INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt., **8**, 104–119 (1998)

SNMP through WWW

In this article we propose a bilingual agent to accept either SNMP or HTTP commands and design several HTML pages to facilitate the task of network management. For network elements that support only SNMP, the bilingual agent can act as a proxy, so that the traditional SNMP agent can also be queried through the Web browser. © 1998 by John Wiley & Sons, Ltd.

*By Ching-Wun Tsai and Ruay-Shiung Chang**

Introduction

The task of managing the wide varieties of networking software and hardware is becoming increasingly difficult due to the rapid progress of networking technologies. Traditionally, to ensure a reliable data transportation service, a dedicated management console is utilized to monitor and analyze traffic status on a segment of the network. In a large organization that often employs various networking technologies for the enterprise-wide network, the traditional network management solution becomes too costly. It is necessary to choose an appropriate network management standard for managing devices efficiently.

Chin-Wun Wang is currently a lecturer in the National Hu-Wei Institute of Technology, Taiwan. His research interests include network management and Internet applications.
Ruay-Shiung Chang joined the Department of Information Management, National Taiwan Institute of Technology in 1992 where he is currently a professor. His research interests include Internet, high-speed networks, and network management.

Contract grant sponsor: NSC; contract grant numbers: NSC86-2622-E-007-001 and NSC86-2213E011-056

**Correspondence to: Professor Ruay-Shiung Chang, Department of Information Management, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, Taiwan. Email: rschang@cs.ntust.edu.tw*

The Simple Network Management Protocol (SNMP)[6,10] is a *de facto* network management standard. Due to the simplicity of its architecture, almost every network device supports SNMP. An agent program embedded in a network device collects traffic statistics and records them in a management information base (MIB). A network manager can obtain the information by polling the agent in real-time.

However, recently the growth of the World Wide Web (WWW) has changed the network management world. Network vendors or groups[21,22] have developed many schemes for Web-based management, which is the application of WWW tools for the management of systems and networks. This includes using Web servers and Web browsers for providing static, dynamic and interactive content of management information. A Web server acting in a management role can provide information in a variety of forms including Hyper Text Markup Language (HTML), graphics, executable code (JAVA applets) and binary encoded information. Together, these capabilities allow HTTP (Hypertext Transfer Protocol) to function as a possible protocol for the management of systems and networks.

There are three advantages for Web-based network management. First, there is no specialized management software needed to configure or monitor a device. To utilize the Web-based network management, one only has to run a Web browser on the client side. Then input the proper URL (uniform

CCC 1055–7148/98/020104–16$17.50

resource location) and the network management information will be shown. Second, the version problems that typically occur when an older agent or manager doesn't support the new, and possibly required, features of the other are eliminated. These problems occur either when a new MIB is supported at the agent and not supported at the manager, or when a new version or feature of the manager expects a minimum of MIB support from an older agent. By using a Web-based network management, both agent and manager do not need to be updated simultaneously. The Web-based agent can update the MIBs and the manager can still work without any change. This same problem also occurs in a multiple vendor environment. Many vendors supply devices embedding some level of vendor-specific MIBs. This complicates the job of the management station vendor by requiring them to support additional screens for these vendor-specific MIBs (MIB browsers are simply inadequate for this task). Often this leaves the network manager with no choice but to buy a specific management application for each vendor's device and then learn to use all of them. By delivering the management information via a Web-based agent, the application can use the common familiar interface of a Web browser and the browser exists for all platforms. The next major advantage is that of platform- and location-independence for the application. A network manager can access a Web-controlled network from anywhere, on any platform. All that is needed is a general-purpose Web browser, which is standard software for the vast majority of platforms.

To realize the above advantages, we propose a Bilingual Agent to accept either SNMP or HTTP commands and design several HTML pages to facilitate the task of network management. In order to use a Web browser to access information in devices supporting only SNMP, it is necessary for the Bilingual Agent to act as a proxy between an SNMP agent and a Web server. SNMP operations over HTTP could take advantage of the protocol's ability to transfer large amounts of information and its security features. This means that one HTTP request may contain multiple SNMP requests. The Bilingual Agent will improve performance and reduce traffic for SNMP operations.

The rest of this paper is organized as follows. The next section reviews the previous Web-related applications. The third section describes our proposed system. The fourth section gives the implementation details. Conclusions and future research directions are presented in the final section.

Related Work

Web-based network management through WWW is one kind of WWW access to legacy client/server applications.[5] The related topics have recently been an area of intense research activity.[1,3,11,13,20]

Brooks *et al.*[3] have developed HTTP Stream Transducers. These services, modeled as HTTP Proxy servers, allow the data stream from a Web site to be modified for per-user markup. For example, when a user visits a Web site they have already seen, the Stream Transducer can add HTML to the page to tell the user the date on which they last viewed this document. Stream Transducers can be hooked together in sequence between the user's Web browser and a Web site in order to provide aggregate functionality.

Lotus' InterNotes'[20] product uses CGI mechanisms to allow Web browser access to documents and forms managed by the Notes Server. Documents to be placed on the Web are pretranslated by a program that converts them to HTML. These documents and forms are accessed through a standard Web server as though they were normal HTML documents. In the case of Notes Forms, the Submit button sends the contents of the form to a Lotus-supplied CGI program that incorporates the data back into the Notes database. While this does allow for some Web-based use of their system, the interaction model is limited. Web-based users do not have access to the integrated e-mail and applications which standard Notes clients use.

Barta and Hauswirth[1] describe a toolkit for the creation of 'Interface-Parasite' gateways. These allow a synchronous session-oriented tool, a telnet session for example, to be used via a Web browser. Their toolkit requires no changes to the source code of the Web server application, but it cannot handle an application in which the client can perform actions independently of the Web server.

