Comparing Event Reporting Solutions for Autonomic Management

Pedro Gonçalves
University of Aveiro, ESTGA/Instituto de Telecomunicações
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
Tel: 351-234-377-900 E-mail: pasg@ua.pt

Diogo Loureiro
University of Aveiro, DETI/Instituto de Telecomunicações
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
Tel: 351-234-377-900 E-mail: diogo.loureiro@ua.pt

António Nogueira
University of Aveiro, DETI/Instituto de Telecomunicações
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
Tel: 351-234-377-900 E-mail: nogueira@ua.pt

Received: July 23, 2012 Accepted: November 21, 2012 Published: December 16, 2012
DOI: 10.5296/npa.v4i4.2133 URL: http://dx.doi.org/10.5296/npa.v4i4.2133

Abstract

The increasing complexity of current communication networks will greatly benefit from the deployment of autonomic monitoring and management architectures that allow the network to detect, diagnose and repair failures autonomously, adapt its configuration and optimize its performance and quality of service parameters. Knowing the potential of existing management technologies can help choosing the most appropriate solutions for specific network/services scenarios. As an illustrative solution, this paper proposes and evaluates a

NETCONF (Network Configuration) agent for link-state monitoring, a fault-detection mechanism based on notifications. Then, a comparative study between different management technologies, in terms of their scope, flexibility and market adoption, will be conducted: by performing a series of tests on representative prototypes, the results obtained regarding different performance issues (like signaling overhead, memory requirements, coding efficiency and response times) are analyzed and compared in order to identify which solutions are more appropriate to be used on future autonomic network monitoring and management scenarios. SNMP (Simple Network Management Protocol), the WBEM (Web Based Enterprise Management) and WS-MAN (Web Services Management) web-based approaches and the NETCONF protocol are exhaustively tested and compared.

Keywords: Autonomic management, Fault-detection, Network monitoring, Event notification, NETCONF, WBEM, WS-MAN, SNMP.

1. Introduction

As networks become larger and more complex, they also become harder to manage in an efficient and reliable way. Autonomic network monitoring and management, where the network itself is able to detect, diagnose and repair failures, adapt its configuration and optimize its performance and quality of service, is becoming an increasingly relevant research area. Sensor [1] and mesh networks [2] are just two communication scenarios where network availability is a critical issue. In order to regulate the availability of the service offered to customers, agreements between customers and vendors usually define values of service availability and include minimum penalties for the suppliers whenever they do not achieve them. Efficient tools, like software agents, active networks and policy languages, which are able to alleviate the task of network management from human operators, are fundamental in these environments. However, in order to implement such an autonomic paradigm several technological decisions should be made, choosing which solutions are appropriate to provide the required degree of generalization and flexibility.

Since the emergence of communication networks, several management and administration technologies were proposed. SNMP (Simple Network Management Protocol) [3] was proposed in the late 80s and rapidly became the de facto technology in the Internet area, although several technological limitations (like security faults and protocol inefficiency in large configuration scenarios) have been raised by different authors [4, 5]. In the late 90s, the Distributed Management Task Force (DMTF) developed a new management technology called Web-Based Enterprise Management (WBEM) [6]: following an integrated management approach, a wide data model [7] representing a vast range of information, ranging from network to service management, was defined. Later, with the emergence of service-oriented architectures, two new initiatives emerged: the Web Services Management [8] from DMTF and the WSDM-MUWS (Web Services Distributed Management-Management Using Web Services) [9] from OASIS (Organization for the Advancement of Structured

Information Standards), which used the W3C standards for web-services. Recently, the NETCONF [10] (Network Configuration) protocol was standardized by the Internet Engineering Task Force (IETF) as a new approach that includes the recommendations of both network operators and protocol developers. NETCONF uses XML for encoding, provides support for several secure transport protocols and uses YANG [11], a language that was specifically created to be used with this protocol, to describe management information. The NETCONF technology has been receiving a lot of attention from the academic and industry communities: several management solutions have been developed based on this technology, SDKs for solution development are already available and some leading networking vendors already provide NETCONF management support in their equipment [12].

As an illustrative monitoring solution, this paper will describe the implementation of a network link-state monitor based on NETCONF. This monitor allows event subscription by the element manager, while also allowing the agent that is responsible for monitoring the link status to send events to the manager element. The implementation is based on the gSOAP platform, since it automatically generates the communication interfaces code based on the provided data models. Several functional tests are conducted, as well as tests to evaluate the performance in terms of the notifications transmission time and the amount of signaling traffic between manager and agent.

Then, the paper will present a comparative study between different event-reporting technologies, in terms of their scope, flexibility and market adoption. By conducting a series of tests on representative prototypes from these technologies, the results obtained regarding different performance issues, like signaling overhead, memory requirements, coding efficiency and response times, are analyzed and compared in order to identify which solutions are more appropriate to be used on future autonomic network monitoring and management architectures.

The organization of this article is as follows. Section 2 gives an overview of the most relevant work on the evaluation of management technologies. Section 3 briefly describes the most relevant event reporting technologies, covering SNMP, WBEM, Web Services technologies and NETCONF. Section 4 presents the proposed implementation of the network monitoring agent, discussing also the management scenario that is envisaged for the proposed framework; at the end of the section, the results obtained from the evaluation tests are presented and discussed. Section 5 is dedicated to the evaluation of the different event reporting technologies, presenting the testing scenario and the results obtained regarding the different performance metrics that were selected. Finally, Section 6 presents the main conclusions of the developed work.

2. Related Work

Much work has been done so far in the area of technology evaluation, particularly regarding SNMP. Yoo et al. [13] carried out a performance evaluation of the NETCONF protocol with the various transport options at the time. They also proposed methods for improving the performance of NETCONF-based management solutions. Franco et al. [14] compared the performance of the NETCONF, COPS-PR and SOAP protocols finding out that, although NETCONF and SOAP produce more signaling than COPS-PR, they could use compression techniques in order to compensate for this disadvantage.

