

Electronic Communications of the EASST  
Volume 28 (2010)



Proceedings of the  
Third International DisCoTec Workshop on  
Context-Aware Adaptation Mechanisms for  
Pervasive and Ubiquitous Services  
(CAMPUS 2010)

Middleware for the Internet of Things, Design Goals and Challenges<sup>1</sup>

Koosha Paridel, Engineer Bainomugisha, Yves Vanrompay, Yolande Berbers, Wolfgang De Meuter

6 pages

Guest Editors: Sonia Ben Mokhtar, Romain Rouvoy, Michael Wagner  
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer  
ECEASST Home Page: <http://www.easst.org/eceasst/>

ISSN 1863-2122

<sup>1</sup> This work is partially funded by the SAFE-IS project of the Research Foundation - Flanders and the MoVES project.

## Middleware for the Internet of Things, Design Goals and Challenges<sup>†</sup>

Koosha Paridel<sup>1</sup>, Engineer Bainomugisha<sup>2</sup>, Yves Vanrompay<sup>1</sup>, Yolande Berbers<sup>1</sup>, Wolfgang De Meuter<sup>2</sup>

K.U. Leuven<sup>1</sup>, Vrije Universiteit Brussel<sup>2</sup>

**Abstract:** As the number of wireless devices increases and their size becomes smaller, there can be more interaction between everyday objects of our life. With advances in RFID chips and the introduction of new generations of these devices that are smaller and cheaper, it is possible to put a wireless interface on almost all everyday objects: vehicles, clothes, foodstuffs, etc. This concept is called the *Internet of Things*. Interaction with thousands of wireless devices leads to a continuous and massive flow of events which are generated spontaneously. The question of how to deal with this enormous number of events is challenging and introduces new design goals for a communication mechanism. In this paper we argue that a middleware together with suitable linguistic abstractions is a proper solution. We also point out the challenges in developing this middleware. Moreover, we give an overview of recent related work and describe why they fail to address these challenges.

**Keywords:** Internet of Things, Middleware, Linguistic Abstractions

### 1 Introduction

As a result of the rapid advances in wireless technology and the emergence of a new generation of wireless devices which are cheaper and smaller, many objects in our daily lives are becoming wirelessly interoperable by having tiny and low-powered or even passive wireless devices attached to them (e.g. Passive RFID tags). The Wireless World Research Forum predicts that by 2017, there will be 7 trillion wireless devices serving 7 billion people [8]. This means one thousand devices per person. Imagine a world where you can interact with most of your daily objects, such as a cup of coffee, shoes, books, etc. What makes this environment different from other domains of ubiquitous computing is its much larger scale in terms of the number of things it involves and the massive number of events that can be generated spontaneously by these things.

The characteristics of the Internet of Things bring new challenges in developing applications and make the existing challenges in the area of ubiquitous computing considerably tougher. Some of them are new challenges which only appear in the environment of the Internet of Things, and some are old challenges which have been addressed to some extent in current solutions, except that those solutions are inadequate for the environment of the Internet of Things.

In a ubiquitous computing environment, it is not possible to impose standards and make everyone comply. We therefore need a system to join heterogeneous components together and to ensure interoperability between them. A middleware offers common services for applications

<sup>†</sup> This work is partially funded by the SAFE-IS project of the Research Foundation - Flanders and the MoVES project.

and eases application development. Moreover, it provides transparency with regard to the heterogeneous components for the application layer. Complementary to middleware are programming language approaches for mobile ad hoc networks [3] and wireless sensor networks [6]. These approaches attempt to tackle some of the challenges (such as discovery, network disconnections, and group communication) posed by the Internet of Things environments. However, they miss out on an important characteristic of context-awareness (e.g., context-aware service discovery) and have not been applied to a massive number of devices. In this paper we discuss some of the existing middleware and programming language approaches and identify their shortcomings. These shortcomings justify our position that new middleware and language abstractions are needed for addressing the challenges posed by the Internet of Things.

