Evaluating Management Architectures for Internet of Things Devices

Marcelo Antonio Marotta[1], Cristiano Bonato Both[2], Juergen Rochol[1],
Lisandro Zambenedetti Granville[1], Liane Margarida Rochenbach Tarouco[1]
[1]Institute of Informatics – Federal University of Rio Grande do Sul (UFRGS)
[2] Department of Informatics – University of Santa Cruz do Sul (UNISC)
e-mails:{mamarotta,juergen,granville,liane}@inf.ufrgs.br, cboth@unisc.br

***Abstract*—The Internet of Things (IoT) is foreseen as a global network infrastructure that provides wireless communication among any kind of objects. One immediate challenge holds: how to manage these objects, considering that they may have limited computational resources. This management can be achieved through the use of gateways, *i.e.,* devices that intermediate wireless communications, minimizing resource consumption of the restrained objects. The communication between gateways and objects can be performed over many architectures. Among these architectures, we highlight the Simple Network Management Protocol (SNMP), the Service Oriented Architecture (SOA), and the Resource Oriented Architecture (ROA). However, there is a lack of deeper investigations to define which is the best architecture to model the communication between gateways and objects. Therefore, the main contribution of this paper is a quantitative evaluation of SNMP, SOA, and ROA as means for the communication between gateways and objects. Results analysis pointed ROA as the most interesting architecture to model the management communication.**

***Index Terms*—Sensor networks, Management architectures, Internet of Things**

I. Introduction

The Internet has evolved from the original network conceived to connect personal computers to a wireless network that has been intermediating the communications among diverse kinds of objects, from cellphones to household appliances. Often, the Internet serving as a communication means for these objects is referred to as the Internet of Things (IoT) [1]. The often mentioned promise of IoT encompasses the electronic access to anything, at any time, located at any place, by anyone, belonging to any network, and consuming as well as being consumed by any service [2]. Given the prediction that in 2020 the Internet will accommodate between 50 to 100 billion of connected objects [3], one immediate challenge holds: how to manage these billions of objects, specially considering that today, these objects have limited resources in terms of processing power, memory, and energy capacity.

Managing IoT objects naturally encompasses accessing such objects through communication protocols, using either direct or indirect communication approaches. The access for objects with sufficient resources is performed directly with the Internet using standardized technologies, *e.g.,* HTTP and IPv6, or proprietary ones, such as Electronic Product Code and CodeBlue protocol. For IoT objects with restrained resources, the access is often materialized through the intermediation of a gateway that connects IoT objects located in a local network with the global Internet. Once gateways are often not resource restrained, they can accommodate the communication load that cannot be supported by restricted IoT devices.

Three main management architectures could be considered to provide intermediate access between IoT objects and the Internet for management purposes: *(i)* Simple Network Management Protocol (SNMP), Service Oriented Architecture (SOA), and *(ii)* Resource Oriented Architecture (ROA). Pautasso *et al.* [4] provided a comparison between SOA and ROA architectures, but the authors did not focus on obtaining experimental results. Alshahwan and Moessner [5], on their turn, presented experimental results with SOA- and ROA-based frameworks for cellphones devices. However, no evaluation regarding indirect access to devices was provided. Dunkels *et al.* [6] suggested the application of SOA and ROA for gateway and devices communications, but a deeper evaluation of the proposed solution is missing and the authors did not provide any comparison with SNMP. Sehgal *et al.* [7] provided a deeper evaluation of SNMP and NetConf, *i.e.,* another management architecture for IoT object access, showing SNMP as a promising solution for managing restrained objects. Nevertheless, no comparison regarding SOA and ROA was provided. In summary, although SNMP, SOA, and ROA are key architectures to provide gateway and restrained IoT objects communications, there is a lack of deeper evaluations involving these three architectures.

In this paper, we present an evaluation comparing SNMP, SOA, and ROA architectures for IoT devices management intermediated by gateways. We investigate which architecture is more efficient to provide access to resource-limited IoT devices through a gateway in an IoT scenario. We carried out a set of experiments considering these architectures deployed in SUN SPOT devices [8]. The main contribution of this paper is to present quantitative results about the influence of device components (*i.e.,* device sensors) and network parameters (*i.e.,* gateway timeout and polling time). In our case studies we found that ROA response time is seven times faster than SOA with a similar performance as the SNMP that both consumes 32% less power to process a query than SOA. In addition, we evaluated these architectures with different network parameters, *i.e.,* timeout and polling time, resulting in analysis that defines ROA as the most interesting architecture to model communications between gateway and IoT objects.

