A Virtual Channel-based Framework for the Integration of Wireless Sensor Networks in the Cloud

Javier Barbarán
Software for Critical Systems S.L.
Málaga, Spain
barbaran@softcrits.es

Manuel Díaz, Bartolomé Rubio
Department of Languages and Computer Science
University of Málaga
Málaga, Spain
{mdr,tolo}@lcc.uma.es

***Abstract*—Currently, Cloud Computing is a cutting edge technology as it is changing how people produce, consume and share information. Nevertheless, a bigger evolution of the Cloud is yet to come by means of new applications and improvements but also with the combination of other arising technologies, resulting in a huge advance as part of the Future Internet. The next step of the Future Internet in the next few years is to get every single device connected at any-time and in any-place with any other device. Moreover, Wireless Sensor Networks offer a technology for monitoring and networking that is low power and low cost which is the basic layer in a world of things interconnected. The combination of Wireless Sensor Networks and Cloud Computing will enable unprecedented opportunities in the Future Internet that will include physical devices. This paper presents a new framework that simplifies the integration of Wireless Sensor Networks in the Cloud. The approach is highly reconfigurable, self-managed and takes into account real-time requirements. The basic idea of the framework is the concept of Virtual Channels, which exchange messages between every single device and the Cloud. The framework then publishes a "virtual sensor" service representing one or more physical sensors.**

Keywords— Wireless Sensor Networks; Cloud Computing; Future Internet; Internet of Things; Distributed Communications; Virtual Channels

I. INTRODUCTION

In the last years, thanks to the recent technological advances in low power integrated circuits and wireless communications carried out by academic and industrial sectors, research fields have evolved in many directions at the same time. For instance, back in the middle of the last decade, Wireless Sensor Networks (WSNs) was one of the most promising fields in wireless communications [1] and a lot of research efforts were put mainly into three categories. First, the creation of low-power, low-cost communication protocols such as Zigbee, 6LowPAN or WirelessHart, among others. The second is the improvement of physical devices when a huge amount of different companies emerged offering their own designs. And, last but not least, new software approaches were studied and developed in order to create frameworks and middleware with the aim of dealing with unreliable links and changing topologies, but also by making embedded devices smarter and providing more secure communications, all this trying to make the programmers life easier as the development of software for embedded devices is sometimes complex [2] [3].

On the other hand, Cloud Computing has been established as one of the most important parts of the next generation Internet. In [4] the concept of Cloud Computing is defined as "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered to external customers over the Internet". The goal of this model is to make a better use of distributed resources to achieve a better service for the users. Going deeper into the development of software, Cloud Computing is also changing the way software is created, as more and more applications nowadays are being developed to be executed in the Cloud. As a result, there are specific programming paradigms based on the concept of "Everything as a service" (XaaS) [5]. Platform (PaaS), Infrastructure (IaaS) and Software (SaaS) are developed in the Cloud as services.

Considering how the Cloud has evolved, the virtualization of objects is the next natural step in this field. When talking about object virtualization, we mean that almost every device with communication capabilities, also known as "smart objects", could share information with other devices through the Internet. This paper is focused on the virtualization of WSNs in the Cloud, proposing a framework which aims to achieve a seamless integration. This can be done providing software agents (running in the Cloud), also referred to as virtual sensors, that behave just like physical sensors, publishing in the Cloud the environmental conditions in the area where they are deployed. This way, we provide the so-called Sensor Data as a Service (SDaaS) paradigm.

In this paper, we propose a communication model with the main objective of providing a virtual image of physical sensors in the Cloud, by means of an abstract concept called *Virtual Channels*. This concept is not a new idea. It is based on the original model called TCM [6] which is a generic coordination model for parallel and distributed programming. It was successfully applied in the TC-WSANs project, where channels were used as the coordination and communication mechanism among sensor nodes [7]. Now, we are focused on using this concept in the research area of integrating WSNs in Cloud Computing. On the other hand, the concept of Virtual Channel that we are proposing in this paper must not be confused with the same concept used for designing efficient wireless control algorithms in MAC layers, such as CSMA [8] or others.

A Virtual Channel (VC) is an abstract full-duplex communication pipe with a set of associated communication attributes that may vary from one VC to another, defining the communication behavior between physical and virtual sensors. Therefore, every physical sensor has a VC associated for exchanging messages with the corresponding virtual sensor, in order to support communication and synchronization.

