NTNU
Norges teknisk-naturvitenskapelige universitet
Institutt for telematikk

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

Contact person / **Faglig kontakt**:	Mazen Shiaa
Tlf.:	452 76 156
Date / **dato**:	05.06.2007
Time / **tid**:	15:00-18:00
Remedies / **Tillatte hjelpemidler**:	**D**: No printed or handwritten remedies allowed. **D**: Ingen trykte eller håndskrevne hjelpemidler tillatt.
Language / **Språkform**:	English / Norsk (Bokmål) (Den engelske oppgaveteksten er den originale og gyldige teksten.)
Results / Sensurdato:	13.06.2007

English

Question 1. (10%)

A host with an IPv4 address of (TLV):
01000000 00000100 00010100 00010100 00010100 00010100
Assume the subnet of this host uses a subnet mask of 21 bits:

a) What is the subnet address?

The IP address is 20.20.20.20, which gives a 20.20.16.0 subnet address.

b) What is the maximum number of hosts in this subnet?

$2^{11} – 2 = 2046$ (out of possible 11 bits for host addresses two addresses are reserved for the subnet address and the broadcasting address)

Question 2. (10%)

Assume an snmptable command generates the following result:

snmptable -v 2c -c public snmp-server.org 1.3.6.1.2.1.65.3.8

rsIndex	*rsAddress*	*rsType*	*rsAccess*
1	*129.241.20.1*	*Types.1*	*readWrite*
2	*129.241.20.2*	*Types.2*	*readOnly*
3	*129.241.20.3*	*Types.3*	*readWrite*
4	*129.241.20.4*	*Types.4*	*readOnly*
5	*129.241.20.5*	*Types.5*	*readWrite*

a) What is the OID of the third column?

rsType ::= { 1.3.6.1.2.1.65.3.8.tableEntry 3 }

Where *tableEntry* is the OID of the table entry object. Possible definition of tableEntry:

tableEntry OBJECT-TYPE
SYNTAX TableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Entry of this table"
INDEX { rsIndex } -- this may be extended to include e.g. rsAddress as another index
::= { 1.3.6.1.2.1.65.3.8 1 } -- 1 is just an example

TableEntry ::= SEQUENCE {
rsIndex INTEGER,
rsAddress IpAddress,
rsType RSType,
rsAccessRSAccess}

(if you don't define the tableEntry object as indicated above, and don't define the sequence of the columnar objects in the TableEntry then you can't index the rsType column as 1.3.6.1.2.1.65.3.8.3 or 1.3.6.1.2.1.65.3.8.1.3 – these are considered wrong answers)

b) What is the OID of the entry in the 2nd row and 3rd column? Explain why?

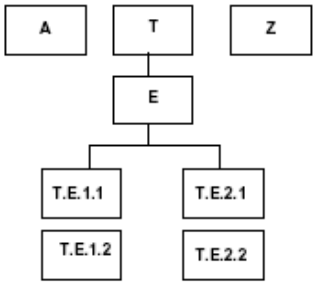
As it has not been indicated how to index this table, the answer must start with choosing an indexing method. There can be several choices for indexing, here we show two cases:

(1) using rsIndex as INDEX: 1.3.6.1.2.1.65.3.8.tableEntry.3.2
(2) using rsIndex and rsAddress as INDEX: 1.3.6.1.2.1.65.3.8.tableEntry.3.2.129.241.20.2

where *tableEntry* is the OID of the table entry object (as defined above). Note that based on your choice for the indexing method you should apply lexicographic ordering – in these two cases the rows are properly ordered.

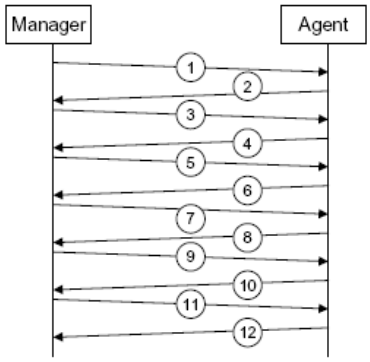
Question 3. (15%)

Assume the following MIB tree:



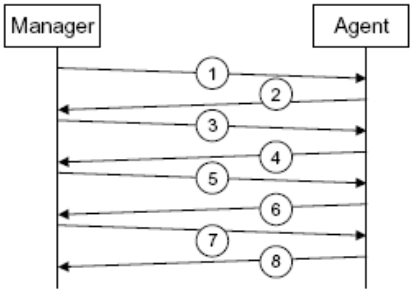
In the following 3 cases the manager will retrieve all the entries of this tree by polling the agent using different SNMP requests and possibly different parameters.

a) Assume you only know that the MIB tree starts at node 'A', what are the SNMP requests 1-12 and their parameters in the following diagram?



