I. J. Communications, Network and System Sciences, **2009, 1, 1-89**
Published Online February 2009 in SciRes (http://www.SciRP.org/journal/ijcns/).

Generality Challenges and Approaches in WSNs

Fadi TIRKAWI, Stefan FISCHER
Institute of Telematics, University of Luebeck, Luebeck, Germany
Email:{*tirkawi, fischer*}*@itm.uni-luebeck.de*

Received September 4, 2008; revised December 1, 2008; accepted December 31, 2008

Abstract

Ignoring the generality in the design of Wireless Sensor Networks (WSNs) applications limits their benefits. Furthermore, the possibilities of future extension and adaptation are also restricted. In this paper, several methods to enhance the generality in WSNs are explained. We have further evaluated the suitability of these methods in centralized and de-centralized management scenarios.

Keywords: Component, Middleware, Standardization, Generality, Wireless Sensor Network, Management, COTS

1. Introduction

The field of wireless sensor networks (WSN) is becoming a very popular area that starts having many applications in different fields. In addition to the benefits which one can get, this expansion also comes with many difficulties. In this paper, we target one of them, namely to provide standard and general paradigms in dealing with WSNs. Supporting platform- and hardware- independent applications for WSN is very important in order to ease the use of the big diversity of WSN applications. Moreover, in WSNs, finding generic management architectures that can be reused for managing different sensor node platforms is a challenge being posed and emphasized by the WSN community for long time. The management architecture should be capable of running over a broad range of WSN platforms and supporting a wide variety of WSN applications. The purpose of this paper is to present generality solution which can provide this feature.

WSN Management provides control and management of the system components such as the network layer, operating system parameters and application settings in real sensor nodes. Such tools that can provide generic remote interactions of the nodes are really missing.

Initially, this generality issue should be viewed in the context of the current traditional network management challenges, so that the learned lessons from generality of the traditional networks should be exploited. Therefore, we are going to discuss some similarities and distinctions of generality between the traditional and wireless sensor networks.

In traditional networks, the generality of the system provides several important benefits such as compatibility, interchangeability, and simplicity. The compatibility enables the users to take the advantage of different components in similar architectures. The interchangeability among components with different architectures is enabled. The simplicity enables having a simple and similar way of interacting with the components and using them. Generality solutions provide other advantages such as resolving the heterogeneity and providing openness. Although generality provides many advantages, it adds additional overhead to the system. However, this overhead should not cause a significant loss of the efficiency.

WSNs are applications-specific, which is, obviously, in contrary to generality. Here, it is very difficult to find general solutions due to the following reasons. The nodes in WSNs have limited resources such as small physical size, small memory, weak computation, limited energy budget, and narrow bandwidth. Moreover, there is wide range of applications in which one can apply WSNs such as environmental tracking applications, medical and industrial applications, home automation applications, surveillance systems, etc. Furthermore, there are diverse hardware equipments used in designing sensor nodes. For example, there are many sensor technologies which can be built in sensor nodes such as: ECG, EMG, humidity, temperature, vibration sensors. This diversity requires a specific design and specific settings. Finally, over the last few years of WSNs evolvement, a large number of software solutions have been introduced which have increased the software diversity and hence have complicated further finding general solutions.

This paper is organized as follows. Section 2 discusses the generality factor in WSNs and its affecting parameters. In Section 3, the proposed generality schemes are explained. Finally, we provide a brief comparison among the different generality schemes and, then we conclude with our comments.

2. Discussion

In WSNs' applications, supporting the generality can be achieved through many solutions. Furthermore, the degree of the supported generality mainly depends on the degree of similarity and the distinction among the different WSNs' applications. Additionally, the generality schemes can be evaluated from many aspects such as the complexity, mobility, openness and scalability.