The remainder of this paper is organized as follows. In Section II we review the state-of-the-art on architectures to SNMP, SOA, and ROA access to devices. A description of our experimental model is presented in Section III Results are analyzed in Section IV. Finally, we conclude and present future work in Section V.

978-1-4799-6606-6/14/$31.00 ©2014 IEEE

II. Background and Related Work

In this section, we first present a brief review about SNMP, SOA, and ROA. Afterwards, details of current work based on comparison involving these architectures are presented.

A. Background

Investigations have been carried out to determine the best architecture to provide access to IoT devices. Among these architectures, SNMP stands as one loosely coupled architecture to manager/agent that uses SNMP messages to access management information from devices [9]. In such an architecture, a device incorporate the role as an agent that held a Management Information Base (MIB) fulfilled with management information, *e.g.,* a sensor value or a network interface status. This information is represented by a Object Identifier (OID) that is gathered or edited through SNMP messages. Managers send to agents SNMP messages that encapsulate Protocol Data Units (PDUs) defining operations to be performed according to a OID. Other features of SNMP are: (*i*) uses UDP as the only transport protocol, (*ii*) messages are binary compressed to avoid network transmission overhead, and (*iii*) implements security and reliability on SNMP v3.

SOA, in its turn, is coupled to the client/server model that uses Uniform Resource Identifier (URI) to access the addresses where services are located. In general, this architecture is implemented using the W3C standard called Web Service (WS). Each WS requires a Web Service Description Language (WSDL) document to provide the communication interface, *i.e.,* description of services, message formats, and data, and Simple Object Access Protocol (SOAP) [10] for communication among services and users. SOAP messages are described in Extensible Markup Language (XML) and must be serialized before being transmitted on the network. Other features of WS are: (*i*) there is no definition of a specific application protocol, (*ii*) network nodes do not support caching of SOA messages natively, (*iii*) services can be extended with new functionalities without the creation of new services, and (*iv*) security is implemented through WS-security.

ROA is a loosely coupled architecture to client/server model that uses URI to directly access the resources of the devices. In general, this architecture follows the architectural style called Representational State Transfer (REST) [11]. This architectural style defines HTTP as the only application protocol and standardizes the interface access considering the methods of this protocol, *i.e.,* GET, PUT, POST, and DELETE. Each message of REST represents a state of the accessed resource, *i.e.,* the current collection of meaningful information, *e.g.,* network parameters and sensor measurements. REST state can be described by XML as SOAP or by JavaScript Object Notation (JSON), a lightweight description language. Other features of REST are: (*i*) network nodes support cache of ROA messages, (*ii*) resources are not extended instead new resources are deployed, and (*iii*) security and reliability are offered using secure HTTP.

B. Related Work

SNMP, SOA, and ROA are used in many contexts, *e.g.,* Internet, business, and industry [12]). These architectures offer interoperability and manageability among networked systems. However, each of then performs the offered management differently. Therefore, it is natural that comparisons of these architectures are not particular to the IoT context. For example, Pautasso *et al.,* [4] presented a comparative evaluation between the conceptual characteristics of REST and WSDL/SOAP. Although the number of features are similar, the application are different, *e.g.,* a REST implementation is less complex and demands less network and hardware resources, such as bandwidth, memory allocation, and processing power. However, the authors remain generic, *i.e.,* their work did not contemplate a quantitative experiment and was also based on statements that may not hold in an IoT context. Different from Pautasso *et al.,* [4], our evaluation compares SNMP, SOA, and ROA in quantitative terms, considering an IoT context.

Alshahwan and Moessner [5] presented a comparison between SOA- and ROA-based frameworks implemented in cellphones *i.e.*, Mobile Web Services Framework - MWSF. The experiments were performed using cellphones linked by their IEEE 802.11b network interface. The parameters evaluated were: (*i*) the response time by size of messages, (*ii*) processing time by number of messages competitors, (*iii*) memory consumption by size of messages, (iv) number of messages lost with concurrent requests, and (v) the maximum number of concurrent requests. In all evaluations the ROA-based framework demonstrated better results, *e.g.,* the lower consumption of battery, the smallest size of messages, and the greater number of concurrent requests without loss. However, this work did not implemented a SNMP-based framework to be compared with the others. In addition, this comparison can be influenced by the implemented framework being not fair to define if SOA is worst than ROA.

Sehgal *et al.,* [7] investigated the management of resource constrained devices for the IoT. The authors presented a comparison of performance between SNMP and NetConf placed on a restrained device. This device was installed with a Contiki OS that enabled the authors to perform hardware queries about network, memory, and processor usage during experiments involving communications with SNMP and NetConf. This work is particularly important, because the authors proved the superiority of SNMP against NetConf. Therefore, we did not explore NetConf as an access architecture to manage IoT devices in this paper. Nevertheless, Sehgal *et al.,* [7] have not considered the battery consumption during experiments. In addition, SOA and ROA were not explored as possible architectures to provide access for IoT objects management.