1. get-request (A)
2. get-response (A)
3. get-next-request (A)
4. get-response (T.E.1.1)
5. get-next-request (T.E.1.1)
6. get-response (T.E.1.2)
7. get-next-request (T.E.1.2)
8. get-response (T.E.2.1)
9. get-next-request (T.E.2.1)
10. get-response (T.E.2.2)
11. get-next-request (T.E.2.2)
12. get-response (Z)

b) Assume you know the MIB tree starts at node ‘A’ and you know the table structure, what are the SNMP requests 1-8 and their parameters in the following diagram?



Possibility (A)

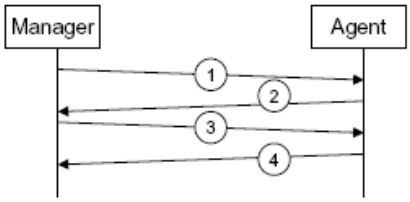
1. get-request (A)
2. get-response (A)
3. get-next-request (A)
4. get-response (T.E.1.1)
5. get-next-request (T.E.1.1, T.E.1.2, T.E.2.1)
6. get-response (T.E.1.2, T.E.2.1, T.E.2.2)
7. get-next-request (T.E.2.2)
8. get-response (Z)

Possibility (B)

1. get-request (A)
2. get-response (A)
3. get-next-request (A)
4. get-response (Z)
5. get-next-request (Z)
6. get-response (T.E.1.1)
7. get-next-request (T.E.1.1, T.E.1.2, T.E.2.1)
8. get-response (T.E.1.2, T.E.2.1, T.E.2.2)

(Note: you only know the start of the tree, which is ‘A’ and the structure of the table, which means the number of columns and rows. However, you don’t really know what comes after A in the tree – it could either be the table (possibility A) or node ‘Z’ (possibility B). Therefore at number (3) you have to send a get-next-request (A). As there is no indication what type of SNMP versions or requests you might use, there are other possible answers to this question, e.g. using a get-bulk-request at the start or at the middle and reasoning about the response – but of course using all the 8 messages as indicated in the diagram.)

c) Assume you know the MIB tree starts at node ‘A’ and you know the table structure, what are the SNMP requests 1-4 and their parameters in the following diagram?



Possibility (A)

1. get-request (A)
2. get-response (A)
3. get-bulk-request (0, 5, A)
4. get-response (T.E.1.1, T.E.1.2, T.E.2.1, T.E.2.2, Z)

(Note: this is a general and the most intuitive way to retrieve the elements of this tree. Actually you might use any number larger than 5 (which comes from the 4 cells of the table and the other node) as well, which will return "endOfMibView" at the end of your response. Also note that this possibility is valid independently of the order of the table and node Z – this means the last message could be get-response (Z, T.E.1.1, T.E.1.2, T.E.2.1, T.E.2.2))

Possibility (B)

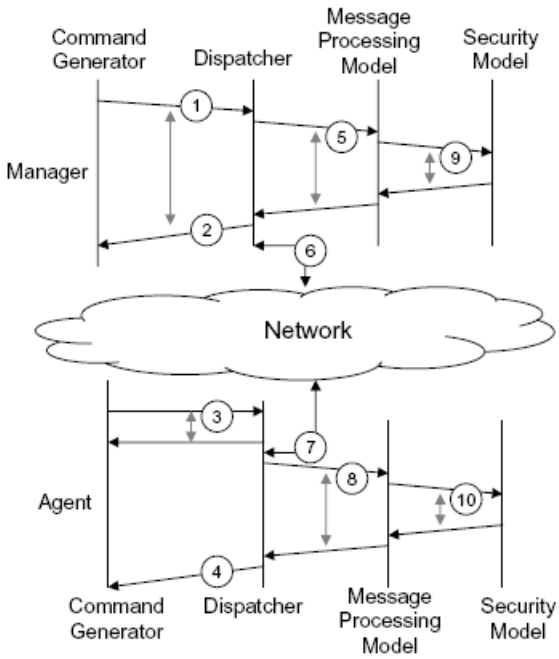
1. get-bulk-request (1, 3, A, T.E.1.1, T.E.2.1)
2. get-response (A, T.E.1.1, T.E.2.1, T.E.1.2, T.E.2.2, "endOfMibView")
3. get-next-request (A)
4. get-response (Z)

