A Secure Web Service-based Platform for Wireless Sensor Network Management and Interrogation

Ahmed Amokrane, Yacine Challal, Amar Balla

A Secure Web Service-based Platform for Wireless Sensor Network Management and Interrogation

Ahmed AMOKRANE (a_amokrane@esi.dz)*
Yacine CHALLAL (ychallal@hds.utc.fr)†
Amar BALLA (a_balla@esi.dz)‡

Abstract: A Wireless Sensor Network (WSN) is composed of small, low cost and low energy consumption devices called sensors. Those sensors are deployed in a monitored area. They capture measurements related to the monitored phenomenon (temperature, humidity...) and send them through a multi-hop routing to a sink node that delivers them to a Base Station for use and decision making. WSN are used in several fields ranging from military applications to civilian ones, for security, home automation and health care... Up to now, most of the works focused on designing routing protocols to address energy consumption issue, fault tolerance and security. In this paper, we address the issue of secure management and interrogation of WSN through Internet mainly. In our work, we designed and implemented a generic approach based on Web Services that builds a standardized interface between a WSN and external networks and applications. Our approach uses a gateway that offers a synthesis of Web Services offered by the WSN assuring its interrogation and management. Furthermore, Authentication, Authorization and Accounting mechanism has been implemented to provide security services and a billing system for WSN interrogation. We designed our architecture as a generic framework. Then, we instantiated it for two use cases. Furthermore, we designed and implemented a Service Oriented routing protocol for WSN.

Keywords: Web Services, WSN, SOA, AAA, Routing

1 Introduction

A Wireless Sensor Network (WSN) is composed of small, low cost and low energy consumption devices called sensors deployed in a monitored area. Those sensors capture measurements related to the monitored phenomenon (temperature, humidity...) and send them by the mean of a multi-hop routing to reach a sink node. The sink delivers them to a more powerful device called Base Station (BS). The BS stores and/or processes data according to the application. WSN can be used in a couple of areas ranging from military tracking and surveillance to civilian applications in home automation (smart home), agriculture, intrusion detection and so on.

This technology has a huge potential application fields in the nearest future and efforts are being conducted to guarantee security services and hence enlarge the spectrum

* École nationale Supérieure d'Informatique, BP 68M, 16309, Oued-Smar, Alger, Algérie
† Université de Technologies de Compiegne, France
‡ École nationale Supérieure d'Informatique, Alger, Algérie

of potential applications. Most of the works focused on energy consumption through designing routing protocols [AKK04] and MAC algorithms [ZLZW10] that reduce the energy consumption. Also, the security issue has been addressed [WAR06].

Today and due to the diversity of applications and the expansion of Internet toward the web of things [W3C], the objectives are slightly moving forward. In fact, the deployed WSNs need to be interconnected with the existing IP networks and mainly Internet. This makes the data they provide accessible for applications and users. Furthermore, those applications are generally heterogeneous and distant. This means that they should cross a couple of networks (Internet) to reach the desired WSN instead of local applications that we usually design for WSN. In this perspective, we have designed a platform based on Web Services [W3C] that offers standardized interfaces for interrogating and managing WSN. Our approach is based on an application level gateway that offers a synthesis of the services offered by the WSN. The platform gives access for users and applications to interrogate and manage the WSN in a secure fashion thanks to the encryption and signature of the exchanged SOAP messages. Furthermore, it offers Authentication, Authorization and Accounting (AAA) services to define who can do what and helps to make a billing system. This is done at the gateway level which helps to save resources in the WSN part.

In a simplified description, our platform offers a set of Web Services for users. Those services give access to interrogate the WSN (*"What is the temperature in my bed room"*), manage the WSN (*"Turn off the sensors in my bed room"*), be notified in case of event (*"Send me an SMS if someone enters my bed room"*), see a summary of who used the platform and add or remove authorization for those services used by the administrators (*"Add my son to the list of users"*, *"Add authorization for my son to see what is the temperature in his bed room"*, *"What was the services invoked by my son during this week"*, *"Who used the platform today"*...).

Our proposed architecture has been though to be generic. In fact, unlike the existing works in literature, we designed a global architecture independently from any application. We have then instantiated it for two application cases. The first one is an application for meteorology and fire detection where sensors capture the temperature and humidity and send alerts in case of fire detection. The second one deals with the domain of agriculture where sensors capture the air and soil temperature and humidity and help to prevent dryness by sending alerts. Furthermore, we designed and implemented a Service Oriented routing protocol for WSN called Directed Service Oriented Diffusion (DSOD).