Dunkels *et al.* [6] presented an approach for IEEE 804.15.4 sensor networks, where IP-based nodes communicate via *Web Services*. Three comparative assessments of SOAP and REST can be highlighted as main results: (*i*) time, (*ii*) power consumption for access to the service control LED, and (*iii*) estimation of the duration of the battery. Although there were four sensors available in the proposed scenario, only one (LED lighting) was considered for evaluation purposes, which limited the sort of results that can be obtained. Therefore, in this paper, we propose to extend Dunkels *et al.* approach, by performing and analysis of the influence of all four sensors in the measurement results. Moreover, our evaluation considers further parameters that may influence in the results of the study, such as, timeout and polling time. Finally, the authors

did not consider SNMP in opposite to this paper that presents also a detailed comparison with this architecture.

III. Comparison Proposal

The comparison of SNMP, SOA, and ROA architectures requires the modeling of a typical IoT scenario to be performed. Therefore, a IoT scenario is proposed in subsection III-A. Afterwards, the definition of a performance comparison can be summarized in the metrics or circumstances that affect each architecture performance, *i.e.,* a comparative aspect. Thus, five comparative aspects are defined in subsection III-B. Finally, we present a prototype implemented for each architecture to perform the comparison among them in subsection III-C.

A. IoT Scenario

A IoT scenario was modeled to enable the assessment of the performance of IoT devices, when interconnected through a gateway, as can be seen in Figure 1. SNMP, SOA and ROA can be observed in the modeled scenario as access architectures between gateways and devices communication. The components and features that composes the modeled scenario are described as follow.

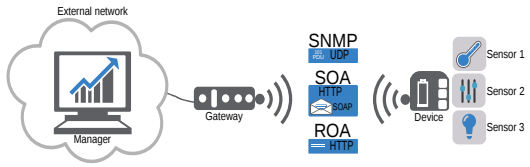

Figure 1: IoT scenario

- **Manager** is placed at an external network, *e.g.,* private local area network, the Internet, or a federative network. The manager has the function of injecting workload, monitor the experiments, and collect results.
- **Gateway** provides the bridge between a manager from an external network and the IoT devices. This bridge receives messages from the manager that are forwarded to the IoT device using the technology implemented in the connection link (*e.g.,* IEEE 802.15.4 and ZigBee). The gateway improves the interoperability between the external entities and IoT devices with different protocols, SNMP, SOAP, and HTTP.
- **Device** is characterized as a real example of an IoT device. One of the main features of these devices is the constrained amount of resources available, such as memory, processing capacity, and power. Moreover, these devices have a set of sensors, *e.g.*, movement, humidity, and blood pressure meter, integrated into it.

The modeled scenario can be adapted to fit in many different IoT contexts that includes, for example, an airport station [13], an hospital [14], or a city neighborhood [3]. This IoT scenario enables the comparison of SNMP, SOA, and ROA architectures deployed between gateways and IoT devices.

B. Comparative aspects

A comparison must be carefully defined according to different comparative aspect. Therefore, five comparative aspects are defined according to the evaluation metric used, methodology applied, and desired goals, described in details as follows.

1) *Response time* is the time spent to request and retrieve an information from a device. Such time impacts the user experience with the system, *i.e.*, the larger time the greater waiting to receive an answer from the device. Therefore, the response time was adopted in this work as a comparative aspect to measure the efficiency of each access architecture.
2) *Power consumption* is specially important for the IoT, because it is a metric that measures a fundamental resource for the operation of some devices, *i.e.,* the battery power. This resource determines the time that a device remains accessible and operant. This time decrease with the power consumption. The architecture adopted for management can increase or decrease the power consumption. In this case, we defined the battery consumption as a comparative aspect to be evaluated.
3) *Analysis of effect*, according to Jain [15] different factors may produce a vary effect that influence the performance of the analyzed architectures. For this work, factors may be defined as the different queried sensors of a device. Therefore, an effect of variation called replication with 2^k factor become one of the comparative aspect. This model evaluates the effect of each factor or relationship between factors that most influenced the final value acquired in the experiment, *i.e.,* which sensor impacted most the results.
4) *Gateway timeout* is a network parameter that plays an important role in the context of the IoT, because it prevents overloading of devices and improves the user experience. For example, many managers may decide to send requests to a device, if the timeout value is large, the device will fill its stack with processes and will drop the new arrived request without answer it. Thus, managers will have to wait all the timeout for finally receive an error message. Therefore, the larger timeout, the worst the manager experience, however, the lesser number of errors and vice versa, presenting a trade-off. The expected goal with this comparative aspect is the finding of thresholds for the timeout parameter for the SNMP, SOA, and ROA architectures.
5) *Manager polling time* is a management parameter to retrieve images of the current status of the device in a defined period. However, in scenarios that involve devices with scarce resource (*e.g.*, an IoT scenario), a wrong polling time may overload the devices with messages retrieval.

