Why Web-based Network Monitoring? Leveraging the Platform

By Ron Jenkins*

The increasing use of network monitoring and the growth of the Internet and intranets are converging trends that make IP network infrastructures the logical means of delivering network monitoring, using browser-based clients. Copyright © 1999 John Wiley & Sons, Ltd.

Introduction

Two significant networking trends have recently combined to evolve a new form of application—web-based network monitoring. The first trend, the increasing use of network monitoring by increasingly smaller networks, has put powerful network control in the hands of those operating even very small networks of only a few nodes.

The reasons for applying network monitoring to increasingly smaller networks are varied: the growing complexity of networks, the expanding centrality of networks to general business operations, the greater trend to include a measure of network management capability in hardware products, and the escalating sophistication of network administrators, among other reasons. These trends ensure that network monitoring will continue to increase in importance to networks generally, and to small networks in particular.

The second trend is the growth of the Internet and intranets. In addition to the increasing use of networks at the LAN level, most organizations are relying in increasing measure on the Internet to interface their businesses with others, and to interface parts of the same business that are geographically dispersed.

The natural consequence of these forces is a convergence of these trends and the inevitable use of the IP network infrastructure—Internet and intranet—to deliver network monitoring to wherever it is needed.

This article looks at the implications of delivering network monitoring using the HyperText Transport Protocol (HTTP), with particular regard to the differences between web-based network monitoring and more traditional approaches. In this article I use the product LANtracer as my example of a web-based network monitor, although many of my observations about web-based monitoring are generally applicable.

Basic Principles and Structure of Web-based RMON Network Monitoring

Typically, a web-based remote monitoring setup consists of a probe or probes that collect network data, a companion software interface that provides a means to convey the probe’s data to HTML pages, a web server that makes those HTML pages available on the Internet or intranet, and one or more client browsers that interpret the HTML into a more readable presentation. Each of these components is detailed below and is shown in Figure 1. As the probe does not read or write HTML directly, it does not present its screens directly, as would a traditional network monitor. Consequently, this configuration differs from the more conventional management environment that

*Correspondence to: Ron Jenkins, Proteon, 5620 Timberlea Blvd., Mississauga, Ontario L4W 4M6, Canada. Email: rjenkins@silcomtech.com
Figure 1. Web-based remote monitoring

uses a single management console to collect and present data, as shown in Figure 2.

—Probe—

The probe is a device, either software- or hardware-based, that collects network data as a basis for analysis. In the case of a software probe, collection is typically a matter of monitoring network traffic across a promiscuous adapter card to gather general statistics from each frame that is travelling on the network segment. In switched environments, the adapter collects traffic within the segment on which the probe is running. The probe also collects broadcasts and multicasts from other segments, but for a multi-segment view, multiple probes will most likely need to be installed. Alternatively, hardware port mirroring can provide a means of switching the port (or segment) from which a probe gets its data.

In the past, hardware probes were the primary means of collecting frame data, but, as processor speeds have increased, software probes have become able to offer the same functionality at substantially reduced cost, and without the need to dedicate hardware to the exclusive purpose of collecting network information.

No essential differences exist between the functioning of probes for web-based applications and those for non-web-based applications, except that web-based probes speak HTTP so that their data...
WHY WEB-BASED NETWORK MONITORING?

Figure 2. Traditional monitoring environment

can be read and processed by web-based tools (HTML, Javascript, Java, and ActiveX).

RMON is a standard means of collecting data in a format that can be statistically interpreted by the presentation software. It offers standardized tools for basic network monitoring, but most applications will extend the basic set of RMON tools to enable features not provided in standard RMON. In the case of LANtracer, a set of MIB (Management Information Base) objects nominally designated by the ‘WebRMON’ MIB provide the extensions necessary. The specific characteristics and rationale for LANtracer’s collection system are detailed later in this article, in the section ‘Implementing Web-based MIBs’, although in general they can be characterized as modifications designed to facilitate web-based delivery of network monitoring.

—Management Application—

The management application is the collection of web pages and applets that reside on the server and that are executed as needed by the client. Web-based monitoring has the added advantage that its management applications will run on ‘thin’ clients more readily than conventional client/server applications.

—Server—

Any web-based presentation requires an HTTP-based web server to deliver its HTML files to the client stations. In the case of web-based management, the server’s bandwidth demands are not extensive compared to those of a typical Internet web server, but in handling a network monitoring interface some security considerations arise that complicate the basic model of HTTP delivery beyond that of something like a web site, for instance. These security considerations are discussed below.