The most relevant aspects of this model, which makes it, in our opinion, a step forward in the integration of WSNs in the Cloud are the following: Synchronization of sensor measurements in real-time in the Cloud. The sensors virtualization hides from the user the details of dealing with physical sensors and also the unreliability of wireless sensor links. Lastly, the virtual channel abstraction provides high expressiveness and flexibility to the framework.

The rest of this paper is organized as follows: a solid state of the art is described in Section 2. In Section 3 we present the operational setting and motivation of our approach. Then, in Section 4, we define the proposed framework. Section 5 concludes the paper and some future work is scheduled.

II. Related Work

To the best of our knowledge, first attempts to combine WSN and Cloud Computing have been published at the beginning of 2009. In the following different proposals are analyzed. We have classified them into four categories taken into account the main objectives of the approaches.

A. WSN-Cloud integration

In this section we consider those proposals that are mainly focused on the way WSNs interact with the Cloud.

In [9] a Cloud Computing model mainly based on pipes and filters is proposed for the system hosted at the cloud infrastructure. Pipes do not transform data (coming from sensors), but generally buffer it and provide a uniform interconnection mechanism of filters. Filters do specific processing and transformation on the input data, such as data refinement or suppression. A simple filter could, for example, be responsible for storing the data in the database or deleting data below or above a certain threshold

The work presented in [10] is focused on finding the shortest path between a sensor node (from de WSN) and a cloud server node (from the Cloud). An efficient ant colony optimization technique is used, so that all data from the sensor node can be uploaded to nearby cloud server, and if necessary efficient query execution is done when required by sensor nodes. Failures of sensor nodes or server nodes are also taken into account. Alternative paths can be selected for faster response time.

In [11] an architecture has been developed according to the Sensor Web Enablement (SWE) standard defined by the Open Geospatial Consortium (OGC) [12]. The layer for the interaction with the sensing infrastructures makes use of the Contiki Operating System. It gives a uniform platform for communicating with heterogeneous sensors. The proposal is a contribution to the design of a pervasive infrastructure supporting facilities for contextualization and geo-awareness.

The WSN-Service Orchestration Architecture WSN-SOrA is presented in [13]. It orchestrate service provisioning for embedded networked systems and enables WSNs to act as cloud ready infrastructures that facilitate on-demand provisioning for potentially multiple individual backend systems. The Service Oriented Architecture (SOA) principle is established at all tier: node network, gateway and back-end/enterprise core, that is, at WSN-Cloud interaction level and at WSNs-Cloud users interaction level. In this sense, this approach has also to be included in section C of this state of the art.

In [14] a generic architecture is presented, based on a lightweight component model and dynamic proxy-based approach with the aim of connecting WSN with the Cloud. In particular, their architecture converts the gateway into the most important part as the middleware core is moved to this device, therefore reducing the work of the sensors. Also, when data from sensors arrive to the gateways it creates a dynamic component for each sensor that is detected. In order to connect the Cloud and the gateway, the authors use LooCI middleware [15] that is based on the event-bus abstraction to bind reusable components.

A quite different approach is presented in [16] where the authors propose a virtualization of sensors in the Cloud by reflecting data. In this approach most of the work is done by sensors where sensor readings can be retrieved from the node that did the reading itself, but also from those nodes that helped the replica of the data. To do that their proposal is based on a previous work called tinyDSM [17] which is a middleware approach that allows defining and enforcing data replication in WSN.

Our approach shares the same aim with the other proposals mentioned above, but our model is based on the use of Virtual Channels for connecting physical and virtual sensors. Our model also provides real-time communication without adding extra complexity to sensors with less computing and battery capabilities, as the runtime of the model is designed in a way that the Virtual Channels are going to be managed in the Cloud, offering lightweight proxies for gateways and sensors.

B. WSN inside the Cloud

A different vision of WSNs and Cloud integration is given in some approaches, where WSNs are considered as entities inside the Cloud, instead of a different system that interacts with the Cloud.

In [18] authors propose what they call Tangible Cloud Computing, which extends the current domain of the Cloud to include the physical world (WSNs). This means that networks of physical devices (sensor nodes) should be able to expose their functionality as standardized Cloud services. As first class entities in the Cloud, devices in the Tangible Cloud can also be used together with 3rd party cloud resources.

