The Simple Times™

THE BI-MONTHLY NEWSLETTER OF SNMP TECHNOLOGY, COMMENT, AND EVENTS^{SM}
VOLUME 1, NUMBER 3 JULY/AUGUST, 1992

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:

Technical Article

David L. Partain, Linköping University

In this issue: *An Implementation of SNMP Security*

In his *Security and Protocols* column in ***The Simple Times***, Keith McCloghrie discusses SNMP Security. In his first two columns, he briefly explained the new security mechanisms, outlined the protection that these extensions provide, and showed how the mechanisms are integrated into SNMP. In this issue, I report on my implementation of SNMP Security, demonstrating that the process is certainly feasible, and hopefully encouraging further fielding of SNMP Security software.

In keeping with the SNMP tradition of ensuring implementation experience prior to standardization, several implementations of SNMP Security were written while the proposals were Internet-Drafts. Implementation experience is essential to verify the soundness of the technology and to highlight those areas in the specifications which are unclear or perhaps not reasonably implementable. So, I wrote an implementation as a part of my Master’s work under Dr. Jeffrey D. Case at the University of Tennessee at Knoxville. The implementation is based on the 4BSD/ISODE SNMP package.

In this article, I briefly discuss modifications made to an SNMP implementation to incorporate the SNMP Security features. Further, we’ll examine the cost of realizing these changes, along with improvements made to the specifications during the implementation period. In this way I hope to provide future implementors with modest guidance for implementing SNMP Security in a performant and correct fashion.

The implementation as a whole progressed quickly and without serious difficulty. The largest portion of the time was spent in understanding the software platform and not in the actual coding. The methodology I chose was straightforward: I first altered the wrappers in the SNMP message and then implemented the party concept from the SNMP administrative model. These two changes essentially implemented noAuth/noPriv. I then added the Digest Authentication Protocol, the next logical step, followed finally by the Symmetric Privacy Protocol.

noAuth/noPriv

Recall from the original SNMP specification that the PDU is wrapped within a Message, which contains not only the PDU, but also the community string and the SNMP version number:

```
Message::=
    SEQUENCE {
        version
            INTEGER { version-1(0) },

        community
            OCTET STRING,

        data
            PDUs
    }
```

This is the sole wrapper. SNMP Security's innermost wrapper, the SnmpMgmtCom (SNMP management communication) includes the PDU along with the identities of the source and destination parties, but neither a community string nor a version:

```
SnmpMgmtCom ::=
    [1] IMPLICIT SEQUENCE {
        dstParty
            OBJECT IDENTIFIER,
        srcParty
            OBJECT IDENTIFIER,
        pdu
            PDUs
    }
```

The SnmpMgmtCom is in turn wrapped in a SnmpAuthMsg (SNMP authenticated message), which contains the authentication information (AuthInformation, which is used in an authentication protocol-specific manner), and the SnmpMgmtCom:

```
SnmpAuthMsg ::=
    [1] IMPLICIT SEQUENCE {
        authInfo
            AuthInformation,
        authData
            SnmpMgmtCom
    }
```

Finally, the SnmpPrivMsg (SNMP private message) contains the identity of the destination party and a possibly encrypted serialization of the SnmpAuthMsg:

```
SnmpPrivMsg ::=
    [1] IMPLICIT SEQUENCE {
        privDst
            OBJECT IDENTIFIER,
        privData[1]
            IMPLICIT OCTET STRING
    }
```

Thus, the first major step is to change from one SNMP wrapper to a wrapper inside a wrapper inside a wrapper.

The additional change to SNMP for noAuth/noPriv is the implementation of the party database. Recall that SNMP Security's administrative model is based upon the notion of an SNMP party, which can be thought of as the identity of a particular protocol entity running at a particular network location and in a particular security context. Of course, a particular protocol entity may operate as any of several parties (for example, one which uses no authentication and no privacy, and one which uses both), but each party uniquely identifies that protocol entity. This specificity contrasts with the community-based model used in the original SNMP, and is necessary in order to uniquely identify the source and destination of a message. An implementation of SNMP Security must of course implement a party database with all the relevant information for that party. There are many possible strategies for implementing the party database, but care should be taken to provide a stable database which is recoverable, i.e., after crashes.

The cost of implementing only noAuth/noPriv in comparison to the original SNMP is apparent. Three wrappers cost more in processing speed, agent complexity, and protocol complexity than a single wrapper. While this is true, each wrapper serves an essential role in SNMP Security. The destination party from the SnmpPrivMsg determines which privacy protocol to use, as this is based upon the destination. The source party in the SnmpMgmtCom determines the authentication protocol. Thus, each of the wrappers is necessary, despite the additional cost. Further, the cost of the party database, which could become quite large, cannot be avoided if SNMP Security is to be used at all.

The Digest Authentication Protocol

Implementation of authentication involved: including the code to generate the MD5 message digest; adding clock maintenance; and, coding the various steps taken to provide incoming and outgoing authentication.

In order to perform the MD5 message digest procedures, I extracted the appropriate code from the reference C Programming Language implementation in the MD5 specification (RFC 1321). The code integration required little effort. Naturally, optimization of the MD5 code for a particular hardware platform is desirable, if at all possible.

Implementation of the authentication steps required that I first manage loosely synchronized party clocks. Each SNMP party has its own party clock, and any outside parties which communicate with that party must keep their view of that party's clock loosely synchronized with its true value. This is done to protect against message reordering and replay attacks. I chose to maintain the party clock as an offset from the system clock where the agent was running, which eliminated the burden of having to maintain the clock manually.

The initial specification for the granularity of this clock was 100 ticks per second. Additionally, the *nonce*, essentially a sequence counter within one clock tick, permits 2^{32} uniquely identified messages to be sent per clock tick. Since it is unlikely that a party would exchange 2^{39} messages per second, the clock granularity was later changed to one tick per second. This still allows for 2^{32} messages to be sent per second while avoiding clock roll-over for over 100 years for those basing their clocks on a 32-bit system clock.

Since the correct operation of the MD5 message digest generation depends upon the private authentication key, implementors must take precautions to ensure that the keys with which they are dealing are in fact the required 16 octets. The initial MIB specification did not include this requirement (although it was stated elsewhere and is now in the MIB), and it is wise to take great care in ensuring correctness of the keys, just as with any value which must be of a specified length. This is also true with respect to the length of the private key used for the Symmetric Privacy Protocol.

