Middleware approaches for Wireless Sensor Networks: An overview

J.Radhika [1] and S.Malarvizhi [2]

[1] Dept. of Electronics and Communication Engineering
SRM University, Kattankulathur, 603203, India

[2] Dept. of Electronics and Communication Engineering
SRM University, Kattankulathur, 603203, India

Abstract

Wireless Sensor Networks (WSN) has become highly attractive to researchers and industries due to its wide range of applications. Middleware is a distributed software that tries to provide a common platform for hiding the heterogeneity of the system, enabling multiple applications to work on a distributed network. This survey paper focuses on the current state of research in middleware design and general issues in designing a middleware for WSN. It also examines the various approaches of middleware design, compares and suggests different types of applications where each approach can be used.

Keywords: *WSN, Middleware, Middleware approaches, Distributed networks, QoS, Heterogeneity, Scalability, Distributed software.*

1. Introduction

Wireless sensor networks consist of a large number of small scale nodes capable of limited computation, wireless communication and sensing. WSN supports a wide range of applications like object tracking, infrastructure monitoring, habitat monitoring, battle field monitoring, health care monitoring etc.

Developing applications for WSN is a tedious job as the application developers have to meet considerable number of constraints due to the rigid integration of sensor nodes to the physical world. Designing a middleware is a novel approach for addressing these constraints wherein the middleware can act as binding software between applications and operating systems (OS).

The necessity of designing a middleware software for WSN is to bridge the gap between the high level requirements from applications and the complexity of the operations in the underlying network, there are some other issues which could be well addressed by designing a middleware and they are listed below. [1]

i. Resource management at the middleware level is more easy and flexible compared to the OS layer level and application layer level because resource management at the OS level becomes platform dependent and will not be common for all applications if it is at the application layer level.

ii. Adding security features is also more appropriate at the middleware level supporting multiple applications.

iii. Integration of a WSN with other networks is possible with a middleware.

iv. Middleware can provide run time environment for supporting and co coordinating multiple applications.

Section 2 discusses the basic challenges involved in middleware design, Section 3 discusses taxonomy of middleware approaches, Section 4 deals with some existing middlewares for WSN and section 5 compares different middleware approaches for WSN. Different approaches for WSN were suggested by authors [1,2,3,4,5,6,7,8 and 9], but the suggested approaches were not specified for its suitable applications. In this survey paper, we have suggested the different applications where each approach could be used.

2. Design Principles for WSN Middleware

Based on literature review [2, 3,4,5,6 and 7] the following can be defined as the key design principles.

2.1 Data centricity

A conventional communication system normally depends upon the relationship between the sender and receiver. Communication in a WSN is data centric, wherein the application is not interested in nodes itself but rather in the data it senses. Hence middleware designed for WSN should support data centric communication.

2.2 Energy efficiency and resource management

Researches in microelectronics have made it possible to produce very tiny sensor devices. Due to small size, sensor-nodes suffer with limited resources like energy, memory etc. So the middleware on these devices should be lightweight, energy efficient and should smartly manage restrained resources in order to provide the required services while increasing the device's life.

2.3 In-network Processing and scalability

Involving the nodes also in taking decisions about how to operate a network by considering the data that they transmit is called in-network processing. Ex. Data Aggregation, Encoding and compression. In- network processing is a very important feature required for a WSN as it is a resource constrained network. Middleware designed for WSN should support in network processing and should provide acceptable level of performance even if the network grows in future.

2.4 Quality of Service (QoS)

Due to constraints in resource and topology, traditional QoS mechanisms cannot be used in WSN. In sensor networks nodes work collaboratively in monitoring and controlling the physical environment. They process each other's data thus they have to be aware of the data that they are forwarding. Hence applications for WSN have more requirements than traditional networks. Middleware designed for WSN should support application level QoS metrics like accuracy, coverage and deployment, detection reliability etc., apart from network level QoS metrics like throughput, delay etc.

2.5 Dynamic network Organization

Topology of WSN may change frequently and hence, middleware must be designed in such a way so as to support the robust sensor network operation by adapting to the changes in the networks.