Gonçalves et al. [15] evaluated the encoding overhead of several management protocols in a well-defined configuration management scenario. The study considered signaling volume and memory requirements. A general gain in performance of the binary-based technologies over XML-based was observed. Moura et al. [16] presented a performance evaluation of Web Services for management applications, observing a performance gain of the DMTF standard over its OASIS equivalent. Chourmouziadis et al. [17] compared the performance of WS and SNMP event reporting in a policy-based QoS management platform. Neisse et al. [18] described the implementation of a SNMP to WS gateway and evaluated the bandwidth consumption of these different technologies. Pras et al. [19] methodically analyzed SNMP message encoding and the signaling produced in a configuration provisioning scenario. They compared the signaling volume, the computation resources and the time-to-relay of both SNMP and WS, studying additionally the compression effect over the signaling volume. Pavlou et al. [20] performed a performance evaluation of SNMP, CORBA and WS technologies, also studying the memory requirements, time-to-reply and signaling volume. Lima et al. [21] compared SNMP and WS as notification technologies. Their paper analyses network usage and the time-to-reply of the event reporting technologies. Furthermore, it proposed and evaluated an SNMP to WS gateway that responds to the network element traps and forwards the monitoring information to a management server in the form of a WS notification.

Andrey et al. [22] surveyed the SNMP-related performance studies that were carried out over the last 10 years. These authors discovered that those studies used different techniques and scenarios and addressed different metrics. So, in reference [22] they propose techniques, approaches and metrics that should be followed in order to reach a benchmarking framework that would allow quantifying the performance of SNMP-based application and reuse of the performance values obtained in future works. Schönwälder et al. carried out an SNMP traffic analysis [23]. They verified that the most used versions of SNMP are SNMPv1 and SNMPv2, besides identifying the most frequent messages in real SNMP environments.

The performance comparison studies mentioned above compare two or, at most, three technologies, but do not allow a clear choice of the most appropriate technologies for network management deployment. Moreover, since there is no uniform test scenario, it is not possible to correlate results from the different studies.

3. Overview of Event Reporting Technologies

Currently, there are several alternative technologies for event reporting that were standardized in different contexts and envisaged for different scenarios. We selected some of the most relevant technologies, all of them potential candidates for implementing the dynamic behavior of autonomous management systems.

3.1 SNMP traps

SNMP [3] is an IETF technology standardized in 1990. Its design followed a simplicity-based approach, including a small number of operations, whose messages are transported in UDP. In spite of the different performance issues that have been pointed to SNMP, most of them related to security aspects, SNMP has been widely adopted as the main tool for network administrators.

During the 90s, several versions of the protocol were proposed, improving the standard and overcoming the issues that were appointed to the technology. However, according to Schönwälder et al. [23], the latest SNMPv3 version is not used in network management. SNMP uses SMI (Structure of Managed Information) for management data description through a relational model using MIBs (Management Information Bases). SNMP provides operations to read (*get, getnext, getbulk*) from and write (set) to MIBs and also for synchronous (Inform) and asynchronous (Trap) event notifications. The trap system defines a set of notification types that can be forwarded (*coldStart, warmStart, linkDown, linkup, authenticationFailure, egpNeighborLoss* and *enterpriseSpecific*).

3.2 WBEM

WBEM [8] was initially proposed by companies from the desktop management area, and was later developed by the Distributed Management Task Force (DMTF). WBEM specification includes a set of technologies imported from the web world, such as the HTTP based transport mechanism (CIM operations over HyperText Transfer Protocol (HTTP)) [24] and the XML based specification for the information encoding (CIM-XML). The data model used in the WBEM technology is the Common Information (CIM) [25], a data model proposed by the DMTF that aims to integrate management information of the desktop and of the network areas.

CIM Event model [26] implements WBEM notifications as Indications. Entities interested in receiving indications subscribe the indication reception to the indication producer, which usually consists in a management agent. To subscribe Indications, the manager creates an instance of the *IndicationSubscription* class that references instances of an *IndicationFilter* class, to set the filter rule that will select the objects of the indication and an instance of an *IndicationHandler* class setting the encoding and transport definitions of the indication.

At the time the interest event occurs, the agent creates an indication and sends it to the indication subscriber. Indications can be from two types: Life Cycle Indications, regarding the creation, deletion or modification of CIM classes or instances, and Process Indications to notify about all other events that do not fit in the previous rule, like for example objects that may not be modeled in CIM, such as SNMP traps. A very common handler is the *CIM_ListenerDestinationCIMXML* that uses CIM-XML for encoding and HTTP for transport; other classes can be used, for instance to email the indication or to use a mobile service.

3.3 WS technologies

Following the SOA (Service Oriented Architectures) trend, DMTF and OASIS proposed the WS-Management [8] and the WSDM [27] approaches, respectively. They based their efforts in the W3C (World Wide Web Consortium) standards for web-services, like SOAP and

Web Service Description Language (WSDL), to define solutions that can remotely access and exchange management information in distributed environments.

The event notification mechanism implemented by WS-Management is very similar to WBEM notifications; the only difference refers to the inclusion of an unsubscribe message that is sent by the indication consumer when it no longer wants to receive indications.

OASIS WSDM was divided in two sub-standards: Management Of Web Services (MOWS) and MUWS. MUWS implements monitoring by sending event information but, unlike WS-Management, it does not plan a subscription process. Albeit less flexible, WS-Management shows better performance than WSDM due to a simpler specification and a small number of operations [16].

3.4 NETCONF

NETCONF follows a client-server architecture and is conceptually divided in four layers. The transport layer must provides the main security features of the protocol, providing definitions to use SSH, SOAP, BEEP or TLS, although the SSH implementation is mandatory [19]. Upper in the layer structure, NETCONF uses the RPC (Remote Procedure Call) paradigm, using XML encoding to provide a transport independent data exchange. Above this, stands an operation layer that includes the set of NETCONF operations that can be invoked by the RPC methods. These operations work over the device configuration data defined in the top layer, the content layer.