The cost incurred by the Digest Authentication Protocol lies primarily in the cost of the digest generation code. A message digest over the serialized `SnmpAuthMsg` must be performed for both incoming and outgoing messages. Each implementor would be well advised to optimize this code as much as possible for the deployment platform.

A possible additional cost is incurred if one chooses to serialize the `SnmpAuthMsg` twice on outgoing messages. The `SnmpAuthMsg` is serialized once before the digest is generated with the private authentication key in the authDigest field. The authDigest field is then replaced with the computed digest. If the implementor does not wish to alter the serialized BER stream in place, the `SnmpAuthMsg` must then be serialized again. (In the initial implementation, I chose the twice-serialized approach. In the current implementation, serialization occurs exactly once.)

The Symmetric Privacy Protocol

The final stage of implementation, inclusion of the Symmetric Privacy Protocol, involved the integration of DES encryption code. The export control restrictions with respect to encryption technology, prompted the use of a public-domain DES implementation which is readily available outside the United States. Implementors should ensure that they understand the export and use restrictions on the Data Encryption Standard before shipping any SNMP Security code. (In brief, some countries limit the export and/or use of authentication and privacy functions. Accordingly, any implementor or user should seek the advice of counsel.)

One question, which did not arise when working on my implementation, dealt with which portion of the serialized `SnmpAuthMsg` is to be used for authentication and encryption. Simply put, the entire BER tag/length/value stream should be used.

Interoperability Testing

Upon completion of the implementation, interoperability testing was conducted with independent implementations written by Jeffrey Case and Keith McCloghrie. Interoperation was successful nearly immediately with all combinations of authentication and privacy.

Performance

I conducted several tests with both the SNMP Security implementation and the original SNMP implementation, in order to determine the impact on performance. Each test consisted of retrieving 18,000 variables to estimate the average number of variables retrieved per second that could be exchanged between an agent and manager running on the same SPARCstation I.

protocol	vars/sec	%-of 1157	%-of prev
1157 (SNMP)	60.97	n/a	n/a
noAuth/noPriv	37.97	62%	62%
md5/noPriv	32.13	53%	85%
md5/des	15.06	25%	47%

Based upon these results, it is apparent that there is a significant loss of speed with even noAuth/noPriv. I attribute this to the additional wrapper processing and the added complexity prior to processing the PDU.

Furthermore, in a given time period, the manager will be able to retrieve approximately 85% as many variables with md5/noPriv as when using noAuth/noPriv. This result confirms earlier estimates and appears to be a reasonable price to pay for authentication.

Finally, as would be expected, the use of the Symmetric Privacy Protocol greatly reduces the speed of variable retrieval. According to these tests, only 47% as many variables can be retrieved in a given time period when using privacy as with md5/noPriv. This drops to 40% when compared to noAuth/noPriv. There can be little doubt that hardware implementations of DES or highly optimized software would speed processing, but the degree of speedup is unknown.

Conclusions

From my experience in implementing and testing SNMP Security, I conclude the following:

- The technology proposed by SNMP Security, insofar as it has been tested in the field, is sound and implementable. The implementation process is quite straightforward. This in itself is valuable information.

- A critical factor in writing accurate implementations of SNMP Security is the clarity of the specifications. There can be little argument that the security mechanisms make the Simple Network Management Protocol significantly less simple. It is safe to say, however, that given the changes that occurred to the documents through the implementation process, the clarity of the protocols will lend themselves well to accurate implementations. The specifications for SNMP Security as of January 1992 did not have ambiguities which produced interoperability problems. It is essential that this be the case, and the interoperation of three independent implementations confirmed this to a large degree.

- SNMP Security, as stated by Keith McCloghrie in his column, is "not free." The performance statistics presented earlier demonstrate this. The cost of authentication, and especially privacy, will likely mean that noAuth/noPriv will be the most common form of network management communication. However, SNMP Security also provides the necessary mechanism for those wishing to manage their networks more securely.

- The implementation process confirmed the SNMP community's insistence that implementation precede standardization. Among the improvements, the process removed ambiguities in the specification, such as redundant terminology for the last authenticated message. Further, implementation demonstrated useful simplifications. The clock tick of one second is one such example. Finally, experience demonstrated the value of possible additions. For example, Jeffrey Case suggested the addition of a `partyMaxMessageSize` object to facilitate determination of the maximum message which can be accepted by a party. Such changes to the specifications would have been significantly more difficult to include had standardization already begun.

- Keith McCloghrie stated in his column in the previous issue of ***The Simple Times*** that "an agent implementation which followed the guidelines in the original SNMP protocol specification should be able to (effect) SNMP security with additional code but very few changes to the existing code." My implementation experience verifies this assertion. With the exception of wrapper changes and the removal of *trivial* authentication mechanisms, coding meant additions rather than changes.

Acknowledgements

Since working on this implementation, it has been incorporated into the 4BSD/ISODE SMP package. This software will be openly-available when the SMP specification is made available in early July. An announcement will be made to the `snmp` mailing list at that time.

The MD5 implementation I used is taken from RFC 1321, and is hereby identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm".

Industry Comment

Marshall T. Rose

Welcome to the third issue of ***The Simple Times***.

In this issue, the *Industry Comment* presents a perspective on SNMP evolution. But first, some subscription information: in his *Interoperability* column in the June 8th issue of *Communications Week*, Carl Malamud discussed ***The Simple Times***. In the following two weeks, about 200 more people subscribed for electronic distribution. The interesting part is that by the morning of June 10th, nearly sixty had subscribed — yes, there are clearly a lot of people who read *Communications Week* as soon as it hits their mailbox! Thanks to Carl and the usual trickle of subscription requests, there are now over 1000 electronic subscribers (including several re-distribution lists), with nearly 11% receiving the MIME edition.

Evolving the Internet-standard Network Management Framework

The Internet-standard Network Management Framework has achieved unprecedented success in providing interoperable solutions to the problem of managing networks. At the heart of this framework is the Simple Network Management Protocol which provides an effective means for monitoring and controlling heterogeneous devices. Although it was initially standardized in 1988, this management framework has been the subject of continuous incremental refinement. Paramount to this refinement has been the commitment to provide ongoing protocol-compatibility, so that the management environment evolves gracefully whilst the existing investment is protected.

