A Study of Web-Based SNMP Network Management
with a Simple Java Applet Network Monitoring Tool

Submitted to Committee Members
Dr. W. Homer Carlisle(Chair)
Dr. Kai H Chang
Dr. Alvin S. Lim

Submitted
by
Jianqun Chen

as a Partial Fulfillment of
the Requirement for the
MCSE Project

Department of Computer Science and Engineering
Auburn University, Alabama
January 29, 1998
# TABLE OF CONTENTS

LIST OF TABLES
LIST OF FIGURES
LIST OF ABBREVIATIONS

1. Introduction .......................................................... 1

2. Simple Network Management Protocol (SNMP) ................. 2
   2.1 SNMP Model ....................................................... 2
   2.2 SNMP Protocol Operations ....................................... 6
   2.3 SNMP Data Representation ....................................... 7
   2.4 Management Information Base (MIB) and Object Identifier (OID) ........................................... 7
   2.5 Structure of Management Information (SMI) Definitions ................................................. 10
   2.6 More Issues of SNMPv2 ......................................... 10
   2.7 SNMP Proxy Agents for the Management of non-SNMP Devices ........................................... 12

3. Web-Based Network Management ................................... 13
   3.1 Java Web-Based Management .................................... 14
   3.2 Other Newly Emerging Approaches ................................. 17

4. Conventional SNMP Management vs Web-Based Management .... 20

5. Writing Java Web-Based Network Management Applications for the World Wide Web ........ 23
   5.1 Technical Overview of the Advent SNMP Java Classes Package ........................................ 24
   5.2 A Java SNMP Application ........................................ 26
   5.3 A Simple Java Applet Network Monitoring Tool .............................................. 26
       5.3.1 The Need for Network Monitoring ........................................ 26
       5.3.1 The Environment of the Applet ........................................ 27
       5.3.3 Prototype of the Applet Network Monitoring ........................................ 29
       5.3.4 The Use of the Applet Network Monitoring ........................................ 30

6. Conclusions and Future Work ....................................... 36

7. References ............................................................ 38

8. APPENDIX ........................................................... 40
LIST OF TABLES

Table 1 The Object Groups of the Internet MIB-II .......................... 9
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>SNMP Management Architecture</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>MIB Tree</td>
<td>8</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Download and Execution of a Java Applet</td>
<td>15</td>
</tr>
<tr>
<td>Figure 4</td>
<td>The Environment for SNMP Applet</td>
<td>28</td>
</tr>
<tr>
<td>Figure 5,6,7</td>
<td>Snapshots of the Applet Monitoring Tool</td>
<td>32, 33, 34</td>
</tr>
<tr>
<td>Figure 8</td>
<td>The Interface of the Advent MIB Browser</td>
<td>35</td>
</tr>
</tbody>
</table>
### LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>API</td>
<td>Application Programmer’s Interfaces</td>
</tr>
<tr>
<td>ASN.1</td>
<td>Abstract Syntax Notation One</td>
</tr>
<tr>
<td>AWT</td>
<td>Abstract Windowing Toolkit</td>
</tr>
<tr>
<td>BER</td>
<td>Basic Encoding Rules</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>CMIP</td>
<td>Common Management Information Protocol</td>
</tr>
<tr>
<td>HMMP</td>
<td>Hyper Media Management Protocol</td>
</tr>
<tr>
<td>HMMS</td>
<td>Hyper Media Management Schema</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standardization Organization</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>JMAPI</td>
<td>Java Management API</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>OID</td>
<td>Object Identifier</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>MIB</td>
<td>Management Information Base</td>
</tr>
<tr>
<td>NMS</td>
<td>Network Management System</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>RFC</td>
<td>Request for Comments</td>
</tr>
<tr>
<td>RMI</td>
<td>Remote Method Invocation</td>
</tr>
<tr>
<td>SAS</td>
<td>SNMP Applet Server</td>
</tr>
<tr>
<td>SMI</td>
<td>Structure of Management Information</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SNMPv1</td>
<td>Simple Network Management Protocol Version 1</td>
</tr>
<tr>
<td>SNMPv2</td>
<td>Simple Network Management Protocol Version 2</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Data Protocol</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VRML</td>
<td>Virtual Reality Modeling Language</td>
</tr>
<tr>
<td>WBM</td>
<td>Web Based Management</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
</tbody>
</table>
1.0 Introduction

Networks and distributed computing systems are of growing importance, and indeed, have become critical for the success of many organizations in the world. With the growth of network interconnection today, the management of network and its associated resources and distributed applications becomes more important than ever before. With the size, complexity and heterogeneity of today’s multi-protocol and multi-vendor computer networks, network management has become more challenge. [1]