(Note: you only know the start of the tree, which is 'A' and the structure of the table, which means the number of columns and rows. However, you don't really know what comes after A in the tree – it could either be the table or node 'Z'. In this case the only possible interpretation is that node Z is coming after node A. Therefore at number (1) you have to send a get-bulk-request with a number of repetitions equal to 3 to get to know what is following the table – mind the behaviour and the interpretation of a get-bulk-request in the SNMP protocol. Assuming that the table comes after node A the request in number (1) would generate the following response: get-response (A, T.E.1.1, T.E.2.1, T.E.1.2, T.E.2.2, Z, "endOfMibView"), which contains all the nodes of our MIB tree – and this is not our case in this diagram.)

Question 4. (20%)

The following figure shows a generalized time-sequenced operation for get request message going from manager to an agent. Identify where the get request is sent and received as well as the primitives exchanged between the different modules (marked 1-10).

(In this question choose among the following primitives: encryptData, decryptData, generateResponseMsg, generateRequestMsg, processIncomingMsg, prepareDataElements, returnResponsePdu, sendPdu, processPdu, prepareOutgoingMsg, authenticateOutgoingMsg, authenticateIncomingMsg, registerContextID, sendPduHandle, processIncomingMsg, isAccessAllowed)



1. sendPdu
2. sendPduHandle

3. registerContextEngineID
4. processPdu
5. prepareOutgoingMessage
6. send get-request message
7. receive get-request message
8. prepareDataElements
9. generateRequestMsg
10. processIncomingMsg

(Note: in the agent part the “Command Generator” is actually a “Command Responder”, however this does not affect the overall solution as it has been directly and specifically stated in the question that only a get-request message to be sent from the manager to an agent without showing the response)

Question 5. (20%)

a) Explain two ways to describe XML tags, elements and their relations?

(1) DTD and (2) XML-Schema

(1) DTD: the purpose of Document Type Definition (DTD) is to define the legal building blocks of an XML document and the document structure with a list of legal elements. A DTD can be declared inline in *your XML document*, or as *an external reference*.
Internal DOCTYPE declaration: wrapped in a DOCTYPE definition with the following syntax: <!DOCTYPE root-element [element-declarations]>
External DOCTYPE declaration: wrapped in a DOCTYPE definition with the following syntax: <!DOCTYPE root-element SYSTEM "filename">
Example: use <!DOCTYPE note SYSTEM "note.dtd"> in an XML document to refer to the following note.dtd file:

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

(2) XML-Schema: XML Schema is an XML based alternative to DTD. The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD. An XML schema describes the structure of an XML document. The XML Schema language is also referred to as **XML Schema Definition (XSD).** XML Schema defines: **elements, attributes, child elements,** the **order** of child elements, the number of child elements, whether an element is **empty** or **can include text, data types** for elements and attributes, and **default** and **fixed** values for elements and attributes
Example:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified"><xs:element name="note">
   <xs:complexType>
     <xs:sequence>
               <xs:element name="to" type="xs:string"/>
               <xs:element name="from" type="xs:string"/>
               <xs:element name="heading" type="xs:string"/>
               <xs:element name="body" type="xs:string"/>
     </xs:sequence>
   </xs:complexType>
</xs:element></xs:schema>
```

b) List three ontology languages and explain the differences between them?

(1) RDF, (2) DAML+OIL, (3) OWL

(1) RDF: RDF stands for **Resource Description Framework** and it is a datamodel for objects ("resources") and relations between them. It provides a simple semantics for this datamodel. These datamodels can be represented in an XML syntax. RDF uses **Web identifiers (URIs)** to identify resources. The following RDF document could describe the resource "http://www.w3schools.com/RDF":

