An evaluation of network management protocols

Pedro Gonçalves[1], José Luís Oliveira[2], and Rui L. Aguiar[2]

[1]University of Aveiro , ESTGA/IT,
[2]University of Aveiro, DETI/IT,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
{pasg, jlo, ruilaa}@ua.pt

***Abstract*— During the last decade several network management solutions have been proposed or extended to cope with the growing complexity of networks, systems and services. Architectures, protocols, and information models have been proposed as a way to better respond to the new and different demands of global networks. However this offer also leads to a growing complexity of management solutions and to an increase in systems' requirements. The current management landscape is populated with a multiplicity of protocols, initially developed as an answer to different requirements.**

This paper presents a comparative study of currently common management protocols in All-IP networks: SNMP, COPS, Diameter, CIM/XML over HTTP and CIM/XML over SOAP. This assessment was focused on wireless aspect issues, and as such includes measures of bandwidth, packets, round-trip delays, and agents' requirements. We also analyzed the advantages of compression in these protocols.

***Index Terms*— Network management, WBEM, Web services, SNMP, COPS, Diameter**

I. Introduction

SINCE the introduction of SNMP [1] in the early nineties, a dramatic and continuous change in the services offered by IP networks has also led to a rising demand for new network and system management paradigms to tackle this evolution. During this period, new management architectures (IETF policy framework, WBEM, Web Services, …), new information models (PIB, CIM, SID, …) and new protocols (COPS, Diameter, CIM-XML over HTTP, SOAP, NETCONF, …) have been proposed with the most varied (and sometimes outrageous) claims. However, despite all this panoply of offers, each one typically brings increasingly complex solutions with significant computational and bandwidth requirements. The major concern of building and deploying a management solution is related to the achievement of functional requirements and, most of the time, performance has not been an important and discriminating issue. However, some proposals can have a great impact on the efficiency of installed network equipment and link usage.

The 3GPP consortium has been developing a Next Generation Network architecture (NGN) comprised of a functional entity set, as well as a management architecture for network management [2]. They have been defining the entities and protocols for implementation of management platform interfaces. In this proposal, core entities exchange data through binary protocols like COPS [3] and Diameter [4] while management and application entities support their communication with textual XML-based protocols.

The NGN environment will be much more complex than this preliminary 3GPP blueprint. Our particular interest is the challenges posed by environments like wireless community networks (such as those developed in the EU-funded SWIFT project [5]). In this environment, links are usually wireless, and nodes can be both users and routers (often both – with low cost wireless access routers), establishing wireless mesh networks for service delivery.

The impact of management/control protocols on NGN can then be significant, if one considers either the amount of terminals, services and end-users, or the diversity of infrastructure that will build this NGN. Thus, selecting a management protocol for such a global environment becomes a daunting task. In this case, the efficiency of the management protocol in terms of both network overhead and device load becomes important.

Unfortunately, past research lacks the present analytical, simulated or real-time measurements of this impact. Most new management technologies have been evaluated, but, concerning performance, the assessment has been made typically against SNMP [6-9]. Those results show, for instance, that the messaging requirements of the web-based protocols are greater than SNMP requirements [10]. Broader evaluation studies are clearly missing, including protocols like COPS, WBEM technology, or Diameter.

In this paper, we present a comparative study of the signaling generated by several management protocols (SNMP, COPS, Diameter, CIM/XML over HTTP and CIM/XML over SOAP). We also present a study of the effect that compression has on the signaling volume of management entity communication. These results should be useful for selection of a management solution for environments like the above mentioned wireless mesh networks, or IMS-based NGN.

This paper is organized as follows. Section II gives a general overview of several management technologies selected, section III presents the methodology used and section IV the result of the evaluation conducted. Section V discusses the results that emerged from this assessment, and section VI concludes the paper.

II. Background to the technologies

The current management landscape is populated with a multiplicity of protocols, initially developed as an answer to different requirements. We selected some of the most relevant

978-1-4244-3487-9/09/$25.00

technologies nowadays, all potential candidates as management frameworks for NGN.

The Simple Network Management Protocol was proposed in 1990 as a simple application layer protocol that implements communication between a management console and the managed agents. SNMP implements 5 messages: messages for information request (GET, GETNEXT, GETBULK), one message for information writing (SET) and a message for event notification (TRAP). The protocol messages are coded in small size packets and transported by UDP - in order to allow a lightweight message transport in overloaded networks. Version two of the protocol was proposed in RFC 1441-RFC 1452 with some enhancements to the SNMPv1 data types and with the GETBULK message. Version 3 [11-15] was proposed in 1998 with some enhancements in terms of security and remote configuration. SNMP protocol is widely used today in network management as well as in the equipment management areas, mainly as an equipment monitoring tool.

