An Introduction to the SNMP Protocol: The MIB Tree

1 Background:

The Net SNMP programs are available from `http://net-snmp.sourceforge.net/`. They implement the SNMP protocol versions 1, 2 and 3. They also provide an SNMP library, which you can use in applications, written in a number of languages. There is also a Perl binding to this library, called `SNMP`.

These tools are useful particularly since they implement aspects of the protocol in a simple, understandable way, and are ideal for learning about the SNMP protocol.

1.1 The MIB Tree

We use SNMP to get values from, and to set values in *managed objects*. These *managed objects* are arranged so that each has an *Object ID*, or OID. The OID has two representations: a series of numbers separated by dots, or a series of names, also separated by dots. These OIDs are arranged in a tree structure, called the *MIB tree*. The abbreviation MIB stands for *Management Information Base*. Figure 1 on the following page shows a small part of the tree. The OIDs are a little bit like an IP address in that they uniquely identify a managed object on a particular *agent* (see section 1.3).

1.2 The MIB Files

The MIB tree structure is described in a set of files on your hard disk, in the directory `/usr/share/snmp/mibs`. We will study the structure of these MIB files in later exercises. Each of these MIB files contains descriptions of a set of managed objects that an agent can provide. The tool `snmptranslate` can query the information in these files, and can show information about the MIB tree.

1.3 Agent and Manager

Figure 2 on the following page shows how a *manager* (the network management station, or NMS) sends a request to the *agent* running on the managed device (say, a router, or switch, or server), and how the agent sends back a response. The request contains one or more object IDs (OIDs). The agent sends back the information relating to the OIDs.

2 Procedure:

You can either boot your own copy of Linux, or boot the Linux that is installed on the internal hard disk of the computers in the laboratory for this exercise, since we are not modifying any SNMP objects today.