2.6 Heterogeneity

A major challenge for WSN is to bridge the gap between hardware technology's raw potentials and necessary activities like reconfiguration, execution and communication. Hence middleware has to support system devices interfacing to the various types of hardware and networks [7].

2.7 Application Challenges

Application knowledge's have great influence on design principles of WSN middleware. To design middleware, it must have some resourceful mechanism to inject application knowledge into WSN's infrastructure. This allows the developers to build proper network topologies and also to frame proper QoS. For this reason, including application knowledge into middleware infrastructure is of great importance [7].

3. Classification of Middleware approaches for WSN

Existing middlewares are classified based on the physical approach and types of programming. Some of the different middleware approaches are given below: [1, 8]

3.1 Virtual Machine (VM) Middleware Approach

This approach is flexible and contains virtual machines (VMs), interpreters, and mobile agents. It basically allows developers to write applications in separate small modules. The system injects and distributes the modules through the network using specially made algorithms, so that limitations such as overall energy consumption and resource usage are minimized. The VM then interprets the modules. But this approach suffers from the overhead that the instructions introduce.

3.2 Modular Programming Approach

In this approach, applications are divided as modular programs to facilitate injection and distribution through the network using mobile code. Transmitting these small modules consumes considerably less energy than a whole application. But the code instruction doesn't allow hardware heterogeneity, which makes it incompatible for devices with limited resources.

3.3 Database Approach

This middleware approach treats the whole sensor network as a distributed database. It has an easy to use interface, using SQL like queries to collect target data [10]. It is good at regular queries, but it lacks the support for real time applications, so sometimes it only provides approximate results i.e. data rendering in real time is not applicable.

3.4 Application Driven Approach

This approach allows us to fine tune the network and to take into account resource minimization and maximum data utilization i.e. this approach basically introduces a new dimension in middleware design by supplementing an architecture that reaches the network protocol stack. This allows the programmers to fine-tune the network on the basis of application requirements i.e., applications will dictate network operations management, providing a QoS support advantage which is useful for WSN as it has limited resources. However, the tight coupling with applications might result in specialized middleware. Applications can have different QoS requirements based on different contexts. Providing QoS support to the applications is still an open research issue for WSN.

3.5 Message-oriented Approach

Message-oriented middleware uses the Publish-Subscribe mechanism to facilitate message exchange between nodes and the sink nodes. The strength of this middleware basically lies in its support for asynchronous communication, allowing a loose coupling between the sender and the receiver. This approach is quite suitable in general environments such as wireless sensor networks, where most applications are based on events.

4. Existing Middlewares for WSN

4.1 Mate [10]

It is the Middleware for WSN which uses the Virtual Machine approach and it runs on top of the TinyOS. Mate has a high level user interface and the programs of this middleware are split into 24 instruction packets called capsules which can be fitted into a single TinyOS packet. Mate is basically a stack-based architecture and it has a concise instruction set that comprises three types of instructions performing different operations. This basically means that Mate works as a byte code interpreter. The program is injected into sensor network faster and easily by using Mate. It should also be noted that Mate is energy-efficient for short running applications, but it incurs high CPU overhead for long running applications, making it only suitable for short applications.

4.2 Impala [11]

Impala uses modular programming approach. Impala basically provides mechanisms for network updates that are efficient enough to support dynamic applications. It also has autonomic behavior which increases its fault tolerance and network self-organization. However, the nature of its code instruction is that it doesn't allow hardware heterogeneity, which makes it unsuitable for devices with limited resources. The middleware itself is separated into two layers. The upper layer contains the applications and protocols. These applications use various methods to achieve the task of gathering environment information and routing it to a base station. The lower layer contains three middleware agents: event filter, adapter, and updater. The event filter controls different operations and initiates processing chains such as the timer, packets sent, and device events. Using the Application Finite State Machine (AFSM), the adapter agent handles application adaptation on the basis of different scenarios, such as energy efficiency and other attributes. The updater agent is in charge of achieving effective software updates with resource constraints by taking into account trade-offs such as high node mobility, constrained bandwidth, wide range of updates, propagation protocol, and code memory management. Impala is energy efficient and its software is easily updatable but it does not provide QoS support.

