The Simple Times™

THE BI-MONTHLY NEWSLETTER OF SNMP TECHNOLOGY, COMMENT, AND EVENTS℠
VOLUME 2, NUMBER 5 SEPTEMBER/OCTOBER, 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 16.

In this Issue:

The Simple Times is available via both electronic mail and hard copy. For information on subscriptions, see page 16.

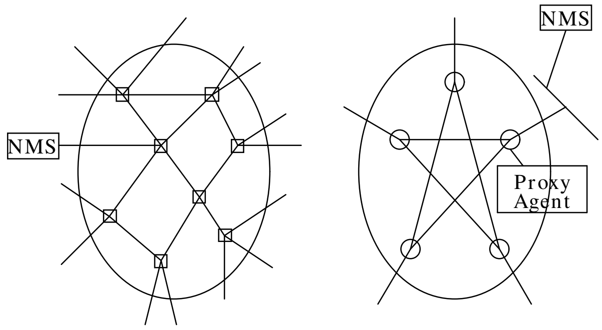
Technical Article

Kenneth R. Rodemann, AT&T Bell Laboratories

In this issue: *Service Management Architecture*

This article presents the Service Management Architecture, an architectural framework for defining MIB modules for Customer Network Management (CNM) of network services over shared networks. Network providers offer a myriad of network services, such as X.25, SMDS, Frame Relay, and ATM. Some of these provide connection-oriented service, while others provide connectionless service. CNM services are becoming an important extension of these transport services to provide customers with a management window into their portion of the shared network. This article focuses on an SNMP-based architectural framework for CNM of connection-oriented network services.

The purpose of this work is to identify the notion of a Service MIB module, and to define an architectural framework for its definition that will permit easy extensibility and interoperability across various network services. In order to explore and understand how Service and Device management differ, consider the fundamental differences in network management functionality between a network service provider and a service customer:



Service Provider Service Customer

or, textually:

- whole-network vs. network-portion view;
- direct vs. indirect management; and,
- physical vs. logical view.

Note that these fundamental differences apply both to public networks and to private networks. In the private network case, the "service provider" is the network administrator and the "service customer" is the network user.

First, service providers are responsible for managing the entire shared network as a whole, while service customers only view and manage their individual portions of the shared service. Because they have a restricted view of the network, customers are unable to perform certain network management functions in the shared environment. For example, a customer which sets routes for optimized throughput of its own traffic may disrupt another customer's traffic. Only the service provider, with a complete view of the entire network, is in a position to determine routes that allow provisioned access to network resources for all customers.

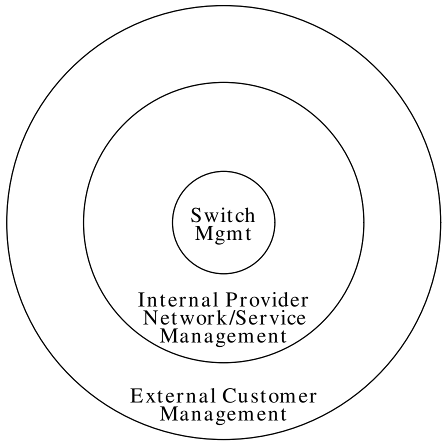
A second fundamental difference in management functionality is that service providers manage the network internals directly, while customers manage their portion of the shared network indirectly. The service provider is responsible for the overall operation of the shared network, so any management control offered to customers must first be approved (perhaps manually) by the service provider before the control request takes effect in the network.

Finally, while service providers see a physical view of the network, customers see a logical view. This logical view includes the customer's configuration of service access points (logical ports) and the virtual connections that run between these logical ports. The customer does not see the individual network switches along the paths of its virtual connections — setting up physical routes is a responsibility of the service provider.

These fundamental differences in network management functionality suggest that there is a wholly different philosophy between Service Management and Device Management. A Device MIB module allows for hands-on management of a physical entity. A Service MIB module provides to customers a logical view of the customer's portion of a shared network service by modeling the service, not the underlying implementation or devices. Much work has been done and experience gained in writing Device MIB modules for hands-on management of physical devices, but defining Service MIB modules is a relatively new area and requires the development of a new architectural framework.

Service Management Architecture

The preceding discussion suggests there are various levels or views at which to manage a network:



For example, a network can be managed at the switch level, at the service provider (internal network) level, or at the service customer (external network) level. The Service Management Architecture focuses on network management at the outer shell — the external customer management view.

The Service Management Architecture models a service network as a single entity or logical device. Even though a customer may have physical connections to multiple switches within the network, these connections are presented to the customer as individual interfaces on the single logical device. To provide this logical view of the network, a service provider supports a Proxy Agent that consolidates the views from the collection of network switches into a single view. The Proxy Agent does this by mapping the switches' physical views into the single logical view. So to satisfy an SNMP request, the Proxy Agent first maps the logical view back to the physical view, then queries the corresponding switches for the requested information.

There exist two views of virtual connections within the Service Management Architecture: service-provider views and customer end-to-end views. Service-provider views consist of single-segment virtual connections established through a single service provider's network. This view is presented by the service provider's Proxy Agent as a logical configuration of service access points (logical ports), access channels, and virtual connections.

Customer end-to-end views consist of multi-segment end-to-end virtual connections that span across multiple service providers' networks. This view is presented by the customer's Network Management Station (NMS) as a concatenation of individual service-provider segments. It is the responsibility of the NMS to consolidate the individual service-provider views to form the customer end-to-end view. Since an adequate definition of the customer end-to-end view requires much more discussion and experience, the remainder of this article focuses on the single service-provider view.

Service-Provider View

A service provider may offer a variety of network services based on differing network technologies and datalink protocols. For example, a service provider may offer services over X.25, Frame Relay, and ATM. The service provider may also permit interworking of services, e.g., by offering virtual connections between Frame Relay ports and ATM ports. The Service Management Architecture promotes effective management of these diverse services by providing a consolidated and consistent management framework for customer network management.

