ISSN 1512-1232

# CENTRALIZED SCHEMES OF FAULT MANAGEMENT IN WIRELESS SENSOR NETWORKS

Muhammad Zahid Khan, Muhammad Asim, Ijaz Muhammad Khan

School of Computing and Mathematical Sciences,
Liverpool John Moores University
Byrom St. Liverpool, L3 3AF, UK
M.Zahid-Khan@2008.ljmu.ac.uk, M.Asim@ljmu.ac.uk, I.M.Khan@2006.ljmu.ac.uk

***Abstract***

*Wireless Sensor Netwroks (WSNs) have a variety of applications, such as military surveillance, industry monitoring, mass vehicle control and smart home etc. In order to have an effective deploymet of WSNs, having an efficeint fault managemet solution is crucial. In this context, fault management has attained growing research interest. Concerning fault management various schemes are reported which could be classified into one of the three types. It is a well known fact that none of these schemes belonging to different categories has been successful in resolving this issue. Thus, there is a need to have a solution which can full-fill typical fault management requirements of a WSN. The contribution of this paper is to establish a clear and concise understanding leadings towards the root of the fault management problem. In this context, this paper critically review schemes which fall into the centralized structrucal approach in WSNs. We believe through such excersize provides a great background to establish new and effective fault management solutions for WSNs.*

***Keywords****: Wireless sensor networks, fault management*

## 1. Introduction

Recent technological advances in wireless networking and communication, the development of MEMS (Micro-Electro-Mechanical-Systems), and its integration with embedded microprocessors have enabled a new breed of wireless networks known as Wireless Sensor Networks (WSNs). WSNs are composed of a large number of self-organized sensor devices (homogenous and heterogeneous) that work in collaboration to monitor the physical environment and object of interest and relay messages to the Sink or Base Station. Sensor devices usually consist of a number of physical sensors, gathering environmental data like temperature or light, a microcontroller, processing the data, and a radio interface to communicate with other nodes [1]. These sensors have strict resource constraints and normally operate on batteries.

The design of WSNs is influenced by many factors including fault tolerance [2, 3]. Because, sensor nodes WSNs are expected to operate autonomously for a long period of time and may not be easily approachable for battery replacement and maintenance due to their physical deployment location. Furthermore, harsh physical environment e.g. rain, fire and falling of hard objects on senor hardware can also completely damage the device, hence faults and failures are normal facts in wireless sensor networks. Thus, in order to guarantee the network quality of service and performance, it is essential for the wireless sensor networks to be able to detect faults, and to perform something akin to healing and recovering from events that might cause faults or misbehavior in the network, hence fault tolerance should be seriously considered in many wireless sensor network applications [5]. A set of functions or applications designed specifically for this purpose is called a *fault-management platform*, which is an integral part of a network management system. Thereby, a network's management system with an efficient fault management platform makes the network fault tolerant in the events of faults and failures.

Fault management techniques are seen to be important aspects in the design of WSNs applications. In essence, some of the fault management issue can be resolved in the light of mobile

ISSN 1512-1232
ad-hoc sensor's network solutions. An ad-hoc network is a collection of mobile device establishing network in the absence of any fixed infrastructure[4, 5]. An example of such solutions is mobile ad-hoc on-demand data delivery protocol[6-8]. Proper implementation of fault management can keep the network running at an optimum level and minimize the risk of failure, consequently, make the network more fault tolerant[2]. In this paper, we overview some of the most dominant centralized architecture based approaches developed for fault management in WSNs. The rest of the paper is organized as follows. In section 2, gives a background about faults and fault management in WSNs, in section 3 dominant fault management schemes based on centralized architecture have been analyzed. It followed by discussion in section 4 and conclusions and future work is given in section 5.

## 2. Background

To comprehend fault management, it is important to point out the difference between faults, and failures. A fault is any kind of defect that leads to an error. A failure is a state which occurs when the system deviates from its specification and cannot deliver its intended functionality. Liu et al. [9], classify fault tolerance into four levels from the system point of view such as: hardware layer, software layer, network communication layer, and applications layer. Faults at hardware layer are caused by malfunctioning of any hardware component of a sensor node, such as processing unit, memory, battery, sensing unit, and network transceiver [9, 10].

