The Simple Times™

THE BI-MONTHLY NEWSLETTER OF SNMP TECHNOLOGY, COMMENT, AND EVENTS℠
VOLUME 2, NUMBER 6 NOVEMBER/DECEMBER, 1993

The Simple Times is an openly-available publication devoted to the promotion of the Simple Network Management Protocol (SNMP). In each issue, ***The Simple Times*** presents: a refereed technical article, an industry comment, and several featured columns. In addition, some issues include brief announcements, summaries of recent publications, and an activities calendar. For information on submissions, see page 13.

In this Issue:

Technology and Commentary
Technical Article 1
Industry Comment 4

Featured Columns
Applications and Directions 4
Ask Dr. SNMP 6
Security and Protocols 6
Standards 7
Working Group Synopses 9

Miscellany
Activities Calendar 12

Publication Information **13**

The Simple Times is openly-available. You are free to copy, distribute, or cite its contents. However, any use must credit both the contributor and ***The Simple Times***. (Note that any trademarks appearing herein are the property of their respective owners.) Further, this publication is distributed on an "as is" basis, without warranty. Neither the publisher nor any contributor shall have any liability to any person or entity with respect to any liability, loss, or damage caused or alleged to be caused, directly or indirectly, by the information contained in ***The Simple Times***.

The Simple Times is available via both electronic mail and hard copy. For information on subscriptions, see page 13.

Technical Article

Allan Leinwand, Cisco Systems

In this issue: *Network Design Using the RMON MIB*

There are a wealth of network devices available to a network designer that help segment traffic on a local area network. It is often useful to obtain some data about the traffic on the current network segment before modification or redesign. A network designer may have to choose between using bridging, switching, or routing technologies when formulating a network design. This article will show you how to use information available in the *Remote Network Monitoring Management Information Base* (RMON MIB), defined in RFC 1271, to facilitate this decision process.

The RMON MIB defines objects designed to help manage network segments. Some of this help comes in many useful forms including current and historical segment statistics, individual host specific statistics and host traffic matrices. Although this list of objects is only a subset of the overall information provided by the RMON MIB, this is the set of information which can significantly aid network design.

Introduction

A remote network monitoring device, known as a probe, connects to one or more network segments. The probe may have its own memory, processor, and network interface card dedicated to performing tasks involved with managing the network segments. With these resources, the probe can gather statistics and store them for later retrieval and analysis. As a further benefit, the probe has the ability to send network events to a network management system based on thresholds a user may define on a network management system.

The probe can monitor all the traffic seen on a network segment because it has the ability to put its interfaces in promiscuous mode (hearing all frames). This enables the probe to monitor all traffic, not just the traffic sent to its own media-specific address (such as ethernet address). This is a significant difference in the monitoring ability of the RMON probe in comparison to those devices that only support MIB-II (RFC 1213). Devices that support

MIB-II only maintain statistics on frames sent to their own address (or broadcasts).

Before talking further about network design, we need to define some MIB groups that exist in the RMON MIB. The first group, the Statistics group, has MIB objects that give statistics about each network segment the probe is monitoring. Examining these statistics will help us design a network using real-time information. Currently, these statistics are for ethernet and token ring only. For each segment being monitored the probe keeps a separate table of statistics. Some of the statistics kept about each segment include: total bytes, total packets, total broadcasts, and total collisions.

The next group, the History group, has objects that are similar to the Statistics group. The difference is that the History group provides a way for a network engineer or network management system to take periodic statistical samples from a segment. As we will see, the History group provides a historical perspective which is often necessary when doing network design. The probe can store statistics about network segments and allow you to retrieve them at a later date for analysis. The History group stores data the probe has gathered at each polling interval for ethernet and token ring media. Like the Statistics group, at the time of this writing, these are the only media supported in the History group, although the potential for other media-specific monitoring does exist.

The Matrix group of the RMON MIB stores statistics about conversations between hosts on the network segment. These statistics include the number of packets and bytes sent during each conversation. These statistics will become useful when trying to find an optimal way to segment a network.

The final RMON MIB group which will help us in network design is the Host group. The Host group contains statistics about each host on the network segment. These statistics include for each host: total bytes sent and received, total packets sent and received, total broadcasts sent, and total errors sent. These host specific statistics will also help us determine how to segment and divide a network segment.

When using these statistics to aid in network design, it is important to remember that different designs may be correct for different situations. For example, some network designers tend to optimize a network for maximum traffic rate, while others design for the mean traffic rate. Regardless, it is often necessary to examine the network over time (like a week or more) before making a definitive judgment about design. Throughout this article we will try to look at how to acquire the necessary information to allow you to make these design decisions properly.

Do you need to segment?

Consider that you may have an ethernet segment which does not have any current segmentation. An example of this may be a workgroup or office floor attached to a series of 10baseT hubs/repeaters. This means you have a single collision domain for these hosts. As your network has grown you have kept in your mind that someday you may need to insert a bridge, switch, router, or similar device to help alleviate ethernet collisions. With the RMON MIB you can have the probe monitor the segment for current and historical statistics which can help you decide if segmentation is necessary.

