SNMP Technology White Paper

Keywords: SNMP, MIB, OID, Agent, NMS

Abstract: With the development of networks, the number of network devices is increasing rapidly. To manage these network devices efficiently, SNMP was addressed. This manual mainly introduces the basic concepts, working mechanisms and applications of SNMP.

Acronyms:

Acronym	Full spelling
MIB	Management Information Base
NMS	Network Management Station
OID	Object Identifier
SNMP	Simple Network Management Protocol
SMI	Structure of Management Information
USM	User-based Security Model
VACM	View-based Access Control Model
PDU	protocol data unit

Table of Contents

1 Overview

1.1 Background

With the fast development of the Internet, two problems were introduced:

- The increase of networks and network devices makes it hard for the administrators to monitor the status of all the devices in time, and find and correct network faults.
- Network devices may be from different vendors. If each vendor provides an independent set of management interfaces (for example, command lines), network management may become more and more complicated.

To solve the above mentioned problems, a set of standards (SNMP) covering service, protocol and management information base (MIB) is introduced.

1.2 Benefits

SNMP is an application-layer protocol between an NMS and several agents. It defines the standardized management framework, common languages in communication, security and access control mechanisms used in monitoring and managing devices in a network. The administrators can query device information, modify device parameters, monitor device status, and enable automatic detection of network faults and generation of reports by using SNMP.

SNMP features the following advantages:

- TCP/IP-based standard protocol, with UDP as the transportation layer protocol.
- Automatic network management. The administrators can search and modify information, find and diagnose network problems, plan for network growth, and generate reports on network nodes.
- Shielding of the physical differences between various devices, realizing automatic management of products from different vendors. Offering only the basic set of functions, SNMP makes the management tasks independent of both the physical features of the managed devices and the underlying networking technology. Thus, SNMP achieves effective management of devices from different vendors.

- Combination of simple request-reply mode and active notification mode, timeout and retransmission mechanism.
- Few packet types and simple packet format, which facilitates resolution and implementation.
- Authentication and privacy mechanisms provided in SNMpv3, which enhances security by user based and view based access control function.

2 Technical Characteristics

2.1 SNMP Network Structure

An SNMP enabled network comprises Network Management Station (NMS), Agent and MIB.

2.1.1 Introduction to NMS

NMS manages an SNMP-enabled network. It uses SNMP to manage and monitor the network devices in the network. NMS can be a server that manages the network or an application performing management function on a device.

NMS can send a request to an agent to query or modify one or more variables. At the same time, NMS can receive traps sent by the agent to obtain the status of the managed device.

2.1.2 Introduction to Agent

As an application module that resides in a network device, an agent maintains the information of the managed devices, responds to NMS requests, and sends the data to the NMS.

Upon receiving an NMS request, an agent completes the querying or modification operations, and sends the result to the NMS. Meanwhile, if the device fails or other events occur, the agent will send unsolicited traps to the NMS to notify it of the status changes of the device.

2.1.3 Introduction to MIB

1. MIB

Any managed resource can be identified as an object, which is known as the managed object. Management Information Base (MIB) is a collection of all the managed objects. It defines a set of characteristics associated with the managed objects, such as the object identifier (OID), access right and data type of the objects. Each Agent has its own MIB. NMS can read or write the managed objects in the MIB. The relationship between NMS, Agent and MIB is shown in Figure 1 .

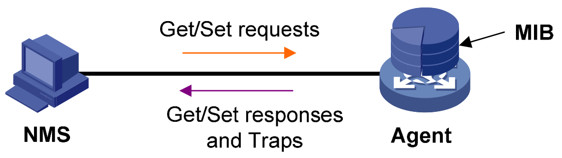

Figure 1 Relationship between NMS, Agent and MIB

2. MIB view

MIB view is a sub-set of MIB. You can bind a community name/username with a MIB view when configuring an agent, thus to control the MIB objects that NMS can access. You can configure the objects in the MIB view as excluded or included; excluded indicates that not all the nodes on the subtree are included in the current MIB view, and included indicates that the current MIB includes all the nodes on the subtree.

