Framework for Extensions to XML-based Network Management

Mohammed Houssaini Sqalli and Shaik Sirajuddin
Department of Computer Engineering
King Fahd University of Petroleum & Minerals
P.O. Box 1447, Dhahran 31261, Saudi Arabia
E-mail: {sqalli, siraj}@ccse.kfupm.edu.sa

Abstract

XML-based integrated network management architecture consists of an XML-based manager, an SNMP/XML gateway and SNMP agents. In this paper, we present a framework for extensions to an existing XML-based network management protocol, which can reduce the processing time between the XML-based manager and the SNMP agents. The extensions consist of new types of messages, including the Multi-Get-Request and Multi-Set-Request. These new types, for instance, allow a manager to send one or more requests to one or more agents bundled in one message. This framework decreases the overall traffic between the XML-based manager and the XML/SNMP gateway.

Keywords: XML-based Network Management, XML, SNMP, XPath, XQuery, SOAP, DOM, XSL, XSLT, DTD and XML Schema.

1 Introduction

Today's network has incompatible infrastructure including different information models, information access methods, and management protocols. The administrator has no choice but to use separate and incompatible management tools to manage the current heterogeneous network [1]. Currently available management tools and framework are based on a centralized approach and confronted with scalability problems when the network expands.

Managing the network components using web-based technology came into existence when the vendors began embedding Hyper Text Transport Protocol (HTTP) servers in their network equipment [2]. Many network equipment vendors, including Cisco, Nortel Networks and 3Com, now routinely embed HTTP servers in their new equipment. When Java applets appeared in Netscape's famous Web browser, in 1995, it introduced the concept of *embedded management application*, and has the advantages of using HTTP rather than Simple Network Management Protocol (SNMP) to vehicle data between managers and agents [2].

XML-based network management applies Extensible Markup Language (XML) technologies to network management. In XML-based network management the management information is defined using XML and the management data is exchanged in the form of an XML document and processed using the standard methods available for XML [3][4][5].

In this paper, we propose an extension to the existing XML-based network management where the manager can send more advanced requests to the agents via a gateway. A manager can, for instance, send one message that includes one request to multiple agents, multiple requests to one agent, or a combination of both.

The paper is organized as follows; first we will give a general background of the network management techniques, then we will address the limitations of the SNMP based network management. The current work on XML-based network management will then be presented. The section that follows will describe the proposed framework. The paper ends with a conclusion.

2 Background

2.1 SNMP-based Network Management Systems

The SNMP is the most widely used protocol to manage network devices on the Internet [1][6][7]. It was first standardized by the Internet Engineering Task Force (IETF) in 1990 [RFC 1157]. A number of Requests for Comments (RFCs) have been written to specify the different elements and versions of SNMP. SNMP uses a general manager and agent interaction model (Request/Response). It uses the Structure of Management Information (SMI) [RFC 1155, RFC 2578] to define managed objects. The SNMP Management Information Base (MIB) [RFC 1213] uses a hierarchal tree structure for organizing the MIB Object Identifiers (OIDs). The first version of SNMP is referred to as SNMPv1 [6][7]. SNMPv1 supports GET, SET, GET_NEXT and TRAP operation, and provides limited management capabilities. SNMPv1 has few limitations including the lack of security, lack of bulk data transfer capability, and lack of manager-to-manager communication.

These issues were addressed in SNMPv2 [RFC 3416] [6][7], which was initially proposed in 1995. SNMPv2 supports GET_BULK_REQUEST, and INFORM_REQUEST. The major changes in SNMPv2 are a manager-to-manager message, enhancements to SMI: SMIv2 [RFC 2578], textual conventions [RFC 2579], conformance statements [RFC 2580], row creation and deletion in tables [RFC 2579], MIB enhancements [RFC 3418], and transport mappings [RFC 3417]. One of the main limitations of SNMPv2 is security, which provides a community based mechanism that uses a plain text string for authentication and access control.

SNMPv3, introduced in 1999, undertook the issue of security including authentication, privacy and access control, as well as the definition of new architecture and framework for SNMP [RFC 3410-3415]. [RFC 3584] described the coexistence between SNMPv1, SNMPv2, and SNMPv3.

The SNMP framework is designed to minimize the number and complexity of management functions by the agents. This makes it extensible to accommodate additional and unanticipated aspects of network operations and management, and independent of implementations of a particular host or gateway [5] . Thus, SNMP provides simplicity, interoperability, and low footprint on agents [5][8]. SNMP has wide support of IP equipment vendors.