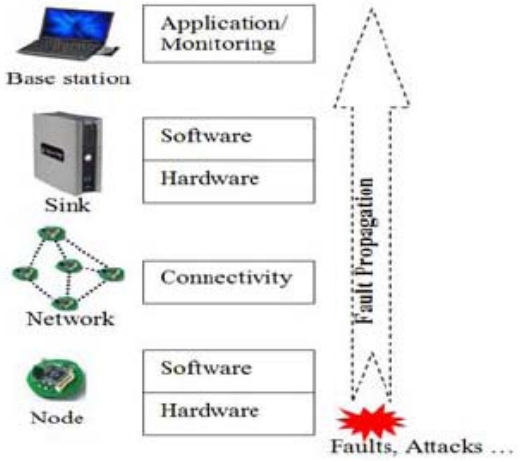


Figure 1. Fault Classification and Propagation

In the next paragraph, we further explain hardware faults, since we primarily concentrate on these faults. In fault management research literature [3, 11], node hardware fault has been categorized into four types such as: *Permanent faults, Intermittent faults, Temporary faults* and *Potential faults*.

- **Permanent faults –** Permanent faults are continuous and stable in nature e.g. hardware faults within a component.
- **Intermittent faults –** An intermittent fault has the occasional (such as a regular or irregular interval) manifestation due to unstable characteristics of the hardware.
- **Temporary or transient faults –** These faults are the result of some temporary environmental impact on otherwise correct hardware, e.g. the impact of cosmic radiation on the sensor.

- **Potential faults –** Potential faults are usually occurring due to the depletion of node hardware resources, such as node's battery energy exhaustion.

A) *Fault Management*

Fault management is a very important component of network management concerned with detecting, diagnosing, isolating and resolving faults and errors. Fault management can be defined as

ISSN 1512-1232
a set of services and functions performed to detect, diagnose, isolate and rectify malfunctions in a network. It also involves compensation for environmental changes, monitoring and examining errors logs, accepting and acting on error detection, tracing and identifying faults. Furthermore, carrying out sequences of diagnostics tests, correcting faults and failures, reporting error conditions and localizing and tracing faults are part of the fault management functions [12]. Important functions of fault-management include:

- Definition of thresholds for potential failure conditions
- Constant monitoring of system status and usage level
- General diagnostics
- Alarm and the notification of any error or malfunctions
- Tracing the location of potential and actual malfunctions
- Automatic correction of potential-problem causing conditions
- Should keep the probability of false alarm as minimum as possible
- Recovery of failures.

Fault management for WSNs is different from traditional networks. Recent research has developed several schemes and techniques that deal with different types of faults at different layers of the network. To provide resilience in faulty situations three main actions (fault detection, fault diagnosis and fault recovery) (Figure -) must be performed [2, 3, 13].

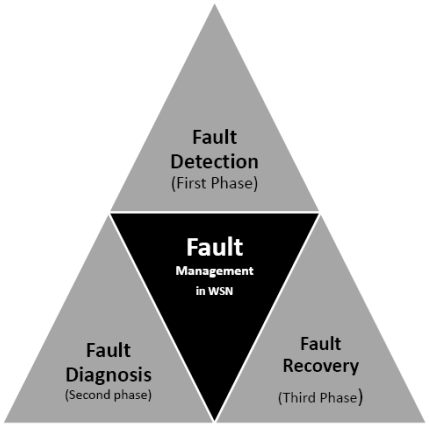


Figure 2. Fault Management Phases

- **Fault Detection -** Fault detection is the first phase of fault management, where an unexpected failure in the network should be properly identified by the networks system. Fault detection in sensor networks largely depends on the type of applications and the type of failures.
- **Fault Diagnosis -** Fault diagnosis is a stage in which the causes of detected faults can be properly identified and distinguished from other irrelevant alarms.
- **Fault Recovery -** After fault detection and fault diagnosis; it is seen in fault recovery that how faults can be treated [2]. The failure recovery phase is the stage at which the sensor network is restructured or reconfigured, in such a way that failures or faults nodes do not impact further on network performance [3].