In the following, we have specified the parameters which influence the generality of the system and the management components. These parameters can be classified into two main categories:

2.1. Compatible System Matrices

By these matrices of components, we mean common components which can potentially be compatible with a large number of system or management architectures. An example for a compatible system matrix is the identification of a sensor node or a group of nodes. In current WSNs applications, there are mainly two ways of naming and identifying nodes. They are the following:

- Data-Centric paradigm: Here, the node is named by one or more attributes. This paradigm is the most preferred in WSNs because data in WSNs is demanded based on certain attributes, not on the node identity itself. It also supports efficient energy consumption as compared with the Address-Centric paradigm [1]. Furthermore, Data-Centric naming provides to a group or to a single node an identification which is based on their attributes such as geographical placement, events, sensor values, time-of-occurrence, etc.
- Address-Centric paradigm: This paradigm is commonly used in traditional networks to identify a particular node depending on an initially assigned address. Each node has a fixed address which can be handled to classify the originator of each received message. This paradigm can be used mostly in small-scale WSNs applications.

Mapping between these two paradigms is possible, whether the applications are based on Data-centric or Address-Centric components. This means that a generic architecture component can be realized to resolve the heterogeneity between applications based on both identification schemes. An example of this generic identification scheme is supported by SP (Sensornet Protocol) [2].

2.2. Incompatible System Matrices

These abstractions represent those components of the system, which can not be mapped to each other. In other words, they represent the components that can not be shared with or reused in distinct WSNs applications. Example of this scheme is the use of contention-based MAC protocols such as CSMS with the scheduling-based MAC protocols such as TDMA or FDMA. Although these two schemes fulfill the same objectives, they are based on entirely different characteristics. Figure 1 represents the compatibility and incompatibility between few of the system abstraction components.

Basically, the main objective of generality is to find a way to overcome the incompatibility between the incompatible matrices. To achieve general management solutions for such heterogeneous matrices of WSNs, we propose the two following schemes:

- Management based on the compatible system abstraction matrices, in which vendors of a specific type of sensor nodes have to set already-agreed existing management matrices for instance, by following either an existing standard or using compatible general abstractions.
- Management based on additional system abstracttions. For example, adding middleware that manages the heterogeneity among incompatible components. Another way is using formal languages to converge the incompatibility of the syntax and the semantic among different components. This can be done using a derivation of ASN.1 which is used to provide efficient communication between heterogeneous applications in traditional networks.

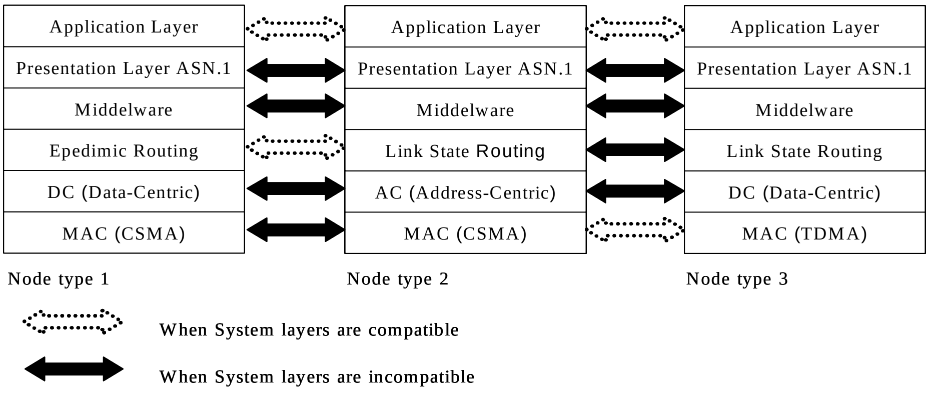

Figure 1. Nodes types heterogeneity.

3. Generality Paradigms

To achieve generic solutions for the heterogeneous matrices of WSNs, we propose five schemes. These are based on a survey of existing solutions and on schemes we have proposed. These solutions can also be integrated with the management of WSNs to provide a generic management.