Top

2.2 XML-based Network management (XNM)

Extensible Markup Language (XML) is a Meta markup language, which was standardized by the World Wide Web Consortium (W3C) for document exchange in 1998 [9][10][11]. XML has many advantages. We can define our own Structure of Management Information in a flexible form using either Document Type Definition (DTD) or XML Schema [11][12][13][14] . XML documents can be transmitted on the Internet using HTTP. XML offers many free APIs for accessing and manipulating the XML data. XML separates the contents of a document and the expression methods, i.e. the management data is stored in XML documents and the presentation or format of the management data is stored in Extensible Style Sheet Language (XSL) documents using Extensible Style Sheet Transformations (XSLT) representation [10][11]. XML supports exchange of management data over all the hardware and software that supports HTTP. XML needs low development cost, since all the APIs and development kits are freely available. XML supports transfer of large amount of data in a single document. All these advantages of XML make it a candidate to solve the problems of scalability and efficiency of existing SNMP based NMS. Figure 1 shows the manager and agent combinations in XML-based network management [4].

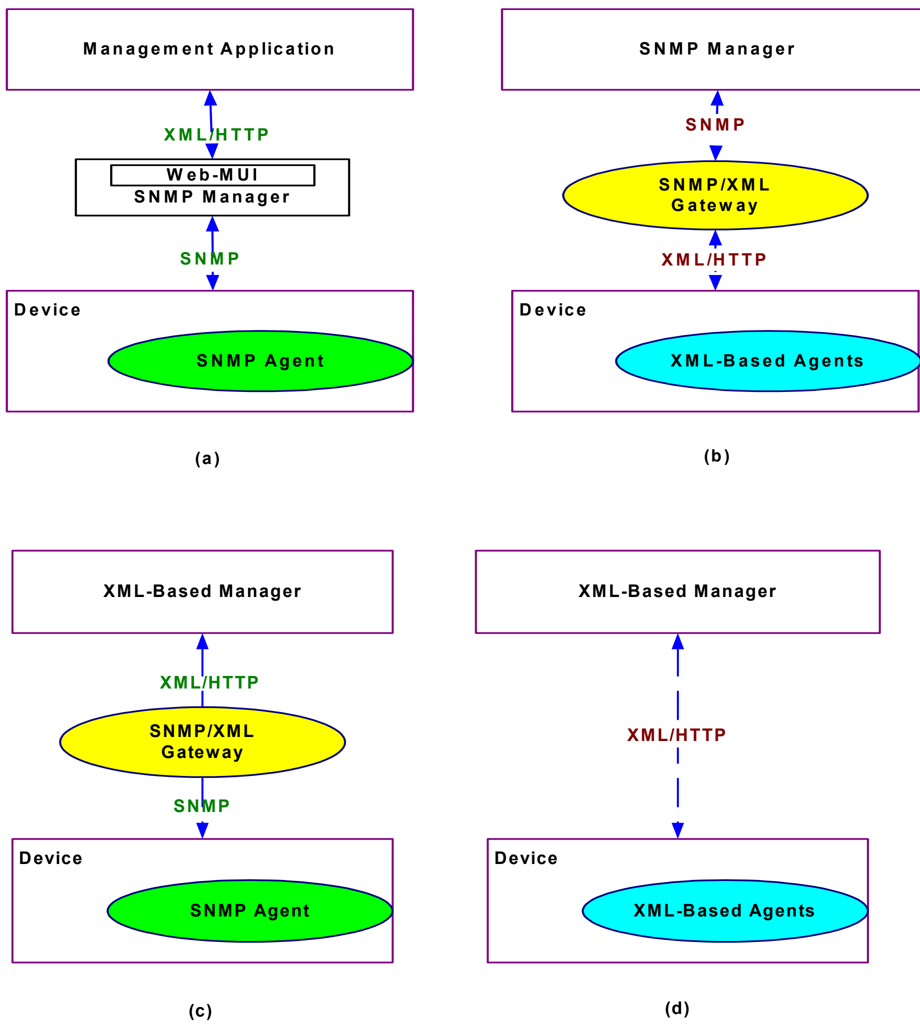

Figure 1: Manager and Agent Combinations in the XML-based Network Management

Top

Figure 1(a) shows the most widely used network management combination. Figure 1(d) is a total XML-based management combination, which is an ideal network management paradigm since there is no XML/SNMP gateway. It gives the maximum benefit compared to the other network management combinations. Figure 1(b) and Figure 1(c) show approaches that need translation from XML to SNMP through a gateway [3][4]. Since most network devices have legacy SNMP agents installed in them, the combination in Figure 1(d) is very difficult to implement in the current network environment. In order to do so, we need to deploy XML-based agents in the network devices. Figure 1(c) shows the most appropriate combination to implement in the current network management framework. This, however, requires the development of an SNMP/XML gateway to exchange the messages between the XML-based network manager and SNMP agents.