Taken the previous work as background, in [19] authors demonstrate Cloud Computing capacity for supporting elastic sensing and modeling applications and show that it is feasible for sensor nodes to use and manage the Cloud-based extensible modeling resources. The elastic nature of Amazon EC2 has been shown as a perfect candidate to support the dynamic loads provided by integrated environmental monitoring and modeling applications.

C. WSN-Cloud users interaction

The proposals of this category are mainly focused on the way the data collected from the sensors can be obtained by Cloud users.

In [20] a content-based publish/subscribe framework is proposed. Sensor data are coming through gateways to a publish/subscribe broker, which delivers sensor information to the consumers of SaaS applications. In order to match published sensor data or events with subscriptions efficiently, a fast and scalable event matching algorithm called Static Group Index Matching is proposed.

The Service Oriented Architecture (SOA) is considered in [21]. The proposed architecture consists of a layered service stack that has management, information, presentation and communication layers with all required services and repositories. The architecture uses SOA and features of cloud like virtualization to deal with heterogeneity. Services are used to allow the interaction between WSNs, subscribers and other clouds.

As it will be mentioned in section V, we are thinking of using one of these two models, publish/subscribe or SOA, in order to support the service part of our framework (SDaaS).

D. Application perspective

A key drawback of current Cloud Computing models is that they do not allow for interaction with the physical world. This precludes use of the Cloud model in a large number of application domains such as environmental monitoring, structural health monitoring, medical computing, industrial automation, etc. The integration of WSNs into Cloud Computing can avoid this disadvantage. In this sense, some proposals are currently appearing with the main objective of applying the integration of WSNs and Cloud Computing to different application scenarios. Some examples are the following:

In the application area of people healthcare several approaches have appeared, such as [22]. Besides of integrating WSNs and Cloud Computing, these proposals also take into account the security aspect in order to provide the data or information on the Cloud with a good protection. The systems proposed monitor human health, activities, and share information among doctors, care-givers, clinics, and pharmacies in the Cloud, so that users can have better care with low cost.

In [23] authors focused on introducing the latest technologies in sensors, wireless networks and Cloud computing to radically revise approaches to agriculture and conduct business feasibility studies to establish a hypothetical model of Cloud services that make a genuine contribution to agriculture. They conducted demonstration tests with the cooperation of two Japanese farming corporations.

An architecture for smart building control and energy management is designed and implemented in [24]. It is argued that a significant proportion (in US about 40%) of total worldwide energy is consumed by buildings. This way, making buildings more energy-efficient is an important step to reduce energy consumption in the combat with global climate change. A high-level system architecture is proposed supporting three main design principles: hierarchical sense and respond; a reference semantic model that facilitates information exchange among the various building subsystems; and a subscription/usage based model over the Cloud in order to minimize the cost of IT over the lifecycle of a building.

III. Operational Setting and Motivation

Typically, WSNs are designed to monitor environmental conditions of the place where they are deployed, and, using this information, some actions can be taken to control or prevent certain situations. However, if we think how many sensor networks are deployed in the world to monitor similar conditions (some of them are even close to each other), it comes to mind that these systems could collaborate with each other. Doing so, we could save batteries, costs and, thereby, the sensor network deployments would be more efficient, profitable, cheap and, of course, more interoperable. The integration of WSNs in the Cloud solves this problem as sensor reading can be shared in the Cloud to provide services for other applications.

In this paper the operational setting considered is focused on the integration of WSNs in the Cloud by means of virtualizing physical sensors in the Cloud. This way the virtual sensors in the Cloud hide the details of the unreliability of wireless sensor links and the complexity of how to exchange messages with the physical ones.

The global vision of the operational setting is depicted in Fig. 1. The sensor layer is composed by a set of different sensor networks, deployed worldwide and implementing different communication protocols. Some of them using ultra low-power protocols, and others using IP-based wireless protocols. In the gateway layer there are gateways that are needed by sensors without IP-based communication protocols, this way these sensors can also connect to the Cloud.

The other piece of this operational setting is the Cloud, where sensors are going to be virtualized. Our framework will be considered in the Cloud as a Sensor Data as a Service (SDaaS). This way, the user of the cloud can configure and build applications on top of our framework, taking advantage of data provided by the real world (physical sensors).