Fault management in WSNs can be classified according to their management system network architecture [14, 15]: *Centralized, Distributed, or Hierarchical.*

- **Centralized Architecture -** In a centralized management architecture, the base station acts as a central controller or a central manager station that collects information from the whole network and control the entire network.
- **Distributed Architecture -** Instead of having a single central controller, distributed management architecture employs multiple manager station throughout the whole network.

ISSN 1512-1232

Each manager controls a sub-region of the network and may communicate directly with other manager station in a cooperative manner in order to perform management functions. Local processing and management reduces network bandwidth requirements and processing at the central controller.

- **Hierarchical Architecture** - Hierarchical management architecture is a hybrid between the centralized and distributed approach. Sub-controller or managers are distributed throughout the network in a tree shape hierarchical manner, having levels of lower and higher level of hierarchy. These managers are referred to as the Intermediate mangers, manage a sub-section of a network and perform the management functions.

## 3. Centralzed architecture based schemes for fault management

Centralized approaches, the central entity (e.g. control center or base station) carry out most of the fault management and maintenance tasks, because it has powerful and unlimited computing and energy resources. The central node adopts an active detection model to detect faults by periodically send queries into the network. On the basis on this information it identifies and localizes the faulty and misbehaving nodes in the network. In centralized fault management systems, usually a geographical or logical centralized sensor node identifies failed or misbehaving nodes in the whole network. This centralized node can be a base station, a central controller or a manager. [16]. Some common centralized fault management approaches are as follows:

The most dominant schemes for fault management approaches based on centralized architecture are: SNMS (Sensor Network Management System) proposed by Tolle and Culler [17], Sympathy (a debugging system for sensor networks) [18], sNMP (sensor Network Management Protocol) [19], and Efficient tracing of failed nodes in sensor networks, proposed by Staddon et al. [20].

SNMS is an interactive system for monitoring the health of sensor networks. SNMS provides two main management functions: query-based network health data collection and event logging. Query-based network health data collection, allows the user to collect and monitor physical parameters of the node environment. For instance, the value of node's remaining battery power can be used to predict node failure. While, the event-driven logging system allows the user to set event parameters which allow nodes to report their data only if they have met the specified event thresholds set by the user. For instance, the physical surrounding like temperature and humidity of the sensor node can also be the indicators of upcoming failure. However, the centralized processing approach in SNMS requires continuous polling of network health data from managed nodes to the base station, and this can burden the energy constrained sensor nodes ultimately minimizing the network lifetime [15]. In addition, another main drawback of SNMS are that it can only perform passive monitoring of the network,

A centralized sink location based scheme Sympathy [21] provides a debugging technique to detect and localize faults that may occur from interactions between multiple node. Sympathy has two main types of nodes: Sympathy-sink and Sympathy-node. Sympathy-sink is a sensor node that make request to the Sympathy-node for event data. The Sympathy-node is the sensor node that monitors network metrics, observe the environmental events, and send the requested data back to the Sympathy-sink. For detecting and debugging Sympathy uses a message-flooding technique to pool event data and current states (metrics) from sensor nodes. Upon receiving the node states metrics, the Sympathy-sink analyses them to process the event context. Failures are detected using flow metrics. Specifically, Sympathy determines whether the sink has received sufficient data from every component on every node over the past epoch. Insufficient data indicates a failure. To conserve energy and to avoid excessive message sending Sympathy node can selectively transmit important events to the Sympathy Central Sink node. Sympathy out forms traditional debugging and fault detection techniques, because they assume that wireless nodes have unlimited resources, and node only fails as a result of local causes. Sympathy takes into account the interactions between the neighbour nodes, provides a mechanism to analyse the context of an event, maintain network

ISSN 1512-1232

states, and identifies events of interest pro-actively. However, Sympathy does not provide an automatic debugging mechanism, in addition it also lacks an adequate fault recovery strategy [15].

