The Simple Times™

THE BI-MONTHLY NEWSLETTER OF SNMP TECHNOLOGY, COMMENT, AND EVENTS℠ PREMIERE ISSUE
VOLUME 1, NUMBER 1 MARCH/APRIL, 1992

The Simple Times is an openly-available publication devoted to the promotion of the Simple Network Management Protocol (SNMP). Each month ***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 14.

In this Issue:

The Simple Times is available via both electronic-mail and hard-copy. For information on subscriptions, see page 14.

Technical Article

Greg L. Satz, cisco Systems, Inc.

In this issue: *A New View on Bulk Retrieval with SNMP*

There has been much discussion over the benefits and drawbacks to the Simple Network Management Protocol (SNMP) since its inception in 1987. Unfortunately too much of the criticism results only in discussion, and too little is acted upon. This article will describe some experimental work to address some of the perceived and real deficiencies of SNMP when retrieving bulk data.

Origins of SNMP

SNMP was developed to provide a general purpose internetworking management protocol. Its primary goal was to be simple so nothing would stand in the way of its ubiquitous deployment. To this end, it has been very successful as it is currently deployed in almost every major internetworking product on the market. However, like many achieved goals, the primary strength can also become a weakness. The simplicity that permitted wide product acceptance was traded off against more powerful function. An extreme example of simplicity versus power can be realized by comparing SNMP against the Common Management Information Protocol (CMIP), the ISO entry to the standard management protocol world. CMIP has a very rich set of primitives and core set of data elements. However, to implement CMIP, a subset of the protocol must be selected. Then, to achieve interoperability, this subset must be agreed upon with other implementors. As SNMP was specified completely and with no options, one implemented what was there and interoperability was assured.

Returning to simplicity, SNMP was built simply for a number of reasons other than time to market: robustness in the face of network failure; low overhead on the devices running the protocol; and ease of debugging the protocol itself (the last thing you want to debug is the management protocol that is supposed to be helping you debug your network). Thus, the SNMP limited itself to the User Datagram Protocol (UDP). This gave the implementor the ability and responsibility to manage lost packets and perform any necessary retransmissions.

As network debugging in the face of changing routes will certainly mean losing packets, retaining this control from the transport service (layer 4) was considered essential. Since a network management protocol will be run continuously it is mandatory that it consume as minimal network resource as possible. UDP allows the necessary control over packet transmissions, packet size and content (packetization). It was a natural choice.

SNMP has three control primitives that initiate data flow from the requester (get, get-next and set). There are two control primitives the responder uses to reply. One is used in response to the requester's direct query (get-response) and the other is an asynchronous response to obtain the requester's attention (trap). All five of these primitives are carried by UDP and are thus limited in size by the amount of data that can fit into a single UDP packet (65507 octets). All implementations of SNMP must be able to receive messages of at least 484 octets in size. Larger maximum values are permitted, via bi-lateral agreement. The relatively small message size was a goal of the design but for some reasonable set of network management functions, it imposes a limitation.

Often in network management, it is necessary to obtain bulk information without knowing at first what it is. In one case, there is a set of problems having to do with packets not going where they are supposed to, due to device misconfiguration that prevents proper protocol operation where one needs to view the entire set of data. For example, in determining some kinds of network failure modes, one has to know the entire contents of the tables containing routing or address mapping information — and these tables can be quite large! Cases such as these are more the exception than the rule, but having the complete information available is necessary if the problem is to be solved. Many a time, a problem is uncovered simply by reviewing a routing table.

Retrieval of Management Information

SNMP has the get-next primitive which permits the viewing of data without requiring prior knowledge. If you know what you are looking for, the get primitive will return it. When you want an entire table of information, the get-next primitive will obtain it.

However, unless employed with care, the get-next primitive can be extremely resource-intensive in real time, network bandwidth, and the agent's CPU time. The simplest use of the get-next primitive is to start at the beginning of a table, await the response and then issue another get-next with the name returned. As an example, say you wanted the next-hop address, next-hop interface and route-type from a routing table containing 1000 entries. Using the simplest form of get-next, this would require 2x3x1000 or 6000 packets (get-next and get-response packets, columns, and rows). A straight-forward optimization would be to request the three columns in a single packet. This puts the number of packets at 2x1000 or 2000 packets. In real time, it is the product of the round trip by the number of requests. In agent CPU time, this is still 6000 lookups in the routing table for both cases.

An Early Approach to Bulk Retrieval

A recent work, RFC 1187, defined two new ways of reducing the real time and packet overhead. The first, the *pipelined algorithm*, creates multiple get-nexts requests running concurrently. Based on table size and network round trip time, the algorithm dynamically determines how many concurrent requests should be running. This reduces the amount of real time by using concurrency to maximize available network bandwidth. The real time is reduced to 1/(number of concurrent requests). So, in the example above if we determine the network path to the agent can sustain three concurrent requests, we can reduce the real time by one-third. However the number of packets on the wire and the number of agent routing table lookups remain at 6000. The same straight-forward optimization of combining rows together could reduce the packet count to 2000. Routing table lookups remain at 6000.

A second algorithm, the *parallel algorithm*, reduces the packet count by combining the multiple concurrent get-nexts into a single packet. Although this minimizes the total number of requests sent over the network, no savings are realized for the agent CPU in lookups.

A Different Approach: SNMP over TCP

To examine these issues in a different way, we built an experimental implementation of SNMP over TCP. In addition to a new transport protocol, an additional primitive was defined, called *get-column*. It is implemented only over the TCP transport and behaves like the get primitive. Each of the arguments names an entire column of the table, rather than a cell in the table. Hence, in the response, many values are returned for each argument. When run over the TCP transport, this set of data is efficiently packetized, based on the TCP negotiated maximum segment size. Due to the work in the late-80's on TCP retransmission and startup packet sequences (e.g., by Karels, Karn, Jacobson, and Partridge), the SNMP data is sent as efficiently as possible without any extra software development. In addition to get-column, SNMP over TCP allows all other SNMP primitives to be used, except for trap.