The utilization of the network segment may be a direct indication that segmentation is necessary. You can compute the current utilization of the segment using objects from the Statistics group and the interface speed (using the Interface group object `ifSpeed` from MIB-II) as follows:

```
bit-rate =
    8 * [delta(etherStatsOctets,t1,t0)
              / (t1 - t0)]

utilization =
    bit-rate / ifSpeed
```

where

```
delta(X,t1,t0)
```

denotes the change between the statistic X at time t0 and time t1.

With ethernet, most experts agree that a consistent utilization above approximately 50 percent requires some form of segmentation to maintain adequate performance. To determine if the current utilization on your segment is consistent, you can use the History group object `etherHistoryUtilization`. This object gives you the percentage utilization of the segment, which you can poll over time using the probe.

Even if your ethernet utilization is consistently below 50 percent, segmentation may be necessary if you have excessive collisions. The definition of "excessive" may be different in many environments, but network administrators typically do not like to have more than 5 percent of their total ethernet packets causing collisions. You can calculate this percentage on your segment using objects from the Statistics group as follows:

```
collision-rate =
    rate(etherStatsCollisions,t1,t0)

packet-rate =
    rate(etherStatsPkts,t1,t0)

collision-percentage =
    collision-rate / packet-rate
```

where

```
rate(X,t1,t0) = delta(X,t1,t0) / (t1 - t0)
```

If you find out that your current collision rate on the segment is "excessive" you may then use objects in the History group to find out if the collision percentage you are seeing is normal for the segment by polling these objects over time and performing the same calculations.

Bridging

Now, let us assume that your research using the methods described above makes you decide to segment your ethernet. One alternative you may consider is to install an ethernet bridge and physically break your single collision domain into two collision domains. In fact, this alternative may or may not achieve the effect you desire on the segment. Bridges need to forward broadcasts and multicasts. You can examine the current number of broadcasts or multicasts on the segment with the following formulas:

```
broadcast-rate =
    rate(etherStatsBroadcastPkts,t1,t0)

multicast-rate =
    rate(etherStatsMulticastPkts,t1,t0)
```

Experience shows that small hosts (i.e., networked PCs) often have difficulties when the broadcast rate is over 30 packets/second (e.g., the internal bus on the PC becomes congested moving broadcast packets from the interface card to the CPU). Of course, if a large percentage of your segment traffic is broadcasts and multicasts, an examination of the applications sending these frames may be necessary!

Now, let us assume that you have found out that bridging may help your segment because broadcasts are not a significant amount of the traffic and you want to reduce your collision domain. The next thing to do is to determine where on your network segment would be the best location for the bridge. Ideally, you would like to keep a major percentage of the traffic on a single side of the bridge.

The probe monitors all of the conversations on the segment using the Matrix group. You can acquire information on the amount of bytes, packets, and errors sent between any two hosts using the objects in the Matrix group. The objects refer to communication from source to destination (the "SD" prefix in the object name) and the traffic from the destination back to the source (the "DS" prefix in the object name). For example, if you collect the object `matrixSDOctets` for each conversation, you can determine which hosts send the most bytes between themselves. Upon installation, you should ensure that both of these hosts are on the same side of the bridge.

Switching

Another technology which is an option for network designers is to use switching to segment a network. Typically, switches provide full ethernet bandwidth to all hosts on the segment. Switches accomplish this by giving all attached hosts a dedicated ethernet and removing all collision domains. In some products, the switch does this by forwarding frames based upon ethernet addresses without any buffering. This usually means that switches work well in many-to-many environments, but can have problems in many-to-one environments. This is because if the switch cannot buffer frames, and the switch attempts to forward two frames to the same destination at the same time, one of these frames will be dropped.

When you consider about using a switch to segment your network, it might be useful to see if a majority of your traffic is many-to-many or many-to-one. If it is the latter, you may want to consider an alternative device. Using the RMON MIB objects you can easily tell the flow of your traffic using the Matrix group, as we saw previously. You can use the same objects as described earlier (i.e., `matrixSDOctets`) to make your determination.

Likewise, to the bridged environment, it is important to examine the amount of broadcasts and multicasts on your segment. If you have a large number of broadcast or multicast packets, switching may not help eliminate your problem as the switch needs to propagate these packets to each host.

You may find that a small subset of hosts on your segment have conversations with a large number of other hosts (by using the Matrix group objects). If this is true, you can use the Host Group objects to discover the percent utilization of the entire segment by a single host:

```
host-rate =
      rate(hostInOctets,t1,t0)
    + rate(hostOutOctets,t1,t0)

byte-rate =
    rate(etherStatsOctets,t1,t0)

host-utilization =
    host-rate / byte-rate
```

You can use yet another group in the RMON MIB, the HostTopN group which sorts the traffic seen on the network segment by host. You could further sort this information by bytes sent by each host. This allows you to quickly identify which hosts communicate most on the

segment. You can use the above formulas by using the HostTopN group to show the top senders of octets, and modifying the host-rate formula as follows:

```
host-rate = rate(hostTopNRate,t1,t0)
```

We may have found that a small number of hosts dominate the traffic on the segment and this may make a majority of our traffic many-to- many. Using this data, we may decide to segment the traffic with a switch. In some environments, with a larger number of hosts, network designers will attach the busiest hosts to a dedicated port on the switch and have other hosts which do not require dedicated bandwidth to share a port on the same switch. By sorting hosts by traffic sent and received, the HostTopN group objects can help you make the decision of how to attach each host to the switch.

