

Tasking Wireless Sensor Nodes in the Internet of Things

Benjamin Le Corre, Jérémie Leguay, Mario Lopez-Ramos

Thales Communications**
160 Bd de Valmy, BP 82
92704 Colombes Cedex, France
{firstname.name}@fr.thalesgroup.com

Abstract. The increasing ubiquity and diversity in the resources and functionalities of the devices of the *Internet of things* claims the need for (1) a device-independent abstraction layer to expose (and consume) device capabilities as services on the network and the need for (2) mechanisms to customize the behaviour of the devices to fulfill new or additional operations. After reviewing state of the art solutions for the dynamic deployment of tasks in WSN, we lay the basis for the extension of the WSN-SOA [1] stack to support the dynamic deployment of service-oriented tasks.

1 Introduction

The increasing ubiquity of network-connected devices in our everyday environments (e.g., at home, at work, in urban settings) is driving an evolution in interaction patterns, from traditional human-computer to machine-to-machine scenarios. For instance, remote monitoring and control solutions, previously limited to industrial systems such as SCADA, are currently available in the market as home automation solutions for the automatic control of lights, shutters or heating.

The advent of the Internet of Things

In the same way, in the years to come we will witness the growth of consumer electronics based on wireless technologies such as RFID (Radio-Frequency Identification) and WSN (Wireless Sensor Networks). WSN are currently mostly devoted to the measurement of simple environmental parameters such as temperature or humidity. However, nodes are increasingly affordable and miniaturised, and power consumption optimisations are expected to make batteries last several years.

These advances will make such sensors ready to be integrated in everyday objects such as furniture or clothes. Users will be able to customize their environment not only depending on the identified objects around them, but also being able to interact with them as input or output peripherals and being able to make them interact with each other, thus realizing the dream of an *Internet-of-Things* world.

Service composition and tasking

The increasing diversity in the resources and functionalities of the devices of the Internet of things arises the need for a device-independent abstraction layer exposing

** This work has been partly funded by the EU ITEA project Do-it-Yourself Smart Experiences

the functionality of each sensor and actuator as a service on the network. In previous work [1], we presented the WSN-SOA stack, a lightweight service-oriented stack for WSN. With such an approach, ubiquitous applications involving from the simplest temperature sensor to highly optimized cloud computing web services can be easily created by using service composition and orchestration. In addition, subscription mechanisms in WSN-SOA remove the need for polling thus saving on batteries, the scarcest resource in wireless sensors.

Although such an approach is sufficient for a wide range of ubiquitous applications, autonomous cooperation between wireless sensor nodes is limited by the inability to modify the behavior of the devices in order for them to fulfill new or additional operations (e.g. event detection, object tracking, actuator control and command, data collection). In this paper, we will first review the existing solutions for the dynamic deployment of processing tasks in WSN. Then, we will lay the basis for extending the WSN-SOA for dynamic deployment of services and tasks.

2 Related work

A number of propositions exist in the literature for the dynamic deployment of treatments in wireless sensor networks. This section reviews all the solutions which enable to reprogram or to task individual or groups of sensor nodes. We evaluate these solutions against the three following criteria:

- **Global overhead:** Asking a sensor or a group of sensors to perform a set of actions consumes bandwidth for the *request phase*, which could include in some cases code deployment, and consumes memory and processing power for the *execution phase*.
- **Required programming effort:** Depending on the programming model, the level of abstractions and the amount of support functions, reprogramming a sensor network can be more or less difficult.
- **Task complexity:** The complexity of the tasks that each framework allows to deploy on the sensor nodes might be different.

Fig. 1 shows a qualitative comparison of the different approaches found in the literature with regards to the previous criteria. An ideal solution would have the maximum score for each axis.

Over-the-air flashing

The Deluge [2] software allows to reprogram sensors by flashing their ROM memory over-the-air. As the entire node program must be transmitted to nodes, it consumes a lot of resources (battery, bandwidth). It potentially requires a significant programming effort as it does not provide any programming support but potentially offers the possibility to deploy tasks of a great complexity. Note also that all the tasks running on nodes are being restarted without restoring their state when nodes are reprogrammed.

