An Evaluation of Service Composition Technologies Applied to Network Management

Ricardo Lemos Vianna, Everton Rafael Polina, Clarissa Cassales Marquezan, Leandro Bertholdo,
Liane Margarida Rockenbach Tarouco, Maria Janilce Bosquiroli Almeida, Lisandro Zambenedetti Granville
Institute of Informatics – Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brazil
Email: {rvianna, polina, clarissa, berthold, liane, janilce, granville}@inf.ufrgs.br

***Abstract*— Service composition is a technique that may help the development of management systems by aggregating smaller services to produce more sophisticated ones. Service composition can be realized by using traditional management technologies, although these technologies have not been conceived taking composition support as one of their main aspects. Current service-oriented architecture (SOA)-related efforts, however, define specific standards for Web services composition, such as the Web Services Business Process Execution Language (WS-BPEL). Web services for network management have been investigated by the management community at least in the last four years, but up to today no research evaluating Web services composition applied to network management has been carried out. In this paper we present such an evaluation where compositions based on the IETF Script MIB, *ad-hoc* Java Web services, and WS-BPEL are compared against one another in a managed network where BGP routers are investigated in order to identify route advertisement anomalies.**

I. Introduction

Service composition [1] is a technique used to aggregate or combine services in order to build up new, more sophisticated ones. It is also a core element of the service-oriented architecture (SOA) [2], which in its turn is the key architecture for the modern Web-based systems. As a technique, service composition can be used to address problems of several computer science disciplines, including network management, where composition is especially interesting when a complex management process requires the execution of smaller activities in order to be successfully accomplished. For example, to track the number of routes an autonomous systems (AS) advertises through its different routers, a composition that combines the routers' information exposed by their management agents is required, so one can be able to detect, for example, possible anomalies on an AS behavior.

Service composition itself is not new in computer science, but the efforts towards the definition of standards for service composition have initiated only around the last five years. With the lack of proper standards, service composition in network management has been manually realized using traditional management technologies, but not without heavy coding efforts usually combined with low flexibility. This is so because network management technologies have no "native" support for service composition on their core components, forcing composition to be implemented via particular solutions.

The current researches and standards for service composition are mainly focused on coordinating the interactions among Web services deployed along the Internet [3] [4] [5]. One of these standards - WS-BPEL (Web Services Business Process Execution Language) [6] - is strongly based on the workflow approach to provide properly orchestrated communications of Web services participating in a composition. One of the most important aspects about such standards, and particularly about WS-BPEL, is that they allow the definition of compositions in an easier and more proper way when compared with the *ad-hoc* compositions that have been carried out so far in network management.

This ease of use, however, is achieved with the price of increased processing delays and additional network bandwidth consumption, due to extensive exchange of XML-based messages. Considering the network management field, Web services-based management is not a new research area, but up to today there is no investigation that has determined whether and how the service composition standards could improve the composition of management services, replacing the composition solutions normally used in network management. We believe that Web services composition can really bring interesting opportunities for network management, but at the same time, possible drawbacks can prevent its use. The main contribution of this paper thus relays on the evaluation of service composition solutions for network management which we have carried out in order to clarify and understand the pros and cons of employing Web services composition for network management.

The remainder of this paper is organized as follows. In Section 2 a review of service composition in the context of network management is presented. Additionally, in Section 2 we also briefly introduce the WS-BPEL standard. Our evaluation has been carried out considering a country-wide backbone where BGP routers need to be investigated to detect route advertisement anomalies. This management environment and the target composition associated to it are presented in Section 3. The investigated service composition has been modeled and implemented considering three different approaches: compositions based on the IETF Script MIB, *ad-hoc* compositions of Web services management gateways, and compositions described in WS-BPEL documents. These composition approaches and their respective implementations

1-4244-0799-0/07/$25.00 ©2007 IEEE

are discussed in Section 4. In Section 5 we present a set of evaluating tests executed over the management environment of BGP routers. Tests and results are analyzed and discussed in order to draw the main conclusions of this paper, which is finally closed in Section 6, where final remarks and future work are presented.

II. Background

With the fast deployment of new services in networked environments, several management activities are initially manually performed by network administrators while no automation for such activities is supported in management software. The complexity of management activities may vary from simple queries directed to managed devices to complex calculations using information retrieved from different remote locations. In this last case, service composition can represent an interesting tool to build up more sophisticated services based on the combination of less complex ones.

As mentioned in the introduction section, service composition itself is not a new technique, and in fact can be realized using traditional management technologies. However, we believe that the employment of technologies specifically created to support service composition could bring important advantages to the network management discipline. In this section we first review how service composition can be implemented using traditional management technologies. After, we briefly introduce *ad-hoc* compositions to then close the section with the presentation of the WS-BPEL standard used in our evaluations.

A. SNMP and Service Composition

Probably, the most frequent service composition in network management occurs when network administrators code their personal bash scripts to perform an activity composed of smaller actions. Often, however, the results of the execution of such a composition are confined to the execution environment, and no other external software can use them to build up new compositions. We consider that proper compositions are characterized not only by the agglutination of smaller services to form a more complex one, but also by the ability of the composed service to expose its results to serve as the basis for the definition of additional and even more sophisticated services in a chain or hierarchy of compositions. In this sense, compositions made coding bash scripts cannot be considered proper compositions.

A more adequate option for service composition in network management is the use of the Simple Network Management Protocol (SNMP) [7] as a mechanism to expose the composed services. For example, RMON [8] and RMON2 [9] MIB objects expose compositions of management information collected by management probes located on dedicated devices or internal to routers and switches. In this case, SNMP is used only to expose the composed information, since the original information is retrieved not using SNMP but sniffing the network segments of interest.

