A Web-based System to Monitor and Analyze Network Management Information in XML

Ricardo Neisse, Lisandro Zambenedetti Granville,
Maria Janilce Bosquiroli Almeida, Liane Margarida Rockenbach Tarouco

Federal University of Rio Grande do Sul - UFRGS
Institute of Informatics
Av. Bento Gonçalves, 9500 - Bloco IV
Porto Alegre, RS - Brazil
{neisse, granville, janilce, liane}@inf.ufrgs.br

Abstract. This paper presents a network management tool to monitor and analyze XML data. The source data to be analyzed is numeric information found in XML documents retrieved via HTTP or HTTPS through a mechanism called SNMP to XML Proxy. The tool is Web-based and shows the analysis results through graphics generated by the RRDtool toolkit. All the monitoring and analysis configurations are stored in a MySQL relational database and can be easily and dynamically changed through the tool Web-based interface.

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

1 Introduction

There are some tools using the RRDTool [1] toolkit to monitor and analyze network management information. The basic functionality of such tools is to query periodically the network devices (polling), usually using SNMP, and to store the data monitored to a future or real time analysis through graphics or algorithms. The RRDTool toolkit provides and Application Programming Interface (API) to perform both functions: to store the monitored data and to plot the resulting analysis graphics.

One limitation of such tools is related with the protocol support to query the devices. In most of the cases, only information accessible using SNMP can be monitored and analyzed, and all the configurations are stored in a plain text file. Depending in the amount of the information and devices monitored, these text-based configuration files became difficult to manage due to the large number and complexity of the configuration directives.

This paper proposed a Web-based tool, implemented as a frontend to the RRDTool toolkit, to do the monitoration and analysis of network management information in the Extensible Markup Language (XML) format. XML are being pointed by some authors [2] [3] as a solution to the representation of network management information with some advantages in relation to the traditional approach of the Internet Engineer Task Force (IETF), based in the Structure of Management Information (SMI) [4] standard of the Simple Network Management Protocol (SNMP).

These article is organized as follows: Section 2 presents an analysis of some polling based monitoring tools, Section 3 shows the SNMP to XML proxy, Section 4 describes the modules of the developed tool and Section 5 finishes this paper with conclusions and future work.

2 Related Work

There are some polling based tools for monitoring and analyzing trends in time-series freely available to use. Examples of such tools are: Multi Router Traffic Grapher (MRTG) [5] and Cricket [6]. In general these tools operates as follow: the network administrator defines the tool configuration in text-based configuration files manually, or using some command line wizards, and accesses the resulting data, usually graphics and algorithms results, through a Web-based interface.

The MRTG tool is mainly used to monitor the traffic load (in and out traffic) on network-links through graphics in HTML pages. Considering networks architectures with QoS support, and a lot of more information to be analyzed, these approach provides a limited support to the analysis of the

network behavior. In practice, MRTG can be used to monitor any information desired, but to achieve such flexibility, the configuration is not easily done without an exhaustive reading of the tool documentation. There are command line wizards only to generate the configuration to the basic interface traffic analysis, any additional monitoration of data out of this pattern must be hard coded.

Based on the MRTG implementation Oetiker developed RRDtool [1], an API to help on the development of MRTG like tools with functions to store performance data and to plot graphics. RRDtool is the API used by Cricket and many other tools [7]. Even so, with many frontends available, most of the solutions using RRDtool do not provide facilities to dynamically configure the monitoring and analysis processes directly in the user interface. Most of the tools are still using text file based configurations.

Cricket, for instance, reads the configuration data from plain configuration files called config tree. The format of the file was designed to minimize redundant information and the configuration express how Cricket accesses the devices, queries the data, and also how to store and to present the information monitored. The resulting of the monitoration process generated by Cricket is visualized by the network administrator in Web pages and, if the administrator decides to change the configuration, the only way to do this is editing the plain configuration files in a text editor.

In spite of the difficult management of the configuration data, the tools available provide facilities only to access information using the SNMP protocol. SNMP is the de facto standard to access network management information but new standards are being proposed, for instance, using XML to code the management information [2] [3]. Thus, flexibility is a requirement and there are no tools with support to monitor and analyze new kinds of information sources.

In these paper we present a Web-based tool and architecture to monitor and analyze network management information presented in XML documents. The tool is an RRDtool frontend and provide many features to allow an easy configuration and management of the monitor and analysis process directly through the user interface, without the requirement of editing text-based configuration files. The following section presents the tool architecture and the modules implemented.

