RMON

Ethernet Switch

ZyNOS 4.0

Support Notes

Version 4.00

July 2011

ZyXEL
Unleash Networking Power

Overview

Remote Monitoring (RMON) is a standard monitoring specification that enables various network monitors and console systems to exchange network-monitoring data. RMON provides network administrators with more freedom in selecting network-monitoring probes and consoles with features that meet their particular networking needs.

RMON was originally developed to address the problem of managing LAN segments and remote sites from a central location. The RMON specification, which is an extension of the SNMP MIB, is a standard monitoring specification. Within an RMON network monitoring data is defined by a set of statistics and functions and exchanged between various different monitors and console systems. Resultant data is used to monitor network utilization for network planning and performance-tuning, as well as assisting in network fault diagnosis.

RMON solutions are comprised of two components: a probe (or an agent or a monitor), and a client, usually a management station. Agents store network information within their RMON MIB and are normally found as embedded software on network hardware such as routers and switches although they can be a program running on a PC. Agents can only see the traffic that flows through them so they must be placed on each LAN segment or WAN link that is to be monitored. Clients, or management stations, communicate with the RMON agent or probe, using SNMP to obtain and correlate RMON data.

Now, there are a number of variations to the RMON MIB. For example, the Token Ring RMON MIB provides objects specific to managing Token Ring networks. The SMON MIB extends RMON by providing RMON analysis for switched networks.

RMON Groups

RMON delivers information in nine RMON groups of monitoring elements, each providing specific sets of data to meet common network-monitoring requirements. Each group is optional so that vendors do not need to support all the groups within the Management Information Base (MIB). Some RMON groups require support of other RMON groups to function properly. RMON MIB is organized into a number of functional groups. Our switches support 4 groups in RFC1757: Statistics, History,

Alarm and Event. Table 1 summarizes the nine monitoring groups specified in the RFC 1757 Ethernet RMON MIB.

Table 1: RMON Monitoring Groups

RMON 1 MIB Group	Function	Elements
Statistics	Contains statistics measured by the probe for each monitored interface on this device.	Packets dropped, packets sent, bytes sent (octets), broadcast packets, multicast packets, CRC errors, runts, giants, fragments, jabbers, collisions, and counters for packets ranging from 64 to 128, 128 to 256, 256 to 512, 512 to 1024, and 1024 to 1518 bytes.
History	Records periodic statistical samples from a network and stores for retrieval.	Sample period, number of samples, items sampled.
Alarm	Periodically takes statistical samples and compares them with set thresholds for events generation.	Includes the alarm table and requires the implementation of the event group. Alarm type, interval, starting threshold, stop threshold.
Host	Contains statistics associated with each host discovered on the network.	Host address, packets, and bytes received and transmitted, as well as broadcast, multicast, and error packets.
HostTopN	Prepares tables that describe the top hosts.	Statistics, host(s), sample start and stop periods, rate base, duration.
Matrix	Stores and retrieves statistics for conversations between sets of two addresses.	Source and destination address pairs and packets, bytes, and errors for each pair.

Filters	Enables packets to be matched by a filter equation for capturing or events.	Bit-filter type (mask or not mask), filter expression (bit level), conditional expression (and, or not) to other filters.
Packet Capture	Enables packets to be captured after they flow through a channel.	Size of buffer for captured packets, full status (alarm), number of captured packets.
Events	Controls the generation and notification of events from this device.	Event type, description, last time event sent

Groups of RMON MIB

The objects are arranged into the following groups:

Statistics (iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).rmon(16).statistics(1))

History	(1.3.6.1.2.1.16.2)
Alarm	(1.3.6.1.2.1.16.3)
Hosts	(1.3.6.1.2.1.16.4)
hostTopN	(1.3.6.1.2.1.16.5)
Matrix	(1.3.6.1.2.1.16.6)
Filter	(1.3.6.1.2.1.16.7)
Capture	(1.3.6.1.2.1.16.8)
Event	(1.3.6.1.2.1.16.9)

All groups in this MIB are optional. (MIB-II is **mandatory**)

RMON Command Example

RMON Event Command Example

This example shows how to configure the Switch's action when an RMON event using the following settings:

- event index number: 2
- enable event logging and SNMP traps: Yes
- the trap's community: public
- who will handle this alarm: operator
- additional description for this event entry: test

This example also shows how to display the setting results.

```
ras# config
ras(config)# rmon event eventtable 2 log trap public owner operator
description test
ras(config)# exit
ras# show rmon event eventtable 2
  Event 2 owned by operator is valid
    eventType: logandtrap
    eventCommunity: public
    eventDescription: test
```

RMON Alarm Command Example

This example also shows how to display the setting results.

```
ras# config
ras(config)# rmon alarm alarmtable 2 variable ifInErrors.1 interval 60
sample-type delta startup-alarm rising rising-threshold 50 2 falling-
threshold 0 2 owner operator
ras(config)# exit
ras# show rmon alarm alarmtable
  Alarm 2 owned by operator is valid
    alarmVariable: ifInErrors.1
    alarmInterval: 60
    alarmSampleType: delta
    alarmStartupAlarm: rising
    alarmRisingThreshold: 50
    alarmRisingEventIndex: 2
    alarmFallingThreshold: 0
    alarmFallingEventIndex: 0
    Last value monitored: 0
ras#
```

RMON Statistics Command Example

This example shows how to configure the settings to display current network traffic statistics using the following settings:

- the Ethernet statistics table entry's index number: 1
- collecting data samples from which port: 12