The Common Open Policy Service Protocol (COPS) [3] was proposed by the IETF as a query / response protocol for policy information exchange. COPS is a binary protocol that transports messages between the COPS manager designated the Policy Definition Point (PDP) and its managed entities – the Policy Enforcement Points (PEPs) using TCP. Client and server maintain a COPS connection and they identify all the messages using a unique handling. Two models of the protocol were proposed: the outsourcing - COPS-RSVP [16] and the provision model - COPS for Policy Provisioning (COPS-PR) [17].

Under the outsourcing model, external events in the Policy Enforcement Point (PEP) must be handled by the Policy Decision Point (PDP) in a pure query/response manner. The protocol is typically implemented between the router (PEP) and its manager (PDP) for admission control purposes. The router sends COPS requests (REQ messages) to the server when it receives network access requests from the network clients, and the server, after taking a decision about the access request, answers with decision (DEC) messages.

Under the provision model [17] COPS implements event notification (REQ) messages from clients to the server, and configuration (DEC) messages from the server to the client. The PDP performs prior configuration of the PEPs that will operate according to pre-defined policies. The COPS client sends REQ messages to the server when it receives a user request or when an internal event is generated in the client. The server sends unsolicited DEC messages to the client, although the model does not make any assumption about the correlation between REQ and DEC messages.

All COPS messages have an 8-byte-long packet header. A typical COPS-REQ message is 24 bytes long and a DEC message has the size of the decision object plus 32 bytes. The COPS protocol did not initially gain much attention from the industry since it was not considered a significant evolution from the SNMP protocol, despite bringing a significant conceptual change.

Then again, the Diameter protocol was proposed within the Authentication, Authorization and Accounting (AAA) framework [4] as the successor for the RADIUS AAA protocol. The Diameter Base Protocol is the core model and several extensions tailored for specific applications were also proposed, such as the Diameter Network Access Server Application (NASREQ), the Diameter mobile IPv4 Application (MobileIP) [18] and the Diameter Session Initiation Protocol [19].

More recently, Diameter gained a preeminent position since it was widely adopted inside the 3GPP IMS platform [20] as the communication protocol for several functional entities. Diameter was proposed as the protocol for the AAA entity communication [21, 22], and it was also proposed as the messaging solution for the QoS negotiation issues [23, 24]. In March 2008, the Diameter Policy Processing Application was published, an extension of Diameter as a policy management protocol, enlarging the original AAA target [25].

Another separate effort, the Web Based Enterprise Management (WBEM) [26] initiative, was born in 1996 with sponsorship from several companies. The goal was to unify desktop management with network management and to create a multi-vendor and multi-platform management framework. WBEM technology consists of three base concepts: it represents the management data in Common Information Model (CIM) [27], it encodes the management information in eXtensible Markup Language (XML) [28] and it transports the management information over HTTP [29]. CIM is an object-oriented model that allows representation of management information, as well as the relationships between management entities.

WBEM solutions include four components: the CIM client typically used by the human operator during management tasks, a CIM Object Manager (CIMOM) that is the main component of the system maintaining the dialogue with the CIM client and the management information, a CIM repository and the CIM providers that perform the interface between the CIM server and specific managed equipment such as a managed server or a router. The definition of a new CIM extension also involves the development of the correspondent CIM provider that will implement the functional logic of the defined objects (configuring, monitoring, …). Providers can be classified as instance (for object representation), indication (notifications), association (to define objects' relations) or method (applied for the invocation of remote methods). The integration of underlying management technologies in WBEM is implemented through specific providers, and adaptors for common protocols are already available [30, 31].

WBEM technology is used mainly in the area of desktop management. Several open-source and commercial implementations based on WBEM technology exist [32-35]. Typically each of the companies that sponsor the open-source project commercializes its own WBEM-based management product.

Web Services (WS) is a very popular distributed systems technology nowadays. It is XML technology based on W3 standards like the Simple Object Access Protocol (SOAP) [36] and the Web Service Definition Language (WSDL) [37] and is supported in many platforms and by many vendors. Web services commonly communicate through the exchange of SOAP messages typically transported in HTTP or SMTP protocol. They offer significant interoperability because of the XML description tags used in the object encoding and they are language independent as well as platform independent. A Web service is described in a WSDL document, which defines the operations and its parameters, the service location, and the implemented protocols for each operation.

Concerning management, two Web service initiatives were promoted: Web Services for Management (WS-Management) [38], specified by the Desktop Management Task Force (DMTF), and the Management Using Web services (MUWS) [39], proposed by the Organization for the Advance of Structured Information Standards (OASIS). The MUWS standard was initially developed by Hewlett Packard and was standardized inside the OASIS. WS-Management is a specification initially developed by a commercial consortium and released in April 2006 by the DMTF. MUWS offers a richer solution to manage distributed systems. However, due to a simpler set of operations and a more lightweight specification, the WS-Management obtains better performance results [10].

Several WS–based management implementations exist: the MUSE [40], that follows the MUWS; and tightly coupled to the DMTF Web services specification the Wiseman [41] and Openwsman [42].