As result of the integration, there will be virtual sensors, in the Cloud, connected seamlessly to their corresponding physical sensors through Virtual Channels. This way, the virtual sensors will offer to the Cloud users the current status and also the latest measurements of the whole sensor network.

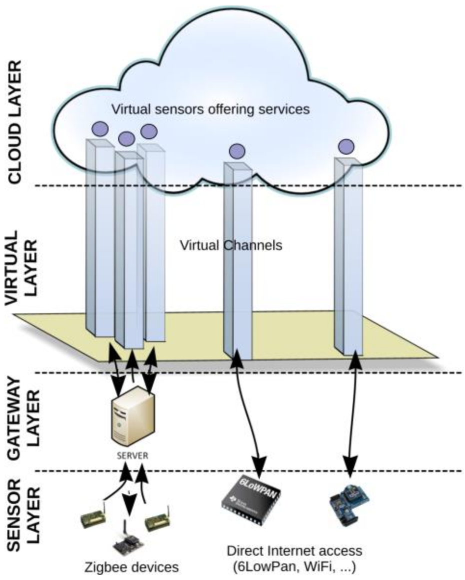

Fig. 1. Global vision of the system.

There are a huge amount of scenarios where the integration of sensors in the Cloud can be applied. Following are two representative examples showing the relevance of this technology:

- Suppose a sensor network deployed along a motorway offering measurements of environmental conditions and the state of road traffic. The information of each physical sensor is sent to its corresponding virtual sensor in the Cloud. The virtual sensors will offer all this information as a service. Users can have an application running on their phones or tablets that check the information offered by the virtual sensors about the state of traffic, and in case of traffic jam the phone or tablet will adjust the alarm clock accordingly.
- Every storage facility of nuclear waste has security systems composed of radiation sensors inside and outside of the facility. These measurements could be sent to virtual sensors in the Cloud, with the adequate security levels. This way, in case of emergency the measurements could be received in real-time not only by facility security departments but also by local and national authorities in order to take action as soon as the emergency situation is detected.

IV. The Framework

The framework presented in this paper is based on the concept of Virtual Channels (VCs). VCs are responsible for virtualizing sensors in the Cloud, leading to virtual sensors. Therefore, the VC is the tool that provides the communication and synchronization between physical and virtual sensors in real-time.

Fig. 2 shows a general view of the framework. It is composed by a set of VCs communicating physical sensors (in the real world) with virtual sensors (in the Cloud) that will offer services on behalf of physical sensors. Those services may be used by third party cloud applications in order to create more complex applications. The VCs are managed by a system called Virtual Channel Manager (VCM), located in the Cloud that will create, delete, or modify the behavior of each channel. The VCM provides an API, which will be described in section IV.B, with the basic primitives for managing the VCs.

Physical sensors (sensors with direct connection to the Cloud, and gateways connected to those without a direct connection to the Cloud) are going to access these VCs in order to exchange information between them and their virtual representations in the Cloud. In order to access to the Cloud, physical sensors are going to use the Virtual Channel Manager Proxy (VCMP), which basically offers an easy-to-use API, this way, they can request to join or leave the system, sending or receiving messages. It will be also described in section IV.B.

A. Virtual Channels

A Virtual Channel (VC) is an abstract full-duplex communication pipe with a set of associated communication attributes that may vary from one VC to another, defining the communication behavior between physical and virtual sensors.

The VC supports both one-to-many (between a virtual sensor to some physical sensors) and many-to-one (between some physical sensors to a virtual sensor) communication schemes. This way, a virtual sensor can be connected to N physical sensors, by means of a VC. Therefore, a command sent through a VC will be sent to all those physical sensors without the need of sending the message N times, avoiding unnecessary messages circulating in the network. On the other hand, different physical sensors can send their information to the same virtual sensor through the same VC. Additionally, a VC can also connect a virtual sensor with only one physical sensor.

One of the characteristics that contributes to achieving the real-time requirements demanded in WSNs is the priority issue. Some activities are more important than others and should be scheduled in an appropriate way in order to enhance the system response time. In our approach, the priority issue is considered at two levels: channels and messages.