NETCONF is designed to distinguish between status and configuration data, providing operations like <*get-config*> or the more generic <*get*> for data retrieval. For manipulating the management objects, operations such as <*edit-config*>, <*copy-config*> and <*delete-config*> are available.

NETCONF also supports the use of multiple configurations in the same device, defining the use of three main configurations. The startup configuration is applied during the boot process, the candidate configuration can be freely manipulated, without any consequences to the device operative status, while the running configuration holds the active configuration of the device.

NETCONF event management follows a subscription-notification model, where the manager subscribes the notifications streams to the agent. When an event occurs the agent sends the notification to the stream subscriber. For better granularity, the manager is able to set a filter to the subscripted stream, avoiding receiving unwanted notifications. Fig. 1 illustrates the event notification model.

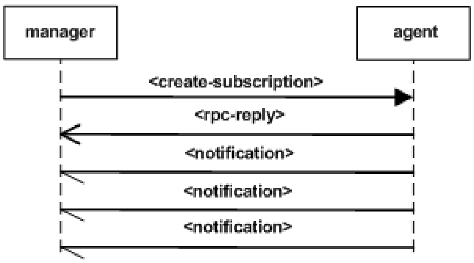


Figure 1. NETCONF event notification model.

A reduced version of the original NETCONF protocol, named NETCONF Light [28], which includes a subset of the original protocol functionalities, was proposed for devices with limited computing resources, particularly those with low memory. Regarding the differences from the original protocol, we can highlight that NETCONF light operations lack support for configuration filtering functionalities. Despite maintaining the original protocol operations, the light version removes the possibility of defining a filter that limits the operations scope over the equipment configuration.

This version requires a small number of sessions and only one data repository, the running configuration. The *get*, *get-config*, *copy-config*, *lock*, *unlock*, *close-session* and *kill-session* operations are mandatory. The *delete-config* operation no longer makes sense, since there is only the running configuration. Filtering is optional, since it is a very resource-consuming feature. Consequently, the *edit-config* operation may not be supported when managing only complete configurations, which should not be considered as a problem since it is probably a reduced size configuration.

Table 1 summarizes the key features of the analyzed management technologies. Since our application scenario belongs to the network management area and the standardization entity (IETF) developed a new technology (NETCONF) that has been receiving a tremendous attention from academia and industry, we chose NETCONF as the basis for the development of our monitoring solution. Moreover, this technology offers a tremendous flexibility, allowing performing notification subscription and avoiding unwanted notifications, while keeping a centralized agent configuration.

Table 1. Comparison between fault management technologies.

Technology	Standard entity	Model	Scope	Usage
SNMP	IETF	trap or inform- response	Network management	Network equipment
WBEM	DMTF	subscription- indication	Enterprise management	Desktop networking
WS-Management	DMTF	subscription- indication	Enterprise management	Desktop networking
WSDM	OASIS	subscription- indication	Enterprise management	Desktop networking
NETCONF	IETF	subscription- notification	Network management	Network equipment

4. Network monitoring agent

The monitoring solution followed RFCs 4147 and 5277, and the transport solution to the

NETCONF operations chosen approach was NETCONF over SOAP. The NETCONF light version was chosen due to its applicability in resource-constrained devices, and a subset of the operations defined in the standard were developed.

In order to minimize the overhead created by SOAP, we choose gSOAP since it offers better performance than other web service frameworks [29, 30]. For the same reason, we decided to implement our software using the C language, thus reducing the required computational resources.

The gSOAP framework generates NETCONF SOAP communication logic based on XML Schemas provided by RFC 4147 for the NETCONF base operations and RFC 5277 for the event notifications operations. Additionally, it supports RPC-XML and asynchronous message exchange, which are essential features for this implementation. NETCONF base operations were implemented as synchronous web services, while event notification messages were implemented in the form of a long-run web service.

Fig. 2 illustrates the subscription creation process that was implemented: the manager performs a subscription creation and sends a <*create-subscription*> operation to the agent which, after validating the request, sends back the answer to the manager. Accepted subscription requests lead to the creation of independent threads: the manager creates a listener thread to receive notifications, while the agent creates a monitoring thread to detect link failures. Unlike the <*create-subscription*> message, <*notification*> messages are asynchronous. Once a link failure event is detected, the agent sends a notification message to the managers that subscribed the event stream.

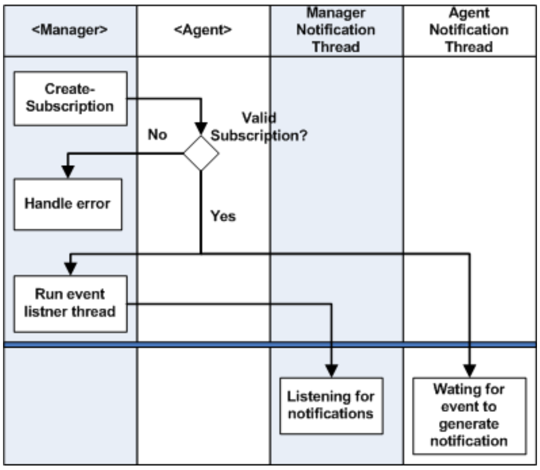

<Manager>
<Agent>
Manager Notification Thread
Agent Notification Thread
Create-Subscription
Valid Subscription?
No
Handle error
Yes
Run event listner thread
Listening for notifications
Wating for event to generate notification

Figure 2. The <create-subscription> process.

4.1 Management scenario

Currently, many configuration management solutions have a layered approach with three layers: the Service Layer that defines concepts related to the day-to-day business flows of a service provider, using service models typically defined using SID (Shared Information/Data Model), UML (Unified Modeling Language) or proprietary languages; the Resource Layer that provides a mapping from the Service Layer to actual device manipulations, modeling individual devices (like switches, routers or DSLAMs) using XML, UML or proprietary languages; the Mediation Layer that maps changes to the local data structures in the Resource Layer to actual configuration change commands on the devices.