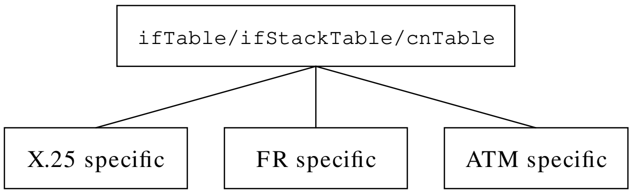
For consolidated management of diverse network services, the Service Management Architecture presents a single generic view of network configuration. This generic view includes configuration information for all logical ports, access channels, and virtual connections, regardless of network service or underlying datalink protocol. The MIB tables `ifTable` and `ifStackTable` contain the generic configuration information for logical ports and access channels, while the newly-proposed `cnTable` contains the generic configuration information for virtual connections.

For consistent management of diverse network services, the Service Management Architecture provides consistency guidelines for the design of CNM MIB modules specific to a given network service or datalink protocol. These consistency guidelines include:

- separate tables for logical port, access channel, and link management information;
- use of VC *flow* tables for virtual connection parameters;
- use of VC *endpoint* tables for virtual connection performance statistics; and,
- back references from protocol-specific VC flow tables to the generic `cnTable` for generic configuration correlation

(VC flow and VC endpoint tables are defined below).

Note the hierarchical relationship between the protocol-generic tables (`ifTable`, `ifStackTable`, and `cnTable`) and protocol-specific tables (for X.25, Frame Relay, and ATM), as diagramed:



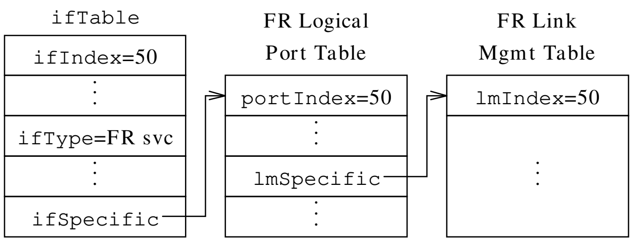
We now consider the table relationships in more detail for logical ports and for virtual connections.

Logical Port Tables

The Service Management Architecture follows the recommendations of the *Evolution of the Interfaces Group of MIB-II* document being developed by IETF's Interfaces MIB WG, which proposes that each layer in an interface's protocol stack have its own entry in `ifTable`, with the hierarchical relationships between interface layers given in the new `ifStackTable`. Because the access channel is layered "below" the logical port, the logical port and the access channel each have their own `ifTable` entry and `ifIndex`. Note that since the Service Management Architecture models a service network as a single entity, each logical port's `ifIndex` value is unique within the service provider's offering.

Consider a typical Frame Relay interface stack, with a Frame Relay port layered above a DS1 access channel. This interface stack is represented by two entries in the `ifTable`, one for the physical access channel (DS1) and one for the Frame Relay logical port. The `ifStackTable` gives the layering relationship between these two `ifTable` entries. Of course, there may be other layers involved; e.g., a Frame Relay service may run over an ATM service, which itself runs over a physical layer. This interface stack would be represented by three `ifTable` entries.

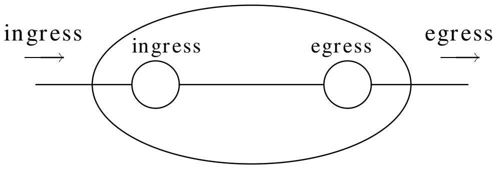
The `ifTable` contains entries for logical ports of various service protocols (e.g., Frame Relay logical ports and ATM logical ports). Each of these service protocols is described in a protocol-specific MIB module that contains a logical port table, which is indexed by the `ifIndex` of the associated ports. The `ifSpecific` variable of a logical port's `ifTable` entry points to the associated protocol-specific logical port table. That table, in turn, may contain pointers to other protocol-specific tables, such as a link management table. An example table relationship for a Frame Relay logical port might be:



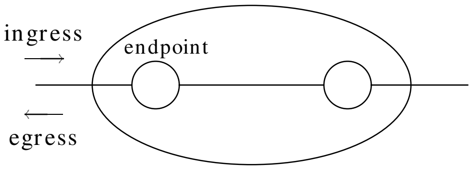
There are a number of other `ifTable` variables that may apply to the network service logical port. When defining a protocol-specific MIB module, the module definition should define the proper use for each of these `ifTable` variables. Note that the protocol-specific use must be consistent with the intended use of the `ifTable` variable — redefinition of `ifTable` variables is not allowed.

Virtual Connection Tables

Virtual connections are logical data transport connections between a pair of logical ports. The Service Management Architecture models virtual connections with two types of tables — VC *flow* tables and VC *endpoint* tables:



VC Flow Table



VC Endpoint Table

Entries in VC flow tables represent unidirectional flows of virtual connections. These entries are indexed by both endpoints of the connection, with one endpoint designated as *ingress* and the other as *egress*. An entry's MIB variables represent the connection flow from the ingress endpoint to the egress endpoint. Note that bidirectional point-to-point connections are represented by two entries in a VC flow table, one entry per flow, with the corresponding ingress and egress ends flipped. The consistency guidelines of the Service Management Architecture recommend using VC flow tables for the (fairly) static virtual connection characteristic parameters.

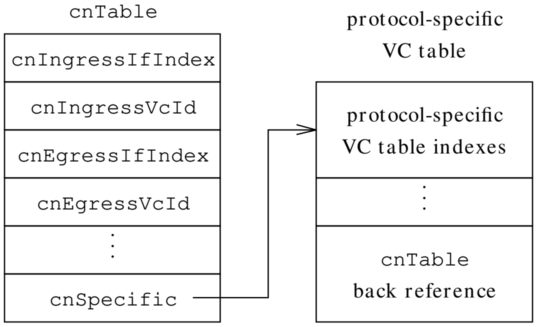
Entries in VC endpoint tables represent connections as seen from a single endpoint. These entries are indexed by a single endpoint of the connection, with an entry's MIB variables referring to the ingress and egress at that endpoint. Note again that point-to-point virtual connections are represented by two entries in a VC endpoint table, one entry for each endpoint. The consistency guidelines of the Service Management Architecture recommend using VC endpoint tables for the real-time virtual connection performance statistics.

