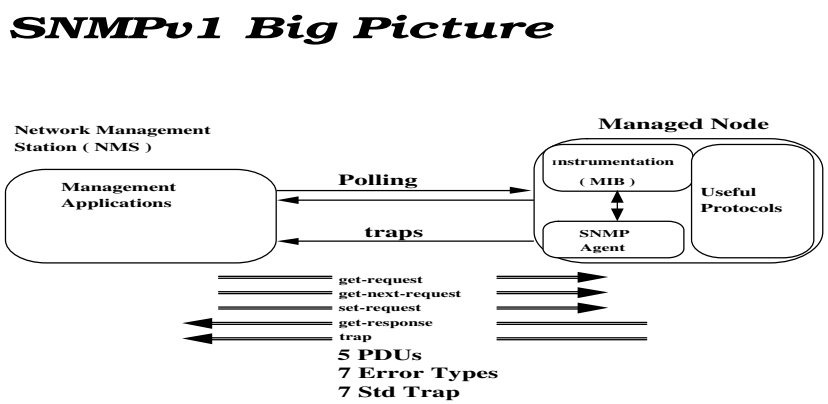
A Guided Tour Through SNMP Version 2

by Bill Norton, Merit Network Inc.

Abstract

The Simple Network Management Protocol version 2 (SNMPv2) has been adopted as a proposed standard by the Internet Engineering Task Force. Through features like enhanced security, extended messages, more detailed error indications, more efficient bulk data dumps, etc., the new SNMPv2 network management framework provides the basis for the next generation of network management tools. This paper discusses the major differences with and enhancements to SNMPv1. The audience is expected to have at least a cursory understanding of SNMPv1 as defined in the Internet Standard Network Management Framework.

SNMP Version 1 Review



The first version of SNMP was a messaging protocol with a vocabulary of five messages as displayed in the above figure. The network management station would issue a get-request message to the managed node to retrieve some instrumentation from the managed node through a SNMP agent. The SNMP agent, upon receiving the message would check perform some authorization checks, retrieve the instrumentation, and return the data to the NMS. The get-next message is similar, but returns the lexigraphically next piece of management information than the one requested. Through this traversal operator, the entire instrumentation can be explored. Instrumentation can be "set" to affect an operation. The last message is an asynchronous notification (trap) that something has occurred that warrants further investigation.

Since the preferred transport mapping for SNMP is UDP, the NMS must deal with the potential loss of SNMP messages. In particular, since SNMP traps are unreliable, the NMS must implement a trap-directed polling approach. In this approach, the NMS must poll for all important information and not count on receiving all asynchronous alerts. If the NMS does happen to receive a trap, it should initiate diagnostic routines. This approach is termed trap-directed polling.

Each SNMP message is encoded using the Asbstract Syntax Notation 1 (ASN.1), which provides a platform independent method for exchanging data structures. Associated with the abstraction language is a data encoding methodology known as the Basic Encoding Rules (BER). The BER detail how data structures defined using ASN.1 are encoded into a serialized stream of octets.

SNMP v2 Evolution

History

- **Several IETFs - NM AD-SNMPv1 problems?**
- **March 92 - RFP to evolve Framework**
- **July 92 - SMP BOF**
  - **Gang of Four - Case, McCloghrie, Rose, Waldbusser**
- **Nov 92 - SNMPv2 WG begins - few changes**
- **Feb 93 - SNMPv2 moved to draft**
  - **SNMPv1 - defined in 3 RFCs**
  - **SNMPv2 - 12 RFCs - That s progress for you**
- **Few implementations as commercial produ**

This Internet Standard Network Management Framework is elegant in its simplicity, but contains a slew of deficiencies. These deficiencies were enumerated by the IETF after several years of deployment experience. As a result, in March 1992 the IETF Network Management Area Director (at that time, Chuck Davin) issued a Request for Proposal seeking technology to evolve the framework . In July 1992, a gang of four responded to the RFP with a series of specifications called *SMP: The Simple Management Protocol.* The proposal was presented at a BOF and quickly accepted as the starting point for what would become SNMP version 2. While the working group was given control of the spec, changes were not accepted by the authors. Antagonistic tactics were used, threats were made, and the community involvement in the progress of the specification was minimized. While the series of 12 rfcs that define SNMPv2 have been drafts for a year, few commercial implementations have been marketed. One can only postulate that the lesson learned is that the bastardization of the standardization process can not produce market-accepted solutions.