Web technologies are accelerating rapidly today. A trend that has begun on the Internet is to integrate all kinds of information systems into the World Wide Web(WWW) or intranet environment. This is also very interesting for network management and monitoring. One of the emerging technologies is the use of Web technology for the network management purposes. Such an application of the Web technology for network and device management is called Web-based Network Management. The Web-based management is the use of a Web server and browser technology, and other Internet technologies, for monitoring, troubleshooting and reporting enterprise network and system management information. Web-based management is a relatively new technology with a few products on the horizon, but it is expected to be a revolutionary network management solution that will change the way how users manage their networks. [7]
2 Simple Network Management Protocol (SNMP)

The Internet and the OSI networking communities have developed a network management architecture, which define the structure of network management information and protocols to retrieve and manipulate management information provided by agents running on the network elements. The well-known network management protocols include the Simple Network Management Protocol (SNMP) of the Internet Protocol Suite (TCP/IP), which is widely used in data network, and the ISO Common Management Information Protocol (CMIP) for use in public telecommunication networks. [20]

2.1 SNMP model [2][3][4]

The Simple Network Management Protocol (SNMP) is an application-layer protocol designed as an Internet protocol for exchanging network management information between network devices, allowing network administrators to manage network performance, retrieve and modify network status information, solve network problems, plan for network growth by using SNMP-transported data, such as the packets per second and network error rates.

Today, SNMP is the most popular protocol for managing diverse commercial internetworks as well as those used in universities and research organizations. There are two versions of SNMP: Version 1 (SNMPv1) and Version 2 (SNMPv2). Most of the changes introduced in Version 2 increase SNMP’s security capabilities. SNMP-related standardization activity continues as vendors continue to add SNMP-based management applications.[3]
The Figure 1 illustrates the SNMP management architecture. The SNMP management architecture is based on the interaction of many entities:

(1) Managed nodes (or called Managed devices) -- managed nodes can be any devices such as hosts, routers, bridges, printers or any other devices that are connected to networks. They are capable of communicating status information to the outside world. To be managed directly by SNMP protocol, a node must also be capable of running an SNMP management process and have such a management process running. Currently, all computers meet this requirement, as
do increasingly many bridges, routers and peripheral devices designed for network use. [18]

(2) SNMP Agents -- SNMP Agents are software modules that are running as management processes on managed devices. They compile information about the managed devices in which they reside, collect and store management information into a local management database, for example, the number of error packets received by a network element, and provide this information (proactively and reactively) to management entities within the network management system (NMSs) via the network management protocol.

(3) Managed object -- A managed object is a parameter or an attribute on the managed device that its agent monitors. Instances of the managed objects are simple current values of the parameters or attributes. For example, a list of currently active TCP circuits in a particular host computer is a managed object, an object instance is a single active TCP circuit in a particular host computer. Managed objects can be scalar (defining a single object instance) or tabular (defining multiple, related instances).

(4) Management information based (MIB) -- A MIB is a tree data structure collecting all possible managed objects at a managed node. Each managed node maintains a MIB that reflects the status of the managed resources at the node. Collections of related managed objects are defined in specific MIB modules. More details about MIB tree and manage group objects are included in section 2.4.

(5) Management stations (or called Managers) -- Management stations are devices running special management software application to monitor and control network elements. Usually, the management stations are dedicated powerful computers with fast CPUs, large memory and disk space. At least one management station must be present in each managed environment.
Many management applications have a graphical user interface to allow the network administrator to inspect the status of the network and take action when required. The management application contains the management functions, and has one or more processes to communicate with agents over the network. It performs tasks required by the human user to query the state of an agent’s local managed objects or change them if necessary, and then presents management information to the user through the user interface.

The management station interacts with agents using the SNMP protocol. The Internet network management protocol, SNMP, is implemented using the client/server model. The agent (server) has processes running on a managed node and continually listens for requests from a manager, and responds to requests for information from a manager machine. The manager is the client. Communication between a manager and an agent is via this request/response exchange. The manager can use a get-request to request status information of a managed node from its agent. The agent reads the values of objects in its MIB and replies with a get-response. The manager can also send a set-request to control resources by instructing the agent to modify values in its database that are provided by the manager. Each request causes the agent to generate exactly one corresponding response. However, sometimes events happen that are not planned. Managed nodes can crash and reboot, lines can go down and come back up, congestion can occur and so on. Each significant event is defined in a MIB module. When an agent notices that a significant event has occurred, it immediately reports the event to all management stations in its configuration list. This report is called a SNMP trap. The report usually just states that some event has occurred. It is up to management station to then issue queries to find out the details.
Since SNMP itself is the simple request/response protocol, Management stations can send multiple requests without receiving a response. Agents can be contacted by multiple managers. A manager program can send requests to one or more agents.