This example also shows how to display the data collection results.

```
ras# config
ras(config)# rmon statistics etherstats 1 port-channel 12
ras(config)# exit
ras# show rmon statistics etherstats index 1
  Statistics 1 owned by  is valid
    Monitor on interface port-channel 12
    etherStatsDropEvents: 0
    etherStatsOctets: 1576159
    etherStatsPkts: 19861
    etherStatsBroadcastPkts: 16721
    etherStatsMulticastPkts: 1453
    etherStatsCRCAlignErrors: 2
    etherStatsUndersizePkts: 0
    etherStatsOversizePkts: 0
    etherStatsFragments: 0
    etherStatsJabbers: 0
    etherStatsCollisions: 0
    Packet length distribution:
      64: 17952
      65-127: 666
      128-255: 671
      256-511: 509
      512-1023: 26
      1024-1518: 37
ras#
```

RMON History Command Example

This example shows how to configure the settings to display historical network traffic statisticsusing the following settings:

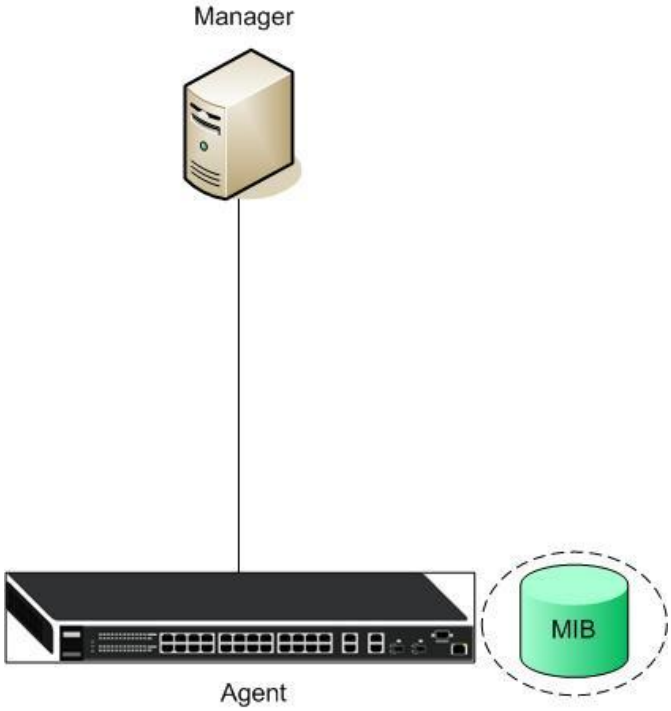
- the history control table entry's index number: 1
- how many data sampling data you want to store: 10
- time interval between data samplings: 10 seconds
- collecting data samples from which port: 12

This example also shows how to display the data collection results.

```
ras# config
ras(config)# rmon history historycontrol 1 buckets 10 interval 10 port-
channel 12
ras(config)# exit
ras# show rmon history historycontrol index 1
  History control 1 owned by  is valid
    Monitors interface port-channel 12 every 10 sec.
    historyControlBucketsRequested: 10
    historyControlBucketsGranted: 10
    Monitored history 1:
      Monitored at 0 days 00h:08m:59s
      etherHistoryIntervalStart: 539
      etherHistoryDropEvents: 0
      etherHistoryOctets: 667217
      etherHistoryPkts: 7697
      etherHistoryBroadcastPkts: 5952
      etherHistoryMulticastPkts: 505
      etherHistoryCRCAlignErrors: 2
      etherHistoryUndersizePkts: 0
      etherHistoryOversizePkts: 0
      etherHistoryFragments: 0
      etherHistoryJabbers: 0
      etherHistoryCollisions: 0
      etherHistoryUtilization: 72
    Monitored history 2:
      Monitored at 0 days 00h:09m:08s
      etherHistoryIntervalStart: 548
      etherHistoryDropEvents: 0
      etherHistoryOctets: 673408
      etherHistoryPkts: 7759
      etherHistoryBroadcastPkts: 5978
      etherHistoryMulticastPkts: 519
      etherHistoryCRCAlignErrors: 2
      etherHistoryUndersizePkts: 0
      etherHistoryOversizePkts: 0
      etherHistoryFragments: 0
      etherHistoryJabbers: 0
      etherHistoryCollisions: 0
      etherHistoryUtilization: 0
ras#
```