VC flow tables and VC endpoint tables extend naturally to handle point-to-multipoint and multipoint-to-multipoint connections as well. For example, consider a full-duplex point-to-multipoint connection from point A to points B, C, and D. This virtual connection consists of 6 unidirectional flows, so a VC flow table will have 6 entries. Likewise, this connection has 4 endpoints, so a VC endpoint table will have 4 entries.

Within the Service Management Architecture, the MIB table structure for virtual connections is similar to the structure for logical ports. All virtual connections, regardless of protocol type, are placed in the protocol-generic `cnTable`, with each entry in `cnTable` containing a pointer to the associated entry in a protocol-specific virtual connection table. Note that this pointer points to an actual *entry* in the protocol-specific table, not just the top of the table. This is necessary because the protocol-specific VC table may be indexed differently than the protocol-generic `cnTable`.

The `cnTable` is modeled as a VC flow table and, as such, is indexed by both the ingress and egress endpoints of a virtual connection flow. These connection endpoints must be identified generically, because the `cnTable` contains entries for virtual connections of various datalink protocols. Thus, each connection endpoint is identified by a tuple (logical port `ifIndex`, VC id), where the VC id is a logical identifier unique to the associated logical port. This VC id is assigned by the service provider as it sees fit. The service provider *may* map the VC id directly to the addressing scheme used in the underlying protocol (e.g., DLCI for Frame Relay), but this is not necessary. The `cnTable` is therefore indexed on these four fields: ingress `ifIndex`, ingress VC id, egress `ifIndex`, and egress VC id.

The `cnTable` contains an OID that points to associated entries in protocol-specific VC tables. To allow for management of interworking service objects, the reverse references must also be in place, i.e., references from the protocol-specific VC tables to entries in the `cnTable`. These backward references may be either OIDs or indexes into `cnTable`. Consider the table relationship between the `cnTable` and a protocol-specific VC table:



Summary

This article presents the Service Management Architecture, an architectural framework for defining Service MIB modules for customer network management. The work is motivated by fundamental differences in management views and functionality between a service provider and a service customer. Differences between service provider and service customer include whole-network vs. network-portion view, direct vs. indirect management, and physical vs. logical view. These fundamental differences suggest a difference in philosophy between Service Management and Device Management.

Of the various views for managing a network, the Service Management Architecture focuses on external customer management at the outer level. A service network is modeled as a single entity or logical device by a Proxy Agent supported by the service provider. This Proxy Agent consolidates the views from the many switches in the network into a single logical view for the service customer.

The Service Management Architecture presents two views of virtual connections: service-provider views and customer end-to-end views. Service-provider views consist of single-segment virtual connections established through a single service provider's network, while customer end-to-end views consist of multi-segment end-to-end virtual connections that span across multiple service providers' networks. The Service Management Architecture focuses mainly on the service provider view, postponing the details of the customer end-to-end view for future work.

The Service Management Architecture provides a consolidated and consistent management framework for customer network management. By presenting a generic view of network configuration using `ifTable`, `ifStackTable`, and `cnTable`, the Service Management Architecture consolidates diverse network services into a single view; and, by providing consistency guidelines, the Service Management Architecture permits consistent management at the protocol-specific level of these diverse network services. The result is an architectural framework that permits easy extensibility and interoperability across various network services.

Industry Comment

Marshall T. Rose

Due to external events (the INTEROP Europe conference and the next IETF meeting), this is a short issue. So, once again, I have an excuse for "no comment".

Applications and Directions

Steven L. Waldbusser

In this issue: *A Network Management Perspective on the age-old Token Ring vs. Ethernet Debate*

Ever since token ring was introduced, there have been debates between token ring proponents and fans of ethernet. Owing largely to the culture clash between these two groups, these debates have been long on religion and short on reality and experience. This article will attempt to be different by taking a network management perspective as well as dealing with the changing technology scene. In particular it will not focus on the relative architectural merits of these two technologies, but rather will focus on the characteristics of real products and experiences of real networks.

We will compare ethernet implemented with SNMP-managed 10baseT hubs and token ring implemented with passive hubs (MAUs) because these are the technologies most often used for these networks. Both support a star-shaped wiring scheme and provide similar bandwidth. However, 10baseT ethernet has the advantage in cost, reliability, and network management.

Commodity Networking Products

A diverse, energetic market exists for ethernet products, keeping costs very low. This has also encouraged the production of many clever products from pocket ethernet adapters to ethernet printers, fax servers and switching hubs. These conditions have not existed in the token ring market, where fewer vendors have captured higher margins. The following table shows the typical price differential, in US dollars/port for a couple of types of purchases:

Type	Adapter	Hub	Total
Low End 10baseT	60	80	140
Low End token ring	285	55	340
High End 10baseT	120	125	245
High End token ring	600	75	675

This shows that token ring is about twice as expensive as ethernet. A similar price differential exists for other network equipment such as bridges and routers.

Analogies can be drawn between the design principles that have contributed to the success of SNMP, and the relative successfulness of token ring and ethernet. For example, token ring has shifted the complexity from the centralized hub to the many network adapters. This increases the total cost of the system because there are many more end nodes than hubs. 10BaseT has extremely simple and cheap interfaces and a more complex hub, not unlike SNMP's simple agents and more complex management stations. Further, ethernet as a whole is much simpler than token ring, making it cheaper, more reliable, and easier to understand.