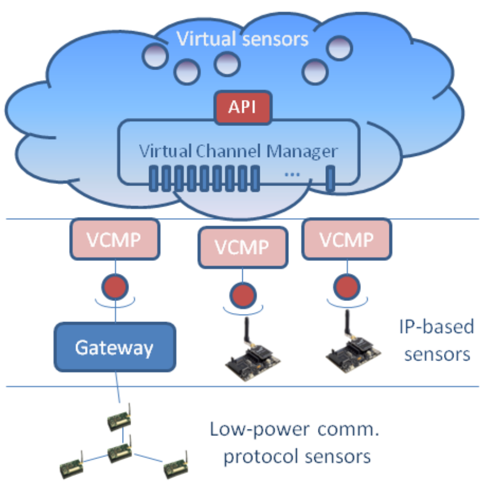

Fig. 2. Framework architecture.

On the one hand, a priority can be associated to a VC by means of an attribute. This way, a priority based scheduling can be established to get data from different channels. On the other hand, when a channel producer sends a message, some attributes may be associated with it. The way this message is treated and ordered inside the VC depends on the specified message attributes. The possible attributes that we have initially considered to be associated with a message are the following:

- Priority: It establishes the priority of the message. Messages are ordered inside the channel by priority. Messages with higher priority are the first obtained by the channel consumers. This allows an entity to attend to the highest priority messages or events first.
- Deadline: It establishes the maximum time the message is going to stay in the channel.
- Remove: It indicates if older messages of the same kind should be removed from the channel before adding the new one. For example, in some cases only the most current reading from a physical sensor is needed by a virtual sensor. This attribute precludes a virtual sensor from getting old readings from the same physical sensor.

VCs have a set of attributes associated that describe their communication behavior and are also useful for the management of VCs in the Cloud. Here is a list of some of the attributes associated to a channel:

```
<Name> Chan16 </Name>
<Description> VC 1 phys.sens.6 </Description>
<QoS> 1 </QoS>
<Security> 2 </Security>
<AlivePeriodMs> 10000 </AlivePeriodMs>
<Priority> 5 </Priority>
```

As it can be seen in the list of attributes, a channel will have a unique name, a description usually telling which physical sensors are connected to it, and also more specific attributes, such as quality of service (QoS) that depending on the level will enable/disable the acknowledgement of messages sent from virtual or physical sensors. Along with all these, there is a security attribute that depending on the level assigned, it defines whether the encryption of messages is activated or not and also the credentials needed by sensors to access the VC.

Due to the weakness of wireless links, physical sensors can disappear for a while because of lost connectivity, or even forever because it has ran out of battery or simply was destroyed. In this case the AlivePeriodMS attribute is the amount of time before the VC is going to send a beacon packet to the sensor, and after N messages without an answer the channel will notify the system that the sensor has disappeared and therefore the channel will be destroyed.

B. *Virtual Channel Manager*

The Virtual Channel Manager (VCM) is the core component of the framework, running in the Cloud, which is in charge of the management of the VCs. It also forwards all the messages from physical to virtual sensors and vice-versa.

The way a VCM chooses a channel to get messages from, is attending to the level of priority of every VC. The second level of priority proposed in the model is that messages of a VC with a higher priority level will be dispatched sooner. However, the runtime model of the VCM is not so easy. The VCM gives its attention to each VC, starting from those with higher priority. In addition, and to avoid that messages are indefinitely postponed, the VCM gives a "window of time" of its attention to dispatch VC messages. Once this period of time has elapsed, the VCM gives attention to the next VCs with higher priority, and so on.

Typically, the VCM will be accessed by sensors or gateways on behalf of those sensors without IP-based communication capabilities. The model proposes a proxy-based scheme, so every sensor (directly or through a gateway) attempting to access the Cloud has to use a proxy, which is the Virtual Channel Manager Proxy (VCMP). It offers a generic API hiding the details related to the VCs. The VCMP internally will deal with VC details using the VCMP_API. The operations offered by the VCMP are the following:

```
interface VCMP_API {
      void Join();
      void Leave();
      void SendMessage(message_t message);
      bool MessagesToRead();
      message_t[] ReadMessages();
      message_t ReadMessage();
}
```

As commented before, the VCM offers a more specific API that allows its clients to request operations over VCs. This API will be used by VCMP and virtual sensors in order to access to VCs, sending or receiving messages. Following are the functions offered:

```
interface VCM_API {
      ch_id CreateChannel();
      bool RemoveChannel(ch_id chan);
      ch_id JoinChannel(ch_id chan);
      void SendMessage(ch_id chan,
                       message_t message);
      bool MessagesToRead(ch_id chan);
      message_t[] ReadMessages(ch_id chan);
      message_t ReadMessage(ch_id chan);
}
```

The VCM and the VCMP negotiate the access of new physical sensors to the system. In this case the VCM creates a generic VC called Guests Virtual Channel (GVC) that will accept messages from unknown physical sensors. Those physical sensors will provide their credentials through GVC and then, attending to the directives set up by the framework programmer, the VCM will decide whether to join the sensor (creating a VC) or not.