3.1. Middleware

The middleware aims at providing transparent common level abstractions of one or different levels of the local or remote nodes. In other words, middleware techniques are used to resolve two main challenges. The first is interfacing homogenous applications on different platforms, and the second is interfacing two heterogeneous applications running on homogenous platforms. In WSNs, Middleware can be used to reduce and address the limitations of the application specificity. It also supports the commonness of the systems, deployment, development and maintenance. Many middleware solutions for WSNs have been proposed. Salem, *et al.* [3] have covered in their classification large number of the WSN middleware proposals such as: Mate' [4], TinyDB [5], SINA [6], and many others.

In Mate', a middle abstraction layer is provided. This layer interprets the applications as byte code. Mate' has 24 one-byte-long instructions which can be injected into the network and then propagated to the nodes. Such an abstraction hides the original platform (hardware and operating system) to unify the interpretation of the Mate' instructions; hence, it provides the portability of the applications, which are built using Mate' instruction set, among different platforms that support Mate' virtual machine.

As it is not feasible to accomplish common middleware for all kinds of WSNs' nodes, WSNs should be classified into subcategories. As a proposal, WSNs applications can be divided into high rate WSNs (industrial application, medical applications, home surveillance, etc.) and low rate WSNs (habitat monitoring, agriculture monitoring, etc.). For each of these main categories, common middleware specifications can be given.

In order to provide a general solution by using middleware, the sensor node designer should adopt one of the middleware techniques, so that these particular sensor node types support the generality of all nodes using the same middleware.

The middleware approach can be applicable for both centralized and de-centralized management solutions. In centralized management, where nodes are globally managed by one or multiple external entities such as special strong central nodes or a cluster head, the management framework on these strong nodes should follow the middleware specifications followed on the individual nodes. This can easily be achieved, as these central nodes have unlimited resources as compared to ordinary senor nodes. Therefore, they can accommodate multiple middleware of diverse existing sensor nodes. Hence, such strong nodes can provide support for different heterogeneous sensor nodes at the same time. In decentralized management, where nodes are locally establishing the management among each other, vendors should adopt the middleware which is used by other existing individual nodes, so that these different nodes have a common interface. Here, a general management is difficult to achieve due to nodes' restrictions. Figure 2 shows a representation of the system layers for both centralized and de-centralized management.

3.2. Dynamic and Mobile Agents

Mobile agents, in this context, are small pieces of code which can be exchanged between the nodes in order to resolve the heterogeneity. These mobile agents can also be added while the compilation time as the management entities in [7]. Here, the sensor nodes' manufacturer provides an agent which comprises the node specifications. These specifications can be later used to enhance compatibility with the other nodes specifications.

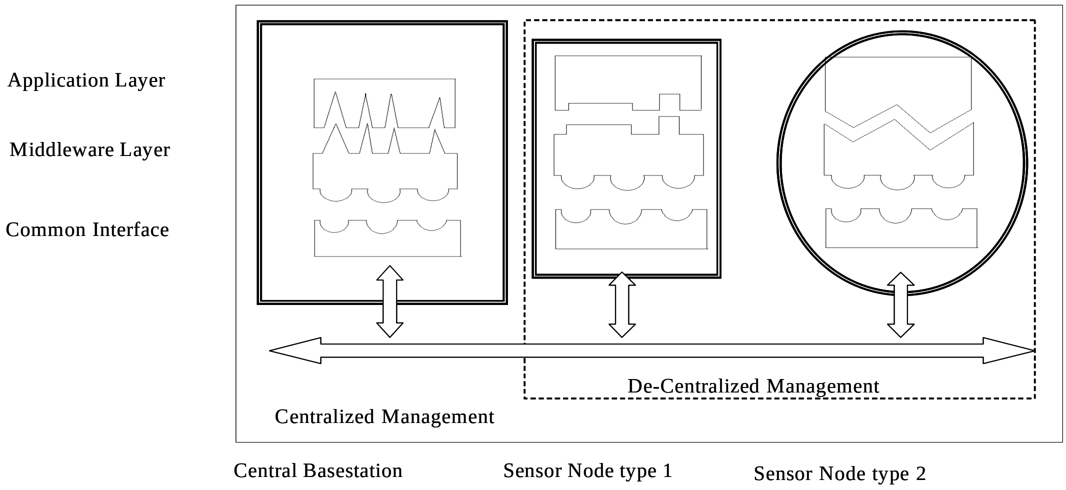

Figure 2. Representation of a middleware to have a general interface between nodes from multiple vendors.