The XML schema [12][13][14] is machine readable and human readable. An XML schema document is basically an XML document. It offers many data types compared to DTD. XML schema provides modularity and offers greater control and flexibility than the DTD. It is complete and more complex than the DTD model. XML schemas are used to define the SMI and the constraints that MIB variables have to satisfy. SMI can be defined according to the user requirements. Below is the code in ASN.1 notation, which shows the OBJECT TYPE macro of the SNMP MIB [3].

```
NodeName OBJECT-TYPE
SYNTAX “SyntaxType”
ACCESS “AccessType”
STATUS “StatusType”
DESCRIPTION “DescriptionText”
REFERENCE “ReferenceType”
INDEX “IndexList”
DEFVAL “DefaultValue”
: : = {parentNodeName nodeNumber}
```

This macro is used to represent the table nodes or the data nodes of the MIB. [3] The equivalent conversion of the OBJECT TYPE macro expressed in XML schema is the following:

```
<xsd:element name = “NodeName”>
<xsd:complexType> <xsd:simpleContent>  <xsd:restriction base = “xsd:string”>
<xsd:sequence> (lower part node definition part)   </xsd:sequence>
<xsd:attribute name = “oid” type =“xsd:string” use = “fixed” value =“OidValue” />
<xsd:attribute name =“Access” type =“xsd:string”use = “fixed” value   =“AccessType”/>
 <xsd:attribute name = “Status” type =“xsd:string” use = “fixed” value =“StatusType” />
<xsd:attribute name = “Description”type = “xsd:string” use = “fixed”value = “DescriptionText” />
<xsd:attribute name = “Reference”type = “xsd:string” use = “fixed”value = “ReferenceType” />
<xsd:attribute name = “Index” type =“xsd:string” use = “fixed” value =“IndexList” />
<xsd:attribute name =“Defval” type =“xsd:string” use =“fixed” value =“DefaultValue” />
</xsd:restriction> </xsd:simpleContent> </xsd:complexType> </xsd:element>
```

XML Path Language (XPath) [11][15][16] is to address parts of an XML document. It also provides basic facilities for manipulation of strings, numbers, and Boolean. XPath gets its name from the use of a path notation as in URLs for navigating through the hierarchical structure of an XML document. XML Query language (XQuery)[11][16] [17] is designed to support all types of XML data sources like structured and semi structured documents, relational databases, and object repositories.

Top

3 Limitations of SNMP

3.1 Scalability

The most important drawback of the SNMP based Network Management is scalability to support a large network. The main factor is the network overhead. In SNMP based NMS networks, the overhead is the proportion of a link capacity used to transfer management data. The network overhead must represent a small percentage of the overall capacity of the link [5]. The network capacity must be utilized for user data transfer not for management data.

3.2 Capacity of Managers Local Segment

Capacity of the manger local segment is limited due to the centralization of management. Data received from all the agents is accumulated at one single point in the network segment where the NMS is connected. This becomes a bottleneck. In order to reduce the network overhead, efficiency of the communication protocol has to improve [5].

3.3 Processing Time

In SNMP-based network management, processing time is nothing but latency. It is between the issuing of the request for the MIB variable and the time it receives the response from the agent. The latency must be low. If it is very high then operational problems are detected very slowly and corrected lately. Latency can be divided into two types: end-host latency and network latency. End-host latency is due to the marshaling and unmarshaling of the data, compression and decomposition of the data, and security key computation. Network latency is the time spent in the network links and network equipments. It depends on the capacity of the error rates of the links, and on the speed of the routers traversed between the agent and the manager. The amount of data moved on the links has direct impact on the network latency [5].

3.4 Centralized Network Management

The SNMP-based Network Management architecture is based on a centralized management paradigm, where a central management system controls a large number of managed agents. This does not scale to support today's large network, and it will be very tedious when the managed network itself is used to communicate with the management agents (in-band management). There is a possibility of the network entering into a critical state [5].

4 Related Work on XML-based Network management

XML-based network management got fame when J.P. Martin-Flatin proposed SNMP MIB to XML translation models. He proposed two translation models, model-level mapping, and metamodel-level mapping [5]. In model-level mapping, a DTD is specific to a particular SNMP MIB. The XML elements and attributes in the DTD have the same name as the SNMP MIB. In metamodel-level mapping, a DTD is defined as one generic prototype and can be used by all the SNMP MIBs.

Today's Network is equipped with legacy SNMP based agents, and it is difficult to manage legacy SNMP agents through an XML-based manager. The interaction between the XML-based manager and SNMP-based agents is provided by converting the XML-based request to an

Top

SNMP-based request through an SNMP/XML gateway. This gateway is developed by POSTECH at their DPNM laboratory [3][4]. POSTECH also defined and implemented the SNMP MIB to XML translation algorithm in the SNMP/XML gateway. This gateway provides modules to manage networks equipped with SNMP agents [3]. The implementation of the gateway requires two types of translations: specification translations and interaction translations. The specification translation is concerned about the translation of the SNMP MIB to XML. POSTECH uses an automatic translation algorithm for SNMP MIB to XML. The interaction translation methods for XML/SNMP gateway are the process level interaction translation, the message level interaction translation, and the protocol level interaction translation.