SNMPv2 Enhancements

What is new in SNMPv2?

- **New PDUs**
- **Added Security**
- **Greater information in Errors**
- **Row creation mechanism defined**
- **Textual conventions defined**
- **Manager-of-managers (M2M MIB)**
- **Proxy relationships**
- **SNMPv2 MIB**
- **more...**

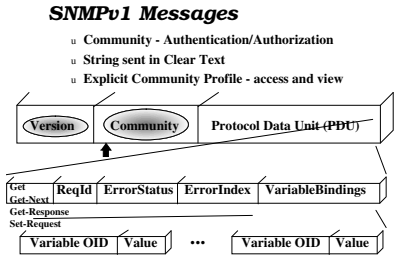
SNMPv2 defines three additional message types: get-bulk, snmpv2Trap, and informRequest. The get-bulk PDU is an optimized get-next, performing multiple get-next operations in a single operation. The snmpv2Trap provides asynchronous alerts to the NMS under the new framework. Finally, the inform request provides acknowledged asynchronous notifications, designed to be used in a manager-manager interaction.

The greatest technological advantage to using SNMPv2 over SNMPv1 is the security framework added to the model. This will be discussed in detail later on.

Other features of the new framework include sticky topics like how conceptual rows are created. In the past, several methods existed, and none were consistent or universally accepted. *Textual conventions* are defined adding semantics to managed objects that are designed to facilitate the creation of virtual rows.

Manager to manager communications are enhanced through the informRequest PDU and the Manager-Manager MIB. The Manager-manager MIB defines a method for a manager to direct a subordinate manager to monitor certain managed objects and generate an alert when the threshold has been exceeded.

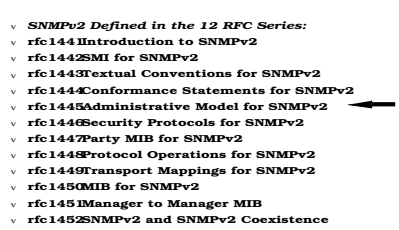
SNMPv1 Messages



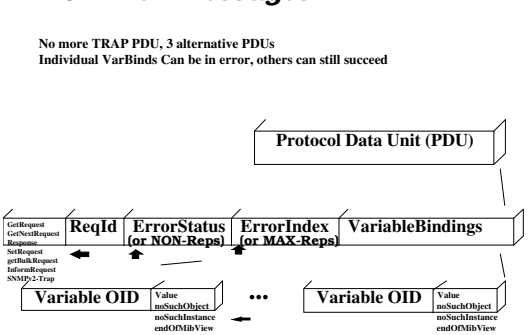
Recall that SNMPv1 messages contain three fields: *version*, *community*, and *Protocol Data Unit (PDU)*. The PDU should be thought of as requests, and there are five types of requests defined, four of which are listed above. Requests are tagged with a Request ID so that the NMS can correlate responses with outstanding requests. Skipping over the error fields, we find the Variable Bindings (VarBinds). The VarBinds consist of a variable-value tuple that will, if the request is successful, contain both the data requested. Requesters always ignore the error-status and error-index fields, and responders always set these fields appropriately. If an error occurs processing or performing the requested action, the responder sets error index to the VarBind causing the error, and error-status indicating the error.

This is all defined in the 3 RFCs known as the Internet Standard Network Management Framework.