In March of this year, the Internet Engineering Steering Group (IESG) issued a call for proposals to evolve the Internet-standard Network Management Framework. A fundamental observation made in this call is the understanding that the existing framework provides the foundation for stable and effective network management of the Internet. Further, these management capabilities are used pervasively and continuously. In other words, SNMP is an integral part of the Internet community's installed base.

At present, the Internet-standard Network Management Framework consists of three core technologies: a notation for describing management information (termed the "Structure of Management Information" or SMI), a collection of modules which define management information (each termed a "Management Information Base" or MIB), and, the management protocol, SNMP. Historically, the balance between stability and extensibility has been achieved by allowing only one kind of change: new MIB modules may be defined and existing ones may be revised.

In one response to the IESG's call, four people developed the *Simple Management Protocol* (SMP) Framework. The SMP specification and four independent, interoperable implementations are scheduled for release at the beginning of July. (Perhaps before you read this issue of ***The Simple Times***.) When the deadline nears for the IESG's call for proposals, the SMP authors will submit the current revision of the SMP specification for consideration.

Because other proposals may be forthcoming, rather than examining the SMP Framework, the *Industry Comment* looks at the issues associated with evolving the Internet-standard Network Management Framework.

Build on Success

An essential goal in any evolutionary scheme must be to build on the success of the current framework. To optimize the likelihood of this, it is important that the evolution be based on the same architectural principles as its predecessor. Although some may argue as to the precise details, there are three goals which provided the underlying guidance for the SNMP architecture:

- the impact of adding network management to managed nodes must be minimal, reflecting a lowest common denominator;
- network management must tend towards universal deployment; and,
- when all else fails, network management must continue to function, if at all possible.

Historically, it is clear that the SNMP philosophy of shifting the burden of management away from the managed nodes and towards the management stations, has allowed us to tend toward the first two goals. Further, the minimal communications infrastructure required by the SNMP (i.e., a connectionless-mode transport service), has allowed us to achieve the third goal.

A second part of building on the success of the current framework is for an evolutionary scheme to maximize backward-compatibility. That is, for each change under consideration, a careful cost/benefit analysis must be undertaken. Whilst the advantages of a feature are often evident, the impact on the installed base is often hidden. This means that for each change, the following questions must receive intense scrutiny:

- will the change affect management stations or agents?
- will the change result in a few or a large number of modifications?
- will those modifications be large or small?

Obviously, to be consistent with the philosophy of the current framework, the ideal change is one which has a minimal (or preferably no) impact on agents, and in which the modifications are well-localized.

In brief, when evaluating any evolutionary scheme, independent of its technical details, great attention must be given to the meta-issues of consistency and compatibility with the current framework.

Management Information

In February of this year, RFC 1303, *A Convention for Describing SNMP-based Agents*, was published. This describes a notation by which an implementor could document the features and limitations of an agent. This informational document met with a lot of interest, because it enables three different kinds of interactions: First, within an agent vendor company, RFC 1303 provides a means for engineering to concisely describe to marketing the features of their agent products. Second, RFC 1303 provides a means for users to evaluate and compare agent products. Third, RFC 1303 provides a means for management station implementors to customize their software to know about different kinds of agents. The way this third interaction works is simple: the RFC 1303 notation is machine-parseable, so an administrator runs a compiler that feeds the definitions into the management station. Because each kind of agent contains a unique identity code and RFC 1303 definitions include this information, as a part of its operations, the management station interrogates the agent and then sees if it has a RFC 1303 definition which corresponds to the kind of agent it is talking to.

As a part of evaluating RFC 1303, a compiler with an "agent evaluator" back-end was built. The algorithm used by the evaluator looks at the RFC 1303 definition of the agent's capabilities, assigns a rating from zero to one-hundred which represents the "goodness" of the agent implementation. The algorithm is limited in that it can evaluate the agent implementation only in a generic sense. In the prototype, when the rating is determined, it is displayed to the user and a different audio file is rendered. If the rating is zero, the message might result in:

> "You have an excellent agent — not!"

Similarly, a rating of twenty might be several people laughing, whilst a rating of forty might result in:

> "Bogus!"

and a rating of eighty might result in:

> "We can name that tune!"

(Of course, this example of the use of RFC 1303 is purposefully humorous.)

However, RFC 1303 is not without its drawbacks: in addition to requiring that implementors and users understand this new notation, management station implementors must build a compiler for the RFC 1303 notation and then instrument their software accordingly. Further, vendors of agent products might resist publication of descriptions of their agent implementations as this might provide marketing-fodder information to their competitors. Another drawback is that RFC 1303 limits its scope to agent implementations and doesn't consider user requirements. That is, the RFC1303 notation describes the capabilities of an agent, but doesn't have a way to describe the capabilities expected of an agent if it is going to operate in a particular user-environment. So, it would seem that we need both a way of specifying compliance issues in addition to agent capabilities. If we had notations for describing both kinds of information, then one could imagine that, in the future, one could write a program which could automatically compare both kinds of specifications in order to give a rough feeling for how well an agent implementation would work in the user's environment.

Administrative Framework

In terms of authentication and authorization, any evolutionary scheme will likely include the work on SNMP Security.

As Keith McCloghrie has pointed out in his *Security and Protocols* column (and as David Partain confirmed in his technical article on *An Implementation of SNMP Security*) security has both benefits and costs. The challenge for implementors is to provide turn-key solutions which hide the details and allow users to get on with the business of managing their networks.

However, one should keep in mind that even though the long-awaited SNMP Security work is largely consistent with the SNMP philosophy, it still needs a small bit of work. For example, SNMP Security, as presently specified, mandates ordered delivery for intra-party traffic. As Keith McCloghrie points out in his column in this issue: ordered delivery is largely unnecessary, and perhaps harmful, for retrieval operations; further, ordered delivery for intra-party traffic is inadequate for coordinating multiple managers performing modification operations. So, one might expect some additional work in this area.

Management Protocol

In terms of the management protocol, two issues seem to be at the forefront.

First, SNMP's set-request hasn't seen a great deal of operational use. There are probably two reasons: one is that some vendors have used the lack of SNMP Security as an excuse to avoid implementing the set-request. (This is, of course, specious as most vendors use a TELNET-based mechanism to modify a managed node. In addition to being no more secure than the original SNMP, because TELNET uses TCP, during times of network stress it is less likely to be able to control a device in comparison to using SNMP's set-request.) In addition to the "security" issue, when an SNMP set-request fails, very limited diagnostic information is returned. In brief, the management station asks the agent to do something, and the agent says:

> "No!"

