**借助万维网实现简单网管协议**

**在本文中，我们建议双语代理接受SNMP或HTTP命令并设计多个HTML页面以便于网络管理任务。 对于仅支持SNMP的网络元素，双语代理可以充当代理，因此也可以通过Web浏览器查询传统的SNMP代理。 © 1998 John Wiley＆Sons，Ltd。**

由Ching-Wun Tsai和Ruay-Shiung Chang \*

**摘要**

由于网络技术的快速发展，管理各种网络软件和硬件的任务变得越来越困难。 传统上，为了确保可靠的数据传输服务，使用专用管理控制台来监视和分析网络段上的流量状态。 在一个经常为企业范围网络采用各种网络技术的大型组织中，传统的网络管理解决方案变得过于昂贵。 有必要选择适当的网络管理标准来有效地管理设备。

简单网络管理协议(SNMP)6,10是事实上的网络管理标准。由于其架构简单，几乎每个网络设备都支持SNMP。 嵌入在网络设备中的代理程序收集流量统计信息并将其记录在管理信息库（MIB）中。 网络管理员可以通过实时轮询代理来获取信息。

然而，最近万维网（WWW）的发展改变了网络管理世界。 网络供应商或团体21,22已经开发了许多基于Web的管理方案，这是WWW工具在系统和网络管理中的应用。 这包括使用Web服务器和Web浏览器提供管理信息的静态，动态和交互式内容。 以管理角色行事的Web服务器可以以各种形式提供信息，包括超文本标记语言（HTML），图形，可执行代码（JAVA小程序）和二进制编码信息。 这些功能共同使HTTP（超文本传输协议）可用作管理系统和网络的可能协议。

基于Web的网络管理有三个优点。首先，配置或监控设备不需要专门的管理软件。要利用基于Web的网络管理，只需在客户端运行Web浏览器即可。然后输入正确的URL（统一资源位置），将显示网络管理信息。其次，消除了当旧代理或管理器不支持另一个的新的，可能需要的特征时通常发生的版本问题。当代理支持新的MIB且管理器不支持时，或者管理器的新版本或功能期望来自旧代理的最小MIB支持时，会出现这些问题。通过使用基于Web的网络管理，代理和管理器不需要同时更新。基于Web的代理可以更新MIB，管理员仍然可以在不做任何更改的情况下工作。在多供应商环境中也会发生同样的问题。许多供应商提供嵌入某种级别的供应商特定MIB的设备。这使管理站供应商的工作变得复杂，要求他们为这些特定于供应商的MIB支持额外的屏幕（MIB浏览器根本不适合此任务）。这通常会让网络管理员别无选择，只能为每个供应商的设备购买特定的管理应用程序，然后学会使用所有这些应用程序。通过基于Web的代理提供管理信息，应用程序可以使用Web浏览器的常见熟悉界面，并且浏览器适用于所有平台。下一个主要优点是应用程序的平台和位置独立性。网络管理员可以从任何平台上的任何位置访问Web控制的网络。所需要的只是一个通用的Web浏览器，它是绝大多数平台的标准软件。

为了实现上述优点，我们建议双语代理接受SNMP或HTTP命令并设计几个HTML页面以促进网络管理的任务。 为了使用Web浏览器访问仅支持SNMP的设备中的信息，双语代理必须充当SNMP代理和Web服务器之间的代理。 HTTP上的SNMP操作可以利用协议传输大量信息及其安全功能的能力。 这意味着一个HTTP请求可能包含多个SNMP请求。 双语代理将提高性能并减少SNMP操作的流量。

本文的其余部分安排如下。 下一节将回顾以前与Web相关的应用程序。 第三部分描述了我们提出的系统。 第四部分给出了实现细节。 结论和未来的研究方向将在最后一节中介绍。

**相关工作**

通过WWW进行基于Web的网络管理是一种WWW访问传统客户端/服务器应用程序.5相关主题最近是一个密集的研究活动领域.1,3,11,13,20

Brooks等人开发了HTTP Stream Transducers。 这些服务建模为HTTP代理服务器，允许为每个用户标记修改来自网站的数据流。 例如，当用户访问他们已经看过的网站时，Stream Transducer可以向页面添加HTML，以告知用户他们上次查看此文档的日期。 流传感器可以在用户的Web浏览器和Web站点之间依次连接在一起，以提供聚合门功能。