The remainder of this article is organized as follows. Section 2 summarizes the existing works that have treated similar problems. Our architecture is described globally in Section 3 and then detailed in Section 4. The instantiation of our framework through the two use cases is presented in Section 5. Section 6 summarizes some of the tests and the obtained results mainly on DSOD performance. Finally, in Section 7 we conclude this paper.

2 Related works

Few works addressed the issue of interfacing WSN with traditional IP networks. We classified them into two categories. On one side, we have those which use a gateway between the WSN and the external IP network (Internet). In fact, the idea of using application level gateways have been used in [PSL+06] and [NFH+08]. Moreover, gateways connected to Internet using GPRS have been introduced in [MSH07]. However, those

solutions, in contrary to our framework, don't use SOAs and are based on application messages. Also, D.I.Tapia et al. in [TFR+09] have introduced proprietary languages based on XML to describe nodes and services. Their problem is the lack of standardization. In [NDMS05, EJ09, CKB06] approaches based on Web Services have been proposed. In the [NDMS05] and [EJ09], the solution is tightly related to the application (emergency medical response, only event driven networks are considered). Dedicated to IEEE 1451 sensors, an architecture based on a gateway implementing Web Services has been introduced in [DHK+08]. The drawbacks of such an approach is the fact that it cannot be adapted to all applications due to constraints in the WSN. In other works, middlewares have been proposed either in the sensors [FRL09] or implemented in a gateway [THT09, DPR+05] as well as virtual machines in sensors [RGD07]. These concepts are interesting to build context aware architectures. However, introducing a middleware or virtual machines is complex and it is generally done in a proprietary way, in response to application needs.

On the other side, we have the other approaches that don't use gateways. They implement the services in the sensing nodes. Those approaches are based on the 6LowPAN [INWG08], an implementation of IPv6 on top of IEEE 802.15.4. This makes possible the addressing of sensors using IPv6 addresses. However, those works are still in progress. Among them, the authors in [MZP+09, GCF+09] have mentioned the possibility of using Devices Profile for Web Services (DPWS) which is still under development [OAS09]. In a slightly different way, [BAMF08] used a reduced implementation of the Web Services through simplifications in WSDL files and the transportation of request messages directly on HTTP instead of SOAP messages. Furthermore, [LPR09] and [YNM+07] used RESTful web services which is an implementation based on HTTP requests instead of SOAP. We note that these simplifications remain proprietary and don't respect the interoperability we are interested in. Unlike these approaches, our framework is based on the standardized Web Services which insures interoperability.

The major problem with the listed works is the fact that they are either tightly related to applications (like [NDMS05]) or they introduce proprietary languages and communication protocols such as [BAMF08] and [TFR+09]. Furthermore, the efforts toward generic approaches are sometimes neglected. However, we have seen that the concept of gateways implementing an SOA (based or not on Web Services) has been introduced. However, no generic approach has been proposed. Therefore, we propose our approach based on a generic architecture including AAA services. To our knowledge, except [YNM+07], no other work has introduced the AAA to insure security and accounting in WSN.

3 Our architecture

3.1 General overview

Our architecture is composed of three main parts:

- WSN Part: it is composed of the deployed sensors that capture the measurements related to the monitored phenomenon.
- The Gateway: it insures the AAA to protect the WSN and insures the translation of the requests (made by the users through Web Services) into requests according to the operating system and technology run by the sensors, and vice versa for the responses.

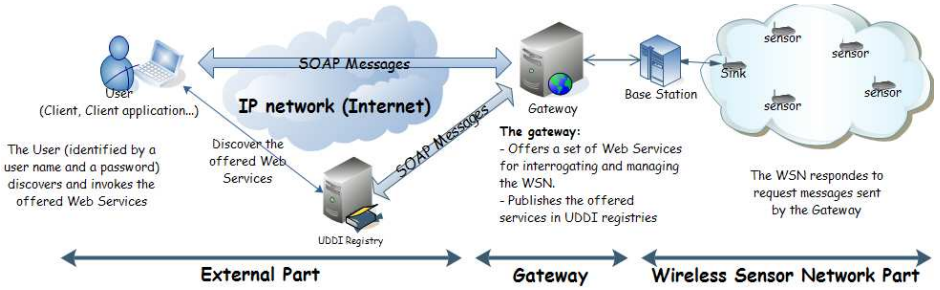


Fig. 1: Global architecture of our platform

- The external part: it is the part made of the users and the IP networks (Internet) that transport the requests (SOAP messages) from the users to the Gateway and vice versa.