In SNMP, data structures are defined using a subset of Abstract Syntax Notation One (ASN.1), and then encoded using the Basic Encoding Rules (BER). The BER has two ways for encoding the length of data: the *definite* and *indefinite* length encoding schemes. The definite length encoding scheme includes all the length fields for each datum. The indefinite length encoding scheme uses markers to denote the end of the encoding. The benefit of the definite scheme is that decoding is easy but encoding is hard. With the indefinite scheme, the burden is actually reversed — if the entire ASN.1 object can be processed in a single packet. Otherwise, the indefinite scheme requires some extra state be maintained to suspend and resume the decoding operation when new information arrives via TCP. (This assumes a multi-threaded decoder which can decode more then one ASN.1 object simultaneously). For this reason, the definite scheme is used so the decoder can determine when all the data is received to begin decoding.

Complexity of Implementation

Most of the hard work is done in the agent where a simple state machine is used to handle input spooling, variable processing, and output spooling. Most of the difficulty is in dealing with a half-closed TCP connection, which occurs when the requester sends data and then closes his half of the connection. This may not be an issue on the application side, as most operating system implementations of TCP do this automatically. The state machine consists of four states: Idle, Input, Output and Abort. The Idle state is required to determine the half-closed connection and to time out and possibly recycle unused connections. Input is used to capture and process the SNMP request. The Output state takes the encoded output and sends it down the TCP connection. The Abort state exists to handle error conditions caused by improper use of SNMP.

All of the agent complexity lies in accumulating the SNMP request and spooling the encoded response from and to the requester. On input, the complete request must be gathered prior to ASN.1 decoding (unless the decoder is able to handle and postpone incomplete encodings on the fly.) Once completely received the operation can be processed and the results encoded. If the encoder can be suspended and resumed, then encoding can be given to TCP on the fly; otherwise, the entire result must be encoded to a memory store and then given to the TCP in chunks. For a get-column primitive with many requests, the memory requirements can be substantial.

For the management station, the implementational complexity is roughly half compared to that of the agent. The output requests tend to be small, so output to the TCP connection is simplified. However the input response can be large and thus require enough memory to hold it and some state to fill it up. Again, a smart decoder would make this easier.

For support of the get-column primitive, extra code is required to validate that the argument given is part of a valid table. Since each argument will produce many results, to save on CPU time in the agent, the results are not returned in any particular order. This permits each implementation to return the complete column in its native order. However, each get-column request is expected to be atomic. This allows multiple columns to be consistent across the rows within a single request. Further, each column within the same table must be returned in the same order as the other columns, so the rows are aligned. Errors are returned for the same reasons as the get primitive.

Pros and Cons

The benefits of using SNMP over TCP are numerous. Agent performance is enhanced tremendously as the tables can be indexed, based on internal and often private information, without adding any extra code to the lookup algorithms. This reduces the table retrieval function to the same computational loading as a user interface (show or display) command. Almost all modern implementations of TCP can maximize network bandwidth but are sensitive enough to back off when there is congestion. No extra coding is required to pipeline or parallelize an unreliable transport. The savings here should not be underestimated. Each vendor which already has a modern TCP stack will save many lines of code and a number of bugs by using an already fielded and debugged protocol stack. This will save the vendors and their customers much time and money. Another benefit is the addition of reliable transfer of network management information. Even traps could be adapted to use a TCP transport, though this is perhaps more controversial.

There are two drawbacks to this scheme: the get-column primitive modifies the protocol and requires a TCP transport. The later TCP requirement is easily dealt with by falling back to UDP if TCP is not available. Of course, the get-column primitive adds some extra complexity to the table retrieval logic. We need a broader understanding to determine if this is a serious limitation or not.

Software development moves forward by evolving the unknown into the known. At the time of SNMP's inception, it was not possible to conceive of a reliable transport based network management protocol. Today's

problems require more sophistication in the amount of data required to analyze a problem. This puts the burden back on the protocol to gather it quickly and efficiently. The two proposals in RFC 1187 do this without protocol changes but are not scalable. The approach described in this article achieves the same end by using a well-known transport present in a great many internetworking platforms. It also reduces the agent processing overhead at the same time.

The experimental implementation described in this article is presently available in the cisco software 8.3(1) and later releases; the TCP listener will respond to port 1993 and the get-column primitive is number 5.

Industry Comment

Marshall T. Rose

Welcome to ***The Simple Times***.

Normally the *Industry Comment* section contains an editorial on the state of network management. Instead, for the premiere issue, we'll look at what this newsletter is all about. In the next issue, we'll have something more thought-provoking.

How this all came about

The Simple Times came about because one of my clients wanted to have a means of keeping track of developments in the world of SNMP. Although there is a mailing list, not everyone has Internet-mail connectivity. Keeping in mind that SNMP is used in many domains which are unrelated to the Internet-world, this isn't as far-fetched as it might seem. Further, although many attended meetings of the SNMP Working Group solely for the purposes of gathering information, that working group has recently disbanded. So, after discussing the matter with some colleagues, we decided that a newsletter would be the best approach.

After securing commitments from my colleagues to help with the project, the next major problem was to determine the business model for the newsletter. There are basically two ways of doing it: professional or volunteer. The professional approach requires that the newsletter be run as a business — with cost-recovery through subscription fees and perhaps even advertising. The volunteer approach is tricky in that, outside of volunteer time, it requires that all other expenses be sunk-costs.