The process level interaction translation method is also known as the Document Object Model (DOM)-based translation [11][18][19], where the mapping is done from the DOM interfaces to SNMP operations. The management information is translated from XML-based manager to SNMP operations by using the DOM structure and its interfaces. Here the DOM tree acts as an intermediate storage for management information. It has good performance when we have an internal gateway in the management system. The access to the DOM data can be done through the DOM API [11].

The message level interaction translation is also known as HTTP-based Translation. The XML/SNMP gateway translates URI-based HTTP requests from XML-based manager to SNMP requests. The URI-based requests are extended with XPath and XQuery to increase efficiency. The equivalent representations of the SNMP operation *get* and *set* in the HTTP-based interaction translations are shown below in Table 1 [3]. When the manager sends a request through the HTTP-based protocol, the message consists of 5 parameters: host name, operation type, community name, XPath, and value. The XPath field gives the name of the MIB node that the manager is interested to get [3]. XPath field is used to get the OID value for this MIB node.

SNMP-GET	http://(gateway_address)/[XPath-Expression]?host = agent_name & community = community_name &operation = get
Example: SNMP-GET	http://kfupm/ node [@name=”sysServices”?host = coe-siraj&community = public &operation = get
SNMP-SET	http:// (gateway_address)/ [XPath-Expression] ?host = agent_name & community = community_name &operation = set & XPath = node name&value = value_string
Example: SNMP-SET	http://kfupm/ node [@name=”sysServices”?host = 172.16.174.30 & community = public &operation = set & value = 72.

Table 1: HTTP-based interaction translation method and SNMP operations

The protocol level interaction translation is also known as SOAP-based translation. Simple Object Access Protocol (SOAP) is a protocol to exchange the XML messages over HTTP or SMTP. SOAP can be used to translate XML requests into SNMP requests. SOAP defined a standard method to transfer XML-encoded messages over HTTP. The “getRequest” or “setRequest” has a version element specifying the version of the SNMP, a “community” used for security identification, and “path” used to identify objects nodes in the DOM tree using the XPath expression. Table 2 gives the SOAP-based GetRequest, SetRequest and Response.

Top

Get Request	<m:getRequest xmlns:m=”http://example.org/xmlsnmp”> <m:community>public</ m:community><m:version>1</m:version> <m:path>// ifSpeed[1]</m:path> </m:getRequest>
Set Request	<m:setRequest xmlns:m=”http://example.org/xmlsnmp”> <m:community>media</m:community>><m:version>1</m:version> <m:path>//hostInfo/hostName</m:path><m:value>zeus</m:value> </m:setRequest>
Response	<m:response xmlns:m=”http://example.org/xmlsnmp” > <rpc:result xmlns:rpc=”http://www.w3.org/2001/12/soap-rpc”> <ifSpeed> 64000 </ifSpeed> </rpc:result><m:response>

Table 2: SOAP-based SNMP requests

JUNOscript [20] is introduced by the Juniper Networks for their JUNO network operating system. Juniper Network uses this JUNOScript in the XML-based network management, which is designed to minimize both implementation costs and the impact on the managed device. XML-RPC's are used to access operational and configuration data. The requests sent by the client are passed to JUNOScript server, which will parse the request and sends it to the corresponding module, and return the result to the client application.

In this paper, we propose to extend the work of POSTECH & Juniper Networks. The framework we describe in this paper allows a manager to send requests to multiple agents using a single message. We define new types of messages that could be sent by a manager, namely Multi-Get-Request, Multi-Set-Request, and Response. These messages can be widely used in configuration management. The implementation for both Multi-Get-Request and Multi-Set-Request can be achieved through an HTTP-based interaction method and a SOAP-based interaction method. We describe how a manager can send in one message either one request to multiple agents, multiple requests to one agent, or multiple requests to multiple agents at the same time.

5 Framework for Extensions to XML-based Network Management

5.1 Motivation

The main drawback of the SNMP gateway is lack of scalability and inefficiency of processing the management data from the agents. We propose a framework to increase the efficiency of processing management data, decrease the communication cost and reduce the traffic between the XML-based manager and the XML/SNMP gateway. It takes advantage of the XML, DOM, and Java servlets.

In this framework, the XML-based manager can bundle in one message one or more SNMP requests sent to one or more agents. This type of messages will be useful when we want to issue the same request to many agents or multiple requests to the same agent such as a Get-Request followed by a Set-Request.

An SNMP Get-Request operation gets the value of one MIB object from one agent at a time. If we want to get the same MIB value for n different agents then we need to execute the Get-Request operation n times. The SNMP Get-Bulk-Request operation can get the values of multiple MIB objects by traversing sequentially a MIB subtree of one agent. But, Get-Bulk-Request does not provide a way to get the data from different agents in a single request. We propose a

Top

procedure to get data from multiple agents at the same time. Similarly one can set the same MIB value in different agents at the same time.

