Network Management (NETW-1001)

Dr. Mohamed Abdelwahab Saleh

IET-Networks, GUC

Spring 2017
1. Architecture of NMSs
2. OSI Network Management
3. Telecom Management Network
4. SNMP
5. SMI and MIB
Remote Management

- A Network Management System (NMS) should implement the “FCAPS” management functions, but how?
- What are the functional and physical components of a NMS?
- This is addressed by several standards. We will examine the management framework for the Open Systems Interconnection (OSI) concept and the one for the Telecommunications Management Network (TMN).
- Both frameworks require that management tasks for network devices can be performed remotely. In other words, management information, such as device configuration, can be transmitted to/from devices in order to collect data or perform operations.
OSI Network Management Concepts

The OSI network management model differentiates between:
1. Management system.
   It is comprised of managers, which are processes that monitor and perform operations on managed network elements.
2. Managed objects.
   These are network elements, they represent networking components/resources to be managed, e.g., a switch or an interface on a router. They are characterized by their:
   - Attributes that distinguish them.
   - Notifications they may send to managers.
   - Operations they accept from managers.
   - Communications with other managed objects.
OSI Network Management and Layers

- A set of managed objects + their attributes is called the “Management Information Base” (MIB).
- Agents and managers communicate using protocols and can reside in different open systems or the same system.
- In the case of “system management” application layer protocols are used.
- Sometimes, N-layer protocols are used. In this case, this is called N-layer management, since any entity at layer N+1 or above will not be managed.
- A protocol should distinguish between management information and other data that it may transfer.
A Simple Example

- Routers used for small home networks can be configured using a Web interface.
- This means that the router runs a web server, i.e., implements an HTTP server.
- This situation is depicted in the figure below:

```
HTTP client, e.g. FireFox  HTTP protocol  Router running HTTP server
```

Usually you can access the router configuration page by going to 192.168.0.0.

The manager in this case is you/your HTTP client, the agent is the HTTP server running in the router. Note that the router is a layer-3 device, however the HTTP server is implemented just for management purposes.
The OSI network management model is characterized by:

- The managing system contain processes called managers, while the managed system has agents.
- Managers and agents communicate via protocols, usually at the application layer.
- An agent executes received operations and sends notifications.
- The actual network resources to be managed are called managed objects.
- A set of managed objects and their attributes is called a Management Information Base (MIB).
- The management functions are the FCAPS functions.
Telecommunications Management Network

The Telecommunication Management Network (TMN) comprises a set of principles for the management of telecommunication networks.

- One of the main features is the separation between the network to be managed, i.e., network of telecommunication equipment, and the network of management systems.
- The management is therefore not centralized but takes place in a cooperative manner between managing systems.
- The main motivation is to facilitate the remote and automatic management of telecommunications equipment using standard interfaces.

TMN defines 3 architectures according to point of view:

- Functional architecture.
- Information architecture.
- Physical architecture.
Notice the following:

- Managed telecom. equipment in the telecom. network are called Network Elements (NE).
- TMN contains the DCN, Operations Systems (OS), and defines interfaces to the Work Station (WS) and NEs, hence the boundaries of the dashed rectangle.
The functional architecture of TMN distinguishes between:

- **Function blocks**: These are the smallest building blocks of TMN functionalities, i.e., they are atomic units that perform a certain service or a certain collection of services.

- **Management Application Functions (MAFs)**: These are the management services offered by a TMN system and they are implemented in function blocks.

- **TMN Management Functions**: They are used for interaction between different MAFs in different function blocks. A set of these functions may be implemented in a single block.

- **Reference points**: They describe a function block’s interface, i.e., its TMN management functions that are exposed for interactions with other function blocks.
The following function blocks are defined:

- **Network Element Function (NEF):** Provides the ability to a managed network entity to be monitored and controlled.
- **Operation Systems Function (OSF):** Provides the management capabilities, i.e., these are the “brain” blocks of the system taking care of the management tasks.
- **Transformation Function (TF):** Provides the capability to “translate” between entities using different protocols of different information models.
- **Workstation Function (WSF):** It provides an interface to human users.