3. OID and subtree

MIB stores data using a tree structure. A node of the tree is a managed object and can be uniquely identified by a path starting from the root node. As illustrated in the following figure, the managed object **system** can be uniquely identified by a string of numbers {1.3.6.1.2.1.1}. This string of numbers is the OID of the managed object **system**.

A subtree can be identified by the OID of its root node. For example, the OID of the subtree with the root node being **private** is the OID of node **private** — {1.3.6.1.4}.

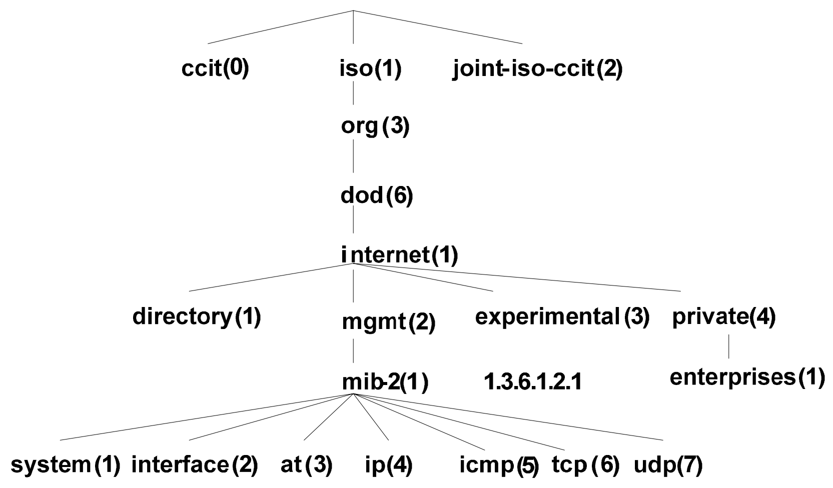

Figure 2 MIB tree

4. Subtree mask

A subtree OID used with a subtree mask defines a view subtree. A subtree mask is in hexadecimal format. After it is converted to binary bits, each bit corresponds to a node of the OID, where:

- 1 means full match, that is, the OID of the MIB object to be accessed must be identical to the subtree OID.
- 0 means wildcard match, that is, the OID of the MIB object to be accessed can be different from the subtree OID.

For example, provided the subtree mask 0xDB (11011011 in binary) and the subtree OID 1.3.6.1.6.1.2.1, their relationship is as shown in Figure 3 . The view determined by them includes all the nodes under the subtree whose OID is 1.3.*.1.6.*.2.1, where * represents any number.

Subtree OID	1	3	6	1	6	1	2	1
Subtree mask	1	1	0	1	1	0	1	1

Figure 3 Subtree OID and subtree mask

Note:

- If the number of bits in the subtree mask is greater than the number of nodes of the OID, the excessive bits of the subtree mask will be ignored during subtree mask-OID matching.
- If the number of bits in the subtree mask is smaller than the number of nodes of the OID, the short bits of the subtree mask will be set to 1 during subtree mask-OID matching.
- If no subtree mask is specified, the default subtree mask (all ones) will be used for mask-OID matching.

2.2 SNMP Versions

Currently, SNMP includes three versions: SNMPv1, SNMPv2c and SNMPv3.

2.2.1 SNMPv1

SNMPv1 is the first version of the SNMP protocol, providing a minimum network management function. The Structure of Management Information (SMI) and MIB of SNMPv1 are rather simple and have many security defects.

SNMPv1 uses community name for authentication. A community name plays a similar role as a password and can be used to control access from NMS to Agent. SNMP packets with community names that do not pass the authentication on the device are simply discarded.

2.2.2 SNMPv2c

SNMPv2c also uses community name for authentication. Compatible with SNMPv1, it extends the functions of SNMPv1. SNMPv2c provides more operation modes such as GetBulk; it supports more data types such as Counter32; and it provides various error codes, thus being able to distinguish errors in more detail.

2.2.3 SNMPv3

By adopting User-based Security Model (USM) and View-based Access Control (VACM) technologies, SNMPv3 enhances security. USM offers authentication and privacy functions; while VACM controls users' access to specific MIBs.

