World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:2, 2015

Fault Tolerance in Wireless Sensor Networks – A Survey

B. R. Tapas Bapu, K. Thanigaivelu, A. Rajkumar

***Abstract*—**Wireless Sensor Networks (WSNs) have wide variety of applications and provide limitless future potentials. Nodes in WSNs are prone to failure due to energy depletion, hardware failure, communication link errors, malicious attacks, and so on. Therefore, fault tolerance is one of the critical issues in WSNs. We study how fault tolerance is addressed in different applications of WSNs. Fault tolerant routing is a critical task for sensor networks operating in dynamic environments. Many routing, power management, and data dissemination protocols have been specifically designed for WSNs where energy awareness is an essential design issue. The focus, however, has been given to the routing protocols which might differ depending on the application and network architecture.

***Keywords*—**Resiliency, Self-diagnosis, Smart Grid, TinyOS, WSANs.

I. Introduction

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation and traffic control. In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions are yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the size of the sensor network and the complexity required for individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station).

Tapas Bapu B.R. is with St.Peter's University, Chennai, India as a Research Scholar & with S.A. Engineering College, Chennai, India as an Associate Professor (phone: +91-9884179733; e-mail: tapasbapu@saec.ac.in).

Thanigaivelu K, Professor, and Rajkumar A, Post Graduate Scholar, are with S.A. Engineering College, Chennai, India (e-mail: drthanigaivelu@saec.ac.in, rajt905@gmail.com).

Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a processing unit with limited computational power and limited memory, sensors (including specific conditioning circuitry), a communication device (usually radio transceivers or alternatively optical) and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication devices (e.g. RS-232 or USB). The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user [1].

A. Advantages

A data aggregation process can enhance the robustness and accuracy of information obtained by an entire sensor network. Certain redundancy exists in the data collected from sensor nodes thus data fusion processing is needed to reduce the redundant information. Data aggression also reduces the traffic load and conserves energy of the sensors.

As far as the advantages of cooperative communications in WSN are concerned, their advantages are as follows:

1. Exploiting the WSN Architecture

i) WSN is deployed with hundreds or thousands of structural or randomly placed nodes making it ideal for cooperative communications. Selections of nodes in a cooperative group which are approximately at same distance from the intended receive nodes, results in equal energy consumption per bit in cooperative communications.
ii) By selecting closest nodes within a cooperative group, we can decrease energy consumption per bit in intra-cooperative node communications.
iii) In WSN the data flow direction is from sensor network to Base Station (BS) i.e., most of the time BS act as a receiver (neglecting small amount of signaling data from BS to network). By increasing the number of antennas at BS, we can take advantages of receive diversity at BS, because BS has no energy constraint.
iv) Usually the base station is located at some height in order to get reliable communication with the network, which results in a dominant LOS component. By exploiting this LOS communications we try to get maximum capacity in this backbone link [11].

As far as the selection of cooperative nodes in a network is concerned, the cooperative communications in WSN can be more beneficial if transmit cooperative nodes are at equal

distance from the intended receive nodes. This strategy evenly distributes the transmission energy in LOS component because in LOS component the power loss is inversely proportional to the square of the distance between the transmitter and receiver [3].

In addition to this the cooperative nodes should be at minimum distance from each other to save energy in local communication within the cooperative nodes group. These measures reduce the required bit energy for desired BER.

II. Applications

The applications for WSNs involve tracking, monitoring and controlling. WSNs are mainly utilized for habitat monitoring, object tracking, nuclear reactor control, fire detection and traffic monitoring. Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT) systems. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military C4ISRT [10].

Area monitoring is a common application of WSNs, in which the WSN is deployed over a region where some incident is to be monitored. For example, a large quantity of sensor nodes could be deployed over a battlefield to detect enemy intrusions instead of using landmines. When the sensors detect the event being monitored, the event needs to be reported to one of the base stations, which can than take some appropriate action. Wireless sensor networks are used extensively within the water/wastewater industries. Facilities not wired for power or data transmission can be monitored using industrial wireless I/O devices and sensor nodes powered by solar panels or battery packs.