In an all-SNMP composition solution, however, SNMP compositive agents can be coded to contact remote agents and combine the information retrieved from them. The results of such processing (i.e., the results of the composition) are then exposed to other higher-level agents also via SNMP. Astrolabe [10] and the work developed by Praveen Yalagandula and Mike Dahlin [11], for example, use SNMP-based compositions to build hierarchical levels of management information, where information from the leafs of the system are composed to express the whole status of the managed network. This approach resembles the management by delegation (MbD) model [12], where intermediate entities in a management hierarchy are dual-role: they are managers when accessing lower-level agents, and agents when exposing information for higher-level managers. These intermediate entities are usually referenced as mid-level managers in the management literature. Figure 1 depicts a set of cascading mid-level managers used to compose management services.

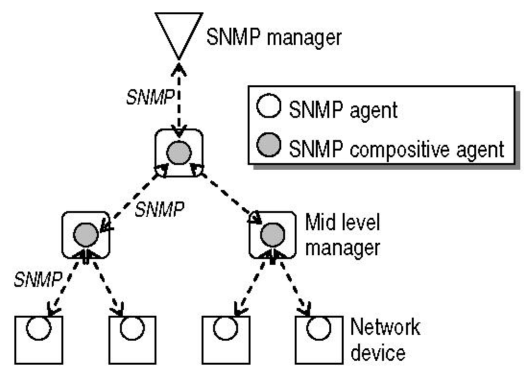

Fig. 1. Compositions coded in SNMP agents

A key problem of the previous approach is that if the composition needs to be changed for some reason (e.g., a different calculation is required in the mid-level managers), the SNMP agents need to be recompiled, which is usually expensive. A more flexible SNMP-based approach is the use of the IETF Script MIB [13]. In this case, SNMP is used as a script transfer and execution control mechanism. A network manager initially transfers via SNMP a script to mid-level managers that execute the management script using an internal runtime engine, such as a Java virtual machine or a TCL interpreter. Figure 2 shows the general approach when using the Script MIB.

It is important to notice that the composition logic in the Script MIB solution is now coded on the management scripts, instead of internally to the SNMP agents of the mid-level managers. Thus, the installation of new compositions requires only the transfer of new scripts, having no necessity of recompiling the SNMP agents anymore. Another important point is the fact that the selection of the language used to define the compositions depends on the execution environments available on the remote managers. Additionally, in order to have a hierarchy of service composition, the language used needs to have support for SNMP because when a script in execution needs to contact a remote SNMP entity it does so using the language's SNMP support. Considering this, building compositions in a large hierarchy with several levels of mid-level managers is not an easy task because, as mentioned before, SNMP-based technologies, including the Script MIB, have not been defined with composition in mind.

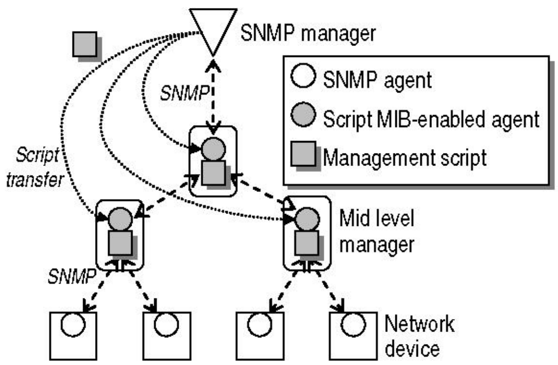

Fig. 2. Compositions based on the Script MIB

Recently, the IETF has been working on the definition of the XML-based NETCONF [14] protocol, devoted to network configuration. Although the employment of NETCONF would form a more elegant composition solution when combined with WS-BPEL (to be presented ahead), very few (if any) actual network devices support NETCONF. Since we are focused in evaluating the composition considering real scenarios, we won't address NETCONF in the remainder of this paper, even though we are aware of the NETCONF importance in the network management field.

B. *Ad-hoc Compositions*

We use the term "*ad-hoc* compositions" to address compositions manually coded using interpreted scripts or programming languages, and being based on no specific solution originally defined to support service compositions. In this paper, we also assume that *ad-hoc* compositions use SOAP (Simple Object Access Protocol) [15] as the protocol to communicate the compositive software with the basic services to be composed. An example of an *ah-hoc* composition is Web meta-search engines, i.e., engines that contact other ones to search for information and aggregate the results to provide a unified view of them to the user that requested the original search. Book search engines are an example of popular meta-search engines on the Web. In these systems, the communication between the meta-search engine and the third-party engines is based on SOAP, but probably not specified using a composition standard such as WS-BPEL.

An example of an *ad-hoc* composition in network management is presented in the XMLNET management system [16]. XMLNET is extensively based on XML, which used as the basic representation of management information. In order to integrate SNMP-enabled devices in the management system, XMLNET uses SNMP to XML gateways. The communication among the Web components of the system is performed using XML-RPC [17] instead of SOAP. XMLNET is developed in Java, and the service compositions are based on a non-standardized language defined by the system authors.

The advantage of the *ad-hoc* composition approach over the SNMP-based approaches presented before is that the use of SOAP as the communication mechanism is usually more appropriate for communications over the Internet. In addition, even for retrieving management information from network devices, SOAP performs better than SNMP if a large number of management variables is exchanged [18]. Although SNMP devices will not be replaced in a short-term by Web services-enabled devices, SNMP to SOAP gateways can effectively integrate SNMP devices in Web services-based management systems [19], thus allowing "legacy" devices to participate in a composition hierarchy using SOAP. Figure 3 presents a sample environment where SNMP devices are integrated in a Web services-based compositive system.