A professional newsletter about SNMP and network management could probably break even or perhaps show a modest profit. However, it takes a lot of resources to make this happen. If a publication has a subscription fee, then someone has to collect it. If a publication has advertising space, then someone has to sell it. Given that

> "there's no such thing as a free lunch, but sometimes it costs less to give away food than to collect money"

we decided that the volunteer approach was a better business model. This way, my colleagues and I can concentrate on putting things in the newsletter, rather than spending time putting out the newsletter.

The final step was to set up minimal-cost distribution channels. ***The Simple Times*** is available both in hard-copy and electronically. To make the hard-copy distribution a no-cost item, we settled on a rather unique distribution mechanism: hard-copy is distributed by SNMP vendors. When an issue comes out, each participating vendor gets a clean-copy (at no charge). They then make copies for their customer lists or special friends, usually after they put their company name in the upper-right corner. (***The Simple Times*** is not-affiliated with any vendor, although several contributors are employees of vendors of SNMP products and services.)

The electronic distribution is via Internet-mail. ***The Simple Times*** is sent out in two formats: MIME and PostScript. MIME, in case you haven't heard of it, is the Multi-media format for Internet-mail. In order to minimize personnel time, a subscriber is added or removed automatically — via e-mail.

Who Contributes

The format for ***The Simple Times*** is rather simple. Each issue, which comes out every other month, has four sections: a technical article, which has been scrutinized by a review board; an industry comment, to provoke thought and discussion; several featured columns; and, the usual miscellany of an activities calendar, recent publications, and the like. The technical article and industry comment are solicited from the community, whilst the featured columns are written by regular columnists.

So, the only thing remaining is to let you know who the volunteers are. We'll start with the people who write the featured columns:

Applications and Directions is written by Steve Waldbusser. Steve is the author of the most widely-used public-domain implementation of SNMP. He's also the driving force behind the Remote Network Monitoring MIB, which is receiving wide adoption for LAN monitoring. In his column, Steve talks about the changing face of management, focusing on the applications we need to get the job done, and the problems we have in achieving those goals.

Ask Dr. SNMP is written by Jeff Case. Jeff is one of the four original authors and implementors of SNMP. Since that time, Jeff has been tireless in promoting and productizing SNMP (hence, he's often referred to as Dr. SNMP). In his column, Jeff answers the latest questions people are asking about SNMP.

Security and Protocols is written by Keith McCloghrie. Keith is the co-author of the Internet-standard SMI and MIB. He is also one of the three co-authors of the "Secure" SNMP. In his column, Keith talks about "Secure" SNMP and other developments from the protocol perspective.

Standards is written by Dave Perkins. Dave, a member of the Network Management Area Directorate of the Internet Engineering Task Force, is working on improving the way MIBs are written. In his column, Dave talks about network management standardization activities within the IETF.

Working Group Synopses is written by Bob Stewart. Bob is the force behind the "character" MIBs being developed by the IETF. In his column, Bob has the unenviable task of surveying the various IETF working groups struggling with network management, and then distilling the key issues under discussion.

The Technical Review Board for ***The Simple Times*** consists of Jeff, Keith, Steve, and myself.

Finally, as coordinating editor, my job is simply to "make the trains run on time". So, when you like the contents of ***The Simple Times***, thank the other volunteers. If an issue comes out late, you know who to blame.

What You can do to Help

The Simple Times is published with a lot of help from the SNMP community. Take a look at the *Submissions* section at the end of this issue. It describes how you can contribute a technical article or industry commentary, or an SNMP-related announcement. If you have something to contribute, I invite you to do so.

Applications and Directions

Steven Waldbusser

In this issue: *Exposing the Myths about Autotopology*

Autotopology is often cited as the most requested network management feature in user surveys and by enthusiastic sales and marketing personnel. The dream of autotopology is that a management station can be installed and will automatically learn the configuration of the entire network and display it in a graphical form suitable for network management. This dream is often held by users of large networks daunted by the prospect of configuring hundreds or thousands of network devices by hand. While many aspects of this dream are possible, many others are impossible or are ill-advised.

Network management users need a graphical network map to display the network in a logical and useful manner. This requires a fair bit of information from the network. Autotopology can ease the burden of entering this volume of information into the Network Management System (NMS) by hand. Some network management architects wish to make their products simpler by using autotopology to make up for the lack of a database — they simply re-discover the network configuration on every startup. Of course, the costs of this strategy are prohibitive for all but the smallest of internets.

There are three aspects to autotopology that should be explored in turn: topology discovery, node discovery, and drawing the map. There are difficulties in each of these areas, as well as ideas that may help make them more solvable.

Topology Discovery

When discovering the network topology, the network management system attempts to find all devices and networks that interconnect the internet, and to discover the interrelationships among all of these. This topology is made up of LANs, WANs, routers, bridges, hubs, repeaters, and so on. Often this discovery can be made by using SNMP to query for routing information from interconnection devices, from which a crude map may be constructed. Unfortunately, this type of discovery will not notice transparent devices such as bridges, hubs, and repeaters. A network manager typically needs to see these devices on a map to solve everyday problems. The standardization and implementation of MIBs for bridges and repeaters will provide access to forwarding and discovery information, and thus provide opportunities to add these devices to the map in the future. However, such opportunities will always be hindered by the inherent transparency of these devices.

Device Discovery

After discovering the network topology, a network management system needs to discover the devices that exist on the internet. One strategy that has been used is to send a broadcast SNMP packet that will compel every device to respond with its identity. This can be dangerous on mid-size or larger networks due to the storm of replies that will choke the network. There is also no guarantee that all devices have an SNMP agent or that the agent will respond. In any event, this mechanism must be used with care.