What we really need is a richer collection of diagnostics, so the management station can determine if the failure is permanent or transient in nature, in addition to receiving a coarse indication of the cause. In other words, under any evolutionary scheme, it would be a good idea for the agent to be able to say:

> "No, because. . ."

Hopefully, introduction of SNMP Security and a somewhat richer diagnostic set will greatly increase the use of SNMP for modifying the behavior of managed nodes.

A second protocol issue that must be dealt with is the question of bulk retrieval. Historically, this has been one of the areas of greatest mis-understanding. The plain fact is that it is impractical to require an agent to provide an arbitrarily large amount of management information in a single transaction. Hence, a solution must be based on the notion of incremental retrieval. Today, there are several strategies, all of which make use of SNMP's get-next operator. Because of this, each strategy, regardless of how cleverly it makes use of parallelism and pipelining, is limited to retrieving a fixed amount of information in each transaction. This would seem to indicate that we need a new SNMP operator for bulk retrieval, one in which the agent helps to decide how much information is returned in a given transaction. However, great care must go into the design of such a facility, as it must not unduly burden the managed node.

The Open Question

Finally, there is one issue which needs a fair bit of thought. Although the Internet-standard Network Management Framework has been very successful in allowing us to instrument our managed nodes, it has been less successful in providing us with management applications.

Although there may be many causes for this, the one thing which seems clear is that management information is currently defined strictly on a micro-level. That is, we produce a lot of MIB modules containing a lot of managed objects; but nowhere do we produce documents describing how those objects can be used to provide for effective management. The result is that the majority of management applications are browsers. These browsers, regardless of the GUI, have little management smarts. (Steve Waldbusser discussed this situation in his *Applications and Directions* column in the previous issue.)

Achieving this task may be nigh impossible: first, the actual details are very often specific to individual environments; and, second, the actual details are also highly technical and (at present) not very amenable to machine-processing. However, figuring out a way of doing this may very well be the most helpful thing of all. The amusing part is that activity is probably outside the scope of any evolutionary scheme!

Applications and Directions

Steven L. Waldbusser

In this issue: *The Truth About SNMP Performance*

One of the most frequently repeated concerns about SNMP is that it won't perform well or won't scale up to the large networks of the future. This is often vocalized as "You shouldn't use SNMP because it isn't efficient enough and will clog the network." Unfortunately, such statements have not been supported by technical rationale or by direct user experience, and this has left consumers very confused. There are many large sites using SNMP today to manage networks. These sites have direct experience that should ease the uncertainty in this area.

There are two areas to be addressed separately:

- overall network load for routine monitoring, and,
- efficiency in downloading large amounts of data.

In addition, there are some up and coming advances that will make the situation even better.

Routine Monitoring

Bill Yeager of Stanford University did some tests with SNMP to find out the real story. He designed and implemented a test to simulate a worst-case scenario using SNMP. His results showed that to monitor all the available performance and error information on a host at the interface, IP and TCP layers every 5 minutes, an average of 16 bytes per second were transferred. When one scales this up to the monitoring of a large site at which 400 routers, hubs, and file servers might be monitored, one finds that this would use 6400 bytes per second of bandwidth. This is equivalent to a half of one percent of an ethernet bandwidth. This is clearly not going to cause any performance problems. Commercial products are often more optimized and present even less of a load on the network. It should also be noted that today's monitoring applications typically look at much less data on each node than tested here — the results of this test would be typical of the increased demands placed by more sophisticated applications in the future.

Carnegie Mellon University also has a large network, on which SNMP has proven to scale very well. An SNMP monitoring tool monitors more than 200 devices on the Carnegie Mellon network, polling each one once every 15 seconds for status information. This also uses much less than one percent of an ethernet's bandwidth — so little in fact, that a second computer performs the identical task as a backup. Both computers spend less than 5% of their processing on these SNMP tasks, which shows that high-powered management station hardware is not required.

Downloading Large Amounts of Data

Another area of concern is the speed at which SNMP can transfer large amounts of data. Some MIBs contain tables with many entries that might account for hundreds of kilobytes of data. It is important to be able to achieve interactive performance when retrieving this data. Because the Remote Network Monitoring (RMON) MIB often stores large amounts of utilization and error information about network devices, it is important that it can be transferred very efficiently with SNMP. When Karen Frisa implemented an RMON MIB agent at Carnegie Mellon, and John Chanak developed some RMON MIB tools, they had the opportunity to transfer large amounts of information using SNMP. For example, a host table on a segment of the Carnegie Mellon campus usually grows to more than 1100 entries. It was possible to download this table in roughly 2 seconds. Because there were 6 variables per entry being downloaded, this resulted in a transfer rate of 3060 variables per second. Similarly, when downloading packets captured by RMON's filter mechanism, 780 packet entries were downloaded per second. If the packet data alone was downloaded (ignoring related data such as length and status), the rate rose to 2082 per second. In this application, enough of each packet was downloaded (64 bytes) to perform a summary decode. These results show dramatically that with SNMP, bulk data can be downloaded quite quickly.

A barrier to the fast download of data is the discovery of previously unknown instances of data. Before asking for the value of a variable, a management station must know its name. The SNMP get-next operator allows the discovery of the names of such variables, but unless a sophisticated algorithm is used, only one instance may be discovered per packet (such a sophisticated algorithm is described in RFC 1187, *Bulk Table Retrieval with the SNMP*). The RMON MIB was specifically designed to allow another method of discovering instances of data quickly. However, many other MIBs exist that weren't designed with this in mind. It would be desirable to provide a fast and easy mechanism to download data from any MIB. The newly defined Simple Management Protocol (SMP) provides such a mechanism, called the get-bulk operator. This operator allows the discovery of many variables per packet, speeding the transfer of data and making it more efficient by packing more in every PDU. Initial testing shows that this operator will improve on the blazing speeds cited above, and will also make normal operations more efficient, requiring less network load than the tests mentioned previously.

On The Horizon

These results show that SNMP is quite capable of scaling to the very large networks of today as well as the larger ones of tomorrow. This scaling can be achieved without overloading critical segments with network management traffic. When fast interactive performance is required for downloading large amounts of data, SNMP can do the job when the management station or the MIB has had the smarts built in. The new Simple Management Protocol will provide the means for any data to be quickly downloaded. Don't let marketeers with a hidden agenda steer you away from the integrated network management possible with SNMP and SMP!

