Network Working Group
Request for Comments: 3577
Category: Informational

S. Waldbusser
R. Cole
AT&T
C. Kalbfleisch
Verio, Inc.
D. Romascanu
Avaya

August 2003

Introduction to the Remote Monitoring (RMON) Family of MIB Modules

Status of this Memo

Copyright Notice

Abstract

The Remote Monitoring (RMON) Framework consists of a number of interrelated documents. This memo describes these documents and how they relate to one another.

Table of Contents

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

2. Definition of RMON

Remote network monitoring devices, often called monitors or probes, are instruments that exist for the purpose of managing and/or monitoring a network. Often these remote probes are stand-alone devices and devote significant internal resources for the sole purpose of managing a network. An organization may employ many of these devices, up to one per network segment, to manage its internet. In addition, these devices may be used to manage a geographically remote network such as for a network management support center of a service provider to manage a client network, or for the central support organization of an enterprise to manage a remote site.

When the work on the RMON documents was started, this device-oriented definition of RMON was taken quite literally, as RMON devices were purpose-built probes and dedicated to implementing the RMON MIB modules. Soon, cards were introduced that added RMON capability into a network hub, switch or router. RMON also began to appear as a software capability that was added to the software of certain network equipment, as well as software applications that could run on servers or clients. Despite the variety of these approaches, the RMON capability in each serves as a dedicated network management resource available for activities ranging from long-term data collection and analysis or for ad-hoc firefighting.

In the beginning, most, but not all, of RMON's capabilities were based on the promiscuous capture of packets on a network segment or segments. Over time, that mixture included more and more capabilities that did not depend on promiscuous packet capture. Today, some of the newest documents added to the RMON framework allow multiple techniques of data gathering, where promiscuous packet capture is just one of several implementation options.

3. Goals of RMON

- Offline Operation

  There are sometimes conditions when a management station will not be in constant contact with its remote monitoring devices. This is sometimes by design in an attempt to lower communications costs (especially when communicating over a WAN or dialup link), or by accident as network failures affect the communications between the management station and the probe.

  For this reason, RMON allows a probe to be configured to perform diagnostics and to collect statistics continuously, even when communication with the management station may not be possible or efficient. The probe may then attempt to notify the management station when an exceptional condition occurs. Thus, even in circumstances where communication between management station and probe is not continuous, fault, performance, and configuration information may be continuously accumulated and communicated to the management station conveniently and efficiently.

- Proactive Monitoring

  Given the resources available on the monitor, it is potentially helpful for it to continuously run diagnostics and to log network performance. The monitor is always available at the onset of any failure. It can notify the management station of

the failure and can store historical statistical information about the failure. This historical information can be played back by the management station in an attempt to perform further diagnosis into the cause of the problem.

- o Problem Detection and Reporting

  The monitor can be configured to recognize conditions, most notably error conditions, and to continuously check for them. When one of these conditions occurs, the event may be logged, and management stations may be notified in a number of ways.

- o Value Added Data

  Because a remote monitoring device represents a network resource dedicated exclusively to network management functions, and because it is located directly on the monitored portion of the network, the remote network monitoring device has the opportunity to add significant value to the data it collects. For instance, by highlighting those hosts on the network that generate the most traffic or errors, the probe can give the management station precisely the information it needs to solve a class of problems.

- o Multiple Managers

  An organization may have multiple management stations for different units of the organization, for different functions (e.g., engineering and operations), and in an attempt to provide disaster recovery. Because environments with multiple management stations are common, the remote network monitoring device has to deal with more than one management station, potentially using its resources concurrently.

4. RMON Documents

The RMON Framework includes a number of documents. Each document that makes up the RMON framework defines some new useful behavior (i.e., an application) and managed objects that configure, control and monitor that behavior. This section lists those documents and describes the role of each.

One of the key ways to differentiate the various RMON MIB modules is by noting at which layer they operate. Because the RMON MIB modules take measurements and present aggregates of those measurements, there are 2 criteria to quantify for each MIB:

1. At which layers does the MIB take measurements?

   For example, the RMON MIB measures data-link layer attributes (e.g., packets, bytes, errors), while the APM MIB measures application layer attributes (e.g., response time). Supporting measurement at higher layers requires analysis deeper into the packet and many application layer measurements require stateful flow analysis.

2. At which layers does the MIB aggregate measurements?

   This criteria notes the granularity of aggregation. For example, the RMON MIB aggregates its measurements to the link, hardware address, or hardware address pair - all data-link concepts. In contrast, the RMON-2 MIB takes the same data-link metrics (packets, bytes, errors) and aggregates them based on network address, transport protocol, or application protocol.

Note that a MIB may take measurements at one level while aggregating at different levels. Also note that a MIB may function at multiple levels. Figure 1 and Figure 2 show the measurement layers and aggregation layers for each MIB.

Measurement Layers

	Data Link Layer	Network Layer	Transport Layer	Application Layer
RMON-1	X			
TR-RMON	X			
RMON-2	X			
SMON	X			
IFTopN	X			
HCRMON	X			
APM				X
TPM			X	

Figure 1

Aggregation Layers

	Data Link Layer	Network Layer	Transport Layer	Application Layer
RMON-1	X			
TR-RMON	X			
RMON-2		X	X	X
SMON	X			
IFTopN	X			
HCRMON	X			
APM		X	X	X
TPM		X	X	X

Figure 2

4.1. RMON-1

The RMON-1 standard [RFC2819] is focused at layer 2 and provides link-layer statistics aggregated in a variety of ways. In addition, it provides the generation of alarms when thresholds are crossed, as well as the ability to filter and capture packet contents. The components of RMON-1 are:

The Ethernet Statistics Group

The ethernet statistics group contains statistics measured by the probe for each monitored Ethernet interface on this device.

