A DYNAMIC SNMP TO XML PROXY SOLUTION

Ricardo Neisse, Lisandro Zambenedetti Granville, Diego Osório Ballvé,
Maria Janilce Bosquiroli Almeida, Liane Margarida Rockenbach Tarouco
Federal University of Rio Grande do Sul - Institute of Informatics
Av. Bento Gonçalves, 9500 - Bloco IV - Porto Alegre, RS - Brazil
{neisse, granville, dob, janilce, liane}@inf.ufrgs.br

Abstract: The network management area has some proposals to use XML to encode information models and managed object instances. In this paper we present a solution to dynamically create SNMP to XML proxies using a SAX parser and the translation facilities from the libsmi tools. We also present an analysis system that uses the management information provided by the proxies in XML.

Keywords: Web-based Network Management, SNMP, HTTP, XML, XPath

1. INTRODUCTION

The information used to manage computer networks are tipically defined according to some rules (e.g. SMIv2, SPPI, XML), and retrieved using some protocol (CLI, SNMP, COPS, HTTP). Currently, an important problem is that the set of different options for the definition of management information and protocols increases the complexity of managing a network, since there is no consensus in a single definition language and protocol.

If a unique definition language could provided (e.g. SMIng [1]) and accepted, the other problem will still remain: which unique protocol should be used? In our view, this question is unsolvable because we believe that several different protocols will be still required to manage older devices. However, from the network administrator point of view, the lack of consensus on a single protocol should not refrain the use of a single representation of the retrieved information. To allow that, protocol and information representation translations is needed.

Although the SNMP is the de facto TCP/IP management protocol, its management information is defined through SMIv1 or SMIv2, which is not suitable when we are searching for a common representation. XML, however, seems to be more appropriated, besides being already addressed by the SMIng working group. We developed a system that automatically generates SNMP/XML proxies that reside in HTTP/HTTPS servers. The proxy generating system receives a SMIv1 or SMIv2 MIB definition as source parameter and creates a PHP4 script file that is the proxy itself. The just created proxy can then contact a target device via SNMP and generates a XML-based result. We have used the `libsmi` [2] package to support the generation of the XML files, and the `expat` package to provide the PHP4 support for SAX (Simple API for XML). We have validated the proxy system through its use in a RRDTool-based [3] monitoring front-end.

2. ARCHITECTURE AND IMPLEMENTATION

Figure 1 shows, how a proxy operates **after** its creation. A network management station (NMS) retrieves information throughout a SNMP/XML proxy hosted by a HTTP/HTTPS server. Each server can hosts several proxies, and the selection of which proxy should be used in done in the URL passed from the NMS to the server. Additionally, the selected proxy receives the address of a target device and an SNMP valid community that are used to access the target device via SNMP. Normally, one single access to a proxy generates several SNMP accesses to the target device, mainly when the information to be retrieve is stored in MIB tables. After the SNMP information is retrieved from the target device, the proxy compiles such information into a single XML and sends it back to the NMS.

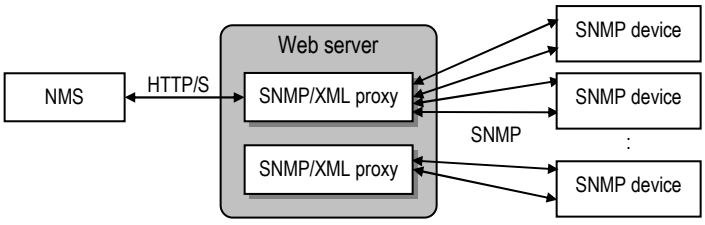

Figure 1. SNMP/XML Proxy operations

Comparing the amount of management information found in the NMS/proxy interactions, it is fewer than the amount of management information found in the proxy/target device interactions. Thus, pushing SNMP/XML proxies closer to the managed devices will reduce the overall amount of management traffic. Also, since we based our implementation in the `smidump` tool, the XML returned to the NMS contains not only the value associated to the management information, but also the whole description of such information originally defined in SMIv1 or SMIv2, allowing a new NMS to discover these definitions on demand.

The proxies are implemented as PHP4 scripts. New MIBs could be supported only through the development of new PHP4 proxies. With the great variety of available MIBs, creating new PHP4 scripts every time a new MIB is required would be a quite slow process. To solve that, we have automated the processes of creating new proxies in our solution.

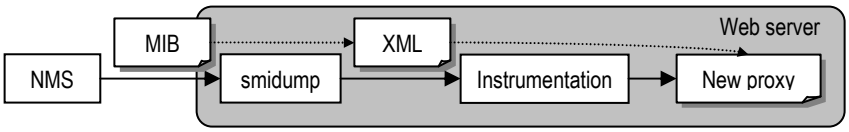

Figure 2. Architecture for SNMP/XML proxy creation

Figure 2 presents the steps to create new PHP4 SNMP/XML proxies. First, a SMIv1 or SMIv2 MIB is uploaded to the server that will host the new proxy. Inside the server, the `smidump` checks the passed MIB and if no errors are found it generates an XML temporary file. This file is then instrumented adding PHP4 code that

can contact SNMP-enabled devices. The proxies is then stored in a standard directory in the server, as well as the original MIB (for documentation purpose) and the XML intermediate file.