C. Prototype

The SNMP prototype to provide access for the devices was developed according to the open source project of Margulies, Siegl, and Toman [16], also called Multiplexor and dynamic MIB of an SNMP v1 agent. This implementation was modified to support sensor, battery, and memory values gathering, among other features.

For SOA prototype, we used the W3C documents to develop the WS placed on the device, providing WSDL document creation and SOAP over HTTP. It is important to make clear

that HTTP was chosen as part of the protocol stack in order to maintain SOA and ROA at the same layer, although SNMP was modeled to perform communication over UDP solely.

Finally, the implementation of Gupta *et al.* [17] was chosen to implement the ROA architecture prototype. The choice of each prototype framework and tools was based on previous usage and recurrence in the literature [5], [6], [4], [7].

In the next section, experimental results collected considering the modeled scenario and described prototypes are presented and discussed.

IV. Evaluation

The hardware and software used during experiments are described in this section. Afterwards, we evaluated and compared SNMP, SOA, and ROA architectures according to different comparative aspects. For this purpose, each evaluation is placed in a subsection ordered according to the aspects discussed in Subsection III-B.

A. Experimental environment

The proposed scenario of Subsection III-A was deployed with a computer with an Intel Core 2 Duo T5800 2Ghz with 4GB of main memory placed as the external manager. The gateway and the device were deployed with a development kit from ORACLE, *i.e.,* Sun^{tm} Small Programmable Object Technology, Sun SPOT [8]. This kit is composed of a base station and two devices (spots). Both, spots and base stations, are composed by a processing main board, called Espot. The Espot has a 400 MHz ARM processor AT91RM9200, 4 MB of memory flash, 512 KB of PSRAM, a IEEE 802.15.4 radio frequency transmitter, and a USB 2.0 port. However, different from the base station, spots have a battery 770mAhr Li-Ion rechargeable and a second plate connected to Espot *i.e.,* Edemo. The Edemo application board is connected to Espot to add new features to the spots. For this purpose, Edemo has a second processor to execute specialized input and output operations, ATmega 168V. The application board has components that can be accessed and modified by the routines implemented in the spots. These components are: 8 RGB LEDs, two push buttons, 3-axis accelerometer, thermometer, and light sensor.

Experiments were performed using the deployed scenario to measure the comparative aspects defined in Subsection III-B. In particular, for response time and battery consumption, we measured three type of requests: (*i*) a simple call, (*ii*) individual sensor, and (*iii*) sensor group request. On the former, a spot is communicated to inform the battery level, being classified as the simplest information to collect from a device. On the individual sensor type, each sensor from the spot is individually requested to calculate the performance influence of each sensor in comparative aspects. Finally, the group sensor request is measured to calculate the influence of the combination of different sensors being requested in a single message.

Replications of 30 experiments to access sensors embedded in the device varying SNMP, SOA, and ROA architectures were performed for each comparative aspect. For each replication performed, averages with an error interval of 95% of confidence level were calculated.

B. Response Time

The average response time of each architecture to perform requests and their respectively standard deviations are displayed in Figure 2. In the horizontal axis, different type of requests are displayed, *i.e.,* simple call, individual sensor, and group sensor requests. In addition, in the vertical axis, the response time measured during each request is presented in seconds. In this figure we can see that the average response time of SNMP and ROA architectures are considerably lower than the average response time of SOA. SNMP and ROA architectures show up seven times faster than SOA to perform a simple call request. For all the other sensors, ROA is at least four times faster than SOA, however, SNMP loses performance when the number of sensors requested raises. The SNMP behavior is explained by the need of multiple SNMP messages to gather all sensor values despite of SOA and ROA that need just one message.