The History Control Group

The history control group controls the periodic statistical sampling of data from various types of network media.

The Ethernet History Group

The ethernet history group records periodic statistical samples from an ethernet network and stores them for later retrieval.

The Alarm Group

The alarm group periodically takes statistical samples from variables in the probe and compares them to previously configured thresholds. If the monitored variable crosses a threshold, an event is generated. A hysteresis mechanism is implemented to limit the generation of alarms.

The Host Group

The host group contains statistics associated with each host discovered on the network. This group discovers hosts on the network by keeping a list of source and destination MAC Addresses seen in good packets promiscuously received from the network.

The HostTopN Group

The hostTopN group is used to prepare reports that describe the hosts that top a list ordered by one of their statistics. The available statistics are samples of one of their base statistics over an interval specified by the management station. Thus, these statistics are rate based. The management station also selects how many such hosts are reported.

The Matrix Group

The matrix group stores statistics for conversations between sets of two MAC addresses. As the device detects a new conversation, it creates a new entry in its tables.

The Filter Group

The filter group allows packets to be matched by a filter equation. These matched packets form a data stream that may be captured or may generate events.

The Packet Capture Group

The Packet Capture group allows packets to be captured after they flow through a channel.

The Event Group

The event group controls the generation and notification of events from this device.

4.2. Token Ring Extensions to RMON MIB

Some of the functions defined in the RMON-1 MIB were defined specific to Ethernet media. In order to operate the functions on Token Ring Media, new objects needed to be defined in the Token Ring Extensions to RMON MIB [RFC1513]. In addition, this MIB defines additional objects that provide monitoring functions unique to Token Ring.

The components of the Token Ring Extensions to RMON MIB are:

The Token Ring Statistics Groups

The Token Ring statistics groups contain current utilization and error statistics. The statistics are broken down into two groups, the Token Ring Mac-Layer Statistics Group and the Token Ring Promiscuous Statistics Group. The Token Ring Mac-Layer Statistics Group collects information from the Mac Layer, including error reports for the ring and ring utilization of the Mac Layer. The Token Ring Promiscuous Statistics Group collects utilization statistics from data packets collected promiscuously.

The Token Ring History Groups

The Token Ring History Groups contain historical utilization and error statistics. The statistics are broken down into two groups, the Token Ring Mac-Layer History Group and the Token Ring Promiscuous History Group. The Token Ring Mac-Layer History Group collects information from the Mac Layer, including error reports for the ring and ring utilization of the Mac Layer. The Token Ring Promiscuous History Group collects utilization statistics from data packets collected promiscuously.

The Token Ring Ring Station Group

The Token Ring Ring Station Group contains statistics and status information associated with each Token Ring station on the local ring. In addition, this group provides status information for each ring being monitored.

The Token Ring Ring Station Order Group

The Token Ring Ring Station Order Group provides the order of the stations on monitored rings.

The Token Ring Ring Station Config Group

The Token Ring Ring Station Config Group manages token ring stations through active means. Any station on a monitored ring may be removed or have configuration information downloaded from it.

The Token Ring Source Routing Group

The Token Ring Source Routing Group contains utilization statistics derived from source routing information optionally present in token ring packets.

4.3. The RMON-2 MIB

The RMON-2 MIB [RFC2021] extends the architecture defined in RMON-1, primarily by extending RMON analysis up to the application layer.

The components of the RMON-2 MIB are:

The Protocol Directory Group

Every RMON-2 implementation will have the capability to parse certain types of packets and identify their protocol type at multiple levels. The protocol directory presents an inventory of those protocol types the probe is capable of monitoring, and allows the addition, deletion, and configuration of protocol types in this list.

Protocol Distribution Group

This function controls the collection of packet and octet counts for any or all protocols detected on a given interface. An NMS can use this table to quickly determine bandwidth allocation utilized by different protocols.

Address Mapping Group

This function lists MAC address to network address bindings discovered by the probe and on which interface they were last seen.

Network Layer Host Group

This function counts the amount of traffic sent from and to each network address discovered by the probe.

Network Layer Matrix Group

This function counts the amount of traffic sent between each pair of network addresses discovered by the probe.

Application Layer Host Group

This function counts the amount of traffic, by protocol, sent from and to each network address discovered by the probe.

Application Layer Matrix

This function counts the amount of traffic, by protocol, sent between each pair of network addresses discovered by the probe.

User History

This function allows an NMS to request that certain variables on the probe be periodically polled and for a time-series to be stored of the polled values. This builds a user-configurable set of variables to be monitored (not to be confused with data about users).

Probe Configuration

This group contains configuration objects that configure many aspects of the probe, including the software downloaded to the probe, the out of band serial connection, and the network connection.

4.4. RMON MIB Protocol Identifiers

The RMON-2 MIB identifies protocols at any layer of the 7 layer hierarchy with an identifier called a Protocol Identifier, or ProtocolID for short. ProtocolIDs also identify the particular configuration of layering in use, including any arbitrary encapsulations. The RMON MIB Protocol Identifiers document [RFC2896] is a companion document to the RMON-2 MIB that defines a number of well-known protocols. Another document, the RMON MIB Protocol Identifiers Macros [RFC2895], defines a macro format for the description of these well-known protocols and others that may be described in the future.

