

Switch Management

Introduction to Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is an application layer protocol for collecting information about devices on the network. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite which enables network managers to monitor, configure, and troubleshoot the network, and to plan for network growth.

One of the reasons why SNMP is referred to as 'simple' is because of its small number of commands. With SNMP, vendors can easily build agents to their products, and this has led to the widespread use of network management today.

SNMP Architecture

SNMP is based on a *manager* and *agents* model. The 'manager' is the console where network management functions are performed, and the 'agents' are the entities or software modules that interface with the actual devices being managed. These managed objects are arranged in a virtual information database known as a Management Information Base (MIB).

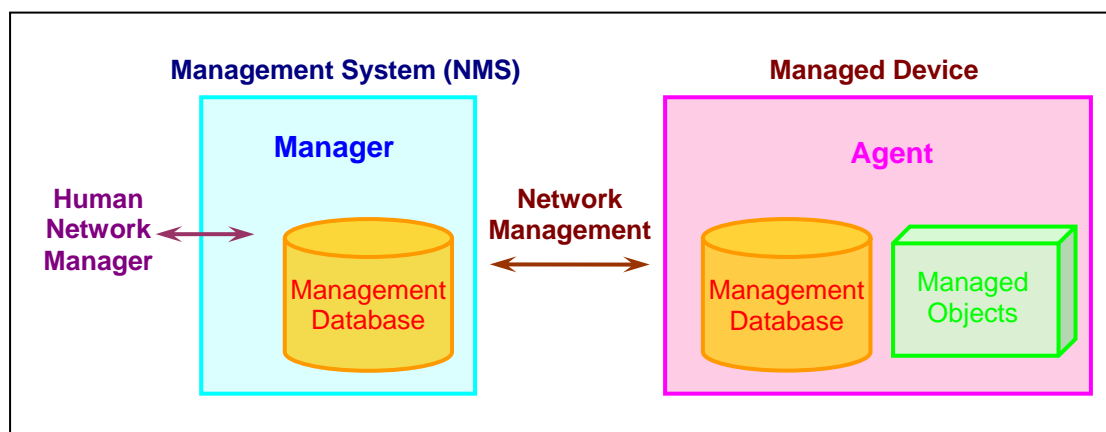


Figure 1. Manager/Agent Model for SNMP

The SNMP manager and agent use a MIB and relatively small set of commands to exchange information of the managed devices that contain network nodes or managed objects. These objects may include bridges, hubs, routers, firewalls, printers, computers, and network servers, configuration parameters, performance statistics, and others. An object identifier (OID) is assigned to distinguish each variable uniquely in the MIB and SNMP messages.

The SNMP manager is typically implemented as a Network Management Station (NMS) using full SNMP protocol. It is able to query, get response from, set variables in, and acknowledge asynchronous events from agents. A typical SNMP agent usually implements the full SNMP protocol. It stores and retrieves management data that is defined by the MIB.

Managed devices are monitored and controlled using four basic SNMP commands: read, write, trap, and traversal operations.

- The NMS uses the **Read** command to monitor managed devices. It can examine different variables that are maintained by managed devices.
- The NMS uses the **Write** command to control managed devices. It can change the values of variables stored within managed devices.
- The managed devices use **Trap** command to asynchronously report events to the NMS. The managed devices will send a trap to the NMS when some types of events occur.
- The NMS uses **traversal operations** to determine which variables a managed device can support and to sequentially gather information into tables.

In a paradigm for network management using a manager and agent architecture, managed objects must be logically accessible. This means the management information must be stored at a place where information may be retrieved or modified. This can be performed using SNMP. In addition, each managed object must also have a name, syntax, and an encoding. The names used must uniquely identify the object. The syntax defines the data type, and the encoding describes how information associated with the managed objects is serialized for transmission between machines.

Management Information Base (MIB)

A Management Information Base (MIB) is a collection of information which define the properties of managed objects. It is organized hierarchically and accessed through the use of a network management protocol such as SNMP. It comprises of managed objects and is identified by object identifiers.