NETCONF [43] is an XML-based protocol for the management and configuration of network elements, where managers retrieve, edit, copy and remove device configuration. The protocol communication is based on Remote Procedure Call technology and it follows the traditional client/server model. NETCONF messages can be transported over several protocols like TCP, SSH, BEEP or SOAP, which has been receiving most attention lately.

NETCONF solution was discarded during the tests since it shares the characteristics of the XML-based protocols, especially with the Web services technologies.

III. EXPERIMENTAL FRAMEWORK

Experimental evaluation of the network management protocols was performed within the same scenario, using a set of applications that are representative of the technologies studied. The scenario selected was based on the configuration of QoS management entities from an IMS platform [2]. The application clients represented the Policy and Charging Resource Function (PCRF) [44] requesting configuration information from the Element Management Layer [45]. The protocol evaluation procedure consisted of a set of tests that measured the impact of management on the network traffic, the number of packets needed to transport the messages, the memory necessary in the management entities and the elapsed time needed by each pair of entities to exchange the information.

A. Test scenario

The test scenario included a manager-agent (client-server) pair placed in two separated systems connected by fast Ethernet. The network segment was isolated from the local LAN in order to avoid any traffic to bias the evaluation results. The management server was installed in a 1.7 GHz Intel Centrino with 1GB RAM, and the client was installed in a 1.8 GHz Intel Dual core with 512 MB RAM, both running 2.6.22 Linux Kernel.

During the tests, the agent requests configuration information from the manager, which responds with a set of objects described in Table 1. This kind of interaction follows the admission control messages that can be exchanged in a QoS-enabled network [46]. Depending on network size, configuration and management policy, a variable number of objects are placed in a local database before the server startup. When the server starts, the configuration update request for each managed object starts too.

Table 1 - Interface object structure

ELEMENT	SIZE (BYTES)	Description
InterfaceID	2	Primary key
Address	4	IPv6 address
Bandwidth	2	Total bandwidth
Uplink	2	Uplink / downlink interface
NetworkID	2	The interface network
Prefix	2	IPv6 prefix size
PrefixADDR	4	IPv6 prefix address
BWunit	2	Bandwidth unit
IfIndex	2	Interface index internal to router

During the tests, the traffic exchanged between the two end-point entities was captured and processed to obtain the number of packets, the number of bytes, the protocol header sizes and the message request size. During the communication between the entities, the server was also monitored in order to measure the memory allocated to the server process. These tests were repeated for each of the management protocols under evaluation.

For each technology an adequate pair of applications was selected, a client and a server.

- We made use of a Diameter API that implements the RFC 3588 for implementation of our Diameter applications.
- For COPS, we used the COPS++ API [47] that implements the RFC 2748.
- An SNMP agent was developed upon the Agent++ open source project [48], and the SNMP manager was a Net-SNMP [49] client application.
- The WBEM environment was deployed with the OpenWBEM framework [32]. To represent the Interface objects, several CIM extensions [50] were developed as well as an instance provider to deal with these CIM instances.

- The Web services scenario was constructed through the Openwsman solution [42]. Openwsman is based on an existing CIMOM, avoiding the development of an independent CIM Server. Therefore the Openwsman server acts internally as a WBEM client, as illustrated in Figure 1.

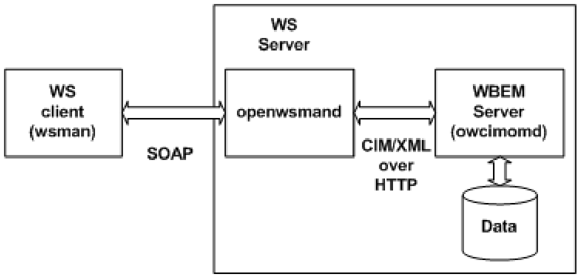

Figure 1- WS-Management test scenario

In Table 2 we summarize the most relevant characteristics of the implementations used in the tests.

Table 2 – Summary of the tested implementation characteristics

TECH.	IMPLEM.	PROTOCOL. VERSION	REQ. SIZE	DEC. HEADER	OBJ SIZE
SNMP	Based on Agent++ [48]	RFC 1157	48	25	220
COPS	Based on COPS++ [47]	RFC 2748	60	32	56
Diameter	Based on Diameter API	RFC 3588 RFC 5224	149	80	96
WBEM	OpenWBEM	[28, 29]	790	574	912
WSMAN	Openwsman	[38]	1291	1012	455

For the compression tests we made use of the *zlib* compression library and the packet flow was gathered through a common open source packet analyzer.

IV. RESULTS

For each scenario under study we have generated suitable management interactions to exchange between 1 and 10000 objects. Also, based on each protocol definition a basic analytical simulator was developed that allowed us to predict the signaling information and the number of packets in each test. The simulation values were used to validate the values obtained during the tests.

A. Managed objects

As expected, the information exchange between the applications increased linearly when we increased the number of objects for all the technologies, as plotted in the diagram in Figure 2, although with different slopes.