Reliability

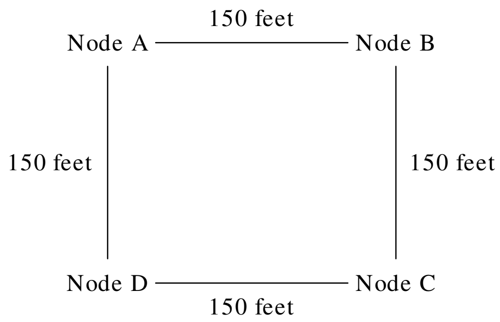
Now that ethernet networks are typically built with 10baseT hub technology, ethernet networking is much more reliable than older coax-based ethernets and has become more reliable than token ring. One reason for this is that token ring is quite complex and requires the continuous, active participation of every node on the ring. In many cases, if a node ceases to function correctly or a cable is damaged, the token ring will cease to function. Ethernet is much simpler and requires only passive participation of all nodes and cables on the net except for those sending and receiving packets.

Token ring has an advantage when running at very high utilization rates. A token ring can run at 100 percent utilization and continue to provide fair access to all stations, while it isn't prudent to run an ethernet at greater than 50% average 0% load. On the other hand, while token ring may work better in the worst case, good network managers never experience the worst case.

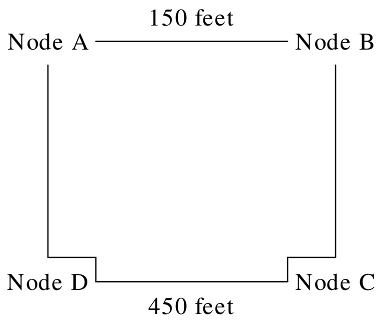
Both 10baseT and token ring advertise the capability to isolate low-level errors by shutting off or routing around the bad link. Token ring hubs have electrical relays that are used to automatically disconnect faulty lines from the hub and to route the ring around the disconnected line. It is important to note some limitations of this fault isolation scheme. First, these relays are not present on the hub-to-hub links (Ring-In and Ring-Out on the MAU), so cable breaks between hubs will take down the ring. Second, these relays often don't disconnect faulty lines as they are supposed to, especially with marginal cable or connector problems or when a video card is plugged into the token ring (a common occurrence!). Third, since these relays are mechanical devices, they sometimes stick in their normal state, unable to route the ring around a host when it powers off or reboots. This makes it necessary for token ring network managers to carry around a small tool that momentarily zaps a stuck relay and unsticks it, restoring the network to operation.

Non-Determinism

The final problem with token ring's failure isolation scheme is that it changes many characteristics of the ring as it is used. The length of cable between stations, and the number of stations repeating the signal changes continuously as hosts are powered up and down or experience failures. This introduces a source of non-determinism that defies a network manager's attempts at carefully engineering these parameters. These changes can sometimes cause the net to fail or to experience more errors. Imagine the following scenario:



In this simplified diagram, each host is the only station on its hub. The signal quality on the network is directly related to the length of cable between nodes. As the cable becomes longer, the signal becomes weaker and the amount of noise increases. The ring in the figure may have been designed to keep all cable runs less than 200 feet. However, if every station on MAU C and MAU D powers down, the token ring will automatically close the relays and cut them out of the ring. This creates the following situation:



This results in a 450 foot link between Host B and Host A, which is not what the network manager expected and might put the ring in a marginal or non-functioning state. This might keep server backups from running at night after many users have powered down their PCs. It is noteworthy that a similar problem is possible with the optical bypass mechanism on FDDI rings. One advantage of token ring that particularly helps in this situation is that it is better suited to running over longer distances than ethernet. When ethernet is run over long distances (more than several hundred feet), collision rates increase, often necessitating the installation of bridges or routers.

Network Management

Token ring has long enjoyed a reputation as the "safer" alternative because it has network management built in. With the proliferation of SNMP-managed hubs, ethernet now has superior network management capabilities. Token ring and 10baseT both provide error statistics for every node on the network, while 10baseT adds statistics detailing how many packets and bytes were sent on each port of the hub. For years token ring has had the ability to list the stations on the ring and show the order in which they are wired. This helps a network manager to isolate problems to a particular node. 10baseT now has a similar capability that goes further by pinpointing which physical hub port each host is attached to. After having pinpointing which host is causing a problem, this helps the manager to physically locate that host. With 10baseT, the network manager can disconnect the problem host from the network with an SNMP command, ensuring that it stays off the network until the problem is solved.

A common tool for solving low-layer network problems is the protocol analyzer. A protocol analyzer is often used to view packets with errors, but due to a lack of capabilities in most token ring chipsets most token ring analyzers can't display those packets. The network manager can only count the number of such errors that occurred. On an ethernet network, these bad packets can be captured and examined to determine their type and length, for example.

Alternatives

This article hasn't described every consideration that one would use in choosing a networking technology. In particular, the ease of integration of token ring into IBM environments and coexistence with installed bases of token ring may be overriding considerations in some situations. One of the most important lessons in the previous paragraphs is that active, SNMP-managed hubs enhance the reliability and manageability of any network, from ethernet to token ring to tin cans and string. The differences in reliability and manageability mentioned above would largely be erased in a token ring environment implemented with active hubs. Of course, this will make the cost disparity between the two technologies even more dramatic.

Conclusions

Many people believe that token ring is being increasingly de-emphasized in corporate networks. The primary reason for this is that network managers are trying to take advantage of the lower cost ethernet solutions available. However, in addition to cost, there are reliability and network management issues that make ethernet an increasingly interesting alternative. Of course, one might have predicted many of these problems by noting that several architectural principles that made SNMP great were ignored in the design of token ring (and FDDI).

Ask Dr. SNMP

Jeffrey D. Case

Dear *Dr. SNMP*,
Why is the ability to reset a managed device NOT a standard MIB-II object? It seems to me that the ability to reset a managed device is a very basic and necessary network management control function.
— *Absolutely, Positively from Memphis*

Dear *Absolutely, Positively*,
Down on the farm (in the Bible belt), we have a saying:

> "If the blind lead the blind, both shall fall into the ditch."

The simple answer is that when MIB-I and MIB-II were written, we didn't see the the need for this. Of course now it seems so clear that even Helen Keller could have seen this need, and we should have. However, we didn't, and it just did not become a part of MIB-I or MIB-II.