2.2 SNMP Protocol Operations

The SNMP protocol operations are defined as the interactions between the manager and managed devices. The following first four operations were defined originally in SNMPv1, the last two operations were added in the SNMPv2:

(1) Get -- Allows the management station to retrieve information from the agent about an attribute of a managed object.

(2) GetNext -- Allows the management station to retrieve the next object in the MIB directly from an agent. In SNMPv1, when a management station wants to retrieve all elements of a table from agent, it initiates a Get operation, followed by a series of GetNext operations.

(3) Set -- Allows the management station to set the value of an attribute of a managed object.

(4) Trap -- Used by the agent to asynchronously inform the manager of some event. Unlike the Get, GetNext, and Set operations, the trap does not elicit a response from the receiver. The SNMPv2 trap message is designed to replace the SNMPv1 trap message.

(5) Inform -- New for SNMPv2. Allows one management station to send trap type information to another management station and request a response.

(6) GetBulk -- New for SNMPv2. Allows the management station to retrieve efficiently large blocks of data such as a group of objects in the MIB without initiating repeated GetNext oper-
2.3 SNMP Data Representation

Most real networks are multivendor, with hosts from one or more manufactures, bridgers and routers from other companies, and printers from still other ones. Exchange of information in a managed network is potentially compromised by differences in the data-representation techniques used by the managed devices. To make multivendor device communication possible, it is essential that these objects be defined in a standard and vendor-neutral way. For this reason, a standard object definition language along with rules to encode messages for transfer over a network is needed. The one used by SNMP is taken from the OSI, called Abstract Syntax Notation One (ASN.1).

ASN.1 is the language used to define both the packet format, exchanged by the SNMP management protocol, and the objects that are to be managed in the MIB in a machine independent format. Objects in the MIB are encoded using the basic encoding rules (BER) associated with ASN.1.

2.1.4 MIB and Object ID

As with any network management system, the foundation is a database containing information about the resources to be managed. The network management information provided by agents to a management station is logically stored in the Management Information Base. A MIB is depicted as a tree, with individual managed objects as leaves. This is shown in Figure 2. Each tree node has a number assigned to it.
Figure 2  MIB Tree
For convenience, the objects of SNMPv2 are currently grouped by protocols (such as TCP, IP, UDP, SNMP etc) into ten categories. Table 1 illustrates the ten nodes under the current Internet-standard MIB-II. The MIB-II is formally defined in RFC 1213 and contains 175 objects. Only the system group objects are included in the MIB tree of Figure 2. The MIB tree is extensible at the experimental and private branches. Vendors are free to define and add additional objects for their products.

Table 1. The Object Groups of the Internet MIB-II

<table>
<thead>
<tr>
<th>Group</th>
<th># Objects</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>7</td>
<td>Name, location and description of the equipment</td>
</tr>
<tr>
<td>Interfaces</td>
<td>23</td>
<td>Network interfaces and their measured traffic</td>
</tr>
<tr>
<td>AT</td>
<td>3</td>
<td>Address translation</td>
</tr>
<tr>
<td>IP</td>
<td>42</td>
<td>IP packet statistics</td>
</tr>
<tr>
<td>ICMP</td>
<td>26</td>
<td>Statistics about ICMP messages received</td>
</tr>
<tr>
<td>TCP</td>
<td>19</td>
<td>TCP algorithms, parameters, and statistics</td>
</tr>
<tr>
<td>UDP</td>
<td>6</td>
<td>UDP traffic statistics</td>
</tr>
<tr>
<td>EGP</td>
<td>20</td>
<td>Exterior gateway protocol traffic statistics</td>
</tr>
<tr>
<td>Transmission</td>
<td>0</td>
<td>Reserved for media-specific MIBs</td>
</tr>
<tr>
<td>SNMP</td>
<td>29</td>
<td>SNMP traffic statistics</td>
</tr>
</tbody>
</table>

The MIB tree provides a map for locating information. With the number assigned at each tree node, object identifiers for managed objects can be derived. Object identifiers are just like telephone numbers, in that they are organized hierarchically with specific digits assigned by different organizations. They uniquely identify MIB objects in the tree. For example, the
object identifier for the object sysUpTime, which represents the time since the network management portion of the system was last reinitialized, is derived as follows:

iso org dod internet mgmt mib-2 system sysUpTime

1 3 6 1 2 1 1 3

The identifier would be written as .1.3.6.1.2.1.1.3

2.1.5 SMI Definitions

The Structure of Management Information (SMI) document, which is specified in RFC1155, defines the structure of the MIB and the rules for describing management information. The SMI is defined using ASN.1, allowing the use of standard ASN.1 data types: Integer, Octet, String, and Object identifier. It also defines the other data types such as network addresses, counters etc.