In the following, we point out the challenges in Section 2. We then give an overview of related work and their shortcomings in Section 3. We conclude in Section 4.

## 2 Challenges

1. **Larger Scale:** Scalability, as described in [7] is the ability of a network to support the increase of its limiting parameters. These are parameters whose increase causes the network performance to degrade. These parameters for a communication system in mobile ad hoc networks are network size, number of events, mobility rate, heterogeneity of devices, number of diverse interactions and number of different topics of interest [2]. We discuss the first four scalability issues here.
  - (a) **Large Network Size:** In the Internet of Things, we are talking about interaction with thousands of devices in one place. The scale is much larger here compared with the situations assumed in other systems. Larger network size makes it harder to build logical structures such as trees [5] and considerably increases the overhead of flooding techniques [10].
  - (b) **Massive Number of Events:** A significant challenge is posed by the enormous number of events generated by objects. As there are interactions with a large number of devices, it can be expected that there will also be a massive number of events occurring all the time. This has never been an issue in the previous solutions. To our knowledge, it has always been implicitly assumed that the number of events in the system is not so high that it could cause problems such as event congestion and lack of event processing capability.
  - (c) **Mobility Rate:** Higher mobility rate causes more breakage of links and causes more routing information becoming out-of-date. The network is a highly unstructured cloud of wireless devices into and out of which nodes are continuously flowing. Moreover, because of the very volatile nature of this network, an object can never be sure if another object is active or is in its proximity. When an object is not reachable, it is unknown whether it remains active. The system should be able to deal with this temporary unavailability. A high mobility rate drastically decreases the performance of methods which maintain a set of logical connections such as routing trees and clusters [10]. The Internet of Things consists of objects with various movement

speeds including objects moving at very fast speeds. Thus, we expect to have more overhead in updating routing information and maintaining logical structures.

- (d) **Heterogeneous Devices:** In the Internet of Things there is a wide variety of hardware and devices, in all shapes and sizes. These devices may vary from a real full-fledged mobile computer to a mobile phone or a PDA, or even wireless devices without any processing power, such as RFID tags. In this kind of environment objects have various wireless communication ranges and therefore need to be dealt differently.
2. **Spontaneous Interaction:** Sudden interactions happen as the objects move around and come into other objects' wireless range. This leads to the spontaneous generation of events. Unlike other context-aware systems and sensor networks, the events are not pulled by the interested parties. Here, an interaction with an object means that an event is produced and is pushed to the system without premeditation.
3. **Zero Infrastructure:** In the Internet of Things setting, devices need to discover each other as well as the resources provided by other devices in the surroundings. The challenge here is that there is no fixed infrastructure to manage resource publication, discovery and communication. Each device should be capable of announcing its presence and the resources it provides without requiring a fixed infrastructure. This is unlike the traditional distributed systems where resource publication, discovery and communication are managed by a dedicated server. The server coordinates all the interactions between devices and is assumed to be in an "always connected" state. Such assumptions do not hold in the Internet of Things setting because devices continuously move from place to place and new surrounding devices need to be discovered.
4. **Multiplicity:** Devices in the Internet of Things are not "self-contained" and often rely on the services available at other devices in the surroundings. Not only does this require selecting the most suitable services from a massive number of available services, but also dealing with massive number of results returned from different services. This challenge is further aggravated by the fact that results from different services may contradict each other. For instance, requesting for location information from nearby devices may return tens of thousands of contradicting location coordinates. Another multiplicity issue is that devices often require to communicate to a group devices at the same time.

## 3 Shortcomings of Current Solutions

### 3.1 Middleware Approaches

As we discussed earlier, in the Internet of Things environment, a system is required to join the heterogeneous components together and to provide interoperability between them. This system should process, filter and route events in a scalable manner, given the other challenges such as volatility of network and massiveness of events. Given these requirements, a middleware is a very suitable solution for routing and delivering events. A middleware offers common services for the applications and eases application development. However, there are already solutions called publish/subscribe (pub/sub) systems for delivering events from event sources to interested

clients. Although they were originally designed for wired networks, they now offer solutions for mobile and volatile environments. In the rest of this section, we describe and give an overview of pub/sub systems and focus on pub/sub systems for mobile ad hoc networks, which are closer to our type of environment. Next, we argue that using a pub/sub mechanism alone cannot address the challenges discussed in Section 2.