Nevertheless, it isn't too late for a MIB variable to become a part of the standard MIB, even if it is too late for it to be a part of MIB-II. Extending the MIB is easy to do, and extensions can become *de jure* and *de facto* standards when there is strong agreement about the need. This is one of the best aspects of the Internet-standard Management Framework based on the SNMP and SNMPv2.

In the meantime, many network administrators are using Uninterruptible Power Supply (UPS) technology, not only to provide uninterrupted power, but also to provide this reset capability. A smart UPS which can be monitored and controlled via the SNMP can be used to toggle the power off and then back on, resetting the device.

Of course, Dr. SNMP will understand if you personally might not want to spend your company's money on UPS technology since your trucks are of a different color.

Security and Protocols

Keith McCloghrie

The performance improvements of SNMPv2's "awesome" `get-bulk` PDU are well known. Less known is that the introduction of the operator requires remarkably few changes to an SNMPv1 agent. In this article, we'll examine these few changes.

The Definition of GetBulk

First, let's recall the functionality of the `get-bulk` PDU. The basic concept is that, with one PDU, `get-bulk` performs repeated `get-next` executions. Two parameters are included in the request: `non-repeaters` and `max-repetitions`. The `max-repetitions` parameter specifies the maximum number of repeated executions for (a subset of) the requested variables. In each repetition, the agent retrieves the variables and their values which lexicographically follow the variables retrieved by the previous repetition. The agent continues processing the repetitions until either the maximum number is reached, or else a maximum-sized `response` PDU is generated, whichever occurs first. The `non-repeaters` parameter specifies how many (if any) of the variables in the request are not subject to repeated executions. This is useful when the values of one or more scalars, e.g., `sysUpTime`, must be retrieved along with variables from multiple rows of a table, in that only one copy of the `sysUpTime` is retrieved along with the multiple rows from the table.

Thus, through using `get-bulk`, a manager can retrieve in one request as many variables, e.g., from the rows of a large table, as will fit in a maximum-sized `response` without knowing the names (i.e., the instance identifiers) of the particular variables it wants to retrieve. When max-repetitions has a value of one, `get-bulk` operates identically to `get-next`, with the exception that `get-bulk` never returns a `tooBig` error; if a `get-next` would return `tooBig`, `get-bulk` will return less than a whole repetition.

Required Method-Routines

The first thing to notice is that for each variable to be retrieved by a `get-bulk` request, the required method-routine(s) are those which retrieve the name and value of the variable which lexicographically follows a particular known variable. These are exactly the same function(s) required to implement a `get-next` request. Thus, the same method-routine(s) can be used without change.

Protocol Engine Changes

For the protocol engine code which processes the `get-bulk` request by calling the appropriate method-routines, it must act differently from the processing of a `get-next` in two ways.

First, after the first repetition, the variable name, which is supplied to the method-routines for finding the next variable, is derived not from the variable-bindings in the request, but from the result of a previous repetition. In particular, for each repeated variable, the variable name supplied on the j-th repetition is the variable name obtained from the (j-1)-th repetition of that variable. Thus, an implementation need only keep a copy of the last variable name obtained from each repeated variable and point to it on second and subsequent repetitions.

Second, the protocol engine code must terminate the processing of repetitions (in the middle of a repetition if necessary) if and when the (ASN.1 encoded) size of the `response` approaches its maximum value. This termination does not need to occur on the exact number of variables for which one more variable would exceed the maximum size; RFC 1448 specifically allows an approximation in this respect, i.e., it allows the `response` to contain some (small number of) variables less than would create a maximum-sized PDU. However, termination is **not** allowed to occur at some fixed number of variables.

To achieve this, an implementation must calculate the size of the ASN.1 encoding (of the name and value) of each variable as the repetitions proceed, and accumulate these sizes. Because the approximation is allowed, the maximum size of the overhead (of the message wrapper and PDU header which will later be needed for the `response` PDU) can be calculated, rather than precisely calculating the actual size in advance. Indeed, a precise calculation in advance of the size of the overhead may not be possible (e.g., because the size of the clock values in an authenticated `response` can vary). Thus, as the repetitions proceed, the maximum size of the overhead plus the accumulated size of the variable encodings is compared to the maximum size of the `response`, and if the next variable would cause the `response` to be too big, then the repetitions are terminated without including that variable.

Summary

Thus, the changes required for an SNMPv1 agent to implement the "awesome" `get-bulk` PDU consist merely of some additional control logic, the retention of some previous results, and the intermediate calculation of the ASN.1 encoded sizes of the partial results.

Standards

David T. Perkins

In the past month, the IEEE 802.3 Repeater MIB has been updated and published as a Draft Standard — the second rung on the standards ladder. A new version of the FDDI MIB has been published. The old version was based on ANSI FDDI SMT 6.2, whereas the new version is based on ANSI FDDI SMT 7.3. The new version enters the standards track at the first rung, Proposed

Standard, since it is a completely new set of objects, even though there is much similarity to the previous version. The Host resources MIB was also published. There has been much interest in getting it completed since the working group was formed 18 months ago. It should be a welcome addition in the expansion of manageable devices since it is the first explicitly directed toward PCs and workstations (i.e., end-systems), instead of routers and bridges (i.e., intermediate-systems). The Token Ring RMON MIB, published as a Proposed Standard, adds support for this media type to the RMON framework. The MAU MIB, also published as a Proposed Standard, completes the document set that defines management for 802.3 devices. And finally, the Source Routing Bridge MIB objects from the original Bridge MIB, RFC 1286, have been published as a separate document after a few cleanups.

Recently Published RFCs

RFC 1500 - Internet Official Protocol Standards (Standard)

This is an irregularly published RFC containing the status of all standards for the Internet community. (Currently, the most recent source for the status of RFCs is the file `1rfc_index.txt` stored in directory `iesg` on host `cnri.reston.va.us`).

RFC 1503 - Algorithms for Automating Administration in SNMPv2 Managers (Informational)

An experimental approach is presented in this document to minimize the amount of information that is required for a user to specify when invoking a management station application in order for SNMPv2 communications to be established and maintained. The document specifies the implementation model, configuration assumptions, operational details, and how to determine and use maintenance knowledge.