A second method of device discovery is to send an ICMP ECHO packet to every possible address on the internet, in sequence. This is referred to as the *Monte Carlo method of network management* (a term coined by Chuck Davin). This method consumes a lot of time and network resources. On a Class A IP network with 4 million possible addresses this would be impossible, but even on a Class B IP network it will generate at least 65,000 packets and take a lot of time (though it's probably the 65,000 ARP broadcast packets that will kill your network). This mechanism should be used with care as well.

A third method of device discovery is to use the passive monitoring capabilities of a Remote Network Monitoring device to discover devices from the source addresses of the packets they send. The Remote Network Monitoring (RMON) MIB defines objects that allow SNMP retrieval of a list of link-level addresses discovered by an RMON probe. The cost of dedicated remote monitoring systems could be a barrier to some users, but as routers, hubs, and other devices implement the RMON MIB, the cost can drop dramatically. The Internet Engineering Task Force (IETF) has chosen RMON as the preferred mechanism for device discovery, and will shortly embark on an effort within the RMON Working Group to add the objects for discovery of the addresses of network protocols such as IP.

To be effective, it is almost always necessary to store the discovered devices and topology in a database. It is also helpful for the NMS to automatically use SNMP to discover configuration information from devices and to store it in the database. This information can be invaluable in tracking down the source of a network failure, at which time it is probably impossible to query the devices directly. A good system will also have facilities for updating its database when it detects additions, replacements, or changes of network devices.

Map Drawing

The next step in autotopology is to draw a graphical map of the network, given the topology and devices that have been discovered. What is likely to result from this operation is a map containing hundreds or thousands of tiny icons without any recognizable structure. The reason that there is no recognizable structure is that the topological information is not sufficient to draw a network as one would view it logically. The NMS user typically views the network in geographical or administrative terms (or both), while the NMS can only discover the connectivity of the network. For example, the NMS doesn't have enough information to know that the LANs in the administrative building should be laid out side-by-side. It is equally likely to scatter them across the map. The user's aid must be enlisted to layout the map in a logical manner. With a well-designed network management station, this can be a simple matter of dragging icons and networks to get them placed correctly.

The Other Shoe Drops

After handling the layout problem, the biggest problem facing the NMS user is that the NMS doesn't have any concept of which devices are interesting to the user, so it either displays all of them or none of them. In fact, both cases are often nearly useless. It is very hard for the management station to understand what will be interesting to the user, so the user needs to configure this aspect of the system. The management station can help by choosing interesting nodes based on predefined criteria and placing these on the map. For example, the management station may choose to select all nodes with any of the following MIB-II characteristics:

- ipForwarding = gateway(1)
- ifNumber > 2
- sysObjectID identifies it as bridge, router, server, etc.
- ipOutRequests/sec > 100

Similar heuristics can be applied to other MIBs. In fact the presence or absence of the particular management objects that make up a MIB might signal that a device is interesting.

Clearly there are several reasons why autotopology cannot be truly automated. Automated systems usually can't get enough topology detail; they can't lay out the map in a logical fashion; and, they can't automatically decide which devices are important enough to deserve an icon on a crowded display. There are some ideas and algorithms which, if used, can make the system more usable and efficient. However, it is important to note that these systems will always require help from the network management user and will never completely satisfy the autotopology dream.

Ask Dr. SNMP

Jeffrey D. Case

Dear *Dr. SNMP*,
Are *community strings* case-sensitive? Some agents don't care, others do. And, now I am
— *CoNfUsEd in Costa Mesa*

Dear *CoNfUsEd in Costa Mesa*,
There is a saying back on the farm:

> "You can put a pig in a suit, but it still acts and smells every bit like a pig."

What this means is that, according to RFC 1157, the community name is a string of octets. Octets are 8–bit units of binary information. As such, they don't even need to be ASCII, let alone case-sensitive ASCII. A strict interpretation of RFC 1157 indicates that a community string containing only ASCII characters is indeed case-sensitive. However, an implementor may decide to use case-insensitive matching in an agent. This should be discouraged, as it may confuse users and even management consultants. Of course, some of our parents brought us up tending to be Case sensitive at all times.

Dear *Dr. SNMP*,
If an agent doesn't support SNMP's set operation, can the vendor claim conformance to the SNMP RFC (or any MIB they claim to implement which has read-write objects)?
— *Jaded in Jersey*

Dear *Jaded in Jersey*,
Your question inspires additional questions in the mind of Dr. SNMP. Suppose a vendor sold you a TCP implementation that could send but could not receive. Would it conform to the TCP specification? Suppose you bought a television which claimed to have a remote control but when you went to use it, you found that it would display the channel on the screen but wouldn't allow you to change it. Would that be satisfactory? Returning to your question, there is a saying back on the farm:

> "That dog won't hunt."

What this means is that agents and manager stations which don't support the set operation don't implement SNMP because the set operation is an integral part of SNMP. Similarly, if a MIB has one or more objects which have an ACCESS of "read-write", then for an agent to implement that MIB, the agent must allow an authorized management station to issue a set request for that object.

Your question points out an issue that is deeply disturbing to Dr. SNMP. Some vendors found it difficult to implement SNMP sets and read-write variables in the MIB. Rather than commit the engineering resources to fully implement SNMP management framework, they simply claim that that did not implement control operations because of weaknesses in the authentication mechanisms. Such a posture has become fashionable as customers of these vendors have let them get away with it. Of course, while the security of SNMP could be (and is being) strengthened, the correct thing to do would be to fully implement the specifications, including sets and read-write MIB variables. Vendors who are concerned about the weaknesses of SNMP security could ship the systems with the default state of these features disabled, with appropriate cautions in the manual text describing how to enable them.