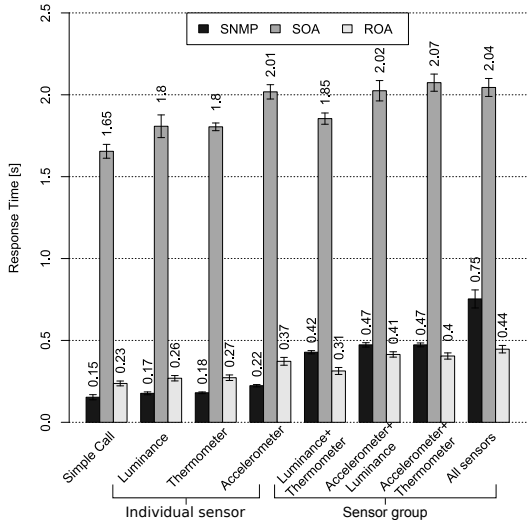

Figure 2: Response time of sensor requests

Considering the individual access to the sensors, the accelerometer has the highest response time for the three access architectures as it is depicted in Figure 2. Diverse parameters can influence the results related to the response time, mainly: (*i*) the number of messages needed to gather sensor values, (*ii*) the size of the messages involved, and (*iii*) the sensor being requested. The number and size of messages needed by SNMP, SOA, and ROA to perform a simple call and all sensor values gathering operations are shown in Table I.

SOA presents proportionally the larger message that is even greater than 3 messages of SNMP together. This proportion is very similar to the performance achieved in Figure 2. In addition, different from SOA and ROA, SNMP needs a total of six messages to process a request to all sensors from a device, prejudicing its performance considerably. It means that most of the results is more related to the amount of data exchanged over the network than the processing of each message. However, there is still doubt about the effect that different sensors may present within the variation between results. Therefore, an effect analysis (2^K factor [15]) was conducted and better explained in Subsection IV-D.

	SNMP		SOA		ROA	
	Messages	Size[Bytes]	Messages	Size[Bytes]	Messages	Size[Bytes]
Request Simple Call	1	40	1	461	1	63
Response Simple Call	1	46	1	473	1	72
Request All Sensors	3	106	1	461	1	63
Response All Sensors	3	139	1	490	1	83

Table I: Size of SNMP, SOA, and ROA messages

C. Power Consumption

In Figure 3, results gathered about the power consumption from each type of requests varying the used architecture are shown. The power consumption is presented in the vertical axis in averages [mAh] as well as the confidence interval for each architecture and type of requests in Figure 3. In addition, each different type of requests are shown in the horizontal axis. In addition, each request performed has a total time of four seconds independently of the time spent by each architecture to perform a sensor request, avoiding unfairness in the power consumption measurement. The power consumption results have a similar behavior as the response time, *i.e.* ROA performs better with mean of 32% lower power consumption than the SOA architecture. Also, ROA shown power results less expensive than SNMP as well. These results may be explained by the SNMP PDU decoding and encoding procedure as well as the need of maintaining a updated MIB.

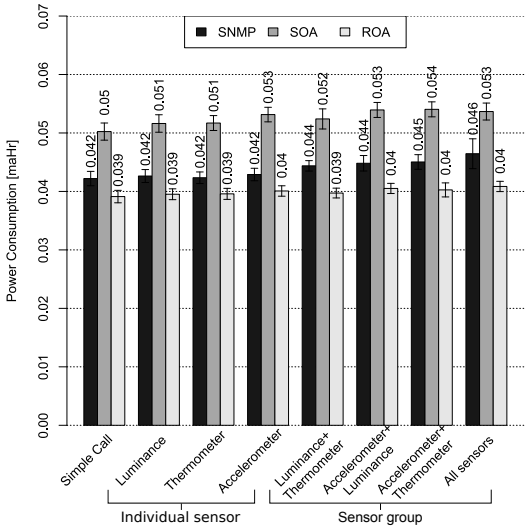

Figure 3: Power consumption of SOA versus ROA

D. Analysis of effect

A device may has many different embedded sensors that can have their measurements gathered by managers. The different performance achieved to retrieve this measurement may be better understood with an effect analysis calculation. Therefore, in Table II the effect of different sensors from a device in terms of response time influence are provided in ascending order. In addition, in Table III, an effect analysis is provided in terms of power consumption.

Analyzing Table II, for the three architectures, considering the individual access time, the accelerometer is definitely the costly sensor to gather values. On one hand, for ROA and SOA, the accelerometer will influence around 90% in the variation of performance between each request. On other hand, SNMP shown to be well balanced to gather information from all sensors being not very influenced by which sensor is being gathered. In addition, each sensor almost do not influence the results when combined showing that their combination is not related when communicated. However, SNMP still needs to perform three messages to gather all sensor values, which affects its response time performance.