Ockerbloom[11] proposes an alternative to MIME (Multipurpose Internet Mail Extensions[2]) types, called Typed Object Model (TOM), that could be employed instead of a MIME extension to allow the use of external tools in the client. Object types exported from anywhere on the Internet can be registered in 'type oracles', specialized Web serves that can communicate among themselves to uncover the definitions of types registered elsewhere. Web clients who find a type they do not understand can ask one of the type oracles how to convert it into a known supertype. In this way, Web clients would not have to be set up to handle a new MIME type. They could simply query the type oracle, which could return information on how to run the external tools.

Trevor *et al.*[13] propose a component-based approach to enable applications for access from the WWW in a more flexible manner by using an application integration toolkit. This facilitates more modular and 'lightweight' application deployment where specific modules can be integrated as required. Further, separation of the protocol-specific communication-handling components from the application modules and provision of an interest-based integration service allows enabling of applications for a variety of other access methods such as electronic mail with few or no changes to the applications themselves.

For integrating network management application into WWW, there are some studies trying to provide Web-based network management.

In July 1996, five major vendors announced an initiative to define *de facto* standards for Web-based Enterprise Management (WBEM).[23] Over 50 vendors publicly endorsed this effort, spearheaded by Microsoft, Compaq, Cisco, BMC, and Intel. The initial announcement called for defining the following components:

(1) HyperMedia Management Schema (HMMS)—an extensible data description for representing the managed environment that is to be further defined by the Desktop Management Task Force (DMTF).
(2) HyperMedia Object Manager (HMOM)—a data model consolidating management data from different sources—a C++ reference implementation and spec, defined by Microsoft and Compaq, to be placed in the public domain.
(3) HyperMedia Management Protocol (HMMP)—a communication protocol embodying HMMS, running over HTTP and with planned interfaces to SNMP and DMI (Desktop Management Information).

Sun[16] has also announced a programming environment for developing Web-based network and systems management software. This environment, called Solstice Workshop, consists of a Java Management API (JMAPI), a small footprint database, and a Java programming environment. Solstice Workshop's big drawing card is JMAPI's extensibility and the popularity of Java's 'write once run anywhere' appeal. JMAPI requires Java, whereas the HMMP/HMMS/HMOM proposal specifies HTML/HTTP, although Java is not specifically excluded.

Prior to these announcements, three developers from Hewlett-Packard (Harrison, Mellquist, and Pell) wrote an Internet draft proposing the use of port 280 for exchanging HTTP management data.[9] This Internet draft describes a very lightweight HTTP Manageable MIB as well as a tunneling facility for SNMP over HTTP.

Of these three proposals, the WBEM is certainly the broadest in scope, addressing not only protocol issues but also data modeling and extensible data description facilities. While JMAPI includes object class definitions, it does not go as far as HMMS with respect to data modeling. The Harrison *et al.* proposal is extremely focused and does not attempt to define new protocols, data models or data description facilities. WBEM is not entirely the same thing as JMAPI since it focuses on new protocols for managing the network and a new hierarchy. JMAPI focuses only on software architecture for management tools and uses existing protocols such as SNMP.

Advent Network Management Inc.[18] proposes 'Advent Network Management Java SNMP Package'. It constructs SNMP PDU using Java. In order to avoid the Java applet security problem, it sets up a proxy server in the Web server to forward the PDU to the managed device. The proxy server is also written in Java. In this architecture, the server and browser must all be Java-capable machine. Web-based Java Remote Network Monitoring[15] is the product for Java SNMP Package. Java applet can provide dynamic information and enrich the HTML document. It gives life to WWW. Java applet is so powerful that the Web browser vendors have to limit its capabilities. Most Web browser providers do not allow the Java applet to access the local files and communicate with any device except the Web Server. In other words, the Java-coded Web-based management monitors a managed device only where the Web Server is located and does not store any information in the local file. The response time will be longer because the Java applet must consist of the additional SNMP class which is to process the SNMP PDU (Packet Data Unit). The SNMP Java applet code is big and if everyone uses the Java-Code SNMP management to monitor the device, the overall network traffic will be heavy.

The DR-Web Extensible Agent[19] is an enhancement of SNMP Research's popular EMANATE Master Agent/Subagent system. With DR-Web, SNMP Research has extended the EMANATE Master Agent to support an HTTP interface as well as the existing SNMP interface. The Dr-Web Extensible Agent converts HTTP requests and SNMP requests to EMANATE events, dispatching them to the appropriate Subagents. This system is so large that it is not suitable for general network devices.

Deri[4] has constructed a proxy server for network management. The Web browser at the client side must set up a proxy server which intercepts the request from the manager, decodes and checks the Uniform Resource Location (URL). If the URL consists of network management operations, the proxy server performs these and sends the response to the manager. If not, the proxy server redirects the request. However, there is a 'far–near' problem. If an SNMP manager tries to get an MIB variable, where the agent is closer to the manager than the proxy server, the network bandwidth would be wasted and performance would be poor.

In reference 17 using the Common Gateway Interface (CGI) to implement network management operations is proposed. The Web server plays the proxy role to perform these. In other words, the Web server receives the HTTP request from the manager that consists of the network management operations. The related CGI program performs the operations as soon as it receives the request, and the Web server then sends the response back to the manager. There are several drawbacks to this scheme. The first is that the presentation of network management information is limited by HTML (Hypertext Markup Language). Second, the Web server's load is high. Each time the Web server receives a request to run a CGI program, a new process must be started on the Web server machine. With more than a few clients accessing data simultaneously, a considerable load is placed on the Web server. The last drawback is its poor response time. Because the Web server machine has to start a new process for each CGI request, Web clients may have to wait for a long time for their requests.