2.1.6 More Issues of SNMPv2

(1) Security

The most serious deficiency in SNMPv1 is its lack of authentication capability which makes it vulnerable to unauthorized changes to the configuration of a network. For this reason, many vendors have not implemented the set operations, thereby reducing SNMPv1 to a monitoring facility. SNMPv2 includes provisions for preventing the following types of security threats: masquerades, modification of information, message sequence and timing modifications etc. Changes have to be made in the message format of SNMPv2 to improve the security of message exchanges.
(2) Interoperability Issue

The changes to the message format and the addition of protocol operations and security features make SNMPv2 incompatible with SNMPv1. There are several ways to deal with this issue [6]:

a. Update the manager systems to support SNMPv2 in a way that allows coexistence of SNMPv2 managers, SNMPv2 agents, and SNMPv1 agents.

b. Achieve coexistence at the protocol level by reaching existing SNMPv1 agents through an SNMPv2 agent that acts as a proxy agent on behalf of an SNMPv2 manager. The proxy agent maps get-bulk PUDs (Protocol Data Unit) to get-next PUDs and SNMPv1 trap PDUs to SNMPv2 trap PDUs, but passes get, get-next, and set PDUs unchanged.

c. Use a bilingual SNMPv2 manager that supports both SNMPv1 and SNMPv2 to achieve coexistence at the protocol level.

(3) Distributed Management Architecture

SNMPv2 supports the centralized network management strategies of SNMPv1 as well as distributed strategies based on a new manager-to-manager MIB. In a distributed architecture, some systems operate both in the role of manager and of agent. When acting as an agent, a system accepts commands from a superior management system. These commands can deal with access to information stored locally or can require the system (now acting as an intermediate manager) to provide summary information about subordinate agents. In addition, an intermediate manager can issue trap information to a superior manager.
2.1.7 SNMP Proxy Agents for the Management of non-SNMP devices

The SNMP model assumes that each managed node is capable of running an SNMP agent internally. In the real world, there are resources that are either not managed by a computer process or managed using a non-SNMP protocol. Still many device manufactures either do not provide support for SNMP or provide limited SNMP support. To handle them, SNMP defined what is called a proxy agent, an agent that watches over one or more non-SNMP devices and communicates with the management station on their behalf, possibly communicating with the devices themselves using some nonstandard protocol.
3. **Web-Based Management**

Web-based management (WBM) is expected to merge Web functionalities with network management to provide administrators with capabilities beyond traditional tools. Since the data that is put on a Web server can be immediately accessible from everywhere on the network and a Web browser offers an ideal user-interface for management applications. Administrators can use a WBM application to monitor and control their networks with any web browser at any network node and take advantage of Web functionalities while their network is connected to the Internet. They no longer have to be tired to a dedicated management workstation to quickly, easily and effectively monitor and control network and individual components. This can also eliminate many interoperability issues that arise with multiplatform structures. [7]

WBM provides graphical interfaces that present information in a more visual and useful fashion than conventional, command-driven telnet screens. Browser operations and Web page interfaces are very familiar to today’s users of the World Wide Web. As a result, it reduces the costs of training administrators with a specific application’s platform or interface, and can enable a wider range of users to utilize network status information.

The WBM functionalities are realized using HTTP as the transfer protocol on the Web. Management information is transported between HTTP servers and HTTP clients. HTTP clients can be a Web browser or any other client-side application that makes use of the services offered by the HTTP server.

Numerous technologies with HTML, CGI or VRML are being deployed to realize WBM capabilities. HTML is used to lay out pages of Web-based information and provide links
called hyperlinks to other pages. HTML is generally textual and static, although graphics and active elements such as Java applets can be embedded within an HTML page. HTML is good at displaying tables of information such as network inventory details and IP listings.

The CGI (Common Gateway Interface) is a Web-based technique for accessing information in a database. A CGI script can be written to query a database and then format an HTML page to present this information to a client. When dealing with a web application, whenever the server gets a request that is handled by a CGI program, it must start the CGI program, which has a fixed amount of overhead. After the CGI program finishes processing a request, it terminates. For a new request, even same as before, a new program has to be started.

Comparatively, Java is perhaps the most powerful and has the best potential to implement Web-based management.

