Comparison of SNMP

Versions 1, 2 and 3

Eddie Bibbs

Brandon Matt

ICTN 4600-001

Xin Tang

April 17, 2006

During its development history, the communities of researchers, developers, implementers and users of the DARPA/DoD TCP/IP protocol suite have experimented with a wide range of protocols in a variety of different networking environments. The Internet has grown, especially in the last few years, as a result of the widespread availability of software and hardware supporting this system. The scaling of the size and scope of the Internet and increased use of its technology in commercial applications has underscored for researchers, developers and vendors the need for a common network management framework within which TCP/IP products can be made to work.

In recognition of this need, several efforts were started to develop network management concepts which might be applied to the Internet and to the internet technology in general. Three of these efforts had made sufficient progress by the end of 1987 that it became clear that some choices had to be made or the community would find itself with a set of incompatible network management tools. These efforts included the High-Level Entity Management System (HEMS), the Simple Gateway Monitoring Protocol (SGMP) and the Common Management Information Service/Protocol.

In the short term, however, the Internet desperately needs tools to apply to the operational management problems associated with it rapid growth. Given the present state of advanced implementation of the SGMP and its relative simplicity, the general agreement was that SGMP (or its re-named successor, SNMP) should be quickly brought to more complete specification for widespread implementation and use. Soon after, Simple Network Management Protocol (SNMP) succeeded SGMP for its ease and versatility.

SNMP was a protocol developed to manage nodes (including servers, workstations, routers, switches and hubs and any other peripheral device) on a network. SNMP is an application protocol that is encapsulated, or encased, in the User Datagram Protocol (UDP). UDP is a connectionless transport layer (layer 4) protocol in the OSI model that provides a simple and unreliable message service for transaction-oriented services. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

An SNMP managed network consists of three key components: managed devices, agents, and network-management systems (NMSs). A managed device is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers. An agent is a network management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP. An NMS executes applications that monitor and control managed devices.

Currently, there are three versions of SNMP defined: SNMP v1, SNMP v2 and SNMP v3. Both versions 1 and 2 have a number of features in common, but SNMPv2 offers enhancements, such as additional protocol operations. SNMP version 3 (SNMPv3) adds security and remote configuration capabilities to the previous versions. To solve the incompatible issues among different versions of SNMP, RFC 3584 defines the coexistence strategies.

SNMP v1 is the initial implementation of the SNMP protocol. SNMPv1 operates over protocols such as User Datagram Protocol (UDP), Internet Protocol (IP), OSI Connectionless Network Service (CLNS), AppleTalk Datagram-Delivery Protocol (DDP), and Novell Internet Packet Exchange (IPX). SNMPv1 is widely used and is the *de facto* network-management protocol in the Internet community.

SNMP is a simple request/response protocol. The network-management system issues a request, and managed devices return responses. This behavior is implemented by using one of four protocol operations: Get, GetNext, Set, and Trap. The Get operation is used by the NMS to retrieve the value of one or more object instances from an agent. If the agent responding to the Get operation cannot provide values for all the object instances in a list, it does not provide any values. The GetNext operation is used by the NMS to retrieve the value of the next object instance in a table or a list within an agent. The Set operation is used by the NMS to set the values of object instances within an agent. The Trap operation is used by agents to asynchronously inform the NMS of a significant event.

Now version 1 wasn't without its problems. The main problems of the version 1 are the authentication of the message source, protecting these message from disclosure and placing access controls on the Management Interface Base (MIB- this is a logical database made up of the configuration, status and statistical information stored at a device) database. SNMP v2 was designed in 1993 and was to be an evolution of its predecessor. The Get, GetNext, and Set operations used in SNMPv1 are exactly the same as those used in SNMPv2. However, SNMPv2 adds and enhances some protocol

operations. In SNMPv2, if a multiple requested value, in a get-request , one is not valid or does not exist, there will be answers for the other request that have been well dealt, whereas for version 1, no response at all was given, only the error message. In version 1, traps had a different format than all of the other PDUs. Version2 simplify traps by giving them the same format as the get and set PDUs.