—Browser (Client)—

A web browser provides the means of handling the formatting and presentation of the data from the web-based monitor and probe. To reduce bandwidth, the client side bears the burden of formatting and displaying raw data, using Java, Javascript, or ActiveX that is executed on the client side.
Convenience is the primary motivation for web-based network monitoring.

Reasons for Choosing HTTP Delivery

—Convenience—

Convenience is the primary motivation for web-based network monitoring. The advantages of network monitoring are the same whether it is web-based or not, but when network monitoring is web-based, a network monitoring client can be located anywhere in the world. This freedom from geographic constraints is particularly useful for monitoring WANs.

A secondary convenience results because client ‘seats’ in such a system can be established at no cost, ad infinitum, and with no client-side setup. Traditionally, network management software has assumed that the user would be located at a specific station in the network, and that on this station would be installed the client-side software necessary to use it. Software vendors in this model could charge a premium for additional ‘seats’ depending on the utility those seats represented to the user. By contrast, web-based network management has no restrictions on such incremental costs, as browser-equipped stations can be made readily available anywhere. And, as script and language processing, graphics handling, file management, and security management are already features of the browser software, no additional client-side software needs to be installed or configured.

Additionally, the web-based model provides a single point of software update: when changes or updates are made to the application, the software of all client stations is in effect also automatically updated.

—Cost—

Savings in cost are a second major motivation for using a web-based presentation for network management. Competition for dominance of access to the Internet has led Microsoft and Netscape to give their browsers away for free. This means that a readily available cross-platform display interface can be had at no cost. Additionally, using an IP network infrastructure such as the Internet to deliver network monitoring information can eliminate long-distance charges.

Client–Server Complications for Network Monitoring

Network monitoring and management software typically performs such functions as presenting graphs and data on key network statistics such as segment utilization, packet rates, error rates, and broadcasts; tracking protocol usage; identifying bandwidth bottlenecks; informing the network manager of network events that need attention; reporting on network responsiveness and connectivity; capturing and decoding packets; and so forth. When these functions involve a simple one-way communication between server and client, the situation is relatively simple. When, however, the client must supply information that needs to be stored on the server, when the client needs to change configuration parameters remotely, or when the client station needs to retain state information on variables to which it does not have continuing access, the case becomes considerably more complicated.

The complications that arise in the web-based model are frequently related to the client–server structure as it is implemented in the web browser environment. For example, an applet downloaded from a server cannot access the client’s resources, and so can’t print or store client-specific configuration information directly. Similarly, the security model for Javascript and Java prevents certain multi-server data structures. For example, a single applet cannot use data from more than one server (such as servers running probes on different network segments), a single applet cannot talk to multiple servers, and a change of state in one HTML frame cannot change the content of another frame if the second frame refers to a different server from the first frame. These constraints have implications for web-based applications that are detailed below.

—Storing Information on the Server—

Network management functions such as retaining log files of URL accesses, alarm notifications,
or packet captures require that the client configure and write information to the server. This information needs to be accessible to the user who has created or collected it from either the initial user’s station or from any other client station. Similarly, other users may wish to set up their own individualized log files, and to have them available wherever a client station has access to the network monitoring probe. Accomplishing these goals through a browser interface is complicated because browsers are intended to maintain isolation from the servers they access, and because downloaded executable code must be either ‘signed’ (ActiveX) or kept within a security ‘sandbox’ (Java). LANtracer handles logging and file storage by storing log files on the server. These log files are accessible through the Windows NT file system on the server. When a user logs in, they are given access to the log files on that server.

—Changing Configuration Parameters Remotely—

Configuration parameters can be classified as client parameters and as server and probe parameters. Changing client parameters through a browser interface is relatively easy, but LANtracer’s security model for browser–server interaction prevents changing the operational characteristics of the probe. Such changes must be managed at the station that is actually running the web server and probe. Changes of this sort include resetting the size of probe buffers, reassigning server ports, changing program paths, and assigning remote login passwords and trusted IP addresses.

Similarly, LANtracer makes changes to the server’s resources only locally configurable. Remotely reconfiguring operating system parameters violates the browser–client security model. Thus, variables that enable the server to send network alarm notifications by e-mail or pager to a remote recipient are configured from the server station only.

—Retaining State Information—

For a network monitor to track non-broadcast and non-multicast activity on more than a single segment, multiple probes need to be run, and to track activity on all segments, each segment must run a probe. This is a condition for any RMON monitoring, which uses a ‘listening’ or passive device to gather information. This condition applies to web-based and non-web-based monitoring alike.