3 SNMP to XML Proxy

The tool implemented only monitor and analyze XML data and, currently, XML is not supported by all the network devices, in opposition to SNMP. Thus, to make the tool compatible and useful with old devices, only with SNMP support, some conversion mechanism is required. The operation of our tool is based in a sub-component called SNMP to XML proxy.

SNMP to XML proxies are PHP scripts, accessed through HTTP or HTTPS, to translate SNMP-retrieved management information to XML documents. SNMP is used because it is the de facto management protocol widely supported in network devices. However, the information accessed by SNMP is originally defined using SMIv1 or SMIv2, which is not suitable when we are searching for a common representation. XML, on the other hand, seems to be more appropriated as a common language, besides being already addressed by the SMIng working group [8]. XML files can be easily parsed and manipulated because they are text-based, human readable, protocol independent and supported by several current Web browsers and tools.

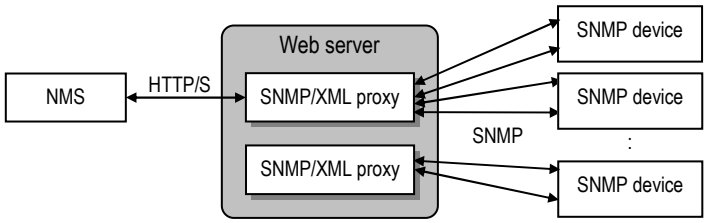


Fig. 1. SNMP to XML Proxy operations

Figure 1 shows, how a proxy operates **after** its creation. A network management station (NMS) retrieves information throughout a SNMP to XML proxy hosted by a HTTP or HTTPS server. Each server can hosts several proxies, and the selection of which proxy should be used is 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. Normally, one single access to a proxy generates several SNMP accesses to the target device, mainly when the information to be retrieved is stored in Management Information Base (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.

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 to XML proxies closer to the managed devices will reduce the overall amount of management traffic. 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 used by the monitoring and analysis tool are not created manually, they are dynamically created using the network management information defined in SMI [9].

4 Tool Modules

The tool is web-based, can be accessed using any browser only with HTML support, and is divided in four modules: SNMP to XML Proxies, SNMP to XML Proxy Builder, Devices, Performance Database, and Analysis Configurations. In the following subsections each one of the modules of the tool are described in details. Configuration samples and screenshots are presented for a better understanding of the tool operation. To access the tool a password is required and all the communication is done using HTTPS.

4.1 SNMP to XML Proxies

The tool manage the SNMP to XML proxies which specify the management information available to monitor and to analyze. New proxies can be added to represent any management information needed. The dynamic inclusion of new proxies gives more flexibility to the tool for future extensions. Figure 2 presents an snapshot of the SNMP to XML Proxies Module.

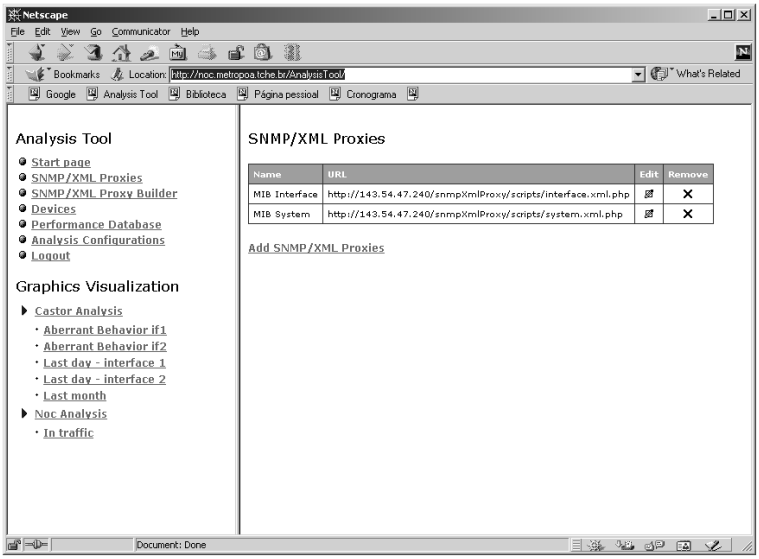


Fig. 2. XML to SNMP Proxies

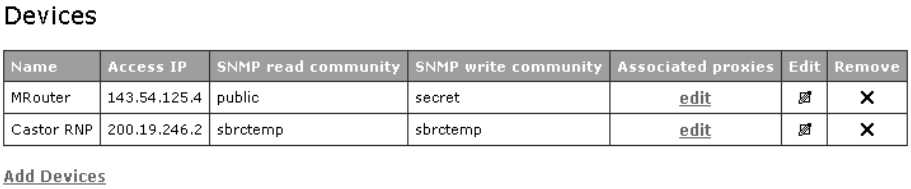
As presented in the figure 2, the tool if full web-based, and can be accessed using any web browser. The tool was designed in two HTML frames, the left frame presents the modules and shortcuts to visualize the graphics, and the right frame shows the information selected. To reduce the space occupied by the screen snapshots, starting at these point, only the right frame will be present in these paper.