Wireless sensor networks can use a range of sensors to detect the presence of vehicles for vehicles detection. Wireless sensor networks are also used to control the temperature and humidity levels inside commercial greenhouses.

A. Open Research Issues

The enabling applications provide some key attributes that determine the driving force behind WSN research. Existing applications such as environmental monitoring, health monitoring, industrial monitoring, and military tracking have application-specific characteristics and requirements. These application-specific characteristics and requirements coupled with today's technology lead to different hardware platforms and software development. A variety of hardware platforms and technology have been developed over the years; however, more experimental work is necessary to make these applications more reliable and robust in the real world. WSNs have the potential to enhance and change the way people interact with technology and the world. The direction of future WSNs lies in identifying real business and industry needs. Interactions between research and development are necessary to bridge the gap between existing technology and the development of business solutions. Applying sensor technology to industrial applications will improve business processes as well as open up more problems for researchers.

III. Fault Tolerance as a Factor Influencing Sensor Network Design

Some sensor nodes may fail or gets blocked due to lack of power, have physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures. The reliability $R_k(t)$ or fault tolerance given by (1) of a sensor node is modeled in using the Poisson distribution to capture the probability of not having a failure within the time interval (0; t):

$$R_k(t) = \exp(-\lambda_k t) \quad (1)$$

where λ_k and t are the failure rate of sensor node k and the time period respectively. Note that protocols and algorithms may be designed to address the level of fault tolerance required by the sensor networks. If the environment where the sensor nodes are deployed has little interference, then the protocols can be more relaxed. For example, if sensor nodes are being deployed in a house to keep track of humidity and temperature levels, the fault tolerance requirement may be low since this kind of sensor networks is not easily damaged or interfered by environmental noise. On the other hand, if sensor nodes are being deployed in a battlefield for surveillance and detection, then the fault tolerance has to be high because the sensed data are critical and sensor nodes can be destroyed by hostile actions. As a result, the fault tolerance level depends on the application of the sensor networks, and the schemes must be developed with this in mind [12].

Given a sensor network represented as a unit-disk graph, to compute and deploy the minimum number of additional devices to ensure that the resulting unit-disk graph satisfies the fault-tolerance constraint called vertex k-connectivity. A graph is vertex k-connected if there are at least k vertex-disjoint paths connecting every pair of vertices, or equivalently, the graph remains connected when any set of at most k-1 vertices is removed. The network has to be made resilient to k node failures [2].

IV. Fault Detection Techniques

The goal of fault detection is to verify that the services being provided are functioning properly, and in some cases to predict if they will continue to function properly in the near future. The simplest way to perform such a task is through visual observation and manual removal of incorrect values. This technique has obvious drawbacks: human interaction leads to errors, it has a high cost and it is not efficient. Hence, we investigated automatic fault detection techniques for WSN.

We classified the techniques we investigated according to the parties involved in the process. Through self diagnosis the node itself can identify faults in its components. With group

detection, several nodes monitor the behaviour of another node. Finally, in hierarchical detection the fault detection is performed using a detection tree where a hierarchy is defined for the identification of failed nodes. Often in a hierarchical detection the detection is shifted to a more powerful device such as the sink [1].

A. Self-Diagnosis

In many cases, nodes can identify possible failures by performing self-diagnosis. A self-diagnosis based on the measurements of accelerometers to determine if the node suffers from an impact that could lead to hardware malfunctions was considered.

Using a similar approach, nodes could detect when they are being moved to a different location. Another approach would be to keep track of the identities of the nodes in the neighbourhood. A considerable change in the neighbourhood could indicate that either the node itself or some of its previous neighbours have been moved.

Faults caused by battery exhaustion can be predicted when the hardware allows the measurement of the current battery voltage. By analyzing the battery discharge curve and the current discharge rate, an algorithm can determine an estimation of the time to death of the battery. Nodes can also identify that their current connection to surrounding nodes is unreliable by probing the link connection therefore identifying that it is isolated.

