Chapter 17: SNMP

Alpha Bridge Technologies

Table of Contents

Chapter 17 SNMP

17.1 SNMP Overview

SNMP (Simple Network Management Protocol) is an important network management protocol on TCP / IP networks, implementing network management by exchanging packets on the network. The SNMP protocol provides the possibility of centralized management of large networks. Its goal is to ensure the management information is transmitted between any two points. SNMP is convenient for the network administrator to retrieve information from any node on the network, make modifications, find faults, and complete fault diagnosis, capacity planning and report generation.

SNMP structure is divided into two parts: NMS and Agent. NMS (Network Management Station) is a workstation that runs client programs while Agent is a server-side software running on a network device. The NMS can forward GetRequest, GetNextRequest, and SetRequest packets to the Agent. Upon receiving the NMS request message, the agent performs Read or Write operations according to the packet type and generates a Response packet to return to the NMS. On the other hand, when the device encounters an abnormal event such as hot / cold start, the agent will forward a trap packet to NMS to report the events.

The system supports SNMP v1, SNMP v2c and SNMP v3. SNMP V1 provides a simple authentication mechanism, does not support the administrator-to-manager communications, and v1 Trap has no confirmation mechanism. V2c enhanced v1 management model (on security), management information structure, protocol operation, manager and communication ability between managers to increase the creation and deletion of the table, the communication ability between managers, reducing the storage side of the agent. V3 implements the user authentication mechanism and packet encryption mechanism, which greatly improves the security of the SNMP protocol.

This function cooperates with the network management software to log on to the Switch and manage the Switch.

17.2 Configure SNMP-Agent

17.2.1 SNMP-Agent Configuration List

Configuration Task	Description	Detailed Configuration
Configure the Basic Parameters	Required	17.2.2

Configure the Community Name	Required	17.2.3
Configure the Views	Optional	17.2.4
Configure the Group	Optional	17.2.5
Configure the User	Optional	17.2.6
Display SNMP Configuration	Optional	17.2.7

17.2.2 Configure the Basic Parameters

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Enable/disable SNMP Traps/informs	[**undo**] **snmp-agent enable** { informs | traps } [*notificationtype-list*]	
Configure sysContact	[**undo**] **snmp-agent scontact** *syscontact*	
Configure sysLocation	[**undo**] **snmp-agent location** *syslocation*	
Configure SW	[**undo**] **snmp-agent name** *SW*	
Configure maximum length of snmp protocol packets	[**undo**] **snmp-agent max-packet-length** *length*	
Configure host	[**undo**] **snmp-agent host** *host-addr* [version { 1 | 2c | 3 [auth | noauth | priv] }] *community-string* [udp-port *port*] [notify-type [*notifytype-list*]]	
Configure snmp trap-source	[**undo**] **snmp-agent trap-source** *ipaddress*	
Configure snmp-agent engineoid	[**undo**] **snmp-agent engineoid** { **local** *engineid-string* | **remote** *ip-address*	
	[udp-port port-number] engineid-string }	

17.2.3 Configure the Community Name

SNMP adopts the community name authentication scheme. SNMP packets that do not match the community name will be discarded. SNMP community is named by a string, known as the community name. Different communities can have read-only or read-write access permission. A community with read-only access can only query system information. However, in addition toquery the system information, the community with read-write access permission can perform the system configurations. It defaults to no community name.

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Configure the community name	**snmp-agent community** *community-name* { ro | rw } { deny | permit } [view *view-name*]	
Display the community name	**display snmp-agent community**	
Remove the community name	**undo snmp-agent community** *community-name*	

17.2.4 Configure the Views

It is used to configure the views available to access control and the subtrees that they contain.The iso, internet, and sysview exist by default. Delete and modify the internet is not supported.

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Configure the views	**snmp-agent view** *view-name oid-tree* { included | excluded }	
Delete the views	**undo snmp-agent view** *view-name* [oid-tree]	

17.2.5 Configure the Group

This configuration task can be used to configure an access control group. By default, there are two snmpv3 groups: (1) The initial group with the security level of auth; (2) The initial group with the security level of noauthpriv(No authentication is required and no encryption is required).

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Configure the group	**snmp-agent group** *groupname* { 1 | 2c | 3 [auth | noauth | priv] [context context-name] } [read *readview*] [wrete *writeview*] [notify *notifyview*]	
Delete the group	**undo snmp-agent group** *groupname* { 1 | 2c | 3 [auth | noauth | priv] [context *context-name*] }	

17.2.6 Configure the User

It is used to configure the user for the local engine or for the remote engine that can be identified. By default, the following users exist: (1)initialmd5, (2) initialsha, (3) initialnone.

The above three users are reserved for the system and cannot be used by the user. When Configure a user, you need to ensure that the engine to which this user belongs is identifiable. When an identifiable engine is deleted, the users it contains are also deleted.