1. USM

USM introduces the concepts of username and group. You can set the authentication and privacy functions. The former is used to authenticate the validity of the sending end of the authentication packets, preventing access of illegal users; the latter is used to encrypt packets between the NMS and Agent, preventing the packets from being intercepted. USM ensures a more secure communication between SNMP NMS and SNMP Agent by authentication with privacy, authentication without privacy, or no authentication no privacy.

2. VACM

VACM defines the five elements: groups, security level, contexts, MIB views, and access policy. These five elements together control users' access to management information. Only a user with access rights can manage the objects. You can define different groups on the same SNMP entity; these groups are bound with MIB views. In addition, you can define multiple users in one group. When a user accesses the management information, he can access only the objects defined by the corresponding MIB view.

2.3 SNMP Operations

SNMP provides the following five basic operations:

- Get operation: The Get operation is a request sent by the NMS to the agent to retrieve one or more values from the agent.
- GetNext operation: The GetNext operation is a request sent by the NMS to retrieve the value of the next OID in the tree.
- Set operation: The Set operation is a request sent by the NMS to the agent to set one or more values of the agent.
- Response operation: The Response operation is a response sent by the agent to the NMS.
- Trap operation: The Trap operation is an unsolicited response sent by the agent to notify the NMS of the events occurred.

The device sends the first four kinds of packets to UDP port 161, and the agent sends traps to UDP port 162. By using two different port numbers, a single device can act as an agent and an NMS at the same time.

2.4 SNMP Messages

The following packets are defined based on different versions and operations of SNMP.

2.4.1 SNMPv1 Message

SNMP message

Version	Community	SNMP PDU

Get/GetNext/Set PDU

PDU type	Request ID	0	0	Variable bindings

Response PDU

PDU type	Request ID	Error status	Error index	Variable bindings

Trap PDU

PDU type	enterprise	Agent addr	Generic trap	Specific trap	Time stamp	Variable bindings

Variable bindings

Name1	Value1	Name2	Value2	…	Namen	Valuen

Figure 4 SNMPv1 message format

The above figure indicates that an SNMP message is composed of Version, Community, and SNMP PDU. The following describes the main fields in an SNMP message:

- Version: SNMP version.
- Community: Community name, used for the authentication between an agent and the NMS. Community name falls into read and write. If NMS performs Get or GetNext operation, read community name is used for authentication; if NMS performs Set operation, write community name is used for authentication.
- Request ID: It is used to match a response to a request. SNMP assigns a unique ID to each request.
- Error status: It is used in a response to indicate the errors when the agent processes the request, including noError, tooBig, noSuchName, badValue, readOnly, and genErr.

- Error index: Provides the information of the variables that caused the error when an error occurs.
- Variable bindings: It is composed of a variable name and value.
- enterprise: Type of the device that generates traps.
- Agent addr: Address of the device that generates traps.
- Generic trap: It includes coldStart, warmStart, linkDown, linkup, authenticationFailure, egpNeighborLoss and enterpriseSpecific.
- Specific trap: Specific trap information of a vendor.
- Time stamp: The amount of time between the time when the SNMP entity sending this message reinitialized and the time when traps were generated, that is, the value of sysUpTime.

2.4.2 SNMPv2c Message

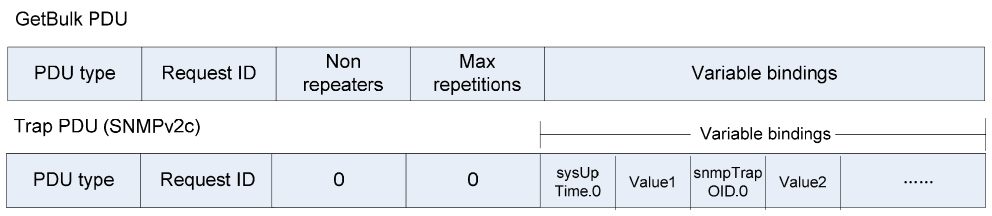

Figure 5 SNMPv2c message format

Compared with SNMPv1, GetBulk packets are added in SNMPv2c. GetBulk operation corresponds to GetNext operation. In a GetBulk operation, the setting of Non repeaters and Max repetitions parameters enables NMS to obtain data of many managed objects from an agent.