```
<?xml version="1.0"?><RDF>
  <Description about="http://www.w3schools.com/RDF">
    <author>Jan Egil Refsnes</author>
    <homepage>http://www.w3schools.com</homepage>
  </Description>
</RDF>
```

(2) DAML+OIL: a language for expressing far more sophisticated classifications and properties of resources than RDF and providing facilities for data typing based on the type definitions provided in the W3C XML Schema Definition Language (XSDL). It allows one to express classifications by inference rather than by explicitly listing which resources go into which buckets. It adds **primitives**, like *DatatypeProperty*, as needed, and it defers to RDFS otherwise. There are some changes to the semantics of *rdfs:domain* and *rdfs:range* in DAML+OIL systems: the most important of which is that a **property** can have **multiple ranges**.

(3) OWL: derived from DAML+OIL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL has more facilities for **expressing meaning and semantics** than XML, RDF, and RDF-S, and higher ability to **represent machine interpretable content** on the Web. OWL is a **revision of the DAML+OIL** web ontology language incorporating lessons learned from the design and application of DAML+OIL.

c) What are the parts of a UDDI entry and what are they used for?

There are three parts of an UDDI entry.

a. White pages
b. Yellow pages
c. Green pages

The White papers describe the company offering the service: name, address, telephone number and other contact information. The yellow pages include information about the type of business and industry category. Each business service entry represents a family of technical services, i.e. it shows the services a certain business provides. The Green pages contain information about the specific services offered. It should describe the interface to the service in enough detail for someone to write an application to use the Web Service. Often, a WSDL file that describes a SOAP interface to the XML Web Service is contained, but other kind of services can in principle also be described.

Question 6. (25%)

Answer the following statements by indicating TRUE or FALSE (no explanation is required):

1) In ASN.1, an object name is a unique name for the leaf node of an OBJECT IDENTIFIER.

TRUE

2) SNMPv1 protocol uses only UDP as transport protocol

TRUE

3) SNMP access policy is the pairing of the SNMP MIB view with the SNMP access mode

FALSE

4) SMIv2 introduced MODULE-IDENTITY, OBJECT-TYPE, and NOTIFICATION-TYPE

TRUE

5) RowStatus is an enumerated Integer used by the manager when creating and deleting rows

TRUE

6) Trap and get-request messages have the same message format in SNMPv1

FALSE

7) In SNMPv3 notification originator sends notifications to the manager

FALSE

8) Abstract service interface is used to provide access control

FALSE

9) The OSF function block of the TMN executes in the same telecommunication network that it manages

FALSE

10) SOA is a management module used to create web services

FALSE

(Don’t answer randomly as there will be discount points for wrong answers. The following algorithm will be applied in correcting this question:

- every good answer is given 1 point,
- unanswered statements are not counted,
- single wrong answer is given 0 point,
- any further wrong answer is given -1 point,
- and the overall point count will provide the weight for this question)

NORSK

Oppgave 1. (10%)
En host som har følgende IPv4 adress (TLV):
01000000 00000100 00010100 00010100 00010100 00010100