Lotus'InterNotes'20产品使用CGI机制允许Web浏览器访问由Notes Server管理的文档和表单。 要放在Web上的文档由一个程序预翻译，该程序将它们转换为HTML。 这些文档和表单可以通过标准Web服务器访问，就像它们是普通的HTML文档一样。 对于Notes表单，提交按钮将表单的内容发送到Lotus提供的CGI程序，该程序将数据合并到Notes数据库中。 虽然这确实允许对他们的系统进行一些基于Web的使用，但是交互模型是有限的。 基于Web的用户无权访问标准Notes客户端使用的集成电子邮件和应用程序。

Barta和Hauswirth1描述了一个用于创建“接口 - 寄生虫”网关的工具包。 这些允许通过Web浏览器使用面向会话的同步工具（例如，telnet会话）。 他们的工具包不需要更改Web服务器应用程序的源代码，但它无法处理客户端可以独立于Web服务器执行操作的应用程序。

Ockerbloom11提出了MIME（多用途Internet邮件扩展2）类型的替代方案，称为类型对象模型（TOM），可以使用它来代替MIME扩展，以允许在客户端中使用外部工具。 从Internet上的任何地方导出的对象类型都可以在“类型oracles”中注册，这是专门的Web服务可以相互通信以揭示其他地方注册的类型的定义。 找到他们不理解的类型的Web客户端可以询问类型oracles中的一种如何将其转换为已知的超类型。 这样，就不必设置Web客户端来处理新的MIME类型。 他们可以简单地查询类型oracle，它可以返回有关如何运行外部工具的信息。

Trevor等人提出了一种基于组件的方法，通过使用应用程序集成工具包，使应用程序能够以更灵活的方式从WWW访问。这有助于实现更模块化和“轻量级”的应用程序部署，可根据需要集成特定模块。此外，将协议特定的通信处理组件与应用程序模块分离并提供基于兴趣的集成服务允许为各种其他访问方法（例如电子邮件）启用应用程序，对应用程序进行很少或不进行任何更改-cations本身。

为了将网络管理应用程序集成到WWW中，有一些研究试图提供基于Web的网络管理。

1996年7月，五家主要供应商宣布了一项为基于Web的企业管理（WBEM）定义事实标准的计划.23由微软，康柏，思科，BMC和英特尔牵头的50多家供应商公开支持这一努力。最初的公告要求定义以下组件：

（1）HyperMedia管理模式（HMMS） - 用于表示托管环境的可扩展数据描述，该环境将由桌面管理任务组（DMTF）进一步定义。

（2）HyperMedia对象管理器（HMOM） - 一种数据模型，用于整合来自不同来源的管理数据 - 由Microsoft和Compaq定义的C参考实现和规范，将放置在公共域中。

（3）HyperMediaManagement Protocol（HMMP） - 一种包含HMMS的通信协议，通过HTTP运行，并具有SNMP和DMI（桌面管理信息）的计划接口。

Sun16还宣布了一个用于开发基于Web的网络的编程环境和系统管理软件。这个名为Solstice Workshop的环境由Java Management API（JMAPI），小型数据库和Java编程环境组成。 Solstice Workshop的大型绘图卡是JMAPI的可扩展性和Java的“一次运行一次写入”的吸引力。 JMAPI需要Java，而HMMP/HMMS /HMOM提议指定HTML/HTTP，但没有特别排除Java。

在这些公告发布之前，来自惠普公司（Harrison，Mellquist和Pell）的三位开发人员撰写了一份互联网草案，建议使用端口280来交换HTTP管理数据.9该互联网草案描述了一个非常轻量级的HTTP可管理MIB以及用于SNMP over HTTP的隧道设施。

在这三个提案中，WBEM无疑是最广泛的范围，不仅解决协议问题，还解决数据建模和可扩展数据描述设施。虽然JMAPI包含对象类定义，但就数据建模而言，它并不像HMMS那样深入。哈里森等人。 proposal非常集中，不会尝试定义新协议，数据模型或数据描述工具。 WBEM与JMAPI并不完全相同，因为它专注于用于管理网络和新层次结构的新协议。 JMAPI仅关注管理工具的软件架构，并使用现有协议（如SNMP）。

