NTNU
Norges teknisk-naturvitenskapelige universitet
Institutt for telematikk

EKSAMEN I TTM4128 – TJENESTE- OG RESSURSADMINISTRASJON
EXAM TTM4128 – SERVICE AND RESOURCE MANAGEMENT

Contact person / **Faglig kontakt**:	Finn Arve Aagesen
Tlf.:	99505728
Date / **dato**:	19.05.2008
Time / **tid**:	0900-1200
Remedies / **Tillatte hjelpemidler**:	**D**: No printed or handwritten remedies allowed. **D**: Ingen trykte eller håndskrevne hjelpemidler tillatt.
Språkform/Languages:	Norsk (Bokmål)/English (Den engelske oppgaveteksten er den originale og gyldige teksten.)
Sensurdato/**Results**:	9. juni/9 June

TTM 4128 Eksamen 19de Mai 2008

Vedlagt :

Forkortet versjon av RFC 1157
Forkortet versjon av RFC 1213
ASN.1 Enkodings-skjema

Oppgave 1: SNMPv1 (15%)

1.1 Forklar kort hva SNMP kan brukes til.
1.2 Hva er SNMP SMI ? List noen (3 eller flere) av de ting som SMI for SNMPv1 definerer.
1.3 Hva er en "MIB group"?
1.4 Hvilke MIB-objekter kan aksesseres for å bestemme om en "managed network element" er "bridge", "gateway" eller "router". Hvilken "MIB group" tilhører disse objekter?
(Navnene trenger ikke være eksakt riktig)
1.5 Hvordan identifiseres MIB-objekt-typer?
Hvordan identifiseres MIB-objekt-instanser?

Oppgave 2: SNMPv3 (15%)

2.1 List kort de "security threats" som håndteres av SNMPv3.
2.2 Hvilke "security services" er definert for å håndtere de definerte "threats"?

Oppgave 3: Tabell-aksess ved bruke av SNMPv1 (40%)

A "Manager is managing" en routing-tabell i en "router". "Manager"og "agent" kommuniserer ved bruk av SNMP versjon1. Forkortede versjoner av RFC 1157 and RFC 1213 er vedlagt. Vi har ip::=OBJCET INDENTIFIER{mgmt(2) mib-2(1) 4}

Følgende instans av ipRoutingTable finnes i en betraktet "router":

ipRouteDest	ipRouteIfIndex	ipRouteNextHop	ipRouteMetric1	ipRouteMetric2	ipRouteType
10.0.0.99	1	89.1.1.42	5	-1	3
9.1.2.3	2	99.0.0.3	3	-1	4
10.0.0.51	3	89.1.1.42	5	-1	3

3.1 (5%) Hva er OBJECT IDENTIFIER og OBJECT-SYNTAX av "columnar objects" av denne instans av ipRoutingTable?

3.2 (15%) "Manager" er interessert i følgende "managed objects".

ipRouteDest	ipRouteNextHop	ipRouteMetric1
10.0.0.99	89.1.1.42	5
9.1.2.3	99.0.0.3	3
10.0.0.51	89.1.1.42	5

Vis hvordan en "manager" kan aksessere objektene {ipRouteDest, ipRouteNextHop, ipRouteMetric1} i denne tabell ved bruk av hensiktsmessig sekvens av SNMP v1 "messages", når en SNMP "message" fra "manager" til "agent" brukes for å aksessere en rad i tabellen.

Skriv sekvensen av "messages" sent mellom "manager" og "agent", og inkluder "message" fra "agent" som indikerer at det ikke er flere instanser av objektene {ipRouteDest, ipRouteNextHop, ipRouteMetric1} i tabellen.

Hver "message" skal indikeres med SNMP "message name" og med relevante "VarBind element" som parametre. Dette er den samme konkvensjon som er brukt i boka, i forelesninger og i "RFCs" .

3.3 (10%) Vi betrakter første "Message" fra "Manager" til "Agent".

3.3.1 Definer typen til VarBind-elementene VarBind1, Varbind2, etc. i VarBindList ved bruk av ASN.1.

3.3.2 Definer instansene til VarBind-elementene definert i 3.3.1 med tilordnede verdier.

3.4 (10%) De BER-kodede VarBind-elementer beteges varbind1BER, varbind2BER, etc.

Definer varbind1BER.

OBJECTIDENTIFIER kodes med hver "sub identifier value" kodet som en oktett. Et unntak er iso(1) og organization (3), som kodes i en oktett som 43.