In SNMPv2c, trap message format is different from that in SNMPv1. SNMPv2c trap PDU adopts the format of SNMPv1 Get/GetNext/Set PDU, and sysUpTime and snmpTrapOID are used as variables in variable bindings to create a packet.

2.4.3 SNMPv3 Message

SNMPv3 message format is modified, but the PDU format is the same as that in SNMPv2c. Figure 6 shows the SNMPv3 message format.

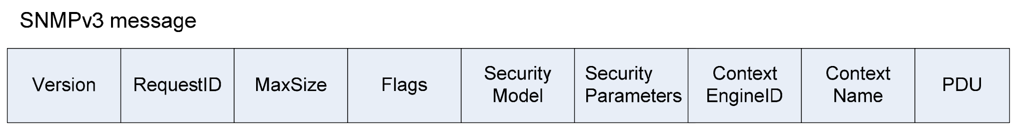

Figure 6 SNMPv3 message format

The entire SNMPv3 message can be authenticated, and EngineID, ContextName, and PDU are encrypted. RequestID, MaxSize, Flags, SecurityModel and SecurityParameters form the SNMPv3 message header.

The following describes the main fields in an SNMPv3 message:

- RequestID: Request ID.
- MaxSize: The maximum size of the message that the sender of the message can contain and the maximum size of the message that the sender of the message can receive.
- Flags: Message flag which occupies one byte. Only the lowest three bytes are valid. 0x0 indicates no authentication no privacy, 0x1 indicates authentication without privacy, 0x3 indicates authentication with privacy, and 0x4 indicates to send a report PDU.
- SecurityModel: Message security model, in the range 0 to 3. 0 indicates any model, 1 indicates SNMPv1 security model, 2 indicates SNMPv2c security model, and 3 indicates SNMPv3 security model.
- ContextEngineID: Uniquely identifies an SNMP entity. For a message received, this field decides how this message will be processed; for a message sent, this field is provided by the sender.
- ContextName: Identifies a context. Each contextName must be unique within an SNMP entity.

SecurityParameters includes the following fields:

- AuthoritativeEngineID: Specifies the snmpEngineID of the authoritative SNMP engine involved in the exchange of the message, used for identification, authentication and encryption for an SNMP entity. This field refers to the source for a trap, response, or report, and to the destination for a Get, GetNext, GetBulk, or Set operation.

- AuthoritativeEngineBoots: Specifies the snmpEngineBoots value at the authoritative SNMP engine involved in the exchange of the message. It indicates the number of times that this SNMP engine has initialized or reinitialized itself since its initial configuration.
- AuthoritativeEngineTime: Specifies the snmpEngineTime value at the authoritative SNMP engine involved in the exchange of the message. It is used for time window check.
- UserName: Specifies the user (principal) on whose behalf the message is being exchanged. Usernames configured on NMS and Agent must be the same.
- AuthenticationParameters: A key used in authentication calculation. If no authentication is performed, this field is null.
- PrivacyParameters: A parameter used in privacy calculation. For example, the value used for generating the initialization vector (IV) in the DES CBC algorithm. If no privacy is adopted, this field is null.

2.5 SNMP Mechanism

2.5.1 SNMPv1 and SNMPv2c Mechanism

SNMPv1 and SNMPv2c adopts almost the same mechanism. New error codes and GetBulk operation are added in SNMPv2c. The following describes the SNMPv1/v2c mechanisms.

1. Get operation

NMS wants to obtain the value of the node sysName of a managed device (the sysName object is in the accessible view), using **public** as the read community name:

(1) NMS sends a Get request to Agent. The main fields in the request are set as follows: version to 1, community to public, name 1 in variable bindings in the PDU to sysName.0.

(2) Agent sends a get response to NMS to tell NMS whether the values are successfully obtained. If succeeded, the field Value1 in Variable bindings in the response PDU is the device name (for example, Agent010-H3C); if failed, the reason for the error is filled into the Error status field, and error location is filled into the Error index field.