Scenario

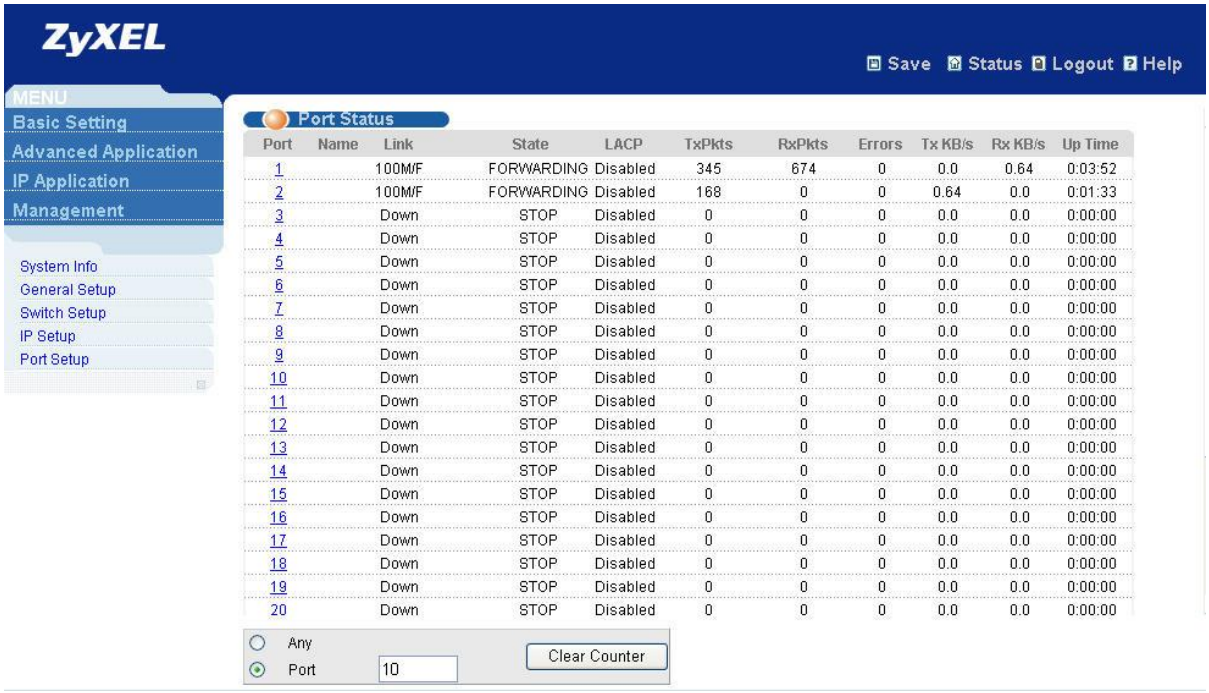
Below is a simple topology which could give us a common view about SNMP.



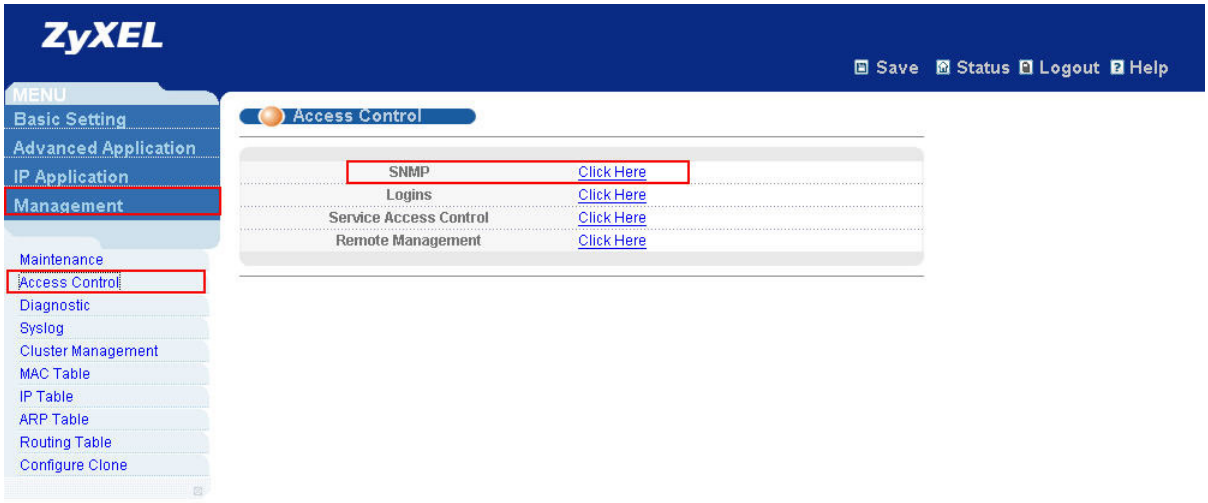
There are three SNMP components in this topology: Manager, Agent, and MIB. In this sample, we use SNMPc as the manager server. SNMPc could be installed on a Windows system.