Ming-Jeng Lee *et al.*[10] propose a browser- oriented remote management system. Because these data structures of SNMP and CMIP are designed for programs to process, they call the services provided by CMIP and SNMP API-level services. As a result of using a protocol providing an API-level service, agent-specific manager applications are needed to decode and reorganize the management information properly before it is displayed. To overcome this shortcoming, they construct a network management protocol providing browser-level service, which is called the Remote Management Protocol (REMP), and a language for describing management information, called Management Interface Language (MIL). An agent provides management information to managers in a way similar to a device's showing control panels to users. In fact, the browser will display each managed object as a control panel on GUI. The management information is specified according to the Management Interface Language (MIL). There are some disadvantages in this scheme:

(1) It is platform- and location-dependent. Because the existing browser software is not used in this scheme, platform and location will limit the new browser software.
(2) This scheme does not adapt to the existing network management environment, because a new agent must be installed in advance.

In this article we will develop an elegant Bilingual Agent which supports both the traditional SNMP-based network management and the new HTTP-based network management. Table 1 summarizes the many Web-related network management systems.

The Proposed System

—Architecture—

Figure 1 shows the proposed architectural framework for delivering Web-based network management solutions. The framework features three distinct elements: the almost universal Web browser itself; Web-enabled management applications; and devices that run the Bilingual Agent. The manager may be the Web browser that runs at any machine. The Bilingual Agent running on the workstation plays the role of a managed device or a proxy. The

Characteristics \ Approaches		Bilingual Agent	Java-based[15,16,18]	WBEM[23]	Dr-Web[19]
New browser				✓	
New agent		✓	*	✓	✓
CGI		✓			
Proxyserver		✓	✓		
Transport Protocol	SNMP	✓	✓		
	HTTP	✓			✓
	HMMP			✓	
New data model				✓	
Support for old SNMP manager		✓			✓

*Due to security considerations, Java-based network management only communicates with the device where the Web server is located. If the managed device is not the Web server, a proxy will be installed in the device where the Web server is located.

Table 1. Characteristics of current Web-related network management systems

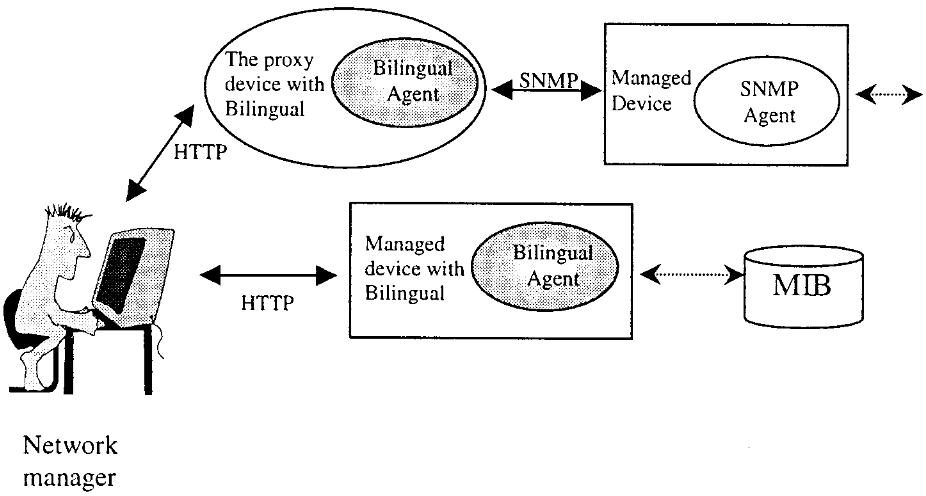

Figure 1. The proposed Web-based network management architecture.

manager sends the HTTP requests with embedded SNMP request information to the agent. The agent receives the requests and sends the HTTP response back after proper processing. The HTTP response contains one or more SNMP responses. The manager shows the result after receiving the HTTP response.

The proposed Web-based network management is focused on compatibility with current Internet protocols and does not attempt to define new protocols, data models or data description facilities. HTTP is the current transport protocol in the WWW. The proposed architecture still adopts HTTP as the transport protocol. The manager is a Web browser (like Netscape Navigator or Internet Explorer). Web browsers are ubiquitous, easy to use, and much less expensive than special-purpose management applications.

The Bilingual Agent is the core of the proposed Web-based network management architecture and plays the role of the agent and the proxy in it. It accepts the command through HTTP and retrieves the value from MIB, then returns the results. In order to comply with current network management environments, the Bilingual Agent also plays the role of the proxy between the Web browser and the existing SNMP agent devices. The Bilingual Agent acting as the proxy sends the SNMP request after receiving the HTTP request which the embedded SNMP requests. It will return the HTTP response after receiving the SNMP response from the managed device.

The Bilingual Agent has four main modules and Figure 2 shows the overall architecture. The HTTP Interface module processes the HTTP request from the Web browser. This module not only plays the role of the general Web Server but also deals with network management operations from the Web browser. The function of the SNMP Interface module is similar to that of the traditional SNMP Agent. In order to query the traditional SNMP agent through the Web browser, a proxy will be added to the Bilingual Agent. The proxy is an HTTP/SNMP conversion infrastructure. The proxy receives the HTTP request and then sends the SNMP request to the managed device. Then the proxy waits for the SNMP response and sends the HTTP response to the Web browser. The proxy is implemented by a Common Gateway Interface (CGI).[14] The proxy module is invoked by the HTTP Interface module and executes the SNMP operations. The MIB module stores the network management information and provides the interface for retrieving the data.

Figure 2 also shows the internal control flow of the Bilingual Agent. In the following, we describe the four possible ways of how the Bilingual Agent works. Later sections will show the HTTP messages for SNMP through WWW.

The control flow (a1, a2, a3, a4) shows the traditional Web Server retrieving the normal HTML document. The HTTP interface module receives the HTTP request from the browser and verifies that the requested object is an HTML document. The HTTP interface module retrieves the document and sends the response back to the browser.