Where web-based monitoring differs in this regard is that the browser interface is constrained from certain presentations and interactions of data from more that one server, as described earlier. As a consequence, presenting state information for different segments is handled by placing probe indicators from different segments in discrete frames, as shown in Figure 3. To see multiple segments simultaneously and in detail, multiple instances of a browser can be started.

LANtracer exploits the browser’s ‘cookies’ feature to retain server IP addresses for the various servers monitoring on each segment configured with a probe. This technique enables the client station to display activity on multiple segments, collected from multiple probes, and to switch between those segments as the user desires. In this way too, a single client station can monitor Ethernet, Fast Ethernet, and Token Ring segments from a single point.

Mechanics of HTTP Delivery

—Strengths—

Apart from the savings in cost, increased geographic versatility, readily available client software, and simple client setup discussed earlier, delivering network monitoring through HTTP offers additional technical advantages. Presentations and reports based on HTTP are readily customized for specific applications by anyone conversant with HTML and perhaps a little Javascript. Indeed, the content of report screens can be easily cut, combined and customized by simply editing the source HTML of the pages involved in any text editor.

HTTP also offers a standardized cross-platform protocol for the network management probe to communicate with the client station, freeing the purchaser of such software from proprietary client-side solutions.

—Challenges—

The primary challenges posed by web-based delivery of network management are consequences
of what is inherent in the web model and infrastructure generally. Specifically, challenges include overcoming slow connections, going beyond the limited functionality of basic HTML, and maintaining client–server security.

—Overcoming Slow Connections—

As is the case with any data transmission, slow connections are naturally alleviated by reducing the quantity of data sent between the client and the server, and by increasing transmission speeds. In the case of LANtracer, virtually nothing is sent from the client to the server, except for the routine polling requests and the infrequent configuration additions and changes to alarm thresholds and logging parameters. Communication from the server to the client is somewhat greater, consisting of HTML pages, Java applets, and network data collected by the software probe. The HTML pages are generally small, and the browser caches the Java applets after their first download. Greater speed can be achieved by installing applets locally, in which case they need not be downloaded and are not flushed when the browser’s cache is emptied.
Finally, the volume of network data transmitted depends on the number of stations in the actively displayed segment, but even complex pages are deliverable in a few seconds over a 28.8-baud connection.

—Going Beyond the Limited Functionality of Basic HTML—

The limited functionality of HTML necessitates creating many of the client-side web pages dynamically using Javascript, and sometimes Java. This solution is quite effective, but requires careful software programming and extensive browser testing to ensure that features of the product that rely on sophisticated programming are supported on the range of browsers on the market. Although HTML is formally defined, Javascript and Java handling differ between Microsoft and Netscape, as well as between versions and sub-versions of browsers by these firms.

—Maintaining Client–server Security—

Maintaining security between client and server is not difficult when the client’s role is simply to observe pages presented by the server, without any interaction with the server side, or when the client station is the station that is running the software probe. However, introducing greater functionality into a remote client-side station requires that a means of authenticating users and sending commands to the server be implemented.

LANtracer has a security system that divides remote users into two security classes. The first class of remote user can only ACCESS information that LANtracer provides, but cannot make any configuration changes. The second class, privileged users, can make configuration changes to LANtracer that include adding, deleting, or altering the protocol monitors and histories that LANtracer creates, and capturing and decoding packets.

LANtracer identifies remote users as privileged in two ways: either the IP address they are using is identified as ‘trusted’, or they have privileged remote login status by being assigned a login name and password.

Trusted addresses are IP addresses which the server considers to have privileged access status independent of whether the user has completed the server login page or not; that is, a browser at a trusted IP address can be used to add, delete, or alter the protocol monitors and histories that LANtracer creates, and to capture and decode packets without its user having a login name and password.

The server login page provides an interface for a remote user to attain privileged access status, independent of whether the user’s address is ‘trusted’ or not. When a user performs a ‘login’ using this mechanism, the user is granted temporary privileged status, which permits the user to add, delete, or alter the protocol monitors and histories that LANtracer creates, and to capture and decode packets without their station having a trusted IP address.

Newer developments such as Java beans and newer versions of Java relax the security model for scripting and applets subject to the user’s assent, and will extend the flexibility of web-based applications such as LANtracer.

Implementing Web-based MIBs

—How do Web-based MIBs Differ from Conventional SNMP MIBs?—

To operate as a web-based application, a network monitor needs extended management information bases (MIBs) to allow it to interface with web tools and protocols. This article looks in some detail at the implementation of MIBs by LANtracer, as a working model of how such an idea can be implemented, although other approaches are also possible. In the case of LANtracer, the web MIBs do not differ from a standard MIB; both sorts are expressed in ASN.1 format and can be compiled by a MIB compiler. However, standard SNMP RMON devices encode management information in a format (Basic Encoding Rules, or BER) that requires a separate application to decode, format, and present in humanly readable format. Web-based devices, on the other hand, must be able to present the same information in a method that is much more directly humanly readable (an HTML document).