4.2 SNMP to XML Proxy Builder

The SNMP to XML Proxy Builder module allows the automatic creation of SNMP to XML proxies that reside in HTTP or HTTPS servers. The proxy generating system receives an SMIv1 or SMIv2 MIB definition as source parameter and creates a PHP4 script file that is the proxy itself. The just created proxy, when accessed via HTTP or HTTPS, contacts a target device via SNMP and generates a XML-based result. The proxy implementation [9] use the `smidump` tool from the `libsmi` [10] package to support the generation of the XML files, and the `expat` [11] package to provide the PHP support for the Simple API for XML (SAX) [12] parser.

4.3 Devices

The tool manage a list of devices that can be monitored and analyzed. For each device the user should inform a description, the IP address and the SNMP read and write communities. Devices can be routers, switches, or any device supporting SNMP agents that can be accessed using the SNMP to XML proxy. The figure 3 shows a screenshot sampling one possible device list.


Devices

Name	Access IP	SNMP read community	SNMP write community	Associated proxies	Edit	Remove
MRouter	143.54.125.4	public	secret	edit		✕
Castor RNP	200.19.246.2	sbrctemp	sbrctemp	edit		✕

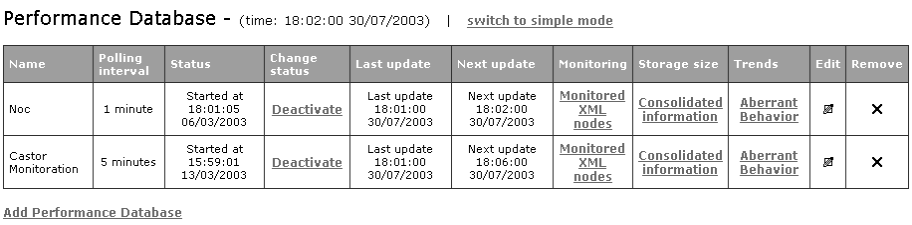
Add Devices

Fig. 3. Devices

Each device can be associated with a set of SNMP to XML proxies, indicating which device support queries using some specific proxy. In the monitor configuration only the proxies supported by a device will be presented to the user, which reduce the amount of information presented to the user of the tool and turn the process of configuration simpler. These process of association could be automatic implemented using some discovery mechanism.

Performance Database After defining which SNMP to XML proxies are supported by the devices, the network administrator should define which information will be monitored. This is done through the performance database configuration. These module can operate in the simple and expert mode, the user can alternate between one and other at every moment.

In the simple mode some configurations are hidden and receive default values. In the expert mode the user must configure the polling interval, how the information will be consolidated and the consolidation to trend analysis (aberrant behavior analysis). Figure 4 presents a snapshot of a list sampling some performance database configurations in the expert mode. In the simple mode the only information requested to the user is the XML nodes monitored, the other information receives default values.


Performance Database - (time: 18:02:00 30/07/2003) | switch to simple mode

Name	Polling interval	Status	Change status	Last update	Next update	Monitoring	Storage size	Trends	Edit	Remove
Noc	1 minute	Started at 18:01:05 06/03/2003	Deactivate	Last update 18:01:00 30/07/2003	Next update 18:02:00 30/07/2003	Monitored XML nodes	Consolidated information	Aberrant Behavior		✕
Castor Monitoration	5 minutes	Started at 15:59:01 13/03/2003	Deactivate	Last update 18:01:00 30/07/2003	Next update 18:06:00 30/07/2003	Monitored XML nodes	Consolidated information	Aberrant Behavior		✕

Add Performance Database

Fig. 4. Performance database

The first option after the creation of the performance database is the specification of which information will be monitored. This is done through XML nodes (Monitored XML Nodes in the Figure 4). XML

Nodes are references to the content of an XML Element or Attribute in an XML document returned by an SNMP to XML proxy. This monitored node must be obligatory numeric, as the tool only support numeric information. Figure 5 show a sample of a configuration of monitored nodes.