As the RMON Framework has grown, other documents have been added to the framework that utilize ProtocolIDs.

4.5. Remote Network Monitoring MIB Extensions for Switched Networks (SMON MIB)

Switches have become pervasive in today’s networks as a form of broadcast media. SMON [RFC2613] provides RMON-like functions for the monitoring of switched networks.

Switches today differ from standard shared media protocols because:

1) Data is not, in general, broadcast. This MAY be caused by the switch architecture or by the connection-oriented nature of the data. This means, therefore, that monitoring non-broadcast traffic needs to be considered.

2) Monitoring the multiple entry and exit points from a Switching device requires a vast amount of resources - memory and CPU, and aggregation of the data in logical packets of information, determined by the application needs.

3) Switching incorporates logical segmentation such as Virtual LANs (VLANs).

4) Switching incorporates packet prioritization.

5) Data across the switch fabric can be in the form of cells. Like RMON, SMON is only concerned with the monitoring of packets.

Differences such as these make monitoring difficult. The SMON MIB provides the following functions that help to manage switched networks:

smonVlanStats

This function provides traffic statistics per Virtual LAN for 802.1q VLANs.

smonPrioStats

This function provides traffic statistics per priority level for 802.1q VLANS.

dataSourceCaps

This function identifies all supported data sources on a SMON device. An NMS MAY use this table to discover the RMON and Copy Port attributes of each data source.

portCopyConfig

Many network switches provide the capability to make a copy of traffic seen on one port and sending it out to another port for management purposes. This occurs in addition to any copying performed during the normal forwarding behavior of the switch.

The portCopyConfig function provides control of the port copy functionality in a device.

4.6. RMON MIB Extensions for Interface Parameters Monitoring (IFTOPN)

Many network switches contain hundreds of ports, many with only one attached device. A common operation when managing such a switch is to sort the interfaces by one of the parameters (e.g., to find the most highly utilized interface). If the switch contains many interfaces it can be expensive and time consuming to download information for all interfaces to sort it on the NMS. Instead, the ifTopN MIB [RFC3144] allows the sorting to occur on the switch and for only the top interfaces to be downloaded.

4.7. RMON Extensions for Differentiated Services (DSMON MIB)

This MIB [RFC3287] defines extensions of RMON for monitoring the traffic usage of Differentiated Services [RFC2474] codepoint values. The 6-bit DiffServ codepoint portion (DSCP) of the Type of Service (TOS) octet in the IP header provides for 64 different packet treatments for the implementation of differentiated network devices. DSMON-capable RMON probes collect and aggregate statistics based on the inspection of the DSCP value in monitored packets.

The DSMON MIB defines a DSCP counter aggregation mechanism to reduce the total number of counters by configuring the agent to internally aggregate counters based on the DSCP value. This mechanism is designed to overcome the agent data collection limitation, perform data reduction at the agent and applications level, and optimize the application for cases in which some codepoint values are not used, or lead to similar packet treatment in the monitored network domain.

The components of the DSMON MIB are:

The Aggregate Control Group

The Aggregate Control Group enables the configuration of the counter aggregation groups.

The DSMON Statistics Group

The DSMON Statistics Group contains per counter aggregation group distribution statistics for a particular RMON data source.

The DSMON Protocol Distribution Group

The DSMON Protocol Distribution Group reports per counter aggregation distribution statistics for each application protocol detected on a particular RMON data source.

The DSMON Host Group

The DSMON Host Group contains host address distribution statistics for each counter aggregation group, detected on a particular RMON data source.

The DSMON Capabilities Group

The DSMON Capabilities Group reports the DSMON MIB functional capabilities of the agent implementation.

The DSMON Matrix Group

The DSMON Matrix Group contains host address pair distribution statistics for each counter aggregation group, detected on a particular RMON data source.

4.8. RMON for High Capacity Networks (HCRMON MIB)

This MIB [RFC3272] defines extensions to RMON for use on high capacity networks. Except for the mediaIndependentTable, each of the tables in this MIB adds high capacity capability to an associated table in the RMON-1 MIB or RMON-2 MIB.

The mediaIndependentTable provides media independent utilization and error statistics for full-duplex and half-duplex media. Prior to the existence of the HCRMON MIB, a new table needed to be created for RMON monitoring of each data-link layer media. These tables included many statistical attributes of the media, including packet and octet counters that are independent of the media type. This was not optimal because there was no way to monitor media types for which a media-specific table had not been defined. Further, there were no common objects to monitor media-independent attributes between media types.

In the future, for media other than ethernet and token ring, the mediaIndependentTable will be the source for media-independent statistics. Additional media-specific tables may be created to provide attributes unique to particular media, such as error counters.

4.9. Application Performance Measurement MIB (APM MIB)

The APM MIB [APM] provides analysis of application performance as experienced by end-users.

Application performance measurement measures the quality of service delivered to end-users by applications. With this perspective, a true end-to-end view of the IT infrastructure results, combining the performance of the application, desktop, network, and server, as well as any positive or negative interactions between these components.

Despite all the technically sophisticated ways in which networking and system resources can be measured, human end-users perceive only two things about an application: availability and responsiveness.