Anta at subnettet hvor denne hosten befinner seg bruker 21 bits subnetmaske:

a) Hva er adressen til dette subnettet?
b) Hva er maksimalt antall noder i dette subnettet?

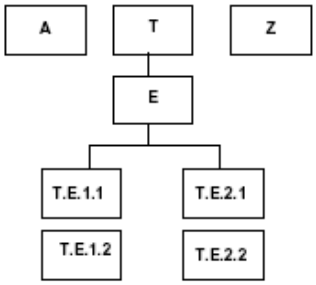
Oppgave 2. (10%)
Anta en snmptable kommando som generer følgende resultat:

snmptable -v 2c -c public snmp-server.org 1.3.6.1.2.1.65.3.8

rsIndex	*rsAddress*	*rsType*	*rsAccess*
1	*129.241.20.1*	*Types.1*	*readWrite*
2	*129.241.20.2*	*Types.2*	*readOnly*
3	*129.241.20.3*	*Types.3*	*readWrite*
4	*129.241.20.4*	*Types.4*	*readOnly*
5	*129.241.20.5*	*Types.5*	*readWrite*

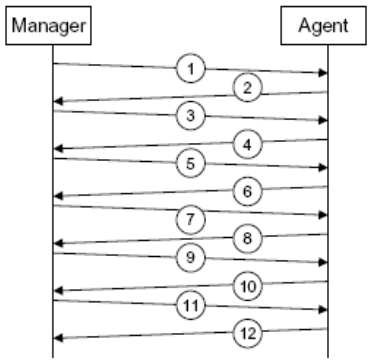
a) Hva er OID av tredje kolonnen?
b) Hva er OID av objektet i andre rad og tredje kolonne? Forklar hvorfor?

Oppgave 3. (15%)
Anta følgende "MIB tree":

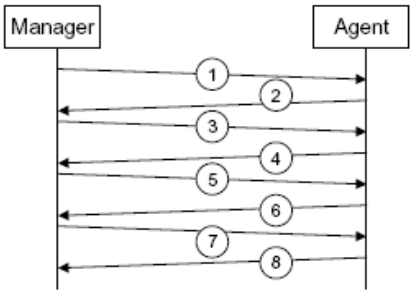


I de følgende 3 tilfellene vil manageren få verdiene til nodene av dette "MIB tree" ved å bruke forskjellige SNMP meldinger og muligens forskjellige parametere.

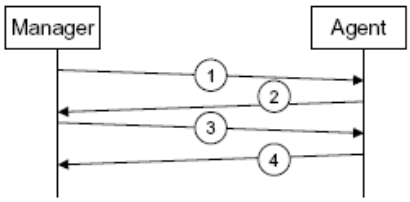
a) Anta du vet at "MIB tree" starter ved node 'A', tegn SNMP meldingene 1-12 og deres parametere i figuren under?



b) Anta du vet at "MIB tree" starter ved node 'A' og du kjenner strukturen til tabellen, tegn SNMP meldingene 1-8 og deres parametere i figuren under?



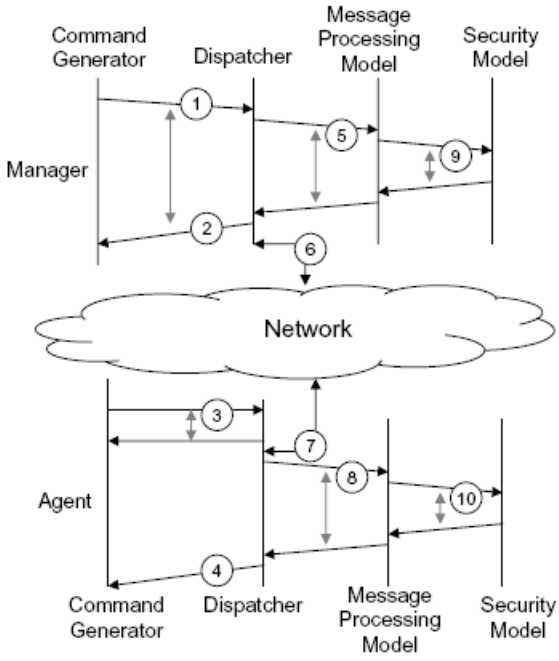
c) Anta du vet at "MIB tree" starter ved node 'A' og du kjenner strukturen til tabellen, tegn SNMP meldingene 1-4 og deres parametere i figuren under?



Oppgave 4. (20%)

Følgende figuren viser en generell "time-sequenced" operasjon for "get request" melding som går fra en manager til en agent. Identifiser hvor "get request" bli sent og mottat, og samtidig de primitivene som utvekles mellom de forskjellige modulene (tegnet 1-10).

(I dette spørsmålet velg blant følgende primitiver: encryptData, decryptData, generateResponseMsg, generateRequestMsg, processIncomingMsg, prepareDataElements, returnResponsePdu, sendPdu, processPdu, prepareOutgoingMsg, authenticateOutgoingMsg, authenticateIncomingMsg, registerContextID, sendPduHandle, processIncomingMsg, isAccessAllowed)



Oppgave 5. (20%)

a) Forklar to måter for å beskrive XML tags, elementer og sammenhengen mellom dem?
b) List tre ontologi språk og forklar forskjellen mellom dem?
c) Hva er innholdet i en ”UDDI entry” og hva blir de brukt til?

Oppgave 6. (25%)
Svar følgende ved å skrive TRUE eller FALSE (ingen forklaring er kreves):

1) I ASN.1, et “object name” er et unikt navn for den ”leaf node” av en OBJECT IDENTIFIER.
2) SNMPv1 protokollen bruker kun UDP som transport protokoll
3) ”SNMP access policy” er kombinasjon av “SNMP MIB view” med “SNMP access mode”
4) SMIv2 introduserer MODULE-IDENTITY, OBJECT-TYPE, and NOTIFICATION-TYPE
5) RowStatus er en “enumerated Integer” som brukes av manageren ved oppretting og fjerning av rader
6) Trap og get-request meldinger har same melding format i SNMPv1
7) I SNMPv3 “notification originator” sender notifikasjoner til manageren
8) “Abstract service interface” er brukt for å oppnå access control
9) Den “OSF function block” av TMN eksekverer i det samme telekommunikasjon nettverk som den styrer
10) SOA er en “management module” som brukes for å lage “web services”

(følgende algoritm ville bli brukt I dene oppgaven:

- hver riktig svar gir 1 poeng,
- ubesvarte spørsmål teller ikke,
- et feil svar gir 0 poeng,
- ethvert feil svar I tilleg gir -1 poeng,
- og total poengsumm vil være basis for vekting av oppgaven)