Ask Dr. SNMP

Jeffrey D. Case

Dear *Dr. SNMP*,
I recently read that the new Simple Management Protocol (SMP) will have a mechanism for the efficient retrieval of large amounts of data. Will this new mechanism be the one described in RFC 1187, *Bulk Table Retrieval with the SNMP*, or will it be the one described in the premiere issue of ***The Simple Times***? Will it be fast and efficient?
P.S. Will *Dr. SNMP* soon become *Dr. SMP*?
— *Impatient in Indianapolis*

Dear *Impatient in Indianapolis*,
Back on the farm, we have a saying:

"That thing's faster than a scalded dog."

The answer is yes, it will be fast and efficient. The new bulk retrieval mechanism uses neither of the mechanisms you ask about. Instead, it uses a new operator, called get-bulk, which has been optimized to communicate requests for the transfer of large quantities of data. Responses are communicated via the usual SNMP response mechanism.

Regarding your postscript, SMP really is SNMP. But, the SMP authors couldn't just call it that, because the name "SNMP" belongs to the Internet community. The SMP authors tried to be careful to avoid offense by using a different name. (They were perhaps too sensitive.) It is anticipated that, over time, it will be acceptable to call SMP what it really is, SNMP version 2.

Dear *Dr. SNMP*,
Recently I've been hearing about this new management protocol, X.700, which is supposed to be well-suited for wide-area network (WAN) management. The same people who told me about X.700 also tell me that since SNMP is a protocol for managing local-area networks (LANs), that I need both protocols for network management.
— *Vacillating in Virginia*

Dear *Vacillating in Virginia*,
Back on the farm, we have a saying:

> "You can give a doggie bone to a pig but it still won't hunt."

(Dr. SNMP thinks this is right up there with "A leopard can't change its spots" and "A zebra can't change its stripes.")

What this means is that calling something by a different name won't yield fundamental changes in its characteristics and behavior. Your question brings four important ideas to mind.

First, there are those who are attempting to avoid the excess baggage associated with the name of the OSI management framework, anchored by CMIP. Consequently, they are attempting to use a fresh, new name (X.700), in order to avoid many of the negative connotations and emotions associated with the "CMIP" label. Dr. SNMP is somewhat sympathetic toward the notion of using new names for existing protocols. However, the rationale for using "SMP" instead of "SNMP version 2" is entirely different than the motives used in renaming CMIP to X.700: SMP really is SNMP, with a few problems corrected and a few important enhancements. SMP uses the same basic well-engineered framework enjoyed by SNMP but these minimal changes lead to dramatic results. In contrast, the X.700 framework really is the CMIP framework, albeit without any problems corrected and without any important enhancements. The motivations for the name changes are quite different.

Second, Dr. SNMP is less charitable toward the dis-information campaign that seems to be underway. This dis-information campaign attempts to position SNMP as a LAN management technology. The truth is that while SNMP has become popular as a LAN management technology, the original impetus for the design was the monitoring and control of wide area network components, especially IP routers. SNMP can, has, and will continue to perform this function in many production networks.

Third, the dis-information campaign attempts to position X.700 as a superior WAN management technology. The truth is that the requirements of WAN management are incompatible with the characteristics and performance of connection-oriented transports in the lossy environments frequently encountered in wide area networks. In other words, CMIP is actually better suited for use in LANs than it is for use in WANs.

Finally, the dis-information campaign appears to be too well organized and orchestrated to be an accident. Perhaps the next dis-information you hear will be that you need a different protocol for manager-to-manager communications than is used for manager-to-agent communications.

It is difficult for Dr. SNMP to see how dissimilar communications technologies and techniques will be helpful. Building translators between these frameworks (SNMP and CMIP) is a difficult problem that many have underestimated. The entity models are different. The information models are different. The naming mechanisms are different. The SMIs are different. The protocol operations are different. The transport assumptions are different. Other than that, they both have managers and agents!

In conclusion, you do not need two different protocols for managing LANs and WANs.

Security and Protocols

Keith McCloghrie

In previous issues, we have looked at the protections provided by SNMP Security and discussed how introducing the concept of a SNMP party allows the three primary mechanisms: the MD5 message digest algorithm, the DES encryption algorithm, and loosely synchronized clocks to be integrated into the protocol. In this issue, we'll discuss some issues involved in implementation and deployment.

Using Parties

Each SNMP party is unique to the particular SNMP protocol implementation where it executes. Thus, many parties need to be defined. The chosen way to do this is to identify them by OBJECT IDENTIFIERs (OIDs), of which there is an infinite supply! This allows anyone to obtain a branch in the OID tree, and allocate party OIDs within that branch.

However, to simplify matters, a set of six initial OIDs have been assigned for use with each IP address, three for local execution at an agent, and three for the agent to communicate with (i.e., at a manager). The three have different settings of authentication and privacy algorithms, with an appropriate MIB view and access control parameters defined for each. The extension of these six to the number actually required in an agent can, of course, be done through the use of SNMP requests acting on appropriate MIB objects.

The definitions of new encapsulation schemes for SNMP, e.g., over OSI, are also defining their own conventions for initial Party OIDs using their own addressing schemes. Note, however, that the use of an address as part of an OID is purely administrative, as one means of providing uniqueness; there is no requirement to have a relationship between protocol stack addresses and party identifiers.

Indeed, for agents which need more than six parties, the party OIDs for the additional parties would typically not be allocated from the `initialPartyId` branch, but from some other branch (e.g., from a OID subtree within the vendor's tree of the management station which is being used to create them). The only requirement to be met when assigning OIDs is to make them unique across the network.

These initial parties need six secrets. As it turns out, all six are the same length. Thus, at initial distribution, all six secrets can have the same value. This does not impair security because all six values should be immediately changed by the management station as soon as secure communication begins. Changing the secrets thereafter is desirable on a relatively frequent basis. When changed, there is no need for humans to be informed of the new values. In fact, it is better security if humans are not informed. Humans are typically lazy, and thus are unlikely to change secrets at the desired frequency. Thus, it is a good practice to have the secrets which are in frequent use changed automatically.

Some parties may be set-up for special use, for example: for use in emergencies by network fire-fighters who may wish to access an agent from wherever they may happen to be at the time. The secrets for these parties do not need to be changed periodically, but can be left unaltered ready for use at a moment's notice.

