTETSTRING
5	NULL
6	OBJECTIDENTIIER
7	ObjectDescriptor
8	EXTERNAL
9	REAL
10	ENUMERATED
11-15	reserved for addenda
16	SEQUENCE, SEQUENCE OF
17	SET, SET OF
…	…

"Encoding" Tag Classes:
- 00 Universal
- 01 Application
- 10 Context specific
- 11 Private

P/C (6th bit):
- 0 primitive encoding
- 1 constructed encoding



Tag Number:
- <31 directly
- >=31 as below:



ASN.1 BER Encoding Summary - 2

Definite form:

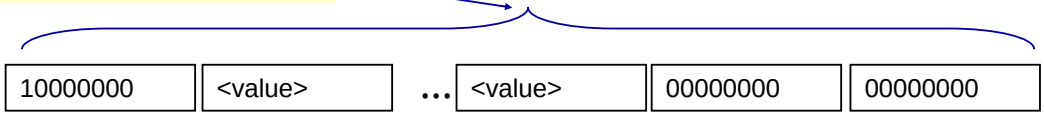
- short: length < 128 → 0 | <length>
- long: length > 128 → 1 | <length of length field> | <length field> … <length field>

Length field < 127

Length field

Max Length of value field = 2**(126*8)= 2**1008

Indefinite form: Only for constructed types



Start Octet | value octets | Final Sequence

OBJECT IDENTIFIER is encoded with each sub identifier value encoded as an octet. An exception is iso (1) and organization (3), which is encoded as (43).

Instances of NULL is encoded with L-field is 00 hex and with empty V-field

Appendix 2: SNMPv2-PDU DEFINITIONS

```
SNMPv2-PDU DEFINITIONS ::= BEGIN
   IMPORTS  ObjectName, ObjectSyntax, Integer32 FROM SNMPv2-SMI;
   -- protocol data units
   PDUs ::=
      CHOICE {
         get-request          GetRequest-PDU,
         get-next-request     GetNextRequest-PDU,
         get-bulk-request     GetBulkRequest-PDU,
         response             Response-PDU,
         set-request          SetRequest-PDU,
         inform-request       InformRequest-PDU,
         snmpV2-trap          SNMPv2-Trap-PDU,
         report               Report-PDU,
      }

   -- PDUs
   GetRequest-PDU ::=       [0] IMPLICIT PDU
   GetNextRequest-PDU ::=   [1] IMPLICIT PDU
   Response-PDU ::=         [2] IMPLICIT PDU
   SetRequest-PDU ::=       [3] IMPLICIT PDU
   GetBulkRequest-PDU ::=   [5] IMPLICIT BulkPDU
   InformRequest-PDU ::=    [6] IMPLICIT PDU
   SNMPv2-Trap-PDU ::=      [7] IMPLICIT PDU
   Report-PDU ::            [8] IMPLICIT PDU

   max-bindings             INTEGER ::= 2147483647

   PDU ::=
      SEQUENCE {  request-id       Integer32,
                  error-status     INTEGER {
                  noError(0),
                  tooBig(1),
                  noSuchName(2), -- for proxy compatibility
                  badValue(3),     -- for proxy compatibility
                  readOnly(4),      -- for proxy compatibility
                   genErr(5),
                  noAccess(6),
                  wrongType(7),
                   wrongLength(8),
                  wrongEncoding(9),
                   wrongValue(10),
                  noCreation(11),
                   inconsistentValue(12),
                   resourceUnavailable(13),
                  commitFailed(14),
                  undoFailed(15),
                  authorizationError(16),
```

```
                notWritable(17),
                inconsistentName(18)        },

                error-index             INTEGER (0..max-bindings),
                variable-bindings   -- values are sometimes ignored
                                        VarBindList
                }

    BulkPDU ::=             -- MUST be identical in
        SEQUENCE {          -- structure to PDU
            request-id          Integer32,
            non-repeaters       INTEGER (0..max-bindings),
            max-repetitions     INTEGER (0..max-bindings),
            variable-bindings   -- values are ignored
                                VarBindList
        }

-- variable binding

VarBind ::=     SEQUENCE {name          ObjectName,
                                        CHOICE      {
                        value   ObjectSyntax,
                        unSpecified
                        -- in retrieval requests
                                            NULL,
                        -- exceptions in responses
                        noSuchObject[0]     IMPLICIT NULL,
                        noSuchInstance[1]   IMPLICIT NULL,
                        endOfMibView[2]     IMPLICIT NULL
                                            }
                        }

 -- variable-binding list

 VarBindList ::= SEQUENCE (SIZE (0..max-bindings)) OF VarBind

 END
```