The architecture is summarized in Fig. 1. From an operational point of view, the Gateway offers a set of Web Services classified into five classes described in more details in the following section. The users can list the available Web Services and invoke some of them by sending SOAP requests through the IP network toward the Gateway. The latter implements AAA. If the user is authenticated ans has the authorization to invoke the service, the Gateway translates it into requests sent to the sensors. Then, the answer is sent back in a SOAP massage to the user in the reverse path. If the user is not authorized or not authenticated, an error message is sent back.

3.2 Taxonomy of the offered Web Services

We classified the gateway Web Services into five classes as follows:

- Web Services for WSN real time interrogation: they are services that request a real time data from the sensors or a particular sensor. It could be, for instance, *"The mean temperature at home"*, *"The humidity in sensor X"*, *"The temperature in my bed room"*.... In this case, the request is translated by the Gateway into another one depending on the sensors Operating System and technology and sent to the appropriate sensor. The response, given by the sensor, is sent back to the user.

- Web Services for the WSN management: they are services that deal with the management of the WSN. It could be turning on/off some sensors, modifying the sensing interval of a sensor, launching and stopping the capture-store function (capture data by sensors at regular intervals of time and store them in a database located in the Gateway for further requests and statistics).

- Web Services for archive interrogation (earlier captured data): they are services that request the data that has been captured earlier. It could be the statistics for a period of time, for instance, *"The average temperature in my bed room yesterday"*. A request for such a service is translated into a request that interrogates a database of the captured-and-stored data.

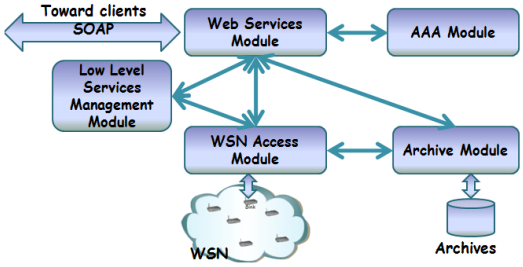


Fig. 2: Gateway Modules

- Web Services for subscription to events: they are services that could be invoked by users or applications that want to subscribe to some events in the WSN. For instance, fire department would be interested in receiving alert messages if a fire is detected by the sensors. Those alerts are sent by the Gateway by mean of SMS or e-mails after receiving a message notifying the occurrence of the event by one or several sensors.

- Web Services for administration: they are services dedicated to the administrators. They help them to set, modify and remove users and authorizations. They focus on the management of the AAA functionalities.

These services are augmented by the AAA. This means that for each service call, the user is authenticated (Authentication), checked against the authorizations (Authorization) and then recorded (Accounting). This insures the establishment of filtering rules. Furthermore, it is done at the gateway level. This helps to save resources in the WSN part since non authenticated or non authorized users are rejected at the gateway level. Moreover, the SOAP messages are encrypted and signed between the user application and the Gateway. This insures the interaction in a secure way.

4 Detailed architecture

The whole platform is divided into three parts that are independent from each other. They interact through interfaces defined by Web Services (WSDL) for the external part and the Gateway, and by the specific packet format of exchanged messages between the Gateway and the WSN.

4.1 The Gateway

We will look at the gateway from two different points of view: in terms of modules representing the main blocks, and in terms of a layered model.

4.1.1 Gateway Modules

The gateway as a whole is divided into four (4) main modules. These modules are interconnected in a mesh structure to simplify the design and for better efficiency. The different modules and the interconnections between them are shown in Fig. 2. They are:

1. Web Services Module: it represents the different Web Services offered by the Gateway. It assures the interface with the Internet and handles the requests coming from the clients (user applications). The first security filtering is done at this level (decrypting and signature checking). Moreover, it encrypts and signs the responses sent back to the clients. It is also responsible for the advertising of the offered Web Services (in UDDI registries).

2. AAA Module: it is responsible for the implementation of the Authentication, Authorization and Accounting functionalities. It is coupled with a dedicated AAA server. It is a critical part in the gateway since it checks the security credentials related to the user, mainly a user name and a password.

3. Archive Module: it is responsible for storing and retrieving the captured data. It assures the interface with the underlying storing technology which could be a relational data base, files, a cloud storage system...

4. WSN Access Module: it is responsible for the interface with the WSN. Depending on the Operating System run in the sensors, it sends the appropriate request messages and handles the response messages.

In addition, we built a Low Level Services Module which is a registry of the services offered by the sensors in the case where the latter implement a Low Level Service Oriented Architecture (see Section 4.2). The offered Web Services will be built on top of those Low Level Services.