SNMP						
A+L+T	A+T	L+T	A+L	Luminance	Thermometer	Accelerometer
1.49%	2.54%	2.61%	2.68%	25.73%	26.15%	38.81%
SOA						
A+T	A+L+T	L+T	Luminance	A+L	Thermometer	Accelerometer
0.0%	0.03%	0.20%	1.13%	1.51%	5.80%	91.34%
ROA						
A+T	A+L+T	L+T	Luminance	A+L	Thermometer	Accelerometer
0.0%	0.05%	0.20%	1.23%	1.77%	5.94%	90.82%

Table II: Analysis of the factors' effect in the response time

SNMP						
A+L	L+T	A+T	A+L+T	Thermometer	Luminance	Accelerometer
0.56 %	0.89%	2.68%	3.33%	23.86%	25.10%	43.59%
SOA						
A+L+T	A+T	A+L	L+T	Luminance	Thermometer	Accelerometer
0,23%	2,58%	2,75%	3,36%	6,16%	8,08%	76,84%
ROA						
L+T	A+T	A+L+T	A+L	Thermometer	Luminance	Accelerometer
0,06%	0,11%	0,91%	1,31%	7,76%	13,32%	76,52%

Table III: Effect analysis - power consumption

The effect analysis of factors in the Table III presents the accelerometer as the major factor of influence. However, it impacts less the power consumption variation than the response time, *i.e.,* the accelerometer influences the power consumption variation in 76% in relation to the other factors and 90% in the response time of previous experiments. However, for SNMP, the accelerometer request impact more intensively in terms of power consumption than the response time.

E. Gateway timeout influence

A summarization of the influence of the gateway timeout parameter in the response time for each of the three architectures is presented in Figure 4. In the horizontal axis, values of timeout are presented in log scale, ranging from 0 to 120s. In the vertical axis, the response time averages performed by each architecture are presented. The results are shown in averages of thirty concurrent simple call requests. For a great timeout, more than 60s, the three architectures presented a vary response time, where ROA converged to 2s, SNMP to 3.2s, and SOA to 24s of response time. The response time gets less variable whith a small timeout, a strange behavior that is better explained in Figure 5.

The small variation presented in the response time for a small timeout is easily explained by the number of errors occurring for each request. In Figure 5, the same timeout settings are presented by the horizontal axis. In addition, in the vertical axis, errors occurred during experiments are shown for each architecture. As can be seen, when the timeout is smaller than 1s, almost all the three architectures start to present a lot of request unsolved, returning error messages. However, an error message still being a response, which explains the low response time, even if SOA is involved. For SOA, no request is answered at a timeout smaller than 1.7s, whereas, SNMP faces the same problem at 0.2s, as well as ROA at 0.1s.

Analyzing the results found with timeout, concurrence explains the superiority of SNMP and ROA architecture to the

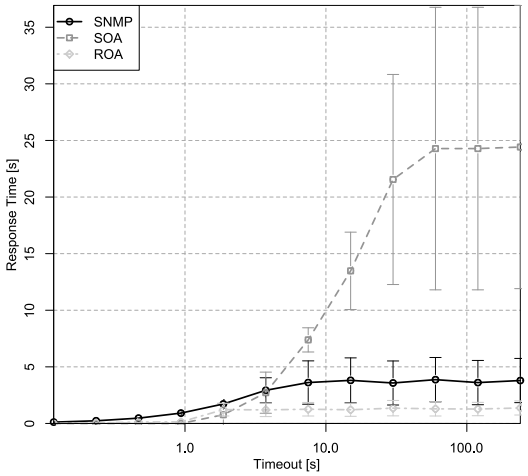

Figure 4: Gateway timeout influence

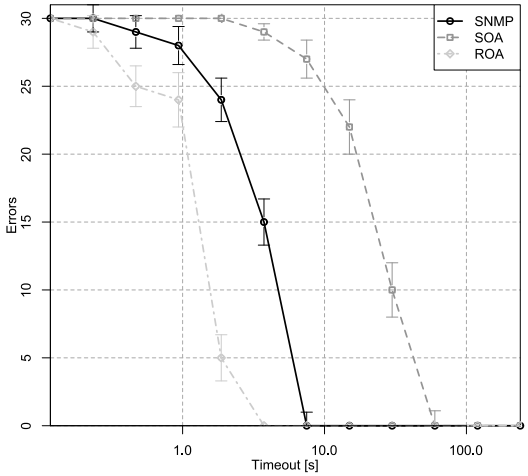

Figure 5: Errors during gateway timeout experiments

scenario presented. With a timeout of 4s, all thirty requests are met with success through SNMP and ROA. Moreover, with a timeout around 1s, more than 80% of the requests are met with success. For SOA architecture the timeout required to achieve thirty requests successfully can be met at 60s, about fifteen times the timeout necessary to ROA and SNMP to perform the same and with a timeout of 20s, the SOA error rate is still greater than 50%.