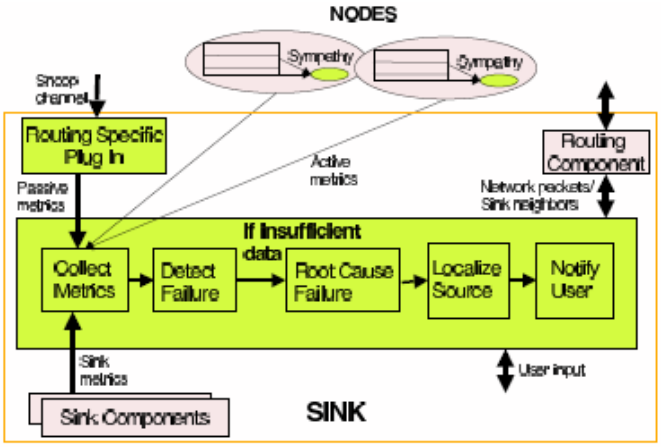


Figure 3. Sympathy System: high level versions of interface between components, and an overview of Sympathy's failure localization algorithm

Furthermore, for detecting and debugging Sympathy uses a message-flooding technique to pool event data and current states (metrics) from sensor nodes. A continues code is need to be running on a resource constrained sink node for monitoring, and it expect that all the live nodes in the network need to generate traffic of some kind (such as routing updates, time synchronization beacons etc.), which consumes a significant amount of energy of a sensor node.

Staddon *et al.* [22] while tracing failed nodes in the network – proposed a similar centralize management approach, whereby, the central manager monitors the health of individual sensor nodes to detect node failures in a network. The central manager or base station to construct an overview of network by integrating each piece of network topology information (i.e. node neighbor list) embedded in node usual routing message. This approach uses a simple divide-and-conquer rule to identify faulty nodes. It assumes that base station is able to directly transmit messages to any node in the network and rely on other nodes to route measurements to the base station. This first step enabled the base station to know the network topology and for this purpose it executes route-discovery protocols. Once the base station knows the node topology it then detects the faulty node by using a simple divide-and-conquer strategy based on adaptive route update messages. However, there is an excessive message exchange between sensor nodes and the central manager. Furthermore, this approach assumes that each node has a unique identification number, which is not suitable for large-scale WSNs.

Deb et al. [19] proposed a centralized based sensor network management framework called sNMP (Sensor Network Management Protocol). sNMP, framework defines sensor models (network topology, energy map and usage patterns etc), that represents the current state of the network and defines various network management functions. It also provides tools and algorithms for retrieving network states through the execution of different network management functions. The human manager in sNMP periodically monitors the network states, and maintains the network by identifying which part of the network has a low performance, and takes the corrective actions as necessary. The periodic monitoring of the network states helps in analyzing the network dynamics to predict potential failures and then to take preventive actions. However, the centralize-processing approach requires continuous polling of data from nodes to the base station, which puts an extra burden on energy-scarce sensor nodes [15]. In addition, another main drawback of sNMP framework is that it requires an external human manager to perform periodic monitoring.

SPINs (Security Protocol for Sensor Networks) [23] is a common routing protocol that can also detect failed or malicious nodes through routing discovery and the update phase. However, it does not provide any fault diagnosis and recovery mechanism. Similarly, MOTE-VIEW is a visualization tool designed for monitoring and managing WSNs. MOTE-VIEW uses a centralized

ISSN 1512-1232

management architecture, where all data, monitoring and management processing is performed by the central server. The architecture uses the large amount of generated data to monitor the health and status of individual sensor nodes and the network as a whole. MOTE-VIEW performs passive monitoring, which does not allow network to be self-configured themselves in the event of node failures [15].

Research work as MANNA [24], WinMS [25] etc proposed management architecture to look after the overall network by a central manager . MANNA [24] is a policy-based approach using external managers to detect faults in the network. MANNA assigns different management roles to various sensor nodes depending on the network characteristics (Homogenous vs. heterogeneous). These distinguish nodes exchange request and response messages with each other for management purpose. To detect node failures, agents execute the failure management service by sensing GET operations for retrieving node states. Without hearing from a node, manager declares it as a faulty node. MANNA has a drawback of providing false debugging diagnosis. There are several reasons a node can be disconnected from the network. It can be disconnected from its cluster and not able to receive any GET message. GET message can be lost during environmental noise. Random distribution and limited transmission range can also cause disconnection. In addition, MANNA requires manual configuration and human intervention to setup agents, which is not practical for sensor networks deployed in the inaccessible terrain. Also, this scheme performs centralized diagnosis and requires an external manager.