4.1.2 Architecture layered model

We propose a generic model composed of three layers. The model as well as the location of the different modules in the different layers are shown in Fig. 3. The role of the different layers is as follows:

- Web Services Layer: it is the top layer of the model. It insures the interface with the users and handles the SOAP requests. It contains the Web Services Module and the AAA Module since it handles the Web Services and the AAA functionalities.

- Task Layer: it is a set of tasks that are launched by the Web Services Layer to interrogate the WSN or the archives. Each task is in charge of sending the request message (toward the archive or WSN) and wait for the answer. Furthermore, a scheduling mechanism is implemented at this level to insure the priorities for the different requests. This means that some services and/or users have more priority than others, so their tasks are scheduled first. It implements also mechanisms to aggregate requests, for instance, two calls for the same service will send only one message toward the WSN (saving resources in the WSN part).

- Access Layer: it is the bottom layer and it insures the interface with the WSN and the storage platforms. It contains the WSN Access and Archive modules.

From an operational point of view, the incoming requests (SOAP messages of Web Services) are received by the Web Services Layer. The Authentication and Authorization is checked at this level. If the user is not authorized or not authenticated, an error message

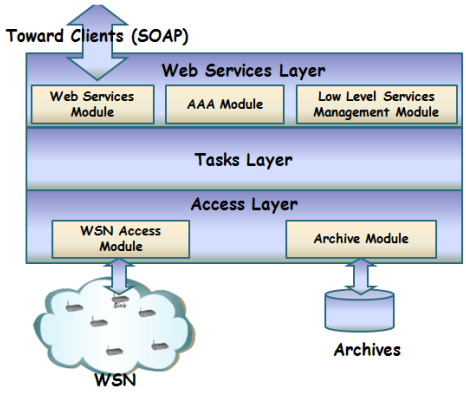


Fig. 3: Our Layered Model for the Gateway

is sent back. Otherwise, a task is created and launched by the Web Services Layer. The task is scheduled in the Tasks Layer and sends the appropriate request to either the WSN or the archives through the Access Layer. The response is sent back from bottom to up and then back to the user.

Let's note that the Administration Web Services are handled only in the Web Services Layer since they involve only the AAA Module.

4.2 The WSN

The WSN part is composed of the deployed sensors. It implements the usual protocol stack of WSNs. It uses a routing strategy that does not impact the way our platform behaves. However, according to the taxonomy of the Web Services we have given earlier (see Section 3.2), it is to note that the routing should be done according to the three ways of collecting data from the WSN: Query Driven (In case of WSN real time interrogation and management services), Event Driven (In case of subscription to events services) and Time Driven (In case of functionalities like capture-store).

Furthermore, according to the capabilities of the sensors, we can design the WSN part in two fashions: the first one as being a usual WSN with nodes that receive and send simple messages. The second one implements a SOA in the sensors.

4.2.1 A Service Oriented WSN and Directed Service Oriented Diffusion

In this section, we present our Service Oriented data gathering scheme for WSN. This scheme is implemented on the WSN side and extends naturally the SOA approach implemented on the gateway. This notion has been introduced also in [JMKV08]. We refer to this approach as being a Low Level Service Oriented Architecture to make a difference with the Web Services and the SOA at the Gateway level (High Level). The sensors in this case offer services (we call them Low Level Services) that could be invoked by other sensors or external nodes by means of Service Invocation Messages (Requests) and the responses (If they do exist) are given in Response Messages. Hence, the exchanged messages carry services invocations and responses as illustrated in Fig. 4.

Through our work, we have designed and implemented a Service Oriented routing protocol. It is a proof of concept of the possibility of implementing an SOA in the WSN part

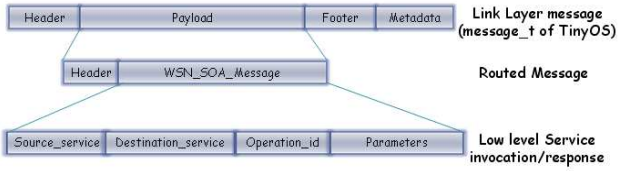


Fig. 4: Message structure in the Low Level SOA in WSN

in a generic and lightweight fashion. The protocol we have designed is called Directed Service Oriented Diffusion (DSOD). To our knowledge, no Service Oriented routing protocol have been introduced so far, even if the idea of SOA has been announced in [JMKV08]. DSOD manages Service invocation from sink to nodes and also between nodes in the WSN making possible the composition of Low Level Services in the WSN.