3.1 Java Web-Based Management

Java, introduced by Sun Microsystems, is gaining increasing acceptance in the industry. It is a simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded and dynamic language. [15]

Java is an interpreted programming language. Its source code is translated into Java bytecode for a Java Virtual Machine (JVM) using a Java compiler javac rather than into native machine code. When a Java program is executed, Java bytecode is interpreted by the interpreter JVM at run-time. Java is portable across platforms and operating system because Java bytecodes are platform-independent. Java program bytecode can be executed on any platform executing a JVM. The Java VM has already been ported to most common platforms. It is also
included in the recent versions of the Netscape Web browser as well as the Microsoft Internet Explorer, therefore it is available on almost every desktop PC as well as in major UNIX systems. Java applications can be executed by virtually everybody who uses the Web. Java stores objects as separate compiled class files that can be loaded over the network at run-time. It has great support for distributed application development.

Java can be used to develop stand-alone applications which are not Web accessible in the same way as C++ or other object-oriented language. But Java has built-in Web-enabled capabilities, that is, Java applets can be invoked in a Java-enabled Web browser. Figure 3 presents the principle of a Java applet.

![Figure 3 Downloading and Execution of a Java Applet](image)

When a Java applet is requested from a server, its Java bytecode is downloaded to the client machine. After the bytecode is transferred, it is first verified for security problems and then executed locally by the Web browser of the client machine using its Java Virtual Machine.
Technically, applets have browser-imposed security restrictions while Java applications do not. The restrictions do not allow applet access any local system resources such as memory or disk, and limit access to network resources. Therefore, applets can be securely passed and executed with minimal risk to a receiving client machine and without breaking network security. Currently, an applet in a Web browser can not communicate directly with any host on the network, except the Web server from which the applet is downloaded. These restrictions may be relaxed through more recently emerging techniques such as "signed applets".

Java applets are a powerful technique for managing and presenting the dynamic data needed for a WBM application. Java applets have the advantage that they can continue running when the web page is already loaded in the browser. This brings dynamic information to a Web browser allowing real-time information to be updated and presented from polling and traps. This is useful for such tasks as displaying real-time graphs of network activity or dynamically presenting device status. Applets can also add animation. By Contrast, other management tools offer Web access using passive HTML and VRML with CGI, through HTTP. This provides access to only static network information. Even though these tools are good at displaying tables, they do not allow administrators to easily view dynamic information or perform real-time management functions through a Web browser. A program using CGI has to be restarted in order to change or modify the data it displays, whereas a Java management application could be triggered by another user event. This is desirable from a user point of view.

Java’s Abstract Windowing Toolkit(AWT) provides many possibilities for graphical presentation. Java applets can also enrich HTML documents with more graphical capabilities. This is especially beneficial for the creation of user friendly interfaces to Web applications
including any management application.

Another notable capability of Java-enabled Web network management applications is their persistence. The idea is that the browser can go to another page, while an application continues doing its job.

All these attractive features make it valuable to develop cross-platform Web-based network management application. Java makes the web technology a natural focus for network management. One potential use for Java as a network management technology could lie in building applications that monitor Internet connectivity. According to SUN, Java can ultimately be used to monitor and configure Java-based clients such as its JavaStation over TCP/IP networks and the Internet.

SNMP support is not included in the Java Development Toolkit (JDK), but a complete Java classes package for SNMP is available from Advent Network Management, Inc. Applications can be developed using these classes.

3.2 Newly Emerging Approaches

Web-Based network management techniques are still evolving. There is a race among network management vendors based on two initiatives recently announced for Web-Based network management:

(1) Web-Based Enterprise Management (WBEM)

Many companies including Cisco, Intel and Microsoft are proposing a framework, called "Web-Based Enterprise Management" (WBEM). This initiative aims at the introduction
of new management protocols and a new hierarchy for integrated management of SNMP or other management agents. WBEM seeks to adopt a set of new standards, Hyper Media Management Protocol (HMMP), which is a communication protocol and a Hyper Media Management Schema (HMMS) which defines how data should be structured.[14] They are based on HTTP working with Web browser to simplify network management and reduce costs. WBEM does not attempt to replace existing management standards such as SNMP or CIMP, but to provide a framework embracing existing management standards and protocols. This would allow the integration of distributed management services provided by different management platforms and applications. The key purpose of the WBEM initiative is to consolidate and unify the data provided by existing management technologies.