A manager may also be required to get a MIB object value from different agents that satisfy some conditions. For instance, when a manager is interested in initializing the same value to n different agents, it needs to execute n different SNMP set operations. This will increase the traffic between the XML-based network manager and the agents. In the framework proposed, the SNMP/XML gateway will check the conditions requested by the manager and sends back only relevant information.

A manager may be required to set a MIB object value after checking some conditions. In this case, it may need to first get the MIB value using an SNMP Get-Request operation then issue an SNMP Set-Request operation. In this paper, we define one message type that bundles multiple SNMP requests. This message type will reduce the traffic between the XML-based manager and the SNMP/XML gateway. This will increase the efficiency of the XML-based manager.

This message type will be useful when we have more than one SNMP operations for one task. For instance, in the SNMP row creation, a manager has to use the first SNMP get operation to find the latest row index, then it has to use the SNMP set operation to create the new row. In the current SNMP paradigm, this type of row creation first involves an SNMP Get-Request operation then an SNMP Set-Request operation, thus there will be more traffic and it is time consuming.

5.2 Architecture

Our framework is based on the XML/SNMP gateway architecture, which was shown in Figure 1(c), communication is between an XML-based Manager, XML/SNMP Gateway, and SNMP Agents. Figure 2shows the actual communication between the XML-based Manager, XML/SNMP gateway and SNMP Agents.

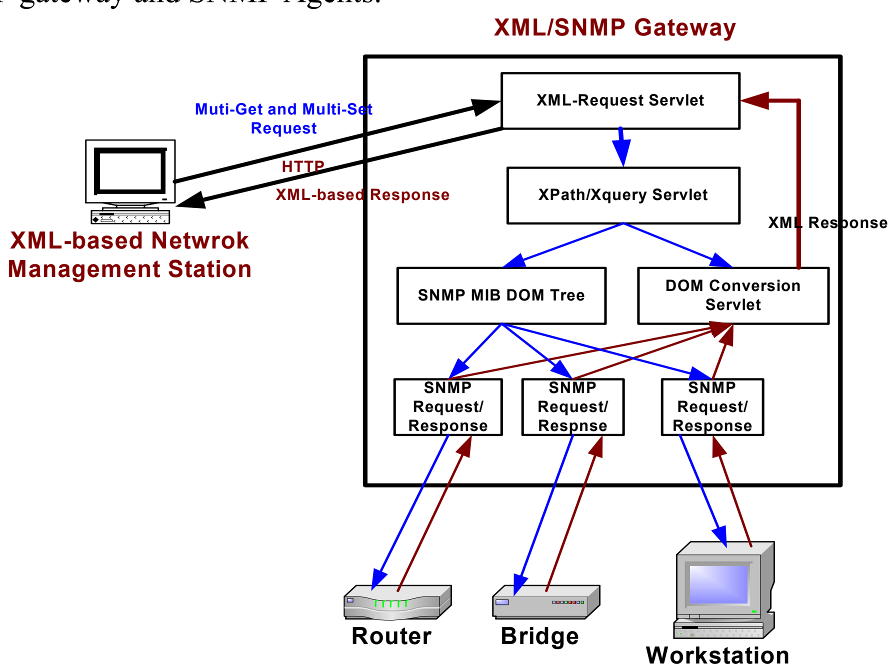

Figure 2: Framework to Implement Extensions of the XML-based Network Management

Top

The XML-based manager sends an SNMP request in the form of an XML document over HTTP protocol to the SNMP/XML gateway. This request is received by the XML-Request servlet, which parses the request and forwards this to the XPath/XQuery servlet, where the XPath and XQuery parts are separated. The MIB nodes of the Multi-Get-Request or Multi-Set-Request are represented as the hierarchical DOM tree. The XPath part is used to find the MIB node part in the SNMP MIB tree. The SNMP MIB DOM tree representations will provide the OID value of the requested MIB node, which is used to send the SNMP-based request to multiple agents by the SNMP Request/Response servlet. The response received from the SNMP agents are passed to SNMP Request/Response servlet. The responses are grouped at the DOM tree created for the multi agents and the grouped data is converted to XML data and the file is passed to the XML-Request servlet. The XML data is passed to the XML-based manager, where we will apply XSL sheet to transform the data to a desired form.

This framework will provide many advantages. It will decrease the traffic between the XML-based manager and the SNMP/XML gateway. The processing time to process the requests and the overall length of the requests will also decrease. It will provide a way to access multiple agents and also send multiple requests in a single message.

5.3 Extensions to Existing XML-based Network Management

5.3.1 Manager Sending One Request to Multiple Agents

In this case, the request coming from the manager is addressed to multiple agents. The request is passed to the request handler module, which is a Java Servlet running at the server side. The servlet module parses the XML request, takes the XPath part of the request, and finds the MIB node referenced. Then a DOM tree is created for representing the MIB node as the child node of the parent host. We present here an example of the manager sending one request to multiple agents through an HTTP-based protocol. The hierarchical DOM tree representation of the Manager sending one Request to Multiple Agents is shown in Figure 3 for this example. The example hierarchical DOM tree representation of the manager sending one request to multiple agents with data is shown in Figure 4.