The control flow (c1, c2, c3, c4) shows a manager querying the Bilingual Agent. The manager sends the request from the Web browser. The request contains the network management operation command. The request format will be discussed in a later section. The HTTP Interface module receives the request, parses it and obtains the operational parameters. The MIB module then retrieves the variable values and returns the result to the HTTP Interface module. The HTTP Interface module sends the response back to the manager (Web browser).

The control flow (b1, b2, b3, b4, b5, b6) shows a manager querying the traditional SNMP Agent through a browser. The HTTP Interface module receives the request, parses it and invokes the Proxy module. The Proxy module sends the SNMP PDU to the managed device and waits until receiving the response or time-out. The HTTP Interface module then sends the response to the manager after receiving the result from the Proxy module.

The control flow (d1, d2, d3, d4) shows the Bilingual Agent being queried by traditional SNMP commands. The manager sends the request to the Bilingual Agent. The SNMP Interface module

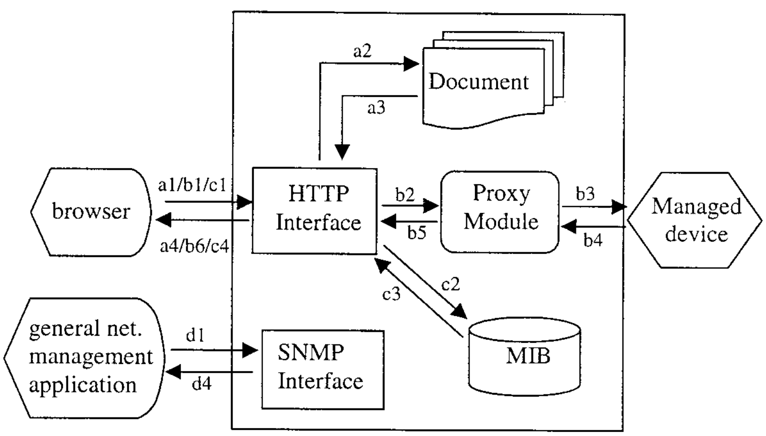

Figure 2. The overall architecture of Bilingual Agent.

invokes the MIB module after processing the SNMP PDU. The MIB module returns the variable values to the SNMP Interface according to the object identifier contained in the SNMP PDU. The manager will receive the response after the SNMP Interface module sends back the response PDU.

—URL Convention for SNMP through WWW—

In order to manage network resources using HTTP, a mapping between management operations and URLs has to be defined. The proposed SNMP-URL is compliant with the standard URL syntax and therefore it can be handled by conventional Web servers and Web browsers. The SNMP-URL is composed of three elements, http:/ /<host>/<cooperation>?<parameters>, where:

(1) <host> identifies the host where the Bilingual Agent runs
(2) <operation> specifies the protocol operation
(3) <parameters> contain the operation parameters.

There are three operations for the SNMP-URL and these are shown in Table 2. The parameters contain four keywords to identify the individual parameter, shown in Table 3.

Since the SNMP-URL is supposed to be used not only by software applications (for instance CGI scripts, JAVA Applet) but also by human operations from within their Web browsers, it is very important to have a clean and simple syntax. Binary values are not allowed since humans cannot handle them. The object identifiers contained inside the SNMP-URL can be in both numeric and symbolic form. SNMP

	Operations	Meanings
1	Snmpget	To retrieve a variable value like GET
2	Snmpgetn	To retrieve a variable value like GETNEXT
3	Snmpset	To set a variable like SET

Table 2. The operations list for SNMP-URL

	Keywords	Meanings
1	host	To specify the managed device; if the managed device is the Bilingual Agent itself, this parameter can be omitted
2	community	To specify the community name
3	oid	To specify the object identifier
4	value	To specify the setting value, if the operation is set

Table 3. The parameters description list for SNMP-URL

uses abstract syntax notation 1 (ASN.1) to describe the types of data. Table 4 is an example of some data types.

The following examples show how to use the SNMP-URL. Suppose the Bilingual Agent is running on host hercule.cs.ntit.edu.tw:

(1) Get the value of .1.3.6.1.2.1.1.1.0 (sysDescr.0)
http://hercule.cs.ntit.edu.tw/snmpget?oid=.1.3.6.1.2.1.1.1.0&community=public
(2) Set the .1.3.6.1.2.1.1.6.0 (sysLocation) to ‘laboratory’
http://hercule.cs.ntit.edu.tw/snmpset?oid=1.1.3.6.1.2.1.1.1.0&community=public&value=laboratory
(3) Get the value of .1.3.6.1.2.1.1.1.0 (sysDescr.0)

ANS. 1	Example String
INTEGER	‘1996’
OCTET STRING	‘hercule’
OBJECT	‘1.3.6.1.2.1.1.1.0’
IDENTIFIER	‘140.118.9.68’
IpAddress	‘2342343342’
Counter32	‘2343223233’
Gauge32	‘2343223432’
TimeTicks	‘<BER-
Opaque	encoded data>’

Table 4. The ASN.1 data types

from another device (IP = 140.118.9.87) http://hercule.cs.ntit.edu.tw/snmpget?oid=.1.3.6.1.2.1.1.1.0&community=public&host=140.118.9.87

(4) Set the .1.3.6.1.2.1.1.6.0 (sysLocation) to 'laboratory' for another device (IP = 140.118.9.87) http://hercule.cs.ntit.edu.tw/snmpset?oid=.1.3.6.1.2.1.1.1.0&community=public&host=140.118.9.87&value=laboratory

—SNMP Requests over HTTP—

Once the SNMP-URL is defined, it is necessary to specify the format of the information returned by the HTTP protocol. Suppose the following URL is specified:

http://hercule.cs.ntit.edu.tw/snmpget?oid=.1.3.6.1.2.1.1.1.0&community=public

An HTTP client will send the following data to the Web server running on hercule.cs.ntit.edu.tw:

GET snmpget?community=public&oid=.1.3.6.1.2.1.1.1.0 HTTP/1.0
[empty line]

The HTTP response, returned by the HTTP server, is always positive, unless the requested URL cannot be found or if some other problem arises (for instance, authentication). If the HTTP response is positive, it will contain the SNMP response which can be either positive or negative. If the SNMP response is positive, the Status-Line does not change and the Entity-Body will contain a set of pairs (<identifier>, <value>) separated with carriage return. Identifiers are object identifiers, usually in symbolic form, whereas values are strings encoded using the encode scheme used by the HTTP protocol. This example has only one SNMP response embedded in a HTTP response. If the response contains multiple SNMP responses, it will insert one empty line between two SNMP responses. Therefore, the HTTP response can contain one or multiple SNMP responses. If the SNMP response is negative, the extension Status-Code will be added. (Refer to HTTP/1.1[7] for the Status-Code used). Table 5 shows the extension Status-Code for SNMP. The Entity-Body will show the description for the Status Code.

The Bilingual Agent running on hercule.cs.ntit.edu.tw sends the following data to the Web browser after receiving the request in the above example:

HTTP/1.0 200 OK
Date: Tue. 25 Mar 1997 10:30:16 GMT
Content-type: text/html
Content-length: 28
[empty line]
system.sysDescr.0
Unix 4.3BSD

Suppose the Bilingual Agent does not implement the object identifier. The Bilingual Agent sends the following data after receiving the HTTP request in the above example:

HTTP/1.0 424 NO SUCH OBJECT
<empty line>

The server does not implement this object identifier.

Web Page Implementation

To demonstrate the Bilingual Agent, several homepages and functionalities have been designed. The following is a brief description (visit http://hercule.cs.ntit.edu.tw for testing).

—Smart-Web View—

We will implement a Web-based network management system which contains both the agent and the manager. For the agent side, the Bilingual Agent is written in C based on the CERN World-Wide Web Library.[8] For the manager side, Smart-Web View is proposed. Smart-Web View is an HTML document that contains four network management tools and related network resources links. Figure 3 shows the displayed homepage of Smart-Web View.

Figure 4 shows the organization structure of Smart-Web View. The network management tools have four functions: Simple-Query, MIB-Browser, Discovery and Trap-Log.

Simple-Query is a tool that simplifies the inputting of SNMP commands; it avoids the complexity. The SNMP-URL format has been defined in the previous section. Figure 5 shows the displayed homepage of Simple-Query. Figures 6, 7 and 8 are examples of a query and its result.

Status Code	Reason-Phrase	Meaning
420	Command Format Error	The URL does not correspond to the SNMP-URL
421	NOT WRITABLE	Manager tried to modify a read-only variable
422	BAD VALUE	Server does not set the bad value
423	NO SUCH OBJECT	Server does not find such object identifier
424	NO SUCH INSTANCE	Server has the object identifier, but does not implement the detail
425	END_OF_MIB	Server does not find such object in all MIB
426	BAD COMMUNITY	The community is illegal

Table 5. The extension status-code for SNMP

Figure 3. Th home page of Smart-Web View.

—The MIB-Browser—

The MIB-Browser lets the user select an MIB entity to see its value. It also provides the related description for the selected MIV or TABLE. Figure 9 shows the displayed homepage of the MIB-Browser. The MIBs currently include System, Interface, AT, IP, ICMP, TCP, UDP and SNMP groups.

The title frame shows the tool name, version and copyright notice and the host frame contains two text buttons. The first button is to input the IP address of the managed device. The second button is to input the community name of the managed device. The MIB frame allows users to choose (double click) the appropriate MIB name and it will display the related description file. The old frame shows the related MIB's object identifiers and descriptions. Every object identifier references one

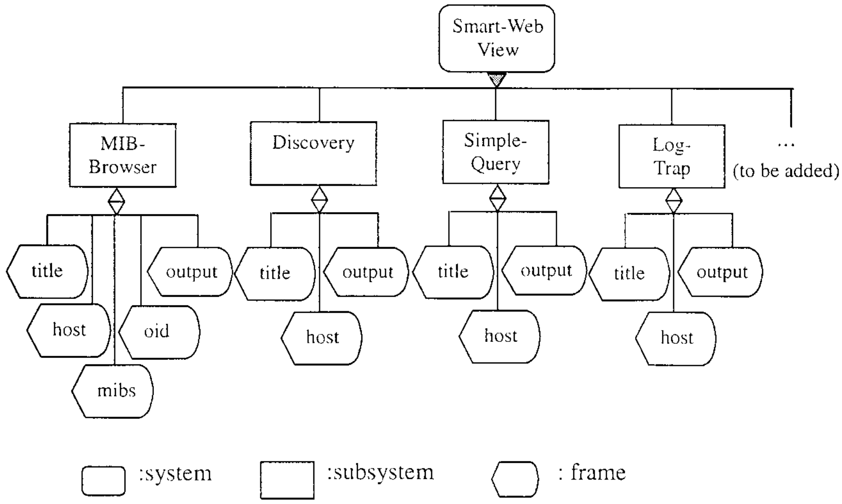

Figure 4. Smart-Web View's organization structure.