Routing

A third alternative to segmenting the network is to use a router. The router limits both the collision and broadcast domains on the segment which may be an advantage, given your previous analysis. Also, if you determine through the Matrix group that a large percentage of your traffic is headed off the local area network (such as through a router to another site) it may make sense to break the network in two and attach the two new segments directly to the router. You could detect this situation by examining the Matrix group statistics and see if the destination of a large percentage of your traffic is headed for the ethernet address of the router.

Likewise, the router ethernet address would be the source of a large percentage of traffic and appear as one of the busiest hosts in the HostTopN group.

Summary

We have seen how the objects in the RMON MIB collected by a probe can be useful in network design. These objects, if used correctly, can give valuable hints in figuring out if a network segment needs a new design. Also, some information collected by the probe is helpful in determining the optimal type of device and the location it should be installed on the segment.

Industry Comment

Marshall T. Rose

It's the end of the year. I'm busy working on the test suite for next month's SNMP Testing Summit. So, *no comment*.

Applications and Directions

Steven L. Waldbusser

In this issue: The Trend Towards Hierarchical Network Management

One of the keys to SNMP's success has been its centralized nature, which has helped to reduce its cost and increase its reliability. As SNMP-managed environments have grown and SNMP has been introduced into new environments, new goals have emerged. SNMP users are looking for new features such as further scalability, distributed intelligence, and eased integration. These goals can be achieved when network management is less centralized, especially when it forms a hierarchical management architecture, while still preserving the nature of SNMP's high manager to agent ratio. The RMON MIB and SNMPv2's Manager-to-Manager MIB are current examples of this, and new MIBs are under development that expand on this theme.

The RMON MIB

The Remote Network Monitoring MIB was SNMP's first foray into hierarchical network management. An RMON probe is a network management device that promiscuously monitored packets, gathering and storing statistical information about them for later use by one of the network management applications that controlled it. In addition, the controlling applications could tell the probe to check the values of certain parameters and to notify users if problems were detected.

SNMPv2's Manager-to-Manager MIB

When SNMPv2 was written, it was decided to go further with the hierarchical management strategy by building explicit support into the protocol for manager-to-manager communications. A new PDU, the `inform` PDU, was created for this task. In addition, an initial MIB, the Manager-to-Manager MIB, was written to provide the first hierarchical manager-to-manager application. The Manager-to-Manager MIB allows a remote manager to poll MIB variables, to check their values, and to notify other managers if problems were detected.

Future Hierarchical Management MIBs

Other MIBs will expand these capabilities. For example, Jeff Case and David Levi have written an "Aggregate MIB" that polls MIB objects and aggregates their values by applying mathematical and logical functions to them, allowing the results to be retrieved as MIB variables by other managers. A "History MIB" has been proposed that will poll MIB variables and store a historical summary for a period of time, allowing off-line performance analysis.

Remote ping and traceroute MIBs have been proposed that will allow these operations to be performed remotely. In the future, distributed expert systems applications will be able to send fault reports to higher level managers when network errors are detected.

Each of these developments brings hierarchical management to a particular network management function, providing the benefits of scalability, distributed intelligence, and enhanced integration.

Scalability

SNMP has shown great flexibility in scaling to very large networks. (For example, see this column in the July/August, 1992 issue of ***The Simple Times*** for a detailed analysis of how well SNMP can perform). However, there are cases where many remote devices need to be managed over a slow WAN link, congesting the link with polling requests. Most of the aforementioned technologies help out in this area by keeping the routine polling on the remote network where it must be performed, and only sending error reports or summary information across the WAN to the higher level network manager.

In particular, the Manager-to-Manager MIB can be configured to poll one or more MIB variables on any number of devices. If the values of these variables cross pre-configured thresholds, an event report is forwarded to higher-level network managers. If any event report is dropped, it will be retransmitted. For example, a mid-level manager implementing the Manager-to-Manager MIB could be placed in a west coast office to monitor a network of thousands of machines, while only error reports are sent across the country to the management station in the New York network operations center.

The RMON MIB was designed to handle an even more difficult performance problem. A third-party network service organization may wish to manage a customer's network without having to install an expensive network link to the customer's site. Attaching a probe to a modem on a dialup phone line is an obvious choice, but long distance charges could still be expensive. The RMON MIB will allow the network manager to configure the probe to collect and store data and check thresholds continuously, then to hang up the phone line. If the probe detects a problem, it can dial the phone line, establish a PPP link, and send the error information to the manager. The manager will also periodically call the probe to download daily performance statistics and for other maintenance tasks, often at night when the rates are low.

Distributed Intelligence

The threshold checking functions of the RMON MIB and Manager-to-Manager MIB allow diagnostics to occur in these mid-level management devices, wherever they may be located in the network, without constant manager supervision or traffic. This can increase the efficiency of polling by moving the pollers closer to the managed devices. This in turn allows more network parameters to be checked more often.

If these MIBs are coupled with the aggregate MIB, the periodic diagnostics can become even more intelligent. Mathematical and logical functions of several variables may be continuously checked for health, alerting a manager when problems seem evident. In the future, expert systems may be distributed to each network with similar notifications when things go wrong.