From a general point of view, each nodes maintains three data structures: *Interest_cache*, *Data_cache* and a *Registry*. A service invocation (from a sink) toward a node (destination) is translated into a Service Invocation Message. Each node that receives the message checks its local table called *Registry*. A Registry contains routes (The next hop) toward services in the network. Each service is identified by (*Service_Id*, *Node_Id*). If no route exists then the Invocation Message is flooded in the network. Each node that receives the message and sends it to its neighbors, keeps trace of the invocation in its *Interest_cache*. The *Interest_cache* is used to find the route back to the sink (source) when a response is given. This is illustrated in Fig. 5. Note that, the same message is forwarded only once (Checked in the *Interest_cache* before forwarding). When the Response is given, it is sent back through the reverse path of the service invocation thanks to the *Interest_cache*. Furthermore, the response message is added to *Data_cache* in each node it passes through to avoid infinite loops and duplication. If a route exists (in the Registry) toward the destination, then the message is sent to the destination through the existing route as illustrated in Fig. 5.

If more than one route exist in the Registry of a node, then only one is chosen, that one having the lowest result of the cost function. The cost function is given by $f(next\_hop) = amount\_of\_remaining\_energy\_in\_next\_node/number\_of\_hops\_to\_destination$. The usage of this cost function will help to choose different routes and load balancing between neighbors to save energy in the network as we will see it in Section 6.

Let's recall that the messages are limited in terms of the number of hops. The same holds for the entries in the registries where the routes are dropped if they are not updated for a predefined period of time.

The Registries are maintained in two fashions:

- Implicitly: it is done thanks to the messages that pass through the nodes. Let's recall that each message contains a source and destination service. So each time a message is received by a node, it adds (or updates) the source service to the Registry.
- Explicitly: by mean of explicit messages that a node sends to its neighbors to inform them of the services it offers periodically or after receiving a message asking for that. For instance, the Low Level Services Module in the Gateway sends those explicit messages to build a registry of the available Low Level Service in the WSN.

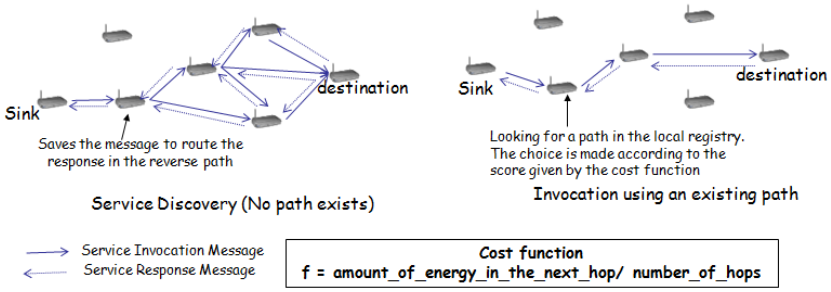


Fig. 5: DSOD: Principle

4.3 The user applications

They could be any kind of application that can send, receive and process SOAP messages. We have designed, for example, applications in Java, C# et PHP. They are much more like user friendly Graphical interfaces. In the context of our work, they are more a proof of concept of interoperability.

5 Instantiation of our generic framework

5.1 First instantiation: Fire detection and meteorology

In this use case, the sensors are deployed in a monitored area (forest...) to capture data related to temperature and humidity. They can answer requests for real time data (Temperature and humidity). Approximately they can detect fires but the accuracy of such a detection is out of scope of this paper (see , for instance, [HB09]). In the WSN part, the sensors run TinyOS [TIN]. This part is emulated using TOSSIM [TOS], the emulator for TinyOS. It uses Directed Diffusion [CRD00] as a routing protocol.

5.2 Second instantiation: Agriculture

The sensors are deployed in agriculture fields to capture data related to temperature and humidity in air and soil. They can answer requests for real time data and detect events like fires and dryness (that help to irrigate at the right time). Again the accuracy of such a detection is out of scope of this paper. The instantiated schema as well as some available services of the five calsses are illustrated in Fig. 6. In the WSN part, the sensors run TinyOS. It is emulated using TOSSIM. The WSN part implements our Service Oriented routing protocol DSOD.

For both of the two applications, the Gateway has been developed using Java. The Web Services are hosted in Apache Axis [AXI], hosted in Apache Tomcat web server [TOM].

6 Tests and Results

In this section, we present some screenshots of our client applications. They are illustrated in Fig. 7. They represent some of the results of the interaction with the platform.