4.3 SINA [12]

SINA, Sensor Information and networking architecture is a middleware which allows sensor applications to issue queries and command tasks into the network, collect replies and results from the network, and monitor changes within these networks. SINA is mainly composed of three components: hierarchical clustering, which consists of grouping of the nodes based on their closeness or on their energy levels into clusters; attribute-based naming which replaces the standard id-based naming which cannot be used in data-centric networks like WSN; and location awareness: nodes should also know their physical location, and that is possible by using GPS. In SINA, a sensor network is theoretically viewed as a collection of datasheets wherein each datasheet contains a collection of attributes of each sensor node. This database-like system can then be queried using SQTL and SQL-like language.

4.4 MiLAN [13]

MiLAN, Middleware Linking Applications and Networks is based on application driven approach providing QoS support to applications. MiLAN contains a network protocol stack which is used to configure and manage the network. It also uses a graph based approach to allow application to know how it performs using the collected data from different combinations of heterogeneous sensors and how to choose combination of sensors to satisfy its QoS requirements. MiLAN was originally designed for medical advising and monitoring. The major disadvantage here is that it doesn't address mobility and lacks support for OS and hardware heterogeneity due to its tight coupling with the network stack. It also requires that applications should have prior knowledge of the type of sensors and no evaluation results are provided for this middleware.

4.5 MidFusion [14]

MidFusion is also a middleware based on application driven approach trying to provide QoS support. But it is suitable only for networks which are Bayesian modeled.

Other middlewares providing QoS support are [15 16, 17, and 18]. [15] provides QoS support for applications by concentrating on fault detection and management. [16] tries to provide partial QoS support to the applications by providing a set of predefined application profiles with different QoS requirements for each application. If the network is unable to support the current QoS requirement of an application profile, the middleware tries to meet the next application profile with slightly lesser requirements which are predefined. [17] provides a cluster based architecture for QoS Support and concentrates on delay as well as energy minimization .[18] provides a service oriented QoS architecture which is based on active regulation with feedback and negotiation.

4.6 Mires [19]

Mires is an adaptation of the message-oriented middleware, used for traditional fixed distributed systems. It provides an asynchronous communication model suitable for WSN applications, which are event driven and has more advantages over the traditional request-reply model. It also adopts a component-based programming model using active messages to implement its publish-subscribe-based communication infrastructure. Mires' architecture basically includes a core component (a Publish-Subscribe service), a routing component, and some additional services, such as data aggregation. The Publish-Subscribe service manages the communication between middleware services. It also manages the topics list and the subscribed application to give the right topic to the related application. Mires sends only messages referring to subscribed topics, hence reducing the number of transmissions and energy consumption. Mires does not provide QoS support and security.

4.7 TinyLIME [20]

This is also basically a database approach but unlike other database approaches it has a different programming paradigm. TinyLIME is a database middleware built over TinyOS which is based on LIME [21].TinyLIME extends LIME by adding features specialized for sensor networks which are not supported by LIME. But, TinyLIME is designed for environments in which clients only need to query data from local sensors. It does not provide multi-hop propagation of data through the sensor network. These assumptions severely limit the kinds of applications for which TinyLIME is suitable

4.8 TinyDB [22]

It is designed and implemented as Acquisitional Query Processing (ACQP) system for collecting data from a sensor network. Comparing it to traditional technology, it has capabilities such as low power consumption and accurate query results. These are important advantages in a resource limited network environment. TinyDB is a distributed system. It runs on the top of TinyOS, with Structured Query Language (SQL) like interface to execute data from sensor nodes.

4.9 Cougar [23]

Cougar applies database approach in sensor network. Basically, there are two types of data: stored data and sensor data. Signal processing functions in each node generate the required sensor data, and the data are communicated or stored in local storage facilities as relations in database system.Signal processing functions are modeled by using Abstract Data Type in Cougar. Cougar also uses SQL like language to implement queries.