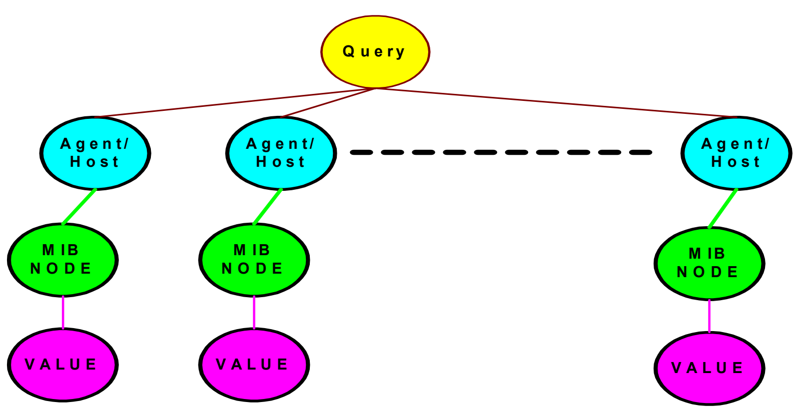

Figure 3: Hierarchical DOM tree of the Manager Sending one Request to Multiple Agents

Top

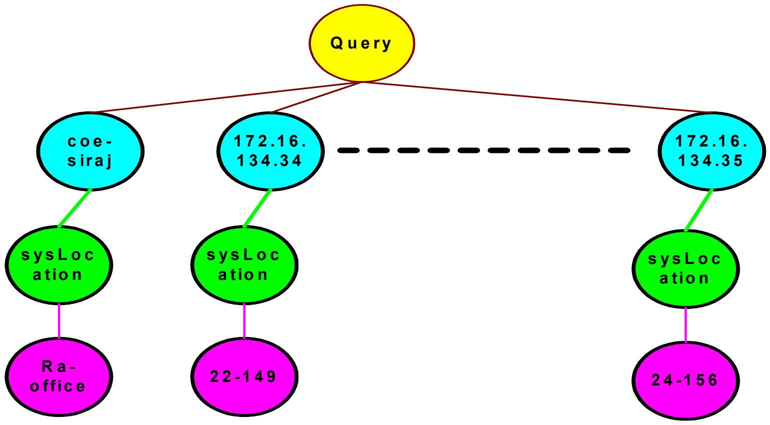

Figure 4: Example of Manager Sending Single Request to Multiple Agents

HTTP-based Approach: In this approach the message exchange between the XML/SNMP gateways will use the HTTP protocol. The Multi-Get-Request consists of "community", "XPath", and "Version" as the common tags for each agent. Then, it includes a number of agents' addresses. Table 3 gives the Multi-Get-Request operation with XPath and XQuery.

Multi-Get-Request With XPath	http://(gateway_address)/SNMP?& community = community_name &operation = Multi-Get &XPath = node_name, &host = agent_name &host = agent_name &host = agent_name& host = agent_name………………
Example-Multi-Get-Request with XPath	http://kfupm.edu.sa/kfupm-snmp-multiget?& community= community_name &operation = Multi-Get &XPath = "sysLocation", &host = coe-siraj & host = 172.16.134.24 & host = 172.16.134.35.
Example Multi-Get-Request With XQuery	http://kfupm.edu.sa/kfupm-snmp-multi-get-XQuery? &Question= <XQuery> <query><XPath> //sysLocation</XPath> <version> 1</version> <community> public</community> </query> <agent-list> <agent> coe-siraj </agent> <agent>172.16.134.24 </agent>172.16.134.35 <agent> </agent> </agent-list> </XQuery>

Table 3: HTTP-based Multi-Get Request

SOAP-based Approach: In this approach we will use the SOAP-based protocol over the HTTP protocol to exchange XML-based requests. The Multi-Get-Request consists of "community", "XPath", and "version" as the common tags for each agent. Then we have a list of agents' addresses. Table 4 gives the Multi-Get-Request and Multi-Set-Request for XQuery. These requests utilize the XPath inside the XQuery.

Multi-Get-Request	<m:multi-get-Request xmlns:m="http://kfupm.edu.sag/snmp-multi-get > <m:community>public</ m:community> <m:version> 1 </m:version> <m:XPath> // sysLocation </m:XPath> </m:agent-list> <m:agent> coe-siraj </m:agent> <m:agent> 172.16.134.34 </m:agent> <m:agent> 172.16.134.25 </m:agent> </m:agent-list> </m:multi-get-Request>
Multi-Set-Request	<m:multi-set-Request xmlns:m=" http://kfupm.edu.sa/snmp-multi-set "><m:community>media</m:community> <m:version> 1 </m:version> <m:XPath> //</m:XPath><m:value></m:value> </m:multi-set-Request>

Table 4: SOAP-Based Multi-Get and Multi-Set Request

Top

5.3.2 Manager Sending Multiple Requests to One Agent

The XML based manager sends one request, which consists of different SNMP operations. This request is passed to the XML-Request servlet, which parses the request and forwards it to XPath/XQuery servlet, where the path and query are separated. Then a DOM tree is created. Figure 5 shows the DOM tree representation for the request received from the XML-based manager, which represents the MIB nodes in a hierarchical manner.