Availability - The percentage of the time that the application is ready to give a user service.

Responsiveness - The speed at which the application delivers the requested service.

The APM MIB includes the following functions:

The APM Application Directory Group

The APM Application Directory group contains configuration objects for every application or application verb monitored on this system.

The APM User Defined Applications Group

The APM User Defined Applications Group contains objects that allow for the tracking of applications or application verbs that are not registered in the protocolDirectoryTable.

The APM Report Group

The APM Report Group is used to prepare regular reports that aggregate application performance by flow, by client, by server, or by application.

The APM Transaction Group

The APM Transaction Group is used to show transactions that are currently in progress and ones that have ended recently, along with their responsiveness metric.

One important benefit of this table is that it allows a management station to check on the status of long-lived transactions. Because the apmReport and apmException mechanisms act only on transactions that have finished, a network manager may not have visibility for some time into the performance of long-lived transactions, such as streaming applications, large data transfers, or (very) poorly performing transactions. In fact, by their very definition, the apmReport and apmException mechanisms only provide visibility into a problem after nothing can be done about it.

The APM Exception Group

The APM Exception Group is used to generate immediate notifications of transactions that cross certain thresholds. The apmExceptionTable is used to configure which thresholds are to be checked for which types of transactions. The apmTransactionResponsivenessAlarm notification is sent when a transaction occurs with a responsiveness that crosses a threshold.

The apmTransactionUnsuccessfulAlarm notification is sent when a transaction, for which exception checking was configured, fails.

The APM Notification Group

The APM Notification Group contains 2 notifications that are sent when thresholds in the APM Exception Table are exceeded.

4.10. RMON MIB Protocol Identifier Reference Extensions

The protocol identifier defined in RMON-2 [RFC2021] can identify any protocol at any layer and its encapsulation. The protocol identifier macro document [RFC2896] defines a convenient human readable and machine parseable format for documenting well-known protocols.

For the most part, the protocol identifiers used by RMON-2 implementations have described protocols at any layer, including the application layer, but have not gone any deeper into the application. In order to differentiate an application's behavior while performing different tasks (logging in vs. downloading, for example), it is important to have a separate protocol identifier for each application "verb". The macro defined in [RFC2896] is inconvenient for defining application verbs because it assumes that most protocols are identified by an integer type field and many or most applications use other means for identifying verbs, including character strings.

These extensions define another macro for defining application verbs that are children of an application. The parent application can be defined with the original protocol identifier macro and the application verbs are defined with the new macro.

4.11. Transport Performance Metrics MIB (TPM MIB)

The TPM MIB [TPM] monitors selected performance metrics and statistics derived from the monitoring of network packets and sub-application level transactions. The MIB is defined to compliment the APM reports by providing a 'drill-down' capability to better understand selected applications' performance. The metrics are defined through reference to existing IETF, ITU and other standards organizations' documents. The monitoring covers both passive and active traffic generation sources.

The TPM MIB includes the following functions:

The tpmCapabilities Group

The tpmCapabilitiesGroup contains objects and tables that show the measurement protocol and metric capabilities of the agent.

The tpmAggregateReports Group

The tpmAggregateReportsGroup is used to provide the collection of aggregated statistical measurements for the configured report intervals.

The tpmCurrentReports Group

The tpmCurrentReportsGroup is used to provide the collection of uncompleted measurements for the current configured report for those transactions caught in progress. A history of these transactions is also maintained once the current transaction has completed.

The tpmExceptionReports Group

The tpmExceptionReportsGroup is used to link immediate notifications of transactions that exceed certain thresholds defined in the apmExceptionGroup [APM]. This group reports the aggregated sub-application measurements for those applications exceeding thresholds.

4.12. Synthetic Sources for Performance Monitoring MIB (SSPM MIB)

The Synthetic Sources for Performance Monitoring MIB [SSPM] covers the artificial generation of a) application-level, b) transport-level, and c) link-level traffic for the purpose of monitoring system performance. There are situations where it is useful to be able to control the generation of synthetic traffic when evaluating system performance. There are other situations where system performance evaluation can rely upon naturally generated application-level traffic, in which case one needs only monitor existing traffic and not instrument synthetic traffic. The SSPM MIB provides the ability to configure and control the generation of this synthetic traffic.

4.13. RMON MIB Extensions for High Capacity Alarms

There is a need for a standardized way of providing the same type of alarm thresholding capabilities for Counter64 objects, as already exists for Counter32 objects. The RMON-1 alarmTable objects and RMON-1 notification types are specific to 32-bit objects, and cannot be used to properly monitor Counter64-based objects. Extensions to these existing constructs are needed which explicitly support Counter64-based objects. These extensions are completely independent of the existing RMON-1 alarm mechanisms.

This MIB [RFC3434] contains the following functions:

The hcAlarmControlObjects group

Controls the configuration of alarms for high capacity MIB object instances.

The hcAlarmCapabilities group

Describes the high capacity alarm capabilities provided by the agent.

The hcAlarmNotifications group

Provides new rising and falling threshold notifications for high capacity objects.

4.14. Real-Time Application Quality of Service Monitoring (RAQMON) MIB

There is a need to extend the RMON framework to monitor end devices such as IP phones, pagers, Instant Message Clients, mobile phones, and PDA devices. This memo proposes an extension of RMON Framework to allow Real-time Application QoS information of these types of end

devices to be retrieved with SNMP, independent of the technology used to perform the measurements. An end-to-end user experience of the quality of service (QoS) and performance for such an application is a combination of device performance, transport network performance and specific application context.