Configuration ZyXEL switch using the Web GUI

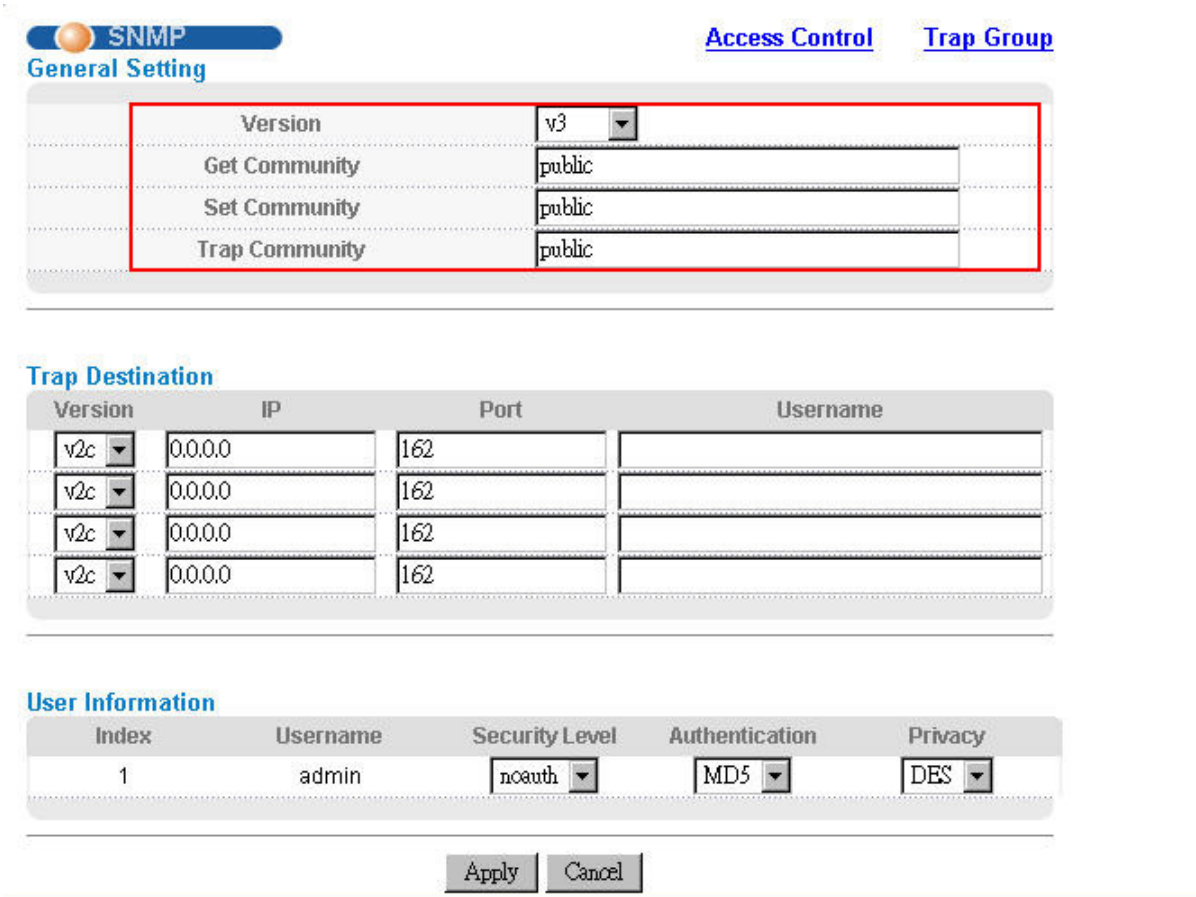
1. Connect the MGMT port to a PC or Notebook with the RJ45 Cable.
2. By default, the MGMT IP address of the out-band port is 192.168.0.1/24
3. Set your NIC to 192.168.0.100/24
4. Open an Internet browser (e.g. IE) and enter http://192.168.0.1 into the URL field.
5. By default, the username for the administrator is “admin” and the password is “1234”.
6. After successfully logging in you will see a screen similar to the one below.



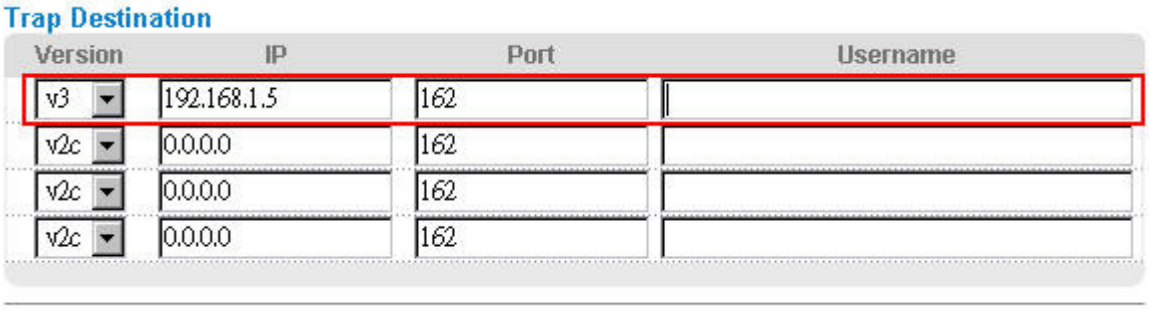
7. To enter the “**SNMP**” page, click “**Management**” “**Access Control**” “**SNMP**”



8. In the “SNMP” page, we can choose what SNMP version, SNMPv2c, SNMPv3 or both. Here we choose to use SNMPv3. Then configure the “**Get Community**”, “**Set Community**”, and “**Trap Community**” values. The term “**Community**” is nothing more than “**password**”. “**Get Community**” which means: The password to get the SNMP messages and so on. Here we use the default communities to let users easier to understand. By default, the communities are “**public**”.