WinMS [25] provides a centralized fault management approach. It uses the central manager with global view of the network to continually analyses network states and executes corrective and preventive management actions according to management policies predefined by human managers. The central manager detects and localized fault by analyzing anomalies in sensor network models. The central manager analyses the collected topology map and the energy map information to detect faults and link qualities. It has the ability to self-configure in case of failure, without prior knowledge of network topology. Also, it analyses the network state to detect and predict potential failures and perform action accordingly. One of the main advantages of WinMS is that it has a lightweight TDMA protocol design; that it adaptively adjusts the network by providing local and central recovery mechanisms. and provides energy-efficient management. Furthermore, WinMS uses a pro-active technique to instruct nodes to send data less frequently to conserve energy. A disadvantage of WinMS is that the initial setup cost for creating a data gathering tree and node schedule is dependent of the network density [15].

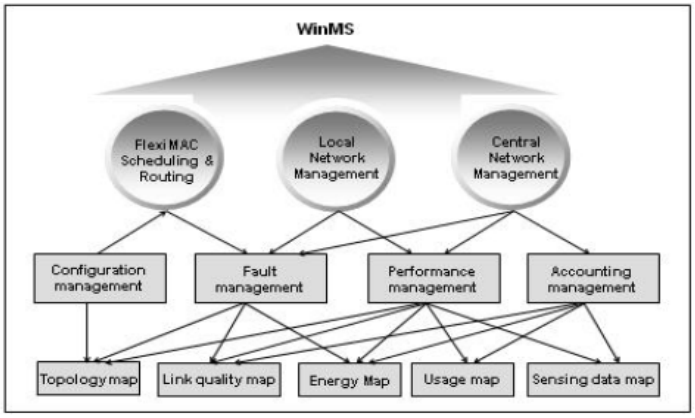


Figure 4. WinMS Architecture

## 4. Discussion

In the previous section we discussed and analyzed some of the most dominant schemes in fault management based on centralized architecture. In a centralized architecture, the base station acts as a central controller or a central manager station that collects information from the whole network and control the entire network. The base station or the central manager has rich and

ISSN 1512-1232
unlimited resources, hence to perform complex management tasks, reducing the processing burden on resource constrained nodes in the sensor network. The central manager has the global knowledge of the entire network (i.e. the topology map, residual energy of the nodes, communication coverage map etc.), therefore it can provides an accurate and reliable management decisions. However, centralized architecture incurs a high message overhead (bandwidth and energy) from data polling, and this limits its scalability. Since the whole network data traffic is towards the base station which results in high congestion rate, which degrade the performance of the network. Moreover, the central controller is the single point of potential failure. Finally, if a network is partitioned, then nodes that are unable to reach the central base station are left without any management functionality and that node is simply isolated.

It is clear from the literature survey that different centralized architecture based approaches for fault management in WSNs suffers from the following problems [26-30]:

- Most existing solutions mainly focus on failure detection, and there is still no comprehensive solution available for fault management in WSNs from the management architecture perspective.
- Different mechanisms proposed for fault recovery i.e. [31], are not directly relevant to fault recovery in respect of the network system level management i.e. network connectivity and network coverage area etc.
- Failure recovery approaches are mainly application specific, and mainly focus on small region or individual sensor nodes thereby are not fully scalable.
- Some management frameworks require the external human manager to monitor the network management functionalities, such as MOTE-VIEW, sNMP and TinyDB.
- Another important factor that needs to be considered is vulnerability to message loss. For example, in MANNA [24], if a cluster-head does not hear from its cluster member than it announced it as a faulty node. However, a message can be lost due to various reasons. It can be lost during transmission and cause a correct node to be declared as faulty.

It can be concluded from the above discussion that centralized approaches are suitable for certain application. However, it poses various limitations such as these are not scalable and cannot be used for large-scale WSNs. In a nutshell, centralized fault management approaches suffer from many problems such as: insufficient scalability, availability and flexibility when a network becomes more distributed [12].