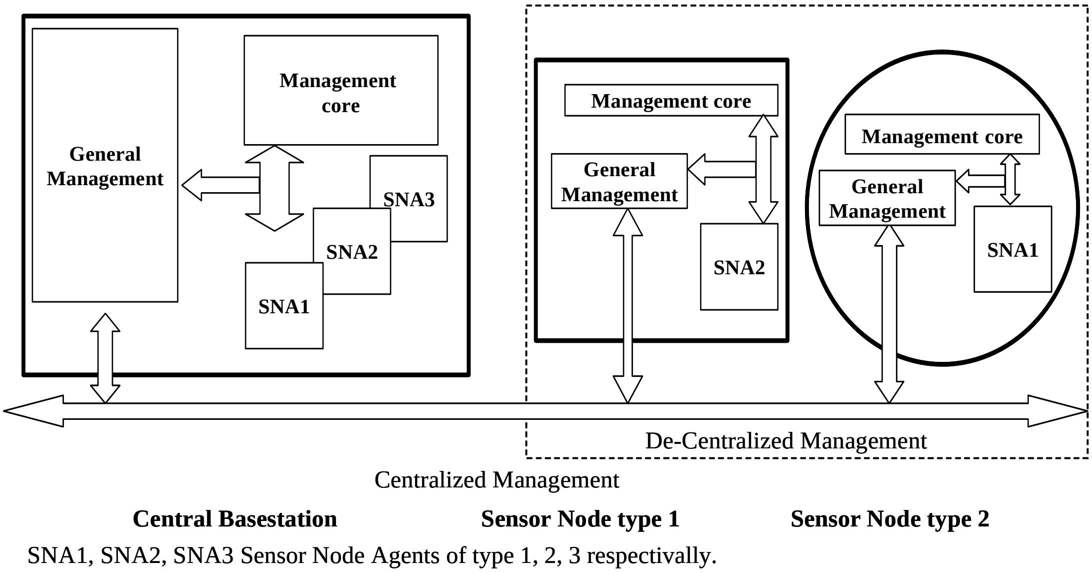

Figure 3. Representation of the mobile agent paradigm to enhance the generality of WSNs applications.

In Agilla [8], the users are able to inject the mobile agents, which are special code segments, into the nodes. These agents propagate into the nodes to perform the application-specific tasks. This fluidity of these agents has the potential to convert the nodes in WSN into a shared, general-purpose computing platform. Such platforms are capable of running several autonomous applications in parallel.

In decentralized management, the sensor nodes initially announce their specifications by exchanging their agents. Then, the received agents are configured with the management core in order to establish a general management compatible with other heterogeneous nodes. This method is very limited due to the restricted resources on the sensor nodes. Also, agents should be sent as binary code due to the complexity correlated with sending agents as sources.

In centralized management, this method is more efficient due to the unlimited resources available on central nodes as compared to the sensor nodes. As shown in Figure 3, the management core on the base station can launch the agents as dynamic libraries to resolve the specificity of other nodes.

3.3. Semantic Methods

This method is based on agreement of the functional meanings of data fields and functionality of components. Here, the exchanged messages among heterogeneous application are semantically interpreted in order to produce a general messaging structure. This general messages format can be then used to provide general management. Figure 4 represents an example of parsing semantically two different messages from heterogeneous platforms (TinyOS and Contiki), in order to have common message structure.

Due to the complexity of applying formal language concepts to generate semantically common messages from different messages, this method can be mainly set on base stations in centralized management. The complexity and the effectiveness of this method depend on the degree of overlapping of the functionality and data structures among the heterogeneous WSNs.