Moreover, the following classes of reference points are defined:

- **q Class:** Between OSF, TF and NEF.
- **f Class:** Between OSF and a WSF.
- **x Class:** Between OSFs of two TMNs or between the OSF of a TMN and its equivalent functionality of another network.
During the exchange of management information, the following roles can be identified (similar to the OSI model):

- The managing role.
- The managed role.

Moreover, the exchanged information is structured according to *information models* which contain *information elements*. An information element contains the attributes of a certain managed network resource.
The physical architecture describes actual network devices:

- **Operations System (OS):** Performs the operations system function.

- **Transformation devices:** Perform transformation functions, they are:  
  - **Adaptation device:** Allows the connection of non-TMN equipment to a TMN.  
  - **Mediation device:** Allows the connection of two TMN-enabled devices that use different communication mechanisms. Transformation devices are also categorized according to the reference point (interface) of their connection (q, f, or x).

- **Network Element (NE):** This is a piece of telecommunication equipment that can be monitored and controlled. It implements network element functions.

- **Workstation (WS):** It provides an interface for human interaction and therefore implements work station functions.
Across the f, q, and x reference points, information elements are exchanged.

**Figure**: Overview of TMN showing physical/functional architectures
SNMP – Overview

What does an agent do? ⇒ Monitor and control objects.
What’s an object? ⇒ Something we need to monitor/configure. It can be a physical network element or any useful information about a device. It can be the temperature of switch, the status of a port in a router, etc.

A Management Information Base (MIB) is a collection of objects.
The Simple Network Management Protocol (SNMP) implements the ideas in the OSI model.

Two roles exist: The manager and the agent. They exchange five types of messages that are listed below. The first three are from manager to agent, and the last two are from agent to manager.

- **GET**: Manager asks for a variable.
- **GET_NEXT**: Manager asks for a variable.
- **SET**: Manager changes the value of a variable.
- **GET_RESPONSE**: Agent replies to a GET request.
- **TRAP**: Agent notifies manager with an event.

**Why is it called simple?**

⇒ Small number of types of messages.
⇒ It is based on a connectionless network service.
Objects and MIBs

- The database of objects monitored by an agent is called a Management Information Base (MIB).
- So, how to describe an MIB? Or, how do we define objects?
- An MIB is described in a text file, whose syntax follows rules defined by a standard: The Structure of Management Information (SMI), which is an adapted subset of ASN.1.
- The MIB file defines the objects that we can monitor or configure, i.e., it is a “declaration” of what the snmp agent can do.
- Knowing the MIB that an agent supports, the NMS can “talk” to the agent —using a protocol like SNMP— without worrying about how the agent does its job.
- All agents must implement a standard MIB called MIB-II.
The following are general issues that apply to SNMP or, in general, to any NMS:

- **Automation of system responses/services**, how autonomous is the system?
  - Reactive: System notifies you, you decide on your own.
  - Interactive: System notifies you and helps you to decide.
  - Proactive: System notifies you and may initiate some action on its own.

- **How to know the real cause of problems**; 10 alarms caused by one fault.

- **Correlation of fault logs**.

- **Security**: Who manages what?
SNMP Implementation

- SNMP is an application layer protocol.
- Relies on UDP.
- SNMP PDUs are bit-oriented, i.e., it is a binary protocol (not a character protocol).
- UDP is unreliable but has low overhead.
- Follows the system network management concept of OSI, since it is at the application layer.
SNMP GET and SET

Informal definition
The NMS wants to the agent to get/set the value of a certain managed object

So, which managed object and what information?
⇒ The object has to be one that is listed in the MIB implemented by the agent.