Advent Network Management Inc.18提出了“Advent Network Management Java SNMP Package”。它使用Java构建SNMP PDU。为了避免Java applet安全问题，它在Web服务器中设置代理服务器以将PDU转发到受管设备。代理服务器也是Java编写的。在这种架构中，服务器和浏览器必须都是支持Java的机器。基于Web的Java远程网络监控15是Java SNMP Package的产品。 Java applet可以提供动态信息并丰富HTML文档。它赋予WWW生命。 Java applet非常强大，Web浏览器供应商必须限制其功能。大多数Web浏览器提供程序不允许Java applet访问本地文件并与除Web Server之外的任何设备进行通信。换句话说，Java编码的基于Web的管理仅监视Web服务器所在的受管设备，并且不在本地文件中存储任何信息。响应时间将更长，因为Java小程序必须包含用于处理SNMP PDU（分组数据单元）的附加SNMP类。 SNMP Java applet代码很大，如果每个人都使用Java-Code SNMP管理来监控设备，整个网络流量将会很大。

DR-Web可扩展代理19是SNMP Research广受欢迎的EMANATE Mas-ter Agent /Subagent系统的增强版。通过DR-Web，SNMP Research扩展了EMANATE主代理，以支持HTTP接口以及现有的SNMP接口。 DrWeb Extensible Agent将HTTP请求和SNMP请求转换为EMANATE事件，并将它们分派给相应的Subagents。该系统非常大，不适合一般的网络设备。

Deri4构建了一个用于网络管理的代理服务器。客户端的Web浏览器必须设置代理服务器，该服务器拦截来自管理器的请求，解码并检查统一资源位置（URL）。如果URL包含网络管理操作，则代理服务器执行这些操作并将响应发送给管理器。如果不是，则代理服务器重定向请求。但是，存在一个“远近”的问题。如果SNMP管理器尝试获取MIB变量，其中代理比代理服务器更靠近管理器，则会浪费网络带宽并且性能会很差。

在参考文献17中，使用公共网关接口（CGI）来实现网络管理操作。 Web服务器扮演代理角色来执行这些操作。换句话说，Web服务器从管理器接收包含网络管理操作的HTTP请求。相关的CGI程序在收到请求后立即执行操作，然后Web服务器将响应发送回管理器。这种方案有几个缺点。首先，网络管理信息的呈现受到HTML（超文本标记语言）的限制。其次，Web服务器的负载很高。每次Web服务器收到运行CGI程序的请求时，都必须在Web服务器计算机上启动新进程。由于有多个客户同时访问数据，因此Web服务器上会产生相当大的负担。最后一个缺点是响应时间很短。由于Web服务器机器必须为每个CGI请求启动新进程，因此Web客户端可能必须等待很长时间才能完成请求。

Ming-Jeng Lee等人提出了一种面向浏览器的远程管理系统。由于SNMP和CMIP的这些数据结构是为程序设计的，因此它们调用CMIP和SNMP API级服务提供的服务。作为使用提供API级服务的协议的结果，需要特定于代理的管理器应用程序在显示之前正确地解码和重新组织管理信息。为了克服这个缺点，他们构建了一个提供浏览器级服务的网络管理协议，称为远程管理协议（REMP），以及一种用于描述管理信息的语言，称为管理接口语言（MIL）。代理以类似于设备向用户显示控制面板的方式向管理员提供管理信息。实际上，浏览器将在GUI上显示每个托管对象作为控制面板。管理信息根据管理接口语言（MIL）指定。这个方案有一些缺点：