Dear *Dr. SNMP*,
Should a management station be able to reset the counters in an agent to zero?
— *Frustrated in Fremont*

Dear *Frustrated in Fremont*,
There is a saying back on the farm:

> "Sometimes, sometimes not."

What this means is that it depends on whether the MIB object is defined to have an ACCESS of "read-write". If so, then an authorized manager should be allowed to set the variable to any value that makes sense. However, just because you can doesn't mean you should. Counters should be monotonically increasing. This is especially important when multiple managers are in use. The normal SNMP manager application would retrieve the current value of the variable of interest, "remember" it, and then compute delta values using the base value and values obtained from subsequent queries. By the way, Dr. SNMP has carefully examined the 1152 objects currently standardized or being developed in the IETF. Of these, he couldn't find a single counter with an ACCESS of "read-write". Gentle reader, perhaps there is a very good reason for this.

Security and Protocols

Keith McCloghrie

SNMP Security is the major recent protocol development in the SNMP area. In this article, we'll look at why we need SNMP Security, that is, what are the "threats" against which it provides protection, and discuss some of the mechanisms it uses. In future articles, we'll see how the mechanisms are integrated into the protocol and discuss some issues involved in product implementation and deployment.

The original designers of SNMP knew that security would become important, but they also knew that getting agreement on it in 1987 wouldn't be possible. So, they purposefully provided a general *community* field to provide a handle for both authentication and authorization. Hence, the community field in today's SNMP is used, among other things, to identify the authentication algorithm for a message. The only authentication algorithm defined was the *trivial* algorithm, for which a message having a known string in its community field was automatically authentic.

Since SNMP is designed to be datagram-based, a community string is included in every message. So, while today's SNMP is not wide-open (you do have to know the right community string), neither is it secure. Any unauthorized person able to capture a SNMP packet off the network immediately learns a community string for the target device, and can then perform his own management operations on the device. A minor improvement is to add restrictions based on the source address (e.g., the IP address) of a message, but authorized addresses are as easy to learn as community strings and the source IP address of a datagram is easily faked.

The level of concern this raises for network administrators depends on their network environment. The most common variation has been in their willingness to use SNMP for control. Some administrators find the community mechanism completely unacceptable for control; in contrast, others have very little concern. Thus, while SNMP has seen tremendous growth in popularity for monitoring networks, it has not been so widely used for controlling networks. Indeed, some vendors have used this as an excuse not to support SNMP's set operation in their agents.

Threats

The effort to define SNMP Security originally began three years ago. We anticipated being done long before now, but it proved to be more difficult than just defining an extra authentication algorithm or two. Initially, the goal was just to satisfy the need to:

- authenticate a message, so that the recipient can be confident in knowing who generated the message, and thereby apply the appropriate access control (i.e., grant read-write or read-only access to a particular collection of MIB objects.)

However, over the last three years other threats have been identified. These are:

- malicious alteration of a message in transit; e.g., modifying a legitimate set operation to change a bridge's filtering parameters to have an unintended value (e.g., instead of tightening the filtering, the set operation could be modified to disable the filtering).

- replay of a previously-sent legitimate message; e.g., after capturing a message which reboots a device, replaying that message at any time later.

- unauthorized disclosure of a captured message; e.g., examining the contents of a captured set operation which changes passwords for a terminal server, thereby learning the new passwords.

Countering these four threats led to the major goals of SNMP Security: origin authentication, message integrity, replay protection, and privacy (i.e., encryption). In addition, SNMP Security recognizes the importance of allowing network administrators to choose how much security they want and when. At one extreme, an especially security-conscious administrator might want every message to be authenticated and private. A different administrator might not want any security except in rare circumstances.

Mechanisms

The mechanisms specified by SNMP Security to achieve these goals are: the MD5 digest algorithm, the Data Encryption Standard (DES), and loosely synchronized clocks. Both MD5 and DES are used as *symmetric* cryptographic algorithms, that is, they have a secret value which is used by both the sender and the recipient of a message.

MD5 is a cryptographic checksum algorithm which specifies how to generate a 16-octet checksum (called the *digest*) for a block of data. The strength of MD5 relies on the fact that it is not feasible to compute a different block of data which generates the same digest value. For SNMP Security, the block of data is the concatenation of the message to be transmitted and a secret value (which is not transmitted). By having the secret value known only to the sender and receiver, two of SNMP Security's goals are obtained. First, the digest value can be computed only by the sender or receiver, providing origin authentication; and, second, no other entity can compute the correct digest value for an altered message, providing message integrity.

DES has been the U.S. Federal Data Encryption Standard for several years. SNMP Security uses its Cipher Block Chaining (CBC) mode for privacy. In this mode the value of one part of the message affects the encrypted value of all later parts of the message. This is important considering that two SNMP messages often have the same values in some of their fields.

The use of loosely synchronized clocks enables the sender to include a timestamp in a message, and the receiver to use that timestamp to verify that the message has arrived within a defined *lifetime*. This lifetime value must be large enough to accommodate variation in message transmission times across the network. In addition, clocks tend to drift and so they must be synchronized periodically. Fortunately, the frequency of such drift is small compared to potential variations in transmission time, so the clocks need only be loosely synchronized. This provides replay protection, in that the message can be replayed only during the lifetime interval, during which time the duplication of messages can occur for non-malicious reasons in a typical IP-datagram network.

Looking forward

A valid question to ask is: are MD5 and DES stronger (in the cryptographic sense) than the needs of SNMP Security in the normal situation? The answer is: probably. However, security is a tricky business. With advances in technology, what was strong yesterday, may not seem so strong tomorrow. Also, these specifications will be useful only by being approved as Internet standards, which requires that they be approved by the security community. No weaker algorithms meet this requirement. However, SNMP Security specifies that MD5 and DES are used only when needed. To any who might use the overhead of these algorithms as an excuse not to implement them, the answer is that it is the network administrator who should choose when to use the algorithms, not the implementor.