SNMPv2 Specifications

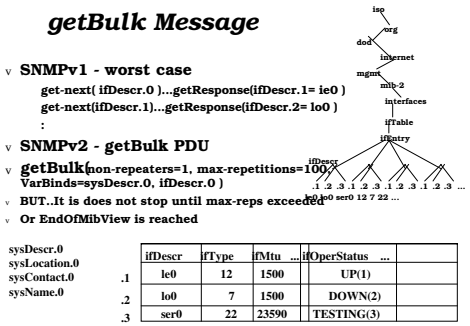


SNMPv2 is quite a bit more complex than SNMPv1, and requires 12 RFCs to adequately define the features. A good starting point for the interested reader is rfc1445-Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 represents an *evolutionary* effort, and this document does a good job of describing the additions.



SNMPv2 messages look a lot like SNMPv1 messages with a few additions and a deletion. There are a few new PDU types as previously described, and the SNMP trap has been replaced by the SNMPv2 trap. Note also that the error-status and error-index field now serve double duty providing the additional getBulk fields. More importantly, the value field of each VarBind can be set to an error value, providing more detailed error information in responses. Remember that SNMPv1 responses indicate success or error, and any error yields all VarBind values to be undefined. SNMPv2 responses provide a mechanism for providing both variable-value tuples and variable-error tuples in the same response.

The "awesome" getBulk PDU



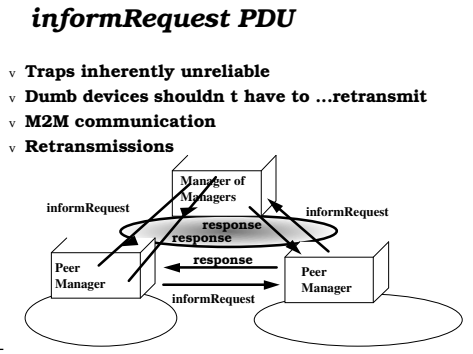
Recall that the management information in the MIB is organized in a tree hierarchy, and get-next requests provides the traversal operator necessary to browse the management data. The problem with using the get-next operator for this operation is that it will take up to n+1-message transactions to retrieve the MIB! (There are ways to optimize get-next operations by using multiple var-binds in a single get-next request.)

The SNMPv2 solution is the getBulk PDU. The NMS specifies the maximum number of repetitions (MAX-REPS) and the object identifier starting point. The Agent performs up to MAX-REPS get-next operations on the MIB and returns all the varbinds that can fit in a single PDU. The NMS can also request the getBulk to also perform single (non-repeated) get operations. This is accomplished by specifying the number of non-repeater varBinds (NON-REPS). The first NON-REPS varBinds are treated like get requests, and the remainder of the varBinds are treated like multiple get-nexts as described above. Pro grammatically, one can view this operator as a subroutine:

getBulk(NON-REPS, MAX-REPS, VarBind [,VarBind, ..., VarBind])

There is some inefficiency with this solution, however. Since the getBulk operations does not stop until MAX-REPS (or until the end of the MIB View is reached), the NMS is bound to get some varBinds returned to him that are outside the range of what he wants. For example, if there was substantial work to be done to retrieve each piece of data (proxy over low speed dial-up, high-impact transactions, etc.), the overhead for retrieving unwanted data could end up wasting a tremendous of resources. This topic was the source of many heated debates.

Manager to Manager Features



The informRequest PDU provides acknowledged asynchronous alerts. Many configurations can benefit from the informRequest PDU including hierarchical and peer-to-peer management configurations. The Manager-Manager MIB defines thresholding, alert generation and event queuing mechanisms. The manager performing these activities can be configured to generate informRequests to peer or parent managers. Retransmissions are configurable.

Security Features in SNMPv2

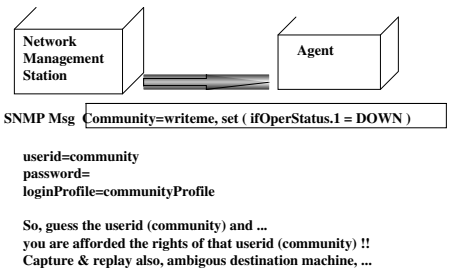
SNMPv2 Security

- **Authentication**
- **Parties**
- **Party Database**
- **How Authentication Works**
- **Privacy Wrapper**
- **Clock Synch**
- **NVM Requirements**
- **Secret Distribution**

SNMPv2 replaces the SNMPv1 security framework (community name) with a substantially more powerful security framework providing authentication and privacy services for management communications. The following slides will touch upon the above terms as they apply to the SNMPv2 security framework.