There are two types of managed objects: *scalar* and *tabular*. Scalar objects define a single object instance, and tabular objects define multiple related object instances. Managed objects or MIB objects are comprise of one or more of these object instances. Object instances are essentially variables and are grouped in MIB tables.

Object identifiers or object IDs (OID) uniquely identify managed objects in the MIB hierarchy. The MIB hierarchy is structured as a tree with a nameless root, which levels are assigned by different organizations. The top-level object IDs represent different organizations, while the lower-level object IDs are allocated by associated organizations. The diagram below illustrates an MIB tree where the top-level object IDs are different food organizations, and the lower-level object IDs are the associated food. To identify the 'seed' illustrated in the diagram, a unique ID for the object can be 'fruit.apple.seed' or the equivalent numeric object descriptor, '1.3.1'.

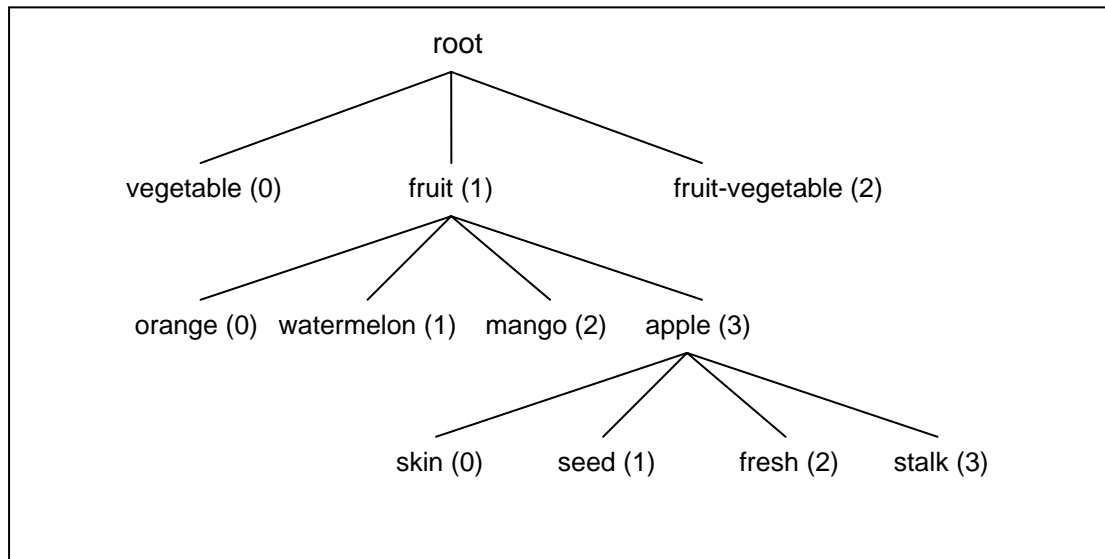


Figure 2. An example of the MIB Tree with Various Hierarchies

According to specification by the Structure of Management Information (SMI), a standard MIB should include the following properties:

- Objects have to be uniquely named
- Objects have to be essential for either fault or configuration management
- Objects have to be general and must not be too device dependant
- Objects have to be maintained in small numbers
- No object should be included that can be easily derived from other objects

Vendors who wish to include managed objects for their own products can define private branches in the MIB tree. Using private MIBs, specific objects can be defined to include extensive information for a more complete management of devices.

Simple Network Management Protocol Version 2 (SNMPv2)

SNMP version 2 (SNMPv2) is an evolution of the initial SNMP (SNMPv1). As with SNMPv1, SNMPv2 functions within the specifications of the Structure of Management Information (SMI).

SNMPv2 offers a number of improvements to SNMPv1, including additional protocol operations and security and enhanced SMI-specific data types. SNMPv2 allows the definition of bit strings which SNMPv1 do not have. In SNMPv1, SNMP only supports 32-bit network address, but in SNMPv2 it also supports other types of addresses. Unlike SNMPv1 which only supports 32-bit counter size, SNMPv2 supports both 64-bit and 32-bit counters.