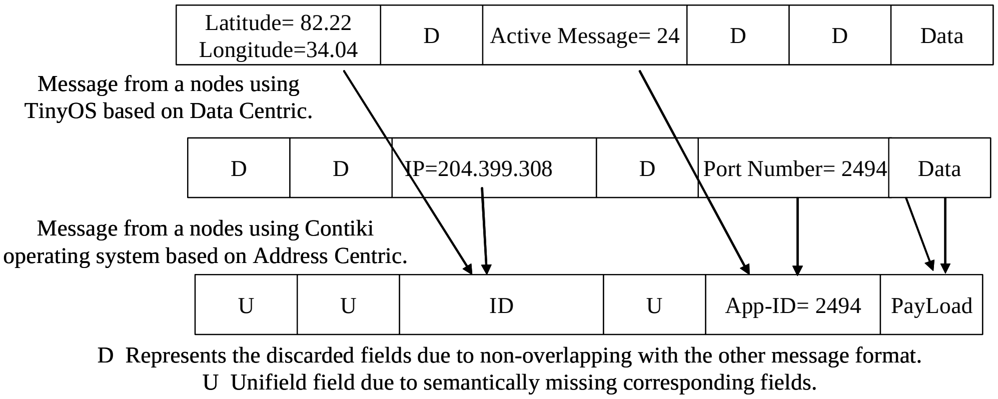
Latitude= 82.22
Longitude=34.04
D
Active Message= 24
D
D
Data
Message from a nodes using
TinyOS based on Data Centric.
D
D
IP=204.399.308
D
Port Number= 2494
Data
Message from a nodes using Contiki
operating system based on Address Centric.
U
U
ID
U
App-ID= 2494
PayLoad
D Represents the discarded fields due to non-overlapping with the other message format.
U Unifield field due to semantically missing corresponding fields.

Figure 4. How to produce semantically a general packets used in management heterogeneous WSNs.

This method is followed in a tool called Message Interface Generator (MIG) [9] in TinyOS applications. It is specific for TinyOS applications. However, it provides generality among heterogeneous application using TinyOS. In this method, the user assigns semantic names to the messages' fields according to well-known naming conventions. On a base station, compatible messages can be semantically generated from the different messages format. Then, these generated messages can be used by the management core on the base station by the management framework which can be based on JAVA, C or any other programming language.

3.4. Standards

Standardization is a way to have compatibility, interchangeability or commonality among multiple systems based on different technical and operational fields. Using standards is another way to have generality among different systems using a particular standard.

In traditional networks, there have been many management standards (technical agreement) defined by different standardization organizations such as International Organization for Standardization (ISO), International Telecommunication Union (ITU), International Electromechanical Commission (IEC), Internet Engineering Task Force (IETF), etc. Out of these groups we find many standards that are proposed for networks management solutions. Simple Network Management Protocol (SNMP) is defined by IETF. Telecommunication Management Network (TMN), which uses Common Management Information Protocol (CMIP), is defined by ITU-T based on the management's specifications of OSI/ISO.

Normally, in all standards, one or more of the system abstraction matrices are fixed. These fixed matrices have to be followed by all vendors or manufacturers. They should fulfill and cover all needed functionally at a specific level of the system. Example for that is the SNMP. SNMP, which is proposed by IETF, has specifications which should be fulfilled by all vendors who are going to introduce a solution compatible with or general to this protocol.

In the WSN field, there are standards that have been adopted such as ISO-18000-7 [10], 6lowpan [11], WirelessHART [12], ZigBee [13] and Wibree [14]. All these standards are not developed explicitly for WSNs; rather, they are mainly proposed for supporting general low power and low rate networks, which is one category of current WSNs.

SP [2] (sensor net protocol) is, so far, the only standard-alike that has been proposed specifically for WSNs. SP is a significant step forward towards generalization of wireless sensor networks. SP represents a unifying abstraction layer that bridges the different network and application protocols to the different underlying data link and physical layers. SP is not at the network layer, which is the IP layer in OSI model, instead it sits between the network and data-link layer (because in sensor nodes data-processing normally occurs at each hop, not just at the end points). SP can be used to identify an individual node, a set of nodes, or a communication structure such as a tree. SP provides generality because it enables the users to use any MAC layer without having to care about the overlaying network layers. Also, it enables users to use any network layer without having to care about the MAC specifications.