NETCONF intends to simplify all these layers by defining how to execute configuration changes, by using stringent YANG models to define device configurations and using technologies such as XMLBeans, Castor, Xgen or JAXB to obtain a set of Java classes that can be used to manipulate the configuration instances.

In fact, as already said, NETCONF is an XML-based protocol specifically designed to configure and manage the most demanding network situations by providing automated configuration management, improved network security and reliability, and robust configuration changes. NETCONF actions are mandatorily communicated across the network in a secure way.

Fig. 3 depicts a hypothetic NETCONF deployment scenario, where different network configurable devices can be centrally configured/controlled using several transport protocols. Setting up routing parameters of network routers or the security values of firewalls, while monitoring their behavior, can be efficiently performed using a remote centralized server. Besides, depending on the capabilities of monitoring probes, their configuration can comprise parameters for flow metering and aggregation, packet sampling, and/or the export of monitoring data.

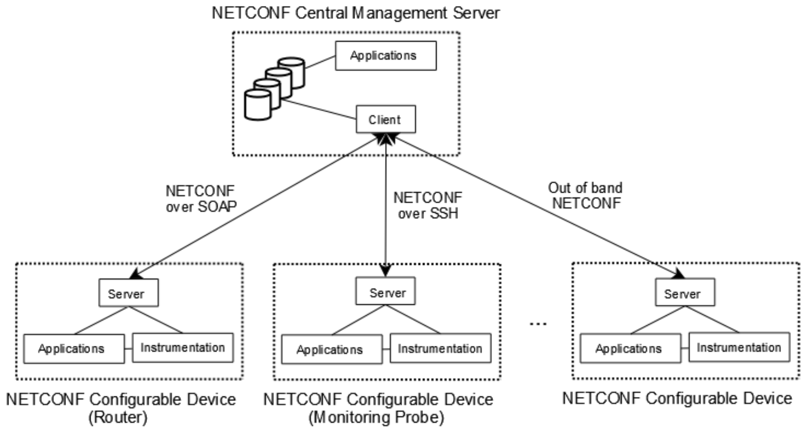


Figure 3. NETCONF deployment scenario.

The use of a device-specific command line interface (CLI) or a configuration file can be cumbersome and complicated, especially if used in heterogeneous networks consisting of different device models. The NETCONF standard simplifies all these tasks, while assuming that real operating networks are composed by various devices from diverse vendors.

4.2 Performance tests

In order to validate the developed management solution, some functional tests were conducted. Additionally, in order to evaluate the applicability of the proposed solution to a real network, we also evaluated the signaling overhead, the protocol encoding efficiency, the response time and the memory usage level.

Tests were conducted in a machine with an Intel Mobile Core 2 Duo, at 2.2GHz and having 2GB of RAM, with a native installation of Ubuntu 10.10. The traffic that was generated during the experiments was captured and analyzed. Fig. 4 illustrates the <notification> message format corresponding to a link-up event.

```
<ns2:notification xsi:type="ns2:NotificationType"> <ns2:eventTime> 2011-10-17T17:16:27Z</ns2:eventTime><ns2:info>link up</ns2:info>
</ns2:notification>
```

Figure 4. NETCONF notification.

The traffic analysis allows us to evaluate the influence that the protocol can have in network performance. So, network data regarding the number of packets and number of bytes transferred during the monitoring operations was gathered. Several link-down and link-up events were caused and all messages exchanged during the communication between the manager and the agent, were captured.

Fig. 5 depicts the traffic analysis results. Both the total number of packets and the amount of signaling data increased with the number of events reported from the agent. The increase was more or less linear, except for small numbers of events where the increase is exponential. Each notification message produces two packets and generates 935 bytes.

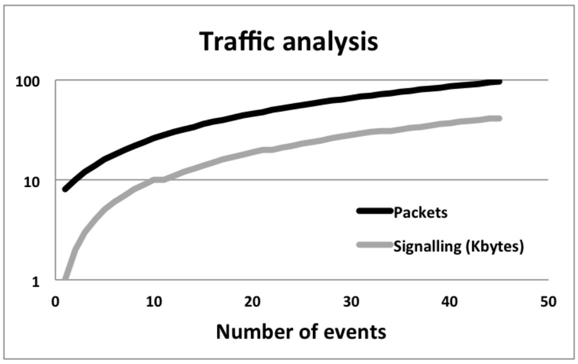


Figure 5. Traffic analysis.

A deeper inspection to the captured traffic confirms the verbose characteristic of NETCONF, which was already visible in Fig. 4. NETCONF technology encodes notifications in a 869 bytes Ethernet packet, which is acknowledged by a TCP ACK packet.

If we measure the individual sizes of the message components (Fig. 6), we can see that useful information is less than 4% of the total information that is sent in the network packet. In addition, we note the preponderance of the component related to the SOAP envelope, which represents more than 50% of the information contained in the Ethernet packet. Although a high overhead generated by the SOAP framework was expected, its dimension is quite impressive, being even higher than the sum of the HTTP component with the component associated to the NETCONF data description (YIN).

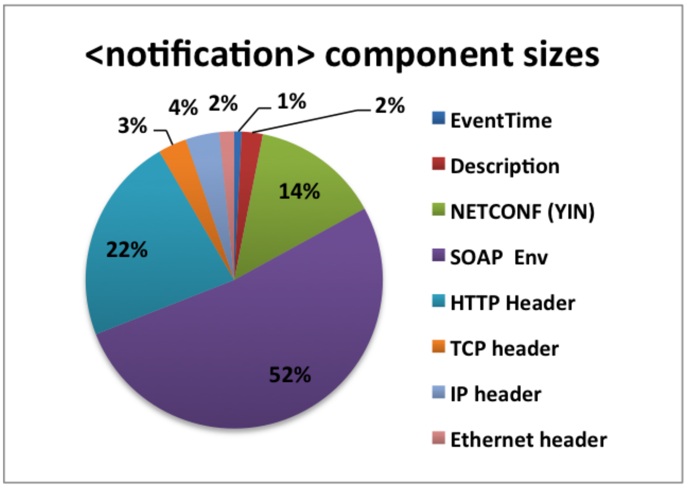


Figure 6. Individual component size of <notification>.