Standards

David T. Perkins

A major function of the IETF is to produce standards for the TCP/IP suite of protocols. Documents generated by the IETF which are completed and meant for distribution are published in the Request for Comments (RFC) document series. Not all RFCs are standards. To become a standard, a document enters the standards track as a proposed standard, moves to draft standard, and finally emerges as a full standard. Other documents in the RFC series are labeled as experimental, informational, or historical.

This column is about the standards that define the Internet-standard Network Management Framework. Each issue of this column will summarize the status of relevant documents. The current issue looks at documents on the standards track, while the next issue will present the Internet standardization process.

Full Standards

RFC 1157 - Simple Network Management Protocol (SNMP)

The format and interpretation of SNMP messages are described in this document, along with the definitions of generic traps. Some parts of this document are modified by standard practice, and by the new work on SNMP Security.

RFC 1155 - Structure of Management Information (SMI)

The method to name and describe managed objects is defined in this standard. Much of the document has been updated by the Concise MIB (RFC 1212) and Trap Definitions (RFC 1215) documents. Further, some parts of this document are modified by standard practice. This document needs to be updated to reflect the state of the art.

RFC 1213 - Management Information Base (MIB-II)

This MIB defines objects for basic management of TCP/IP-based devices, and obsoletes RFC 1156, the original Internet-standard MIB. Areas covered in the MIB include: objects describing the system, objects for device interfaces, address translation objects, IP protocol objects, ICMP protocol objects, TCP protocol objects, UDP protocol objects, EGP protocol objects, SNMP protocol objects, and, a placeholder for interface objects based on transmission medium type.

Draft Standards

RFC 1212 - Concise MIB definitions

This document defines a format for producing concise, yet descriptive, MIB modules. It accomplishes this task by redefining through augmentation of the OBJECT-TYPE macro defined in the SMI document (RFC 1155) and describing how to specify and interpret the instance encodings for SNMP variables.

Proposed Standards

RFC 1229 - Extensions to the Generic-Interface MIB

Objects for generic interfaces are defined in MIB-II (MIB-I). When MIBs for media-specific objects were developed, it was discovered that there was an additional set of objects common across the media types. This MIB defines that collection. These include: interface counter and configuration objects, objects used to perform tests, and, a list of receive addresses for an interface. Note: this document is updated by RFC 1239.

RFC 1230 - IEEE 802.4 Token Bus Interface Type MIB

This MIB defines objects for interfaces that conform to the IEEE 802.4 Token Bus specification. These objects include: those that report operational state and parameter values, initialization values for operational parameters, interface statistics, and, object identifier assignments for well-known tests and chip sets. Note: this document is updated by RFC 1239.

RFC 1231 - IEEE 802.5 Token Ring Interface Type MIB

This MIB defines objects for interfaces that conform to the IEEE 802.5 Token Ring specification. These objects include: those that report operational state and parameter values, interface statistics, timer values, and, object identifier assignments for well-known tests and chip sets. Note: this document is updated by RFC 1239.

RFCs 1232/1233 - DS1/DS3 Interface Type MIB

These two MIBs define objects for DS1 and DS3 physical interfaces based on the AT&T T-1 specifications and Extended Superframe formats for DS1 or ANSI T1.102-1987, ANSI T1.107-1988, and ANSI T1.404-1989 for DS3. These include: configuration objects, error counters over the last 24 hours in 15 minute intervals, interface statistics, and, mappings for fractional DS1 channels (for DS1 interfaces). Note: these documents are updated by RFC 1239.

RFC 1239 - Reassignment of experimental MIBs to standard MIBs

When RFCs 1229, 1230, 1231, 1232, and 1233 were published, they were not updated to specify the correct MIB subtree root. This document corrects that oversight.

RFC 1243 - AppleTalk MIB

This MIB defines objects for the AppleTalk suite of protocols. Included are objects for: LocalTalk Link Access Protocol (LLAP), AppleTalk Address Resolution Protocol (AARP), Datagram Delivery Protocol (DDP), Routing Table Maintenance Protocol (RTMP), Kinetics Internet Protocol (KIP), Zone Information Protocol (ZIP), Name Binding Protocol (NBP), and, AppleTalk Echo Protocol. Also included are objects describing the configuration and status of AppleTalk Ports.

RFC 1253 - OSPF version 2 MIB

This MIB allows routers that have implemented OSPF to be fully managed via SNMP. The groups in the MIB include: global status and configuration objects, area table and area stub metric table, the link state database, address range table and host route table, interface and interface metric tables, the virtual interface table, and, the neighbor and virtual neighbor tables.

RFC 1269 - BGP version 3 MIB

The BGP MIB allows routers implementing the BGP routing protocol to be monitored. This MIB consists of a few objects that specify the global information, a table of information about BGP peers, and, a table of path attributes.

RFC 1271 - Remote Network Monitoring MIB

This MIB defines the objects needed to completely manage via SNMP a device that monitors ethernet (802.3) LANs. It is structured so that other LAN types such as token ring may be added. It is probably the most extensive MIB developed within the IETF, and defines a new model for row creation which can be used in the development of other MIBs. The objects include the following groups: statistics based on the LAN type being monitored, historical samples of statistics, alarm generation objects, tables of statistics based on source and destination addresses in LAN frames, frame capturing after filtering, and, event generation.

RFC 1284 - Ether-Like Interface Type MIB

This MIB defines objects for interfaces that conform to the ether-like (i.e., IEEE 802.3 CSMA/CD, Ethernet, and StarLan) specifications. These objects include: those that monitor and control operational state and parameter values, interface statistics, a table of collision types, and, object identifier assignments for well-known tests and chip sets.