Using Clocks

The clock value for each party must increase with the passage of time, even across reboots. If these clock values are maintained as offsets from a system clock, this is not such an implementation burden as it might appear. While it is vital that clock values are never decreased (in order to maintain protection against replay), speeding them up is explicitly allowed. For example, in times of network stress, a manager can artificially advance its notion of a party's clock so that even though communication delays may have increased dramatically, a message will still be considered authentic when it arrives at an agent.

As was discussed in a previous article, the use of clocks ensures message timeliness within the limit specified by the lifetime. The specifications also include another clock-based mechanism, called ordered delivery. This mechanism specifies that messages delivered out-of-order be discarded as unauthentic. While this has some benefit for set-requests, there is potential for this to be harmful when applied to retrieval requests. As such the inclusion of ordered delivery has been questioned, but no one wants to further delay the specifications, so these arguments are moot at this time. Due to the inclusion of ordered delivery, another variable (called the nonce) is introduced to distinguish multiple requests generated within one tick of the clock (i.e., within one second).

Using Secrets

Both the MD5 authentication and the DES privacy algorithms for a party rely on secrets, which must be known by both the originator and the recipient. If these algorithms are to maintain their level of security, then their secrets must remain secret and not be available to would-be attackers. So, they cannot be transmitted over the network in clear text form. Strictly speaking, this requires the use of encryption. However, the MIB objects for these secrets do not represent them in clear text, but rather as the XOR-encoding of the previous and new values in set-requests, and as zero-length strings in get-requests. Thus it is possible, though not strictly conformant to the specification, to change secrets without using encryption. The more significant security issue for implementations which do not include an encryption capability is the setting-up of new parties, when the XOR-encoding of the new secret (with the null string) provides no protection from eavesdroppers. Indeed, until two SNMP protocol entities share a secret, secure communication across the network is not possible. Thus, security requires that initial secrets be distributed manually, generated perhaps by the management station, and entered into the agent as a piece of initial configuration information. This enables secure communication, so that subsequent distribution of secrets, either for new parties or for the regular changing of secrets of existing parties (which is very desirable from a security standpoint), can be done via SNMP access to appropriately secured MIB objects.

The inclusion of DES may be problematic for some implementors because of export regulations. While products incorporating DES can be exported from most countries, the inclusion of DES may incur additional complications. As such, it is to be expected that some implementors may choose not to include DES in their implementations, especially since conformance to the specification only requires DES for access to party secrets and, as mentioned above, the requirement to use DES for such access is most significant when establishing new parties, as opposed to changing the secrets of existing parties.

Specification Status

Finally, you might be wondering about the status of the specifications. The author has it on good authority that these documents will be published as RFCs with a status of proposed standard by mid-July.

Standards

David T. Perkins

In May and June, there were no new SNMP-related standards that were approved. In the pipeline are the *IP Forwarding Table* MIB and the three documents defining SNMP Security. Only after these documents are published as RFCs with proposed standard status will they be included in this column.

In the previous issue, the process which is used in the IETF to develop standards was described. In this issue, the heritage of the SNMP standards is discussed, and in the next issue, we'll look at the standards process for IEEE committee 802.

Summary of Sources and Internet Standards which they Influenced

From the International Organization for Standardization (ISO), three documents provided some initial influence on the Internet-standard Network Management Framework:

- *Abstract Syntax Notation One* (ASN.1), ISO 8824;
- *Basic Encoding Rules* (BER), ISO 8825; and,
- *Common Management Information Protocol* (CMIP), ISO 9595/9596.

The documents produced by IEEE committee 802 has had significant influence on several media-specific MIBs:

- *Token-Passing Bus Access Method and Physical Layer Specifications*, 802.4, was used as the input to the working group that produced RFC 1230, *IEEE 802.4 Token Bus Interface Type MIB*;
- *Token Ring Access Method and Physical Layer Specifications*, 802.5, was used as the input to the working group that produced RFC 1231, *IEEE 802.5 Token Ring Interface Type MIB*;
- *Media Access Control (MAC) Bridges*, 802.1d, was used as the input to the working group that produced RFC 1286, *Bridge MIB*; and,
- *CSMA/CD Access Method and Physical Layer Specifications*, 802.3, and *802.3 Layer Management*, 802.3h, were used as the input to the working groups that produced RFC 1284, *Ether-Like Interface Type MIB*, and RFC 1271, *Remote LAN Monitoring MIB*.

Finally, from ANSI committee X3T9, revision 6.2 of the *FDDI Station Management (SMT)* document was used as the input to the working group that produced RFC 1285, *FDDI Interface Type MIB*.

From this, it can be seen that IEEE 802 has been a major influence on SNMP standards. The IEEE is currently in the process of developing a management protocol, named LAN/MAN Management Protocol (LMMP). LMMP, sometimes termed CMIP over LLC (CMOL), is based on CMIP running on top of the IEEE 802 connectionless logical link control (LLC type 1). At present, it appears that the LMMP work in IEEE 802 will not be used in the Internet standardization process, but the management information documented in IEEE 802 will continue to be used as input to IETF standards development process.

This issue has listed the SNMP standards that have been significantly influenced by documents from other organizations. The next issue will present the process that develops the IEEE network management standards.

Summary of Standards

Full Standards:

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

Draft Standards:

- 1212 - Concise MIB definitions.

Proposed Standards:

- 1229 - Extensions to the generic-interface MIB;
- 1230 - IEEE 802.4 Token Bus Interface Type MIB;
- 1231 - IEEE 802.5 Token Ring Interface Type MIB;
- 1232 - DS1 Interface Type MIB;
- 1233 - DS3 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;
- 1284 - Ether-Like Interface Type MIB;
- 1285 - FDDI Interface Type MIB;
- 1286 - Bridge 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; and,
- 1318 - Parallel Printer Interface Type 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);
- 1238 - CLNS MIB;
- 1283 - SNMP over OSI; and,
- 1298 - SNMP over IPX.

Informational:

- 1147 - A network management tool catalog;
- 1215 - A convention for defining traps for use with the SNMP;
- 1303 - A convention for describing SNMP-based agents; and,
- 1321 - MD5 message-digest algorithm.

Historical:

- 1156 - Management Information Base (MIB-I).