4.10 MagnetOS [24]

This middleware uses a VM. MagnetOS is a power-aware and an adaptive operating system which is specially designed for sensor and ad hoc networks. It constitutes a layer called a Single System Image, which provides higher abstraction for the heterogeneity of ad hoc networks' distributed nature. This abstraction lets the whole network appear as a single, unified Java VM. It is also imperative to note that MagnetOS provides a robust, power-aware algorithm that uses object migration of the same application to nodes that are topologically closer together. This mechanism reduces application energy consumption and increases longevity.

Table 5. Comparison of Middleware approaches for WSN

Approaches	Openness	Scalability	Heterogeneity	Mobility	Ease of use	Power awareness	Application type
Virtual Machine Approach	Full	Full	Partial	Full	Little or None	Yes	Dynamic applications
Modular Programming	Full	Full	Little or None	Full	Full	Yes	Dynamic applications
Database Approach	Little or None	Little or None	Little or None	Little or None	Full	Partial	Event driven applications
Application Driven Approach	Full	Full	Little or None	Little or None	Full	Yes	Real time applications
Message Oriented Approach	Full	Full	Partial	Partial	Full	Yes	Event driven applications

5. Middleware Comparison

In this section we try to compare the different middleware approaches based on the applications for which they could be used. Table 5 presents the comparison between different middleware approaches .Apart from the comparisons included in [3, 9] we have suggested the different applications where each middleware approach could be used. All the comparison parameters except openness and ease of use are explained already in section 2. Openness is the ability to extend and modify the system easily as functional requirements change and Ease of use refers to what degree does a middleware's interface relieve the user from handling the heterogeneity of the network.[3]

6. Conclusion

Wireless sensor networks, an emerging technology, is expected to change our lives in the near future. In this survey, we went through the design principles, the different middleware approaches and some existing middlewares for WSN and then compared the different approaches by including suggestions about where each approach could be used. In this process we observed that, while scalability and power saving issues can be compensated, it comes at a trade-off with QoS. Therefore, from this survey we can safely conclude that although middleware has been able to compensate for most of its issues by trade-offs with QoS, there is a lot of research yet to be carried out before a perfect middleware for WSN can be built and tested.

References

[1] Miaomia Wang, Jiannong Cao, Jing Li and Sajal K Dasi., "Middleware for Wireless Sensor Networks:A survey," Journal of Computer Science & Technology, Volume No.23, Number 3, 305-326, (May 2008).
[2] Hadim, S. and Mohammed, N., "Middleware for Wireless sensor Networks: A Survey," Proc. The First International Conference on Communication System Software and Middleware (COMSWARE) Pages 1-7(2006).
[3] Hadim, S. and Mohamed, N., "Middleware Challenges and Approaches for Wireless Sensor
Networks," IEEE Distributed Systems Online,Vol 7, Pages 1-23 (2006).
[4] Yu, Y., et al., "Issues in designing middleware for wireless sensor networks," IEEE Network Magazine, vol. 18, Pages 15-21 (2004).