Ease of Integration

A side effect of hierarchical management that surprises some is that it makes management systems easier to integrate. The mid-level managers in a hierarchical management system use SNMP as their high-level "user interface". Because this is a standard, upper-level managers from multiple vendors can easily be integrated to this component. This gives the upper-level application access to the functions implemented in the mid-layer application without worrying about what platform each application uses, what system type they use, and what API's or file formats are defined for the applications.

An example is the Aggregate MIB. If a customer wishes to graph the sum of the disk operations on several servers, he might be out of luck if his graphing tool doesn't support that function. However, he could install an Aggregate MIB implementation and configure it to perform the mathematical operation and to return the result as a MIB variable, which can be graphed by his tool.

This sort of application-to-application interoperability can be applied to diagnostic applications, auto-discovery applications, and data analysis applications, as well.

The advantages that are provided by hierarchical applications will bring good cheer to those who need their higher efficiency, intelligence, or ease of integration.

Even better news is that this technology has already been forging ahead and parts of it are ready to provide solutions today.

Ask Dr. SNMP

Jeffrey D. Case

Dear *Dr. SNMP*,
I just ran a MIB-walker on a new release of an agent and the walker complained that a bunch of variables were of the wrong type. After some investigation we discovered that vendor had a new version of its MIB module with many new features implemented in its new code. They reused many of the OBJECT IDENTIFIERs in their old MIB. As a result, now the new MIB module works only with the new agent, and the old MIB modules work only with the old agent... You know the story. How can this happen?
— *Freaked-out in Fremont*

Dear *Freaked-out in Fremont*,
Down on the farm (in merry ol' England), they have a saying:

> "You cannot make people honest by an act of Parliament."

It seems to Dr. SNMP that the text in the Internet-standard SMI (RFC 1155, page 16) is pretty clear: "New versions may not change the semantics of any previously defined object without changing the name of that object..."

However, every rule seems to have a loophole. The rigorous process for MIB extension and modification, as defined by the rules stated in the SMI, were intended by the authors to pertain to all MIB objects, including those in the mgmt(2) subtree and variables introduced by particular implementations of the protocol (experimental variables and variables found in enterprise MIB documents). These rules clearly prohibit the reuse you have witnessed. Regrettably, depending upon how one defines the term "Internet-standard MIB", one can argue about whether these rules apply to an enterprise MIB module, irrespective of the authors' clear intent. In fact, Dr. SNMP was quite surprised to find this loophole when researching the answer.

Of course, many things lawful are not expedient and Dr. SNMP is unwilling to condone the poor manners of this vendor. It is obvious that this vendor is imprudent, even if its actions do not violate the initial SNMP SMI, in that its actions have caused great difficulty for you, its customer. What is doubly troubling is that the vendor could have done the right thing at less expense.

Thankfully, RFC 1442, the SNMPv2 SMI, is crystal clear (at least in this matter). This is yet another loophole that has been closed by the SNMPv2 framework.

Security and Protocols

Keith McCloghrie

For both SNMPv1 and SNMPv2, the semantics of the `set` PDU require that it be implemented as a multi-phase operation. In this article, we'll examine some of the issues behind the need for multiple phases, and look at one possible implementation strategy.

The semantics of a `set` PDU

In contrast to a retrieval operation, a `set` PDU specifies a new value for each of the variables specified in the PDU's variable-bindings. The definition requires that either all of the variables take on their specified new values, or that none of them do. The latter case occurs when one or more of the new values are invalid. Otherwise, all of the variables must take on their new values *as if simultaneously*. Several consequences ensue from these requirements.

Implications

First, there are no implications which can be assumed from the ordering of the variables within the variable-bindings. For example, the request:

```
set (objectA.1=value1, objectB.1=value2)
```

must have exactly the same result as

```
set (objectB.1=value2, objectA.1=value1)
```

Second, more than one occurrence of the same variable in a single `set` PDU's variable-bindings is ambiguous, since the same variable cannot be set to multiple values at the same time. RFC 1448 does not require any particular behavior from an agent in this circumstance, but rather specifies that the outcome is implementation-specific. This tells manager implementations to avoid sending such `set` PDUs, and at the same time, allows agents to implement as much or as little error-handling code for this condition as they wish.

Third, for some variables, the agent's validation of one variable can be dependent upon the values of other variables, and these other variables might also be modified by the same PDU. In such cases, the validation of the one variable must be performed using the values of the other variables as they would be after the `set` PDU is successful, i.e., using their values, if any, specified by the PDU. For example, suppose an agent imposes the

restriction that the value of objectA.1 must be less than the value of objectB.1, and further suppose that the agent receives a `set` PDU which includes both objectA.1 and objectB.1 in its variable-bindings, with the new value of objectA.1 greater than the existing value of objectB.1, but less than the new value for objectB.1, then this `set` PDU must not fail because of that restriction.

Fourth, for some variables, the setting of a new value requires an agent to obtain additional resources. Further, the agent must keep temporary state information on the additional resources required since two variables in the same `set` PDU might contend for the same additional resources. If the required resources cannot be obtained, then the `set` PDU must fail. For example, when the setting of an object can cause the creation of a new entry in a fixed-length table, multiple instances of that object can be set in the same PDU. Thus, not only does the agent have to check that the table is not already full, but it must also keep track of how many additional entries in the table will need to be assigned for the `set` PDU to succeed.