RAQMON [RAQMON-FRAMEWORK] defines a common framework to identify a set of application QoS parameters and a reporting mechanism using a common protocol data unit (PDU) format used between a RAQMON Data Source (RDS) and a RAQMON Report Collector (RRC) to report QOS statistics using RTCP and SNMP as underlying transport protocol.

See the RAQMON MIB [RAQMON-MIB] for more information about its components.

5. RMON Framework Components

The collection of documents in the RMON Framework are associated by 1) A common purpose and similar collection methodologies; and, 2) Use of common infrastructure components.

These common infrastructure components are:

- MediaIndependent Table
- Protocol Directory
- appDirectory
- DataSource
- Capabilities
- Control Tables

5.1. MediaIndependent Table

While many data-link media types exist and they each have unique features, there are many statistics that are common across most media. For example, counts of packets and octets are interesting for most media. The media independent table contains the most common such statistics and forms a super class from which specific interface types are inherited. This means that the common statistics can be monitored even for media types that are unknown.

For example, if the mediaindependentTable had existed prior to the definition of the etherStatsTable, the etherStatsTable could have omitted the etherStatsDropEvents, etherStatsOctets, etherStatsPkts objects.

The Media Independent Table is defined in the High Capacity RMON MIB [RFC3434].

5.2. Protocol Directory

The second of the RMON infrastructure components is the Protocol Directory Group defined in the RMON-2 MIB [RFC2021]. The main objective of RMON-2 was to extend the remote network monitoring agents capabilities beyond the link layer to higher level protocol monitoring. This required a means to globally identify individual protocol encapsulations. This capability is provided by the Protocol Directory Group, specifically the protocolDirID found in the protocolDirTable in the RMON-2 MIB.

The Protocol Directory allows the agent to provide an inventory of the protocols that the agent can decode, count, categorize and time. The directory and its objects are designed to allow for the addition, deletion and configuration of the protocol encapsulations in the directory list. Protocol Directory entries are identified primarily by an object called the protocolDirID. The protocolDirID is a hierarchically formatted OCTET STRING that globally identifies individual protocol encapsulations. A protocol descriptor macro has been defined in RFC 2895 [RFC2895] to describe the various protocol layers supported in the protocolDirID protocol hierarchy. The protocolDirID is defined as a tree built up from successive protocol encapsulations. Each layer is identified by a 4-octet identifier that identifies the child protocol within the context of the parent protocol identified by the preceding identifiers.

Associated with each protocol layer in the protocolDirID is a 1-octet parameter field. Each parameter identifies potential options specific to that protocol, such as the agent's capability to count fragmented packets correctly and to track sessions for port mapped protocols, e.g., TFTP. These 1-octet parameter fields are concatenated, in order, in the protocolDirParameters object.

The protocolDirTable index is comprised of the protocolDirID, the protocolDirParameters and their associated length fields. The index format is shown in Figure 3.

Figure 3: the protocolDirTable INDEX format.

An example protocolDirTable INDEX for SNMP over UDP over IP over Ethernet is:

```
 16.0.0.0.1.0.0.8.0.0.0.0.17.0.0.0.161.4.0.0.0.0

|  |       |       |        |         | |       |
+--+-------+-------+--------+---------+-+-------+
 c  ether2    ip      udp      snmp    c  param.

 c = 1-subidentifier count field
```

Figure 4: A protocolDirTable INDEX example for SNMP over UDP over IP over Ethernet.

The set of defined protocol layers currently described is found in RFC 2896 [RFC2896]. RFC 2895 [RFC2895] defines a process for submitting new protocols to add to the currently defined set. Periodic updates to RFC 2896 will be published to incorporate new protocol definitions that have been submitted. In fact, RFC 2896 is the second version of the defined protocol macros, obsoleting RFC 2074 [RFC2074]. RFC 2895 also defines how to handle protocols that do not map into this well-defined tree hierarchy built up from encapsulation protocol identifiers. An example of such a protocol encapsulation is RTP, which is mapped to specific UDP ports through a separate signaling mechanism. These are handled by the ianaAssigned protocols, as described in RFC 2895.

The protocolDirTable is defined (and used) in the RMON-2 MIB [RFC2021], and is being used in other RMON WG MIBs, as well as other IETF defined MIBs. Examples include the APM MIB [APM], the TPM MIB [TPM] and the SSPM MIB [SSPM].

As mentioned in previous sections, the protocolDirID is being extended in two ways. First, work is underway on a new set of protocol descriptor macros to extend the protocol encapsulation model to identify application layer verbs [RFC3395]. This extension was motivated by the work on the APM MIB and the TPM MIB. Second, the APM MIB defines the apmAppDirectoryTable that provides a directory of applications that the agent can process. This is discussed further in the following section. Combined, these extensions allow:

+ The APM MIB to define and monitor the end-user’s view of application performance.

+ The TPM MIB to clearly specify the sub-transactions that comprise the application it monitors through the tpmTransMetricDirTable.

+ The SSPM MIB to generate synthetic application transactions by importing the appLocalIndex from the APM MIB.