F. Manager polling time influence

We investigated the effect of different polling time during two minutes of experiment per replication in the response time for a simple call request that is depicted in Figure 6. Horizontal axis present the polling time settled during the experiments in log scale, whereas, the vertical axis presents the response time from each architecture. Analyzing the achieved results, when the polling time is smaller than the average response time from each architecture, the device becomes overloaded with requests, resulting in different performances for each architecture. In addition, given a certain threshold, the device shall discard the new arrived requests. This dismissal allows the device to answer a few requests back. However, the polling processing continues leading the device queues to be fulfilled, reproducing the dismissals. This behavior justifies the peaks and valleys presented in Figure 6. As can be seen, SNMP gradually converges to its average response time with the raising of polling time, as well as SOA and ROA. SOA, in its turn, presented the worst average response time during the polling experiment, achieving averages in the range of 1.94 to 9.80s whereas SNMP and ROA presented averages in the range of 0:32 to 2:48s.

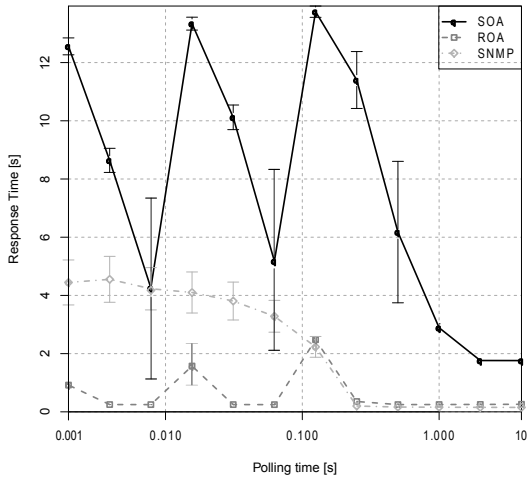

Figure 6: Response time versus polling time

The polling time is important for the management of networked devices. However, in scenarios that involve devices with limited hardware, a wrong polling time may results in excessive power consumption. In Figure 7, horizontal axis presents gateway polling time settled in log scale, whereas, the vertical axis presents the power consumption for each replication of 2 minutes of experiment. The smaller polling time, the higher power consumption. This increase in power consumption is a function of growth in the number of visits per unit of time. For the three architectures SNMP, SOA, and ROA, when the number of requests raises more resources are required, reflecting in a higher power consumption.

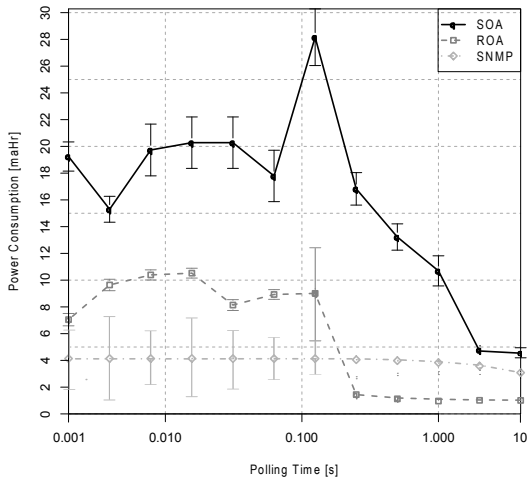

Figure 7: Power consumption versus polling time

It should be noted that, for a polling time greater than 2s, the battery consumption for SNMP, SOA and ROA behaves

similarly. However, for a shorter time, the battery consumption for SOA grows quickly and remains greater than SNMP and ROA to any value less than 1s. For ROA the same happens when comparing to SNMP at the mark of 0.1s of polling time, where ROA starts to consume much more energy than SNMP. However, SNMP presents a larger variation, showing that the energy consumption starts to vary when the polling time get smaller. Therefore, the relationship between polling time and the consumption of battery power is classified as a trade-off and must be analyzed to implement a polling based manager.

In summary, SNMP shown to be the most stable protocol presenting less varying results. However, ROA achieved the best performances in the timeout experiment, response time with group sensor requests, and presented the lower power consumption to gather information from an IoT device. Therefore, according to our investigations, ROA is the most suitable management architecture for IoT devices.

V. Conclusions and Future Work

In this paper, we presented a comparison of three management architectures, SNMP, SOA and ROA. These architectures have been prototyped and deployed in an experimental environment based on Sun SPOT devices. This environment was composed of a gateway that acts as a bridge that connect the Sun SPOT devices to the external network, such as the Internet. Using this experimental environment, we evaluated which architecture is more suitable for the IoT, in terms of response time and power consumption from the IoT device. In parallel to this investigation, we evaluated how some network parameters, *i.e.* timeout and polling time, influence the performance of SNMP, SOA and ROA architectures. Finally, we performed an analysis of effect between the gathering of different device sensors information. Through this analysis, we determined quantitatively which of the sensors is more costly in terms of response time and power consumption.