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Configure the user	**snmp-agent user** *username groupname* [remote *host* [udp-port *port*]] [auth { md5 | sha } { authpassword { encrypt-auth password *authpassword* | *authpassword* } | authkey { encrypt-authkey *authkey* | *authkey* } } [priv des { privpassword { encrypt-privpassword *privpassword* | *privpassword* } | privkey { encrypt-privkey *privkey* | *privkey* } }]	
Delete the user	**undo snmp-agent user** *username* [remote *host* [udp-port *port*]]	

17.2.7 Display SNMP-Agent Configuration

Operation	Command	Remarks
display snmp community configuration	**display snmp community**	
display snmp contact configuration	**display snmp contact**	
display snmp engineid configuration	**display snmp engineid** { local | remote }	
display snmp group configuration	**display snmp group**	
display snmp host configuration	**display snmp host**	
display snmp location configuration	**display snmp location**	
display snmpmax-packet-length configuration	**display snmp max-packet-length**	
display snmp name configuration	**display snmp name**	
display snmp notify configuration	**display snmp notify**	
display snmp user configuration	**display snmp user**	
display snmp view configuration	**display snmp view**	

17.3 RMON

RMON (Remote Network Monitoring) mainly implements statistics and alarm functions, and is used for remote monitoring and management of managed devices by management devices inthe network.

The statistics function means that the managed device can track and count various traffic information on the network segment connected by its port periodically or continuously, such asthe total number of messages received on a network segment in a certain period of time, or the total number of ultra long messages received.

The alarm function refers to that the managed device can monitor the value of the specified MIB variable. When the value reaches the alarm threshold (for example, the port rate reachesthe specified value, or the proportion of broadcast messages reaches the specified value), it can automatically record logs, generate alarm information, and send it to the SNMP module, which sends it to the management device. For details about alarm information, see "SNMP" in"Network Management and Monitoring Configuration Guide".

17.3.1 Working mechanism of RMON

RMON allows multiple monitors to collect data in two ways:

・The first method uses a dedicated RMON probe to collect data, and the management devicedirectly obtains management information from the RMON probe and controls network resources. This method can obtain all the information of the RMON MIB;

・The second method is to directly implant the RMON Agent into network devices (routers, switches, HUBs, etc.) to make them become network facilities with RMON probe function. The management device uses the basic operations of SNMP to exchange data information with the

RMON Agent and collect network management information. However, this method is limited by the device resources and cannot obtain all the data of the RMON MIB. It only collects the information of four groups: event group, alarm group, history group and statistics group.

17.3.2 RMON group

Multiple RMON groups are defined in the RMON protocol. The device implements the statisticsgroup, history group, event group, alarm group, agent configuration group and user history group supported in the public MIB.

17.3.3 Statistics group

The statistics group stipulates that the system will continuously make statistics on various traffic information of ports (currently only supports statistics on Ethernet ports), and store the statistics results in the Ethernet Statistics Table for management devices to view at any time. After the statistics table item is successfully created under the specified interface, the statistics group will count the number of messages of the current interface. The result of its statistics is acontinuous cumulative value.

The statistical information includes the number of network conflicts, the number of CRC checkerror messages, the number of data messages that are too small (or too large), the number of broadcast and multicast messages, the number of received bytes, the number of received messages, etc.

17.3.4 Event Group

The event group is used to define the event index number and the event handling method. The events defined by the event group are used in the alarm group entries and extended alarm group entries. When the monitored object reaches the alarm condition, an eventwill be triggered. The event can be handled in the following ways:

• Log: Record the event related information (event occurrence time, event content, etc.) in the event log table of the RMON MIB of the device, so that the management device can view itthrough the SNMP Get operation.

• Trap: indicates that when an event is triggered, an alarm message will be generated and sent to the SNMP module of the device.

• Log Trap: When an event is triggered, it not only records the log on the device, but alsogenerates alarm information and sends it to the SNMP module of the device.

• None: No processing.

17.3.5 Alarm group

RMON alarm management can monitor the specified alarm variables (such as the total number of messages received by the port etherStatsPkts). After the user defines the alarm tableitem, the system will obtain the value of the monitored alarm variable according to the defined time cycle. When the value of the alarm variable is greater than or equal to the upper limit threshold, an upper limit alarm event will be triggered; When the value of the alarm variable is less than or equal to the lower limit threshold, a lower limit alarm event will be triggered, and the alarm management will handle it according to the definition of the event.

17.3.6 Protocol Specification

Protocol specifications related to RMON include:

- RFC 4502：Remote Network Monitoring Management Information Base Version 2
- RFC 2819：Remote Network Monitoring Management Information Base Status ofthis Memo

17.3.7 Configure the RMON Ethernet statistics function

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Enter the Ethernet interface view	**interface** *interface-type interface-number*	
Create Statistics Table Item	**rmon statistics** *entry-number [* **owner** *text]*	

17.3.8 Configure RMON historical statistics function

When configuring the RMON historical statistics function, you should pay attention to:

•The entry number of the history control table item must be globally unique. If it has been used under other interfaces, the creation operation will fail.