B. Group Detection

The detection of services failing due to incorrectly generated values is only possible if a reference value is available. Detection mechanisms are proposed to identify faulty sensor nodes. Both algorithms are based on the idea that sensors from the same region should have similar values unless a node is at the boundary of the event-region. The algorithm start by taking measurements of all neighbours of a node and uses the results to calculate the probability of the node being faulty.

Another approach proposed in the literature is to let consumer nodes observe whether the service provider is in fact performing the operations that it is supposed to. The misbehaviour detection mechanism is based on the idea of monitoring the communication of the service provider to verify whether messages are forwarded correctly.

Focusing on providing a fault-tolerant approach for clusters in WSNs, it is proposed to support the dynamic recovery of failed gateways (high-energy devices that act as cluster heads). The proposed protocol assumes that a gateway has failed only when no other gateways can communicate with it. The fault detection mechanism is based on constant status updates being exchanged between gateways and further use of a consensus algorithm.

C. Hierarchical Detection

The definition of a detection tree enables a scalable fault detection algorithm in WSN. It is proposed that the usage of the network topology is to forward the fault detection results of child nodes to the parent nodes and up to the sink. Each node forwards the status of the child nodes that it is monitoring to its parent node. The parent performs an aggregation operation on the results of the child nodes together with its results and forwards it to the next level. The approach proposed scales well with the network size; however it consumes resources of the network. Shifting the fault detection task to a more powerful device is an alternative that can help to increase the lifetime of the WSN. Some authors propose an algorithm that puts the burden of detecting and tracing failed nodes to the base station. At first the nodes learn the network topology and send their portion of the topology information to the base station. With this information the base station learns the complete network topology which is used to send route updates as soon as it detects that nodes become silent. This approach is not applicable to event-driven WSN because in such a network sensors only send messages when there is an event that should be reported, for instance when the temperature goes above a certain limit.

The proposed mechanism uses a hierarchical network topology where cluster heads monitor ordinary nodes, and the base station monitors the cluster heads. To perform the monitoring, the base station and the cluster heads constantly ping those nodes that still have battery power left and that are under their direct supervision. If a node does not respond, it is marked as failed. Sympathy is a debugging tool that also utilizes the hierarchical detection approach. SNIF is another example of a debugging tool that identifies the source of problems in WSN. Contrary to Sympathy, this tool does not modify the software of the sensor nodes nor requires additional traffic to be transported through the WSN. To automatically identify network failures this tool proposes a decision binary tree based on the research performed by the authors on failures in real world deployments [9].

V. Fault Recovery Techniques

Fault recovery techniques enable systems to continue operating according to their specifications even if faults of a certain type are present. There are many potential sources for faults in WSNs. Fault tolerance techniques have been proposed in various contexts that increase the reliability of the functionality of sensor nodes in their specific domain. We attempt at giving an overview of this scattered work. The most common of these techniques is the replication of components. Although redundancy has several advantages in terms of high reliability and availability, it also increases the costs of a deployment.

As an alternative, the quality required from the WSN can be downgraded to an acceptable level. Here we classify the recovery techniques for WSN into two major approaches: Active and Passive replication. Active replication means that all requests are processed by all replicas, while with passive replication, a request is processed by a single instance and only when this instance fails, another instance takes over.