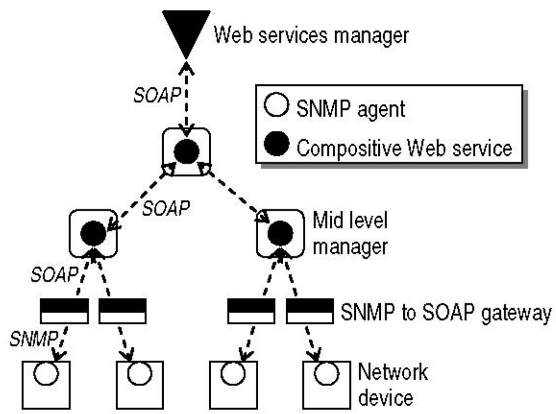

Fig. 3. Web services *ad-hoc* and WS-BPEL compositions

Although more interesting than the SNMP approaches for composition, the *ad-hoc* composition still presents the lack of flexibility usually required in dynamic management environments. If the composition needs to be somehow changed, the composition code needs to be rewritten (and recompiled if a programming language is used rather than a scripting language). This limits the applicability of service composition in more dynamic environment, and that in fact has motivated the development of composition-specific standards, such as WS-BPEL.

C. *Web Service Business Process Execution Language*

WS-BPEL (Web Services Business Process Execution Language) [6] is probably the most relevant and well accepted standard for Web services composition. Until version 1.1, WS-BPEL specification was called BPEL4WS (Business Process Execution Language for Web Services). When the working draft that proposed a new version (2.0) was released, it also changed the specification name to WS-BPEL. The standardization of WS-BPEL is currently under OASIS's arms, which released the newest specification draft in May, 2006. Our evaluation tests are based on BPEL4WS because the composition engine used (ActiveBPEL [20]) has not been updated by its developers to support WS-BPEL yet. Although we use BPEL4WS, WS-BPEL and BPEL4WS share the same main principles.

WS-BPEL models the behavior of a composition through an XML grammar that describes the logic needed to coordinate the services that participate in a process flow. That grammar is interpreted and its stated actions executed by a composition engine, such as ActiveBPEL, that coordinates the activities using a compensation strategy when errors occur.

Basically, WS-BPEL is a new layer built on the WSDL standard, where WSDL define operations, partners, and data types involved in the composition and WS-BPEL establishes how those operations will be sequenced. WS-BPEL supports basic and structured activities. Basic activities can be seen as a component that interacts with things externals to the own process, such as manipulating requests and replies or invoking external Web services. Structured activities, on the other hand, manage the entire process flow, specifying, for example, if some tasks should run sequentially or concurrently.

Using a composition standard one can compose services to create new services as if one was just modeling a workflow[1], in a higher level if compared to *ad-hoc* compositions. In WS-BPEL compositions, the user only needs to care about the logic of the new composed service, instead of worrying about the logic and how to implement it using a particular programming language and API (as in the *ah-doc* approach). In other words, implementation details are hidden from the user by using a composition standard. Other advantage brought by such standards is that they inherit all the advances resulted from previous workflow researches (e.g., formal semantic issues) since workflows and service compositions are very similar. Fault tolerance aspects are also covered by WS-BPEL standard. It has powerful mechanisms, such as roll back when some point of the composition fails, which allows one to create "transactional" services. The previously presented Figure 3 illustrates a WS-BPEL Web services composition for network management as well. Despite the different implementations, *ad-hoc* and standardized compositions share the same general architecture.

III. Management Environment

In order to evaluate the composition solution presented before, we have coded a set of compositions intended to manage some Brazilian internet exchange points (Brazilian IXPs - PPT-Metro project[2]) and their relationship with several autonomous system (AS) peers, in special with the country-wide Brazilian National Education and Research Network (RNP)[3]. Remote autonomous systems (ASes) connected to RNP constantly advertise BGP routes in 12 Internet exchange points (IXPs) located along the 27 RNP's points of presence (POPs). This environment needs to be managed because the same remote AS may advertise different routes in different IXPs, which leads to routing anomalies or peer-agreement violations.

Through the composition of routing and connectivity information obtained from IXPs, and using IPXs different routing views information it is possible to make several inferences about national Internet stability and growth. Moreover, such compositions can indicate regions in growth, if considered the increase of prefixes announced throughout the years in each IXP. Thus, based on the number of new routes advertised, it is possible to measure if the economy of a certain region is increasing or decreasing, and drive the government investments on the Internet initiative. Based on this kind of information it is also possible to establish quality levels for the Internet in each part of the country. For example, based on these established levels it is possible to firm SLAs with partners that will have to adjust their ASes to the quality level on that region.

By checking some BGP parameters and drawn prefixed on a IXP it is possible to measure regions that are suffering some disruption, like a dissemination of a computer virus, like those occurred in 2001 (CodeRedv2) and 2003 (W32.Slammer) that has shut down several minor ISPs around the planet, impacting on the global and national BGP table; that approach can measure the impact on Internet when accident or vandalism involving optical fiber disruption and another infrastructure problems happen. In essence, BGP-related information collected in the country-wide backbone drives the governmental investments on the national Internet initiative. Without the knowledge about the BGP advertisements, the investments may be guided towards wrong directions.