(2) Java Management API (JMAPI)

Another initiative from vendors that holds a great deal of promise uses Sun’s Java Management API set. This is expected to be a new direction for creating powerful Web-based network management tools for the future. The JMAPI is a collection of Java language classes and interfaces that allow developers to use Web browsers as cross-platform network management interfaces and to build management applications. It reads and writes Java objects such as status information directly into a relational database. The JMAPI uses the Java’s distributed protocol, called Remote Method Invocation (RMI) as its remote communication mechanism. [13]

JMAPI is not based upon SNMP, although it supports SNMP and promises to make getting SNMP information easier. Unlike the actual protocol standard SNMP, which deals directly with communicating network hardware and software in a uniform way, JMAPI only
focuses on a software architecture for management tools. It is concerned only with creating a common interface for accessing network information through existing protocols such as SNMP. The basic components for implementing an application using JMAPI are:

a. A Java-enabled browser supporting Java JDK and RMI

b. The JMAPI objects

c. A commercial relational database such as Oracle, Sybase, or Informix integrated with Java Database Connectivity (JDBC)

d. A Web server to coordinate distributed objects

To be noted is that the Java/RMI based JMAPI is not the same as the definition of Java Web-based management stated above. HTTP is only used here to initially transfer Java code from the server to the client and, for all subsequent management communication, RMI is used.

Both JMAPI specification and WBEM initiative are still in their initial stages of development. No WBEM products have been shipped yet, though a software developer kit was scheduled for release at the end of 1997. The immature JMAPI today is still in a beta test state of version 1.0 and has not yet been released.

Currently, there are no official standards for developing Web-based management applications, but using Java with a Web browser still can be a way to make the job of managing network much easier and more efficient.
4 Conventional SNMP Management vs Web-Based Management

Even while the introduction of standards and proposals for the Web-Based management is still at an early stage, its benefits are becoming more and more obvious when compared with the conventional way. In this section, the conventional SNMP network management approach using management stations is compared with the Web-based approach in order to understand the value of Web-based management.

Before Web-Based Management, the conventional approaches have several difficulties and challenges:

(1) Incompatibility and platform-dependence in managing heterogeneous networks

Various network system management applications have created incompatible infrastructures. Their platform-dependence and complexities make the training of system administrators in network management very costly. Usually, system administrators are required to deal with multiple technologies.

Multiplatforms make applications development more difficult and time-consuming for software developers. They must create applications for multiple management environments, and because they cannot easily integrate their solutions, their ability to provide innovative solutions is restrained.

(2) High investment

Management stations are usually dedicated, powerful, costly and specialized computers. As a result, organizations have to invest in a costly hardware or management
platform. To be effective, administrators are required to have extensive training in the use of each specialized management tool which can also cost both money and time. In addition, the management is performed centrally and does not take advantage of the powerful technology they are designed to manage.

Comparatively, Web-Based management approaches provide the following advantages:

(1) Lower cost

A management system based on a popular web browser can ease access to management data for networks of any platforms and will be less costly to learn to use, set up and operate. This can significantly lower the cost for training administration to use a new management system as well as lowering the cost associated with new equipment.

(2) Scalability

With a simple Web browser as the management interface, organizations can take advantage of networking technology they already have in place to manage a wide range of network resources such as routers, hubs, PC, workstations, etc. When the same technologies used for building networks are used to create management applications, the scalability of the applications can match that of the networks. This scalability allows a system administrator to learn or implement just one interface to monitor and maintain various growing heterogeneous system environments.

(3) Platform-independence

WBM can eliminate the need to design different version applications for different management platforms and make applications more efficient and effective. Developers can be concentrated on innovative functionality rather than system differences and they can bring the
applications into use or to market more quickly.

The major benefit for users will be significant: a greater selection of management applications and added functionality that takes advantage of the rapidly evolving web technology.

(4) More flexibility and remote access

The Web based approach will give system administrators a more flexible way of managing networks than they currently have. They can monitor and configure devices with remote access capabilities from any web-enabled PC, workstation or even notebook computer in the network at any time, regardless of its location. This reduces the time and effort required to do the job.

(5) Easier dynamic information

The Web based approach provides access to dynamic management information more efficiently without the overhead of restarting application program.
5. Writing Java Web-Based Network Management Applications for the World Wide Web

Writing a Web-Based management application with Java from scratch is very time consuming because of the large amount of work need on writing SNMP variables, communication classes, MIB classes etc. The Java SNMP classes package by the Advent NM, Inc. provides a means to write a Java application or applet that can communicate with existing network management SNMP agents by assembling the pre-built components. While it is possible to write an agent with this package, users would need to add native methods to get access to the underlying system resources in order to build a complete agent.

5.1 Technical Overview of the Advent SNMP Java Classes Package

The package is designed to enable one to write network management Java applets or Java applications that use SNMP to communicate with managed nodes. Special support for applets is provided to get around the security restrictions. The package contains classes of four categories:

(1) SNMP variable classes

The ancestor of all SNMP variable classes is an abstract classes called SnmpVar. This class contains abstract methods for printing, ASN encoding, ASN decoding, etc.