Problems to Overcome in SNMPv1

Problem with SNMPv1



The most fundamental problem with SNMPv1 security is the fact that all you need to do is guess the community name, and you are afforded all rights associated with that community name. This is analogous to having a network addressable computer with user ids with no passwords. Further, these messages can be captured and replayed at anytime. Resetting a device interface for example, might be performed at a later time by replaying the SNMPv1 set request. There is no temporal setting within which these operations are performed, nor are the endpoints of communication defined. Authentication deals with these two issues.

Authentication

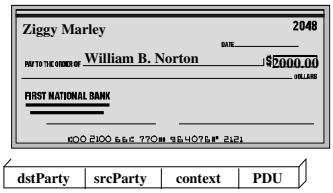
- **Identify the source and destination of the communication**
- **Elements of the new security model**
  - **SNMPv2 Communication**
  - **SNMPv2 Parties**
  - **SNMPv2 MIB Views**
  - **SNMPv2 Access Policies**

The SNMPv2 approach to security is to keep the PDU form, identify the endpoints of communication, and "wrap" this message in the appropriate security packages. If authentication is required, an

authentication wrapper is required. If privacy is required, the message should be wrapped in a privacy package. Let's look at a basic SNMP management communication.

SNMP Management Communication (snmpMgmtComm)

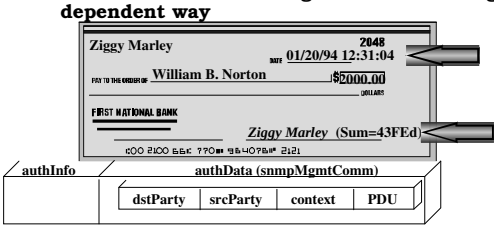
- **a disambiguated SNMPv2 Operation**
- **Parties: 2 endpoints of communication**
- **a check with the payer, payee, and amount filled**



A management communication is similar to a SNMPv1 message with the source and destinations identified. This is analogous to a check with the payee and payer spelled out, instead of writing a check to "Cash". An associated context field is also included to specify an operating environment. This is similar to providing a separate read-only and read-write community with corresponding MIB views, etc. in SNMPv1.

SNMP Authenticated Message (snmpAuthMsg)

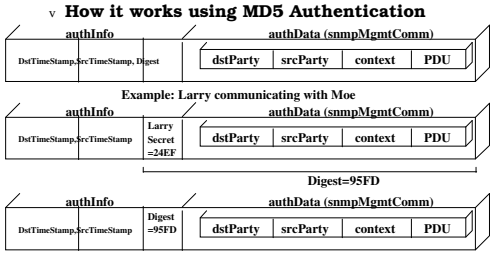
- **represents a SNMPv2 operation reliable origin and integrity**
- **a check that is dated, signed in a technolog dependent way**



All SNMPv2 message require an authentication wrapper. If no authentication is required for the SrcParty, no authentication is performed. If authentication is required for the SrcParty, the authInfo field contains vital information for authenticating the SrcParty. Included in this message are time stamps (analogous to dating the check), checksums, etc. whatever is needed by the authentication methodology in use for the SrcParty. Note that the source of the message is authenticated.

In the old days, letters were delivered with a wax seal across the back. This seal was stamped with a hot metal stamp thus certifying not only that the letter was authentic (indeed from the person who owns the stamp) but also that the contents of the letter have not been altered in any way. In the same way, authenticated SNMPv2 messages are sealed with a secret seal described next.