Oppgave 4. Semantisk WEB (10 %)

Forklar hva RDF er? Gi også et enkelt eksempel på en RDF spesifikasjon.

Oppgave 5. Web-based Management (20 %)

5.1 Hva er "the components of CIM"? Hva er formål og funkjonaliet til disse "CIM components"?

5.2 Forklar WBEM arkitekturen samt funskjonaliteten til komponentene i arkitekturen .

TTM 4128 Exam May 19th 2008

Enclosed :
Shortened version of RFC 1157
Shortened version of RFC 1213
ASN.1 Encoding Scheme

Task 1: SNMPv1 (15%)

- **1.1** Explain shortly what SNMP can be used for.
- **1.2** What is SNMP SMI? List at least 3 important issues defined in SMI for SNMPv1.
- **1.3** What is a MIB group?
- **1.4** Which MIB objects can be accessed to decide if a managed component is a bridge, gateway or a router. What is the MIB group of these objects? (Exact correct names are not needed.)
- **1.5** How are MIB types identified? How are MIB object instances identified?

Task 2: SNMPv3 (15%)

- **2.1** List shortly the security threats that are handled in SNMPv3.
- **2.2** Which security services are defined to handle the defined threats?

Task 3: Table traversal by SNMPv1 (40%)

A Manager is managing an instance of a routing table in a router. The manager and agent communicate by SNMP version 1. Shortened versions of RFC 1157 and 1213 are enclosed. We have ip::=OBJCET INDENTIFIER{mgmt(2) mib-2(1) 4}

The following instance of the ipRoutingTable exists in the considered router:

ipRouteDest	ipRouteIfIndex	ipRouteNextHop	ipRouteMetric1	ipRouteMetric2	ipRouteType
10.0.0.99	1	89.1.1.42	5	-1	3
9.1.2.3	2	99.0.0.3	3	-1	4
10.0.0.51	3	89.1.1.42	5	-1	3

3.1 (5%) What are the OBJECT IDENTIFIER and OBJECT-SYNTAX of the columnar objects of the instance of the ipRoutingTable?

3.2 (15%) The Manager is interested in the following managed objects.

ipRouteDest	ipRouteNextHop	ipRouteMetric1
10.0.0.99	89.1.1.42	5
9.1.2.3	99.0.0.3	3
10.0.0.51	89.1.1.42	5

Show how a manager can traverse the objects {ipRouteDest, ipRouteNextHop, ipRouteMetric1} of this table by using the appropriate sequence of SNMP v1 messages, and when one SNMP message from the manager to the agent is used to access one row of the table.

Write the sequence of messages exchanged between the manager and the agent including the message from the agent that indicates that there are no more instances of the objects {ipRouteDest, ipRouteNextHop, ipRouteMetric1} in the table.

Each message shall be indicated with SNMP message name and with relevant VarBind element as parameters. This is the same convention as used in the book, in the lectures and in the RFCs.

3.3 (10%) We are considering the first Message going from the Manager to the Agent.

3.3.1 Define the type of the VarBind elements VarBind1, Varbind2, etc. in VarBindList by using ASN.1.
3.3.2 Define instances of the VarBind elements defined in 3.3.1 with assigned values.

3.5 (10%) The encoding of the VarBind elements is denoted as varbind1BER, varbind2BER, etc.

Define varbind1BER

An OBJECTIDENTIFIER is encoded with each sub identifier value encoded as an octet. An exception is the iso(1) and organization (3), which are encoded in one octet as 43.

Task 4. Semantic WEB (10 %)

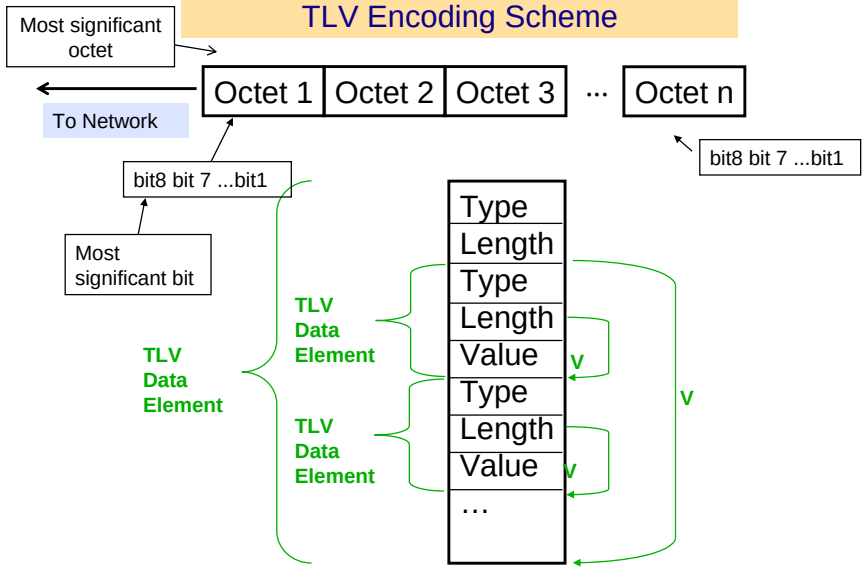
Explain what RDF is? Also give a simple RFD specification example.