(2) SNMP command classes

The Advent SNMP package uses the SnmpAPI classes to manage sessions created by
the user application, manage the MIB modules that have been loaded, and store some key parameters for SNMP communication, e.g. SNMP ports to be used. An SNMP application (manager or agent) often needs to manage multiple sessions to interact with multiple SNMP peers. The SnmpAPI class has a list of sessions attached to it and monitors each of the sessions for time-outs and retransmits via a separate thread. It enables a few methods across all sessions, e.g., checking if the responses have come in on any of the sessions, etc. Multiple threads can work with a single SnmpAPI instance. The SnmpAPI class must be instantiated and started before its usage.

The SnmpSession class is used to manage a session with an SNMP peer. More than one host can be accessed via a single session, but the Advent NM recommends separate sessions for hosts that are often accessed inside an application. Each session runs as a separate thread to receive tasks and provides functions to:

a. Open sessions (on a particular local port if needed)

b. Synchronously or asynchronously send and receive SNMP requests

c. Check for responses and time-outs

d. Close sessions

An SnmpSession needs to be instantiated and opened before it can be used to communicate with an SNMP peer.

Interaction between the SNMP manager and the agent is done via the SNMP protocol data units(PDU). The SnmpPDU class will be used to provide the variables and methods to create and use the SNMP PDU.
(3) SNMP MIB related classes

MIB modules allow an SNMP managed agent to let users know about the structure and format of data available on the agent. The MIB modules are usually specified in a MIB module file, which needs to be parsed to understand the syntax and structure of the data available on the agent.

The `MibModule` class provides a means to parse and use the data available in an MIB module file. Each MibModule instance is created from an MIB module file, and users can load and unload MIB modules by creating and deleting these instances. The instance contains all the nodes of the MIB tree as well as defined traps and textual conventions. A few utility methods and variables are provided, e.g, `getNode()` to search the modules for a node matching a specified Object ID.

(4) Miscellaneous classes

They are the other classes that do not fall into the above categories i.e. the client interface `SnmpClient` class, the Exception classes, `MibException` and `SnmpException`.

In order to get around the security restriction for Java applets, a Java program for the web server called SNMP Applet Server(SAS) is provided in the package, which allows the applet to send and receive SNMP packets to and from any managed devices from the Web server(applet host). The SAS has to be run on the Web server. It suffices to start the SAS server on the Web server once. The applet then automatically detects the TCP port that is used on the Web server and all further SNMP communication between the applet and agents will be relayed via the SAS. That is, the applet uses the Advent library to communicate with the SAS application on the web server, and the SAS will communicate with the agent to do all communica-
tion the applet wants. This is possible, since Java applications do not have to deal with security
problems. The use of the SAS server is transparent to users.

5.2 A Java SNMP Application

By re-using the basic classes files in the SNMP packages, a Java application implementing the
GetBulk SNMP operation was written and the code is included in the Appendix. A testing output
from executing the code is also included. The application shows that a user can develop an SNMP
management application quickly by taking advantage of the reusability feature of the Java SNMP
classes package.

5.3 A Simple Java Applet Network Monitoring Tool

In this section, the Java SNMP network monitoring applet is discussed. The main purpose of
the monitoring tool is to examine the feasibility and methodology of Java Web-based Management
and to show the benefits of a Java Web-based management tool.

5.3.1 The Need for Network Monitoring

The rapid growth of the Internet and the growing demand of many network applications such
as multimedia applications has drastically increased the traffic load of network. The amount of data
transferred through a network and passed to an application on a network node is growing. Observing
or gathering information about the status or behavior of data being transferred and passed to any end
system or subnetwork is one fundamental task of network management. The information can be
further analyzed so that it will be helpful for network configuration and planning. Among the many
aspects of a network that is monitored, the status of the IP packages delivered to a node is one aspect
that to be observed. This can help a system administrator to locate a machine that transmits excessive data or a host that is not able to communicate with others so that an action can be taken to solve the fault or control the congestion.

The Java applet discussed here can be used to simplify the monitoring of IP packages delivered to the higher protocol layer at any managed host or hub running a SNMP agent. An MIB browser by the Advent NM is used in the applet, which contains the Get, GetNext, Set SNMP operations to specify the managed node and the managed objects to be monitored.

5.3.2 The Environment of the Applet

Figure 4 depicts the complete environment in which the applet is running. First, the applet is loaded from the Web server into the client machine with the Web browser. It then can communicate with the managed node, which has an SNMP agent, via the SNMP applet server application. The applet server is running on the Web server machine and relays all SNMP messages between the applet and the agent.