A. Active Replication in WSN

Active replication in wireless sensor networks is naturally applied in scenarios where all or many nodes provide the same

functionality. One example is a service that periodically provides sensor data. Nodes that run this service activate their sensors and forward their readings to an aggregation service or to a base station. When some nodes fail to provide that information, the recipient still gets the results from other nodes, which is often sufficient. Fault recovery in the presence of active replicas is relatively straightforward. Nevertheless, for a consistent survey we present some of these approaches here:

1. **Multipath Routing:** Usually, it is desirable to avoid that a single failing node causes the partitioning of a sensor network. Thus, a network should be k-connected, which allows k−1 nodes to fail while the network would still be connected. Multipath routing can be used to actively replicate routing paths [8].
2. **Sensor Value Aggregation:** Sensor value fusion is a research area that seeks to provide high level information derived from a number of low-level sensor inputs. Here, the inherent redundancy of sensor nodes can be used to provide fault-tolerant data aggregation. This is achieved through a tradeoff between the precision (the length) of the resulting sensor reading interval and the number of faulty sensors. This ensures that despite of node failures, the resulting reading interval will contain the correct sensor reading of a region.
3. **Ignore Values from Faulty Nodes:** A simple but efficient solution to not propagate a failure of one specific node to the entire network is to ignore the data that it is generating, as applied in. The major challenge in this case is the identification of the malfunctioning nodes.

B. Passive Replication in WSN

When passive replication is applied, the primary replica receives all requests and processes them. In order to maintain consistency between replicas, the state of the primary replica and the request information are transferred to the backup replicas. Given the constraints of WSNs, applications should be designed to have only little or no state at all, which minimizes the overhead for transferring state information between nodes or eliminates it altogether. The process of recovering from a fault when using passive replication consists of three main steps: fault detection, primary selection and service distribution.

1. **Node Selection:** After it has been established that certain functionality is not available any longer due to a failure in the primary replica, a new service provider must be selected. After this selection phase, one or several nodes become service providers. Several approaches to how the selection is performed have been proposed. We differentiate them according to who makes the decision on which party should become a service provider. Self-organization techniques have proven to increase the reliability and fault-tolerance of distributed systems. In the extreme case, each node makes an individual decision (possibly taking information from its neighbours into account) or local group of nodes work together, coordinating their actions. On the other end, hierarchical systems assign tasks in a top-down manner. We discuss these options in turn.

i. Self Election: In LEACH, nodes periodically execute a probabilistic algorithm to establish whether they should serve as cluster head to their neighbours. In this probabilistic rotation system, nodes keep changing their role in the network. When a cluster head node fails, it will take only one rotation period until another node starts providing the functionality of the failed or absent node. Role assignment algorithms determine which of a certain role, such as coverage, clustering and in-network aggregation should be assumed by a node. A deterministic algorithm for autonomous role assignment is proposed that takes into account properties of the node such as battery status and location but also its neighbourhood and the roles chosen by neighbouring nodes. This facilitates the localized self-configuration of a sensor network and can re-establish service provisioning if executed after some nodes have failed.

ii. Group Election: A reallocation of nodes that were part of a cluster that suffered a cluster head failure is proposed. The cluster head, called gateway, is considered to be a resourceful node. The solution presented considers that all the gateways in the network maintain a list of the nodes that are currently in their cluster and another backup list of nodes that could become a part of their cluster. When a gateway fails, the nodes from its cluster are reallocated to other gateways that have the nodes in their backup lists. If more than one gateway has a specific node in its backup list the node is assigned to the cluster head that has the smallest communication cost.

iii. Hierarchical Election: In a hierarchical election, a coordinator selects the new primary node. This applies to the rebuilding of routing paths as well as the selection of a new cluster head. The former describes an algorithm to select the node that is closest to the base station. The latter approach applies fuzzy logic in the base station to select which node will become a cluster head. This algorithm makes use of a fuzzy descriptor, the node concentration, energy level in each node and its centrality with respect to the entire cluster. Although these centralized algorithms could perform a better selection of the nodes than a local algorithm due to its global view of the network, such approaches require that nodes send periodical messages to the base station.