Task 5. Web-based Management (20 %)

5.1 What are the components of CIM? What is the purpose and functionality of these CIM components?
5.2 Explain the WBEM architecture and the functionality of its components.

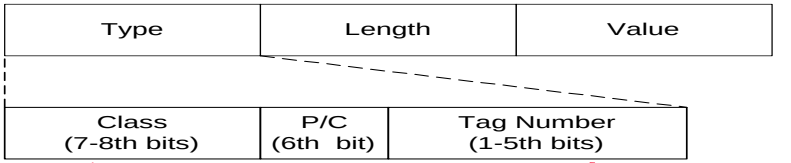
TTM4128 Exam: ASN.1 Encoding

TLV Encoding Scheme



The Most significant bit within the Most significant octet is transmitted first

Type Encoding



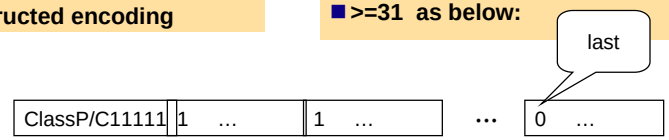
"encoding" class of types and "syntax" tags

Class	8th bit	7th bit
Universal	0	0
Application	0	1
Context-specific	1	0
Private	1	1

"encoding" tag no

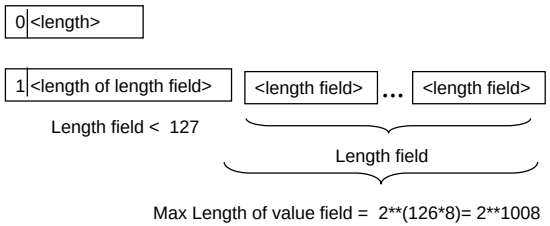
P/C (6th bit):
- **0 primitive encoding**
- **1 constructed encoding**

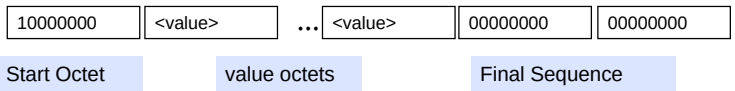
Value (bit 1-5):
- **<31 directly**
- **>=31 as below:**



Length (of Data) Field Encoding

Definite form
- short: length < 128
- long: length > 128
- Indefinite form: only for constructed types





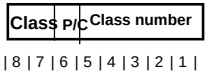
Indefinite form can only be used for constructed encoding, and not for primitive types. This because the final sequence can be part of data.

Type Field Encoding - Summary

Universal Class	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 addenda	reserved for
16 SEQUENCE OF	SEQUENCE,
17	SET, SET OF
...	...

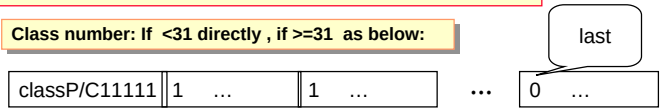
"Encoding" Tag Classes:
- **00 Universal**
- **01 Application**
- **10 Context specific**
- **11 Private**

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



IpAddress ::= [APPLICATION 0] -- in network-byte order
IMPLICIT OCTET STRING (SIZE (4))