（1）它取决于平台和位置。由于此方案中未使用现有的浏览器软件，因此平台和位置将限制新的浏览器软件。

（2）该方案不适应现有的网络管理环境，因为必须事先安装新的代理。

在本文中，我们将开发一个优雅的双语代理，它支持传统的基于SNMP的网络管理和新的基于HTTP的网络管理。表1总结了许多与Web相关的网络管理系统。

**建议系统**

—结构—

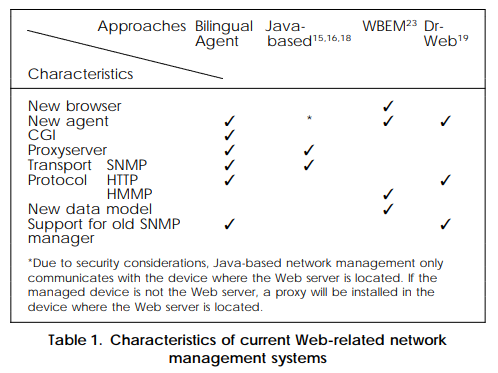
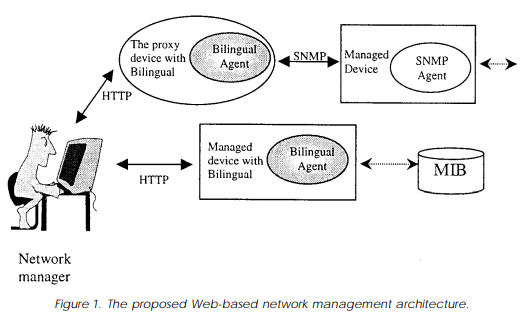


图1显示了用于提供基于Web的网络管理解决方案的建议框架。 该框架具有三个独特的元素：几乎通用的Web浏览器本身; 支持Web的管理应用程序; 和运行双语代理的设备。 管理器可以是在任何机器上运行的Web浏览器。 在工作站上运行的双语代理扮演受管设备或代理的角色。 管理器将带有嵌入式SNMP请求信息的HTTP请求发送给代理。 代理接收请求并在正确处理后发回HTTP响应。 HTTP响应包含一个或多个SNMP响应。 管理器在收到HTTP响应后显示结果。



建议的基于Web的网络管理侧重于与当前Internet协议的兼容性，并不试图定义新的协议，数据模型或数据描述设施。 HTTP是WWW中的当前传输协议。所提出的架构仍然采用HTTP作为传输协议。管理器是Web浏览器（如Netscape Navigator或Internet Explorer）。 Web浏览器无处不在，易于使用，并且比专用管理应用程序便宜得多。

双语代理是提议的基于Web的网络管理体系结构的核心，并在其中扮演代理和代理的角色。它通过HTTP接受命令并从MIB中检索值，然后返回结果。为了符合当前的网络管理环境，双语代理还扮演Web浏览器和现有SNMP代理设备之间的代理角色。作为代理的双语代理在收到嵌入式SNMP请求的HTTP请求后发送SNMP请求。它将在从受管设备收到SNMP响应后返回HTTP响应。

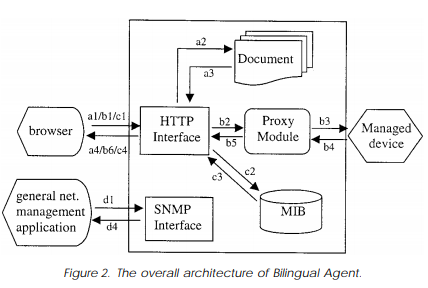
双语代理有四个主要模块，图2显示了整体架构。 HTTP接口模块处理来自Web浏览器的HTTP请求。该模块不仅扮演一般Web服务器的角色，还处理来自Web浏览器的网络管理操作。 SNMP接口模块的功能类似于传统SNMP代理的功能。为了通过Web浏览器查询传统的SNMP代理，将向双语代理添加代理。代理是HTTP / SNMP转换基础结构。代理接收HTTP请求，然后将SNMP请求发送到受管设备。然后代理等待SNMP响应并将HTTP响应发送到Web浏览器。代理由公共网关接口（CGI）实现.14代理模块由HTTP接口模块调用并执行SNMP操作。 MIB模块存储网络管理信息并提供用于检索数据的接口。

图2还显示了双语代理的内部控制流程。在下文中，我们描述了双语代理如何工作的四种可能方式。后面的部分将通过WWW显示SNMP的HTTP消息。控制流程（a1，a2，a3，a4）显示传统Web服务器检索普通HTML文档。 HTTP接口模块从浏览器接收HTTP请求，并验证所请求的对象是否为HTML文档。 HTTP接口模块检索文档并将响应发送回浏览器。