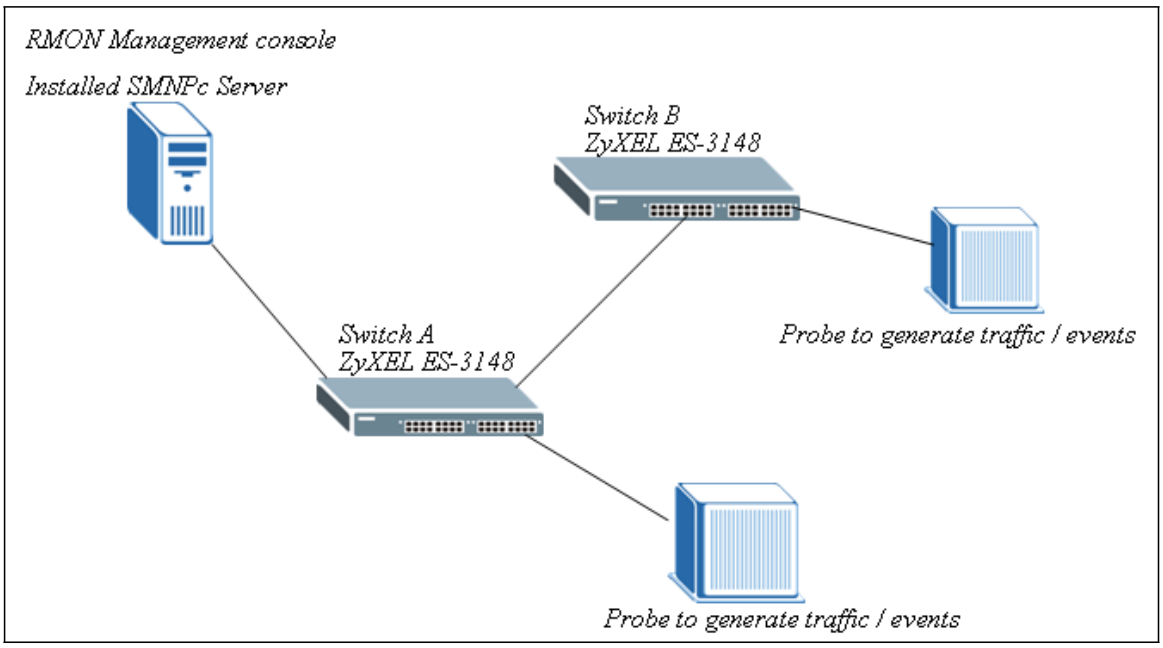


9. We can configure which kind of events should trigger the SNMP trap message. In the “Trap Destination” section, we can choose the SNMP version of the trap message, the destination we want to trigger to, destination port, and the username.



10. Configure the “**User information**”. There we can choose Security Level, Authentication methods, and encryption methods. Here we use “**noauth**” for no authentication.Click “**Apply**”

Scenario



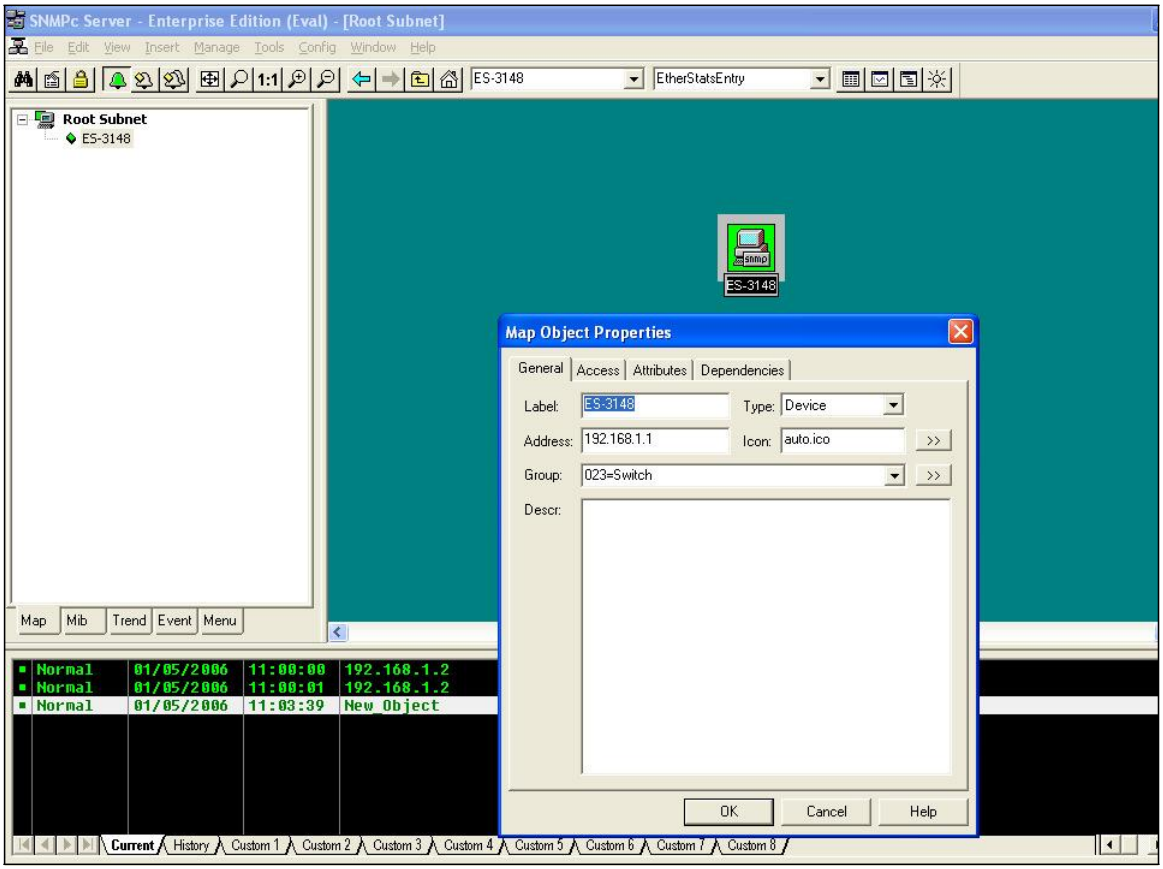
In this illustration, SNMPc Enterprise Edition Version 5.1.6c is installed on the PC. And this PC is defined as “RMON management console”. This PC can ping both ZyXEL ES-3148 (both Switch A & Switch B). And there are some probes / networking devices to generate the traffic to the ZyXEL Switches in order to verify the RMON result. Since the work flow and the technology of RMON on the two switches are the same, only one of the ZyXEL ES-3148 Switch will be demonstrated at this time.

Since RMON is an extension of the SNMP, SNMP must be enabled first in the ZyXEL ES-3148. By default SNMP is enabled and it has set Community (Get,Set,Trap) to “public”. And Trap Destination to 0.0.0.0; It is not mandatory to change the default value in order for SNMP & RMON to work. Therefore, modification is not necessary in this case.