The binary protocols showed a much better performance than the XML based protocols, especially COPS. The binary protocols advantage has to do with the absence of tags as well as with efficiency in terms of information encoding. For instance, binary protocols encode the IPv6 Address in 4 bytes, and the XML-based protocols use 39 bytes for the address encoding.

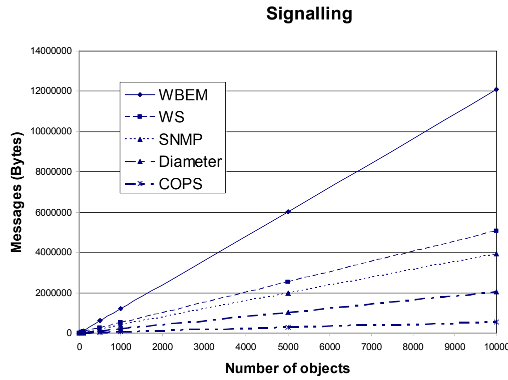

Figure 2 - Signaling test results

After packet inspection it was found that SNMP implementation does not make use of the complete transport packet payload size. This is one of the reasons for its performance. By not using the complete payload the total number of packets increases and thus the transport overhead also increases. During the 10000 object experiments, about 2Mb of Diameter signaling and 4Mb of SNMP signaling was measured. In terms of the number of captured packets we captured in the same experiments about 2000 Diameter packets and 18000 SNMP packets. Using our analytical simulator we expected a performance of the SNMP implementation 40% better than we actually verified during the tests. This is an issue that can be improved with a different SNMP implementation.

The Web services solution showed a remarkable performance compared to WBEM in terms of signaling, especially considering that Web services maintain the XML encoding and the XML interoperability advantages. Moreover, Web services wastes less than 50% of WBEM signaling. After an individual packet analysis we observed the rationale of this gain in performance: the difference has to do with a more efficient object encoding done by the Web services technology.

The WBEM solution is extremely verbose, as can be observed in Figure 3. For each object property, it repeats the property definition with the tags indicating where the name of the property, the property data type and its value are defined. Furthermore, the tags PROPERTY NAME, TYPE, and VALUE are of a considerable size, especially because they are repeated for each property, and in each object instance. The last reason is that WBEM repeats the complete information from the instance key. In the beginning of the object description, WBEM implementation describes the property key with its name, its type and its value. After the instance key description, all the data describing the key property is repeated completely.

```
<INSTANCENAME CLASSNAME="D_Interface"><KEYBINDING
NAME="InterfaceID"> <KEYVALUE
VALUETYPE="numeric">39650</KEYVALUE></KEYBINDING></
INSTANCE NAME></INSTANCEPATH><INSTANCE
CLASSNAME="D_Interface"><PROPERTY NAME=
"InterfaceID" TYPE="uint16"
><VALUE>39650</VALUE></PROPERTY><PROPERTY
NAME="ADDR" TYPE="string"
><VALUE>2001:690:2380:778f:250:daff:fed6:
499c</VALUE></PROPERTY><PROPERTY NAME="Bandwidth"
TYPE="uint16"
><VALUE>10</VALUE></PROPERTY><PROPERTY
NAME="Uplink" TYPE="uint16"
><VALUE>1</VALUE></PROPERTY><PROPERTY
NAME="NetworkID" TYPE="uint16"
><VALUE>20064</VALUE></PROPERTY><PROPERTY
NAME="Prefix" TYPE="uint16"
><VALUE>12</VALUE></PROPERTY><PROPERTY
NAME="PrefADDR" TYPE="string"
><VALUE>214:748:364:217:222:2222:0001</VALUE></PROP
ERTY><PROPERTY NAME="BWUnit"
TYPE="uint16"><VALUE>2</VALUE></PROPERTY> <PROPERTY
NAME="IfIndex"
TYPE="uint16"><VALUE>11</VALUE></PROPERTY>
</INSTANCE> </VALUE.OBJECTWITHPATH>
```

Figure 3 – OpenWBEM instance encoding

WBEM solutions typically offer a binary communication option in order to avoid the encoding efficiency problem, such as the *owbinary* option implemented by OpenWBEM and the communication protocol used by Microsoft in the Microsoft Operations Manager application. Such an option alleviates the network from so much volume of signaling information. However it removes the semantic richness of the XML encoding which is one of the claimed advantages of WBEM technology.