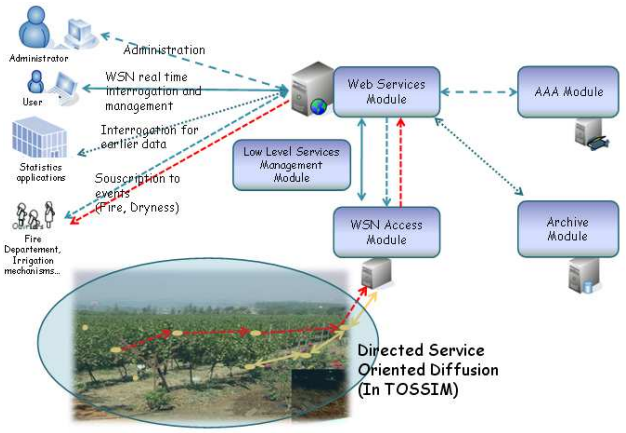
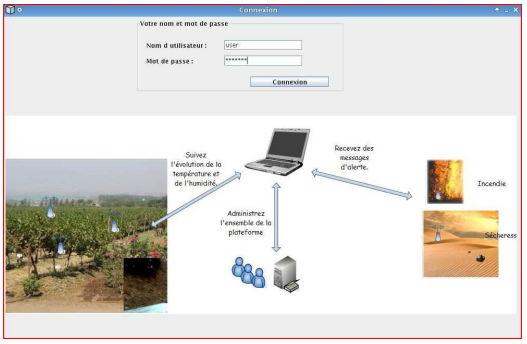
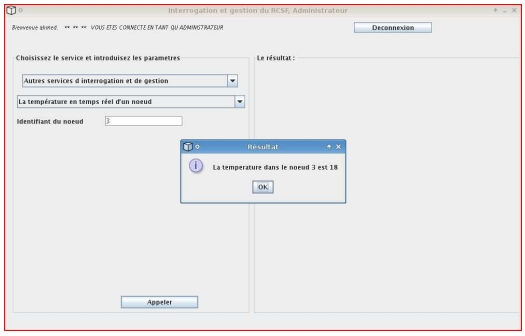


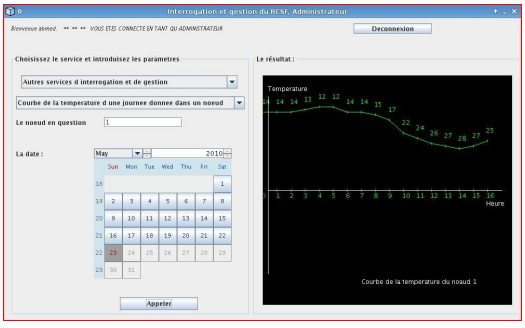
Fig. 6: Agriculture application instantiation of our generic framework



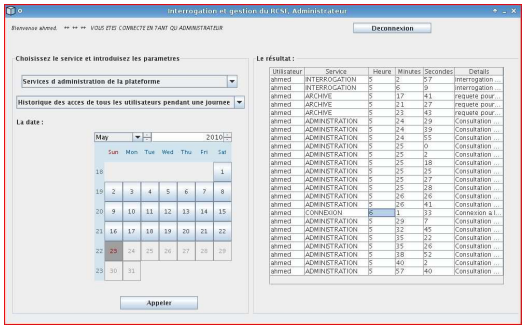
Connexion screen



Real time interrogation service



Data captured summary for one day (Archive Service)



Summary of accesses (Administration Service)

Fig. 7: Screenshots of a Client Application in the Agriculture use case

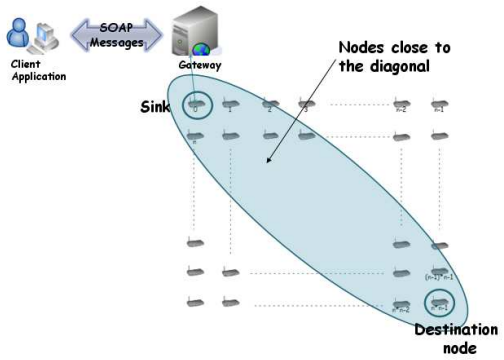


Fig. 8: The First test scenario

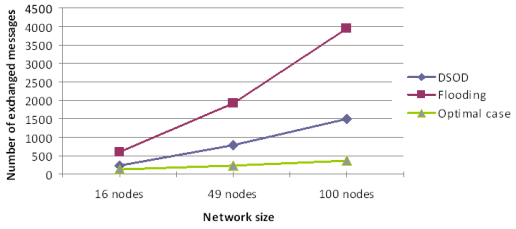


Fig. 9: The total number of exchanged messages for 20 service calls

We have also tested our Service Oriented routing protocol DSOD. The results we have obtained are promising for a new paradigm in WSN. We present here some of those tests and results.