We therefore contend that there is still a need of a new fault management scheme to address all the problems in existing fault management approaches for WSNs. We must take into account a wide variety of sensor applications with diverse needs, different sources of faults, and with various network configurations. In addition, it is also important to consider other factors i.e. mobility, scalability and timeliness.

## 5. Conclusion and Future Work

The contribution of this paper is to present an in-depth critical overview of some of the most dominant centralized architecture based schemes for fault management in WSNs. fault management has been widely considered as a key part of today's network management. Recent rapid growth of interests in WSNs has further strengthened the importance of fault management, or in particular, played a crucial role. Faults in WSNs are not exception and tend to occur more frequently. In addition to typical network faults, wireless sensor networks have to deal with faults arising out of unreliable hardware, limited energy, connectivity interruption, environmental variation and so on. Thus, in order to guarantee the network quality of service and performance, it is essential for WSNs to be able to detect failures and to perform something akin to heal and recover the network from events that might cause faults or misbehaviour. A set of functions and applications designed specifically for this purpose is called a fault management platform [16, 29].

Centralized architecture based approaches are suitable for certain application. However, it poses various limitations such as these are not scalable and cannot be used for large-scale WSNs. In

ISSN 1512-1232
a nutshell, centralized fault management approaches suffer from many problems such as: insufficient scalability, availability and flexibility when a network becomes more distributed [12].

As a part of our on-going research, our future direction focuses on re-defining some of the evaluated dominant schemes into a more comprehensive effective fault management mechanism for WSNs.

**Acknowledgment**

The author acknowledges University of Malakand, Pakistan, and Higher Education Commission of Pakistan for funding towards his PhD project.

**References**

1. L.M.d.Souza, H.Vogt, and M.Beigl, "A survey on Fault Tolerance in Wireless Sensor Networks," *[Online]. Available:http//:digbib.ubka.uni-karlsruhe.de/volltexte/documents/11824,* 2007.
2. L. Paradis and Q. Han, "A Survey of Fault Management in Wireless Sensor Networks," *Journal of Network and System Management, Springer Science + Business Media, LLC,* vol. 15, pp. 171-190, June 2007.
3. Y.Mengjie, H.Mokhtar, and M.Merabti, "Fault Management in Wireless Sensor Networks," *IEEE Wireless Communications,* vol. 14, pp. 13-19, 2007.
4. H.Bakht, M.Merabti, and R.Askwith, "Mobile ad-hoc on-demand data delivery protocol," in *Proceedings of the 3rd Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, UK., 2002.
5. H.bakht, "Sensor network and ad-hoc networking," *Computing-Unplugged,* October 2001.
6. H.bakht, "Mobile ad-hoc on-demand data delivery protocol," *IEFT draft,* June 2001.
7. H. Bakht, M. Merabti, and B. Askwith, "A Study of Routing Protocols Mobile Ad Hoc Networks."
8. H.bakht, "Simulation Based Comparison of on--Demand Routing Protocols for Mobile Ad--hoc Network," *Annals. Computer Science Series,* 2011.
9. H.Liu, A.Nayak, and I.Stojmenovic, "Fault-Tolerant Algorithms/Protocols in Wireless Sensor Networks," in *Guide to Wireless Ad Hoc Networks*, ed: Springer-Verlag London, 2009, pp. 265-295.
10. B.Khelifa, H.Haffaf, M.Madjid, and D.Llewellyn-Jones, "Monitoring Connectivity in Wireless Sensor Networks," *International Journal of Future Generation Communication and Networking,* vol. 2, p. 10, June. 2009.
11. M.Yu, H.Mokhtar, and M.Merabti, "Self-Managed Fault Management in Wireless Sensor Networks," in *Proceedings of the The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '08)*, 2008, pp. 13-18.
12. M.Al-Kasassbeh and M.Adda, "Network fault detection with Wiener filter-based agent," *Journal of Network and Computer Applications,* vol. 32, pp. 824-833, 2009.
13. L.M.d.souza, H.Vogt, and M.Beigl, "A survey on Fault Tolerance in Wireless Sensor Networks," *www.digbib.ubka.uni-karlsruhe.de/volltexte/documents/11824.* , n.d.
14. I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communication Magazine,* pp. 102-114, 2002.
15. W.L.Lee, A. Datta, and R. Cardell-Oliver, *Network Management in Wireless Sensor Networks*: Handbook on Mobile Ad Hoc and Pervasive Communications American Scientific Publishers, 2006.
16. M.Yu, H.Mokhtar, and M.Merabti, "A survey on Fault Management in wireless sensor network," presented at the Proceedings of the 8th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK, 2007.
17. G.Tolle and D.Culler, "Design of an application-cooperative management system for wireless sensor networks," presented at the Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.