Figure 4 illustrates the object encoding performed by the Openwsman application.

```
<p:D_Interface
xmlns:p=http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/D_Interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><p:InterfaceID>39639</p:InterfaceID><p:ADD
R>2001:690:2380:778f:250:daff:fed6:499c</p:ADDR><p:B
andwidth>10</p:Bandwidth><p:Uplink>1</p:Uplink><p:Ne
tworkID>20064</p:NetworkID><p:Prefix>12</p:Prefix><p
:PrefADDR>214:748:364:217:222:2222:0001</p:PrefADDR>
<p:BWUnit>1800</p:BWUnit><p:IfIndex>11</p:IfIndex></
p:D_Interface>
```

Figure 4 - Openwsman instance encoding

Openwsman also uses an XML schema to describe the object properties. However, during the object encoding it omits the details related to the object semantics. For instance, the property limits are defined within a single <p:> tag, much more efficient than the correspondent syntax in the WBEM specification. Finally, Openwsman omits the key property definition since it was already stated in the instance schema. The encoding differences previously discussed make a significant difference to the final message size when the number of objects exchanged increases.

B. *Protocol overhead*

Another metric under study was the protocol efficiency in terms of the ratio between the useful information vs. the transferred information. Figure 5 illustrates those results. The most efficient protocol was COPS due to its small object header. As the number of objects grows, the efficiency of the data exchange also increases, but only to an upper limit. The maximum efficiency ratio is above 2% for WBEM, around 5% for SNMP and WS-Management solutions, 11% for Diameter solution and 39% for COPS.

The SNMP protocol shows a better efficiency ratio for a small number of objects. This behavior has to do with the fact that SNMP has a smaller packet header than the other protocols. When the number of objects increases this advantage is lost for two reasons: the first was previously referred to and is dependent on our implementation; the second is that the SNMP transport protocol (UDP) does not maintain a session control and it makes the SNMP communication repeat the packet header, which does not happen with the other protocols.

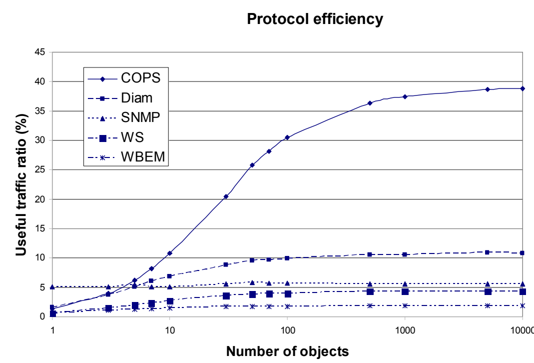

Figure 5 - Useful traffic ratio

C. *Memory allocation*

During the tests we monitored the server processes and we registered the amount of memory they use. The averages of maximum values were plotted in Figure 6.

It is a fact that the memory allocation is strongly application dependent, and this is the main reason why COPS and Diameter obtained better results than the other solutions. In the case of COPS and Diameter we have developed simple (albeit functional) programs to perform this study while for the remaining protocols we have explored, and adapted, complete applications. From the figure we can also observe the increase of memory usage with different slopes, from one technology to another. The increase of the used memory is only dependent on the communication protocol and does not suffer any influence from the application complexity.

The memory measures show smaller slopes for the COPS and Diameter applications. The bigger slopes are obtained by the WS-Management application, which is in line with the fact that Openwsman produces a translation from the WBEM to the SOAP encoding. Additionally, we observed during the tests that the WS-Management server does not implement any kind of parallel processing. It waits for a complete WBEM message arrival to start producing the WS-Management message for its client. This behavior makes the application have a second copy of each object in memory and it explains the memory waste in the WS-Management application.

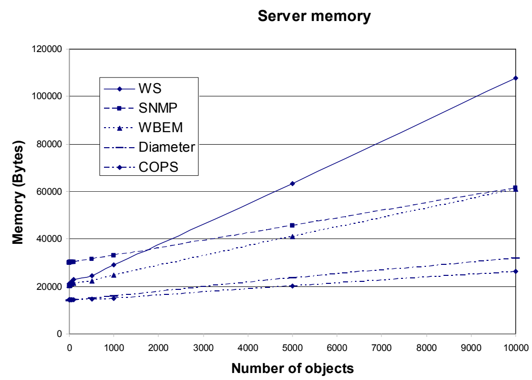

Figure 6 - Memory allocation

D. Round-trip delay

We also measured the time elapsed between the configuration request and the last configuration message packet. The averages of those values are plotted in Figure 7. Observing this diagram we can detect that the faster answers were obtained by the COPS application. In order of increasing swiftness we have the WBEM application, then SNMP, the WS-Management and finally the Diameter. The WS-Management performance analysis has to consider the fact that Openwsman has to wait for the WBEM transference and it makes the Openwsman much slower than it in fact is. However, according to [10] that fact did not undermine its performance.

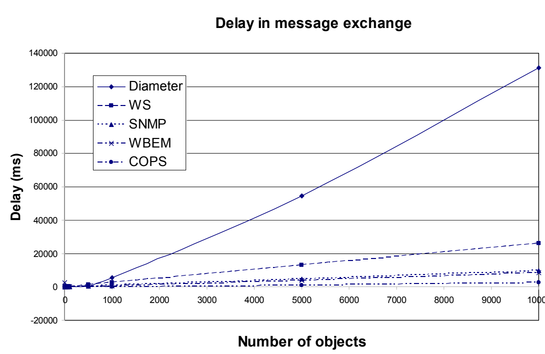

Figure 7 –Delay in message exchange test results

The SNMP performance was disappointing, and it was even worse than WBEM. The same reasons we referred to before can help to explain these delays: it does not use the transport resources efficiently and since the UDP does not maintain a session between the client and the server, SNMP client has to repeat the SNMP requests after it receives each answer from the server.