RFC 1512 - FDDI Interface Type (SMT 7.3) MIB (Proposed standard)

This defines managed objects for interfaces using FDDI based on ANSI FDDI SMT 7.3. This document is a companion to RFC 1285 which is based on ANSI FDDI SMT 6.2. The SMT group has an object which is the number of SMTs in a device and a table that fully describes each SMT. Next, the MAC group has information on MACs in the device. The optional enhanced MAC counters group has additional MAC counters. The PATH group consists of two tables and a scalar object. The tables list the paths and how they are configured. Finally, the PORT group contains objects which describe the characteristics of all ports.

RFC 1513 - Token Ring Extensions to the RMON MIB (Proposed standard)

The RMON MIB, RFC 1271, defines a framework for remote monitoring by a network probe and those necessary objects specific for IEEE 802.3 (ethernet) networks. This MIB defines those comparable objects for IEEE 802.5 (token ring), and objects for additional monitoring functions used only for token ring. Those objects used to fill out the media specific framework of the RMON MIB include the MAC layer frame statistics table and associated history table, and the logical link layer frame statistics table and associated history table. The additional objects include those based on stations in a ring, and those for monitoring source routing frames.

RFC 1514 - Host Resources MIB (Proposed standard)

This MIB defines a framework and base set of objects independent of hardware and software for managing primarily end-node (host computers) on a network. The first group, System, augments the system group from MIB-II. The Storage group describes the storage areas such as file systems, primary memory (RAM), and swap space. The next group has several tables used to describe devices such as processors, network interfaces, printers, disk drives, etc., on a host system. Executing software is described by the next group. Associated with the table of executing software is an optional table of memory and processor consumption for each executing program. The last group contains objects that describe the locally installed software on the host.

RFC 1515 - IEEE 802.3 Medium Access Unit (MAU) MIB (Proposed standard)

This MIB defines the objects used in managing IEEE 802.3 Medium Attachment Units (MAUs). There are three classes of MAUs identified by this MIB. Each has a group of objects defining its characteristics. The classes are repeaters, interfaces, and broadband DTEs.

RFC 1516 - IEEE 802.3 Repeater MIB (Draft standard)

This is an updated version of RFC 1368. Several descriptions were updated for clarity. The behavior for object `rptrAddrTrackLastSource` was incompletely defined, and instead of updating the description, a new object, `rptrAddrTrackNewLastSrcAddress`, was created to replace it.

RFC 1525 - Source Routing Bridge MIB (Proposed standard)

The source routing bridge MIB objects under the `dot1dSr` branch from the original bridge MIB, RFC 1286, were removed to create this separate MIB module. The definitions were updated to track the changes in the

IEEE 802.5M SRT Addendum to the IEEE 802.1D Standard for MAC Bridges. The MIB contains two tables: the port table, describes each port on a source routing bridge; and, the port-pair table, contains the bridge number and state for each port pair.

Challenges in the Transition to SNMPv2

The SNMP management standards community is entering a period of transition. Change usually brings discomfort, especially when the path seems unclear. The transition is from a world of only SNMPv1 to one of both SNMPv1 and SNMPv2. To provide comfort, a directional beacon to show the way has been lighted by the Area Director, via a message sent to the SNMP mailing list last August. The *State of the Area* report provided guidelines for the transition and information on a mail responder that checks SNMPv2 MIBs and another mail responder that converts MIBs from the SNMPv2 to SNMPv1 format.

Even with the beacon there are still challenges. The SNMPv2 rules for the Structure of Management Information (SMI) has some really nice features that would be desirable to use. However, at the INTEROP August 1993 Conference and Exhibition in San Francisco, there were few companies making commitments for shipping SNMPv2 end-user products! Some companies were visibly upset when asked about future SNMPv2 products instead of current SNMPv1 products.

A standards organization has to be careful that it keeps its constituency happy. The transition to SNMPv2 needs to make sure it doesn't outpace it's users. The primary authors were quite careful in planning the transition. Some of the proponents for adding creation and deletion operations to SNMPv2 didn't consider the transition consequences of these additions. So, it may be a jerky ride of speeding up, and slowing down while transitioning. More direction lights are needed, such as additional MIB compilers and tools, as well as mangement station vendors smoothing the path. This column will blow the horn when the first MIB is transitioned to SNMPv2.

In the next issue, we'll give you an update on the state of the transition and the IETF standards process which is again being updated.

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 (NCP) MIB;
- 1474 - PPP Bridge Network Control Protocol (NCP) 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;
- 1227 - SNMP MUX protocol and MIB;
- 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);
- 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 mailing list. To subscribe, simply append "`-request`" on to the local-part of the address. For example, the submission address for the SNMP general discussion list is

```
snmp@psi.net
```

so to subscribe, you'd send a message to

```
snmp-request@psi.net
```

If you are interested in a group's activities and do not subscribe to the mailing list, you should!

SNMP General Discussion

Submissions: `snmp@psi.net`

There was a question on whether SNMP requires that the index of a table be included in the `SEQUENCE` defining the entry. The answer was that the structure is technically correct. However, the openly-available SMIC compiler will, if given the "-7" option, generate an error for tables containing such a definition.

Another topic was the timeliness of data reflected in MIB information. The question was what are acceptable ranges of data currentness. One responder replied that a good network manager would have to be aware of the data currentness and adjust reactions to NM data based on the age of the data. Another writer proposed adding enterprise-specific objects to indicate the age of the data. This discussion led to a suggestion to redefine `ifLastChange` to be the number of time units since an interface came up, rather than the value of `sysUpTime` which reflects the time units since the agent came up. Another writer suggested having an object, `ifTimeEstablished`, which records the value of `sysUpTime` at the time the interface is established.