RFC 1285 - FDDI Interface Type MIB

This MIB defines objects for interfaces that conform to the FDDI specifications. The ANSI committee that wrote the FDDI specifications also included management objects in those specifications. This MIB is a translation of the SMT revision 6.2 mandatory objects. The goal of the translation was to keep the object semantics the same even though the names or structure of the objects were changed. The objects in the MIB include: those for SMT monitoring and control, MAC status and control, port status and control, attachment status and control, and, object identifier assignments for well-known chip sets.

RFC 1286 - Bridge MIB

Devices which provide transparent bridging that is consistent with the IEEE 802.1d specification, and source routing bridges consistent with the IBM Token Ring Architecture may be monitored and can have a few objects configured via this MIB. The objects defined by IEEE for transparent bridging were mapped trying to keep the semantics unchanged even though the names or structure may be changed. The groups for both

types of bridging include: global bridge parameters, statistics and configuration per bridge port, spanning tree objects, per port spanning tree configuration and statistics, source routing information per port, forwarding database for transparent or SRT bridges, per port statistics for transparent bridging, and, a destination address filtering database. The MIB also includes traps for new root and topology change.

RFC 1289 - DECnet phase IV MIB

This MIB defines objects so that a device using DECnet phase IV can be managed via SNMP. The objects come from the July 1983 DECnet Architecture Network Management Functional Specification version 4.0.0. This MIB contains quite a few objects. They are combined in the following groups: system, network management, session layer, end communication layer, routing layer, circuit parameters and counters, DDCMP circuits, DDCMP multipoint control, ethernet, counters, adjacency, line, nonbroadcast line, and area.

RFC 1304 - SMDS Interface Protocol (SIP) MIB

For use with IP over SMDS, this MIB provides statistics at SIP level 3, SIP level 2, and for the Physical Layer Convergence Protocol (PLCP) layer. MIB objects for the physical layers DS1 and DS3 are provided in other MIBs. There are MIB object for the mapping information as defined in RFC 1209 for running IP over SMDS. Finally, a table is provided to log SIP level 3 errors.

Working Group Synopses

Robert L. Stewart

Many IETF working groups address SNMP-relevant issues. In some cases, the group is under the Network Management Area, and the work is entirely directed toward SNMP, such as developing a MIB for a particular transmission medium. In other cases, the SNMP work is a small component of a primary task, such as developing a protocol for the Internet community.

The following synopses summarize the charter and current status of both types of groups and provide the address to use if you wish to join the group's Internet mailing list. Future columns will report SNMP-relevant occurrences in these groups, and also introduce any new groups that may appear.

SNMP WG

In November of 1991, the IETF's SNMP Working Group disbanded because its chartered work was complete. However, there is still a mailing list for ongoing general discussions of SNMP.

Request Address: `snmp-request@psi.com`

Border Gateway Protocol WG

The group's primary tasks are development of the Border Gateway Protocol and promotion of its use. Their milestones include development of a relevant MIB, RFC 1269, a Proposed Standard as of September 1991.

Request Address: `iwg-request@rice.edu`

Bridge MIB WG

The group's primary task is development of a MIB for bridges, giving due consideration to the work of IEEE 802.1d. They have produced RFC 1286, a Proposed Standard as of December 1991.

Request Address: `bridge-mib-request@nsl.dec.com`

Character MIB WG

The group's primary task is development of a MIB for character stream devices such as terminals and printers. The group has three documents: Character MIB, RS-232-like MIB, and Parallel-printer-like MIB. Final drafts, dated January 1992, have been submitted for IESG and IAB approval as Proposed Standards to be published as RFCs.

Request Address: `char-mib-request@decwrl.dec.com`

DECnet Phase IV MIB WG

The group's primary task is development of a MIB for DECnet Phase IV nodes. The have produced RFC 1289, a Proposed Standard as of December 1991.

Request Address: `phiv-mib-request@jove.pa.dec.com`

Ethernet-like MIB WG

The group's primary task is preparing the Ethernet-like MIB for promotion to Draft Standard. Due to concerns over proper consideration of IEEE 802 standards, the charter includes collecting very specific implementation reports and constrains the group from changing object semantics or adding new objects. The Ethernet-like MIB is RFC 1284, a Proposed Standard as of December 1991. The group had its first meeting in November, and has begun to collect implementation experience.

Request Address:
`enet_mib-request@europa.clearpoint.com`

FDDI MIB WG

The group's; primary task was the development of a MIB for FDDI devices. They have produced RFC 1285, a proposed Standard as of January, 1992. The group is now inactive, pending implementation experience.
Request Address: `fddi-mib-request@cs.utk.edu`

IEEE 802.3 Hub MIB WG

The group's primary task is development of a MIB for IEEE 802.3 multiport repeaters, giving careful consideration to IEEE draft standard P802.3K. The IEEE 802.3 Repeater MIB is an Internet Draft dated July 1991. Due to the number of edits at the November meeting, the group is not ready to recommend for advancement until the March meeting. Discussions are active on the mailing list.
Request Address: `hubmib-request@synoptics.com`

Inter-domain Policy Routing WG

The group's primary task is development of architecture and protocols for policy routing among domains under different administrations. The Inter-Domain Policy Routing Protocol MIB is an Internet Draft dated July 1991.
Request Address: `idpr-wg-request@bbn.com`

IP over AppleTalk WG

The group's primary tasks are connection of AppleTalk systems to IP internets and distribution of AppleTalk services in IP internets. Their milestones include development of a relevant MIB, RFC 1243, a Proposed Standard as of July 1991. The working group is now collecting implementation experience. The have also produced an Internet Draft on SNMP over AppleTalk, dated December 1991.
Request Address: `apple-ip-request@apple.com`