The management of BGP routes has been already addressed in the past. For example, Musunuri and Cobb [21] have investigated the divergences on AS tables and presented a survey listing possible solutions. Dimitropoulos and Riley [22], in turn, have presented an investigation on modeling AS relationships by simulating the Internet topology. We focus here on the necessity of monitoring remote ASes through different IXPs in order to detect possible anomalies. That is accomplished by management service composition.

In our solution, service composition for the management of the RNP's BGP border routers happens in two contexts. First, the composition of management information found in a single router is required to compute the number of routes a specific AS has advertised to a specific gateway. Another level of composition happens when information from different routers need to be aggregated to calculate the overall advertisement activity an AS is posing in the whole RNP backbone. Figure 4 depicts the managed environment highlighting the composition contexts.

Each BGP router may connect different ASes. Each AS, in its turn, may be connected to the RNP backbone through different BGP routers in different IXPs. A top-level manager is responsible for monitoring the advertisement patterns of each remote AS connected to RNP possibly via multiple IXPs. To do that, the top-level manager acts as a BGP monitor that contacts the mid-level manager of level 1 requesting a table of advertisement information for a giving AS. For example,

[1]In fact, there are graphical tools that aid users to create workflows and generate associated WS-BPEL documents

[2]http://www.ptt.br

[3]http://www.rnp.br

considering the network shown in Figure 4, the request of the advertisement table of AS number 3 would result in the Table I.

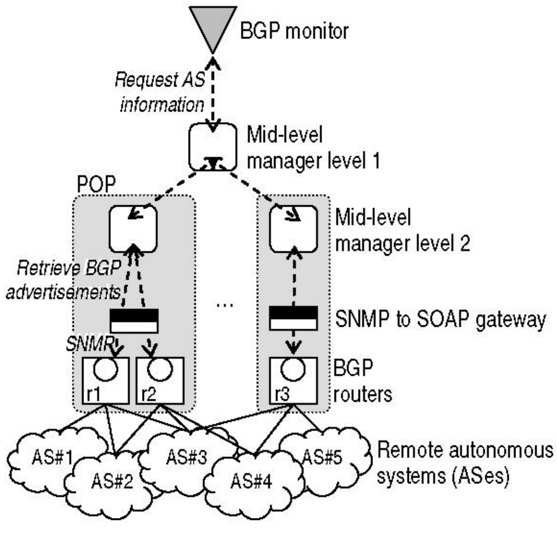

Fig. 4. RNP country-wide backbone with service composition

The difference in the number of advertisements sent to each BGP router may indicate, as mentioned before, routing anomalies that should be addressed. In order to produce the Table I output, the mid-level manager of level 1 composes the management information retrieved from mid-level managers of level 2 in each RNP POP that host one of the 12 IXPs. Mid-level managers of level 2, in turn, retrieve management information from the POP local routers accessing, via SNMP, the IETF BGP4 MIB [23]. If the composition implemented by the level 2 mid-level manager is based on Web services, then an intermediate SNMP to SOAP gateway is placed between the mid-level manager and the target device. If this is the case, POPs with more than one border BGP gateway share a single gateway to convert SNMP to SOAP messages. If the composition solution is solely based on SNMP, no gateway is required.

TABLE I

ADVERTISEMENT TABLE FOR AS NUMBER 3

AS♯3 advertisement table	
BGP router	Number of advertisements
Router r1	181
Router r2	663
Router r3	36

In the following section we describe three implementations used to investigate SNMP and Web services-based technologies for service composition applied to the management of the RNP's BGP routers advertisements.

IV. IMPLEMENTATION

In order to support the management of the previously presented environment, we have coded three solutions based, each one, on the Script MIB, on *ad-hoc* composition, and on WS-BPEL, respectively. All compositions perform operations in two levels, using the support of mid-level managers of levels 1 and 2, as shown in Figure 4.

At the bottom of the architecture, each level 2 mid-level manager contacts local BGP routers agents to retrieve the advertisements of an AS of interest. That is always performed via SNMP, since BGP routers do not natively support Web services-based management interfaces. Two objects of the BGP4 MIB are of special interest here: `bgpPeerRemoteAs` and `bgp4PathAttrPeer`. With proper treatment of these objects one can retrieve the list of advertisements associated to an AS.

The retrieval and manipulation of the values associated with `bgpPeerRemoteAs` and `bgp4PathAttrPeer` are performed by the level 2 mid-level manager. The result of this manipulation is a single pair listing the BGP router and associated number of advertisements. This composed information is exposed to the level 1 mid-level manager that performs the second service composition. The communication between level 1 and level 2 mid-level managers depends now on the composition solution. If Script MIB is used, SNMP is the communication mechanism employed. For *ad-hoc* and WS-BPEL compositions, SOAP is used instead.

Further details specific to each composition solution are presented in the following sub-sections.

A. Script MIB Composition Details

In order to support compositions based on the Script MIB, the managed environment needs to provide Script MIB-compliant agents at the mid-level managers, as well as an execution engine to run the compositive scripts. To implement the mid-level managers we have used Jasmin [24], which is an implementation of the Script MIB developed by the Technical University of Braunschweig and NEC C&C Research Laboratories. Jasmin implements the Script MIB published in the RFC 2592, which was later updated by the RFC 3165.

Jasmin supports both Java and TCL runtime engines, so that the top-level manager and the level 1 mid-level manager can delegate Java and TCL management scripts to the Jasmin-based mid-level managers. Our compositions have been coded in two Java scripts: one of them to be executed in the level 1 mid-level manager, and the second one to be placed in the level 2 mid-level manager.

