

Network Management & Monitoring

Introduction to SNMP

Network Startup Resource Center
www.nsrc.org



These materials are licensed under the Creative Commons Attribution-NonCommercial 4.0 International license
(<http://creativecommons.org/licenses/by-nc/4.0/>)

Overview

- What is SNMP?
- Polling and querying
- OIDs and MIBs
- Notifications
- SNMPv3

What is SNMP?

SNMP – Simple Network Management Protocol

- Structured protocol, structured information
- For querying network device state and receiving notifications
- Also can be used to change state
- Industry standard, hundreds of tools exist that use it
- Supported on any decent network equipment
- Transport : UDP ports 161 and 162 (notifications)

Uses for SNMP

Typical queries

- Bytes In/Out on an interface, errors
- CPU load
- Uptime
- Temperature or other vendor specific OIDs

For hosts (servers or workstations)

- Disk space
- Installed software
- Running processes
- ...

Windows and UNIX have SNMP agents

SNMP Versions

v1 (1988) Original specification

- Historic

v2 (1996) Failed Standard

- Security+new data types+new operators
- 64-bit counters, get-bulk, v2 notifications
- View-based access control model (VACM) introduced
- Historic, no current implementations left

v2c (1996) De facto standard

- v2 data types and operators
- v1 security (community string) (simple security model)
- Historic

v3 (1998) Robust security

- User/view based security (USM/VACM)
- Full Internet Standard

We will use SNMP v2c and v3 in this class

SNMP roles

Terminology—We will be using Manager and Agent

Manager (the monitoring station)

- Sometimes known as the SNMP client
- SNMPv3 calls it the Command Generator and Notification Receiver

Agent (running on the equipment/server)

- Sometimes known as the SNMP server
- SNMPv3 calls it the Command Responder and Notification Originator

How does SNMP work?

Basic operators

- **get** (manager -> agent)
 - Query for a value
- **getnext** (manager -> agent)
 - Get next value (e.g. list of values for a table)
- **getresponse** (agent -> manager)
 - Response to **get**, **getnext**, or **set**, includes error returns
- **set** (manager -> agent)
 - Set a value, or perform an action
- **trap** (agent -> manager)
 - Spontaneous notification from equipment (line down, temperature above threshold, ...)

How does SNMP work?

Query/response based

- Monitoring generally uses **get**, **getnext**, **getbulk**
- Changing state uses **set**
- Response is always a **getresponse**
- **getbulk** requires v2c or v3

Notifications are delivered as **traps** or **informs**

- **traps** are unacknowledged
- **informs** are acknowledged (v2c, v3)
- Use v2c format **traps**
- No one uses **informs**

The SNMP database

The information offered by a device is available in its Management Information Base (MIB)

- SNMP uses Object Identifiers (OIDs) to organize this information
- OIDs are keys to identifying each piece of data
- OIDs are organized into a tree structure that is the MIB
- MIB files document parts of the MIB on a device

OIDs

OID: Object Identifier

- A unique key to select a particular item of data in the device
- The same piece of information is always found at the same OID. That's simple!
- An OID is a variable-length string of numbers, e.g.
 .1.3.6.1.2.1.1.3
- Allocated hierarchically in a tree to ensure uniqueness (*similar to DNS*)

If Email Addresses were OIDs

user@nsrc.org

would have been something like:

user@nsrc.enterprises.private.internet.dod.org.iso

user@99999.1.4.1.6.3.1

except that we reverse the ordering, putting iso(1) first:

.1.3.6.1.4.1.99999.117.115.101.114

Note the portion after 99999—it spells “user” in ascii dotted decimal!

Don't worry about the deeply branched tree. What matters is that OIDs are unique.

- Ensures vendors don't have conflicting OIDs
- The numeric OID is what gets sent on the wire

OIDs and MIB files

Read from left to right

OID components separated by '.'