控制流程（c1，c2，c3，c4）显示管理员查询双语代理。管理器从Web浏览器发送请求。该请求包含网络管理操作命令。请求格式将在稍后的部分讨论。 HTTP接口模块接收请求，解析它并获得操作参数。然后，MIB模块检索变量值并将结果返回给HTTP接口模块。 HTTP接口模块将响应发送回管理器（Web浏览器）。

控制流程（b1，b2，b3，b4，b5，b6）显示管理员通过浏览器查询传统SNMP代理。 HTTP接口模块接收请求，解析它并调用代理模块。代理模块将SNMP PDU发送到受管设备，并等待直到收到响应或超时。然后，HTTP接口模块在从代理模块接收结果后将响应发送到管理器。

控制流程（d1，d2，d3，d4）显示传统SNMP命令查询的双语代理。经理将请求发送给双语代理。 SNMP接口模块在处理SNMP PDU后调用MIB模块。 MIB模



块根据SN​​MP PDU中包含的对象标识符将变量值返回到SNMP接口。在SNMP接口模块发回响应PDU后，管理器将收到响应。

—通过WWW的SNMP URL约定—

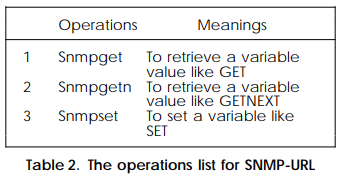
为了使用HTTP管理网络资源，必须定义管理操作和URL之间的映射。 建议的SNMP-URL符合标准URL语法，因此可以由传统的Web服务和Web浏览器处理。 SNMP-URL由三个元素组成，http://<host>/<cooperation>?<parameters>, 哪里：

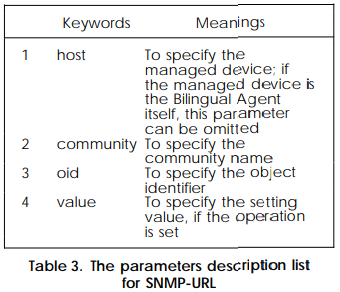
（1）<host>标识双语代理运行的主机

（2）<operation>指定协议操作

（3）<parameters>包含操作参数。

SNMP-URL有三种操作，如表2所示。参数包含四个用于标识各个参数的关键字，如表3所示。





由于SNMP-URL不仅应该被软件应用程序（例如CGI脚本，JAVA Applet）使用，而且还应该被Web浏览器中的人工操作使用，因此拥有简洁明了的语法非常重要。 不允许二进制值，因为人类无法处理它们。 SNMP-URL中包含的对象标识符可以是数字形式和符号形式。 SNMP使用抽象语法表示法1（ASN.1）来描述数据类型。 表4是一些数据类型的示例。以下示例显示如何使用SNMP-URL。 假设双语代理正在主机hercule.cs.ntit.edu.tw上运行：

（1）获取.1.3.6.1.2.1.1.1.0（sysDescr.0）的值。

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

（2）将.1.3.6.1.2.1.1.6.0（sysLocation）设置为`laboratory'。

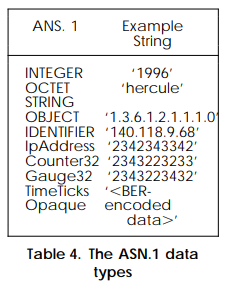
http://hercule.cs.ntit.edu.tw/snmpset?oid=1.1.3.6.1.2.1.1.1.0＆community=public＆value=laboratory

（3）获取.1.3.6.1.2.1.1.1.0（sysDescr.0）的值从其他设备（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）将.1.3.6.1.2.1.1.6.0（sysLocation）设置为另一个设备的“laboratory”（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



—HTTP上的SNMP请求—

定义SNMP-URL后，必须指定HTTP协议返回的信息的格式。 假设指定了以下URL：

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

HTTP客户端将以下数据发送给在hercule.cs.ntit.edu.tw上运行的Web服务器：

GET snmpget?community=public&oid=.1.3.6.1.2.1.1.1.0 HTTP/1.0

[empty line]