Mid-level managers' software infrastructure is composed of Jasmin version 1.0.0, Java Development Kid 118, SNMP support provided by the ucd-snmp package version 4.2.6, and Linux Suse distribution 6.4 (2.2.14). Although newer versions of these softwares are available, the Jasmin software, which is not maintained by their developers anymore, imposes some restrictions on the versions of the other software packages used.

The compositive script at the level 1 mid-level manager requests the script on the level 2 manager to be executed setting the `smLaunchStart` Script MIB object. Than the level 1 mid-level manager loops consulting the level 2 mid-level manager waiting for the end of the script execution.

That is done checking the `smRunState` object. When its state evolves to `terminated`, the level 1 mid-level manager retrieves the result of the composition on the level 2 mid-level manager accessing the `smRunResult` object. In fact an SNMP trap message is issued to indicate that the execution of a management script is over. However, since SNMP traps are UDP messages not acknowledge by the receiving manager and UDP messages may get lost more easily in hostile network environments such as the one where our system is intended to run, the safer way to ensure that a script execution is over is by polling the remote mid-level manager.

B. Ad-hoc Composition Details

Ad-hoc compositions have also been coded in Java, but instead of being transferred to a Script MIB-based mid-level manager, they have been statically installed as a regular Java software. Since no special transfer mechanism is required, the *ad-hoc* composition software infrastructure is not limited by the previous Jasmin package requirements. On the other hand, since in the *ad-hoc* composition the communications are based on SOAP, proper SOAP support needs to be provided.

In order to build up the software infrastructure of a mid-level manager to support *ad-hoc* compositions, the following software has been installed: net-snmp version 5.1.1, J2SDK 1.4.2, Apache Tomcat 5.0.28, and Apache Axis 1.2RC2.

Since the final BGP router exposes the management information through the SNMP BGP4 MIB, an intermediate SNMP to SOAP gateway has been used. Such gateway has been automatically generated using a gateway creation tool [25] that we have developed for previous Web services for management investigations. The gateway presents Web services operations for each BGP4 MIB object, allowing a higher-level manager to access the BGP4 information by invoking such operations.

Although a gateway has been introduced, it has been physically placed on the same host that runs level 2 mid-level managers. When such manager wants to retrieve BGP information from an SNMP managed device it first contacts the local gateway via an internal SOAP call to the gateway, which then forwards the request now using SNMP. Figure 5 shows the physical placement of each manager in an *ad-hoc* composition setup.

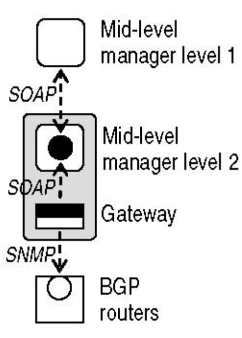

Fig. 5. Physical placement of mid-level managers and gateway

C. WS-BPEL Composition Details

As mentioned before, we have used ActiveBPEL to implement the composition support at mid-level managers. To install ActiveBPEL, the same software infrastructure used in the ad-hoc compositions has been used, with the additional installation of the AtiveBPEL itself, version 1.1.

The WS-BPEL compositions running on both level 1 and 2 mid-level managers have been specified with the use of a software tool named Network Information Aggregator (NIA) [26]. NIA helps network operator to define management services by using as information descriptor the traditional SNMP MIB modules.

The actual retrieval of management information from within the final BGP routers, again, requires the SNMP to SOAP gateways mentioned before.

V. Evaluation

The tests performed in this work aim at determining the response time and bandwidth consumption for each service composition implementation. To execute these tests we have deployed the mid-level managers presented before on the managed network and proceed with the measurements. Our experiments were carried out in a lab environment composed by two computers, connected via an 100Mbps switch, whose hardware setups for the top-level and mid-level managers are presented in Table II. It is important to mention that we intended to evaluate the possible solutions without the introduction of optimization. This question will be more apparent ahead.

TABLE II
TESTING HOST DESCRIPTION

	Top-level manager	Mid-level manager
Processor	AMD Athlon 2GHZ	AMD Athlon 2GHz
Cache	256KB	256KB
Memory	1GB	235MB
Swap	500 MB	800MB

Two different service composition levels has been observed in our evaluation, i.e., *device composition* and *network composition*. In the device composition, just one BGP router is contacted, and the number of advertisements of such router varies from 10 to 130. In network composition different services are contacted to form the more sophisticated one. In this case, the level 1 mid-level manager contacts all level 2 mid-level managers to build up an advertisement table given a specific AS of interest. In this last case we fixed the number of advertisements of each router to 10, but vary the number of mid-level managers from 1 to 10.

Figure 6 shows the environment setup to measured network usage and response time for the device composition.

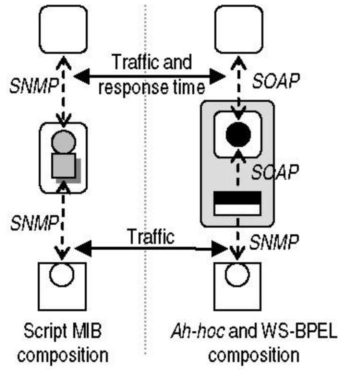

Fig. 6. Tests setup for device composition measurements

The environment for the measurements for the network composition is complementary depicted in Figure 7.