In order to evaluate the response times of the monitoring system, we changed the agent code to generate a higher number of event notifications. The time that is necessary to send notifications increased with the number of notifications sent by the agent. The conducted tests have also shown that the time required to send a notification is approximately equal to 200 ms.

5. Technology evaluation

This section describes the tests that were carried out to evaluate the technologies studied in previous sections and analyzes the results obtained from those tests. The test scenario reflects a fault notification operation where a router notifies the management platform about events that occurred in a communication link.

Notification subscription messages are typically exchanged between the equipment's and the management at equipment startup. As so, we considered those messages infrequent and that its effect was no significant impact in the network performance, disregarding its effect in the present study.

5.1 Test scenario

Tests were performed using a pair of applications created by each of the studied technologies. During the tests, the traffic exchanged between the monitoring agent and the management console was captured and the captured traces were analyzed in order to obtain the number of packets, the number of bytes and the response times. The applications under evaluation were placed in different machines connected through an isolated Ethernet LAN in order to avoid external disturbances. Tests were repeated twelve times and the tests results were processed. In all cases, the 95% confidence intervals are negligible.

The client and server applications corresponding to each technology were developed using the C/C++ language:

- A NETCONF client and server interfaced by a SOAP transport that were developed using gSOAP;

- A SNMP client and server developed in C using NET-SNMP;
- A WBEM client and server implemented in C using an *openpegasus* open-source implementation.

The applications installed at the agent monitored the network connection state and periodically sent link-up and link-down notifications.

Tests were conducted using machines with an Intel Core 2 Duo @ 2.2GHz processor with 2GB of RAM, with a native installation of Ubuntu 10.10. In order to generate and analyze traffic, bash scripting and *Wireshark* 1.6.2 were used.

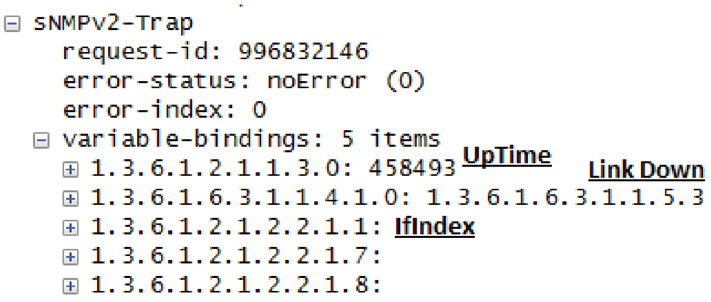
5.2 Encoding efficiency

The SNMP trap message (Fig. 7) is transported over UDP and contains an authentication header, a PDU header and a variable binding list containing the trap event information. SNMP has two different trap PDUs, one for SNMPv1 and another for SNMPv2, and according to Lima et al. [19] the expressions that describe SNMPv1 (1) and SNMPv2c (2) trap sizes are the following:

$$L_{\mathrm{trapSNMPv1}} = 37 + L_{\mathrm{Community}} + L_{\mathrm{Enterprise}} + n.(3 + L_{\mathrm{OID}} + L_{\mathrm{value}}) \tag{1}$$

$$L_{\mathrm{trapSNMPv2c}} = 63 + L_{\mathrm{Community}} + L_{\mathrm{trapOID}} + n.(3 + L_{\mathrm{OID}} + L_{\mathrm{value}}) \tag{2}$$

The messages values are generally encoded in binary formats, especially numerical values, which results in very compact messages.



```
⊟ sNMPv2-Trap
    request-id: 996832146
    error-status: noError (0)
    error-index: 0
  ⊟ variable-bindings: 5 items
    ⊞ 1.3.6.1.2.1.1.3.0: 458493 UpTime     Link Down
    ⊞ 1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.6.3.1.1.5.3
    ⊞ 1.3.6.1.2.1.2.2.1.1: IfIndex
    ⊞ 1.3.6.1.2.1.2.2.1.7:
    ⊞ 1.3.6.1.2.1.2.2.1.8:
```

Figure 7. SNMP trap indication.

WBEM indications are sent as an *ExportIndication* message, are encoded according to the CIM-XML standard and are transported by an HTTP message. Fig. 8 illustrates a CIM indication that reports a link-down event. Indication instances are encoded as CIM objects containing several properties, each one encoded in XML and containing a pair of tags. WBEM Indications encoding is very verbose, as can be seen in Fig. 8, mainly due to the XML technology. Each XML property defines its data type, and all values are encoded in text.

```
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="1010" PROTOCOLVERSION="1.0"> <SIMPLEEXPREQ><EXPMETHODCALL NAME="ExportIndication"><EXPPARAMVALUE
NAME="NewIndication"><INSTANCE CLASSNAME="RT_TestIndication"><PROPERTY NAME="IndicationIdentifier"
TYPE="string"><VALUE>63486176495792572</VALUE></PROPERTY><PROPERTY NAME="IndicationTime"
TYPE="datetime"><VALUE>20111018180135.792583+060</VALUE></PROPERTY><PROPERTY.ARRAY NAME="CorrelatedIndications"
TYPE="string"><VALUE.ARRAY></VALUE.ARRAY></PROPERTY.ARRAY><PROPERTY NAME="IndicationDescription"
TYPE="string"><VALUE>eth0: link-down</VALUE></PROPERTY><PROPERTY NAME="MethodName"
TYPE="string"><VALUE>generateIndication</VALUE></PROPERTY></INSTANCE></EXPPARAMVALUE></EXPMETHODCALL></SIMPLEEXPR
EQ></MESSAGE></CIM>
```

Figure 8. CIM-XML indication used in WBEM tests.