The lower performance of Diameter cannot be explained by message complexity, but is only due to an implementation fault. This is especially clear, as its performance shows significant degradation when the number of exchanged objects increases.

E. Compression gain

The signaling compression tests showed that many efficiency gains could be easily obtained by means of a compression library, and as such this should be considered for NGN solutions.

From the beginning, XML protocols had greater compression gains since the verbosity of its encoding makes the compression algorithm (61 % - WS, 45% - WBEM, 34 % - Diameter, 21% SNMP and 4% - COPS) more efficient. The compression gain rises with the increase of the number of objects exchanged during the test, reaching a limit around 1000 objects exchanged, but by then the difference is not as remarkable between protocols (91% for WBEM and WS, 90% - Diameter, 86% - COPS, 63% - SNMP).

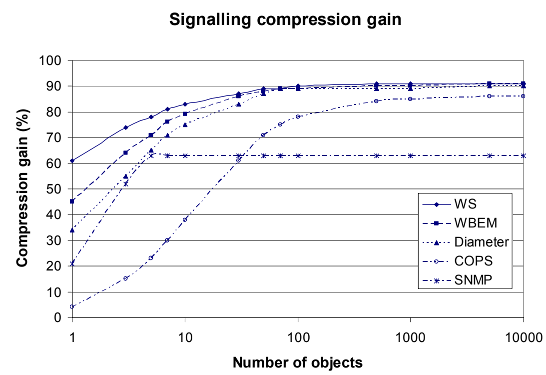

Figure 8 - Signaling compression gain

Figure 8 plots the results from the compression tests. An exception in the increasing gain was observed during the SNMP tests. Since SNMP client and server pair performs successive requests/responses, the compression gain is limited by the number of objects encoded in each response.

V. DISCUSSION

Analyzing the results we observed, in general, a much better performance of the binary over the XML-based protocols.

COPS obtained the best performance in all the analyzed parameters. It is a binary protocol that uses a very efficient encoding scheme. In a similar way, Diameter shows a good performance in terms of number of packets, in terms of message signaling and in terms of memory usage. Compared to COPS, the Diameter protocol seems to be more complex; it uses bigger request messages, longer headers and it requires more bytes to encode the same objects.

The SNMP protocol has been demonstrated to be more efficient for a small number of objects but it does not scale very well, mostly because of implementation issues. The original simplicity behind SNMP philosophy always considered that a small packet could pass easily through an overloaded router queue and then solve a network configuration problem. This is probably what influenced the

Table 3 - Overall comparison

Tech.	Sing Vol..	Numb. of packets.	Prot. Overhead..	Memory.	Delay	Compres.	Scalability	Advantage	Issues
SNMP	3	5	3	4	3	5	5	Widely used. Several open source implementations	Does not scale well. Usage usually limited to monitoring
COPS	1	1	1	1	1	4	1	Extremely efficient	Reduced set of applications
Diameter	2	2	2	2	5	3	2	Modularity Becoming very popular	Excess of information and requires a complex state machine in the server
WBEM	5	4	5	3	2	2	4	Interoperability Large number of open source implementations	Overhead and memory. Needs compression
WSMAN	4	3	4	5	4	1	3	Interoperability Easy to develop	Overhead

SNMP API developers for this particular implementation, since simulated values obtained from the RFC specification indicated signaling values 40% lower than the ones obtained during the tests.

Web services technology showed itself to be a very promising technology since it presented considerable gains in terms of performance over WBEM technology, maintaining the interoperability characteristics and offering a much easier application development. The efficiency achieved by the Web services implementation is due to better object encoding. The Web services implementation avoids the intricate semantic description of each object's property, making use of the instance schema to describe the object semantics.

The compression tests showed a promising technique to obtain better performance in terms of protocol overhead. This gain increases with the number of objects exchanged and is greater for the XML protocols. Although compression processing represents a cost, it should be considered particularly in high signaling networks. Moreover, a tradeoff should be made between the signaling and the compression/decompression time costs, and clearly, some scenarios, where only a small number of objects are exchanged per request, are not suitable for compression (for instance, in an admission control process where a router asks for permission to route a traffic flow).

Table 3 summarizes the evaluation results of the different technologies in the set of tests carried out, with extra empirical information that came from our personal assessment of the protocols and their implementation.

Although XML-based protocols show the worst performance in terms of information transport, they show important advantages over the binary solutions such as improved interoperability. Also they provide much richer and more powerful frameworks for the development of management solutions.