Virtual machines

Agilla [3] is a virtual machine-based middleware which allows the use of mobile agents. Mobile agents are defined in the form of programs, with a complex syntax similar to

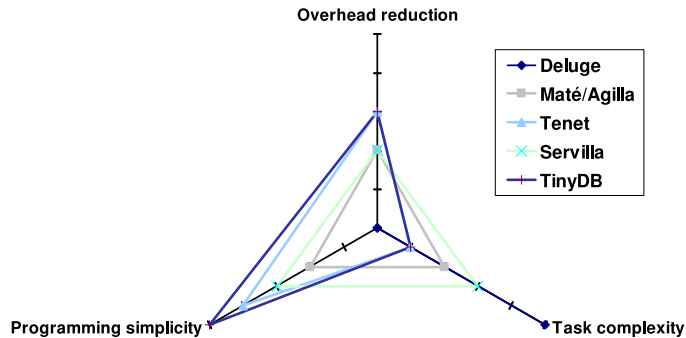


Fig. 1. Evaluation of existing solutions.

the assembly, interpreted on the sensors. Maté [4] takes a similar approach, being a byte code interpreter for TinyOS. In both solutions assembly programs can be deployed remotely by using a mobile agent injector for the first one and a code capsule for the second one. These approaches reduce the global overhead and the programming effort but the task complexity is limited by the available instructions (opcodes) in the nodes and by the fact that no communication support is offered for collaboration between sensor nodes.

Action chains

The Tenet [5] software aims at distributing processing on sensor nodes in the form of tasks. Each task consist in a chain of *tasklets* which are called sequentially. Tenet reduces communication costs with respect to virtual machine-based solution by relying on pre-deployed unitary complex operations. Tenet supports tasking with a very simple programming language but is limited by the number of tasklets available at nodes and does not offer any support for sensor nodes collaboration.

Service orientation

Servilla [6] is a service-oriented middleware running across heterogeneous wireless sensor nodes (e.g., TelosB and Imote2) on top of which one can execute scripts. Servilla relies on Agilla but provides an easier C-style scripting language and service-oriented features such as dynamic service discovery and invocation. Thanks to its service-oriented approach and its programming language, Servilla enables to deploy more complex tasks in terms of functionality and sensor nodes collaboration while keeping a global overhead similar to virtual machine-based solutions.

Macroprogramming

Whereas previous approaches focused on the modification of the local behavior of nodes, a different class of solution focuses on the global behavior of a distributed sensor network as a whole, also referred as *macroprogramming*. In this class, Cougar [7] and TinyDB [8] are considering a sensor network to be a database. They provide easy SQL-like interfaces for extracting the data of interest from sensor nodes. These solutions offer a very easy way of tasking a sensor network while consuming a fairly little global overhead. However, they are limited to data collection operations.

3 Towards the deployment of service-oriented tasks

From the survey presented in Sec. 2, we found that all the solutions face a general trade-off between programming simplicity, tasks complexity and overhead reduction. However, we argue that without adding overhead or programming complexity, Servilla and Tenet would benefit from the support of a rich SOA middleware to easily support complex collaboration between sensors. The WSN-SOA stack [1] is a good candidate as it features service discovery and invocation, eventing operations and dynamic service hosting.

The WSN-SOA protocol stack was introduced as part of a multi-tier service-oriented architecture for heterogeneous sensor networks. This architecture seamlessly integrates devices with resources ranging from those of a PDA to extremely constrained sensors such as Crossbow's MICAz. WSN-SOA enables the use of request/response and publish/subscribe paradigms in networks of constrained sensors, but it is only able to run the services that were hard-coded when the program was flashed into memory. There is no means to change the default behaviour apart from changing values for thresholds, periods, etc.