Fifth, experience has shown that for some variables, the success of setting a new value cannot be absolutely guaranteed prior to the actual assignment of the new value. This is more often true for variables which have some type of action semantics, or some interaction with other parts of the (dynamic) system being managed. The implication here is that, if and when an actual assignment fails, an "undo" function is required to be executed for any other assignments already made.

A Possible Implementation

One possible implementation strategy is to have a four-phase implementation, where the variables in the variable-bindings are processed in each phase. These four phases are: a local-test phase, a global-test and allocation phase, an assignment phase, and, an undo phase.

In the first phase, local checking of the specified values for each of the variables is performed, i.e., checking which is not dependent on the values of any other variable which might also be set in the same PDU. This includes checking for the correct syntax, and that the new value is within the maximum range permitted by the agent under any circumstances. This phase records state information concerning which variables are being set (and to what values) by this PDU.

In the second phase, global checking is performed on the specified values, i.e., checking the values against the values of other variables, either those contained in the state information recorded in the first phase, or for any of the other variables not being set in this PDU, then their existing values. This phase also allocates any resources which will be needed for the assignment to be successful.

In the third phase, the actual assignments are made using the allocated resources, if any, obtained in the second phase.

The fourth phase is called when, and only if, an error was returned by the processing in any of the preceding phases. This phase attempts to undo assignments, if any, which have already been made, and releases any allocated resources which will not now be used.

For an SNMPv2 `set` PDU, a failure in the third phase will result in a `commitFailed` error, except when the fourth phase also fails, in which case it will result in an `undoFailed` error. For an SNMPv1 `set` PDU, both of these types of failures will result in a `genErr` error. Note that an agent should take all possible measures to avoid having to return either the `commitFailed` or the `undoFailed` errors. This is especially true of the latter, since an agent which returns the `undoFailed` error code is admitting that it has disobeyed the protocol specification!

For simple variables, there may be no need for global checking, no need for the allocation of any resources, and no possibility that the assignment can fail. For such variables, the second and fourth phases can be null.

Standards

David T. Perkins

Since the last issue, there have been no new SNMP related standards published. In the pipeline are several including the DNS server and resolver MIBs, an update of the DECnet Phase IV MIB, and the new IF table MIB.

Update on Transition of the SMI

In the last issue, some of the challenges in the transition from SNMPv1 to SNMPv2 were specified. Since then, there has been some continued low-level of grumbling about the changes being too hard for the benefit in the protocol area. However, in the SMI area, work is going well on the transition. In fact, recent e-mail has requested that the transition be speeded up and include updating all the MIBs since the new SMI has quite useful features. The new `TEXTUAL-CONVENTION`, `MODULE-IDENTITY`, `OBJECT-IDENTITY`, `OBJECT-GROUP`, `MODULE-COMPILIANCE`, and, `AGENT-CAPABILITIES` macros allow much information to be specified now in a parseable format not available in SNMPv1. The updates to the `OBJECT-TYPE` macro and the replacement of the `TRAP-TYPE` with the `NOTIFICATION-TYPE` macro are also much appreciated.

Switching topics, the IETF standards process is still undergoing update and review. At the Houston IETF,

a presentation was given on the new section about copyrights. Many people had questions. It appeared that this section needed to be reworded to explicitly state that it was allowable to create derivative works (i.e., stripped MIBs from RFCs) and publish them without any restrictions.

In the next issue, we'll present some opinions on why no standards have been created in the IETF on programmatic interfaces.

Summary of Standards

SNMPv1 Framework (Full Standards):

- 1155 - Structure of Management Information (SMI);
- 1157 - Simple Network Management Protocol (SNMP);
- 1212 - Concise MIB definitions; and,
- 1213 - Management Information Base (MIB-II).

SNMPv2 Framework (Proposed Standards):

- 1441 - Introduction to SNMPv2;
- 1442 - SMI for SNMPv2;
- 1443 - Textual Conventions for SNMPv2;
- 1444 - Conformance Statements for SNMPv2;
- 1445 - Administrative Model for SNMPv2;
- 1446 - Security Protocols for SNMPv2;
- 1447 - Party MIB for SNMPv2;
- 1448 - Protocol Operations for SNMPv2;
- 1449 - Transport Mappings for SNMPv2;
- 1450 - MIB for SNMPv2;
- 1451 - Manager-to-Manager MIB; and,
- 1452 - Coexistence between SNMPv1 and SNMPv2.

Full Standards:

- 1213 - Management Information Base (MIB-II).

Draft Standards:

- 1398 - Ether-Like Interface Type MIB;
- 1493 - Bridge MIB; and,
- 1516 - IEEE 802.3 Repeater MIB.