SNMPv2 uses similar request/response protocol operations as SNMPv1 such as **Get**, **GetNext**, and **Set**. Although both SNMPv1 and SNMPv2 also have **Trap** operation, they do not use the same message format. The Trap operation of SNMPv2 replaces SNMPv1 Trap. SNMPv2 defines also two new protocol operations:

- **GetBulk**. NMS uses the GetBulk operation to retrieve large blocks of data efficiently. GetBulk can fill a response message with as much requested data as will fit.
- **Inform**. NMS uses the Inform operation to send trap information to another NMS and then receive a response.

SNMPv2 is incompatible with SNMPv1. This is because the message formats and protocol operations are different. The two protocol operations, GetBulk and Inform, are not specified or available in SNMPv1 and the header and protocol data unit (PDU) formats are different.

Simple Network Management Protocol Version 3 (SNMPv3)

SNMP version 3 (SNMPv3) is an interoperable standards-based protocol for network management. It provides secure access to devices and combines authentication and encryption packets over the network. SNMPv3 provides the following security features:

- Message integrity. It ensures a packet is not tampered during transmission.
- Authentication. It determines the message comes from a valid source.
- Encryption. The contents of a packet are scrambled so that it can prevent from being seen by any unauthorized source.

SNMPv3 supports different security models and security levels. The security model is an authentication strategy in which a user or group resides. The three security models are SNMPv1, SNMPv2c and SNMPv3. A security level is the permitted level of security within a security model. The combination below determines which security mechanism to employ when handling an SNMP packet.

Model	Level	Authentication	Encryption
v1	noAuthNoPriv	Community String	No
v2c	noAuthNoPriv	Community String	No
v3	noAuthNoPriv	Username	No
v3	authNoPriv	MD5 or SHA	No
v3	authPriv	MD5 or SHA	DES

For SNMPv1 and SNMPv2c, a community string match is used for authentication. For SNMPv3 with noAuthNoPriv level, a username match is used for authentication. For SNMPv3 with authNoPriv level, authentication based on HMAC-MD5 or HMAC-SHA algorithms may be used. HMAC or Keyed-Hashing for Message Authentication is a mechanism for message authentication using cryptographic hash functions. MD5 and SHA are examples of such hash functions.

SNMPv3 also provides DES 56-bit encryption with authentication based on CBC-DES (DES-56) standard. DES refers to Data Encryption Standard and CBC refers to Cipher Block Chaining. CBC is a mode of operation for using DES encryption.

Introduction to Remote Monitoring (RMON)

Remote Monitoring (RMON) is a standard monitoring specification developed by the Internet Engineering Task Force (IETF) in 1992 to support monitoring and protocol analysis. It provides open, comprehensive network fault diagnosis, planning, and performance tuning features for network management. It is designed to collect and process data using remote probe devices. Data collected is used with analysis tools to transform raw data into useful information to help network managers manage their networks and fine tune network performance.

RMON is an SNMP standard for a MIB that controls the remote probes or agents. It uses the agent software embedded in network devices to collect network traffic information and device statistics. The information collected is then recorded in a MIB. Network managers can obtain this information by sending queries to the agent's MIB using a polling process. Information obtained from MIB however only record aggregated statistics and does not provide historical analysis of the daily traffic. If network managers want to have a more comprehensive view on the daily traffic, they will have to continually poll the SNMP agents. Continual polling however has two distinct disadvantages. In large networks, polling can generate substantial network traffic and this can cause serious congestion. Polling can also place heavy burden at the network management console as a result of extensive logging and collection of data from many segments.

RMON helps network managers determine how to segment networks through identifying, analyzing, monitoring, and troubleshooting problems in the network. Strategic proactive management tasks such as baselining and capacity planning are also easier with RMON agents. RMON saves time and manpower by eliminating the need to travel to a problem site, set up equipment, and begins to collect information.

Architecture of RMON

RMON is based on client/server architecture. The 'client' is the application running on the network management station that presents RMON information to the user. The 'server' is the monitoring device which uses a software program called a RMON 'agent' or 'probe' to collect information.

RMON agents are the key element of the monitoring system in the server. Multiple clients can use the agent at the same time. The network managers can configure the agent to offer different views for members of the management team.