In the first scenario, we consider a grid topology. A number of invocation messages (real time interrogation Web Services) is sent to the Gateway from a user application. The targeted node is always the same as illustrated in Fig. 8. We measure the total number of exchanged messages in all the network and in the nodes located near to the diagonal. The amount of exchanged messages gives us an idea on the load and energy consumption in the WSN. Let's recall that the diagonal is the shortest path in terms of the number of hops between the sink and the destination according to the topology in Fig. 8. For these nodes also, we measured the average number of exchanged messages as well as the standard deviation. These results will help us to get an idea of the load balancing of the routing task between the neighbors.

In terms of the total number of exchanged messages, it is summarized in Fig. 9. We note that it is less than the flooding case and not far from the optimal case. Is is done thanks to the Registries in the nodes helping to route through existing paths instead of flooding. Furthermore, the amount of messages increases in a linear way with the network size. This assures scalability for our protocol.

For the nodes located near to the diagonal, we have recorded the largest number of messages. This is normal since our cost function uses the number of hops to determine the next hop. We have also registered small variances as given in Tab. 1. This illustrates

Network size	Mean	Standard deviation
16	1.6	0.64
49	1.58	0.51
100	1.6	0.52

Tab. 1: The distribution of the number of messages per node for the nodes located near to the diagonal

Network size	Number of invocations	Number of distinct services invoked	Number of distinct services responding	Proportion of lost messages
16	100	13	13	0 %
49	100	34	34	0 %
100	100	67	67	2 %

Tab. 2: The results of responding services

the load balancing while routing messages between the nodes.

In the second scenario, a client application sends messages to randomly chosen nodes. We measured the amount of lost requests (Services invocations without responses). The results are summarized in Tab. 2. The coverage is around 100% since it is based on flooding to discover the services in the network. Also, no service invocation has been lost even if request messages are lost. This is due to the fact that DSOD takes care of resending the invocation messages through different routes if no answer is received.

7 Conclusions

Through our work, we have given an architecture that integrates the WSN in IP networks, mainly Internet. Our approach is based on Web Services implemented in a gateway. The usage of Web Services provides opportunities due to the fact that they are standardized and widely used in Internet today. The offered services are augmented by Authentication, Authorization and Accounting (AAA) functionalities to insure filtering clients and requests at the gateway level.

We have given a global overview and detailed decomposition of the design of our framework. Furthermore, it was though to be generic, not limited to some applications. Then, we instantiated it into two applications to provide a proof of concept. We have seen that the interoperability is always insured through the different client applications we developed.

Moreover, we have given an implementation of a SOA in WSN which we called Low Level SOA. We have also designed and implemented a Service Oriented routing protocol for WSN called Directed Service Oriented Diffusion (DSOD). It is the first routing protocol in this new paradigm for WSN and the results we obtained are promising.

References

[AKK04] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6 – 28, december 2004.

[AXI] Apache axis web site. `http://axis.apache.org/axis/`.

[BAMF08] Priyantha Nissanka B, Kansal Aman, Goraczko Michel, and Zhao Feng. Tiny web services: design and implementation of interoperable and evolvable sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 253–266, New York, NY, USA, 2008. ACM.

[CKB06] Xingchen Chu, Tom Kobialka, and Rajkumar Buyya. Open sensor web architecture: Core services. In *In Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing*, pages 1–4244. Press, 2006.

[CRD00] Intanagonwiwat Chalermek, Govindan Ramesh, and Estrin Deborah. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 56–67, New York, NY, USA, 2000. ACM.

[DHK+08] A. Dersingh, M. Helmer, A. Kamath, R. Liscano, E. Sadok, and A.Thommandram. A multi-tier sensor web network based on the IEEE 1451 instrumentation standard. *SWAN Project(Sensor Web Application Automation Network)*, 2008.

[DPR+05] Flavia C. Delicato, Paulo F. Pires, Luiz Rust, Luci Pirmez, and José Ferreira de Rezende. Reflective middleware for wireless sensor networks. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1155–1159, New York, NY, USA, 2005. ACM.

[EJ09] Churcher Gavin E. and Foley Jeff. Applying and extending sensor web enablement to a telecare sensor network architecture. In *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*, COMSWARE '09, pages 6:1–6:6, New York, NY, USA, 2009. ACM.

[FRL09] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu. Enhanced coordination in sensor networks through flexible service provisioning. In *Proceedings of the 11th International Conference on Coordination Models and Languages*, COORDINATION '09, pages 66–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[GCF+09] Moritz Guido, Cornelius Claas, Golatowski Frank, Timmermann Dirk, and Stoll Regina. Differences and commonalities of service-oriented device architectures, wireless sensor networks and networks-on-chip. In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, WAINA '09, pages 482–487, Washington, DC, USA, 2009. IEEE Computer Society.