The size of a WBEM indication message can be defined as given by equation (3); taking the individual component sizes into consideration, it can be simplified to (4):

$$L_{WBEMNotif} = L_{WBEMEnv} + L_{InstEnv} + \sum\left(L_{PropHead} + L_{PropName} + L_{PropValue} + L_{PropFoot}\right) \qquad (3)$$

$$L_{WBEMNotif} = 235 + 52 + 5.\left(40 + L_{PropName} + L_{PropValue} + 20\right) \qquad (4)$$

NETCONF notification message (Fig. 9) is encoded within a 659 byte SOAP envelope and contains the notification object. The SOAP envelope contains a header that includes several XML schemas defining the indication structure and a body that comprises the notification. An envelope composes the notification object and a set of XML encoded properties per notification attribute.

```
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:ns3="urn:ietf:params:xml:ns:netconf:soap:1.0" xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"><SOAP-ENV:Body><ns2:notification
xsi:type="ns2:NotificationType"><ns2:eventTime>2011-10-23T10:58:24Z</ns2:eventTime><ns2:info>link
up</ns2:info></ns2:notification></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Figure 9. NETCONF notification.

The inclusion of the XML Schema represents 381 bytes, having a considerable weight in the SOAP envelope since it represents 61% of the notification message. NETCONF notification sizes can be calculated from equation (5), while the notification properties size can be calculated from expression (6). Unlike the SOAP envelope, the notification properties are efficiently encoded: they are composed by the property name and the text formatted property value and a set of XML tags with 13 bytes.

$$L_{Notification} = L_{SOAPEnv} + L_{NotificationEnv} + \sum L_{Field} \qquad (5)$$

$$L_{Field} = 13 + L_{PropName} + L_{Data} \tag{6}$$

Considering (5) and (6), and taking into consideration the individual component sizes, the size of a Notification message can be calculated by expression (7):

$$L_{Notification} = 452 + 69 + 2.(13 + L_{PropName} + L_{Data}) \tag{7}$$

Although sharing the XML technology, NETCONF performs a more efficient encoding than WBEM. As observed in [22], the main difference in the technology encoding strategies is that WBEM includes a property description inside each instance property, while NETCONF includes a schema describing the structure of each instance. Unlike what was observed in the configuration scenario experiments described in [22], the performance obtained by the NETCONF technology does not show to a big improvement when compared to WBEM. In configuration provisioning scenarios, the amount of information transferred from the network manager to the network elements tends to be higher and, as the number of configuration elements increases, the NETCONF encoding efficiency improves.

5.3 Network traffic

Network traffic analysis will allow us to evaluate the overhead that the different management technologies impose to the management solutions. In this analysis, we have gathered data regarding the number of packets and number of bytes transferred by the monitoring operations.

The analysis of Fig. 10 and Fig. 11 show what was already observed in some studies related to evaluating the performance differences between binary and XML-based technologies.

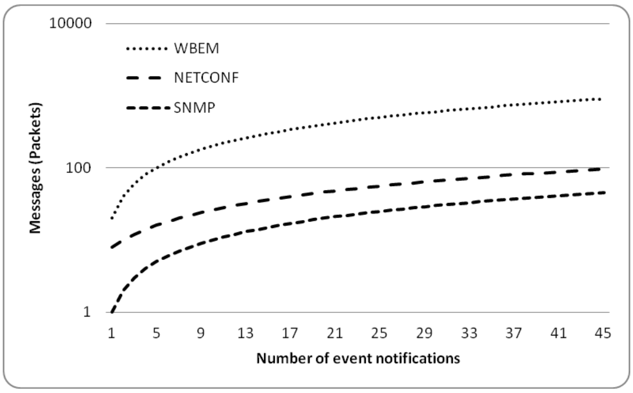


Figure 10. Number of packets transferred.

WBEM transfers the biggest volume of data, measured both in packets and bytes, followed by NETCONF and, finally, by SNMP. WBEM initiates one TCP session per indication, while NETCONF has only one session per subscription reducing the overhead of the TCP protocol. Finally, SNMP presents the best values because (i) it uses UDP as the transfer protocol, which has a very small overhead when compared to TCP; (ii) and uses a binary encoding format that is also lighter than the XML-based protocols. Additionally, and contrary to the SNMP Inform operation, SNMP trap does not require any confirmation from the SNMP manager.

WBEM is severely penalized by the high verbosity of CIM, showing a high growing rate of the response times as a function of the number of indications that are sent. Meanwhile, NETCONF presents better results due to the comparatively small amount of data that it has to transfer, a direct consequence of the less verbose YANG encoding language. Finally, SNMP has the best results regarding this performance metric due to the UDP use and the small amount of data that is transferred.

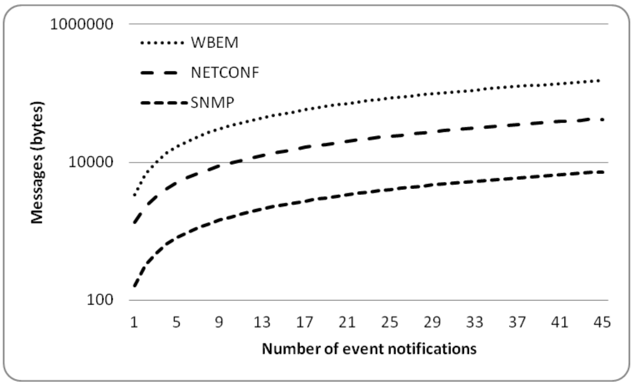


Figure 11. Number of bytes transferred.

5.4 Response times

In order to measure the response times of the protocols, bash scripts were used to generate an increasing number of event notifications. The results obtained from the measured data were plotted in a graph and are presented in Fig. 12.

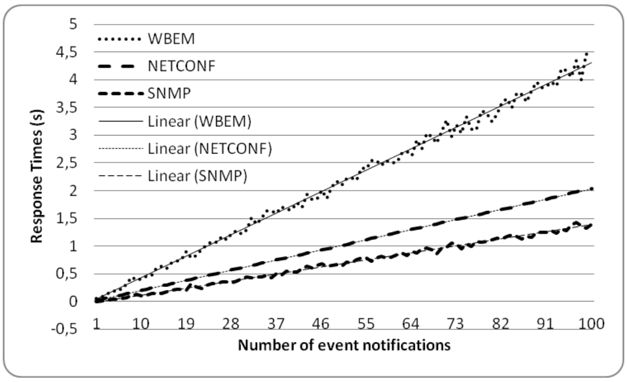


Figure 12. Response times.

A detailed analysis of Fig. 12 shows a considerable variation on the obtained time values. This variation is due to the fact that other processes were also running in the computer where the agent was executed during the test execution. The response times are relatively small, which amplifies the effect caused by the execution of other system processes.

Despite this noise effect, it is possible to draw a trend line and observe a linear increase of the response times with the number of event notification, for each technology. SNMP shows the lowest response time, due to its encoding simplicity and the fact that it just comprises a message with no acknowledgement. WBEM shows the worst response time due to the complexity of its encoding and because its indication requires an acknowledge from the management station to the network agent.