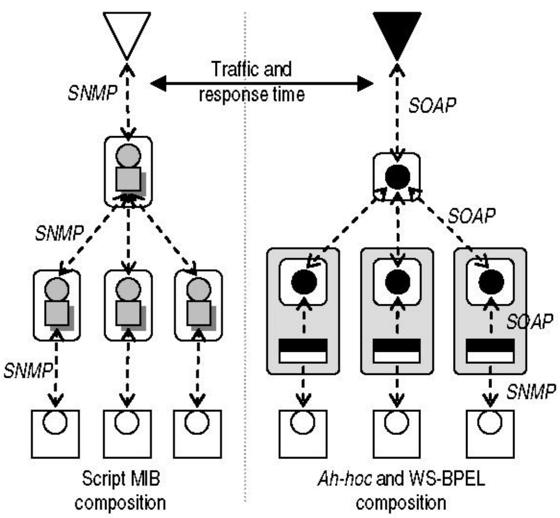

Fig. 7. Network composition tests

In the following subsections we present the results of this evaluation.

A. Network Usage

Network usage was measured in two different points for each service composition implementation. Regarding the Script MIB implementation, it was measured between the top-level manager and the mid-level manager, and between the mid-level manager and the final device's SNMP agent. Network usage for the *ad-hoc* and WS-BPEL service compositions has been measured between the Web services manager and the composition (mid-level manager), and between the SNMP to SOAP gateway and the SNMP agent (Figure 6).

Specifically on the Script MIB evaluation, the network usage includes the traffic introduced by the preparation of the script for execution, the traffic for monitoring the agent to detect the end of the script execution (using a polling operation of 10ms of interval), and the traffic generated to retrieve the execution results. The polling interval is 10ms because a previous software used by the RNP operators polled the remote entities using this interval. The time spent to transfer the script to the Script MIB agent was not computed because we considered that the script has been already deployed in the Jasmin agent. Furthermore, our measurements include all overhead from the lower layer protocols, i.e., transport, network, and data link layers. Figure 8 shows the network usage when retrieving 10 to 130 routes from a BGP router.

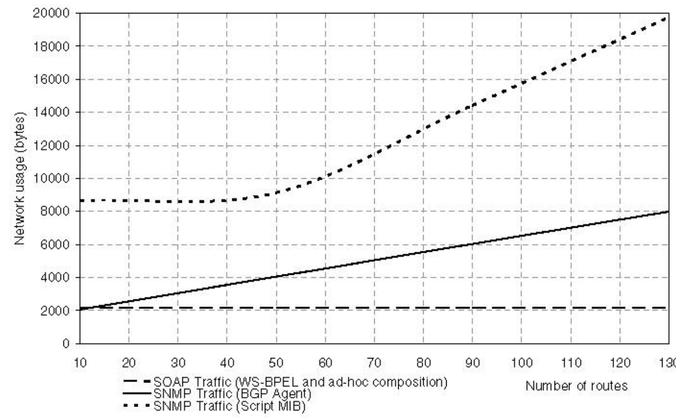

Fig. 8. Network usage for device composition

The solid line shows the SNMP traffic generated to retrieve the routing information directly at the BGP routers. Above the level 2 mid-level manager we have additionally measured the network usage imposed by the Script MIB, the *ad-hoc* composition, and the WS-BPEL composition. As one can observe, the Script MIB composition consumes more bandwidth due to the polling mechanism used by the Script MIB client to detect when the execution of the compositive script has finished. Both *ad-hoc* and WS-BPEL compositions, in turn, consumed a constant bandwidth regardless of the number of routes advertised. This is so because the composition executed in the level 2 mid-level manager reduces all routes retrieved from the BGP routers to a pair (router, number of advertisements), which consumes a fixed number of bytes to be transferred to the mid-level manager.

This network usage pattern is similar to the one previously published by Fioreze *et al.* [19]. Since the level 2 mid-level manager composes all advertised routes to produce a single pair of information, all SNMP traffic is confined to the mid-level manager and agent segment. The exception of this occurs when the Script MIB is used. Since there is no way for a Script MIB client (in this case, the level 1 mid-level manager) to safely learn that a script execution has finished except by polling the Script MIB agent, the bandwidth consumption will be increased.

Figure 9 presents the network usage considering the network composition. In this case, the traffic observed is that one between the top-level manager and the level 1 mid-level manager. Again, due to the polling mechanism used to access the Script MIB, the traffic generated in this composition is greater than in the other options. The traffic from the *ad-hoc* and WS-BPEL compositions is again constant.

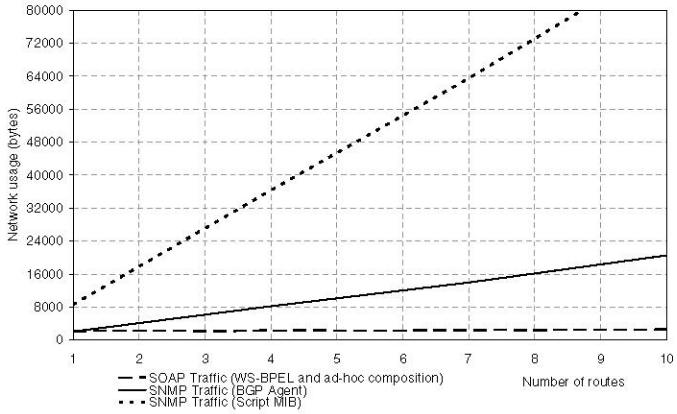

Fig. 9. Network usage for network composition

From these results we can conclude that, independently of using ad-hoc or WS-BPEL compositions, this Web services-based options are better than the solution based on the Script MIB. It is important to highlight, however, that this is most a consequence of the inability of the Script MIB of safely notify its clients about the end of a script execution.