5.3. Application Directory and appLocalIndex

APM, TPM and related applications collect certain types of statistics for each application or application verb they are decoding. Some applications and application verbs are defined in the protocol directory and thus get their own protocolID and a corresponding protocolDirLocalIndex. Other application verbs are defined more dynamically by entries in the apmHttpFilterTable or apmUserDefinedAppTable. These dynamically defined applications do not have protocolDirID's assigned to them.

The APM MIB [APM] defines an important index called the appLocalIndex. For all application monitoring in the APM and TPM MIBs, applications are identified by integer values of the appLocalIndex. However, there is no single registry of applications (as there is for protocols) because there are a few different mechanisms through which an application may be registered. For each value of appLocalIndex, a corresponding entry will exist in one of several tables:

1. The protocolDirTable - Some values of appLocalIndex correspond to protocolDirLocalIndex values assigned in the protocolDirTable. Each of these corresponds to a protocol defined by a protocolID.

2. The apmHttpFilterTable - Some values of appLocalIndex correspond to apmHttpFilterAppLocalindex values assigned in the apmHttpFilterTable. Each of these corresponds to an application verb defined as a set of HTTP transactions that match a set of filters.

3. The apmUserDefinedAppTable - Some values of appLocalIndex correspond to index values of the apmUserDefinedAppTable. Each of them corresponds to an application or application verb defined in a user-defined way.

Each value of appLocalIndex will only be registered in one of these tables. In effect, the appLocalIndex number space is the union of these number spaces, where these tables must work together to avoid assigning overlapping (duplicate) appLocalIndexes.

Each unique appLocalIndex value is also registered in the apmAppDirectoryTable, where a number of attributes of the application may be configured.

5.4. Data Source

Most RMON functions use a DataSource as a pointer to the entity from which data is to be collected. The DataSource is an object identifier that identifies one of three types of data sources:

ifIndex.<I>

Traditional RMON dataSources. Called 'port-based' for ifType.<I> not equal to 'propVirtual(53)'. <I> is the ifIndex value.

smonVlanDataSource.<V>

A dataSource of this form refers to a 'Packet-based VLAN' and is called a 'VLAN-based' dataSource. <V> is the VLAN ID as defined by the IEEE 802.1Q standard. The value is between 1 and 4094 inclusive, and it represents an 802.1Q VLAN-ID with a global scope within a given bridged domain, as defined by 802.1Q.

entPhysicalEntry.<N>

A dataSource of this form refers to a physical entity within the agent and is called an 'entity-based' dataSource. <N> is the value of the entPhysicalIndex in the entPhysicalTable.

5.5. Capabilities

Probe Capabilities objects have been introduced in the RMON MIB modules with the goal of helping applications determine the capabilities of the different probes in the domain. These objects use a BITS syntax (with the exception of some of the objects in the TPM and SSPM MIBs), and list in an explicit manner the MIB groups supported by the probe, as well as functional capabilities of the specific RMON agents. By reading the values of these objects, it is possible for applications to know which RMON functions are usable without going through a trial-and-error process that can result in loss of time and bandwidth in the operational flow. These objects have the MAX-ACCESS of read-only, which defines their use as an indication of what is supported by a probe, and not a means to configure the probe for operational modes. An RMON agent SHOULD initiate the capabilities objects at agent initialization and SHOULD NOT modify the objects during operation.

The probeCapabilities object in the RMON-2 MIB describes the capabilities of probes that support RMON, Token-Ring RMON and RMON-2.

The smonCapabilities object in the SMON MIB describes the SMON-specific capabilities of probes that support the SMON MIB.

The dataSourceCapsTable in the SMON MIB defines the capabilities of the SMON data sources on probes that support the RMON MIB.

The interfaceTopNCaps object in the Interface TopN MIB defines the sorting capabilities supported by an agent that supports the Interface TopN MIB.

The dsmonCapabilities object in the DSMON MIB provides an indication of the DSMON groups supported by an agent that supports the DSMON MIB.

The tpmCapabilitiesGroup contains objects and tables, which show the measurement protocol and metric capabilities of an agent that supports the TPM MIB.

The sspmCapabilitiesTable indicates whether a device supporting the SSPM MIB supports SSPM configuration of the corresponding AppLocalIndex.

The hcAlarmCapabilities object provides an indication of the high capacity alarm capabilities supported by an agent that supports the HC-Alarm MIB.

5.6. Control Tables

Due to the complex nature of the available functions in the RMON MIB modules, these functions often need user configuration. In many cases, the function requires parameters to be set up for a data collection operation. The operation can proceed only after these parameters are fully set up.

Many functional groups in the RMON MIBs have one or more tables in which to set up control parameters, and one or more data tables in which to place the results of the operation. The control tables are typically read-write in nature, while the data tables are typically read-only. Because the parameters in the control table often describe resulting data in the data table, many of the parameters can be modified only when the control entry is invalid. Thus, the method for modifying these parameters is to invalidate the control entry, causing its deletion and the deletion of any associated data entries, and then create a new control entry with the proper parameters. Deleting the control entry also gives a convenient method for reclaiming the resources used by the associated data.

To facilitate control by multiple managers, resources have to be shared among the managers. These resources are typically the memory and computation resources that a function requires.

Two facilities are used to ease cooperation between multiple managers as they create and use control tables. The first is the use of EntryStatus or RowStatus objects that guarantee that two managers can avoid creating the same control entry. The second is the use of OwnerString objects in control tables that provides the following benefits:

1. Provides information to facilitate sharing of already existing control entries instead of creating a new but identical entry.