5.5 Discussion

The results obtained corroborate the expected performance differences between binary and XML encoding data transfers. XML uses tags, which implies more verbosity and data to be transferred. The overhead imposed by the management of TCP sessions was also predictable, being even more significant when transferring small amounts of data. Regarding these performance metrics, the conducted tests clearly indicate that SNMP presents the best performance: it sends less packets and bytes and takes less time to do it. Among the XML approaches, WBEMs CIM does an extensive use of tags, severely penalizing its performance when compared to NETCONFs YANG/YIN. NETCONF uses persistent TCP connections, thus reducing the session management overhead and improving its performance.

Table 2. Test results.

Metric	WBEM	NETCONF	SNMP
Timestamp (ms)	25	20	17
Packets	4	2	1
Event description size (byte)	15	7	25
Encoded data size (byte)	40	27	42
Transport overhead (byte)	192	192	44
Total Information (byte)	3374	1347	161
Efficiency (Encoded object size/Total information)	1.19%	2.00%	26.09%

6. Conclusion

Autonomic network monitoring and management architectures will allow the network to detect, diagnose and repair failures autonomously, as well as to adapt its configuration and optimize its performance and quality of service parameters. In order to choose the most appropriate solutions for specific network/services scenarios, it is crucial to know the potential of existing management technologies.

The first part of this paper documented the implementation of a network-monitoring agent. NETCONF over SOAP transport was chosen and the implementation was based on the gSOAP platform. The gSOAP platform automatically generates the communication interfaces code based on the provided data models, which greatly facilitates the development process. Several functional tests were conducted, as well as tests to evaluate the notifications transmission time and the signaling traffic between manager and agent. The time to produce notifications seemed to be appropriate, although the message size was enormous. NETCONF encoding imposes a very considerable amount of overhead on monitoring signaling: although it was somehow a predictable result, the amount of overhead was very impressive. By adding the contribution of all components related to the NETCONF technology, we could easily achieve a percentage of signaling due to NETCONF higher than 80%. The SOAP envelope size and its weight in the overall signaling (52%) represents a very considerable overhead for the management platform. This effect could be mitigated by the use of a tailored SOAP implementation, allowing slightly better results.

The second part of the paper presented a comparative study between different management technologies (SNMP WBEM and NETCONF) in terms of their scope, flexibility, market adoption and performance (measured by metrics like signaling overhead, memory requirements, coding efficiency and response times), by conducting a series of tests on representative prototypes. Although SNMP presents better performance results, regarding efficiency and response times, its functionality is clearly outdated by things like the lack of support for bigger configurations or the complex security features. In spite of the CIM verbosity, WBEM has no way to distinguish between status and configuration data, has no support for multiple data stores or transactions. The conducted tests confirmed a better performance of NETCONF when compared to WBEM. Although the tests usually used small event notification messages, reference [16] shows that the SNMP performance rapidly deteriorates when it has to transfer bigger messages. However, in event reporting scenarios messages have small sizes, which avoids WBEM and NETCONF to take advantage of their TCP transport gains. NETCONF provides a wide range of capabilities, like the support for resource-constrained devices, giving it an advantage to manage heterogeneous networks or to implement autonomous systems. Although NETCONF is intended to satisfy a wide range of capabilities that are expected from current network management and monitoring, its adoption does not depend only on its capabilities. It needs the attention of equipment vendors and developers, that should timely provide implementation support and quality data models. NETCONF is a young protocol, still needing a lot of standardization effort, but it seems to have the potential to overcome technologies like WBEM and become the de facto technology for configuration management and monitoring.

References

[1].J. J. P. C. Rodrigues and P. A. C. S. Neves, "A survey on IP-based wireless sensor network solutions", International Journal of Communication Systems, vol. 23, pp. 963-981, 2010. http://dx.doi.org/10.1002/dac.1099

[2].L. M. L. Oliveira, A. F. de Sousa, and J. J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks", International Journal of Communication Systems, vol. 24, pp. 1445-1466, 2011. http://dx.doi.org/10.1002/dac.1228

[3].J. Case, M. Fedor, M. Schoffstall, and J. Davin, Simple Network Management Protocol (SNMP), RFC 1157, 1990. Available at http://www.ietf.org/rfc/rfc1157.txt