SNMPv2 also defines two new protocol operations: GetBulk and Inform. The GetBulk operation is used by the NMS to efficiently retrieve large blocks of data, such as multiple rows in a table. GetBulk fills a response message with as much of the requested data as will fit. The Inform operation allows one NMS to send trap information to another NMS and to then receive a response. In SNMPv2, if the agent responding to GetBulk operations cannot provide values for all the variables in a list, it provides partial results.

The last area that SNMP v2 was to improve was security, and this led to the proliferation of SNMPv2 version "variants". Since there are in fact several different "SNMPv2"s, there are also several message formats for SNMPv2, not just one. This is confusing enough, but would be even worse without the modular nature of SNMP messages "coming to the rescue". The protocol operations in SNMPv2 were changed from SNMPv1, which necessitated some modifications to the format of SNMPv2 PDUs. However, the protocol operations are the same for all the SNMPv2 variations. The differences ***between*** SNMPv2 variants are in the areas of security implementation. Thus, the result of this is that the ***PDU*** format is the same for all the SNMPv2 types, while the overall ***message*** format differs for each variant. (This is why the distinction between a PDU and a message is not just an academic one!)

Now, later on a new version of SNMP was released to cover some of the security issues that plagued version 2. SNMP v3 Framework augments the original SNMP and the SNMPv2 specifications with additional security and administration capabilities. SNMP Version 3 (SNMPv3) adds security and remote configuration capabilities to the previous versions. The SNMPv3 architecture introduces the User-based Security Model (USM) for message security and the View-based Access Control Model (VACM) for access control. The architecture supports the concurrent use of different security, access control, and message processing models. SNMPv3 also introduces the ability to dynamically configure the SNMP agent using SNMP SET commands against the MIB objects that represent the agent's configuration. This dynamic configuration support enables addition, deletion, and modification of configuration entries either locally or remotely.

The general message format for SNMPv3 still follows the same idea of an overall message “wrapper” that contains a header and an encapsulated PDU. However, in version 3 this concept is further refined. The fields in the header have themselves been divided into those dealing with security and those that do not deal with security matters. The “non-security” fields are common to all SNMPv3 implementations, while the use of the security fields can be tailored by each SNMPv3 security model, and processed by the module in an SNMP entity that deals with security. This solution provides considerable flexibility while avoiding the problems that plagued SNMPv2.

There are five types of application which can be associated with an SNMP engine. These applications are Command Generators, Command Responders, Notification

Originators, Notification Receivers, and Proxy Forwarders. Along with these new applications, there was a new User-based Security Model for SNMP v3. t defines the Elements of Procedure for providing SNMP message-level security. The USM protects the user against four threats, which are modification of information, masquerade, message stream modification, and disclosure (optionally). The USM uses MD5 (Message Digest Algorithm) and the Secure Hash Algorithm to provide data integrity, to directly protect against data modification attacks, to indirectly provide data origin authentication, and to defend against masquerade attacks. It also uses Data Encryption Standard (DES) to protect against disclosure.

Another new feature not in previous installments is the addition of the View-based Access Control Model for SNMPv3. It defines the Elements of Procedure for controlling access to management information. The VACM can simultaneously be associated in a single engine implementation with multiple Message Processing Models and multiple Security Models.

The latest development with network management and SNMP is expanding to all forms of hardware. Today's latest software includes desktops as well. The Desktop Management Interface (DMI) is a good example. The DMI is another piece of software that is next to the SNMP protocol but not replace it. The DMI functions reach for more of the software level where SNMP is mostly hardware. DMI is very similar to SNMP. Instead of a MIB the DMI uses the Management Information Format know as a MIF. Simple algorithms are used to transform MIF into a MIB and transport through SNMP protocol. The DMI will provide a better API in the Operating System to get information out of the hardware and software. The reason for DMI is new hardware. The latest

firmware in today's technology provides specific application calls that can be used to get information about the hardware. The term DMI can be used in place of API for hardware. DMI is providing a standard procedure for using SNMP to talk to the hardware. The DMI is very useful when wanting data that is very dynamic. For example, if an administrator would like to see the amount of I/O on a nodes hard disk or the current utilization of the processors, the management station will make a request using the DMI management interface (MI) through the SNMP protocol to the DMI component interface (CI) which will make the appropriate API call to the hardware. The CI will then get the answer and use the DMI's MIF to transport the information back to the management station. This can be done due to the standard DMI calls through the CI. These calls are standard for any platform. The management station can request processor use for any node using a standard procedure call for all nodes to their CI. This will decrease the amount of platform specific information the management station is required have. The reason for DMI is to get network management back to an industry standard.