C. *Sensor virtualization*

The virtualization of sensors is considered in two different levels. Those sensors without IP-based communication protocol are in the lowest level of virtualization, where they need the presence of a gateway that brings those sensors into the Internet. The higher level of virtualization is composed by those sensors using IP-based communication protocols accessing the Internet directly. Both, sensors and gateways use the VCMP to access the Cloud as shown in Fig. 2 and described in section IV.B.

The virtual sensors offer services in the Cloud that will be consumed by Cloud application's programmers. The potential of this model is shown at this point, that users of these services do not have to know anything about channels or unreliable wireless links, instead users just use a simple interface offered as a service.

The services offered and the behaviors of the virtual sensors are defined by the framework programmer. Therefore, the programmer uses the VCM API to access VCs in order to exchange messages with physical sensors.

V. CONCLUSIONS AND FUTURE WORK

We have presented a new framework for the integration of Wireless Sensor Networks in the Cloud by means of virtualization of sensors. The communication and synchronization between physical and virtual sensors are based on Virtual Channels, which are abstract full-duplex communication pipes with a set of associated attributes.

The approach provides synchronization of sensor measurements in real-time in the Cloud. It also provides abstraction to the user, as virtual sensors hide the details of dealing with physical sensors from users. Finally, the Virtual Channel abstraction provides a high level of expressiveness and flexibility.

We are involved in the development of a prototype for implementing and evaluating the framework. On the one hand, the WSNs-Cloud integration part (VCM, VCMP and VCs) will be based on our previous work TC-WSANs, where channels were used to establish the communication and synchronization among nodes in a Wireless Sensor and Actor Network. On the other hand, for the service part, that is the way the framework will offer the sensor services (SDaaS), through the virtual sensors, to the Cloud application's programmers, we are analyzing two different possibilities: the publish/subscribe model and the service oriented architecture.

ACKNOWLEDGMENT

This work was supported by the Spanish projects TIC-1572 (“MIsTIca: Monitoring of Critical Infrastructures based on Wireless Technologies”) and TIN2011-23795 (“WiCMaS: Wireless based Critical Information Management Systems”).

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless Sensor Networks: A Survey” Computer Networks Journal, vol. 38(4), pp. 393-422, 2002.

[2] B. Rubio, M. Díaz, J.M. Troya, “Programming Approaches and Challenges for Wireless Sensor Networks” IEEE 2nd Int. Conf. on Systems and Networks Communications (ICSNC’07), Special Track on Sensor Networks (SENET’07), Cap Esterel, France, 2007.

[3] L. Mottola, G.P. Picco, “Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art” ACM Computing Surveys, vol. 43(3), 2011.

[4] I. Foster, Y. Zhao, I. Raicu, S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared” Grid Computing Environmets Workshop (GCE’08), pp. 1-10, Austin, USA, 2008.

[5] B.P. Rimal, C. Eunmi, I. Lumb, “A Taxonomy and Survey of Cloud Computing Systems”, Int. Joint Conf. on INC, IMS and IDC, pp. 44-51, Seoul, South Korea, 2009.

[6] M. Díaz, B. Rubio, J.M. Troya, “A Tuple Channel-based Coordination Model for Parallel and Distributed Programming” Journal of Parallel and Distributed Computing, vol 67(10), pp. 1092-1107, 2007.

[7] J. Barbarán, M. Díaz, I. Esteve, D. Garrido, L. Llopis, B. Rubio, J.M. Troya, “TC-WSANs: a Tuple Channel-based Coordination Model for Wireless Sensor and Actor Networks” IEEE Symposium on Computers and Communications (ISCC’07), pp. 173-178, Aveiro, Portugal, 2007.