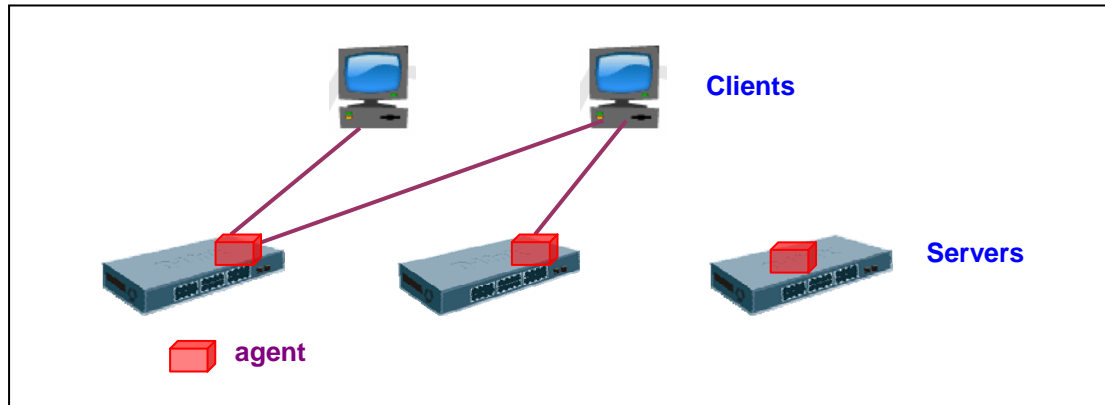


Figure 3. RMON Architecture

There are standalone and embedded agents to choose from. The standalone agents are portable and self-contained in a hardware device. RMON agents can be embedded in network devices such as switches, routers and network interface cards. Agents in routers can monitor activities on the LAN interfaces using remote access. Agents in switches can rove all the ports on the switch. Products with embedded agent however may face some degradation in performance when the agent is used actively. Optional network interface card with basic RMON capabilities that can help off-load network devices' agent activity to conserve resources is available if required.

RMON1 and RMON2

Originally developed for Ethernet and Token Ring LANs, the RMON standard divides monitoring functions into nine groups to support Ethernet topologies and add a tenth group for Token Ring. The RMON standard is created to be deployed as a distributed computing architecture where the agents and probes communicate with a central management station (client) via SNMP. RMON agents have defined SNMP MIB for all nine or ten RMON groups and allows interoperability between different vendors of RMON-based diagnostic tools.