HTTP服务器返回的HTTP响应始终为正，除非找不到请求的URL或出现其他问题（例如，身份验证）。如果HTTP响应为正，则它将包含SNMP响应，该响应可以是正数也可以是负数。如果SNMP响应为正，则状态行不会更改，实体正文将包含一组以回车符分隔的对（\_identifier\_，\_value\_）。标识符是对象标识符，通常是符号形式，而值是使用HTTP协议使用的编码方案编码的字符串。此示例在HTTP响应中只嵌入了一个SNMP响应。如果响应包含多个SNMP响应，它将在两个SNMP响应之间插入一个空行。因此，HTTP响应可以包含一个或多个SNMP响应。如果SNMP响应为负，则将添加扩展名状态代码。 （有关所使用的状态代码，请参阅HTTP / 1.17）。表5显示了SNMP的扩展状态代码。实体主体将显示状态代码的描述。在her-cule.cs.ntit.edu.tw上运行的双语代理在收到上述示例中的请求后将以下数据发送到Web浏览器：

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

假设双语代理没有实现对象标识符。 在上面的示例中，双语代理在收到HTTP请求后发送以下数据：

HTTP / 1.0 424 NO SUCH OBJECT

[empty line]

服务器未实现此对象标识符。

**网页实施**

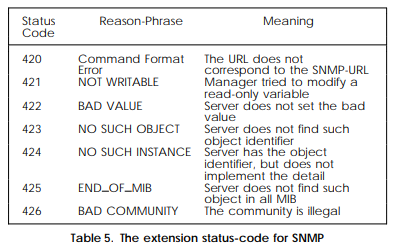
为了演示双语代理，已经设计了几个主页和功能。 以下是简要说明（访问http://hercule.cs.ntit.edu.tw进行测试）。

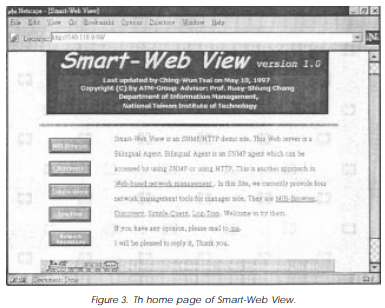
—智能Web视图—

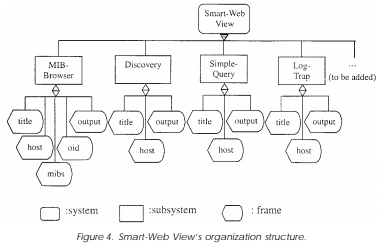
我们将实现一个基于Web的网络管理系统，其中包含代理和管理器。 对于代理方，双语代理是基于CERN万维网库以C语言编写的.8对于管理者方面，提出了Smart-Web View。 Smart-Web View是一个HTML文档，包含四个网络管理工具和相关的网络资源链接。 图3显示了Smart-Web View的显示主页。

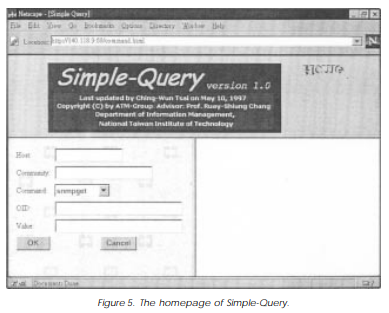
图4显示了Smart-Web View的组织结构。 网络管理工具有四个功能：简单查询，MIB浏览器，发现和陷阱日志。

Simple-Query是一种简化SNMP命令输入的工具; 它避免了复杂性。 SNMP-URL格式已在上一节中定义。 图5显示了Simple-Query的显示主页。 图6,7和8是查询及其结果的示例。









—MIB浏览器—

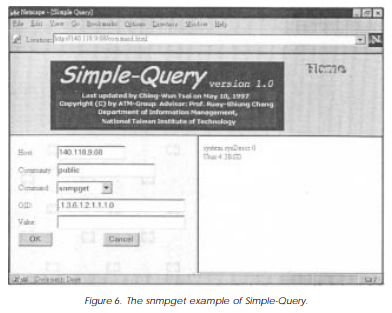
MIB-Browser允许用户选择MIB实体以查看其值。 它还提供了所选MIV或TABLE的相关描述。 图9显示了MIB-Browser的显示主页。 MIB目前包括系统，接口，AT，IP，ICMP，TCP，UDP和SNMP组。