A set of questions on initial values for hardware-maintained counters started a long discussion on the intelligence of agents. In SNMP, counters are not guaranteed to start at an initial value of zero. Instead, the general paradigm is that the relative (i.e., delta) values over time are of interest to network managers. The thread drifted to agent intelligence when a writer commented that this philosophy was frustrated — the writer wanted information such as available bandwidth, capacity percentages, packet rates, averages, and so on. A response observed that this type of information should be provided by the NMS based on the raw data collected by the SNMP agent.

There was a question on whether `ifIndex` values should start at 1 and be contiguous to `ifNumber`. A well-known responder pointed out that the Interfaces MIB WG, which is responsible for evolving the interfaces group, recommends, but does not require that values of `ifIndex` be assigned contiguously.

There was a question on how to deal with responses that may be too large to send as a single datagram. If IP fragmentation isn't acceptable (i.e., if buffer space isn't available), the proper reply is to use the `tooBig` value in the `error-status` field. Otherwise, if the response can be buffered for fragmentation, then a response should be sent. Note that the segmentation doesn't occur at the SNMP level.

There was a very long discussion thread on the merits of connection-oriented (CO) versus connectionless-mode (CL) methods of transport for SNMP. Also mixed in the discussion were messages on the need for out-of-band signaling versus in-band signaling. One responder noted that in Volume 1, Number 2 of ***The Simple Times***, *Dr. SNMP* discussed some research in this area. Another responder suggested that RFC 816, *Fault Isolation and Recovery*, is recommended reading. The reasons why SNMP uses a CL-mode transport service were summarized:

- CL-mode transport requires only the most minimal communications stack;
- The goal of a CO-mode transport is to hide problems in the network from the application which is probably not a good idea for every kind of network management;
- When the network goes south, it is exponentially difficult to complete the three-way handshake used by CO-mode transport for connection-establishment; and,
- If the management application needs "reliability", then the manager can employ an appropriate re-transmission algorithm on top of a CL-mode transport service, and the agent still needn't be burdened with maintaining the state associated with a CO-mode transport service.

Appletalk/IP Working Group

Submissions: `apple-ip@cayman.com`

No SNMP-related traffic to report.

AToM MIB Working Group

Submissions: `atommib@thumper.bellcore.com`

Only minor modifications have been discussed for the SONET MIB. A discussion on the need for the set of 15–minute Interval Tables, and the 24–hour Total Tables resulted in the Total Tables being removed, since they can also be calculated by a manager from the Interval Tables. A suggestion to combine the Configuration and Current Tables was also accommodated. Some final polishing is still needed on alignment with the results of the Interfaces MIB WG. Several ATM MIB issues have been discussed. A question on the use of the `ifLinkUpDownTrapEnable` was answered with that the default should be `disabled` to avoid too many alarms, following the philosophy that only the lowest protocol layer should generate the `linkUp` trap. A proposed test facility to replace the counters per VC, discussed in Amsterdam, received mixed reactions. The observation was made that the exact needs for this feature are still not well understood. A lengthy discussion took place on the exact use of the refined `ifTable` as per the latest Interfaces MIB WG proposal. This resulted in some proposed modifications for the Interfaces MIB WG, observing that ATM, with its fixed packet length, could be treated differently from the regular `ifPacketGroup`. A proposal to introduce additional counters was not received well. The point was made that the ATM Forum had also not accepted these additional counters. A suggestion was also made to change the number of `ifEntries` needed to manage an ATM interface into one `ifEntry` for the ATM layer, and one `ifEntry` for AAL5 (only if AAL5 supports the internet layer directly). AAL3/4 is already covered through RFC1304 (SIP MIB). AAL1 would not really need an `ifEntry`. A discussion was also held on the exact purpose of `atmInterfaceSpecific`, and questioned whether and how enterprise-specific ATM MIBs should be accommodated this way. Polishing is still needed on alignment with the results of the Interfaces MIB WG. New versions of the SONET and ATM MIBs have been posted as I-Ds. Since these I-Ds use SNMPv2's SMI, peer versions using SNMPv1's SMI will be maintained on `thumper.bellcore.com`. Automatic translations can also be obtained by mailing an SNMPv2 MIB to `mib-v2tov1@dbc.mtview.ca.us`.

BGP Working Group

Submissions: `iwg@ans.net`

The WG has posted a new version of the BGPv4 MIB as an I-D. This MIB defines objects for managing the Border Gateway Protocol version 4. The WG continues to gather implementation experience.

Bridge MIB Working Group

Submissions: `bridge-mib@nsl.dec.com`

The WG has concluded with the publication of RFC 1525 (Proposed standard), which defines managed objects for IEEE source routing bridges. This is a complementary document to RFC 1496 (Draft standard), which defines managed objects for IEEE 802 bridges in general. The mailing list will remain active as a forum for implementors.

Character MIB Working Group

Submissions: `char-mib@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.

DECnet Phase IV MIB Working Group

Submissions: `phiv-mib@jove.pa.dec.com`

The WG's I-D, reflecting implementation experience with RFC 1289, is being reviewed by the NM-Directorate.

FDDI MIB Working Group

Submissions: `fddi-mib@cs.utk.edu`

The WG has concluded with the publication of RFC 1512 (Proposed standard), which defines managed objects for FDDI devices implementing the ANSI SMT 7.3 draft standard. The mailing list will remain active as a forum for implementors.

Frame Relay Service MIB Working Group

Submissions: `frftc@nsco.network.com`

The WG is rapidly developing a MIB module, and anticipates completing its final I-D after the Houston IETF.

Host Resources MIB Working Group

Submissions: `hostmib@andrew.cmu.edu`

The WG has concluded with the publication of RFC 1514 (Proposed standard), which defines managed objects for host systems. The mailing list will remain active as a forum for implementors.

IEEE 802.3 Hub MIB Working Group

Submissions: `hubmib@synoptics.com`

The WG has concluded with the publication of of RFC 1515 (Proposed Standard), which defines managed objects for IEEE 802.3 medium access units, and RFC 1516 (Draft Standard), which defines objects for IEEE 802.3 repeaters. The mailing list will remain active as a forum for implementors.