Monitored XML Nodes | switch to simple mode

XML/SNMP proxy	Name	XPath expression	Data type	Min	Max	Heartbeat	Edit	Remove
Device: 200.19.246.2								
MIB Interface	ifInOctets1	//val[@oid="IF-MIB::ifInOctets.1"]/@value	COUNTER	0	0	3600	✎	✕
MIB Interface	ifInOctets2	//val[@oid="IF-MIB::ifInOctets.2"]/@value	COUNTER	0	0	3600	✎	✕
MIB Interface	ifOutOctets1	//val[@oid="IF-MIB::ifOutOctets.1"]/@value	COUNTER	0	0	3600	✎	✕
MIB Interface	ifOutOctets2	//val[@oid="IF-MIB::ifOutOctets.2"]/@value	COUNTER	0	0	3600	✎	✕

Add XML node

Fig. 5. XML nodes monitored in expert mode

The identification of the XML node is done through an XPath expression [13]. These expression does not need to be manually typed by the user, the tool provides a selector concept, called XPath Selector, who makes easy the specification of the XPath expression and do not require from the user any previous XML knowledge. Figure 6 presents the XPath Selector in action. If the user has XML knowledge, he/she can manually enter the XPath expression as his/her needs requires.

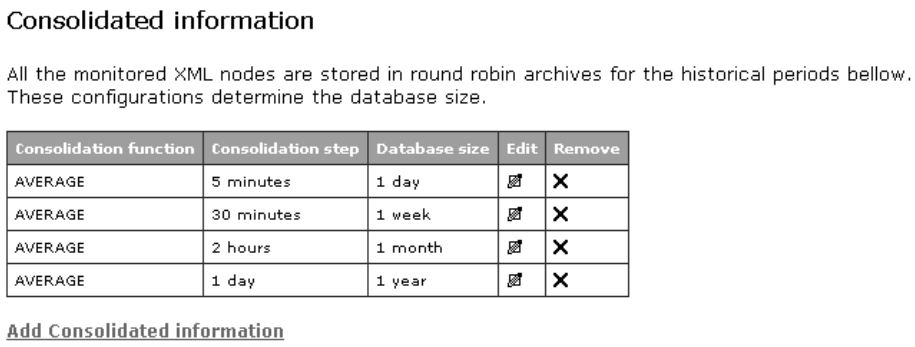
Fig. 6. Wizard to generation of the XPath expression

In addition of the XPath expression, an XML node has a name, that must be unique in the performance database, a data type (COUNTER, GAUGE, DERIVE or ABSOLUTE), a maximum value, a minimum value and a minimum heartbeat, which is the minimum interval allowed between updates to a null (unknown) value not to be stored in the performance database. The options minimum value, maximum value, and minimum heartbeat are visible only in the expert mode.

A correct definition of the minimum heartbeat is important because, in some cases, is common to the tool to have problems in querying the devices, or, to query the devices and get an invalid response, due to network problems. In these cases, to avoid a null valued to be stored in the performance database, the last information fetched is re-used. Minimum heartbeat defines a validity, in seconds, in which the information queried could be used in subsequent updates with faulty queries. The standard value used in the simple mode is one hour.

The consolidated information defines the amount and granularity of the information to be in fact stored by the tool. The consolidation functions provided are: AVERAGE, MAX, MIN and LAST, which calculate, respectively: the average, maximum, minimum and the last value for a specified time interval. These time interval must be obligatory greater and multiple of the polling interval, otherwise it will be automatically rounded to match.

The files with the consolidated information do not grow indefinitely, it is defined a maximum historic size to be stored. For instance, the half hour average of the last six months could be one configuration. When these period of six months is reached the oldest information stored start to be rewritten, what turn easy the management of the monitor files. Figure 7 shows the standard configuration generated when a performance database is created in the simple mode. In the expert mode the user can choose and alter these information if desired.


Consolidated information

All the monitored XML nodes are stored in round robin archives for the historical periods bellow. These configurations determine the database size.