Working Group Synopses

Robert L. Stewart

Once again the working groups supplied plenty of discussion, with the prize going to the SNMP mailing list, which doesn't even have a working group. Be aware that the following synopses present my condensation of many statements by many people. Although I try to present the flavor and summary of the discussion as it occurred, I do not include either direct quotes or attribute, nor do I edit for correctness. If you want to know who really said what, subscribe to the mailing lists. There's no substitute for being there.

SNMP Mailing List

Someone designing a MIB asked the best way to report an absolute time, suggesting either the UNIX format, `DisplayString`, or elapsed `TimeTicks`. A respondent pointed out that all three approaches had both good and bad points. This issue was left unresolved.

Someone asked about good objects to monitor to detect network congestion. One opinion held that monitoring `icmpInSrcQuenchs` and `icmpOutSrcQuenchs` was a good idea. Another respondent noted that SNMP is not well suited to congestion control which requires intelligence in hosts, routers, and bridges. The response that SNMP can monitor congestion got agreement, if using `ifOutDiscards` or `ifOutOctets`, with the caveat that source quench itself is optional and defaults to being disabled. A different respondent said they use the ICMP objects because agents running on UNIX often cannot supply `ifOutOctets`.

Someone asked for a UNIX library for MIB-I, MIB-II and private MIBs, and wondered if the API could be standardized. Someone else suggested built-in SNMP for UNIX, e.g., an extensible agent, and SNMP and ASN.1 libraries, all of which could be produced by an IETF working group. A respondent noted that implementation specifications are not consistent with the IETF mission.

Some asked if, in promiscuous mode, should all packets be counted in the MIB-II interfaces table, or just those packets addressed locally? The Interface Extensions MIB says that its receive address table holds only the addresses used in non-promiscuous mode. So, the behavior depends on whether the interface being modeled is a MIB-II interface. In the case of repeater ports, which are not MIB-II interfaces, counting is not done; but, if both a bridge and router are using the same interface, and it is thus promiscuous, all packets are counted.

Someone asked how does an NMS determine the maximum message size that an agent can accept, and on error, should the NMS retry progressively smaller sizes or simply drop to the minimum size? The recommended procedure is to guess initially, and then start decreasing until the complaints stop. Unfortunately, a long, string-valued, variable may make the problem insoluble. A followup asked that since the request is usually smaller than response, how does an agent know the maximum response size? The answer was that the SNMP Security Party MIB includes a maximum message size.

Someone asked if enumerated INTEGERs could be negative, even though they may not be zero-valued. The answer was that use of negative values appears to be within the letter, but not the spirit, of the SMI. So, negative-valued enumerations should not be used.

There was a fairly long discussion on the parallel processing of incoming SNMP requests which produced many warnings for agent implementors.

Someone asked if order of processing of the variable bindings in a set-request was important. In particular, can a manager make any assumption about the order in which processing will occur. A respondent indicated that the processing must occur "as if simultaneously". Another respondent observed that there should be no external way to detect the order of processing.

Someone offered to the public domain a utility to monitor a logfile and send SNMP traps under conditions from a regular expression. The host is `wuarchive.wustl.edu` and the file is `/pub/log2sd.tar.Z`

There were some questions about EMANATE, a technology for multiplexing agents on a single host, which was announced in the trade press. These questions were referred for off-line discussion as EMANATE is a commercial product used as a local-mechanism and is out of customary IETF concerns.

Someone asked for real data on SNMP overhead. A respondent indicated that a test using a home-grown tool based on the CMU package showed that polling for key objects at reasonable periods of 5 and 10 minutes generated negligible Ethernet overhead and very useful information. The experimenter concluded that monitoring up to 200 systems this way was not intrusive. (See this issue's *Applications and Directions* column for further discussion.) Another respondent suggested traps backed by polling for a "last changed" time-stamp, carefully designed to avoid fast-changing objects such as normal message counters.

A message concerning the *Communications Week* story on SMP, a proposed new version of SNMP, indicated that the SMP specification and four implementations will be available before a planned Birds-of-a-Feather (BOF) at the Boston IETF. The message then went on to plead for no follow-up questions to the four SMP authors as they have a lot of work to do by then. The *Communications Week* reporter asked for community reactions. Although there was some concern over previous contributions of the proposers giving their work too much weight, the general consensus was that having relatively complete work by competent people was a good starting place for the open process.

Someone asked how an agent should respond if it does not implement an object in a MIB group which is mandatory for the agent's managed node. The immediate, first reply was that the agent should return noSuchName. Then began an interminable, thundering flame war, replete with denigration of individuals and companies over a topic that has been discussed much the same way several times over the past four years. The "dogmatic architectural purists" maintained the strict position, with justification based on the intention to have a stable, predictable base for advanced network management applications. The "heretical pragmatic iconoclasts" held that returning a static value promoted easy compliance and was necessary due to many existing implementations that lacked real information and had to interoperate with NMSs that treated noSuchName as a serious failure. Those of the "Inquisition" called such responses lies and declared such NMSs broken, and the "heretics" retreated behind interoperability and marketing pressure. This discussion will no doubt replay itself in the future, perhaps at the Boston IETF.

Someone noted that statistics in the *Internet Monthly Report* showed an error rate on one network that was 2000 times better than a second network, and then asked if this was possible. One respondent indicated that the information was probably true but could be misleading; for example, 9870 errors on a DS1 could indicate a single burst. Another respondent said the numbers were for a single interface, not a network, and that the "good" interface was an Ethernet, while the other interfaces were DS1 circuits. A followup asked if a new MIB object was needed to indicate what is expected as "normal". The responses which followed indicated that the concept of "normal" was murky.

Finally, there was a very long discussion about the effect of export restrictions on SNMP's new security mechanisms. The discussion started with several questions: are there any changes in U.S. export regulations, is authentication useful without privacy, will public domain implementations have privacy, can an implementation be compliant without privacy, how will the market react if privacy is omitted, why hasn't the press caught on to this situation, and can the IAB or IETF help. The ensuing discussion was rife with fear, uncertainty, and doubt.

Bridge MIB Working Group

The working group will meet in Boston to consider changes and elevation to Draft Standard.

Chassis MIB Working Group

Someone observed that when adding a repeater port to a box that has other functions, it gets a LOT more complicated, e.g., requiring implementation of the Chassis MIB, the Repeater MIB (possibly for multiple repeaters), and the Party and View MIBs. However, under such a model, how can one identify a particular repeater port given the IP address of the agent. The one respondent indicated that a MIB view model would work, and that the Party MIB could be useful for this.