Binary protocols should preferably be used in network entity communication during network operation as in admission control messages or during the user registration phase so as not to compromise network scalability. The XML based protocols should be used in the network configuration phase since they are more powerful and the configuration actions are not as frequent as the operation messages.

Additionally, hybrid solutions could be used, making use of several management technologies and performing the protocol translation between the elements [6, 7, 51] through the use of a proxy or a gateway. As network device complexity increases, even for low cost units, it may become feasible to support two different protocols in parallel. In those conditions, a mixture of COPS/Diameter and Web Services seems to answer most management requirements.

VI. Conclusion

We carried out performance analysis of several common management protocols like SNMP, Diameter, COPS, CIM-XML over HTTP and CIM-XML over SOAP, in terms of signaling bandwidth, exchanged messages, round-trip delay and transport efficiency. Our aim was to consider the impact of these protocols in novel NGN scenarios where IMS solutions and wireless mesh infrastructures will be common.

The test results showed a considerable performance advantage of the binary protocols over the XML-based protocols. In the latter, Web services present much more efficient object encoding than WBEM and thus showed better performance – although compression would be a good technique to apply in general to this environment. COPS was the most efficient of the protocols under study. The Diameter protocol is an acceptable compromise, most especially because of its flexibility and of the large set of applications that use it.

Some of the presented results are of course implementation dependent. Nevertheless, the selected applications are typical examples of these protocols, and clearly indicate the trends that can be found in these different management protocols.

One interesting point arising from this analysis is that hybrid solutions, with Web Services for early configuration, and COPS or Diameter for running operation, may be an interesting compromise between flexibility and efficiency for NGN.

REFERENCES

[1]. J. Case, M. Fedor, M. Schoffstall, and J. Davin, "RFC 1157 - Simple Network Management Protocol (SNMP)", May 1990.

[2]. 3GPP, "3GPP TS 32.102-Telecommunication management; Architecture", Jan 2008.

[3]. D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "RFC 2748 - The COPS (Common Open Policy Service) Protocol", Jan. 2000.

[4]. P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "RFC 3588 - Diameter Base Protocol", Sep. 2003.

[5]. "SWIFT home", http://www.ist-swift.org/, 2008-07-30.

[6]. A. Pras, T. Drevers, R.v.d. Meent, and D.A.C. Quartel, "Comparing the performance of SNMP and Web services based management," IEEE electronic Transactions on Network and Service Management, vol. 2, Nov. 2004.

[7]. R. Neisse, R. Vianna, L. Granville, M. Almeida, and L. Tarouco, "Implementation and bandwidth consumption evaluation of SNMP to Web services gateways," in Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, 2004, pp. 715-728 Vol.711.

[8]. G. Pavlou, P. Flegkas, S. Gouveris, and A.A.L.A. Liotta, "On management technologies and the potential of Web services," Communications Magazine, IEEE, vol. 42, pp. 58-66, 2004.

[9]. R. Vianna, R. Almeida, L. Tarouco, and L. Granville, "Evaluating the Performance of Web Services Composition for Network Management," in Communications, 2007. ICC '07. IEEE International Conference on, 2007, pp. 1943-1948.

[10]. G.C.M. Moura, G. Silvestrin, R.N. Sanchez, L.P. Gaspary, and L.Z. Granville, "On the Performance of Web Services Management Standards - An Evaluation of MUWS and WS-Management for Network Management," in 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007) Munich, Germany, 2007, pp. 459-468.

[11]. B. Wijnen, D. Harrington, and R. Presuhn, "RFC 2271 - An Architecture for Describing SNMP Management Frameworks", January 1998.

[12]. J. Case, D. Harrington, R. Presuhn, and B. Wijnen, "RFC 2272 - Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", January 1998.

[13]. D.B. Levi, P. Meyer, and B. Stewart, "RFC 2273 - SNMPv3 Applications", January 1998.

[14]. U. Blumenthal, and B. Wijnen, "RFC 2274 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", January 1998.

[15]. B. Wijnen, R. Presuhn, and K. McCloghrie, "RFC 2275 - View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", January 1998.

[16]. J. Boyle, R. Cohen, D. Durham, R. Rajan, S. Herzog, and A. Sastry, "RFC 2749 - COPS usage for RSVP", Jan. 2000.

[17]. K.H. Chan, D. Durham, S. Gai, S. Herzog, K. McCloghrie, F. Reichmeyer, J. Seligson, R. Yavatkar, and A. Smith, "RFC 3084 - COPS Usage for Policy Provisioning (COPS-PR)", Mar. 2001.

[18]. P. Calhoun, T. Johansson, C.E. Perkins, T. Hiller, and P.J. McCann, "RFC 4004 - Diameter Mobile IPv4 Application", August 2005.

[19]. M.A. Garcia-Martin, M.-C. Belinchon, M.A. Pallares-Lopez, C. Canales-Valenzuela, and K. Tammi, "RFC 4740 - Diameter Session Initiation Protocol (SIP) Application", November 2006.