2. Provides information to allow the ultimate human owners of control entries to identify each other so they can cooperate in cases of conflict over resources.

3. Provides information to allow software to identify control entries that it owns but has forgotten about (e.g., due to a crash or other error) so that it can re-use or free them.

4. Provides information to allow an administrator to make an informed decision to override someone else's control entry when circumstances make it necessary.

5. Provides information to identify control entries that are set up automatically when the device starts up.

See the RMON MIB [RFC2819] for further information on the use of control tables, EntryStatus/RowStatus, and OwnerStrings.

6. Relationship of the SSPM MIB with the APM and TPM MIBs

While APM and TPM may monitor actual traffic generated by end-users on the network, they may also monitor synthetically generated traffic. The SSPM MIB provides a mechanism for the generation of synthetic traffic but no mechanism for monitoring - the task of monitoring the generated traffic is deferred to the APM and TPM MIBs.

Figure 5 shows an overview of the components of the SSPM MIB architecture, including the roles played by the APM and TPM MIBs. The RMON documents address the "Control-Level" in this diagram and some aspects of the "Synchronization Control-Level". The underlying "Instrumentation-Level" is implementation dependent and outside the domain of the RMON specifications.

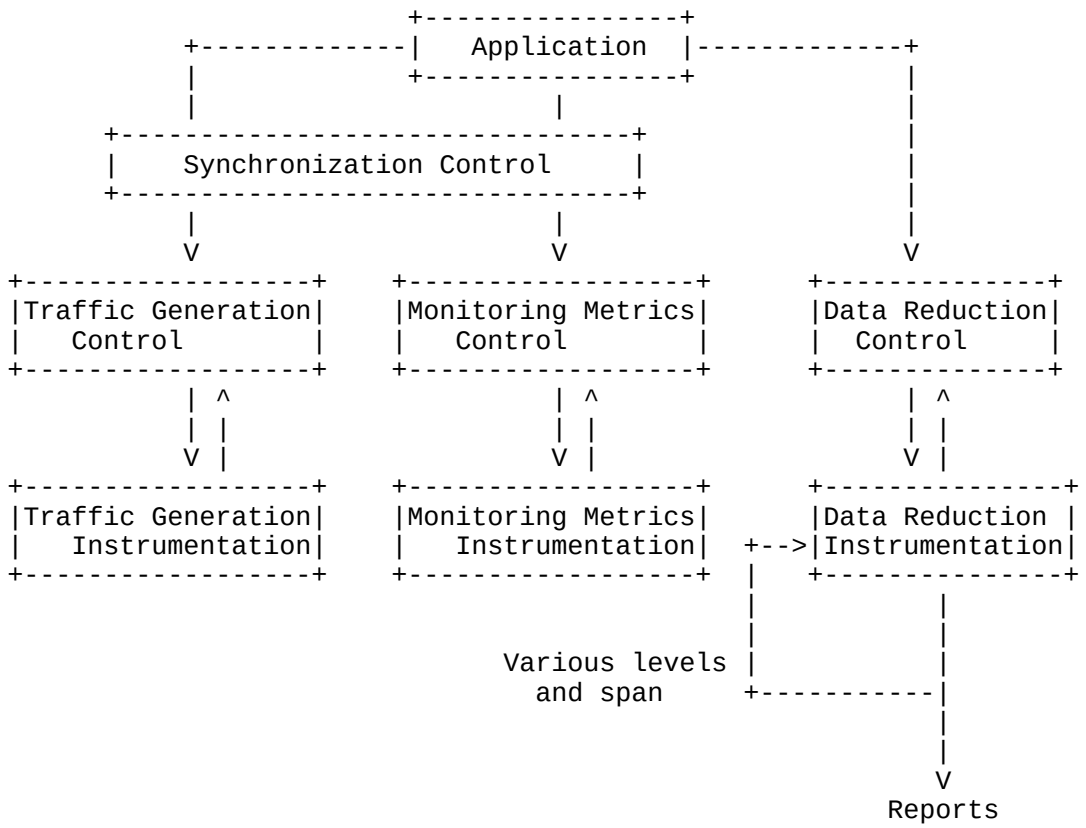

Figure 5: An SSPM Performance Monitoring System

It is the responsibility of the network management application to coordinate the individual aspects of the performance management system.

Within the APM, TPM, and SSPM set of RMON MIB modules:

+ APM MIB [APM] is responsible for the aspects of the "Monitoring Metrics Control" directly related to the end-user's perceived application-level performance. The APM MIB also handles aspects of "Data Reduction Control" and "Reports". Finally, when TPM MIB relies upon the control tables in the APM MIB for its own control, then APM MIB is providing some aspects of "Synchronization Control" of the reports from these two MIBs.

+ TPM MIB [TPM] is responsible for the aspects of the "Monitoring Metrics Control". TPM MIB also handles aspects of "Data Reduction Control" and "Reports" related to sub-application-level transactions. Synchronization control with APM MIB is provided by opting to rely on the APM MIB control tables within the TPM MIB.

+ SSPM MIB [SSPM] is responsible for the "Traffic Generation Control" in the event that synthetic traffic is to be monitored. The other, most common, option is to monitor natural, user-generated traffic.