Proposed Standards:

- 1229 - Extensions to the generic-interface MIB;
- 1231 - IEEE 802.5 Token Ring Interface Type MIB;
- 1239 - Reassignment of experimental MIBs to standard MIBs;
- 1243 - AppleTalk MIB;
- 1253 - OSPF version 2 MIB;
- 1269 - BGP version 3 MIB;
- 1271 - Remote LAN Monitoring MIB;
- 1285 - FDDI Interface Type (SMT 6.2) MIB;
- 1289 - DECnet phase IV MIB;
- 1304 - SMDS Interface Protocol (SIP) Interface Type MIB;
- 1315 - Frame Relay DTE Interface Type MIB;
- 1316 - Character Device MIB;
- 1317 - RS-232 Interface Type MIB;
- 1318 - Parallel Printer Interface Type MIB;
- 1354 - SNMP IP Forwarding Table MIB;
- 1381 - X.25 LAPB MIB;
- 1382 - X.25 PLP MIB;
- 1389 - RIPv2 MIB;
- 1406 - DS1/E1 Interface Type MIB;
- 1407 - DS3/E3 Interface Type MIB;
- 1414 - Identification MIB;
- 1418 - SNMP over OSI;
- 1419 - SNMP over AppleTalk;
- 1420 - SNMP over IPX;
- 1461 - Multiprotocol Interconnect over X.25 MIB;
- 1471 - PPP Link Control Protocol (LCP) MIB;
- 1472 - PPP Security Protocols MIB;
- 1473 - PPP IP Network Control Protocol MIB;
- 1474 - PPP Bridge Network Control Protocol MIB;
- 1512 - FDDI Interface Type (SMT 7.3) MIB;
- 1513 - Token Ring Extensions to RMON MIB;

- 1514 - Host Resources MIB;
- 1515 - IEEE 802.3 Medium Attachment Unit (MAU) MIB; and,
- 1525 - Source Routing Bridge MIB.

Experimental:

- 1187 - Bulk table retrieval with the SNMP;
- 1224 - Techniques for managing asynchronously generated alerts;
- 1228 - SNMP Distributed Program Interface (SNMP-DPI); and,
- 1238 - CLNS MIB.

Informational:

- 1215 - A convention for defining traps for use with the SNMP;
- 1270 - SNMP communication services;
- 1303 - A convention for describing SNMP-based agents;
- 1321 - MD5 message-digest algorithm;
- 1470 - A network management tool catalog; and,
- 1503 - Automating Administration in SNMPv2 Managers.

Historical:

- 1156 - Management Information Base (MIB-I);
- 1161 - SNMP over OSI;
- 1227 - SNMP MUX protocol and MIB;
- 1230 - IEEE 802.4 Token Bus Interface Type MIB;
- 1232 - DS1 Interface Type MIB;
- 1233 - DS3 Interface Type MIB;
- 1252 - OSPF version 2 MIB;
- 1283 - SNMP over OSI;
- 1284 - Ether-Like Interface Type;
- 1286 - Bridge MIB;
- 1298 - SNMP over IPX;
- 1351 - SNMP Administrative Model;
- 1352 - SNMP Security Protocols;
- 1353 - SNMP Party MIB; and,
- 1368 - IEEE 802.3 Repeater MIB.

Working Group Synopses

Frederick J. Baker, Deirdre C. Kostick, and Kaj Tesink

This column is a summary of activities. There is no substitute for actually participating in a working group. Even if you cannot go to the meetings, you can subscribe to the mailing lists. Included in each working group's summary is the address of the group's subscription address. If you are interested in a group's activities and do not subscribe to the mailing list, you should!

SNMP General Discussion

To subscribe: `snmp-request@psi.net`

There was a question on how to parse indexes for a SNMPv1 agent implementation. The implementor is working with a table that is doubly indexed by an `INTEGER` and a `DisplayString`. The problem is how to determine where the `INTEGER` value ends and the `DisplayString` begins. The answer offered was that there is "no problem" if RFC 1212 and ASN.1 rules are followed. The `INTEGER` is a single sub-identifier that may be larger than a single octet. The encoding of the octet will indicate whether it is a continuation of the sub-identifier value. The `DisplayString` consists of "n+1" sub-identifiers. The first sub-identifier is "n" which identifies the length of the string. Then, each octet of the string is encoded as a separate sub-identifier.

There was a question on how to determine if a link between two routers was down. The writer proposed monitoring the `ifOperStatus` values for the two interfaces on both sides of the link. If `ifOperStatus` values for both were up, then the writer assumed that the link was up; otherwise, the link would be down. One response was that multiple, alternate routes could be possible. So one interface could be down, but the routers could still communicate. Another response indicated that other traffic counts should be monitored rather than relying solely on `ifOperStatus`. Another response was that pings or traceroutes were a good alternative-monitoring method.

There were some lengthy SNMPv2-related discussion threads on the list.

Appletalk/IP Working Group

To subscribe: `apple-ip-request@cayman.com`

The Appletalk/IP WG is currently inactive. During the AppleTalk Networking Forum (ANF) meeting in Houston (which met concurrently with the IETF), there was a Zone Changing WG meeting. One of the items discussed was the possibility of using an SNMP-based approach to

perform zone changes. Review of the current Appletalk MIB (RFC 1243) to determine the level of support for SNMP-based zone changes was an open issue.

AToM MIB Working Group

To subscribe: `atommib-request@thumper.bellcore.com`

The SONET MIB has been completed. The AToMMIB WG has recommended the I-D to the AD for further processing by the NM-D and the IESG as a Proposed Standard. (Yes, there were really six acronyms in the previous sentence.) In the ATM MIB the use of `ifTable` by the ATM level has been agreed to with a minor editorial amendment. The mapping for the AAL5 level was also resolved; an AAL5 entity in a switch will be modeled as being connected to a virtual interface, while on a host AAL5 will be stacked directly on top of the ATM level. In addition, a small table with AAL5 error statistics has been introduced.