B. Response Time

The response time is the time difference between the first message requesting an operation and the last message with the response associated to the request. For the device composition, the response time is observed between the level 1 and 2 mid-level managers. In this case, this level 1 manager requests to a single level 2 manager the number of advertisements a specific AS has issued in a BGP router. Internally, the level 2 manager sends several requests to the BGP router agents until the desired information is ready to be sent back to the level 1 mid-level manager. Figure 10 presents the response time associated to device compositions. In order to guarantee the statistic validation of the results, the experiments have been performed considering a confidence interval of 95% and running over 30 interactions.

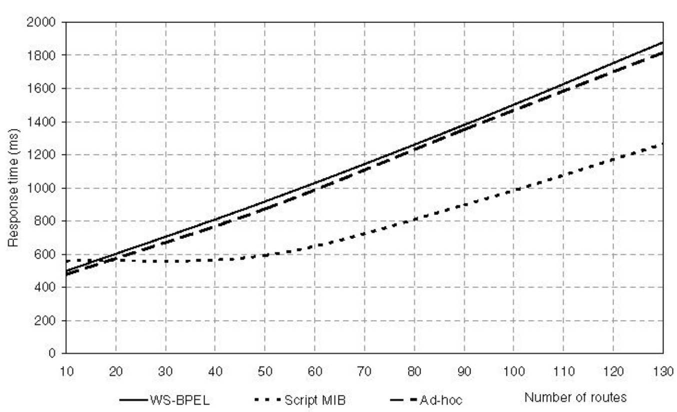

Fig. 10. Device composition response time

It is interesting to notice that the Script MIB composition often presents lower response time, although it also consumes more bandwidth as observed before. The low response time is due to low processing power required to process SNMP messages, which contrasts with the *ad-hoc* and WS-BPEL solutions, where more processing power is needed to handle all verbose, XML documents that form the SOAP messages.

Another interesting point comes from the difference between the response time associated to the *ah-hoc* and WS-BEPL compositions. Although WS-BPEL requires an additional execution engine to operate (in the case of our investigation the execution engine is provided by ActiveBPEL), no significant increase in the response time is observed when compared with the *ad-hoc* composition.

Figure 11 presents the response time now considering network compositions. Again, this response time has been calculated observing the communications between the top-level manager and the level 1 mid-level manager.

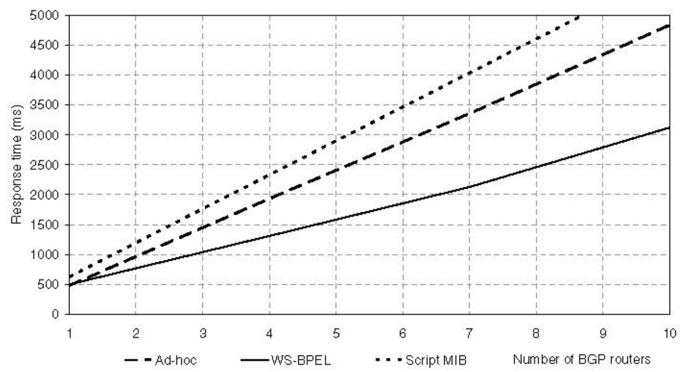

Fig. 11. Network composition response time

In the network response time, the Script MIB has presented worse performance than *ad-hoc* and WS-BPEL compositions. In this case it is so because we have fixed the number of routes in 10 and varied the number of BGP routers. For 10 routes, the Script MIB has performed worse than *ad-hoc* and WS-BPEL for device composition (look at Figure 10 again). This small difference in the device composition has been propagated, resulting in a more significant difference at the network composition level.

Notice that the performance of WS-BPEL is better than *ad-hoc* in this network composition. Here, this is the result of the native parallel requests issues by the WS-BPEL specification, which is something difficult to be achieved in *ad-hoc* composition because it involves explicit handling of processes and executing threads inside a usually simple code that defines the composition.

These final results make it clear that the final performance of a compositive hierarchy or chains depends not only on the composition technology used, but also on how the elements participating in such a hierarchy are implemented.

VI. Conclusions and Future Work

In this paper we have presented a study on the employment of composition technologies in network management. We have evaluated the use of traditional management technologies such as the IETF Script MIB, as well as technologies specifically defined to support workflow compositions, like WS-BPEL. Our evaluations have been executed considering a management environment composed of BGP routers that need to be monitored in order to detect anomalies related to the advertisement of BGP routes from remote autonomous systems.

We have considered three main technologies: Script MIB, which is a flexible IETF solution for the deployment of management script on remote managers in a possible hierarchy of managers; *ad-hoc* compositions often implemented on Web systems such as meta-search engines; and WS-BPEL, a recent standard devoted to the specific creation and support of Web services-based compositions.

Previously to our work, it was natural to believe that WS-BPEL – which requires a strong software infrastructure to be deployed – would perform poorer when compared with both Script MIB and *ad-hoc* compositions. However, from our evaluation results it is now evident that the performance issues of WS-BPEL compositions are not as critical as initially supposed. In addition to the performance results associated to it, WS-BPEL also has the advantage of being specifically created for service composition, thus more properly dealing with composition questions, such as native parallel execution support, better design and expressiveness of compositions, and an increasing set of tools available to automate service composition.

This paper concentrated on the performance issues of *ad-hoc* and WS-BPEL service composition for network management contrasting with the Script MIB traditional solutions. Future work of our research will address other aspects of these solutions, such as language expressiveness and scalability, which are as critical as the performance issue.