The "Monitor Metrics Control" is essentially hard-coded in the APM MIB. Within the TPM MIB, a metrics table is used to identify the metrics monitored within a specific implementation of the TPM MIB. The "Data Reduction Control" is essentially hard-coded within the MIB structure of the APM MIB and the TPM MIB. These MIBs strictly specify the statistics to be reported within a set of report tables.

Both the TPM MIB and the SSPM MIB rely upon the APM MIB's appLocalIndex to specify the application being monitored or generated. The APM MIB provides the end-user view of the application performance, e.g., the Whois transaction time. The TPM MIB, through its tpmTransMetricDirTable, identifies a set of sub-application level transactions and their metrics, which are associated with the application. E.g., an implementation of the TPM MIB could report the DNS lookup time, the TCP connect time (to the Whois Server), the Whois Req/Resp download time. The SSPM MIB could be configured to generate synthetically, these Whois transactions.

The testing model then is to first configure the traffic generation instrumentation through the SSPM MIB control function. This defines aspects of the synthetic traffic such as application type, targets, etc. Once the traffic generation is configured, the network management application can setup the monitoring instrumentation through the APM MIB and TPM MIB. These control the reporting periods, the type of data aggregation, etc. Once the tests are complete, the network management application retrieves the reports from the monitoring metrics control MIBs, e.g., APM MIB and TPM MIB.

7. Acknowledgements

This memo is a product of the RMON MIB working group. In addition, the authors gratefully acknowledge the contributions by Lester D'Souza of NetScout Systems, Inc.

8. References

8.1. Normative References

[RFC2819] Waldbusser, S., "Remote Network Monitoring Management Information Base", STD 59, RFC 2819, May 2000.

8.2. Informative References

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2578] McCloghrie, K., Perkins, D. and J. Schoenwaelder, Eds., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D. and J. Schoenwaelder, J., Eds., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D. and J. Schoenwaelder, J., Eds., "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC1513] Waldbusser, S., "Token Ring Extensions to the Remote Network Monitoring MIB", RFC 1513, September 1993.

[RFC2021] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2 using SMIv2", RFC 2021, January 1997.

[RFC2895] Bierman, A., Bucci, C. and R. Iddon, "Remote Network Monitoring Management Information Base Protocol Identification Reference", RFC 2895, August 2000.

[RFC2896] Bierman, A., Bucci, C. and R. Iddon, "Remote Network Monitoring MIB Protocol Identifier Macros", RFC 2896, August 2000.

[RFC2613] Waterman, R., Lahaye, B., Romascanu, D. and S. Waldbusser, "Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0", RFC 2613, June 1999.

[RFC3144] Waldbusser, S., "Remote Monitoring MIB Extensions for Interface Parameters Monitoring", RFC 3144, August 2001.

[RFC3287] Bierman, A., "Remote Monitoring MIB Extensions for Differentiated Services", RFC 3287, July 2002.

[RFC3273] Waldbusser, S., "Remote Network Monitoring Management Information Base for High Capacity Networks", RFC 3273, July 2002.

[APM] Waldbusser, S., "Application performance measurement MIB", Work in Progress.

[RFC3395] Bierman, A., Bucci, C., Dietz, R. and A. Warth, "Remote Network Monitoring MIB Protocol Identifier Reference Extensions", RFC 3395, September 2002.

[TPM] Dietz, R. and R.G.Cole, "Application Performance Measurement Framework Transport Performance Metrics MIB", Work in Progress.

[SSPM] Kalbfleisch, K., Cole, R.G. and D. Romascanu, "Definition of Managed Objects for Synthetic Sources for Performance Monitoring Algorithms", Work in Progress.

[RFC3434] Bierman, A. and K. McCloghrie, "Remote Monitoring MIB Extensions for High Capacity Alarms", RFC 3434, December 2002.

[RFC2233] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB Using SMIv2", RFC 2233, November 1997.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC2330] Paxson, V., Almes, G., Mahdavi, J. and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.

[OWDP] Shalunov, S., Teitelbaum, B. and M. Zekauskas, "A One-way Active Measurement Protocol", Work in Progress.

[RAQMON-FRAMEWORK] Siddiqui, A., Romascanu, D. and E. Golovinsky, "Real-time Application Quality of Service Monitoring (RAQMON) Framework", Work in Progress.

[RAQMON-MIB] Siddiqui, A., Romascanu, D., Golovinsky, E. and R. Smith, "Real-Time Application Quality of Service Monitoring (RAQMON) MIB", Work in Progress.

9. Security Considerations

This document is a description of existing documents and as such it does not have any security impact. In order to understand the security-related issues of the different RMON documents, the reader is directed to the Security Considerations sections of the respective documents.

10. Authors' Addresses

Steve Waldbusser

Phone: +1 650-948-6500
Fax: +1 650-745-0671
EMail: waldbusser@nextbeacon.com

Carl W. Kalbfleisch
NTT/VERIO
8700 Stemmons Freeway, Suite 211
Dallas, TX 75247
United States

Phone: +1 972-906-2034
EMail: cwk@verio.net

Robert G. Cole
AT&T Labs
Network Design and Performance Analysis Department
330 Saint John Street, 2nd Floor
Havre de Grace, MD 21078
United States

Phone: +1 410-939-8732
Fax: +1 410-939-8732
EMail: rgcole@att.com

Dan Romascanu
Avaya
Atidim Technology Park, Bldg. #3
Tel Aviv, 61131
Israel

Phone: +972-3-645-8414
EMail: dromasca@avaya.com

11. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.