3. ANALYSIS TOOL

We have also developed an XML analysis tool that uses the SNMP/XML proxies. We have used the RRDTool [3] to store performance data and the MySQL to store configuration data. Basically, the tool is a generic monitor that collects XML files addressed in URLs. Any information available in XML can be monitored, which includes, obviously, the SNMP data indirectly provided by the SNMP/XML proxies.

The tool is also based on Web technology and accessed through HTTP/HTTPS. The network administrator defines which information should be monitored, and which proxies have to be used. Other information required is the IP of the target device, the SNMP community string and an XPath expression which locates, inside the retrieved XML, the specific information to be analyzed. All this configuration data is then stored in the MySQL. For example, the configuration data required to monitor the incoming traffic in the interface 2 of the IP 200.132.73.54 throughout the interfaces.xml.php proxy hosted by noc.metropoa.tche.br are:

Target device:	200.132.73.54
SNMP/XML proxy:	interfaces.xml.php
SNMP comunity string:	public
Proxy Web server:	noc.metropoa.tche.br
XPath expression:	val[@oid="interfaces.ifTable.ifEntry.ifInOctets.2"]/@value

Figure 3 presents one possible configuration for the analysis tool and a proxy interaction. In this case, both analysis tool and the proxy are located within the same server. Due to this configuration there are no network traffic overhead between the proxy and the analysis tool.

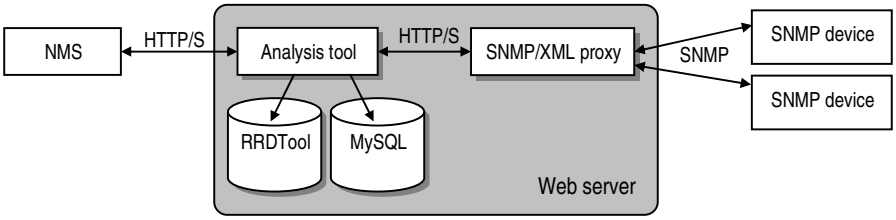

Figure 3. Analysis tool accessing an SNMP/XML proxy

Figure 4 presents a real traffic data analysis generated through the Aberrant Behavior Detection (ABD) [4] algorithm of a university campus link in the Brazilian National Research Network backbone. The thick line is the observed value of the incoming traffic and the thin lines are min and max bound values (confidence band).

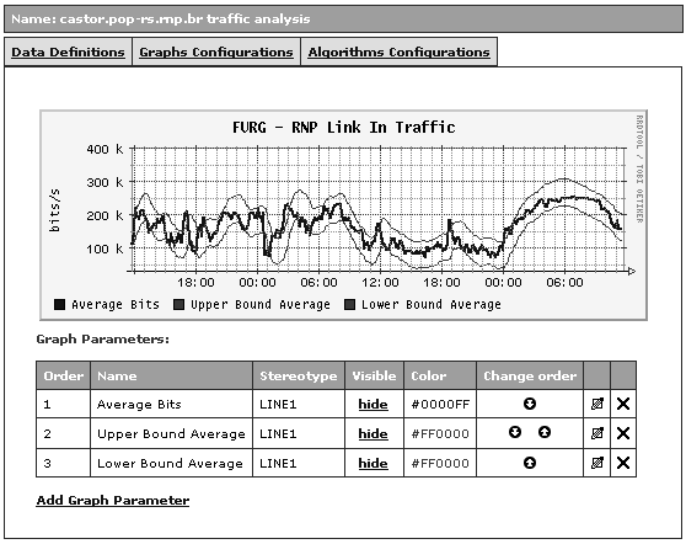

Figure 4. Analysis tool snapshot for the Anomalous Behavior Detection

4. CONCLUSIONS AND FUTURE WORK

We presented in this paper a dynamically SNMP/XML proxy creating solution that produces SNMP/XML proxies from standard SMIv1 or SMIv2 MIBs. Since the created proxies reside inside Web servers, they act as intermediate managers that uses SNMP to retrieve management information and generates XML document as a result.

We have also presented the monitoring tool that uses the SNMP/XML proxies to analyze the network behavior. Proxies and the management tool could be located into a different device, differently from the example presented in figure 3, and no modifications are need to the architecture, as the access to the proxy is done through HTTP/HTTPS and, therefore, it is transparent to the analysis tool the physical location of the proxy.

One improvement for the SNMP/XML proxy is the implementation of a filter that would receive and XPath expression as an extra parameter in order to specify only the specific data that should be fetched and transferred to the management application. This would reduce the traffic between the NMS an the SNMP/XML proxy and also will reduce the processing overhead in the target device.

REFERENCES

[1] F. Strauss, J. Schoenwaelder. SMIng - Next Generation Structure of Management Information, draft-ietf-sming-02, July 20, 2001.

[2] F. Strauss. Libsmi - A library to access SMI MIB information, http://www.ibr.cs.tu−bs.de/projects/libsmi/.

[3] Oetiker T. Round Robin Database Tool (RRDTool) http://www.rrdtool.org

[4] Brutlag, J. D. Aberrant Behavior Detection in Time Series for Network Monitoring, Proceedings of the 14th Systems Administration Conference (LISA 2000) New Orleans, Louisiana, USA December 3-8, 2000.