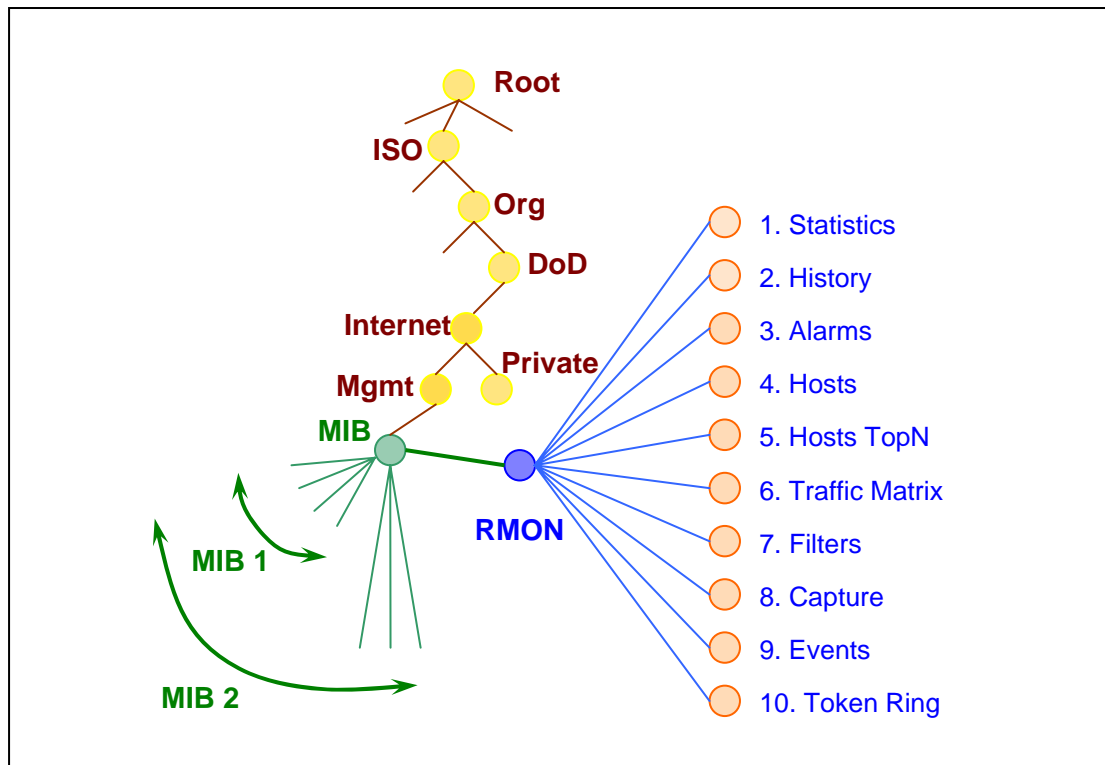


Figure 4. RMON Groups

The basic RMON standard is broadly accepted by the networking and data communications industry. RMON1 however can only specify monitoring and diagnostics of network traffic at the data link layer and do not monitor end-to-end enterprise-wide and application-layer traffic. RMON1 agent is also unable to identify network hosts and sources beyond the router connection even though it is able to view traffic on the local LAN segment.

RMON2 answers the need to analyze traffic and troubleshoot at higher layers. Developed as an extension to RMON1, ROM2 is a complementary technology to RMON1. With RMON2 agents, all RMON groups can now map into major network-layer protocols such as IP, IPX, DECnet, AppleTalk, Banyan VINES to give a complete end-to-end view of the network traffic.

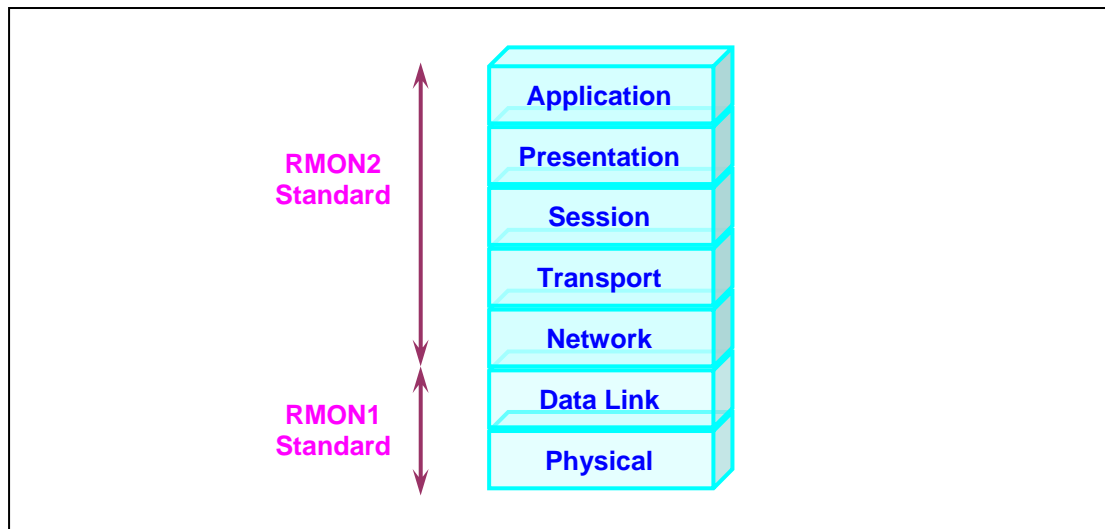


Figure 5. RMON Standard

RMON2 defines the specification for monitoring application-layer traffic. It enables the network managers to monitor network applications such as Telnet, Lotus Notes, Microsoft Mail and others. Using RMON2, the network managers can proactively monitor and troubleshoot any key application-layer traffic in the enterprise network. The RMON groups such as Alarms, Statistics, and History groups may be used for troubleshooting and maintaining network availability based on application-layer traffic.