• Under the same interface, multiple historical control table items can be created, but the values of entry number and sampling interval of different table items must be different, otherwise the creation operation fails.

• The maximum number of control history table entries allowed to be created for the entire device is 100. When the total number of control history table entries is more than 100, thecreation operation fails.

• When creating a historical control table item, if the specified bucket number parametervalue exceeds the historical table capacity actually supported by the device, the historical control table item will be added, but the value of the bucket number corresponding to the table item isthe historical table capacity actually supported by the device.

Operation	Command	Remarks
Enter the global configuration mode.	**system-view**	
Enter the Ethernet interface view	**interface** *interface-type interface-number*	
Create History Control Table Entry	**Rmon** **history entry-number buckets number int** **erval interval [owner text]**	

17.3.9 Configure RMON alarm function

If alarm information needs to be sent to the management device (NMS) when an alarmevent is triggered, you must ensure that the SNMP Agent has been correctly configured beforeconfiguring the RMON alarm function. For the configuration of SNMP Agent, see SNMP in the Network Management and Monitoring Configuration Guide.

Operation	Command	Remarks
Event Table Entry	*Description string, event type (log, trap,* *logtrap or none) and community name* *(security string)*	
Alarm table item	*Alarm variable, sampling interval, sampling* *type (absolute or delta), upper threshold*	
	(threshold value1) and lower threshold *(threshold value2)*	
Extended alarm table item	*Alarm variable formula (primalarm formula), sampling interval, sampling type (absolute or delta), upper threshold (threshold value1)* *and lower threshold (threshold value2)*	

Configuration Steps

Operation	Command	Remarks
(Optional) Create an event table entry.	**rmon event entry-number [description string] { log | log-trap security-string | none | trap security-string } [owner text]**	
Create alarm table item	**rmon alarm** *entry-number alarm-variable sampling-interval {* **absolute** | **delta** *}* *[* **startup-alarm** *{* **falling** | **rising** | **rising-falling** *}]* **rising-threshold** *threshold-value1 event-entry1* **falling-threshold** *threshold-value2 event-entry2 [* **owner** *text]*	
Create extended alarm table item	**rmon prialarm entry-number prialarm-formula prialarm-des sampling-interval { absolute | delta } [startup-alarm { falling | rising | rising-falling }] rising-threshold threshold -value1 event-entry1 falling-threshold threshold-value2 event-entry2 entrytype { forever | cycle cycle-period } [owner text]**	

17.3.10 RMON display and maintenance

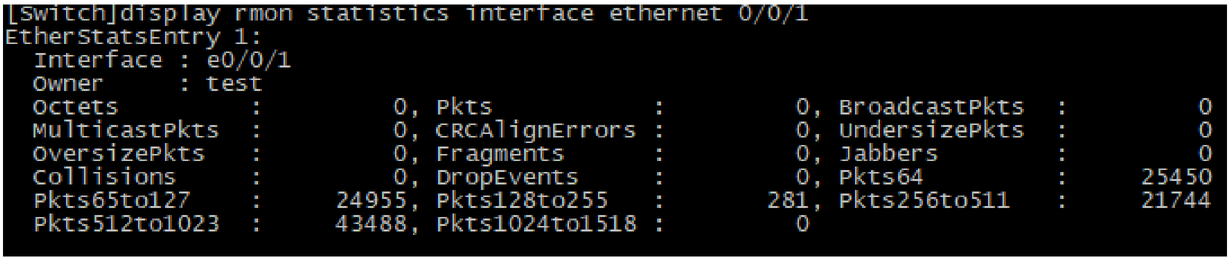
After completing the above configuration, execute the display command in any view to display the running status of RMON after configuration. Verify the configuration effect by viewing the display information.

Operation	Command	Remarks
Display RMON statistics	**display rmon statistics** [*interface-type interface-number]*	
Display RMON historical control table and historical sampling information	**display rmon history** [*interface-type interface-number]*	
Display relevant information of RMON alarm table items	**display rmon alarm** [*entry-number]*	
Display relevant information of RMON extended alarm table items	**display rmon prialarm** [*entry-number]*	
Displays information about RMON event table entries	**display rmon event** [*entry-number]*	
Display information about event log entries	**display rmon eventlog** [*entry-number]*	

17.3.11 Example of typical configuration of RMON

1) Now it is necessary to conduct performance statistics on the messages received by Gigabit Ethernet 0/0/1 through the RMON statistics table. The administrator can check the statistical data at any time to understand the status of the interface receiving messages.

```
<Sw> system-view
[sw] interface gigabitethernet 1/0/1
[sw-Ethernet0/0/1] rmon statistics 1 owner user1
```