DECnet Phase IV MIB Working Group

A NMS provider asked for an agent to test against, but received no public responses.

In response to several questions he had received, the editor said there is no counter for multicast bytes sent because it was not in Phase IV DECnet; however, such an object will be added to the list for the next edit. The editor also solicited responses from those interested in interoperability testing in the Fall.

Finally, although the working group has concluded its charter, the mailing list will remain for implementation discussion.

Domain Name Service MIB

The mailing list received a new DNS MIB draft in PostScript and ASCII forms. The new version of the MIB is now almost 50% smaller.

Ethernet MIB Working Group

The chair asked if the group needs a meeting at the Boston IETF, solicited the participation of IEEE 802.3 people in the group, and asked if there were any objections to advancing the MIB forward. There were no public responses.

The mailing list has moved to `enet_mib@ftp.com`

FDDI MIB Working Group

Someone asked if the FDDI trap document had been completely dropped. The answer, as agreed to by the working group, was yes.

Host MIB Working Group

Someone asked if the presentation made at the San Diego IETF could be distributed to the mailing list. In response, a strawman proposal was distributed.

The working group was officially formed, chartered to define "SNMP MIB objects that instrument characteristics common to all internet hosts".
Request Address: `hostmib-request@andrew.cmu.edu`

IP over Large Public Data Networks (IPLPDN) Working Group

Questions about IP over X.25 were referred to the X.25 MIB working group.

Someone seeking information on an ISDN MIB got two responses: first, a masters student is writing such a MIB as a part of thesis work, with the final form due May 14 prior to submission to the IETF; and, second, a company is working on IP over ISDN experiments, with results one or two months out.

The Boston meeting agenda includes the Frame Relay MIB and the X.25 MIB.

Multiport Repeater MIB Working Group

Someone suggested changing the syntax of `rptrPortAdminState` to be consistent with other MIBs. As all other standard MIBs are that way, the editors agreed. A followup asked a similar question about `rptrPortAutoPartitionState`. The editors agreed, if there were no objections from working group. There were no public responses.

For the meeting at the Boston IETF, the only agenda item is MAU MIB.

Someone wanted to know if on-line detection of health or failure could be done reliably and with common meaning. A respondent indicated that such information is very implementation-specific. Further discussion brought a proposed wording change, pending editor approval.

A new Internet-Draft is available with all changes.

NOCtools Working Group

A message sought tools and volunteers to update entries, asking that replies be sent to
`noctools-request@merit.edu`

OSPF Working Group

For the meeting at the Boston IETF, the agenda includes discussion on the MIB and traps.

PPP Working Group

A solicitation for MIB comments received the response it looks good so far but is still too big.

Someone asked that since the counter `pppLinkPacketTooLongs` counts DISCARDED packets longer than MRU, should it count packets that are too long but not discarded? In response, the MIB was changed to include "too long for any reason which results in a loss of information or lack of communication". If such frames are discarded, then their count is also included in `ifInErrors`.

There was a long and detailed discussion of reusing `ifIndex` for internal PPP layers which included analysis of code and data size. Someone argued against filling the MIB-II interfaces table with internal PPP details, and also noted that counting everything was, in general, a bad policy. There were also objections to optional objects or objects relating to unfinished PPP features being required in all MIB implementations.

Distribution of a new MIB draft elicited comments on various specific objects and a general objection to the MIB's technique for presenting optional or negotiated features. The conclusion was to word the MIB so that it doesn't imply a particular implementation strategy.

The editor announced separate documents available as Internet-Drafts for: LCP, Security, IP, and Bridging, and solicited comments at the Boston IETF.

Remote Network Monitoring (RMON) MIB Working Group

Someone asked if it is valid to add a new control table entry for either invalid or noSuchName entries, which is preferred, and are there other methods? A respondent indicated that new entries must be created with createEntry status, preferably by attempting to set status for a random index. A followup objected that a random index won't work in an agent with a small, fixed table. The original respondent noted that the agent should accept any unused index and map it to internal entries.

Someone asked for a way to stop data collection but leave results in place. A respondent suggested a "freeze" EntryStatus but noted that this is inconsistent with concurrent use by multiple managers and prefers a way to snapshot.

Someone asked how possible inconsistencies among interdependent objects should be dealt with, and went on to suggest various ways to interpret the NMS intent, such as order sensitivity. This drew the response that agents can't do the NMS job and should ignore inconsistencies not prohibited by the MIB specification. (It was further noted that SNMP prohibits order sensitivity.)

A question about RMON extensions for higher level protocols received the reply that such work is scheduled after the Token Ring RMON work completes.

A long discussion of sensing stations in a token ring reached no clear conclusion.

Someone asked if `historyControlDataSource`, (an OID for an interface) and `channelIfIndex` (an INTEGER for an interface) should be the same. A respondent said the two objects are different because the former will someday point to other things but the latter may not.

SNMP Security Working Group

Someone asked that when creating a new MD5 party, if a DEFVAL of the empty string could be used for the shared secret. The response was that the empty string is appropriate for noAuth parties, but is inappropriate for MD5 parties.

Someone asked about a working group meeting at the Boston IETF. After some discussion, it was decided to have an implementor's BOF instead.

Someone asked when a MIB view should be evaluated. A respondent noted that is an implementation decision, but that it is sensible to evaluate VarBinds as they are processed, keeping in mind that evaluation for get-next occurs after the processing, not before.

A late-breaking message said the final Internet-Drafts have been approved for standardization by the IAB.

X.25 MIB Working Group

Someone asked about archives. The response is that they are kept on the host `dg-rtp.dg.com` in the directory `x25mib/`.

All three documents were reissued with various changes, mostly suggested by editor, and with little group discussion.

Activities Calendar

- 24th Meeting of the IETF
  July 13–17, Boston, MA
  For information: +1 703–620–8990
- SMP BOF (at the Boston IETF)
  Wednesday, June 15, 7:00–10:00pm
  For information: +1 703–620–8990
- 25th Meeting of the IETF
  November 16–20, Washington, DC
  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:

Dr. Jeffrey D. Case	SNMP Research, Inc. University of Tennessee
Keith McCloghrie	Hughes LAN Systems, Inc.
David T. Perkins	SynOptics Communications, Inc.
Robert L. Stewart	Xyplex, Inc.
Steven L. Waldbusser	Carnegie Mellon University

Contact Information

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