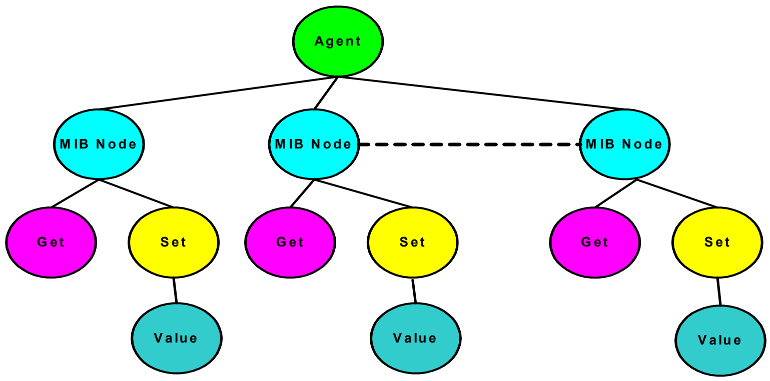

Figure 5: Hierarchical representation of the Multiple Request to one agent

5.3.3 Manager Sending Multiple Requests to Multiple Agents

The XML-based manager sends one request, which consists of different SNMP operations. This request is passed to the XML-Request servlet, which parses the request and forwards it to XPath/XQuery servlet, where the path and query is separated. Then a DOM tree is created. Presently only a get operation followed by a set operation has been identified. Figure 6 shows the manager sending multiple requests to multiple agents.

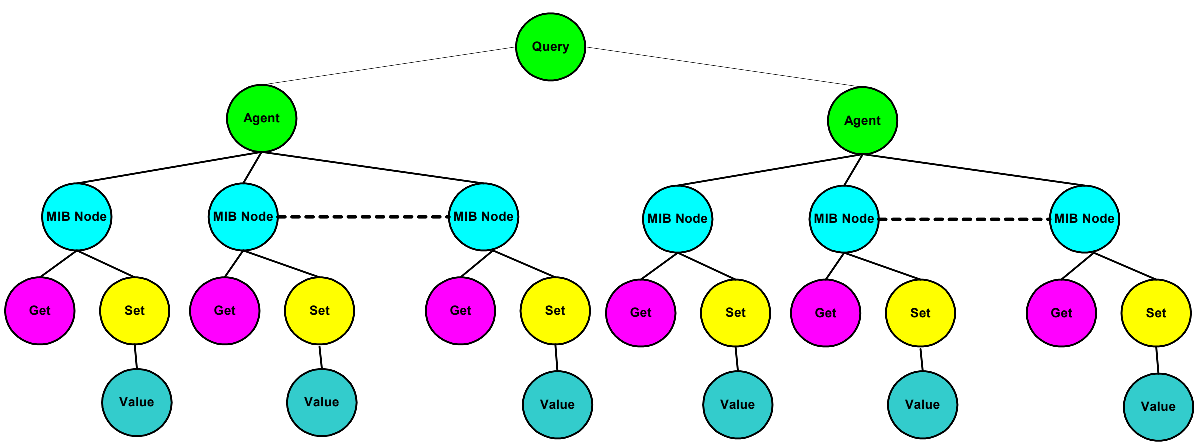

Figure 6: Hierarchical representation of the Manager sending Multiple Requests to Multiple Agents

Top

5.3.4 Other Extensions

The communication between the XML-based manager and RMON is similar to the communication between the XML-based manager and SNMP agents, since the RMON is going to be an agent for the top-level manager. Requesting multiple RMON devices to do the same type of monitoring and sending statistics back to the manager can be supported by this framework.

The gateway is going to receive many traps/alarms from the agents but the manager could request the gateway to send only the summary of the alarms by filtering the related ones or sending those that satisfy certain conditions. Thus the gateway can also act as a filter.

5.4 Applications

Configuration Management [5][6][21] is concerned with setting and getting the parameters of managed devices. With the multi-get-request and multi-set-request we can get and set many objects on many agents at the same time. Thus, this proposed framework increases the efficiency of the processing, and thus the efficiency of the configuration management process. The new extensions can be used, for instance, to set an alarm threshold value in multiple agents or to find the location (i.e., sysLocation) of n agents at the same time. It will also be useful when the manager is interested in initializing many agents with the same value. This approach will give us a way to download or upload a part of the configuration data and the configuration data is in textual format, hence wide range of text processing tools can be used format the data. The above mentioned are the requirement of the future Internet management [8].