[HB09] Mohamed Hefeeda and Majid Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc and Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[INWG08] RFC 4944 IETF Network Working Group. Transmission of ipv6 packets over ieee 802.15.4 networks, 2008. `http://tools.ietf.org/html/rfc4944`.

[JMKV08] Leguay Jeremie, Lopez-Ramos Mario, Jean-Marie Kathlyn, and Conan Vania. Service oriented architecture for heterogeneous and dynamic sensor networks. In *Proceedings of the second international conference on Distributed event-based systems*, DEBS '08, pages 309–312, New York, NY, USA, 2008. ACM.

[LPR09] Schor Lars, Sommer Philipp, and Wattenhofer Roger. Towards a zero-configuration wireless sensor network architecture for smart buildings. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 31–36, New York, NY, USA, 2009. ACM.

[MSH07] R.S. Marin-Perianu, J. Scholten, and P.J.M. Havinga. Prototyping service discovery and usage in wireless sensor networks. In *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, pages 841–850, Los Alamitos, October 2007. IEEE Computer Society Press.

[MZP+09] Guido Moritz, Elmar Zeeb, Steffen Preter, Frank Golatowski, Dirk Timmermann, and Regina Stoll. Devices profile for web services in wireless sensor networks: Adaptations and enhancements, 2009.

[NDMS05] Hashmi Nada, Myung Dan, Gaynor Mark, and Moulton Steve. A sensor-based, web service-enabled, emergency medical response system. In *Proceedings of the 2005 workshop on End-to-end, sense-and-respond systems, applications and services*, EESR '05, pages 25–29, Berkeley, CA, USA, 2005. USENIX Association.

[NFH+08] Tomasz Naumowicz, Robin Freeman, Andreas Heil, Martin Calsyn, Eric Hellmich, Alexander Brandle, Tim Guilford, and Jochen Schiller. Autonomous monitoring of vulnerable habitats using a wireless sensor network. In *Proceedings of the workshop on Real-world wireless sensor networks*, REALWSN '08, pages 51–55, New York, NY, USA, 2008. ACM.

[OAS09] OASIS. Web services discovery and web services devices profile (ws-dd) tc, 2009. `http://www.oasis-open.org/committees/ws-dd`.

[PSL+06] Jose Pinto, Alexandre Sousa, Paulo Lebres, G.Manuel Gonzalves, and Joao Sousa. Monsense - application for deployment, monitoring and control of wireless sensor networks. *ACM Workshop on Real-World Wireless Sensor Networks. REALWSN'06*, 2006.

[RGD07] Muller Rene, Alonso Gustavo, and Kossmann Donald. A virtual machine for sensor networks. *SIGOPS Oper. Syst. Rev.*, 41:145–158, March 2007.

[TFR+09] D I. Tapia, Juan A. Fraile, Sara Rodriguez, Juan F. de Paz, and Javier Bajo. Wireless sensor networks in home care. In Joan Cabestany, Francisco Sandoval, Alberto Prieto, and Juan Corchado, editors, *Bio-Inspired Systems: Computational and Ambient Intelligence*, volume 5517 of *Lecture Notes in Computer Science*, pages 1106–1112. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-02478-8_138.

[THT09] Masaaki Takahashi, Basit Hussain, and Bin Tang. Demo abstract: Design and implementation of a web service for liteos-based sensor networks. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, IPSN '09, pages 407–408, Washington, DC, USA, 2009. IEEE Computer Society.

[TIN] Tinyos web site. `http://www.TinyOS.net`.

[TOM] Apache tomcat web site. `http://tomcat.apache.com`.

[TOS] Tossim web page. `http://www.cs.berkeley.edu/~pal/research/tossim.html`.

[W3C] W3c: World wide web consortium. `http://www.w3c.org`.

[WAR06] Yong Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 8(2):2–23, 2006.

[YNM+07] Kawahara Yoshihiro, Kawanishi Nao, Ozawa Masahiro, Morikawa Hiroyuki, and Asami Tohru. Designing a framework for scalable coordination of wireless sensor networks, context information and web services. In *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, pages 44–, Washington, DC, USA, 2007. IEEE Computer Society.

[ZLZW10] Dalong Zhang, Qing Li, Xiaoyi Zhang, and Xiaomei Wang. De-ass: An adaptive mac algorithm based on mobility evaluation for wireless sensor networks. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1 –5, september 2010.