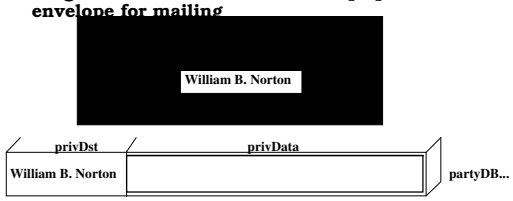
SNMP Authenticated Message



First, we must assume that there are secrets known by only two parties that will communicate with authentication. Messages from A to B will be encoded with A's authentication method. A will place its notion of both the source and destination timestamps into the packet. A will place A's secret in the digest field of the packet and compute a digest across the entire message. This calculated digest will then overlay the digest field in the message. When B receives the message, the time stamps are checked to see that this message has not been seen before (replay attack), and the digest is saved elsewhere. B's notion of A's secret is placed in front of the message, and the digest is calculated. If the digest matches the digest originally contained in the message, the message is deemed authenticated and of known integrity (not modified).

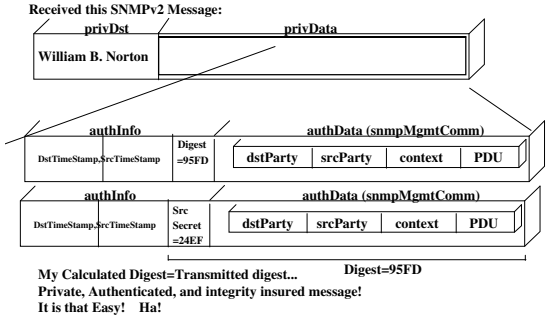
SNMP Privacy Message

- **a SNMPv2 message protected from Disclosure**
- **useful for exchanging secrets, captured RMON packets, etc**
- **a signed check in a more or less opaque, addressed envelope for mailing**



SNMPv2 privacy provides management communication an encrypted wrapper, thus preventing the content of management communication from disclosure. The only unencrypted piece of information is the dstParty, which is needed to determine which privacy method and secrets to use.

SNMP Security Realized



Walk through the process of private, authenticated communication. Note that noAuth and noPriv are valid authentication and privacy mechanisms for less secure communication.

Access Control List

- **snmpMgmtComm context used as key**

Target	Subject	Context	Privileges
gracie	george	local	35 (Get, GetNext & GetBulk)
george	gracie	local	132 (Response & SNMPv2-Trap)

Defines the rights afforded to the parties in the given context.

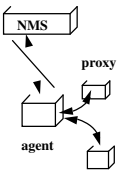
Analogous to the read-only/read-write community name configuration, but more flexible.

Get	1
GetNext	2
Response	4
Set	8
---unused	16
GetBulk	32
Inform	64
SNMPv2-Trap	128

Associated with each SrcParty-DstParty-context triple is a set of privileges. These privileges define what operations are allowed with respect to the two parties when in that execution context.

Context and View Table

- **From Context, determine view**



Identity	View Index	ProxySrcParty	proxyDstParty	....	
local	1				
peer	2	me_n_my	shadow		
box	2	laurel	hardy		

Index	Subtree	Mask(signif. subids)	Type ...
1	1.3.6.1.2.1	B	inclusive
1	1.3.6.1.4.1.25	1111101 B	inclusive
1	1.3.6.1.2.1.2.1	B	exclusive
2	1.3.6.1.4.1		inclusive

Collectively, MIB View

Associated with each context is a view, that consists of potentially many sub-views. These sub-views can be combined to include or exclude specific branches of the MIB. This is also where proxy relationships are defined. This is basically a proxy forwarding table in this role, and follows the "transparency principal".

Proxy Relationships

- **Foreign Proxy - manage non-snmp boxes**
- **Native Proxy - distribute management**

Wrap-up

Other Features of SNMPv2

- **18 Error Status Values**
- **Instance level granulation**
- **MOM communication**
- **Textual Conventions**
- **SNMPv2 MIB (partyDB, contexts, views,**
- **Manager-to-manager MIB**

SNMPv2 References

- **rfc1441-rfc1452 - anonymous FTP**
- **The Simple Book, 2nd edition, Marshall T. R**
- **SNMP, SNMPv2, and CMIP, William Stallings**
- **The Simple Times, Electronic Publication, st subscriptions@simple-times.org**
- **snmp@psi.com**
- **snmpv2-request@magellan.TIS.COM**