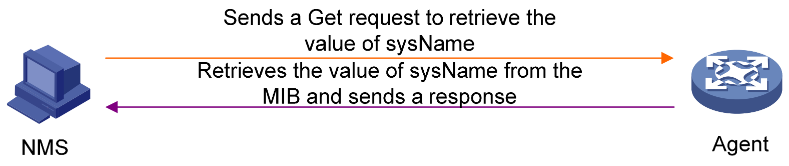
Sends a Get request to retrieve the value of sysName
Retrieves the value of sysName from the MIB and sends a response
NMS
Agent

Figure 7 Get operation

2. GetNext operation

NMS wants to obtain the value of the node sysLocation next to node sysName of a managed device (the sysName and sysLocation objects are in the accessible view), using **public** as the read community name:

(1) NMS sends a GetNext request to Agent. The main fields in the request are set as follows: Version to 1, Community to public, and Name 1 in variable bindings in the PDU to sysName.0.

(2) Agent sends NMS a GetNext response. If succeeded, the value of Name 1 in Variable bindings in the response PDU is the next node sysLocation.0 of node sysName.0, and the value of Value 1 is, for example, Beijing China; if failed, the reason for the error will be added to the Error status field, and position will be added to the Error index field.

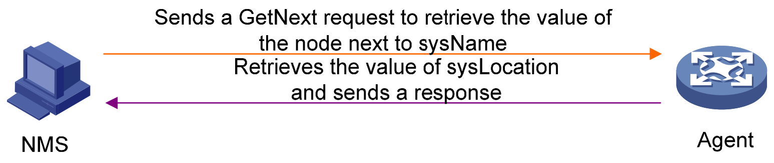
Sends a GetNext request to retrieve the value of the node next to sysName
Retrieves the value of sysLocation and sends a response
NMS
Agent

Figure 8 GetNext operation

3. Set operation

NMS wants to set the value of node sysName of the managed device to Device01, using **private** as the read community name:

(1) NMS sends a Set request to Agent. The main fields in the request are set as follows: Version to 1, Community to private, Name 1 in variable bindings in the PDU to sysName.0, and Value1 to Device01.

(2) Agent sends NMS a Set response. If succeeded, the value of Value1 in Variable bindings in the response PDU is the new name of the device (for example, Device01); if failed, the reason for the error will be added to the Error status field, and position will be added to the Error index field.

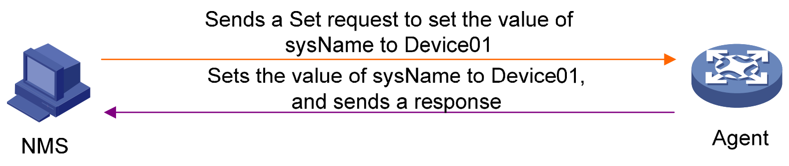
Sends a Set request to set the value of sysName to Device01
Sets the value of sysName to Device01, and sends a response
NMS
Agent

Figure 9 Set operation

4. Trap operation

If abnormalities occur on a device, Agent will notify NMS by sending unsolicited traps. For example, if the cable on a port of the device is plugged out, Agent will send a linkDown trap to NMS. In the trap, the value of the Version field is 1, that of the Community field is public, that of the enterprise field is the value of sysObjectID.0 (for example, enterprises.25506), that of the Generic trap field is linkDown, and the Variable bindings field contains the interface information.

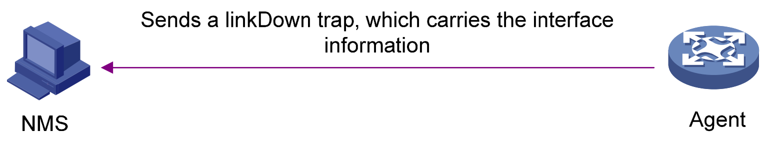
Sends a linkDown trap, which carries the interface information
NMS
Agent

Figure 10 Trap operation

2.5.2 SNMPv3 Mechanism

Compared to the SNMPv1 and SNMPv2c mechanisms, authentication, encryption, and decryption are added in SNMPv3 mechanism. In the following example, Get operations are performed using authentication and encryption.

(1) NMS sends a Get request without any authentication and encryption parameters. In the request, the Flags field is set to 0x4 to obtain the values of the related parameters such as contextEngineID, contextName, AuthoritativeEngineID, AuthoritativeEngineBoots, and AuthoritativeEngineTime.