The Web browser, SAS server and SNMP agent are actually constructing a 3-tier model. The Web browser is the first tier, providing graphical interface to users to access management functions. The SAS server is the second tier which provides the main management application logic and the data needed by the Java management client. Finally, the SNMP agent is the third tier, which provides access to management information from the managed device.
The Advent NM provides a visual network management applet builder (Advent NetMonitor). The author downloaded a 30 day trial version and used it to build the applet. It was convenient to use the tool to visually build the SNMP capable Java applet to monitor and control network devices that support SNMP. Users of the Advent NetMonitor can build their own applets by using any one of the 25 pre-provided components and editing property form for each component. Users can also create and add their own components with some additional Java programming. They can use these components in their applets, or they can add more customization. All of the details of communication between the created applet and SNMP agents can be handled transparently to the users.

Figure 4. The Environment for SNMP Applet

The Advent NM provides a visual network management applet builder (Advent NetMonitor).
5.3.3 Prototype of the Applet Network Monitoring

The prototype of monitoring applet consists of:

a. A server machine, on which the Web server and SAS server will be running. The SNMP classes and applet will also be placed on this machine from which they can be loaded through the network to any client machine. For testing, a Window 95 PC (hostname: sb1pc14.cse.eng.auburn.edu, IP address: 131.204.20.163) was selected as the server machine.

b. A Java JDK on the Web server machine. The Java JDK 1.1.4 has been installed on the server machine sb1pc14.cse.eng.auburn.edu.

c. A Web server on the server machine. A trial version of Java Web Server 1.0.3 was downloaded and started on the server machine sb1pc14.cse.eng.auburn.edu. The default port number used by this Web server is 8080. It can be configured to port number 80.

d. A SNMP Applet Server (SAS). A SAS server was downloaded from the Advent Web site. It should be started and run on the Web server machine to provide a pass-through for the MIB browser applet to talk with the managed device.

e. A SNMP agent running on the host or device to be monitored.

Since the SNMP agent is operating system dependent, to perform any SNMP operation on a managed node, the node should be capable of running an SNMP agent or else a proxy agent is needed. It is the vendor of a network device who provides information on how to get an agent or provides an agent itself. A Windows 95 SNMP agent is an agent the author could find and it was installed. The same Windows 95 PC (Domain name sb1pc14.cse.eng.auburn.edu, IP address: 131.204.20.163) is also selected as the managed node. The monitored node can be any other device as long as an essential SNMP agent is running on it. Due to the limitation of machines available, the server machine was also used as the monitored machine. Machines or devices in our Engineering Network may
have similar SNMP agents available or running, but the root privilege is needed to get information and access.

f. The SNMP classes and applet loaded to the client machine

g. A Java-enabled Web browser on the client machine. In order to use the applet to perform Web-based monitor, a Java enabled Web browser is necessary.

5.3.4 The Use of Applet Network Monitoring Tool

The following steps were performed to test the applet monitoring tool:

a. Start the Window 95 SNMP agent on the computer

b. Start the Web server on the server machine

c. Start the SAS server on the server machine where the Web server is running

d. Configure the Web server with the administration tool by creating file alias to make the URL "http://sb1pc14.cse.eng.auburn.edu:8080/NetMonitor/" point to the actual directory on the C: driver where the applet is located.

e. For a user, click on the "new_applet.html" file, the applet will be downloaded and transferred to the user by the Web server. After necessary verification, the applet is viewed by the user’s Web browser. To specify the managed node, the user just clicks on the applet and fill out the property form with the hostname of managed node and the OID of the attribute to be monitored. The applet then shows the real-time information of the object IpinDelivers in the IP group of MIB tree which represents the total number of input datagrams successfully delivered to the managed node through IP protocol. Its OID can be derived from MIB tree structure: .1.3.6.1.2.1.4.9

The following three figures illustrate the interface that is available to the user. In the first two
figures, the horizontal axis reflects the real system time and the vertical axis reflects the number of IP packages successfully delivered to the managed node. The first one shows the number of IP packages successfully delivered to the managed node from time 18:04:49 to 18:12:49. The second figure shows the number of IP packages delivered continuously in the period of time from 18:14:49 to 18:22:49. They illustrate that the applet is able to show dynamic information in real-time. The third figure shows the interface of the Advent MIB browser, which is used to specify the managed node and the managed object to be monitored, and the MIB module file to be loaded and parsed.
Figure 5  Snapshot of the Applet Monitoring Tool
Figure 6  Snapshot of the Applet Monitoring Tool
Figure 7: Snapshot of the Applet Monitoring Tool