Class number: If <31 directly , if >=31 as below:



```
RFC1157-SNMP DEFINITIONS ::= BEGIN

 IMPORTS
     ObjectName, ObjectSyntax, NetworkAddress, IpAddress, TimeTicks
         FROM RFC1155-SMI;


     -- top-level message

     Message ::=
             SEQUENCE {
                 version          -- version-1 for this RFC
                     INTEGER {
                         version-1(0)
                     },

                 community        -- community name
                     OCTET STRING,

                 data             -- e.g., PDUs if trivial
                     ANY          -- authentication is being used
             }

     -- protocol data units

     PDUs ::=
             CHOICE {
                         get-request
                             GetRequest-PDU,

                         get-next-request
                             GetNextRequest-PDU,

                         get-response
                             GetResponse-PDU,

                         set-request
                             SetRequest-PDU,

                         trap
                             Trap-PDU
                     }

     -- PDUs

     GetRequest-PDU ::=
         [0]
             IMPLICIT PDU

     GetNextRequest-PDU ::=
         [1]
             IMPLICIT PDU

     GetResponse-PDU ::=
         [2]
             IMPLICIT PDU

     SetRequest-PDU ::=
         [3]
             IMPLICIT PDU

     PDU ::=
             SEQUENCE {
                request-id
                     INTEGER,

                 error-status      -- sometimes ignored
                     INTEGER {
                         noError(0),
                         tooBig(1),
                         noSuchName(2),
                         badValue(3),
                         readOnly(4),
                         genErr(5)
                     },
```

```
            error-index       -- sometimes ignored
               INTEGER,

            variable-bindings -- values are sometimes ignored
                VarBindList
        }

Trap-PDU ::=
    [4]
       IMPLICIT SEQUENCE {
            enterprise        -- type of object generating
                              -- trap, see sysObjectID in [5]


                OBJECT IDENTIFIER,

            agent-addr        -- address of object generating
                NetworkAddress, -- trap

            generic-trap      -- generic trap type
                INTEGER {
                    coldStart(0),
                    warmStart(1),
                    linkDown(2),
                    linkUp(3),
                    authenticationFailure(4),
                    egpNeighborLoss(5),
                    enterpriseSpecific(6)
                },

            specific-trap  -- specific code, present even
                INTEGER,   -- if generic-trap is not
                           -- enterpriseSpecific

            time-stamp     -- time elapsed between the last
                TimeTicks, -- (re)initialization of the
                              network
                           -- entity and the generation of the
                              trap

             variable-bindings -- "interesting" information
                VarBindList
        }


-- variable bindings

VarBind ::=
        SEQUENCE {
            name
                ObjectName,

            value
                ObjectSyntax
        }

VarBindList ::=
        SEQUENCE OF
           VarBind

END
```

```
        -- the IP routing table

        -- The IP routing table contains an entry for each route
        -- presently known to this entity.

        ipRouteTable OBJECT-TYPE
            SYNTAX  SEQUENCE OF IpRouteEntry
            ACCESS  not-accessible
            STATUS  mandatory
            DESCRIPTION
                    "This entity's IP Routing table."
            ::= { ip 21 }

        ipRouteEntry OBJECT-TYPE
            SYNTAX  IpRouteEntry
            ACCESS  not-accessible
            STATUS  mandatory
            DESCRIPTION
                    "A route to a particular destination."
            INDEX   { ipRouteDest }
            ::= { ipRouteTable 1 }

        IpRouteEntry ::=
            SEQUENCE {
                ipRouteDest
                    IpAddress,
                ipRouteIfIndex
                    INTEGER,
                ipRouteMetric1
                    INTEGER,
                ipRouteMetric2
                    INTEGER,
                ipRouteMetric3
                    INTEGER,
                ipRouteMetric4
                    INTEGER,
                ipRouteNextHop
                    IpAddress,
                ipRouteType
                    INTEGER,
                ipRouteProto
                    INTEGER,
                ipRouteAge
                    INTEGER,
                ipRouteMask
                    IpAddress,
                ipRouteMetric5
                    INTEGER,
                ipRouteInfo
                    OBJECT IDENTIFIER
            }

        ipRouteDest OBJECT-TYPE
            SYNTAX  IpAddress
            ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
                    "The destination IP address of this route.  An
                    entry with a value of 0.0.0.0 is considered a
                    default route.  Multiple routes to a single
                    destination can appear in the table, but access to
                    such multiple entries is dependent on the table-
                    access mechanisms defined by the network
                    management protocol in use."
            ::= { ipRouteEntry 1 }

        ipRouteIfIndex OBJECT-TYPE
            SYNTAX  INTEGER
            ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
                    "The index value which uniquely identifies the
                    local interface through which the next hop of this
                    route should be reached.  The interface identified
```

```
            by a particular value of this index is the same
            interface as identified by the same value of
            ifIndex."
    ::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The primary routing metric for this route.  The
            semantics of this metric are determined by the
            routing-protocol specified in the route's
            ipRouteProto value.  If this metric is not used,
            its value should be set to -1."
    ::= { ipRouteEntry 3 }