(2) Agent processes the request, and sends a report, which contains the values of the above parameters.

(3) NMS sends a Get request again to Agent. In the request, the main fields are set as follows: Version to 3, the parameter values obtained in step 2 are added to the corresponding fields, the value of Name1 field in Variable bindings in the PDU to sysname.0, AuthenticationParameters is calculated by the configured authentication algorithm, PrivacyParameters is calculated by the configured encryption algorithm, and encrypt the PDU data using the configured privacy protocol.

(4) Agent authenticates the Get request first. If the Get request passes the authentication, Agent will decrypt the PDU. If decryption succeeds, Agent will obtain the value of sysName.0, and fill the value of the field Value1 in Variable bindings in the response PDU with the device name (for example, Agent010). If authentication or decryption fails, or obtaining parameter values fails, the error reason will be filled into the Error status field, and the error location will be filled into the Error index field. At last, Agent will encrypt the PDU, set the parameters contextEngineID, contextName, AuthoritativeEngineID, AuthoritativeEngineBoots, AuthoritativeEngineTime, AuthenticationParameters and PrivacyParameters, and then send a response.

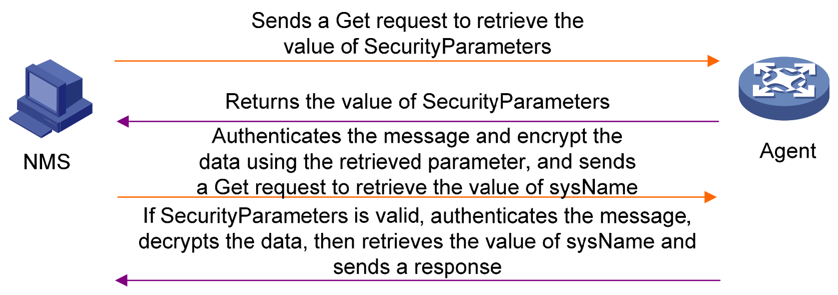
Sends a Get request to retrieve the value of SecurityParameters
Returns the value of SecurityParameters
Authenticates the message and encrypt the data using the retrieved parameter, and sends a Get request to retrieve the value of sysName
If SecurityParameters is valid, authenticates the message, decrypts the data, then retrieves the value of sysName and sends a response
NMS
Agent

Figure 11 SNMPv3 Get operation

3 H3C’s Implementation of SNMP

An H3C device supports SNMPv1, SNMPv2c, and SNMPv3. To make SNMPv1 and SNMPv2c compatible with SNMPv3, you can configure group, user and view for these two versions. In this case, you only need to configure the parameter settings of the community name on the NMS as the username configured on the device. You can enable multiple SNMP versions on the device at the same time, but you need to make them consistent with those on the NMS.

4 Application Scenarios

4.1 Network Diagram

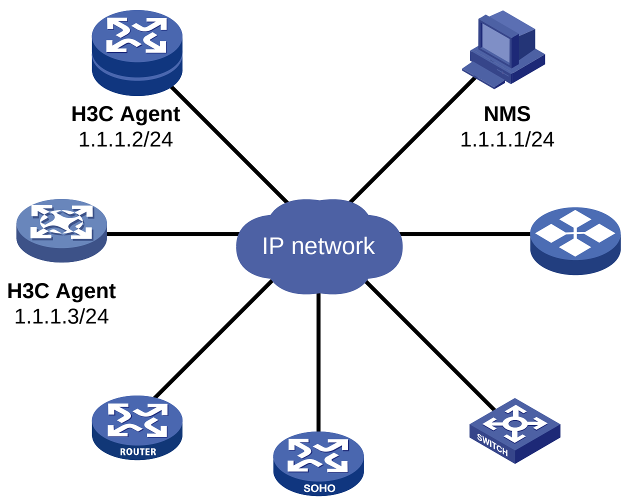

Figure 12 SNMP network diagram

4.2 Networking Environment

In a network, the vendors of the network devices are different, and device models are different. It is required to monitor and manage the network and the devices.

5 References

- RFC 1157
- RFC 2578
- RFC 2579
- RFC 3411
- RFC 3412
- RFC 3415