IP over Large Public Data Networks WG

The group's primary task is to specify the operation of the TCP/IP protocol suite over public data networks, including SMDS, ISDN, X.25, and Frame Relay. The Frame Relay DTE MIB is an Internet Draft dated January 1992. Pending final draft it has been approved by the Network Management Directorate to be submitted for IESG and IAB approval as a Proposed Standard to be published as an RFC.
Request Address: `iplpdn-request@nri.reston.va.us`

IS-IS for IP Internets WG

The group's primary task is development of additions to OSI IS-IS routing for use in IP networks. The Integrated IS-IS MIB is an Internet Draft dated October 1991.
Request Address: `isis-request@merit.edu`

NOC Tools Catalog Revisions WG

The group's primary task is updating the existing catalog of tools for TCP/IP network managers. The current catalog is RFC 1147, an information document dated April 1990.
Request Address: `noctools-request@merit.edu`

Open Shortest Path First IGP WG

The group's primary task is development of the OSPF and promotion of its use. Their milestones include development of a relevant MIB, RFC 1253, a Proposed Standard as of August 1991.
Request Address: `ospfigp-request@trantor.umd.edu`

Point-to-Point Protocol WG

The group's primary task is development of a protocol to support IP over serial links. The Point-to-Point Protocol MIB is an Internet Draft dated September 1990.
Request Address: `ietf-ppp-request@ucdavis.edu`

Remote LAN Monitoring WG

The group's primary task is development of a MIB for monitoring LANs. The Remote Network Monitoring MIB is RFC 1271, a Proposed Standard as of November 1991. Implementation experience, system discovery, and token-ring extensions are under discussion. The group is holding a meeting this February.
Request Address: `rmonmib-request@mti.com`

RIP Version II BOF

The group's primary task is expansion of RIP, including a RIP MIB. The group met as a BOF in November.
Request Address: `ietf-rip-request@ftp.com`

Router Requirements WG

The group's primary tasks are to upgrade the existing Router Requirements document, RFC-1009, and to publish additional necessary documents, now including a forwarding table MIB. The Forwarding Table MIB is an Internet draft dated November 1991. The group recommends to the IESG that it be made a Proposed Standard and published as an RFC.

Request Address:
ietf-rreq-request@jessica.stanford.edu

SNMP Security WG

The group's primary task is to specify security services for SNMP. The group has three documents: SNMP Administrative Model, SNMP Security Protocols, and SNMP Party MIB. All are Internet Drafts since January 1992. These drafts have been submitted for IESG and IAB approval as Proposed Standards to be published as RFCs.
Request Address: snmp-sec-dev-request@tis.com

Trunk MIB WG

The group's task is preparing the DS1 and DS3 MIBs to become Draft Standards. The DS1 and DS3 MIBs are RFCs 1232 and 1233, respectively, both Proposed Standards as of May 1991. Clarifications and possible changes are being discussed on the mailing list.
Request Address:
trunk-mib-request@saffron.acc.com

X.25 MIB WG

The group's primary task is development of MIBs for the X.25 packet and link layers and for using IP over X.25, with consideration for ISO documents 7776 and 8208. The group has three documents: X.25 PLP MIB, HDLC/LAPB MIB, and IP over X.25 MIB. All are Internet Drafts dated October 1991. They are currently being edited per discussions at the November meeting. Discussions are active on the mailing list.
Request Address: x25mib-request@dg-rtp.dg.com

Recent Publications

Abstract Syntax Notation One: Tutorial and Reference
Douglas Steedman, Technology Appraisals Ltd., 1991. ISBN 1–871802–06–7

This is the authoritative, readable reference on ASN.1 language, written by one of its designers. Because the Internet-standard SMI is based on ASN.1, this is a good text for readers of management documents.

Integrated Network Management, II
Iyengar Krishnan, Wolfgang Zimmer, ed.,
North-Holland, 1991. ISBN 0–444–89028–9

This is the proceedings of the IFIP TC 6/WG 6.6 Second International Symposium on Integrated Network Management. It contains many papers regarding the Internet-standard Network Management Framework including:

- Network Management is Simple: You Just Need the "Right" Framework
- SNMP for Non-TCP/IP Sub-networks: An Implementation
- The Architecture of LANCE: A Simple Network Management System
- Secure Management of SNMP Networks

IEEE Network. ISSN 0890–8044

- SNMP Agent Support for SMDS (September, 1991)

ConneXions. ISSN 0894–5926

- Development and Integration of a MIB (June, 1991)
- SNMP Security (June, 1991)
- An SNMP Stereo System (April, 1991)

Activities Calendar

- 23rd Meeting of the IETF
  March 16–20, San Diego, CA
  For information: +1 703–620–8990
- Interop 92 Spring
  May 18–22, Washington, DC
  For information: +1 415–941–3399
- 24th Meeting of the IETF
  July 13–17, Boston, MA
  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.

Technical Review Board:
Dr. Jeffrey D. Case — SNMP Research, Inc.
University of Tennessee
Keith McCloghrie — Hughes LAN Systems, Inc.
Steven L. Waldbusser — Carnegie Mellon University

Featured Columnists:
David T. Perkins — SynOptics Communications, Inc.
Robert L. Stewart — Xyplex, Inc.

Contact Information

Postal: ***The Simple Times***
c/o Dover Beach Consulting, Inc.
420 Whisman Court
Mountain View, CA 94043–2112

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 to only 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 8mm `tar` tape, 1/4in `tar` cartridge-tape, or 3-1/2in MS-DOS floppy-diskette).

Each submission must include the author's full name, title, affiliation, postal address, 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 two forms: PostScript and MIME (the multi-media 822 mail format). 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).