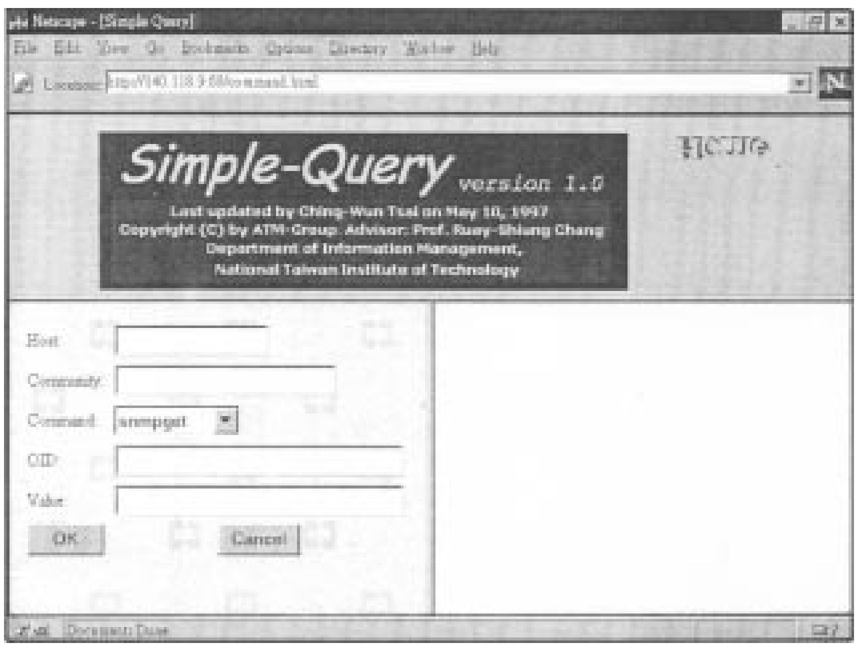

Figure 5. The homepage of Simple-Query.

JavaScript function to compose and send SNMP-URL to the Bilingual Agent in the device. The output window shows the query result. Figures 10 and 11 are examples.

—Discovery—

Discovery is an another useful network management tool whose purpose is to find the SNMP device in a specified IP address range. In addition to checking whether the hosts are alive or not, Discovery also detects the device vendor (system.sysObjectID.0). Figure 12 shows the displayed homepage of Discovery.

The title frame shows the title name, version, last updated date and copyright notice. The input frame includes two text buttons. The first is the beginning search IP address. The second button is the ending search IP address. The output frame shows the

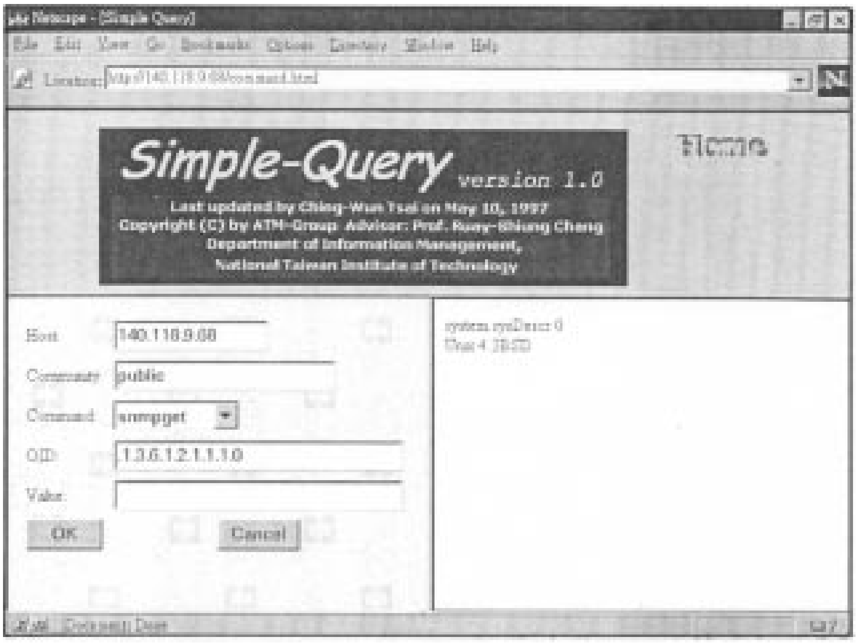

Figure 6. The snmpget example of Simple-Query.

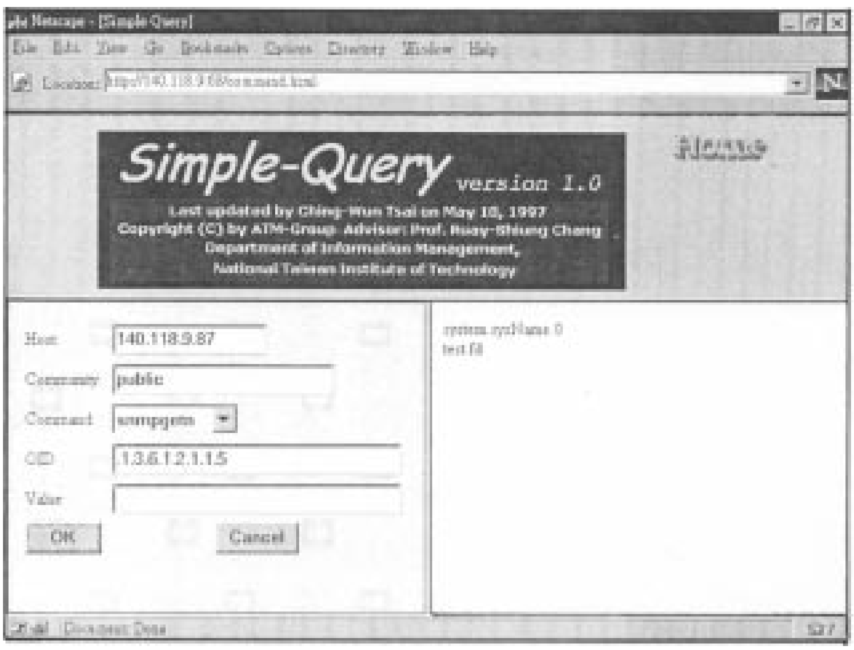

Figure 7. The snmpgetn example of Simple-Query.

detected hosts. Figure 13 shows an example of Discovery.