Fault Management [5][6] is concerned with the detection and isolation of the problems that cause failures in the network. This gateway can be used to isolate minor and major alarms. The gateway can also be used to correlate different alarms and report to the manager a summary of the status of a sub network.

5.5 Future Work

Our objective is to evaluate the response time, network traffic and message length of the multi request. These results will be compared with a centralized NMS having gateways which can send a single request at a time to the agents connected to them. It will also be compared with the centralized management approach, where agents are connected to the centralized management station. In addition, we plan to evaluate the effect of this approach on the scalability and the efficiency of the NMS.

The evaluation of the response time, network traffic and message length of the multi-get request will be performed as follows by varying different parameter:

- Varying the number of agents present in the multi request.
- Varying the number of MIB objects present in the multi request.
- Varying the number of gateways connected to the XML-based manager.

Top

6 Conclusion

In this paper, we have defined and described a framework for extensions to an existing XML-based network management protocol. The framework includes types of messages that can be used in various forms to access the agents. We presented how one manager can send the same request to multiple agents and how it can send multiple SNMP requests in the same bundled message to an agent. A manager can also send multiple requests to multiple agents in the same message. We have shown how these types of messages can be used to increase the efficiency of the XML-based network management. This framework can be used in configuration management and fault management.

Acknowledgments

The authors acknowledge the support of King Fahd University of Petroleum Minerals (KFUPM) in the development of this work. This material is based in part on work supported by a KFUPM research project under Fast Track Grant No. FT/2004-20. This work has also benefited from the reviewers' comments.

References

[1] James Won-Ki Hong, Ji-Young Kong, Tae-Hyoung Yun, Jong-Seo Kim, Jong-Tae Park, and Jong-Wook Baek, "Web-based Intranet services and network management", Communications Magazine, IEEE, Volume: 35 Issue: 10, Oct. 1997, Page(s): 100–110.

[2] Martin-Flatin, J., "Push vs. pull in web-based network management". In Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management (IM'99), pages 3-18, USA.

[3] Jeong-Hyuk Yoon, Hong-Taek Ju and James W. Hong, "Development of SNMP-XML translator and Gateway for XML-based integrated network management", International journal of Network Management, 2003, 259-276.

[4] Mi-Jung Choi, James W. Hong, and Hong-Taek Ju, "XML-Based Network Management for IP Networks", ETRI Journal, Volume 25, November 6, 2003.

[5] Jean-Philippe Martin-Flatin, "Web-Based Management of IP Networks and Systems", Wiley series in communications Networking and Distributed Systems, 2003.

[6] William Stallings, "SNMP, SNMPV2, SNMPv3, and RMON 1 and 2", 3rd Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999.

[7] Mani Subramanian, "Network Management: Principles and Practice", Addison-Wesley, Hardcover, Published December 1999, 644 pages, ISBN 0201357429.

Top

[8] J.P Martin- Flatin, Aiko Pras, and Jurgen Schonwalder, “On the Future of Internet Management Technologies”, IEEE Commmunication Magzine, October 2003.

[9] W3C, “Extensible Markup Language (XML) 1.0”, W3C Recommendation, October 2000.

[10] Brett McLaughlin, “Java and XML”, O’Reilly, Second Edition, 2001.

[11] “Professional Java XML”, Wrox Publications, 2001.

[12] W3C, “XML Schema Part0: Primer”, W3C Recommendation, May 2001.

[13] W3C, “XML Schema Part1: Structures”, W3C Recommendation, May 2001.

[14] W3C, “XML Schema Part2: Data Types”, W3C Recommendation, May 2001.

[15] W3C, “XML Path Language (XPath) Version2.0”, W3C Working Draft, April 2002.

[16] W3C, “XQuery and XPath Functions and Operators Version2.0”, W3C Working Draft, April 2002.

[17] W3C, “XML Query Language (XQuery)”, W3C Working Draft, April 2002.

[18] W3C, “Document Object Model (DOM) Level 2 Core Specification”, W3C Recommendation, November 2000.

[19] W3C, “Document Object Model (DOM) Level 2 Traversal and Range Specification”, W3C Recommendation, November 2000.

[20] Phil Shafer “XML-Based Network Management” – White Paper, Juniper Networks, Inc., 2001, http://www.Juniper.net/solutions/literature/white_papers/200017.pdf

[21] Siu, J.B., and Zhen Sheng Guo, “Web-based network configuration management system**”,** Communication Technology Proceedings, 2000. WCC - ICCT 2000.

Appendix

DOM: Document Object Model
DTD: Document Type Definitions
HTTP: Hyper Text Transport Protocol
MIB: Management Information Base
NMS: Network Management Stations
OID: Object Identifier’s
RFC: Request for Comments
SNMP: Simple Network Management Protocol
SOAP: Simple Object Access Protocol
XML: Extended Markup Language
XPath: XML Path Language
XQuery: XML Query language
XSL: Extensible Style Language
XSLT: Extensible Style Language Transformations

Top