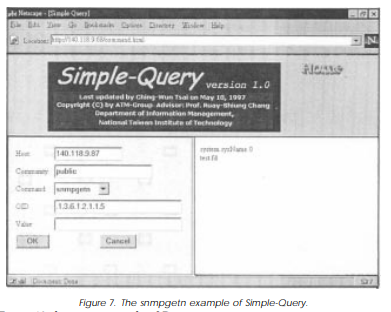
标题框显示工具名称，版本和版权声明，主机框架包含两个文本按钮。 第一个按钮是输入受管设备的IP地址。 第二个按钮是输入受管设备的社区名称。 MIB框架允许用户选择（双击）相应的MIB名称，它将显示相关的描述文件。 旧框架显示相关MIB的对象标识符和描述。 每个对象标识符引用一个JavaScript函数来组合并将SNMP-URL发送到设备中的双语代理。 输出窗口显示查询结果。 图10和11是示例。

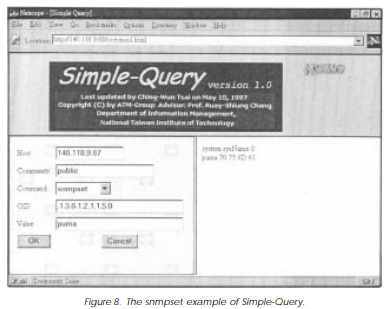
—Discovery—

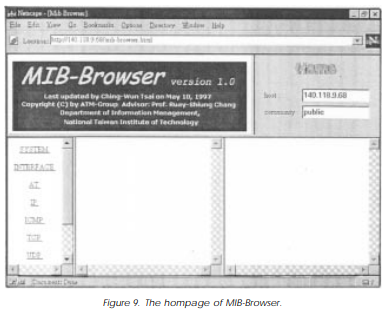
Discovery是另一种有用的网络管理工具，其目的是在指定的IP地址范围内查找SNMP设备。 除了检查主机是否处于活动状态之外，Discovery还会检测设备供应商（system.sysObjectID.0）。 图12显示了Dis-covery的显示主页。

标题框显示标题名称，版本，上次更新日期和版权声明。 输入框包括两个文本按钮。 第一个是开始搜索IP地址。 第二个按钮是结束搜索IP地址。 输出框显示检测到的主机。 图13显示了Discovery的一个示例。



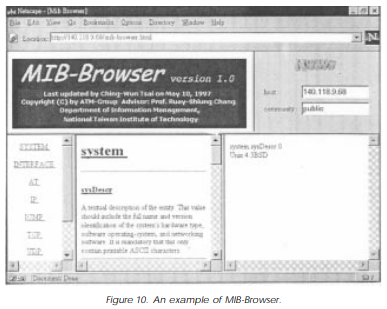


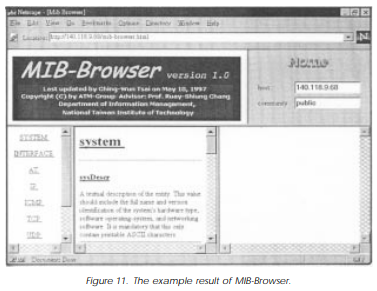




—Log-Trap—

Log-Trap是一种在双语代理中显示收集的陷阱的工具。 由于HTTP旨在接受同步数据而不适应接受异步数据，因此HTTP无法直接接受SNMP陷阱。 因此，我们使用守护进程将SNMP陷阱收集到一个文件中，Log-Trap检索文件数据。 图14显示了Log-Trap。 标题框显示标题名称，版本，上次更新日期和版权声明。 输入框包含一个文本按钮。 按钮是选择陷阱种类。 图15显示了一个示例。





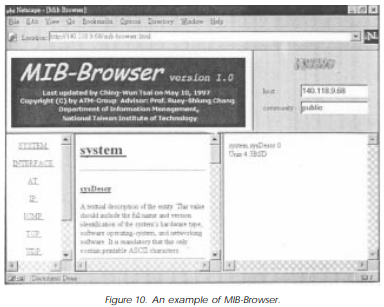
**结论**

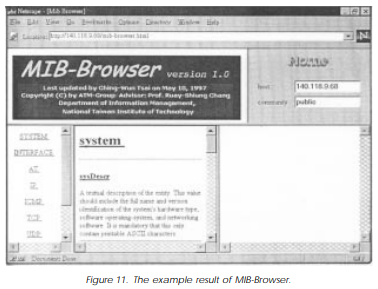
到目前为止，应该清楚SNMP和HTTP共享许多相似之处。两者都为客户端（管理员）提供了修改或检索服务器（代理）内的特定资源的能力。 SNMP提供了用于标识资源的标准机制以及这些资源的标准表示（媒体类型）。 HTTP提供在通信实体之间识别和传输任何媒体类型的资源的能力。