IDPR Working Group

Submissions: `idpr-wg@bbn.com`

The WG is currently working on multicast, multipath, and resource allocation for IDPR. The expectation is that this will result in MIB development, but the MIB enhancements to support IDPR multicast, multipath, and resource allocation will not be defined until the IDPR enhancements themselves have been completed.

Also, the latest version of the IDPR MIB is expected to be released soon and hopefully can be submitted for standards-track evaluation before the end of this year.

IDRP for IP Working Group

Submissions: `idrp-for-ip@merit.edu`

No SNMP-related traffic to report.

Interfaces MIB Working Group

Submissions: `if-mib@thumper.bellcore.com`

The WG has completed a new I-D which is likely to be its final draft. If not, the WG will meet at the Houston IETF.

IPLPDN Working Group

Submissions: `iplpdn@nri.reston.va.us`

The WG has concluded, but without making a recommendation on its I-D reflecting implementation experience with RFC 1315. This is expected to be resolved shortly.

IS-IS Working Group

Submissions: `isis@merit.edu`

No SNMP-related traffic to report.

Mail and Directory Management Working Group

Submissions: `ietf-madman@innosoft.com`

The WG has completed three MIB modules and is awaiting review by the NM Directorate: the Network Services Monitoring MIB defines the generic part of a MIB suitable for monitoring applications which provide some kind of network services; the Mail Monitoring MIB extends the basic Network Services Monitoring MIB to allow monitoring of Message Transfer Agents (MTAs); and, the Directory Monitoring MIB defines the MIB for monitoring Directory System Agents, a component of the OSI Directory.

Modem Management Working Group

Submissions: `majordomo@telebit.com`

No SNMP-related traffic to report.

NOCtools Working Group

Submissions: `noctools@merit.edu`

No SNMP-related traffic to report.

OSPF Working Group

Submissions: `ospfigp@gated.cornell.edu`

The WG is preparing two I-Ds: one to revise RFC 1253; and the other to add support for CIDR routes to RFC 1354.

PPP Working Group

Submissions: `ietf-ppp@ucdavis.edu`

No SNMP-related traffic to report.

RIP Working Group

Submissions: `ietf-rip@xylogics.com`

No SNMP-related traffic to report.

Remote Monitoring (RMON) Working Group

Submissions: `rmonmib@jarthur.claremont.edu`

The WG has been reconstituted with a charter directed at evaluating RFC 1271 for elevation to Draft Standard status.

SNA DLC Services MIB Working Group

Submissions: `snadlcmib@apertus.com`

The WG has begun working on an initial draft, and held an interim meeting at the APPN Implementors Workshop.

SNA NAU Services MIB Working Group

Submissions: `snanaumib@thumper.bellcore.com`

The WG held an interim meeting on at the APPN Implementors Workshop.

SNMPv2 Working Group

Submissions: `snmp2@thumper.bellcore.com`

One writer asked for documentation on the differences between SNMPv1 and SNMPv2 and on the compatibility between the two versions of SNMP. A response indicated that RFC 1452 documents the co-existence mechanisms between SNMPv1 and SNMPv2, including how to proxy SNMPv2 requests to a SNMPv1 agent.

A question was raised on the following scenario: should a response be sent to a successful `set` request which deletes either the source or destination party for the response? One answer was that it would be very hard to require an agent implementation to not make the set operation effective until the reply packet had been safely placed onto the wire. (With internal operating system queueing, it can often be extremely difficult for an agent to know when the response had actually made it out of the network interface.) Another response suggested that the result should be implementation-dependent, allowing the agent do whatever is easiest for it to do.

TCP Client Identity Protocol

Submissions: `ident@nri.reston.va.us`

No SNMP-related traffic to report.

Token Ring Remote Monitoring Working Group

Submissions: `rmonmib@jarthur.claremont.edu`

The WG has concluded with the publication of RFC 1513 (Proposed standard), which defines managed objects for monitoring of IEEE 802.5 token ring networks. This is a complementary document to RFC 1271 (Proposed standard), which defines managed objects for monitoring of IEEE 802.3 ethernet networks. The mailing list will remain active as a forum for implementors.

Trunk MIB Working Group

Submissions: `trunk-mib@saffron.acc.com`

No SNMP-related traffic to report.

Uninterruptible Power Supply Working Group

Submissions: `ups-mib@cs.utk.edu`

The WG is continuing to work on an Internet-Draft and is trying to reach consensus on this draft prior to the next IETF meeting in Houston.

X.25 MIB Working Group

Submissions: `x25mib@dg-rtp.dg.com`

No SNMP-related traffic to report.

Announcements

New mailing for SNMP testing

The list is `snmp-test@netcom.com` (please send requests to be added to the `snmp-test-request@netcom.com`). Since this is a new list, please wait until October 8, 1993 before making submissions. This will allow enough time for people to subscribe.

Corrections to Volume 1, Number 5

In Volume 1, Number 5 of ***The Simple Times***, the technical article, *Accomplishing Performance Management with SNMP*, contained some formulas that were in error. The correct formulas are:

```
input-packet-rate  =
      (delta(ifInUcastPkts,t1,
             ifInUcastPkts,t0)
          + delta(ifInNUcastPkts,t1,
                  ifInNUcastPkts,t0))
    / (t1 - t0)

output-packet-rate  =
      (delta(ifOutUcastPkts,t1,
             ifOutUcastPkts,t0)
          + delta(ifOutNUcastPkts,t1,
                 ifOutNUcastPkts,t0))
    / (t1 - t0)

broadcast-rate =
      (delta(ifInNUcastPkts,t1,
             ifInNUcastPkts,t0)
          + delta(ifOutNUcastPkts,t1,
                  ifOutNUcastPkts,t0))
    / (t1 - t0)

traffic-rate =
      (delta(ifUcastPkts,t1,
             ifInUcastPkts,t0)
          + delta(ifOutUcastPkts,t1,
                  ifOutUcastPkts,t0))
    / (t1 - t0)
```

Activities Calendar

- 28th Meeting of the IETF
  November 1–5, Houston, TX
  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).