A lengthy discussion took place on the modeling of connections. The approach followed modeled connections as a set of unidirectional connections in switches and networks (not applicable to hosts). However, it was also felt desirable to have a single approach for both switches and hosts. A number of criteria for modeling connections emerged. The creation of new connections should allow detailed diagnostics in case of errors or problems. Sets should be safe, i.e., two managers should be precluded from inadvertently performing set operations on the same connection. Arbitrary topologies should be possible, including point-to-point, point-to-multipoint, and multipoint-to-multipoint. The number of table rows for a given connection should be small. It should be possible to provide connection status information, and tracing of a connection topology should be simple. The adopted proposal accommodates all these criteria. The expectation is that with the resolution of these problems the ATM MIB can be completed, and can be recommended to the AD for further processing by the NM-D and the IESG as a Proposed Standard before the end of the year.

BGP Working Group

To subscribe: `iwg-request@ans.net`

No SNMP-related traffic to report.

Bridge MIB Working Group

To subscribe: `bridge-mib-request@nsl.dec.com`

The Bridge MIB WG concluded with the publication of RFCs 1493 and 1525. The mailing list remains active as a forum for implementors, primarily to field questions relating to the existing Transparent Bridge and Source Routing MIBs.

Character MIB Working Group

To subscribe: `char-mib-request@decwrl.dec.com`

In anticipation of RFCs 1316, 1317, and 1318 being evaluated for promotion to Draft Standards, the chair continues to solicit implementation experience. The group is also closely awaiting maturing of the current I-D from the Interfaces MIB WG, in order to assess any impact on the Character MIB RFCs.

DECnet Phase IV MIB Working Group

To subscribe: `phiv-mib-request@jove.pa.dec.com`

The revised MIB is awaiting approval and publication by the IESG as a Draft Standard.

FDDI MIB Working Group

To subscribe: `fddi-mib-request@cs.utk.edu`

The FDDI MIB WG concluded with the publication of RFC 1512. The mailing list remains active as a forum for implementors; however, there was no SNMP-related traffic to report.

Frame Relay Service MIB Working Group

To subscribe: `frftc-request@nsco.network.com`

The WG has completed its MIB and is awaiting review by the NM Directorate.

Several Frame Relay service providers and switch vendors have shown interest in the MIB and seem likely to implement it. In fact, at the Houston IETF, a number of people indicated they had implementations in progress.

As this MIB was (informally) co-developed by the IETF and the Frame Relay Forum, the latter is planning to "standardize" the MIB as part of an implementation agreement.

Host Resources MIB Working Group

To subscribe: `hostmib-request@andrew.cmu.edu`

The Host Resources MIB WG concluded with the publication of RFC 1514. The mailing list remains active as a forum for implementors; however, there was no SNMP-related traffic to report.

IEEE 802.3 Hub MIB Working Group

To subscribe: `hubmib-request@synoptics.com`

No SNMP-related traffic to report.

IDPR Working Group

To subscribe: `idpr-wg-request@bbn.com`

No SNMP-related traffic to report.

IDRP for IP Working Group

To subscribe: `idrp-for-ip-request@merit.edu`

No SNMP-related traffic to report.

Interfaces MIB Working Group

To subscribe: `if-mib-request@thumper.bellcore.com`

The Interfaces Evolution MIB is awaiting approval and publication by the IESG as a Proposed Standard. The MIB is similar to MIB-II's `ifTable`, but, unlike MIB-II, interfaces are explicitly layered. The new MIB also handles problems posed by newer media technologies.

IPLPDN Working Group

To subscribe: `iplpdn-request@nri.reston.va.us`

The WG has concluded, leaving a revised draft of the Frame Relay DTE MIB, without a recommendation. The MIB was discussed at the Houston IETF, and needs a few changes (in addition to being converted to use SNMPv2's SMI). The MIB's authors are following up to close this loop.

IS-IS Working Group

To subscribe: `isis-request@merit.edu`

No SNMP-related traffic to report.

Mail and Directory Management Working Group

To subscribe: `ietf-madman-request@innosoft.com`

This WG has completed three MIB modules: the Network Services Monitoring MIB, the Mail Monitoring MIB, and the Directory Monitoring MIB. These MIBs have been reviewed by the NM-D, and are awaiting approval and publication by the IESG as Proposed Standards. The WG is gathering implementation experience on these MIB modules.

Modem Management Working Group

To subscribe: `modemmgt-request@telebit.com`

This WG is working to apply SNMP-based management to ITU draft V.58, which defines GDMO-based management for all V-series DCEs, and contains about 120 attributes. The approach that has been taken is to define a core subset of these capabilities in the coming months, and to gather implementation experience on this subset. The remaining capabilities may follow later as extensions to the core subset.

NOCtools Working Group

To subscribe: `noctools-request@merit.edu`

No SNMP-related traffic to report.

OSPF Working Group

To subscribe: `ospfigp-request@gated.cornell.edu`

The MIB has been updated for use in a CIDR environment, and is being converted to use SNMPv2's SMI. The group is now looking at the IP Forwarding Table MIB, RFC 1354, to add support for CIDR routes.

PPP Working Group

To subscribe: `ietf-ppp-request@ucdavis.edu`

No SNMP-related traffic to report.