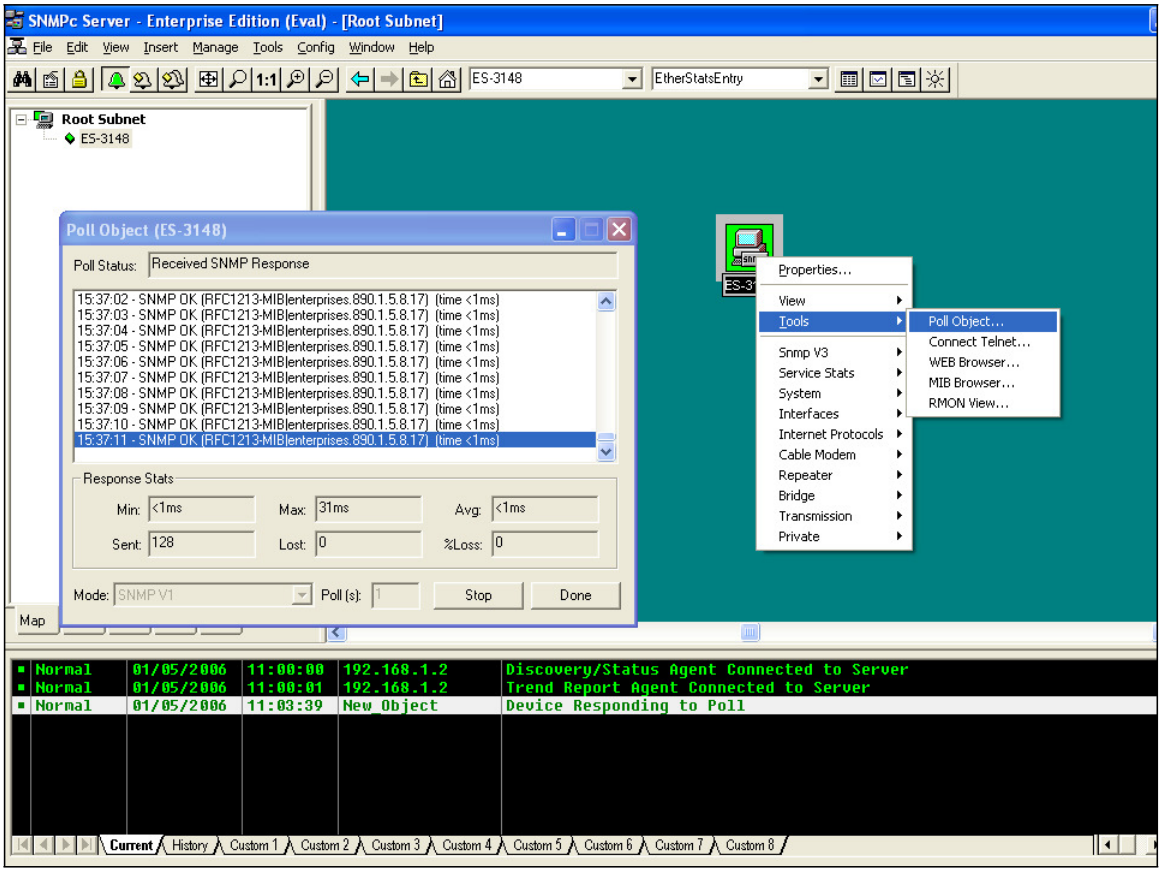
In this scenario, we are going to monitor the Broadcast Packets by using the RMON MIB. The following will demonstrate the steps to monitor the Broadcast Packets by using SNMPc Enterprise Edition Version 5.1.6c.