[4].J. Schönwälder, SNMP over TCP Transport Mapping, RFC 3430, 1990. Available at http://tools.ietf.org/html/rfc3430

[5].D. B. Levi, P. Meyer, and B. Stewart, SNMPv3 Applications, RFC 2273, 1998. Available at http://www.ietf.org/rfc/rfc2573.txt

[6].J. P. Thompson, "Web-based enterprise management architecture", Communications Magazine, IEEE, vol. 36, pp. 80-86, 1998. http://dx.doi.org/10.1109/35.663331

[7].DMTF, "Common Information Model (CIM) Specification - Version 2.28", 2011. Available at http://dmtf.org/standards/cim/cim_schema_v2280

[8].DMTF, Web Services for Management (WS-Management), 2005. Available at http://dmtf.org/standards/wsman

[9].I. Sedukhin, Web Services Distributed Management: Management of Web Services 1.0, 2005. Available at http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0.pdf

[10]. R. Enns, NETCONF Configuration Protocol, RFC 4741, 2006. Available at http://tools.ietf.org/html/rfc4741

[11]. M. Bjorklund, YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), RFC 6020, 2008. Available at http://tools.ietf.org/html/rfc6020

[12]. H. M. Tran, I. Tumar, and J. Schönwälder, "NETCONF Interoperability Testing", Proceedings of the 3rd International Conference on Autonomous Infrastructure, Management and Security: Scalability of Networks and Services, Enschede, The Netherlands, 2009. http://dx.doi.org/10.1007/978-3-642-02627-0_7

[13]. S. M. Yoo, H. T. Ju, and J. W. Hong, "Performance Improvement Methods for NETCONF-Based Configuration Management", in Management of Convergence Networks and Services, Ed. Springer Berlin/Heidelberg, 2006, pp. 242-252. http://dx.doi.org/10.1007/11876601_25

[14]. T. Franco, W. Lima, G. Silvestrin, R. C. Pereira, M. Almeida, L. Tarouco, L. Granville, A. Beller, E. Jamhour, and M. Fonseca, "Substituting COPS-PR: an evaluation of NETCONF and SOAP for policy provisioning", Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06), 2006, pp. 195-204, London, Ontario, Canada. http://doi.ieeecomputersociety.org/10.1109/POLICY.2006.35

[15]. P. Gonçalves, J. L. Oliveira, and R. Aguiar, "A study of encoding overhead in network management protocols", International Journal of Network Management, 2012. http://dx.doi.org/10.1002/nem.1801

[16]. G. Moura, G. Silvestrin, R. Sanchez, L. Gaspary, and L. Granville, "On the Performance of Web Services Management Standards - An Evaluation of MUWS and WS-Management for Network Management", 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007), Munich, Germany, 2007, pp. 459-468. http://dx.doi.org/10.1109/INM.2007.374811

[17]. A. Chourmouziadis and G. Pavlou, "Efficient web services event reporting and notifications by task delegation", 18th IFIP/IEEE International Workshop on Distributed Systems - Operations and Management, San José, CA, USA, 2007. Available at http://dl.acm.org/citation.cfm?id=1783374.1783406

[18]. R. Neisse, R. Vianna, L. Granville, M. Almeida, and L. Tarouco, "Implementation and bandwidth consumption evaluation of SNMP to Web services gateways", IEEE/IFIP Network Operations and Management Symposium (NOMS'04), 2004, pp. 715-728, vol. 1. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.6555

[19]. 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.

[20]. 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. http://dx.doi.org/10.1109/MCOM.2004.1316533

[21]. W. Lima, R. Alves, R. Vianna, M. Almeida, L. Tarouco, and L. Granville, "Evaluating the Performance of SNMP and Web Services Notifications", 10th IEEE/IFIP Network Operations and Management Symposium (NOMS'06), 2006, pp. 546-556. http://dx.doi.org/10.1109/NOMS.2006.1687583

[22]. L. Andrey, O. Festor, A. Lahmadi, A. Pras, and J. Schönwälder, "Survey of SNMP Performance Analysis Studies", International Journal of Network Management, n. 6, vol. 19, 2009, pp. 527-548. http://dx.doi.org/10.1002/nem.729

[23]. J. Schönwälder, A. Pras, M. Harvan, J. Schippers, and R. van de Meent, "SNMP Traffic Analysis: Approaches, Tools, and First Results", 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 323-332. http://dx.doi.org/10.1109/INM.2007.374797

[24]. DMTF, Specification for CIM operations over HTTP version 1.2, 2007. Available at http://www.dmtf.org/sites/default/files/standards/documents/DSP200.html

[25]. J. P. Britton and A. N. deVos, "CIM-based standards and CIM evolution", IEEE Transactions on Power Systems, n. 2, vol. 20, pp. 758-764, 2005. http://dx.doi.org/10.1109/TPWRS.2005.846202

[26]. DMTF, CIM Event Model White Paper - DSP0107, 2003. Available at http://dmtf.org/standards/published_documents

[27]. W. Vambenepe, Web Services Distributed Management: Management Using Web Services (MUWS 1.0), 2005. Available at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm

[28]. V. Perelman, J. Schoenwaelder, and M. Ersue, Network Configuration Protocol for Constrained Devices (NETCONF Light), draft-schoenw-netconf-light-00.txt, 2011. Available at http://tools.ietf.org/html/draft-schoenw-netconf-light-01

[29]. M. R. Head, M. Govindaraju, A. Slominski, L. Pu, N. Abu-Ghazaleh, R. van Engelen, K. Chiu, and M. J. Lewis, "A Benchmark Suite for SOAP-based Communication in Grid Web Services", Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, 2005. http://doi.ieeecomputersociety.org/10.1109/SC.2005.2

[30]. M. Govindaraju, A. Slominski, K. Chiu, P. Liu, R. van Engelen, and M. J. Lewis, "Toward characterizing the performance of SOAP toolkits", Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, 2004, pp. 365-372. http://dx.doi.org/10.1109/GRID.2004.60

Copyright Disclaimer