`.1.3.6.1.4.1.9. ...`

Each OID corresponds to a label

`.1.3.6.1.2.1.1.5 => sysName`

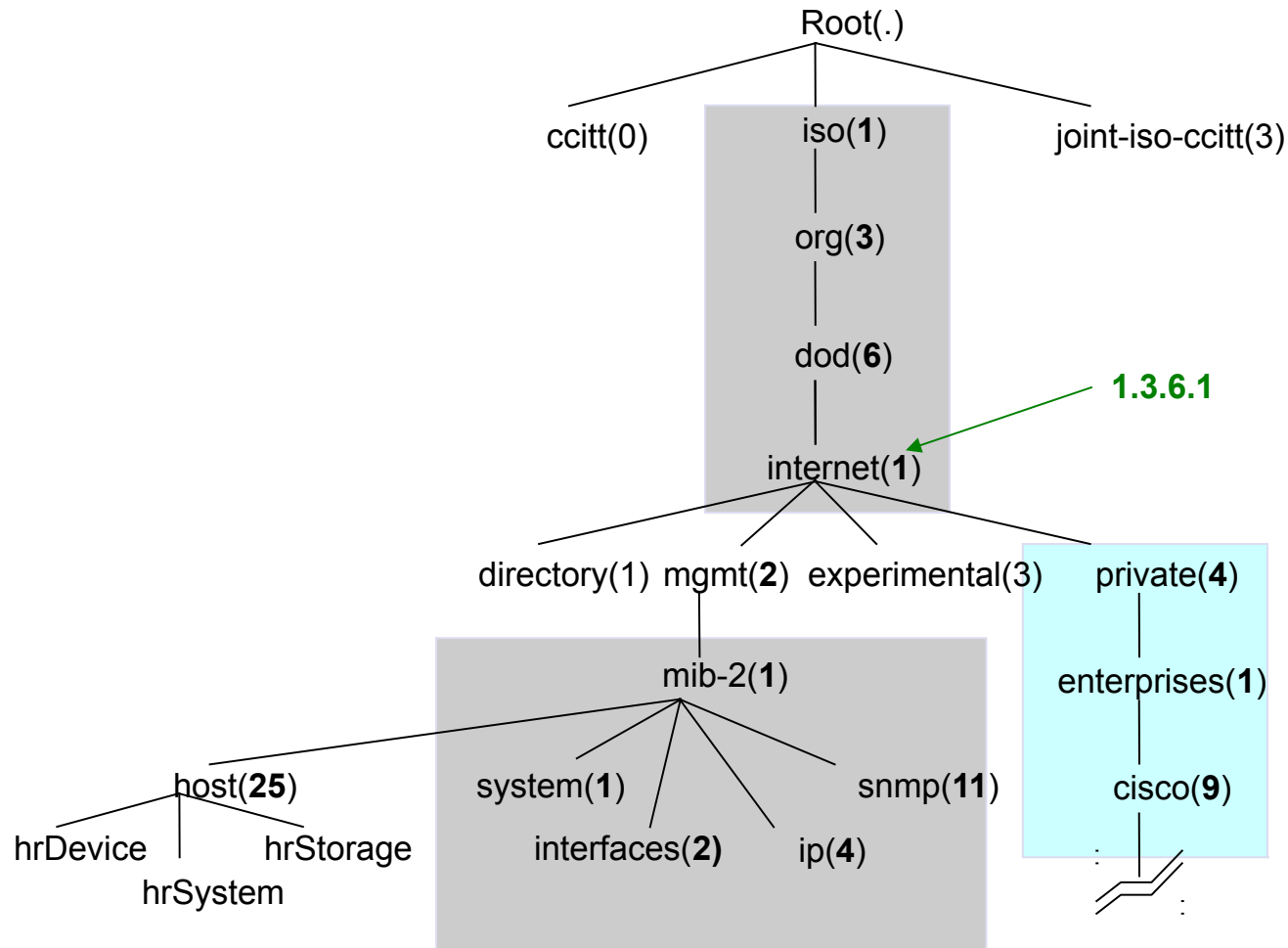
The complete path:

`.iso.org.dod.internet.mgmt.mib-2.system.sysName`

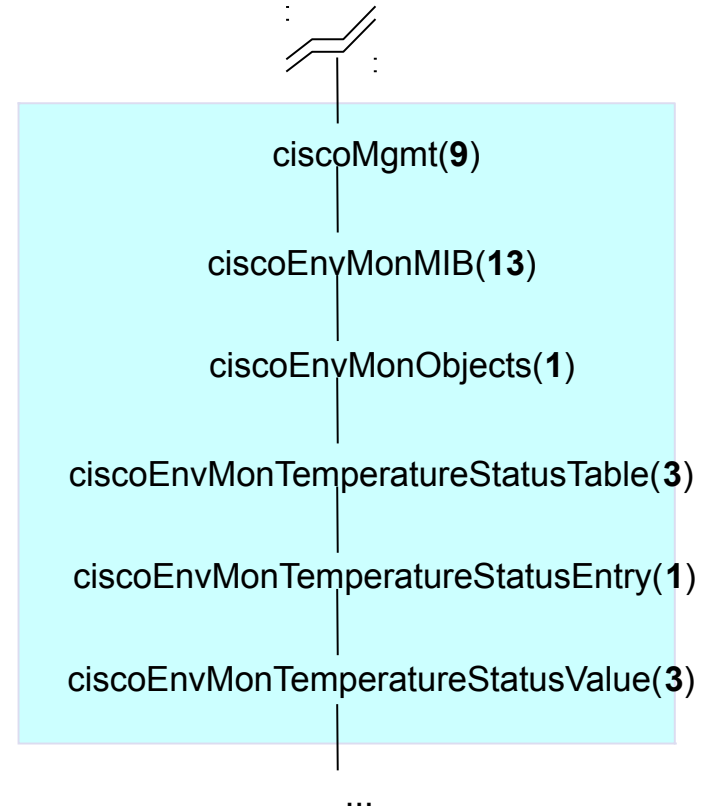
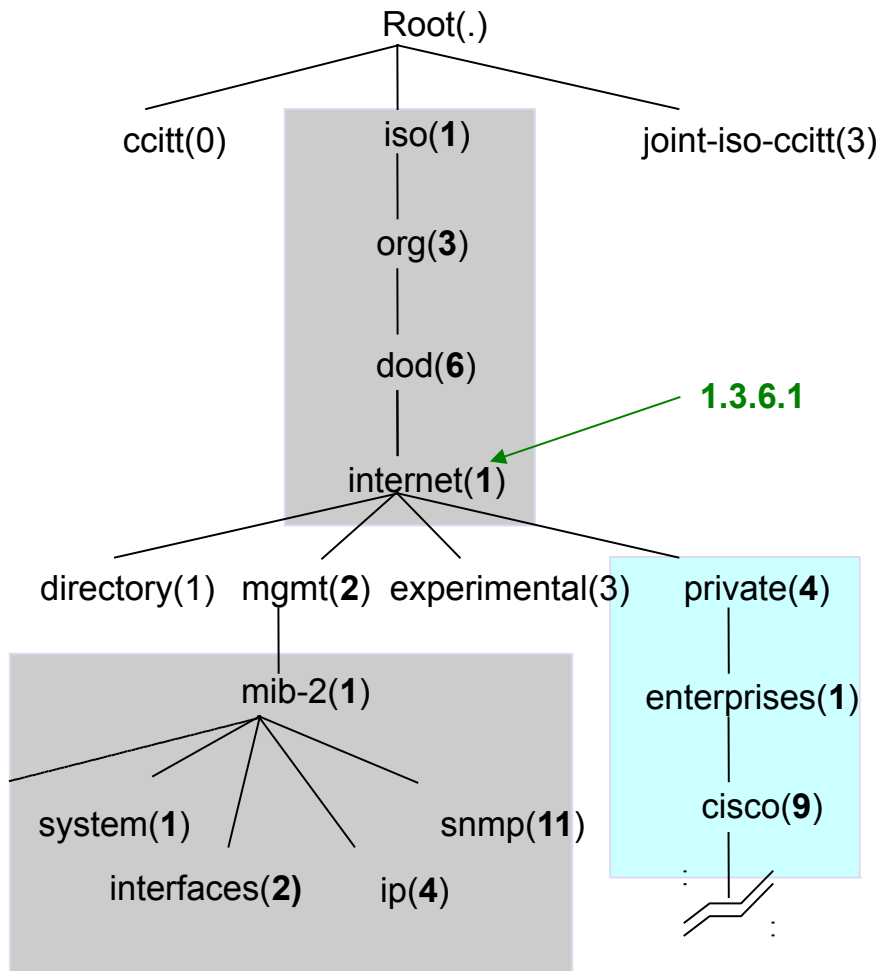
How do we convert from OIDs to Labels (and vice versa)?

- Use the MIBs files!