1. Methodology of Scenario Verification

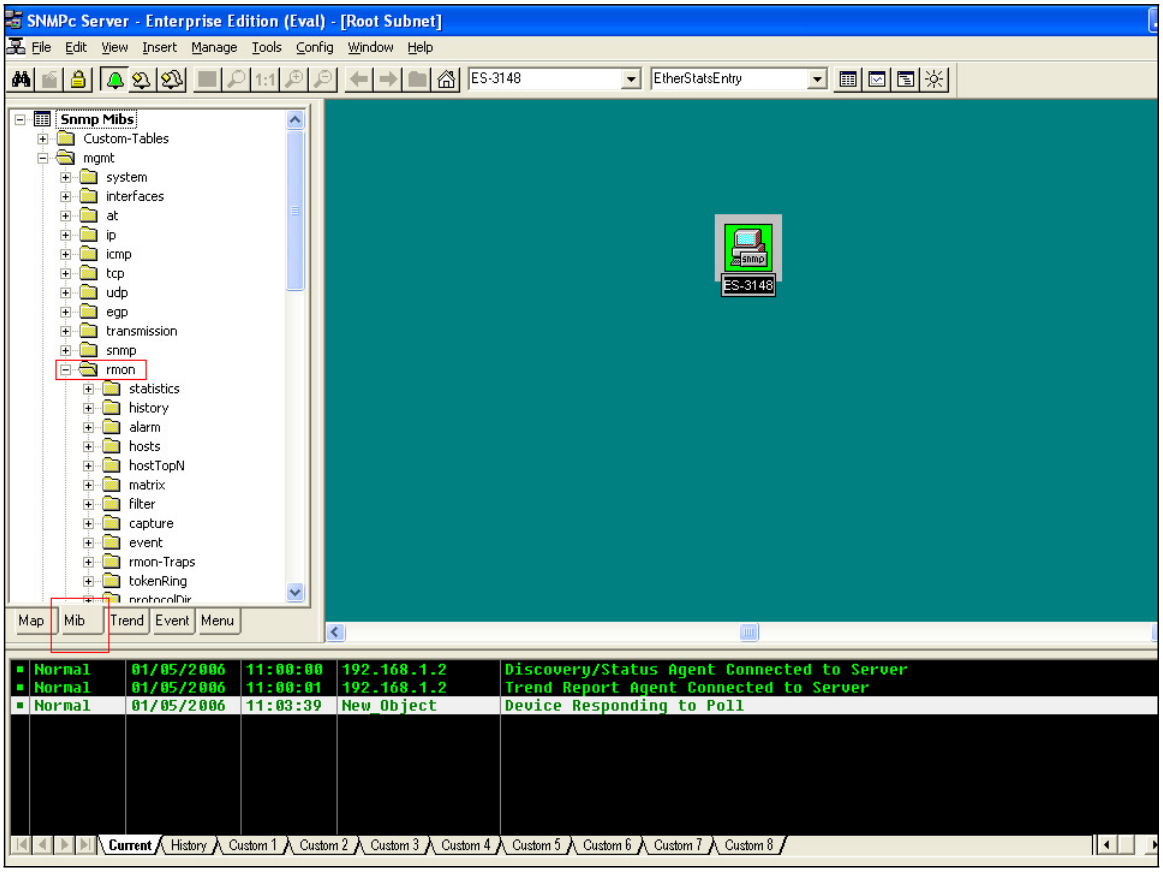
1.Open your SNMPc program first, then pick the ZyXEL-3148 Switch (it is first named as device “root”) and give it the correct IP information to get the SNMP information. Also, you can rename it to whatever you want.



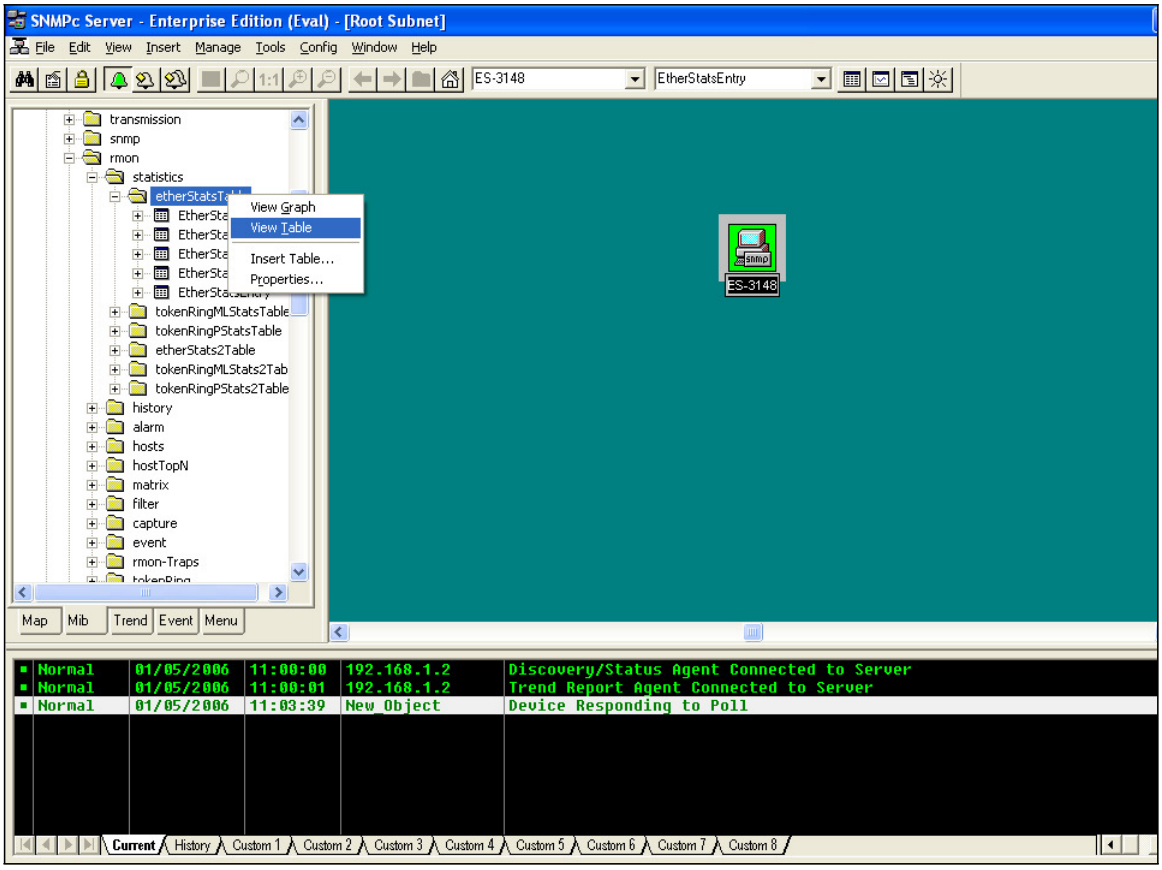
You can verify if your configuration is correct by using the “Poll Object” option. Just right click our mouse on the ES-3148 icon and it is located inside the “Tools”.