ISSN 1512-1232
18. N.Ramanathan, E. Kohler, L. Girod, and D. Estrin, "Sympathy: a debugging system for sensor networks [wireless networks]," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pp. 554-555.
19. B.Deb, S.Bhatnagar, and B.Nath, "Wireless Sensor Networks Management. *http://www.research.rutgers.edu/bdeb/ sensor networks.html*, 2005
20. S.Jessica, B.Dirk, and D.Glenn, "Efficient tracing of failed nodes in sensor networks," presented at the Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002.
21. N.Ramanathan, E. Kohler, L. Girod, and D. Estrin, "Sympathy: a debugging system for sensor networks [wireless networks]," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 554-555.
22. S.Jessica, B.Dirk, and D. Glenn, "Efficient tracing of failed nodes in sensor networks," presented at the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002.
23. A.Peffig, R.Szewczy, J.D.Tygar, Victorw, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," in *Proceedings of the ACM MobiCom' 01*, Rome, Italy, 2001, pp. 189-199.
24. L.B.Ruiz, I. G.Siqueira, L. B. Oliveira, H. C. Wong, J. M. S. Nogueira, and A. A. F. Loureiro, "Fault management in event-driven wireless sensor networks," presented at the MSWiM'04, Italy, 2004.
25. W.L.Lee, A.Datta, and R.Cardell-Oliver, "WinMS: Wireless Sensor Network-Management System, An Adaptive Policy-Based Management for Wireless Sensor Networks," School of Computer Science and Software Engineering, The University of Western Australia, Technical Report UWA-CSSE-06-01, 2006.
26. M.Z.Khan, M. Merabti, and B. Askwith, "Design Considerations for Fault Management in Wireless Sensor Networks," in *Proceedings of the 10th Annual PostGradute Symposium on The Conference of Convergence of Telecommunications, Networking and Broadcasting, PGNet 2009, Liverpool John Moores University*, Liverpool, UK, June 2009, pp. 3-9.
27. M.Z.Khan, M. Merabti, B. Askwith, and F. Bouhafs, "A Fault-Tolerant Network Management Architecture for Wireless Sensor Networks," presented at the 11th Annual PostGradute Symposium on The Convergence of Telecommunications, Networking and Broadcasting, PGNet 2010, Liverpool John Moores University, Liverpool, UK, June 2010.
28. M.Asim, H.Mokhtar, and M.Merabti, "Sensor Networks Management: A Survey," presented at the 11th Annual PostGradute Symposium on The Conference of Convergence of Telecommunications, Networking and Broadcasting (PGNet'10), , Livepool John Moores Univeristy, UK, 2010.
29. L.Paradis and Q. Han, "A Survey of Fault Management in Wireless Sensor Networks," *Journal of Network and Systems Management*, vol. 15, pp. 171-190, 2007.
30. M.Z.Khan, B. Askwith, F. Bouhafs, and M. Asim, "Limitations of Simulation Tools for Large-Scale Wireless Sensor Networks," presented at the Fifth International Workshop on Telecommunication Networking, Applications and Systems (TeNAS 2011), Workshops of International Conference on Advanced Information Networking and Applications (WAINA 2011), Biopolis, Singapore, March 2011.
31. F.Koushanfar, M. Potkonjak, and A. SangiovanniVincentelli, "Fault tolerance techniques in wireless ad-hoc sensor networks," UC Berkeley technical reports2002.

Article received: 2011-12-08