As a consequence, the primary difference between LANtracer’s MIBs and standard SNMP
MIBs is in how they present management data to the user and not in the formats of their definitions. Thus, instead of using BER, the LANtracer web server formats MIB objects into a form that can be used in an HTML document.

—Web-based MIB Groups—

LANtracer implements many RMON- and RMON2-like groups (called WebRMON groups here), such as history, matrix, and various protocol groups. The motivation for changing standard groups into proprietary groups is that many RMON groups tend to require complicated access methods requiring many protocol exchanges between the management software and the agent to integrate information for the user.

With a web interface, the goal is to achieve simple, useful integration of information in a single HTML document. To that end, LANtracer’s MIBs are designed so that this can be achieved. This means that the MIBs can be accessed using simpler methods with related information accessible from a single exchange (HTML document).

—How Does Web-based MIB Data Access Differ from Conventional SNMP MIB Data Access?—

With a conventional SNMP-based system, MIB data is accessed via a protocol exchange with an SNMP agent from SNMP management software. The management software performing this operation usually does so using some SNMP API. The user is held at ‘arm’s length’ by the management software.

This is a good thing, since the data in the protocol exchange is usually unreadable by humans. It is not entirely unreadable; a person used to dealing with this stuff can get good at interpreting this data through a painful learning process. However, understanding even as little as the object identification mechanism is painful, and even when that is presented in ASCII dot notation. The downside is that the view of management information is restricted, since it is based entirely on how the management software presents it.

With a web-based presentation, management information is retrieved using an HTML document. Requests for specific kinds of data are embedded into a document, along with any other documentation the user wishes. The notation for requesting data uses well-established HTML conventions for accessing dynamic server side information. As the server sends out a requested page, it scans the outgoing page for management information requests and replaces those requests with data on the way ‘out the door’. Objects are accessed using easy-to-understand object names with simple operations like GET and SET.

—Standard MIB Groups Implemented by LANtracer—

In addition to web-specific implementations, LANtracer implements the following standard MIB groups. While it is outside the scope of this article to delineate specifics, in general these objects are fully compliant, with the exceptions noted above. The groups are:

(A) Interface group (MIB II, RFC 1213)
(B) Host Group (RMON, RFC 1757)
(C) Ethernet Statistics Group (RMON, RFC 1757)
(D) Token Ring MAC-Layer Statistics Group (Token Ring Extensions to RMON, RFC 1513)
(E) Token Ring Promiscuous Statistics Group (Token Ring Extensions to RMON, RFC 1513)
(F) Token Ring Ring Station Group (Token Ring Extensions to RMON, RFC 1513)

Future directions for web-based network monitoring

—Scripting and Programming—

Web-based solutions to network monitoring and management will only become more effective in the future. Developments such as XML (eXtensible Markup Language), virtual private networks (VPNs), increased security, faster Internet connectivity, and more robust browsers will all assist in making web-based delivery of network management smoother and more powerful.

XML, soon to be supported by Microsoft and Netscape’s browsers, will move a number of programming and formatting functions that are
now the province of scripts and applets to a more versatile and compact markup language than HTML. Essentially, more of the nuts and bolts of information massage will be automatically handled by the client, less explicit formatting will need to be designed into the HTML pages, and the bandwidth requirements to transport such data will be reduced.

VPNs will offer numerous advantages to those designing network management software for delivery over the Internet. Indeed, in the longer term VPNs may well offer solutions to both handling client–server security and to interface weaknesses that are now complicated by the limitations of HTML. With these solutions may come the re-institution of proprietary client-side solutions, however.

Internet security has increased substantially in a very short time, and continues to improve. Browsers now offer sophisticated checks on the security of files they handle, and users have numerous settings on the levels of security they wish to assign to scripting, languages, and file sources.

Conclusion

Viable web-based network monitoring applications are now beginning to appear in the marketplace, and the obstacles to their growth and use diminish daily. The drawbacks and complications of implementing web-based monitoring are technical rather than absolute, and are largely solved. From here, refinements to web-based applications will undoubtedly add power and functions that will match network monitors generally, and that will extend the reach of web-based applications to full-blown SNMP management applications. Indeed, many recent issues of hardware include web-based SNMP managers already. It will not be long before someone relaxing on a sunny tropical beach will be able to manage almost all facets of a remote network.