Security is important when using SNMP. Because SNMP agents broadcasts information and in some agents changed, security must not be overlooked. The initial version of SNMP now referred to, as SNMPv1 did not have a very good implementation of security. SNMP faces all the standard threats of any network application: Modification of Information, Masquerade, Message Stream Modification, and Disclosure. Here is a brief overview of the security measures with each of the versions of SNMP.

SNMPv1 used only one form of security, community names. Community names are similar to passwords. Agents can be set to reply to queries only received by accepted

community names. In SNMPv1 the community name was passed along with the data packet in clear text. This allowed anyone to ease drop and learn the SNMP community name or password. SNMPv2 brought a lot of extra security. First of all everything in the packet except for the destination address is encrypted. Inside the encrypted data is the community name and source IP address. The agent can now decode the encrypted data packet and use the accepted community name and accepted source IP address to validate the request. This type of security is referred to as party and context. Party referring to a specific machine or person and context referring to a name or string associated with the party. SNMP uses DES (Data Encryption Standard) for encrypting the data packets. SNMPv3 provides the latest architecture for SNMP security. It incorporates an SNMP context engine ID to encode and decode SNMP contexts. The context engine ID could take more time than allowed to explain. In short it matches a context name with an object and the security requires the object and context to match. SNMPv3 provides three levels of security. The highest level is with authentication and privacy. The middle level is with authentication and no privacy and the bottom level is without authentication or privacy.

The perfect example of why SNMP security is important is its ability to reboot devices. Administrators cannot let that ability be violated. The latest version of SNMP have brought security a long way from clear text.

Reference:

1) Cisco Systems, 2002, Simple Network Management Protocol;
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm

2) The TCP/IP Guide, 2001, SNMP
http://www.tcpipguide.com/free/t_SNMPMessageFieldDefinitionsGeneralMessageFormatand.htm

3) Javinn.com, 2005, SNMP: Simple Management Network Protocol

http://www.javvin.com/protocol/rfc1155.pdf : Structure and Identification of Management Information for TCP/IP based internets

http://www.javvin.com/protocol/rfc1156.pdf : Management Information Base Network

http://www.javvin.com/protocol/rfc1157.pdf : A Simple Network Management Protocol

http://www.javvin.com/protocol/rfc1441.pdf : Introduction to SNMP v2

http://www.javvin.com/protocol/rfc2579.pdf : Textual Conventions for SNMP v2

http://www.javvin.com/protocol/rfc2580.pdf : Conformance Statements for SNMP v2

http://www.javvin.com/protocol/rfc2578.pdf : Structure of Management Information for SNMP v2

http://www.javvin.com/protocol/rfc3416.pdf : Protocol Operations for SNMP v2

http://www.javvin.com/protocol/rfc3417.pdf : Transport Mappings for SNMP v2

http://www.javvin.com/protocol/rfc3418.pdf : Management Information Base for SNMP v2

http://www.javvin.com/protocol/rfc3410.pdf : Introduction and Applicability Statements for Internet Standard Management Framework

http://www.javvin.com/protocol/rfc3411.pdf : Architecture for Describing SNMP Frameworks

http://www.javvin.com/protocol/rfc3412.pdf : Message Processing and Dispatching for the SNMP

http://www.javvin.com/protocol/rfc3413.pdf : SNMP Applications

http://www.javvin.com/protocol/rfc3414.pdf : User-based Security Model (USM) for SNMP v3

http://www.javvin.com/protocol/rfc3415.pdf : View-based Access Control Model for the SNMP

http://www.javvin.com/protocol/rfc3584.pdf : Coexistence between SNMP v1, v2 and v3

4) Bob Stewart, 1995, SNMP version 2, http://www.ietf.org/html.charters/OLD/snmpv2-charter.html

5) Wikipedia.com, 2005, Simple Management Network Protocol, http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol

6) Fusion, 2002, Fusion SNMP, http://www.dspos.com/DSPOSWeb/network_mgmt/fusion_snmp.htm