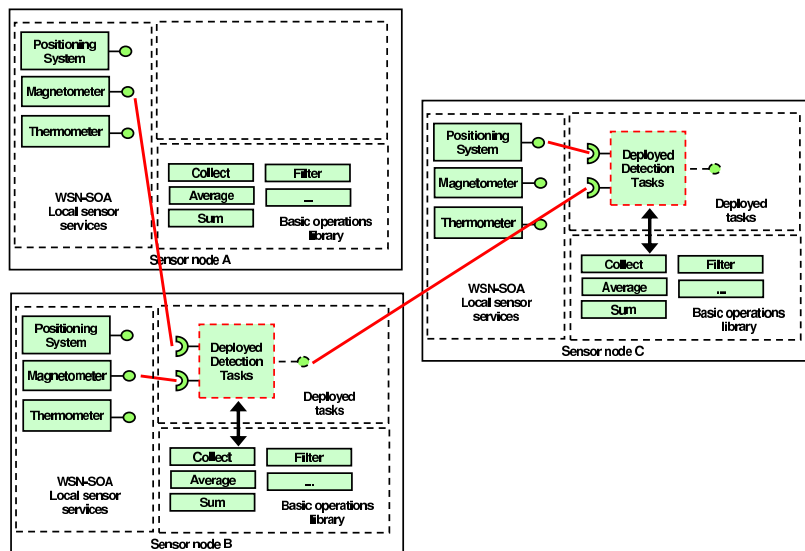


Fig. 2. Example of a service-oriented task deployment.

By combining the advantages of WSN-SOA for distributed applications and tasking mechanisms such as Servilla and Tenet, distributed processing tasks could be deployed to a number of nodes in order for them to cooperate autonomously. The tasks that one would deploy consists in simple scripts which could invoke local or remote services and manipulate data using a library of pre-programmed functions. These tasks could also

expose results of their computation to the other nodes as dynamically created services. Fig. 2 illustrates how tasks would be deployed using an extended WSN-SOA stack. In this example, a task deployed on node C uses a local positioning service and the information exposed as a service by another task deployed on node B which fuses data from magnetometer services running on nodes A and B.

In order to enable the dynamic deployment of service-oriented tasks on top of WSN-SOA, a number of technical issues should be addressed. First, a task management service has to be specified to take care of the lifecycle of tasks: (un)installation and (des)activation operations. This service offers the remote ability to list deployed tasks status. Then, a script or task interpreter with a scheduler has to be integrated. Finally, the programming language has to be enriched with service-oriented operations to invoke remote services, expose data as dynamically created services, subscribe to events and produce event notifications. Note that as sensor nodes like MICAz have highly constrained resources these extensions to WSN-SOA are required to have little memory footprint.

4 Conclusion

After having reviewed state-of-the-art solutions for the dynamic deployment of tasks in WSN, this paper has laid the basis for the extension of the WSN-SOA stack to support the dynamic deployment of service-oriented tasks. The combination of these powerful mechanisms will enable the seamless creation of cooperative behaviours in WSN, thus paving the road for a whole new set of applications in the customization of *Internet-of-Things* environments.

References

1. Leguay, J., Lopez-Ramos, M., Jean-Marie, K., Conan, V.: An efficient service oriented architecture for heterogeneous and dynamic sensor networks. In: Proc. IEEE SenseApp. (2008)
2. Delicato, F.C., Pires, P.F., da Costa Carmo, L.D.R.: A flexible web service based architecture for wireless sensor networks. In: Proc. International Conference on Distributed Computing Systems Workshops (ICDCSW). (2003)
3. Noble, B.D., Satyanarayanan, M., Narayanan, D., Tilton, J.E., Flinn, J., Walker, K.R.: Agile application-aware adaptation for mobility. In: Proc. ACM Symposium on Operating Systems Principles. (1997)
4. Levis, P., Culler, D.: Mat: a tiny virtual machine for sensor networks. In: Proc. ASPLOS-X. (2002)
5. Gnawali, O., Greenstein, B., Jang, K., Joki, A., Paek, J., Vieira, M., Estrin, D., Govindan, R., Kohler, E.: The tenet architecture for tiered sensor networks. In: Proc. ACM Sensys. (2006)
6. Fok, C.L., Roman, G.C., Lu, C.: Enhanced coordination in sensor networks through flexible service provisioning. In: Proc. of Coordination. (2009)
7. Yao, Y., Gehrke, J.E.: Query processing in sensor networks. In: Proc. of CIDR. (2003)
8. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tinydb: An acquisitional query processing system for sensor networks. In: Proc. of ACM TODS. (2005)