[20]. 3GPP, "TS 23.228-IP Multimedia Subsystem (IMS) Stage 2", Jan 2008.

[21]. 3GPP, "TS 29.328-IP Multimedia (IM) Subsystem Sh interface; Signalling flows and message contents", June 2008.

[22]. 3GPP, "3GPP TS 29.229 -Cx and Dx interfaces based on the Diameter protocol; Protocol details", June 2008.

[23]. 3GPP, "3GPP TS 29.214-Policy and charging control over Rx reference point", May 2008.

[24]. 3GPP, "3GPP TS 29.212-Policy and charging control over Gx reference point", May 2008.

[25]. M. Brenner, "RFC 5224 - Diameter Policy Processing Application", March 2008.

[26]. DMTF, "Web-Based Enterprise Management (WBEM) Initiative", http://www.dmtf.org/standards/wbem/, 2008-12-14.

[27]. DMTF, "Common Information Model (CIM) Specification - Version 2.9", 2005.

[28]. DMTF, "Specification for the Representation of CIM in XML", 02/05/2002.

[29]. DMTF, "Specification for CIM operations over HTTP version 1.1".

[30]. W.Q. de Lima, R.S. Alves, R.L. Vianna, M.J.B. Almeida, L.M.R. Tarouco, and L.Z. Granville, "Evaluating the Performance of SNMP and Web Services Notifications," in 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006). , 2006, pp. 546-556.

[31]. S.-J. Lee, M.-J. Choi, S.-M. Yoo, J.W. Hong, H.-N. Cho, C.-W. Ahn, and S.-I. Jung, "Design of a WBEM-based Management System for Ubiquitous Computing Servers," 2006; http://www.dmtf.org/education/academicalliance/mjchoi_2004.pdf.

[32]. "OpenWBEM project", www.openwbem.org, 2006-04-05.

[33]. "C++ CIM/WBEM Manageability Services Broker", http://www.openpegasus.com, 2006-04-05.

[34]. "SBLIM - Standards Based Linux Instrumentation for Manageability", http://www-124.ibm.com/sblim/index.html, 2005-02-16.

[35]. "Java™ Web Based Enterprise Management", http://wbemservices.sourceforge.net/, 2006-04-05.

[36]. M. Gudgin, M. Hadley, and T. Rogers, "Web Services Addressing 1.0 - SOAP Binding", 9 May 2006.

[37]. D. Booth, and C.K. Liu, "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer", 26 June 2007.

[38]. DMTF, "Web Services for Management (WS-Management)", 2006-04-05.

[39]. W. Vambenepe, "Web Services Distributed Management: Management Using Web Services (MUWS 1.0)", 9 March 2005.

[40]. A.W.S. Project, "Apache MUSE - A Java-based implementation of WSRF 1.2, WSN 1.3, and WSDN 1.1", http://ws.apache.org/muse/, 2008-07-22.

[41]. Sun, "Wiseman: A Java implementation of WS-Management", https://wiseman.dev.java.net/, 2006-04-05.

[42]. "Openwsman | WS - Management for All", http://www.openwsman.org/, 02/05/2008.

[43]. R.C. Pereira, and L.Z. Granville, "On the performance of COPS-PR and NETCONF in an integrated management environment for DiffServ-enabled networks," in Telecommunications, 2008. ICT 2008. International Conference on, 2008, pp. 1-6.

[44]. 3GPP, "3GPP TS 23.517-IP Multimedia Subsystem (IMS); Functional Architecture", Dec 2007.

[45]. 3GPP, "3GPP TS 32.101-Telecommunication management; Principles and high level requirements", Dec 2007.

[46]. J.L. Oliveira, P. Gonçalves, W. Dziunikowski, J. Wszolek, S. Rasmussen, R.P. Lopes, and V. Roque, "Policy-Based Network Management in an Integrated Mobile Network," in Advanced Industrial Conference on Telecommunications, Lisbon, Portugal, 2005.

[47]. D. Gomes, "COPS++", http://hng.av.it.pt/~copspp/, 30/12/2008.

[48]. "Agent++ - SNMPv1/v2c/v3 Agent API for C++", http://www.agentpp.com/, 2008/06/09.

[49]. "Net-SNMP", http://net-snmp.sourceforge.net/, 2008-05-02.

[50]. W. Dziunikowski, J. Wszolek, P. Gonçalves, S. Rasmussen, A. Cuevas, R.P. Lopes, V. Roque, and J.L. Oliveira, "Using CIM Extensions to model Managed Entities in Heterogeneous Networks," in IEEE International Workshop on Management Issues and Challenges in Mobile Computing, Nice, France, 2005.

[51]. Y. Jeong-Hyuk, J. Hong-Taek, and W.H. James, "Development of SNMP-XML translator and gateway for XML-based integrated network management," Int. J. Netw. Manag., vol. 13, pp. 259-276, 2003.