The pub/sub paradigm is designed to deliver events from a publishing source to interested clients in an asynchronous way. The pub/sub systems are very interesting solutions because of their decoupling properties, which means that the interacting parties do not need to know each other. An asynchronous way of communication also has the advantage that parties can communicate even if they are not up at the same time. There are already many decent solutions for pub/sub systems in wired networks and infrastructured mobile networks, where the backbone of the network is still wired. Recently, some pub/sub systems have been proposed for mobile ad hoc networks. In [11], the authors propose a pub/sub protocol for mobile ad hoc networks involving the construction of dynamic Voronoi regions. All nodes are divided into broker and non-broker categories, and each region is represented by a broker node. The authors propose mechanisms for constructing these Voronoi regions and for message deliveries. [10] also propose a pub/sub system for mobile ad hoc networks involving a hybrid model which uses flooding and content-based routing techniques hierarchically. They divide the network into clusters of nodes and use the event flooding technique for inter-cluster communication and content-based routing for intra-cluster communication. They try to utilize the advantages of content-based routing to achieve more efficient subscription propagation and maintenance and also reduce the costs of document delivery, while avoiding the high topology maintenance costs of content-based routing by using clusters and using flooding techniques for inter-cluster communication. Also in [5], the authors propose a protocol to organize the nodes of a mobile ad hoc network in a single, self-repairing tree structure which can be used for efficient content-based routing in the network. Their idea of maintaining the tree topology is inspired by a protocol for multicast over mobile ad hoc networks called MAODV.

However, the pub/sub systems focus purely on routing and delivering events. They do not take into account the possibility of having a very large number of nodes and a massive number of events. The current pub/sub systems do not scale for more than a few hundred nodes in mobile ad hoc environments. [5] do simulations with up to 100 nodes, and the performance of their system rapidly decreases as the size of network increases [10]. [11] simulates only with 200 nodes and does not study the effect of increment in the network size. Because of their structure and communication mechanism between brokers it is expected that increasing the network size will decrease the performance of the system rapidly. In [10] the authors run simulations with up to 405 nodes. Their simulation shows a sharp decrease in the document delivery rate when network size is increasing. This behavior is worse when they choose larger cluster size (e.g. 10 nodes in each cluster). On the other hand, when they choose smaller cluster sizes average energy consumption increases. The reason is that when cluster sizes are smaller document flooding will be used more than content-based routing and this means sending more messages and consequently consuming more energy. Also in the mentioned systems document publication rate is chosen between 0.5 to 1 events per second. Only [5] do simulations with more publishing rate up to 24 messages per second and the result is a rapid decrement in document delivery rate. However, in the Internet of Things environment we expect to deal with at least thousands of nodes in one

place that are continuously generating events. As we discussed in section 2, there are lots of challenges that cannot be addressed solely by pub/sub techniques, although they can be used effectively as a component of a bigger solution. The following table shows how the mentioned systems deal with scalability challenges:

System	Network Size	Number of Events	Mobility Rate	Heterogeneity
[10]	+	N/A	-	-
[5]	-	+	-	-
[11]	-	N/A	N/A	-

There have been several publications proposing a middleware to tackle some of the challenges of the Internet of Things. In [4] the authors describe their vision for the Internet of Things, and propose their smart semantic middleware to deal with the heterogeneity of the components and to allow the creation of self-managed systems. They exploit semantic technologies to provide interoperability and agent technologies for the management of complex systems. In [1], the authors propose their Global Sensor Networks (GSN) middleware to support their vision of the Internet of Things. They focus on minimizing the development and deployment costs and on data-oriented integration of data sources in what they call the "Sensor Internet" environment. Neither of these approaches aims to address scalability, event massiveness and network volatility.