Evaluations involving the three architectures in terms of response time enabled to find that SNMP and ROA have a response time of seven times smaller than SOA. In addition, for the power consumption, ROA consumed 30% less power than SOA, and 8% less than SNMP.

The experimental environment makes possible to analyze two management parameters, the timeout and the polling time. These parameters modify the behavior of SNMP, SOA and ROA architectures. For example, the timeout influences the occurrence of errors. This parameter is aggravating to the performance of SOA, when sometimes the architecture could not even perform a successful request to the device, while ROA was found to be less sensitive to changes in timeout as much as SNMP. For the polling time, the smaller time between requests the major power consumption and more unstable becomes the response time for the three architectures. SNMP, SOA, and ROA begin to undergo competition when the polling time becomes smaller than the average response time of each architecture.

Evaluating the influence of sensors effect in the experiments enabled to analyze which sensor of Sun SPOT devices has a greater effect on variations in response time and battery consumption. The accelerometer of the SPOT was the sensor that had the greatest effect among all sensors. For SOA and ROA, the accelerometer had an effect of 90% of the variation in response time and 76% of changes in the variation of power consumption. However, SNMP shown that independent from the sensor requested, its requests performance are not very influenced, showing to be very well balanced to support individual sensor requests.

The analysis of the results of the polling time evaluation showed a trade-off. This trade-off involves the number of messages in a given time versus the power consumption of devices. In the future work, we will explore different strategies of polling time, *e.g.*, static and dynamic polling time configuration, and how much this strategies influence in the IoT devices management.

References

[1] I. Union, "The internet of things," *ITU Internet Reports*, 2005.

[2] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. Jubert, M. Mazura, M. Harrison, M. Eisenhauer10 *et al.*, "Internet of things strategic research roadmap," *Aerospace Technologies and Applications for Dual Use*, p. 9, 2008.

[3] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the internet of things," *European Commision, Cluster of European Research Projects on the Internet of Things*, 2010.

[4] C. Pautasso, O. Zimmermann, and F. Leymann, "Restful web services vs. "big"' web services: making the right architectural decision," in *Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 805–814.

[5] F. Alshahwan and K. Moessner, "Providing soap web services and restful web services from mobile hosts," in *IEEE Fifth International Conference on Internet and Web Applications and Services*, 2010, pp. 174–179.

[6] A. Dunkels *et al.*, "Efficient application integration in ip-based sensor networks," in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, 2009, pp. 43–48.

[7] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of resource constrained devices in the Internet of Things," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 144–149, Dec. 2012.

[8] Oracle Corporation, "Sun SPOT World, Program The World!" http://www.sunspotworld.com/docs/index.html, URL ressources last accessed at 31 january, 2014.

[9] D. Harrington, R. Presuhn, and B. Wijnen, "RFC 3411: An architecture for describing simple network management protocol (SNMP) management frameworks," 2002.

[10] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, and D. Winer, "Simple Object Access Protocol (SOAP) 1.1," 2000.

[11] R. Fielding and R. Taylor, "Principled design of the modern web architecture," *ACM Transactions on Internet Technology (TOIT)*, vol. 2, no. 2, pp. 115–150, 2002.

[12] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Network*, vol. 54, pp. 2787–2805, October 2010.

[13] M. A. Marotta, F. J. Carbone, J. J. C. de Santanna, and L. M. R. Tarouco, "Through the Internet of Things - A Management by Delegation Smart Object Aware System (MbDSAS)," in *IEEE 37th Annual Computer Software and Applications Conference*, Kyoto, Jul. 2013, pp. 732–741.

[14] A. Jara, M. Zamora, and A. Skarmeta, "An architecture based on internet of things to support mobility and security in medical environments," in *7th IEEE Consumer Communications and Networking Conference (CCNC)*, 2010, pp. 1–5.

[15] R. Jain, *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling*, W. P. Computing, Ed. NewYork: John, 1991.

[16] E. R. Margulies, M. Siegl, and M. Toman, "Sun SPOT SNMP agent and multiplexor," http://sunspotsnmp.sourceforge.net, URL ressources last accessed at 31 january, 2014.

[17] V. Gupta, P. Udupi, and A. Poursohi, "Early lessons from building sensor.network: an open data exchange for the web of things," in *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010, pp. 738 –744.