[8] A. Nasipuri, J. Zhuang, S.R. Das, “A Multichannel CSMA MAC Protocol for Multihop Wireless Networks” Wireless Communications and Networking Conference (WCNC’99), pp. 1402-1406, New Orleans, USA, 1999.

[9] W. Kurschl, W. Beer. “Combining Cloud Computing and Wireless Sensor Networks” 11th Int. Conf. on Information Integration and web-based Applications and Services (iiWAS’09), pp. 512-518. Kuala Lumpur, Malaysia, 2009.

[10] R. Sen. “Exchanging of Information between Cloud Computing Server and Sensor Node for Effective Application Development” Int. Symposium on Devices MEMS, Intelligent Systems and Communication (ISDMISC’11), pp. 31-34. Gangtok, India, 2011.

[11] N. Mitton, S. Papavassiliou, A. Puliafito, K.S. Trivedi. “Combining Cloud and Sensors in a Smart City Environment” EURASIP Journal on Wireless Communications and Networking, pp. 2012-247, 2012.

[12] http://www.opengeospatial.org/domain/swe

[13] M.S. Aslam, S. Rea, D. Pesch. “Service Provisioning for the WSN Cloud” IEEE Fifth Int. Conf. on Cloud Computing (CLOUD’12), 2012.

[14] W. Wang, K. Lee, D. Murray, “Integrating Sensors with the Cloud using Dynamic Proxies” 3rd Annual IEEE Int. Symposium on Personal, Indoor and Mobile Radio Commuications (PIMRC’12), pp. 1466-1471, Sydney, Australia, 2012.

[15] D. Hughes, K. Horré, N. Matthys, S. Michels, C. Huygen, W. Joosen, “LooCI: A Loosely-coupled Component Infrastructure for Networked Embedded Systems” 8th Int. Conf. on Advances in Mobile Computing and Multimedia (MoMM’09), pp. 195-203, New York, USA, 2009.

[16] P. Langendoerfer, K. Piotrowski, M. Díaz, B. Rubio, “Distributed Shared Memory as an Approach for Integrating WSNs and Cloud Computing” Workshop on Wireless Sensor Networks: Architectures, Deployments and Trends, pp. 1-6, Istambul, Turkey, 2012.

[17] K. Piotrowski, P. Langendoerfer, S. Peter, “tinyDSM: A Highly Reliable Cooperative Data Storage for Wireless Sensor Networks” Int. Symposium on Collaborative Technologies and Systems (CTS’09), pp. 225-232, Baltimore, USA, 2009.

[18] K. Lee, D. Hughes. “System Architecture Directions for Tangible Cloud Computing” 1st ACIS Int. Symposium on Cryptography and Network Security, Data Mining and Knowledge Discovery, E-Commerce and Its Applications and Embedded Systems (CDEE’10), pp. 258-262. Qinhuangdao, China, 2010.

[19] K. Lee, D. Murray, D. Hughes, W. Joosen. “Extending Sensor Networks into the Cloud using Amazon Web Services” IEEE Int. Conf. on Networked Embedded Systems for Enterprise Applications (NESEA’10), pp. 1-7. Suzhou, China, 2010.

[20] M.M. Hassan, B. Song, E. Huh, “A Framework of Sensor-Cloud Integration: Oportunities and Challenges” 3rd Int. Conf. on Ubiquitous Information Management and Communication (ICUIMC’09), pp. 618-626, Suwon, S. Korea, 2009.

[21] S.V. Patel, K. Pandey. “Design of SOA Based Framework for Collaborative Cloud Computing in Wireless Sensor Networks” Int. Journal of Grid and High Performance Computing, vol. 2(3), pp. 60-73, 2010.

[22] J. Jayashree, J. Vijayashree. “Ubiquitous Life Care Integrates Wireless Sensor Network and Cloud Computing with Security” Global Journal of Computer Science and Technology, vol. 11(15), 2011.

[23] M. Hori, E. Kawashima, T. Yamazaki. “Application of Cloud Computing to Agriculture and Prospects in other Fields” Fujitsu scientific and technical journal, vol. 46(4), pp. 446-454, 2010.

[24] H. Chen, P. Chou, S. Duri, H. Lei, J. Reason. “The Design and Implementation of a Smart Building Control System” IEEE Int. Conf. on e-Business Engineering (ICEBE’09), pp. 255-262. Macau, China, 2009.