The following table shows how RMON2 can complement existing RMON management solutions to provide different perspectives to address different network management issues.

Network Management Issue	Relevant OSI Layer	Mgt Standard
Physical errors & utilization	Media Access Control	RMON1
LAN segmentation	Data Link	RMON1
Interconnection of networks	Network	RMON2
Application usage	Application	RMON2

RMON Groups

RMON Ethernet Groups have Ethernet-specific information concerning collisions, runts, and jabbers. The first three groups provide overall monitoring of current activity including detection of possible problems.

1. Statistics

Statistics contains information of network activity measured by the probe for each monitored interface. Network managers can collect information on the packet volumes, broadcast and unicast traffic, packet size distributions and errors.

2. History

Periodic statistical samples from a network can be stored and retrieved when required using History. With user-definable sampling rates and time intervals, network information can be collected over a period of time to compare behavior, build baselines, and perform accurate trend analysis.

3. Alarm

Alarm allows the setting of rising and falling network thresholds and sampling intervals on any counter or integer accessible by any object from the entire SNMP MIB. It periodically takes statistical samples from variables in the probe and compares them with previously configured thresholds. If the monitored variable crosses a threshold, an event is generated.

The next three groups help the network manager with traffic analysis and offer more detailed views of segment behavior.

4. Host

New hosts are discovered by probe when it sees a new media access control (MAC) address in the segment. Host contains statistics associated with each host discovered.

5. Host TopN

HostTopN provides sorted host table statistics and is used to prepare reports describing hosts that top a list ordered by one of the statistics.

6. Matrix

Information about traffic volumes and errors is kept in conversations between sets of two addresses. Matrix stores statistics for conversations. When a new conversation is detected, Matrix creates a new entry in its table.

The next three groups allow finer details to be monitored. Network managers use this information to monitor activities such as application behavior or protocol interactions.

7. Filter

Filter enables packets to be matched by a filter equation, allowing specific information of the matched packets to form a data stream that may be captured.

8. Capture

Capture allows packets to be captured after they flow through a channel and matches filter equation. It defines the number of filtered packets that should be saved and generated as events.

9. Event

Event controls the generation and notification of SNMP TRAPs to the remote client.

10. Token Ring

RMON2 is for Token Ring which is not covered in this document.

The RMON2 MIB extends the capability of the original RMON MIB to include higher-layer protocols for monitoring network and application layer activities. Each group controls a specific RMON2 agent function. The following are additional MIB groups available with RMON2.

11. Protocol Directory

Protocol Directory is the list of protocols the probe has the capability of monitoring. It is the means for a RMON2 application to learn which protocol a specific RMON2 probe can see. Protocol Directory is especially useful when the application and probe are from different vendors.