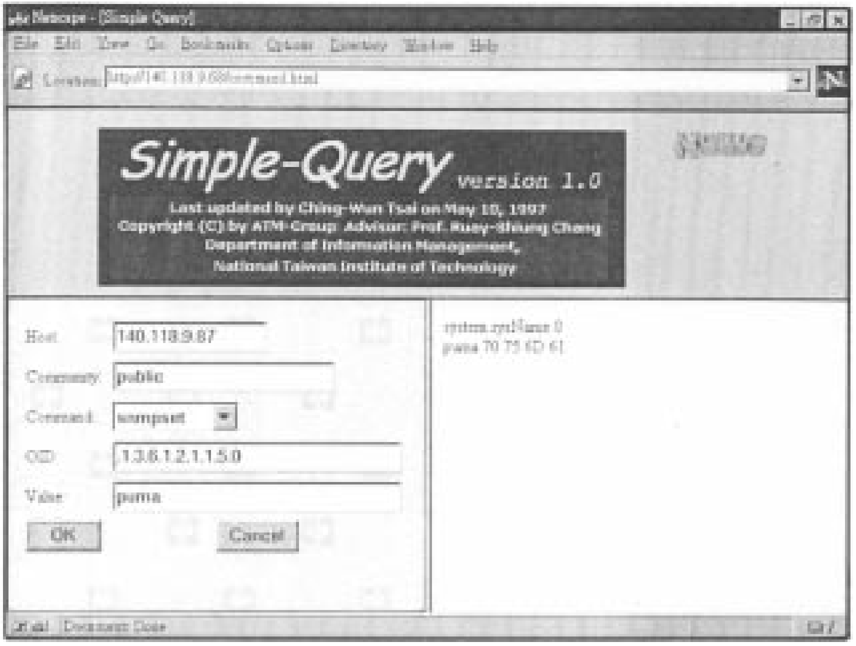

Figure 8. The snmpset example of Simple-Query.

Figure 9. The hompage of MIB-Browser.

—Log-Trap—

Log-Trap is a tool that displays the collected traps in the Bilingual Agent. Because HTTP is designed to accept synchronous data and does not adapt to accept asynchronous data, HTTP cannot accept SNMP traps directly. Therefore, we use a daemon to collect SNMP traps into one file and Log-Trap retrieves the file data. Figure 14 shows the Log-Trap.

The title frame shows the title name, version, last

Figure 10. An example of MIB-Browser.

Figure 11. The example result of MIB-Browser.

updated date and copyright notice. The input frame contains one text button. The button is to choose the trap kinds. Figure 15 shows an example.

Conclusions

By now, it should be clear that SNMP and HTTP share a number of similarities. Both provide the cli-

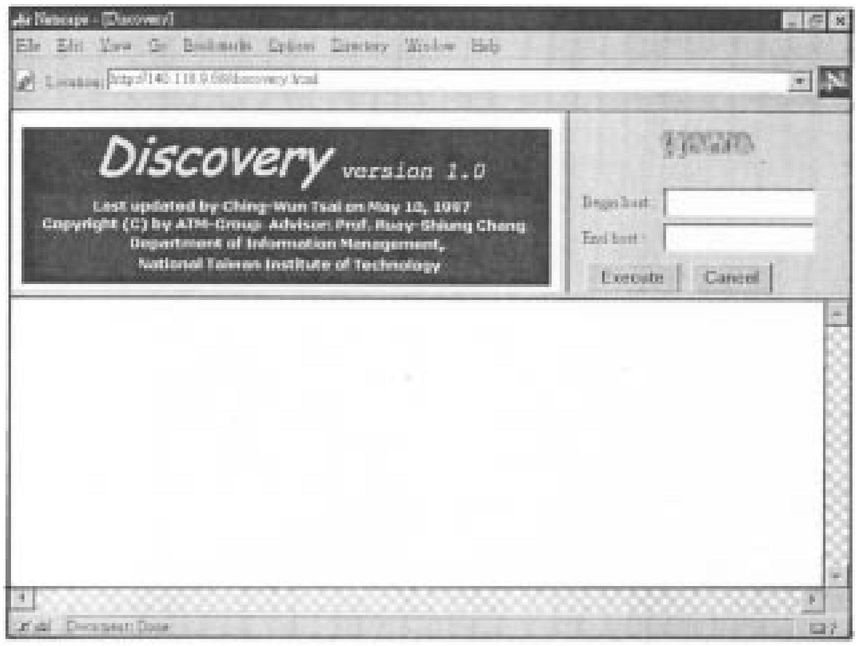

Figure 12. Discovery.

Figure 13. An example of Discovery.

ent (manager) with the ability to modify or retrieve specific resources within a server (agent). SNMP provides a standard mechanism for identifying resources and a standard representation of those resources (a media type). HTTP provides the ability to identify and transport resources of any media type between communicating entities.

SNMP through WWW is both convenient and necessary, especially when the network management operator is not at the place where the dedi-

Figure 14. The Log-Trap.

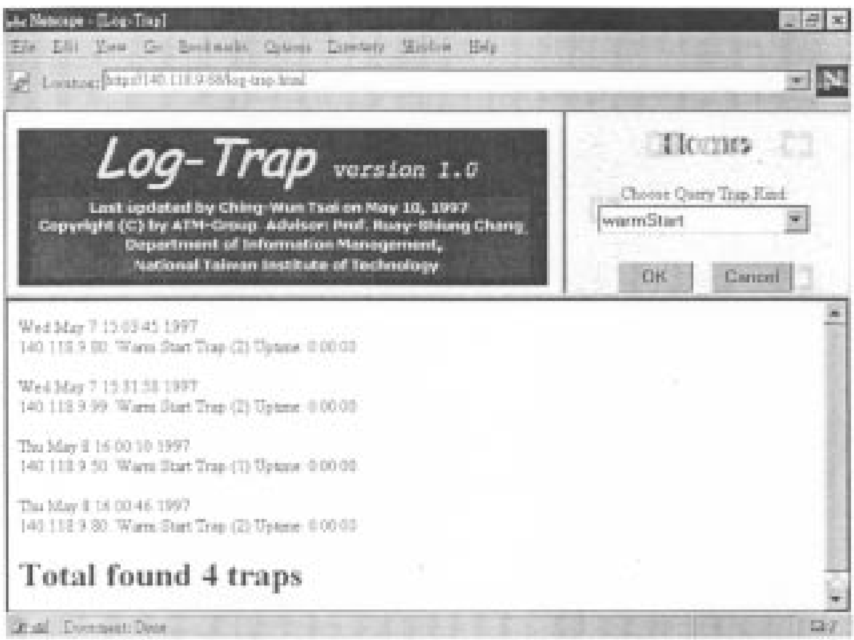

Figure 15. An example of Log-Trap.