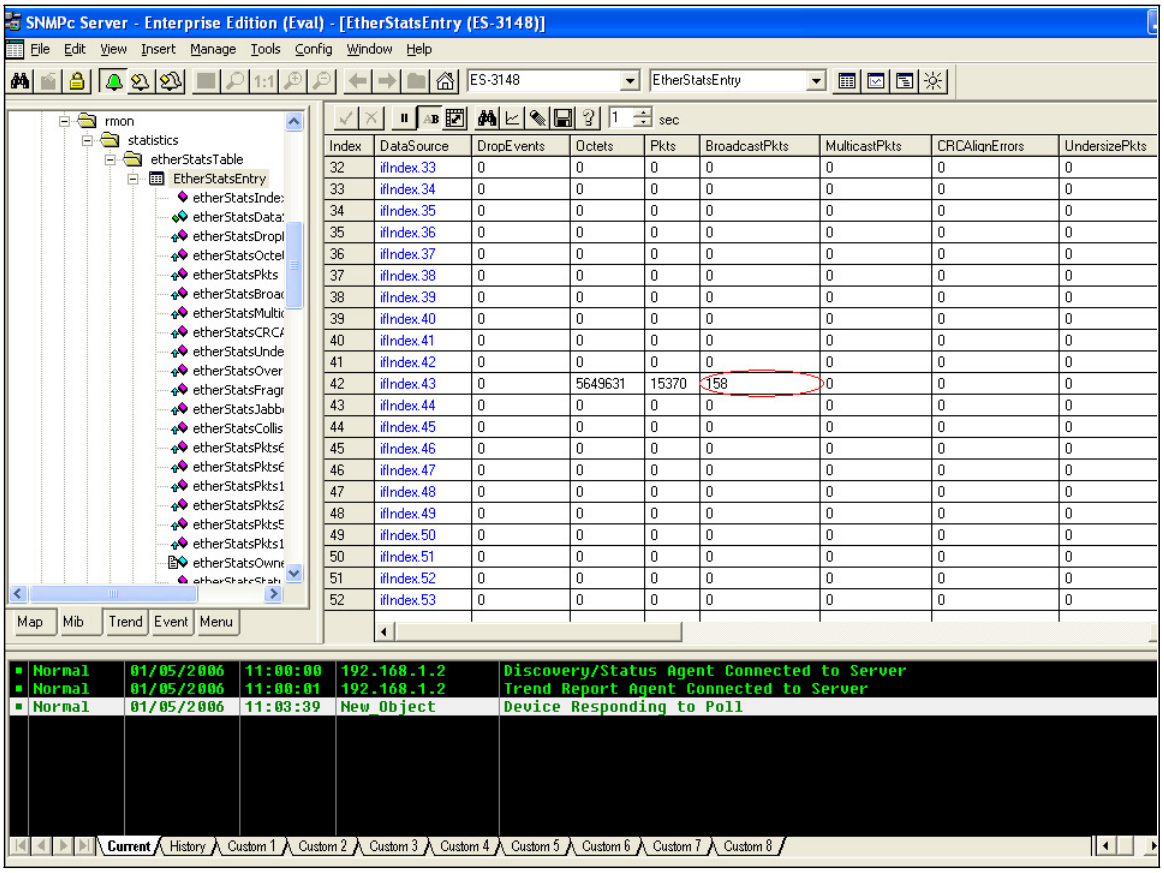
2. Secondly, click on the “Mib” tab and expend the SNMP Mibs’ tree. You will find that there is an “rmon” group over there and again you can expend its sub-tree.



3. Right click the “etherStatsTable” and choose “View Table”

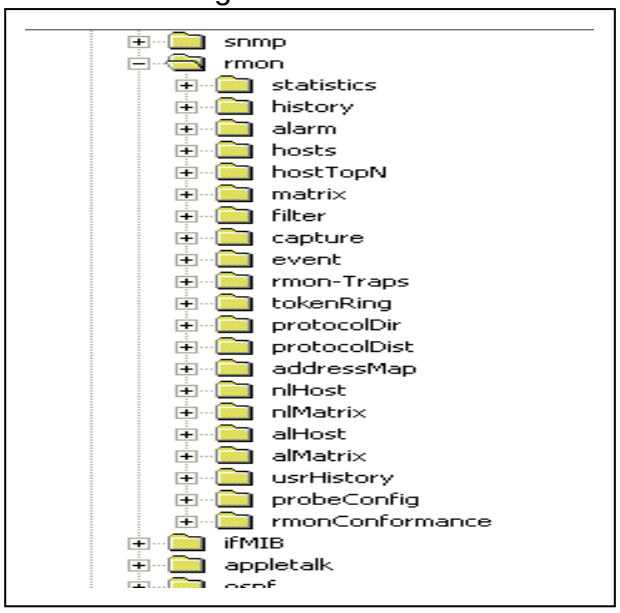


4. Find the interface or port that you are looking for. And you can look at the corresponding field and therefore find the value that you want to monitor. In this case, we are looking for the Boardcast Packets.



Try to generate some broadcast traffic from the probe or your network device, then you should see the BroadcastPkts increasing.

5. In conclusion, if the Switch supports RMON, then you can get the values from the Switch in the RMON Group(s), otherwise, it will return 0 and always stays 0. Without the supporting of RMON, then it is impossible to monitor those elements in the RMON MIB Group



Configuration ZyXEL switch using the CLI

```
vlan 1 name 1
   normal ""
   fixed 1-28
   forbidden ""
   untagged 1-28
   ip address 192.168.1.1 255.255.255.0
exit
interface route-domain 192.168.1.1/24
exit
snmp-server version v3
/*Using the default communities, thus, no more SNMPv3 related configuration needed*/
```