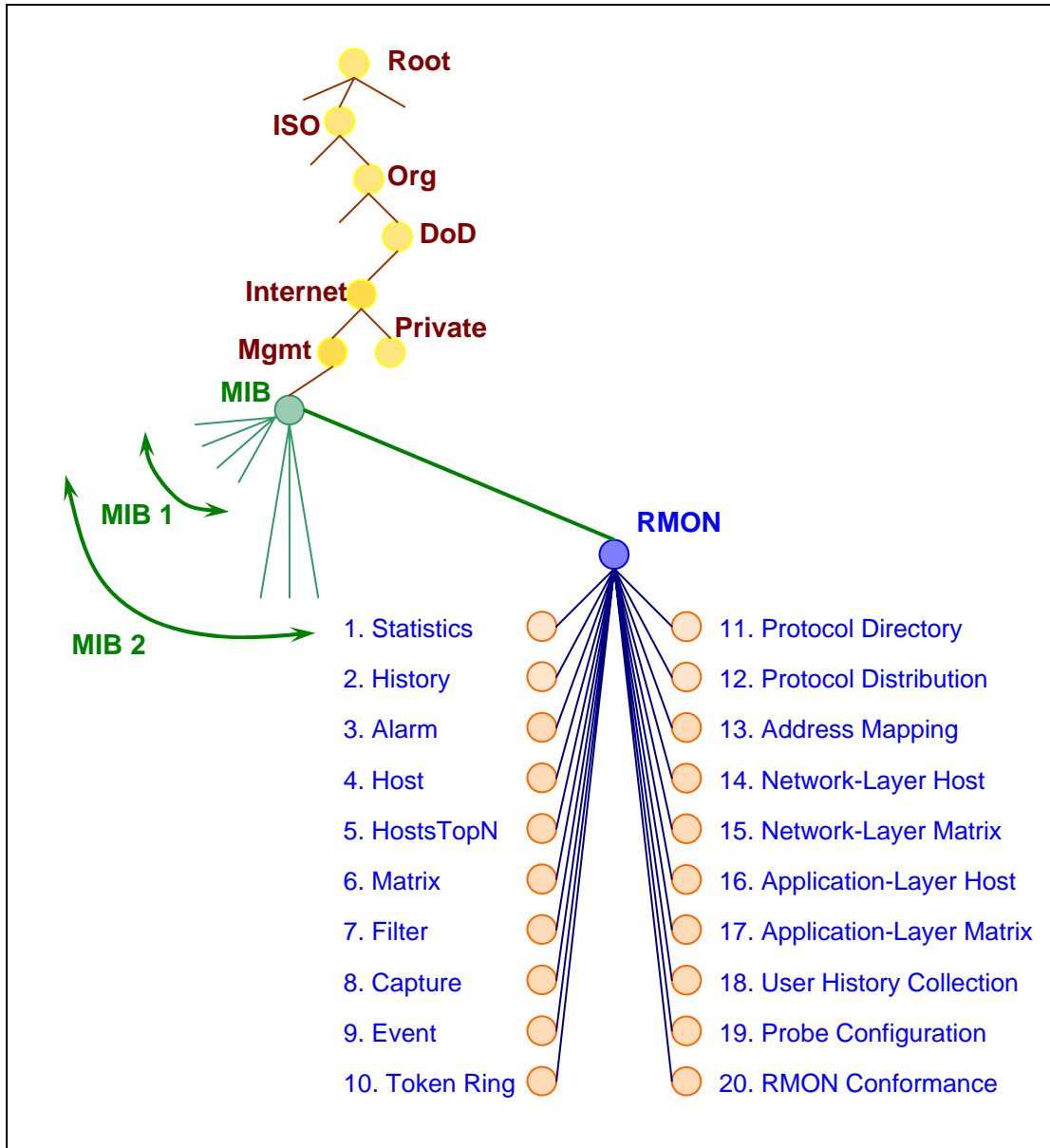


Figure 6. RMON2 Group

12. Protocol Distribution

Protocol Distribution collects traffic statistics such as the number of octets and packets for protocols detected on a network segment to provide distribution and trend information on use of protocols.

13. Address Mapping

Address Mapping maps network-layer addresses to MAC-layer addresses for easier viewing by network managers and interpretation of data.

14. Network-Layer Host

Network-Layer Host is the traffic statistics to and from each discovered host. It is useful for improving the configuration and placement of network resources for optimized performance.

15. Network-Layer Matrix

Network-Layer Matrix is the traffic statistics on conversations between pairs of discovered network addresses or hosts.

16. Application-Layer Host

Application-Layer Host is the traffic statistics to and from each host by protocol, including the application-layer protocols. It provides insight into the use and growth of applications such as Web, Telnet, Lotus Notes, and others.

17. Application-Layer Matrix

Application-Layer Matrix is the traffic statistics on conversations between pairs of hosts by protocol, including application-layer protocols.

18. User History Collection

User History Collection is the periodic statistical samples of user-specified variables, extending the capabilities beyond RMON1 History group that focuses exclusively on Statistics variables.

19. Probe Configuration

Probe Configuration provides a standard way to remotely configure probe parameters such as trap destination and out-of-band management.