REFERENCES

[1] Francisco Curbera, Rania Khalaf, Nirmal Mukhi, Stefan Tai, and Sanjiva Weerawarana. The Next Step in Web Services. *Commun. ACM*, 46(10):29–34, October 2003.

[2] Steve Jones. Toward an Acceptable Definition of Service. *IEEE Software*, 22(3):87–93, May/June 2005.

[3] Boualem Benatallah, Quan Z. Sheng, Anne H.H. Ngu, and Marlon Dumas. Declarative composition and peer-to-peer provisioning of dynamic web services. In *18th International Conference on Data Engineering (ICDE'02), 2002.*, pages 297–308, 2002.

[4] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang. Qos-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, May 2004.

[5] Gerardo Canfora, Piero Corte, Antonio De Nigro, Debora Desideri, Massimiliano Di Penta, Raffaele Esposito, Amedeo Falanga, Gloria Renna, Rita Scognamiglio, Francesco Torelli, Maria Luisa Villani, and Paolo Zampognaro. The c-cube framework: developing autonomic applications through web services. In *DEAS '05: Proceedings of the 2005 workshop on Design and evolution of autonomic application software*, pages 1–6, New York, NY, USA, 2005. ACM Press.

[6] Alexandre Alves, Assaf Arkin, Sid Askary, Ben Bloch, Francisco Curbera, Yaron Goland, Neelakantan Kartha, Canyang Kevin Liu, Dieter Koenig, Vikesh Mehta, Satish Thatte, Dannhy van der Rijn, Prasad Yendluri, and Alex Yiu. Web Services Business Process Execution Language Version 2.0, May 2006. http://www.oasis-open.org/committees/download.php/18714/wsbpel-specification-draft-May17.htm.

[7] D. Harrington, R. Presuhn, and B. Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. Internet Engineering Task Force (IETF), RFC 3411, STD 62, December 2002.

[8] S. Waldbusser. Remote Network Monitoring Management Information Base. Internet Engineering Task Force (IETF), RFC 2819, STD 59, May 2000.

[9] S. Waldbusser. Remote Network Monitoring Management Information Base Version 2. Internet Engineering Task Force (IETF), RFC 4502, May 2006.

[10] Robbert Van Renesse, Kenneth P. Birman, and Werner Vogels. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Transactions on Computer Systems*, 21(2):164–206, 2003.

[11] Praveen Yalagandula and Mike Dahlin. A scalable distributed information management system. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 379–390, New York, NY, USA, 2004. ACM Press.

[12] Y. Yemini, G. Goldszmidt, and S. Yemini. Network Management by Delegation. In *International Symposium on Integrated Network Management*, pages 95–107, 1991.

[13] J. Schnwlder, J. Quittek, and C. Kappler. Building Distributed Management Applications with the IETF Script MIB. *IEEE Journal on Selected Areas in Communications*, 18(5):702–714, May 2000.

[14] R. Enns. NETCONF Configuration Protocol. Internet Engineering Task Force (IETF), RFC 4741, December 2006.

[15] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the Web services Web: an Introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing, Vol. 6, Issue 2*, pages 86–93, March/April 2002.

[16] Dimitris Alexopoulos and John Soldatos. Xmlnet: An architecture for cost effective network management based on xml technologies. *J. Netw. Syst. Manage.*, 13(4):451–477, 2005.

[17] Simon St. Laurent, Edd Dumbill, and Joe Johnston. *Programming Web Services with XML-RPC*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.

[18] A. Pras, T. Drevers, R. v.d. Meent, and D. Quartel. Comparing the Performance of SNMP and Web Services-Based Management. *IEEE eTNSM - eTransactions on Network and Service Management*, 1(2):11, December 2004.

[19] T. Fioreze, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco. Comparing Web Services with SNMP in a Management by Delegation Environment. In *IM 2005 - 9th IFIP/IEEE International Symposium on Integrated Network Management*, Nice, France, May 2005.

[20] ActiveBPEL Engine, 2006. http://www.activebpel.org.

[21] Ravi Musunuri and Jorge A. Cobb. An overview of solutions to avoid persistent bgp divergence. *Network, IEEE*, 19(6):28–34, 2005.

[22] Xenofontas Dimitropoulos and George Riley. Modeling autonomous-system relationships. In *20th Workshop on Principles of Advanced and Distributed Simulation, 2006. PADS 2006.*, pages 143–149, May 2006.

[23] S. Willis, J. Burruss, and J. Chu. Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2. Internet Engineering Task Force (IETF), RFC 1657, 1994.

[24] TU Braunschweig and NEC C&C Europe. Jasmin: A Script-MIB Implementation. http://www.ibr.cs.tu-bs.de/projects/jasmin/, 2003.

[25] R. Neisse, R. L. Vianna, L. Z. Granville, M. J. B. Almeida, and L. M. R. Tarouco. Implementation and Bandwidth Consumption Evaluation of SNMP to Web Sevices Gateways. In *NOMS 2004 - 9th IEEE/IFIP Network Operations and Management Symposium*, pages 715–728, Seoul, South Korea, April 2004.

[26] Ricardo Lemos Vianna, Maria Janilce Bosquiroli Almeida, Liane Margarida Rockenbach Tarouco, and Lisandro Zambenedetti Granville. Investigating web services composition applied to network management. In *Proceeding of the 2006 IEEE International Conference on Web Services (ICWS 2006)*, September 2006. To appear.