ipRouteMetric2 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "An alternate routing metric for this route.  The
            semantics of this metric are determined by the
            routing-protocol specified in the route's
            ipRouteProto value.  If this metric is not used,
            its value should be set to -1."
    ::= { ipRouteEntry 4 }

ipRouteMetric3 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "An alternate routing metric for this route.  The
            semantics of this metric are determined by the
            routing-protocol specified in the route's
            ipRouteProto value.  If this metric is not used,
            its value should be set to -1."
    ::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "An alternate routing metric for this route.  The
            semantics of this metric are determined by the
            routing-protocol specified in the route's
            ipRouteProto value.  If this metric is not used,
            its value should be set to -1."
    ::= { ipRouteEntry 6 }

ipRouteNextHop OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The IP address of the next hop of this route.
            (In the case of a route bound to an interface
            which is realized via a broadcast media, the value
            of this field is the agent's IP address on that
            interface.)"
    ::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),        -- none of the following

                invalid(2),      -- an invalidated route
                                 -- route to directly
                direct(3),       -- connected (sub-)network

                                 -- route to a non-local
                indirect(4)      -- host/network/sub-network
            }
```

```
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The type of route.  Note that the values
            direct(3) and indirect(4) refer to the notion of
            direct and indirect routing in the IP
            architecture.

            Setting this object to the value invalid(2) has
            the effect of invalidating the corresponding entry
            in the ipRouteTable object.  That is, it
            effectively dissasociates the destination
            identified with said entry from the route
            identified with said entry.  It is an
            implementation-specific matter as to whether the
            agent removes an invalidated entry from the table.
            Accordingly, management stations must be prepared
            to receive tabular information from agents that
            corresponds to entries not currently in use.
            Proper interpretation of such entries requires
            examination of the relevant ipRouteType object."
    ::= { ipRouteEntry 8 }

ipRouteProto OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),       -- none of the following

                                -- non-protocol information,
                                -- e.g., manually configured
                local(2),       -- entries

                                -- set via a network
                netmgmt(3),     -- management protocol

                                -- obtained via ICMP,
                icmp(4),        -- e.g., Redirect

                                -- the remaining values are
                                -- all gateway routing
                                -- protocols
                egp(5),
                ggp(6),
                hello(7),
                rip(8),
                is-is(9),
                es-is(10),
                ciscoIgrp(11),
                bbnSpfIgp(12),
                ospf(13),
                bgp(14)
            }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The routing mechanism via which this route was
            learned.  Inclusion of values for gateway routing
            protocols is not intended to imply that hosts
            should support those protocols."
    ::= { ipRouteEntry 9 }

ipRouteAge OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The number of seconds since this route was last
            updated or otherwise determined to be correct.
            Note that no semantics of `too old' can be implied
            except through knowledge of the routing protocol
            by which the route was learned."
    ::= { ipRouteEntry 10 }

ipRouteMask OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
```

```
    DESCRIPTION
            "Indicate the mask to be logical-ANDed with the
            destination address before being compared to the
            value in the ipRouteDest field.  For those systems
            that do not support arbitrary subnet masks, an
            agent constructs the value of the ipRouteMask by
            determining whether the value of the correspondent
            ipRouteDest field belong to a class-A, B, or C
            network, and then using one of:

                 mask           network
                 255.0.0.0      class-A
                 255.255.0.0    class-B
                 255.255.255.0  class-C

            If the value of the ipRouteDest is 0.0.0.0 (a
            default route), then the mask value is also
            0.0.0.0.  It should be noted that all IP routing
            subsystems implicitly use this mechanism."
    ::= { ipRouteEntry 11 }

ipRouteMetric5 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "An alternate routing metric for this route.  The
            semantics of this metric are determined by the
            routing-protocol specified in the route's
            ipRouteProto value.  If this metric is not used,
            its value should be set to -1."
    ::= { ipRouteEntry 12 }

ipRouteInfo OBJECT-TYPE
    SYNTAX  OBJECT IDENTIFIER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "A reference to MIB definitions specific to the
            particular routing protocol which is responsible
            for this route, as determined by the value
            specified in the route's ipRouteProto value.  If
            this information is not present, its value should
            be set to the OBJECT IDENTIFIER { 0 0 }, which is
            a syntatically valid object identifier, and any
            conformant implementation of ASN.1 and BER must be
            able to generate and recognize this value."
    ::= { ipRouteEntry 13 }
```