In WSN, using standards is the most efficient paradigm to obtain generality due to its simplicity and its efficiency. Standards cause lesser overhead in both centralized and decentralized management as compared to the other techniques that provide generality.

3.5. COTS (Commercial Off The Shelf)

COTS paradigm means using the components, hardware and software, available in the market in the design. Use of well-known legacy components based on existing standards enhance the generality, while introducing new components increases the specificity of the system. For instance, using the common AVR microcontrollers would ease dealing with sensor nods rather than designing new application-specific microcontrollers for WSNs, since users have to design more specific development tools and conventions in the latter case. This way is still one of the dominating ways in designing general WSNs applications.

4. Conclusions

To sum up, generality is an important feature used to address the specificity of heterogeneous platforms. As WSNs are application-specific, solutions that support the generality provide a great help in management, deployment and development. Many methods can be applied to support the generality such as using middleware, semantic interpretation, mobile agents and using agreed standards. The selection of one of these methods is based on the degree of application specificity and on the management type whether it is centralized or de-centralized.

	Centralized Management	*Decentralized Management*
Middleware	Not scalable, easy to use, resources-efficient	Scalable, difficult to use, resources-inefficient
Dynamic and Mobile Agent	Not scalable, easy to use, resources-efficient	Scalable, difficult to use, resources-inefficient
Semantic Methods	Not scalable, easy to use, resources-efficient	Scalable, difficult to use, resources-inefficient
Standards	Not scalable, easy to use, resources-efficient	Scalable, ease to use, partially resources-efficient

In general, centralized management schemes are not suitable in large scale WSNs; however, decentralized schemes perform well in such scenarios, as the management is distributed over all the nodes. The implementation and the use of different generalization schemes in centralized management are simpler than in the decentralized management. Since the management in centralized schemes is performed at the central nodes having sufficient resources, the resources of the individual sensor nodes are less consumed. The table below summarizes the comparison of the generalization methods in both centralized and decentralized management.

5. References

[1] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in the Proceedings of the 6th Annual ACM/IEEE MobiCom00.

[2] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, “A unifying link abstraction for wireless sensor networks,” in SenSys 2005.

[3] S. Hadim and N. Mohamed, “Middleware: Middleware challenges and approaches for wireless sensor networks,” 2006 Published by the IEEE Computer Society, Vol. 7, No. 3, March 2006.

[4] P. Levis and D. Culler, “Mat’e: A tiny virtual machine for sensor networks,” in the Proceedings of Architectural Support for Programming Languages and Operating Systems ASPLOS 2008.

[5] S. R. Madden, M. J. Franklin, and J. M. Hellerstein, “TinyDB: An acquisitional query processing system for sensor networks,” ACM Transactions on Database Systems, Vol. 30, 2005.

[6] C. Srisathapornphat, C. Jaikaeo, and C. Shen, “Sensor information networking architecture,” in the Proceedings of ICPPW 2000.

[7] F. Tirkawi and S. Fischer, “Remote interaction tool for wireless sensor networks,” 3rd IEEE International Symposium on Wireless Pervasive Computing, May 2008.

[8] D. C. Fok, G. Roman, and C. Lu, “Mobile agent middleware for sensor networks: An application case study,” in the Proceedings of IPSN 05.

[9] http://www.tinyos.net/tinyos-1.x/doc/nesc/ mig.html.

[10] Standards under development, http://www.iso.org.

[11] “6lowpan,” IPv6 over Low Powr WPAN, http://www. ietf.org/.

[12] “WirelessHART,” http://www.hartcomm2. org.

[13] “Zigbee,” http://www.zigbee.org.

[14] “Wibree,” http://www.wibree.com.