The MIB Tree



The MIB Tree



Interesting parts of the MIB tree

The Internet MIB, `.1.3.6.1`, really only two branches of interest:

- Standard MIBs

`.1.3.6.1.2.1 = .iso.org.dod.internet.mgmt.mib-2`

- Vendor-specific (proprietary) MIBs

`.1.3.6.1.4.1 =`

`.iso.org.dod.internet.private.enterprises`

The IEEE has MIBs of interest in three parts of the tree:

- IEEE 802 MIBs, including LLDP

`.1.0.8802 = .iso.standard.iso8802`

- IEEE 802.3 MIBs, including LAG

`.1.2.840.10006 = .iso.member-body.us.ieee802dot3`

- IEEE 802.11 wireless MIBs

`.1.2.840.10036 = .iso.member-body.us.ieee802dot11`

MIB Files

MIB files define the objects that can be queried, including:

- Object name
- Object description
- Data type (integer, text, list)

MIB files are structured text

- using an ASN.1 subset called the Structure of Management Information (SMI)

Standard MIB files include:

- MIB-II – (RFC1213) – a sub-group of MIBs
- HOST-RESOURCES-MIB (RFC2790)

MIB Sample

```
sysUpTime OBJECT-TYPE
    SYNTAX TimeTicks
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The time (in hundredths of a second) since the
        network management portion of the system was last
        re-initialized."
    ::= { system 3 }
```

sysUpTime OBJECT-TYPE

This defines the object called `sysUpTime`.

SYNTAX TimeTicks

This object is of the type `TimeTicks`. Object types are specified in the SMI we mentioned a moment ago.

ACCESS read-only

This object can only be read via SNMP (i.e., **get**, **getnext**); it cannot be changed (i.e., **set**).

STATUS mandatory

This object must be implemented in any SNMP agent.

DESCRIPTION

A description of the object

```
::= { system 3 }
```

The `sysUpTime` object is the third branch off of the `system` object group tree.

MIB Files

MIB files also make it possible to interpret a returned value from an agent

- For example, the status for a fan could be:
 - 1, 2, 3, 4, 5, or 6
 - What does it mean?
- Look for the Textual Convention (tc) in the MIB

MIB Sample

CiscoEnvMonState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Represents the state of a device being monitored.

Valid values are:

normal(1): the environment is good, such as low temperature.

temperature warning(2): the environment is bad, such as above normal operation range but not too high.

critical(3): the environment is very bad, such as temperature much higher than normal operation limit.

shutdown(4): the environment is the worst, the system should be shutdown immediately.

notPresent(5): the environmental monitor is not present, such as temperature sensors do not exist.

notFunctioning(6): the environmental monitor does not function properly, such as a temperature sensor generates a abnormal data like 1000 C.

SNMP and Security

SNMP versions 1 and 2c are insecure

SNMP version 3 was created to fix this

SNMPv3 authentication is based on a user

- “User-based Security Model” (USM)
 - Authenticity and integrity
 - Keys are used for users and messages have digital signatures generated with a hash function (MD5 or SHA)
 - Privacy
 - Messages can be encrypted with secret-key (private) algorithms (DES or AES)
 - Temporary validity
 - Utilizes a synchronized clock with a 150 second window with sequence checking

SNMPv3 Security Levels

noAuthNoPriv

- No authentication, no privacy

authNoPriv

- Authentication with no privacy

authPriv

- Authentication with privacy

Cisco SNMP Configuration

Read-only

```
snmp-server community NetManage RO
```

- Enables SNMPv1 and v2c

```
snmp-server group ReadGroup v3 auth
```

```
snmp-server user admin ReadGroup v3 auth sha NetManage
```

- SNMPv3 authentication, no encryption

Read-write

```
snmp-server group WriteGroup v3 auth write v1default
```

```
snmp-server user admin-rw WriteGroup v3 auth sha NetManage priv aes 128  
NetWrite
```

- Cisco allows authNoPriv and authPriv queries with this user
- You could also define a read-write user without encryption (priv)
- Note that we recommend using SNMP version 3 if you want write access using the **set** operator

Net-SNMP Configuration

Add a community string by editing `/etc/snmp/snmpd.conf` and adding:

```
rocommunity NetManage 10.10.0.0/16
```

Add the SNMPv3 user

```
# service snmpd stop
# net-snmp-create-v3-user -a SHA -A NetManage admin
# service snmpd start
```

Modify your user configuration file `~/.snmp/snmp.conf`, adding:

```
defVersion 3
defCommunity NetManage
defSecurityName admin
defSecurityLevel authNoPriv
defAuthPassphrase NetManage
defAuthType SHA
```

Querying an SNMP agent

Using Net-SNMP command line tools...