### 3.2 Programming Language Approaches

Complementary to a middleware, there are programming languages that have been designed for highly dynamic environments such as mobile ad hoc networks (MANETs) [3] and Wireless Sensor Networks (WSNs) [6, 9]. Here we discuss a selection of representative language approaches and point out why they fail to cope with the challenges posed by the Internet of Things.

*SpatialViews* [6] is a language extension to Java that is specially designed for resource constrained devices in WSNs. Devices providing the same services and residing in the same location are collectively represented as a *spatial view*. The language runtime takes care of discovery, communication and migration of computations amongst nodes. In this respect, *SpatialViews* to some extent, tackles the challenge of service discovery in environments characterized by *zero infrastructure*. However, service discovery based on location and type of service is not enough considering the massive number of services that may be available at the same time providing the same functionality. Rather than bombard the user with thousands services to select from, we argue that the service discovery mechanism should be based on richer contextual information (e.g., quality, user preference, cost, availability, e.t.c.).

Many-to-Many Invocation (M2MI) [3] enables building applications that run in MANETs and provides language abstractions for broadcasting messages to multiple devices. Messages are broadcast to objects of a specified type or within a defined communication range. M2MI ensures that the application behavior is not hampered by network disconnections. For instance, devices can come and go and still become part of the ad hoc network upon reconnection. This group communication facility somehow addresses the challenge of *multiplicity* described in section 2. However, M2MI has only been applied to simple applications (chat and printer services). In the Internet of Things setting, there is need for not only to dynamically compose groups of communication, but also means to filter the massive number of results from the many invocations.



## 4 Conclusion

In this paper we described challenges in developing a middleware for the Internet of Things. We then discussed some of the existing middleware and programming language approaches and identified their shortcomings. These shortcomings justify our position that new middleware and language abstractions are needed for addressing the challenges posed by the Internet of Things.

## Bibliography

- [1] K. Aberer, M. Hauswirth, and A. Salehi. Middleware support for the “internet of things”. *GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Universität Stuttgart*, 2006.
- [2] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlays for pub-sub with many topics. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, page 118. ACM, 2007.
- [3] Alan Kaminsky and Hans-Peter Bischof. Many-to-many invocation: a new object oriented paradigm for ad hoc collaborative systems. In *OOPSLA*, pages 72–73, New York, NY, USA, 2002. ACM.
- [4] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan. Smart semantic middleware for the internet of things. In *Proc. 5th Intl. Conf. Informatics in Control, Automation and Robotics (ICINCO’08), Volume ICSO*, pages 169–178, 2008.
- [5] L. Mottola, G. Cugola, and G.P. Picco. A self-repairing tree topology enabling content-based routing in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, pages 946–960, 2008.
- [6] Yang Ni, Ulrich Kremer, Adrian Stere, and Liviu Iftode. Programming ad-hoc networks of mobile and resource-constrained devices. *SIGPLAN Not.*, 40(6):249–260, 2005.
- [7] C.A. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. On the scalability of ad hoc routing protocols. In *IEEE INFOCOM*, volume 3, pages 1688–1697. Citeseer, 2002.
- [8] M.A. Uusitalo. Global vision for the future wireless world from the wwrp. *IEEE Vehicular Technology Magazine*, 1(2):4–8, 2006.
- [9] Matt Welsh and Geoff Mainland. Programming sensor networks using abstract regions. In *NSDI’04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, pages 3–3, Berkeley, CA, USA, 2004. USENIX Association.
- [10] S. Yoo, J.H. Son, and M.H. Kim. A scalable publish/subscribe system for large mobile ad hoc networks. *The Journal of Systems & Software*, 82(7):1152–1162, 2009.
- [11] Q. Yuan and J. Wu. Drip: A dynamic voronoi regions-based publish/subscribe protocol in mobile networks. In *IEEE INFOCOM*. Citeseer, 2008.