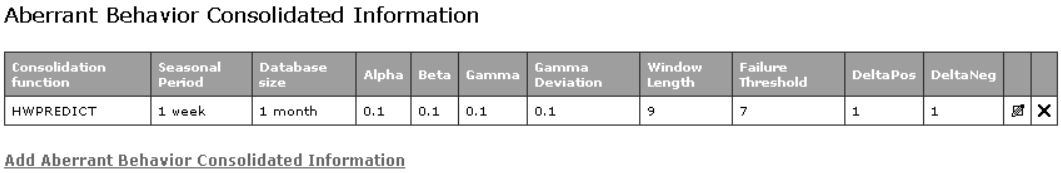
Consolidation function	Consolidation step	Database size	Edit	Remove
AVERAGE	5 minutes	1 day		
AVERAGE	30 minutes	1 week		
AVERAGE	2 hours	1 month		
AVERAGE	1 day	1 year		

Add Consolidated information

Fig. 7. Consolidation of the monitored data

The user of the tool can also define the consolidation of information for trend analysis, available through the Aberrant Behavior Detection Algorithm from RRDTool. More information in the configuration of the parameters of the algorithm are found in an article from Brutlag [14].

Figure 8 presents the standard configuration generated which analyzes the trend considering the period of one week. In these configuration the algorithm verify, for instance, if the actual traffic of the network for one day of the week is in agreement with the past history traffic observed in the previous weeks. These option came pre-configured for the user in the simple mode and can be altered in the expert mode.


Aberrant Behavior Consolidated Information

Consolidation function	Seasonal Period	Database size	Alpha	Beta	Gamma	Gamma Deviation	Window Length	Failure Threshold	DeltaPos	DeltaNeg		
HWPREDICT	1 week	1 month	0.1	0.1	0.1	0.1	9	7	1	1		

Add Aberrant Behavior Consolidated Information

Fig. 8. Consolidation to aberrant behavior detection

After the definition of the monitoration and consolidation of the information, the user can define and configure the graphics and algorithms to visualize and analyze the performance data, what is done through the analysis configuration module. These module is described in details in the next sub-section.

4.4 Analysis Configuration

The Analysis Configuration module is divided in three submodules: Analysis Definitions, Graphics and Algorithms. The purpose of the Analysis definition is to abstract the performance database, inside the analysis configurations, and two types of analysis definitions are provided: static and calculated. The static definitions are simply references to the monitored data in the performance database. The calculated data

are formulas, using the Reverse Polish Notation (RPN), which use the static definitions as parameters and variables. Figure 9 shows samples of some possible static and calculated analysis definitions.

Data Definitions | Graphs Configurations | Algorithms Configurations

Name	Type	Data Source	Edit	Remove
ifInOctets1	Static	Castor Monitoration - 200.19.246.2 - MIB Interface - ifInOctets1 (AVERAGE)		✕
ifOut1DEVP	Static	Castor Monitoration - 200.19.246.2 - MIB Interface - ifOutOctets1 (DEVPREDICT)		✕
ifOut1HWP	Static	Castor Monitoration - 200.19.246.2 - MIB Interface - ifOutOctets1 (HWPREDICT)		✕
ifOutOctets1	Static	Castor Monitoration - 200.19.246.2 - MIB Interface - ifOutOctets1 (AVERAGE)		✕
lowerBound	Calculated	ifOut1HWP,ifOut1DEVP,-		✕
upperBound	Calculated	ifOut1DEVP,ifOut1HWP,+		✕

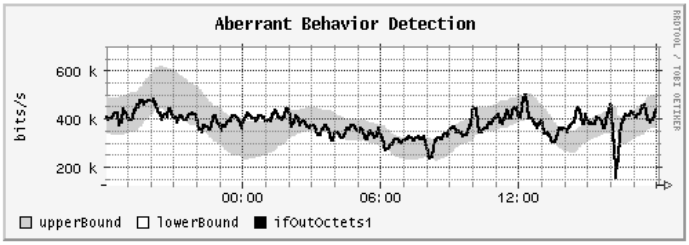
Add analysis definition: Static | Calculated

Fig. 9. Analysis definition data

Each graphic configured in the Analysis tool has a title, a name, a vertical label (unit of the monitored information), and a period of presentation, defined by a start and end date. The values for start and end date can use expressions like now (actual time/date), or more complex expressions like now-1day (last day), now-1month (last month), etc.

The analysis data definition are associated with the graphics and is allowed to configure the presentation order and the presentation format (stereotype). Figure 10 presents a screenshot of one graphic generated by the tool with some analysis definition associated.