RIP Working Group

To subscribe: `ietf-rip-request@xylogics.com`

The MIB has been updated (along with RIP-II itself) to address unnumbered point-to-point links and the Demand RIP protocol. All that remains is to convert the MIB to use SNMPv2's SMI, and then it will be ready

for submission to the IESG for consideration as a Draft Standard.

Remote Monitoring (RMON) Working Group

To subscribe:
rmonmib-request@jarthur.claremont.edu

No SNMP-related traffic to report.

SNA DLC Services MIB Working Group

To subscribe: snadlcmib-request@apertus.com

The SNA DLC MIB WG met at the Houston IETF. They plan to complete a final I-D by the end of December, 1993.

SNA NAU Services MIB Working Group

To subscribe:
snanaumib-request@thumper.bellcore.com

The SNA NAU MIB WG met at the Houston IETF. They plan to complete a final I-D by the end of December, 1993.

SNMPv2 Working Group

To subscribe: snmp2-request@thumper.bellcore.com

The SNMPv2 WG is waiting for reactivation to work on evaluating RFCs 1441-1452 with respect to the standards track. As discussed at the NM Area Open Meeting during the Houston IETF, the timing for reactivation of the WG depends on the level of user-deployment of SNMPv2.

Steve Waldbusser has offered to host an SNMPv2 Interoperability Test at CMU (Pittsburgh, PA). The tests will be scheduled for early 1994. The purpose of the test will be to find and fix bugs, and to discuss common implementation and interoperability problems. Scheduling will take into account the SNMP Testing Summit scheduled for San Jose on January 10-14, 1994.

The University of Twente (the Netherlands) announced the release of their SNMPv2 package. The release is available via anonymous FTP at ftp.cs.utwente.nl in the directory pub/src/snmp. Folks who are interested in the UT software were encouraged to send a note to snmp@cs.utwente.nl.

There was a long thread of SNMPv2-related discussion on the SNMP General list. Differing views were exchanged on the complexity of SNMPv2 and users' needs for SNMPv2 security features. One writer posted a list of proposed changes to SNMPv2. Other folks were also encouraged to develop lists based on their implementation experience to prepare for the potential re-activation of the SNMPv2 WG.

TCP Client Identity Protocol

To subscribe: ident-request@nri.reston.va.us

No SNMP-related traffic to report.

Trunk MIB Working Group

To subscribe: trunk-mib-request@saffron.acc.com

No SNMP-related traffic to report.

Uninterruptible Power Supply Working Group

To subscribe: ups-mib-request@cs.utk.edu

The UPS MIB WG met at the Houston IETF. They are currently working on the final UPS MIB I-D.

X.25 MIB Working Group

To subscribe: x25mib-request@dg-rtp.dg.com

No SNMP-related traffic to report.

Activities Calendar

- SNMP Testing Summit
  January 10–14, San Jose, CA
  For information: +1 415 969 4544
- 29th Meeting of the IETF
  March 28–April 1, Seattle, WA
  For information: +1 703 620 8990

Publication Information

The Simple Times is published with a lot of help from the SNMP community.

Publication Staff

Coordinating Editor:
Dr. Marshall T. Rose — Dover Beach Consulting, Inc.

Featured Columnists:
Frederick J. Baker — Advanced Computer Communications
Dr. Jeffrey D. Case — SNMP Research, Inc.
University of Tennessee
Deirdre C. Kostick — Bell Communications Research
Keith McCloghrie — Hughes LAN Systems, Inc.
David T. Perkins — SynOptics Communications, Inc.
Kaj Tesink — Bell Communications Research
Steven L. Waldbusser — Carnegie Mellon University

Contact Information

Postal: ***The Simple Times***
c/o Dover Beach Consulting, Inc.
420 Whisman Court
Mountain View, CA 94043–2186

Tel: +1 415–968–1052
Fax: +1 415–968–2510
E-mail: st-editorial@simple-times.org
ISSN: 1060–6068

Submissions

The Simple Times solicits high-quality articles of technology and comment. Technical articles are refereed to ensure that the content is marketing-free. By definition, commentaries reflect opinion and, as such, are reviewed only to the extent required to ensure commonly-accepted publication norms.

The Simple Times also solicits terse announcements of products and services, publications, and events. These contributions are reviewed only to the extent required to ensure commonly-accepted publication norms.

Submissions are accepted only in electronic form. A submission consists of ASCII text. (Technical articles are also allowed to reference encapsulated PostScript figures.) Submissions may be sent to the contact address above, either via electronic mail or via magnetic media (using either 8-mm `tar` tape, $\frac{1}{4}$-in `tar` cartridge-tape, or $3\frac{1}{2}$-in MS-DOS floppy-diskette).

Each submission must include the author's full name, title, affiliation, postal and electronic mail addresses, telephone, and fax numbers. Note that by initiating this process, the submitting party agrees to place the contribution into the public domain.

Subscriptions

The Simple Times is available via electronic mail in three editions: *PostScript*, *MIME* (the multi-media 822 mail format), and *richtext* (a simple page description language). For more information, send a message to

```
st-subscriptions@simple-times.org
```

with a `Subject` line of

```
help
```

In addition, ***The Simple Times*** has numerous hard copy distribution outlets. Contact your favorite SNMP vendor and see if they carry it. If not, contact the publisher and ask for a list. (Communications via e-mail or fax are preferred).