cated manager console is located. But the traditional SNMP will survive. The dedicated manager software provides a more effective tool and will be more efficient than the browser using SNMP through WWW. They will coexist in future.

Devices that run both SNMP and HTTP separately will be hard to integrate and combine. This is why the Bilingual Agent is proposed. The Bilingual Agent provides a better performance than the other solutions for SNMP through WWW.

Web-Based management is very popular but there is still much future work to do. First, in order to determine whether an SNMP agent supports HTTP, an SNMP Management Information (HTTP-Enable MIB) will be needed. The manager located at the general network management platform (like OPENVIEW) can detect whether the agent is http-enable or not via querying the HTTP-Enable MIB. Second, there is the security problem. Because the transmission of HTTP does not ensure security, SSL (Secure Socket Layer) technology will have to be added. Third, the use of CORBA (Common Object Request Broker Architecture) to implement SNMP MIB and management applications can be considered. The main benefits of implementing SNMP MIB and management applications using CORBA are as follows. Portability of application across multiple network management platforms is the first advantage. Second, it creates a set of reusable class library of SNMP-based managed resources. Third, it takes advantage of already standardised MIBs defined for network elements. Finally, it reduces the knowledge and skill needed to implement SNMP agents/ applications.

References

1. Robert A. Barta and Manfred Hauswirth, Interface parasite gateways, *Proc. 4th Internat. World Wide Web Conf.*, Boston, MA (1995). http://www.w3.org/pub/Conferences/WWW4/Papers/273/.
2. N. Borenstein and N. Freed, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*, RFC 1521, Bellcore, Innosoft, September 1993.
3. Charles Brooks, Murray S. Mazer, Scott Meeks and Jim Miler, Application-specific proxy servers as HTTP stream transducrs, *Proc. 4th Internat. World Wide Web Conf.*, Boston, MA (1995). http://www.osf.org/www/waiba/papers/www4oreo.htm.
4. Luca Deri, IBM Zurich Research Laboratory of Berne, in *2nd Intel IEEE Workshop on System Management*, 'Surfing Network Resources across the Web'.
5. Stephen E. Dossick and Gail E. Kaiser, 'WWW access to legacy client/server applications', *Computer Networks and ISDN Systems* **28**, 931–940 (1996).
6. Sidnie Feit, *SNMP: a guide to network management*, International Editions, 1995.
7. R. Fielding, H. Frystyk and T. Berners-Lee, *Hypertext Transfer Protocol*—HTTP/1.1, Internet Draft draft-ietf-http-v11-spec-Ox, HTTP Working Group, 4 January 1996.
8. Henrik Frystyk Nielsen and Hakon W. Lie, 'Towards a uniform library of common code A presentation of the CERN World-Wide Web Library', *Computer Networks and ISDN Systems* **28**, 12–23 (1995).
9. Brian Harrison *et al.*, *Web Based System and Network Management*, in "ftp://ds.internic.net/internet-drafts/draft-mellquist-Web-sys-01.txt"
10. Ming-Jeng Lee, Shao-Chi Wang and Min-Sho Kuo, *A Browser-Oriented Remote Management System*, Technical Report, Dept. CSIE, NCU, Taiwan.
11. John Ockerbloom, *Introducing structure data types into Internet-scale information systems*, PhD dissertation, School of Computer Science, Carnegie- Mellon University (1994).
12. Marshall T. Rose, *The Simple Book: An Introduction to Internet Management*, second edition.
13. Jonathan Trevor, Richard Bentley and Gerrit Wildgruber, 'Exorcising daemons: a modular and lightweight approach to deploying applications on the Web', *Computer Networks and ISDN Systems*, **28**, 1053–1062 (1996).
14. D. R. T. Robins, *The WWW Common Gateway Interface*, Version 1.1, Internet Draft draft-robinson-www-interface-01, is February 1996. Available at <URL: http://www.ast.cam.ac.uk/~drtr/cgi-spec. html>
15. Jung-Tsung Tsai *et al.*, *Realizing a Web-based Java-coded remote network monitoring (RMON)*, in *11th International Conference on Information Networking*, Vol. 1, 27–29 January 1997, Taipei International Convention Center, Taipei, Taiwan, ROC.
16. JavaSoft, a Sun Microsystems Business, *The Java-Management API*, http://java.sun.com/products/JavaManagement/profile.htm
17. WWW SNMP MIB Browser, 'http://www.cs.tu-bs.de/ibr/cgi-bin/sbrowser.cgi'
18. Advent Network Management, Inc. *Advent Network Management JAVA SNMP Package*, http://www.adventnet.com/snmp_api.html
19. SNMP Research International, Inc. *Dr-Web: Extensible Agent*, http://www.snmp.com/drWeb.html
20. Lotus InterNotes Web Publisher, Lotus Development Corp. (1995). http://ww.lotus.com/ corpcomm/334a.htm/
21. The Web based Management Page. http://www.mindspring.com/~jlindsay/Webbased.html
22. Web-based Management, http://www.qds.com/net_Webbased.htm
23. Web-Based Enterprise Management, http://wbem.freerange.com/ ■