Data Definitions | Graphs Configurations | Algorithms Configurations



Graph Parameters:

Order	Definition name	Stereotype	Visible	Color	Change order		
3	upperBound	AREA	hide	#D0D0D0	⬇		✕
4	lowerBound	AREA	hide		⬆ | ⬇		✕
5	ifOutOctets1	LINE2	hide	#000000	⬆		✕

Add Graph Parameter

Fig. 10. Visualization of a graphic dynamic configured

Figure 10 presents a real traffic data analysis generated through the Aberrant Behavior Detection (ABD) [14] algorithm of a university campus link in the Brazilian National Research Network. The ABD algorithm provides time series values prediction and is available only in the development version of RRDTool. The thick line is the observed value of the incoming traffic and the gray area are the confidence band. Whenever the thick line crosses outside the confidence band it suggests an abnormal behavior in relation to the expected behavior, indicating an increasing or decreasing traffic faster than the past history.

The tool does not implement now more advanced analysis algorithms, but all the tool organization and architecture forecast the possibility of add analysis extensions with more functionalities. Some

threshold analysis algorithms are under development. Figure 11 presents a sample of an analysis algorithm configuration.

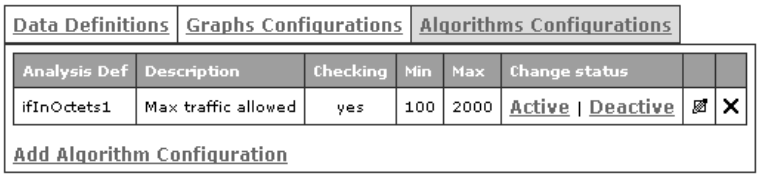


Fig. 11. Analysis algorithm configuration

5 Conclusions and Future Work

Technologies used in the Internet like HTML, HTTP and XML applied in network management tools are in evidence, many different solutions and applications are been proposed, mainly to the representation and access to the management information in replacement to the traditional SNMP and console based management applications. The use of XML documents to present the management information could be a standard in the next few years.

One of the biggest benefits of the tool presented in these article is the possibility of monitor and analyze any information, from various sources and types in an integrated and flexible fashion. These was achieved because the information supported is any information in the XML format. These results can help in the management of large heterogeneous networks where a great number of different information must be considerate in an integrated way.

As future work we highlight the improvement of the utilities in the user interface to turn the definition of the XPath filter expression and the Reverse Polish Notation (RPN) expressions, which give more flexibility to the tool and is likely to be easily defined by the users. This process could be enhanced using wizards to make the expression creation process transparent.

The incorporation of more advanced algorithms to automatic analysis of the time series data monitored and, probably allowing the co-relation of alarms is a good point to be taken in advance. The tool architecture forecast this possibility, but not implement at this moment such algorithms and do not provide any framework to anyone include more algorithms as plug-ins in the tool. The analysis algorithm configuration must also allow the configuration of tasks automatically fired when some event occur, like alarms generated by email when some threshold are crossed.

References

1. T. Oetiker. Rrdtool round robin database tool, 2003. http://www.rrdtool.org.
2. DMTF. Xml as a representation for management information - a white paper, 2001.
3. Hong-Taek Ju, Mi-Jung Choi, Sehee Han, Yunjung Oh, Jeong-Hyuk Yoon, Hyojin Lee, and J.W. Hong. An embedded web server architecture for xml-based network management, 2002. IEEE/IFIP NOMS2002.
4. M. Rose and K. McCloghrie. Structure and identification of management information for tcp/ip-based internets, 1990. RFC 1155, May.
5. T. Oetiker and Dave Rand. Multi router traffic grapher, 2003. http://www.mrtg.org.
6. Jeff R. Allen. Driving by the rear-view mirror: Managing a network with cricket, 1999. Proceedings of the 1st Conference on Network Administration.
7. T. Oetiker. Rrd world frontends, 2003. http://www.rrdtool.org.
8. IETF. Next generation structure of management information (sming) charter, 2002. http://www.ietf.org/html.charters/sming-charter.html.
9. R. Neisse, L. Granville, D. O. Ballve, M. J. B. Almeida, and L. M. R. Tarouco. A dynamic snmp to xml proxy, 2003. IEEE/IFIP IM2003.
10. F. Strauss. Libsmi a library to access smi mib information, 2002. http://www.ibr.cs.tu bs.de/projects/libsmi/.
11. Expat. The expat xml parser, 2002. http://expat.sourceforge.net.
12. Sax. The simple api for xml, 2002. http://www.saxproject.org.
13. James Clark and Steve DeRose. Xml path language (xpath) version 1.0 w3c recommendation, 1999.
14. Jake D. Brutlag. Aberrant behavior detection in time series for network monitoring, 2000. LISA2000.