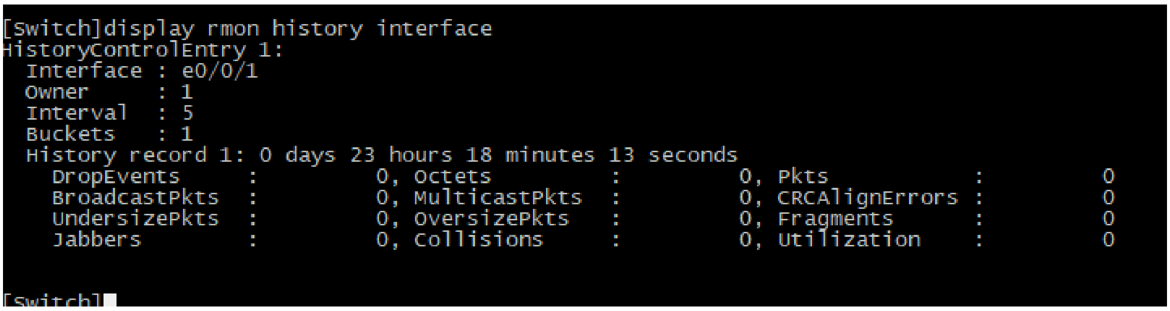
```
[Switch]display rmon statistics interface ethernet 0/0/1
EtherStatsEntry 1:
  Interface : e0/0/1
  Owner     : test
  Octets         :         0, Pkts           :         0, BroadcastPkts   :        0
  MulticastPkts  :         0, CRCAlignErrors :         0, UndersizePkts   :        0
  OversizePkts   :         0, Fragments      :         0, Jabbers         :        0
  Collisions     :         0, DropEvents     :         0, Pkts64          :    25450
  Pkts65to127    :     24955, Pkts128to255   :       281, Pkts256to511    :    21744
  Pkts512to1023  :     43488, Pkts1024to1518 :         0
```

2) Example of typical configuration of historical statistics function

```
<SW> system-view

[Sw] gigabitethernet 0/0/1

[Sw-Ethernet1/0/1] rmon history 1 buckets 1 interval 60 owner user1
```



```
[Switch]display rmon history interface
HistoryControlEntry 1:
  Interface : e0/0/1
  Owner     : 1
  Interval  : 5
  Buckets   : 1
  History record 1: 0 days 23 hours 18 minutes 13 seconds
    DropEvents     :         0, Octets        :         0, Pkts           :         0
    BroadcastPkts  :         0, MulticastPkts :         0, CRCAlignErrors :         0
    UndersizePkts  :         0, OversizePkts  :         0, Fragments      :         0
    Jabbers        :         0, Collisions    :         0, Utilization    :         0

[Switch]
```

3) Example of typical configuration of alarm function

snmp-agent community public ro permit view iso snmp-agent

community private rw permit view iso

snmp-agent host 10.1.1.200 version 2c public udp-port 162 notify-type bridge g bn gbnsavecfg

interfaces rmon snmp

snmp-agent enable traps

```
[SW] interface gigabitethernet 1/0/1

[SW-GigabitEthernet1/0/1]  rmon  statistics  1  owner  user1 [SW-

GigabitEthernet1/0/1] quit

[SW] rmon event 1 trap public owner user1

[SW] rmon alarm 1 1.3.6.1.4.1.8888.1.2.4.4.20.1.1 2 5 delta rising-threshold 100

1 falling-threshold 50 1 owner user1
```

```
<SW> display rmon alarm 1

AlarmEntry 1 owned by user1 is VALID.Sample type

                       : delta

  Sampled variable     :  1.3.6.1.4.1.8888.1.2.4.4.20.1.1  2<etherStatsOctets.1> Sampling
  interval (in seconds)             : 5
  Rising threshold      :  100(associated  with  event  1)  Falling
  threshold             :  50(associated with  event  1)  Alarm  sent
  upon entry startup  : risingOrFallingAlarmLatest value      : 0


[Switch]display   rmon   statistics   interface   ethernet   0/0/1
EtherStatsEntry 1:
  Interface : e0/0/1
  Owner     :
          test
  Octets    :           0, Pkts         :       0, BroadcastPkts  :      0
  MulticastPkts  :       0, CRCAlignErrors :     0, UndersizePkts  :          0
  OversizePkts   :       0, Fragments     :       0, Jabbers       :     0
  Collisions    :       0, DropEvents    :       0, Pkts64       :     275
                                                                       59
  Pkts65to127   :    26028, Pkts128to255   :     297, Pkts256to511 :       22800
  Pkts512to1023  :    45600, Pkts1024to1518 :       0
```