[5] Henricksen, K. and Robinson, R., "A Survey of Middleware for Sensor Networks: State-of-the-Art and Future Directions," MidSens'06, Melbourne, (2006).
[6] Culler, D., Estrin, D. and Srivastava, M., "Overview of Sensor Networks," IEEE Computer Society,Vol.18 Pages 41-49 (2004).
[7] Kay Romer., Oliver Kasten. and Friedemann Mattern., "Middleware Challenges for Wireless sensor Networks," Proc. COMSWARE, volume 6, pages 1321–1329, (2005).
[8] Wassim Masri., Zoubir Mammeri., "Middleware for Wireless Sensor Networks: A Comparitive analysis," IFIP International Conference on Network and parallel Computing workshops Pages349-356 (2007).
[9] Md.Atiqur Rahman., Helsinki. "Middleware for Wireless Sensor Networks: Challenges and Approaches," TKK T-110.5190 Seminar on Internetworking,(2009).
[10] Levis, P. and Culler, D., "Mate: A Tiny Virtual Machine for Sensor Networks," Proc. International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), (2002).
[11] Liu, T. and Martonosi, M., "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," Proc. Ninth ACM SIGPLAN symposium on Principles and practice of parallel programming, California,Vol 38, Pages107-118 (2003).
[12] Srisathapornphat, Jaikaeo, C. and Shen, C.C., "Sensor Information Networking Architecture," Int. Workshops on Parallel Processing, TorontoVol. 18,Pages 23-30, (2000).
[13] Heinzelman, W., Murphy, A., Carvalho, H. and Perillo, M., "Middleware to Support Sensor Network Applications," IEEE Network, Vol.18 Pages6-14 (2004).
[14] Hitha Alex.Han Kumar., Berouoz Shirazi., "Midfusion: An adaptive middleware for information fusion in sensor network applications," Information Fusion, Elsevier. Vol 9, issue no3, Pg no: 332-343, (2008).
[15] Linnyer Beatrys Ruiz., Isabela G. Siqueira., Leonarda, B. E. Oliveira., "Fault management in event driven WSN," Proc. MSWiM '04 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems Pages 149-156, (2004).
[16] Flavia C. Delicate., Paulo F. Piers., Luiz Rust., Luci Pirmez,Jose Ferreria de Rezende., "Refelective middleware for wireless sensor networks," Proc. ACM symposium on Applied computing ,Pages 1155-1159 (2005).
[17] Mohesen Shariff., Majid alkaee Taleghan., Amirhosein Taherkordi., "A middleware layer mechanism for QoS support in wireless sensor networks," Fifth Int. conference on networking, IEEE conference page 118, (2006).
[18] Nan Hua, Ning Yu. and Yi Guo., Research on service oriented and middleware based active QoS infrastructure of wireless sensor network," IEEE tenth international symposium on Infrastructure Pervasive systems,algorithms and networks(ISPAN) Pages.208-213, (Dec. 2009).
[19] Souto, E., Guimaraes, G., Vasconcelos, G., Vieira, M., Rosa, N., Ferraz, C. and Kelner, J., "Mires: a publish subscribe middleware for sensor networks," Personal Ubiquitous Comput 10(1)Pages:37–44, (2005).
[20] Curino, C., Giani, M., Giorgetta, M. and Giusti, A., "TinyLIME: Bridging mobile and sensor networks through middleware," 3rd IEEE Int. Conference on Pervasive Computing and Communications, Hawaï, Pages:61-72 (2005).
[21] Murphy, A. L., Picco, G.P. and Roman, G. C., "LIME: A middleware for physical and logical mobility," Proc. 21 Int. Conference Distributed Computing systems, Phoenix,Pages:524-533 (2001).
[22] Madden, S., Franklin, M., .Hellerstein, J. and Hong, W., TinyDB: An Acquisitional Query Processing System for Sensor Networks, Journal of ACMTODS Pages 122-173. (2005)
[23] Al Demers., et al., "The Cougar Project: a work in progress report," ACM SIGMOD record Vol.32, Pages 53-59(2003).
[24] Barr, R., et al., "On the Need for System Level Support for Adhoc and Sensor Networks," ACM SIGOPS Operating Systems Review, vol.36, issue. 2, pp. 1-5, (April 2002).

First Author
J.Radhika is a Assistant Professor in the Department of ECE and pursuing Ph.D in S.R.M university in the area of Wireless Sensor Networks .She has completed B.E in Electronics and Communication Engineering and Masters in Communication Systems in Anna University, Chennai, India. The research interests include Wireless sensor networks, Middleware design, etc.

Second author
Dr. S. Malarvizhi is a Professor and Head, in the School of ECE, SRM University performing teaching, research, design and consulting in the general area of Electronics and Communication Engineering. She completed Masters in the area of Applied Electronics and Ph.D in Anna University, Chennai, India. Having a total teaching experience of 20 years, she has published in all reputed journals and conferences. She is a member of IEEE and reviewer for many international conferences and journals. Her research interests include Wireless Sensor Networks, MIMO OFDM implementations, VLSI, signal processing, embedded security,etc.