Some typical commands for querying:

- snmpget
- snmpwalk
- snmpbulkwalk (requires v2c or v3)
- snmpstatus
- snmptable

Syntax:

```
snmpXXX -v1 -c<community> host [OID]
```

```
snmpXXX -v2c -c<community> host [OID]
```

```
snmpXXX -v3 -lauthNoPriv -u<user> -aSHA -A<pass> host [OID]
```

However, because you've setup the snmp.conf file, it's much easier

```
snmpxxx host [OID]
```

- Or, if you want to force the version to v2c, for example:

```
snmpxxx -v2c host [OID]
```


Querying an SNMP agent

Let's look at some examples

```
snmpstatus 10.10.0.254
```

```
snmpget 10.10.0.254 ifNumber.0
```

```
snmpwalk -v2c 10.10.0.254 ifDescr
```

Querying an SNMP agent

Community:

- A "security" string (password) to define whether the querying manager will have RO (read only) or RW (read write) access
- This is the simplest form of authentication in SNMP

OID

- A value, for example, `.1.3.6.1.2.1.1.5.0`
- or its name equivalent: `sysName.0`

Let's ask for the system's name (using the OID above)

- Why the `.0`? What do you notice?

Queries Using snmp.conf

Two walks:

```
# snmpwalk 10.10.0.252 sysUpTime
DISMAN-EVENT-MIB::sysUpTimeInstance =
Timeticks: (1946738) 5:24:27.38
# snmpwalk -v2c 3 10.10.0.252 sysUpTime
DISMAN-EVENT-MIB::sysUpTimeInstance =
Timeticks: (1953429) 5:25:34.29
```

First walk used SNMPv3 as it was the default in snmp.conf, second walk specified SNMPv2c, and used the community string from snmp.conf.

Failed Query...Why?

Two gets:

```
# snmpget -v1 10.10.0.252 ifHCInOctets.1
```

Error in packet

Reason: (noSuchName) There is no such variable name in this MIB.

Failed object: IF-MIB::ifHCInOctets.1

```
# snmpget 10.10.0.252 ifHCInOctets.1
```

IF-MIB::ifHCInOctets.1 = Counter64: 475028252

Why? Notice the data type: Counter64. 64-bit counters are only supported in SNMPv2c and v3.

64-bit counters are important because 32-bit interface counters (ifInOctets) can wrap in 34 seconds on Gig interfaces.

How fast can it wrap on 10G?

SNMP failure: no response?

The device might be offline or unreachable

The device might not be running an SNMP agent

The device might be configured with a different community string

The device might be configured to refuse SNMP queries from your IP address

In all of these cases you will get no response

SNMP Best Practices

- Secure your SNMP access and traffic:
 - Management VLAN
 - Access lists
 - Use SNMPv3 with authentication for queries and sets where possible
- Use SNMPv2c traps
 - Better formatted than v1 traps
 - Accurate timestamps
- Do no harm
 - Only poll as fast as you really need
 - Possible to drive CPU load on devices up and affect other protocol processing
 - It does no good to poll every 5 seconds if the device updates the counter every 10

Coming up in our exercises...

- Using `snmpwalk`, `snmpget`
- Config file: `/etc/snmp/snmp.conf`
- Running Linux SNMP agent (daemon)
- Config file: `/etc/snmp/snmpd.conf`
- Loading MIBs

References

Essential SNMP (O'Reilly Books) Douglas Mauro, Kevin Schmidt

- <http://www.amazon.com/Essential-Second-Edition-Douglas-Mauro/dp/0596008406>

Wikipedia

- http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

MIB/OID Browser

- <http://oid-info.com/>

Cisco SNMP on IOS, MIB tools, and MIB/OID browser

- <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/snmp/command/nm-snmp-cr-book.html>
- <http://tools.cisco.com/ITDIT/MIBS/servlet/index>
- <http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en&substep=2&translate=Translate&tree=NO>

Open Source Java MIB Browser

- <http://www.dwipal.com/mibbrowser.htm>

SNMP Link – collection of SNMP resources

- <http://www.snmplink.org/>

Net-SNMP Open Source SNMP tools

- <http://net-snmp.sourceforge.net/>

Integration with Nagios

- <https://web.archive.org/web/20100614010336/http://www.cisl.ucar.edu/nets/tools/nagios/SNMP-traps.htm>

SNMP Versions

v1 Original specification

RFCs 1155,1157,1213

v2 Security+new data types+new operators

RFCs 1901,1909-1910,2011,2576,2578-2580,3416-3418

v2c De facto standard

Documented in RFC 3584

v3 Robust security: USM/VACM

RFCs 3411-3415,3417-3418,3826,5343,5345,5590

RFC 3584 specifies coexistence between versions