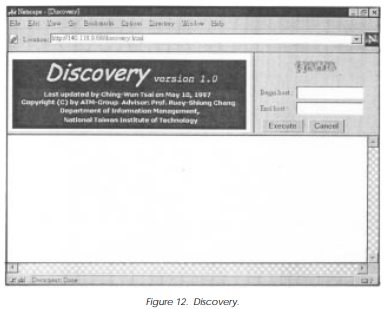
通过WWW的SNMP既方便又必要，尤其是当网络管理操作员不在专用管理器控制台所在的位置时。但传统的SNMP将继续存在。专用管理器软件提供了更有效的工具，并且比使用SNMP通过WWW的浏览器更高效。他们将来会共存。

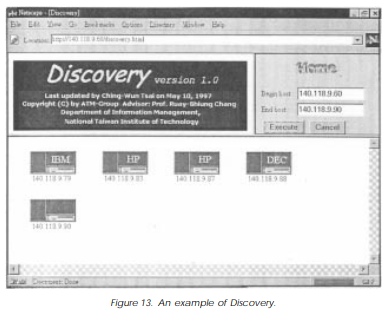
分别运行SNMP和HTTP的设备将难以集成和组合。这就是提出双语代理的原因。双语代理通过WWW提供比SNMP的其他解决方案更好的性能。

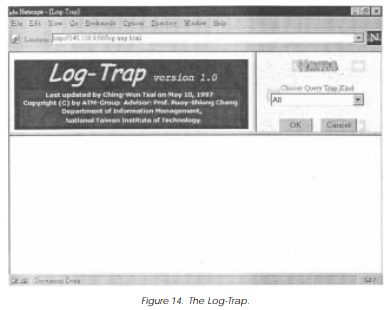
基于Web的管理非常受欢迎，但仍有许多工作要做。首先，为了确定SNMP代理是否支持HTTP，将需要SNMP管理信息（HTTP-Enable MIB）。位于通用网络管理平台（如OPENVIEW）的管理器可以通过查询HTTP-Enable MIB来检测代理是否为http-enabled。其次，存在安全问题。由于HTTP的传输不能确保安全性，因此必须添加SSL（安全套接字层）技术。第三，可以考虑使用CORBA（公共对象请求代理体系结构）来实现SNMP MIB和管理应用程序。使用CORBA实现SNMP MIB和管理应用程序的主要好处如下。跨多个网络管理平台的应用程序的可移植性是第一个优势。其次，它创建了一组基于SNMP的托管资源的可重用类库。第三，它利用为网络元素定义的已经标准化的MIB。最后，它降低了实现SNMP代理/应用程序所需的知识和技能。

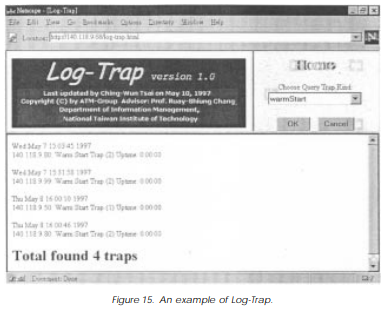












**参考**

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 Inter-net 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 Net-works 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 ProtocolÐHTTP/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 ofthe CERN World-Wide Web Library', Computer Net-works and ISDN Systems 28, 12±23 (1995).

9. Brian Harrison et al., Web Based System and Network Management, in ªftp://ds.internic.net/internet-drafts/draft-mellquist-Web-sys-01.txtº

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 Uni-versity (1994).

12. Marshall T. Rose, The Simple Book: An Introduction to Internet Management, second edition.

13. Jonathan Trevor, Richard Bentley and Gerrit Wildg-ruber, `Exorcising daemons: a modular and light-weight 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-inter-face-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 Jan-uary 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. Kfreerange.com/

|  |
| --- |
| 如果您希望在国际网络管理杂志上订购此文章或任何其他文章的重印，请参阅封面内的特殊重印说明。 |