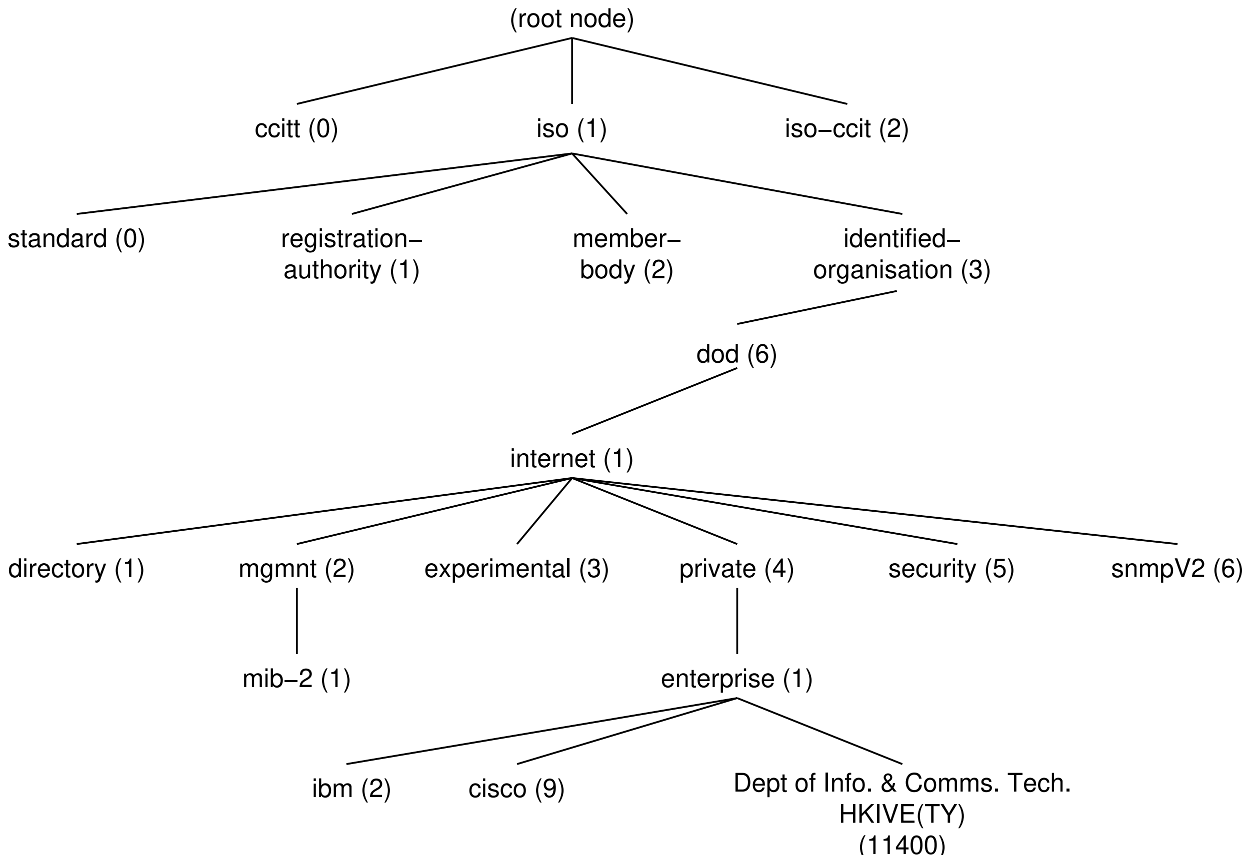

Figure 1: The MIB tree.

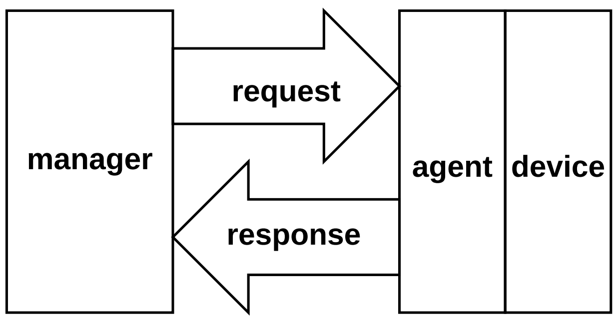

Figure 2: A manager sends an SNMP request to an agent, which responds to the manager.

2.1 Learning about the SNMP MIB Tree with `snmptranslate`

1. Open a command prompt, and type:

   $ snmptranslate .1.3.6.1.2.1.1.3.0

   You will see the text form of this object ID (OID).

2. `snmptranslate` can also translate into numerical form as well, by adding the `-On` flag to its options (that is the letter "Oh", not a zero). Type this at your command prompt:

   $ snmptranslate SNMPv2-MIB::system.sysUpTime.0

3. You can mix text and numbers when you describe an OID; the `-On` flag just *toggles* which type of output is displayed. Compare the output of each of:

```
$ snmptranslate .iso.3.6.1.private.enterprises.2021.2.1.prNames.0
```

and

```
$ snmptranslate -On .iso.3.6.1.private.enterprises.2021.2.1.prNames.0
```

4. Normally the OID output is abbreviated (shorter). You can change this behaviour with `-Of`:

```
$ snmptranslate -Of .iso.3.6.1.private.enterprises.2021.2.1.prNames.0
```

5. The problem with the above commands is that you have to remember the entire OID for what you're looking for. The `-IR` option ("random access lookup") searches the MIB tree for the node you want:

```
$ snmptranslate sysUpTime.0
Invalid object identifier: sysUpTime.0
```

So try this instead:

```
$ snmptranslate -IR sysUpTime.0
```

6. You can use regular expressions to find the exact node you want given only a piece of its name by using the `-Ib` (best match) option:

```
$ snmptranslate -Ib 'sys.*ime'
```

7. To get a list of *all* the nodes that match a given pattern, use the `-TB` flag:

```
$ snmptranslate -TB 'vacm.*table'
```

8. To get extended information about a MIB node, use the `-Td` (description) flag:

```
$ snmptranslate -On -Td -Ib 'sys.*ime'
```

9. To see a diagram of a section of the MIB tree, use the `-Tp` flag:

```
$ snmptranslate -Tp system
```

This is useful to explore the MIB tree.

10. Run `snmptranslate -Tp` without an OID argument. This prints the entire MIB tree from all the MIB files. I suggest pipe the output into `less`.