Informal Syntax
GET(RequestID request-ID,
    ErrorStatus error-status,
    ErrorIndex error-index,
    VarBindList variable-bindings)
## GET and SET Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request-ID</td>
<td>RequestID</td>
<td>An integer identifying the request.</td>
</tr>
<tr>
<td>error-status</td>
<td>ErrorStatus</td>
<td>An enumeration of error codes $\in {0, 1, 2, 3, 4, 5}$. In GET, it is always 0.</td>
</tr>
<tr>
<td>error-index</td>
<td>ErrorIndex</td>
<td>A number giving information about the error, if any.</td>
</tr>
<tr>
<td>variable-bindings</td>
<td>VarBindList</td>
<td>A variable is an <em>instance</em> of a managed object. A variable binding is a variable name along with its value. A variable binding list is a sequence of variable bindings. The value is irrelevant in a GET.</td>
</tr>
</tbody>
</table>
Variable Bindings

Variable name + variable value

A variable is an instance of a managed object. The object is a concept (a type or a class), whereas the instance (variable) is the actual entity whose value is of interest. So, an object may have multiple instances:

- Structure of Management Information (SMI) defines the syntax used to specify objects in MIB files.
- To define an object, we need three pieces of information:
  - Name or Object ID (OID) To identify the object
  - Type or syntax What sort of values can this object have? Is it an integer, a string, a structure, etc.
  - Encoding How is the object represented at the byte level
An object is any network element, e.g., a routing table. An Object ID (OID) is a numeric name of an object.

An OID is a sequence of integers separated by dots, i.e., It has the form X.Y.Z . . .

A collection of OIDs always has a tree structure, an OID specifies a node in this tree.

Given the tree of OIDs, an OID is obtained by following the path leading to the OID starting at the root.

The OID of an instance is the OID of its object X, followed by another sequence of numbers, e.g., X.0.

The hierarchical naming scheme is useful for providing a naming structure and avoiding name collisions.

The Structure of Management Information (SMI) standard defines a tree for OIDs, shown on the next slide.
The figure is taken from the book “Essential SNMP”, 2nd Edition By Douglas Mauro, Kevin Schmidt. To get the OID of the object “SysLocation” we follow the path leading to the object from the root and note the numbers. The path is shown in blue. OID of SysLocation is .1.3.6.1.2.1.1.6, an instance of SysLocation will have OID = .1.3.6.1.2.1.1.6.0
OIDs identify objects and follow the tree-like hierarchical naming convention explained in the previous slide.

We need to know what sort of values can an object have.

This is defined by the object type.

Object types (as defined by SMI can be:

- **INTEGER**: An integer.
- **OCTETSTRING**: A string of bytes.
- **SEQUENCE**: A sequence of other types.
- **IpADDRESS**: An IP address.
- **etc.**
So, knowing the SMI syntax, how do we define objects?

**Objects are defined in an MIB.** The most basic syntax for defining objects is the following:

```plaintext
object_name OBJECT-TYPE
SYNTAX The type
ACCESS Read-only, read-write, not-accessible, ...
STATUS To indicate if this definition is current, obsolete, etc.
DESCRIPTION Text describing the object
::={parent_node index}
```

Words in all upper-case letters are fixed keywords in the definitions of all objects.
An interface table:

```plaintext
ifTable OBJECT-TYPE
SYNTAX     SEQUENCE OF IfEntry
ACCESS     not-accessible
STATUS     mandatory
DESCRIPTION An IfTable is a sequence of Ifentries
 ::= { interfaces    2 }
```

Here the parent node is interfaces whose ID is .1.3.6.1.2.1.2, the definition of ifTable means that it is a child of the interfaces node and its index is 2. Therefore the OID for ifTable is .1.3.6.1.2.1.2.2
Security

- SNMP defines communities, which are classes of users (of managers).
- Three communities are defined:
  - read-only can only read values of objects.
  - read-write Can read and write values objects.
  - trap can only receive traps from agents.
- Community strings are passwords of communities.
- A community string provides access privileges corresponding to a community.
The NMS is aware of objects through the MIBs that are loaded in it.

It sends a GET request, by the manager to the agent asking for a specific variable binding.

The agent checks its MIBs for the OID.

If there is no match an error is generated and an error code is sent.

If there is a match a response is issued and sent to the manager.