2. **Service Distribution:** During this phase, nodes elected to become service providers must activate the service. In some cases the service is already available on the nodes and a simple configuration change to inform the node that this service should be activated is required. However in some cases, for instance when nodes do not have enough memory to store the code of all potential services, it is necessary to inject code into the node through some technique. There are different techniques that can be used for service distribution: completely reprogramming the node, sending entire blocks of executable code or sending small pieces of code such as scripts.

i. Pre-Copy: Pre-copying consists of the making code of all services available on all nodes before deployment. This

allows nodes to change their behaviour according to the role that they are assigned to.

ii. Code Distribution: Several approaches have been proposed for disseminating code throughout the network. Mat´e is an example for a byte code interpreter for TinyOS where code is broken into capsules of 24 instructions. These capsules can be distributed through the network and installed on nodes, which start to execute the new code. Agilla is a Mat´e-based mobile agent middleware for programming wireless sensor networks. These mobile agents can be programmed to move through the network or replicate themselves to other nodes according to changes in the environment. Impala is a middleware for sensor networks that supports software updates and on-the-fly application adaptation. Unlike Mat´e, the focus of Impala is networks that have a high degree of mobility, which can lead to long delays until an update is finished. While Mat´e stops the execution of an application until the update is finished, Impala processes ongoing software updates in parallel.

A hybrid approach between code migration and remote execution is proposed in, where the application code is copied to another node when the battery level reaches a first threshold. As soon as the battery reaches a critical level, the execution state is transferred and control is handed to the remote node. This allows for the full usage of the available energy resources, since control is handed over right before a node fails.

VI. WSANs in Smart Grids

Wireless sensor and actor networks (WSANs) are considered potential tools for monitoring and controlling smart grids. A WSAN is composed of a large number of low-cost, low-power, small, and multifunctional sensor and actor nodes. Sensor and actor nodes communicate wirelessly over short distances. Sensor nodes can collect various kinds of data, e.g., voltage, current, frequency, etc., while actors perform tasks such as closing/opening circuit breakers, turning on/off loads, etc. WSANs are preferred due to their ability to work in extreme environmental conditions, in addition to having enhanced fault tolerance, low power consumption, self-configuration, rapid deployment, and low cost. In environments where high voltages are in use, WSAN can also provide necessary insulation [7].

To carry out information exchange among fault nodes, the designed protocol for any application will allow the implementation of several fault-tolerance techniques. In this case, the central node would send a WHO message periodically, which would be replied by means of an ALIVE message. If the ALIVE message does not arrive, and after several tries, the supervision node will assume a failure in corresponding node [6].

When a node fail is detected, several actions can be carried out. The user will be informed anyway, but it is also possible to restart the system, or even to start a degraded working mode.

The node fall implies the impossibility of managing the inputs and outputs physically connected, but it does not impede the continuation of the reasoning based on the last well-known state of these variables, until the guardian node mechanism detects the failure and take the opportune measures [5].

VII. Discrepancy-Based Fault Detection and Correction

A cross-validation based technique for online detection of sensor faults is introduced. The approach can be applied to a broad set of fault models. They define a fault as an arbitrary type of inconsistent measurement by a sensor, which cannot be compensated systematically. In particular, they consider faults associated with the incorrect measurements that cannot be corrected using calibration techniques. The approach is based on two ideas: (i) compare the results of multi-sensor fusion with and without each of the sensors involved. (ii) use non-parametric statistical techniques to identify the measurements that are not correctable, regardless of the used mapping function between the measured and accepted values [4].

Sensor measurements are inevitably subject to errors. One can identify two types of errors: (i) random fluctuations in data due to a noise in a sensor or in a sensed phenomenon, or (ii) gross errors - faults. A practical method to distinguish a random noise is to run maximum likelihood or Bayesian approach on the multi sensor fusion measurements. A random noise would exist, if running these procedures improves the accuracy of final results of multi-sensor fusion. While there have been several efforts to minimize random errors, very little has been done for fault detection. In multi-sensor fusion, the measurements from different sensors are combined in a model for consistent mapping of the sensed phenomena. Although the new fault detection technique is generic and can be applied to an arbitrary system of sensors that use an arbitrary type of data fusion, they focus on equation-based sensor fusion for the sake of brevity and clarity.

Assume a set of sensors $s_i (0 \le i \le n)$, each measuring a value xi at a time t. The multimodal sensor fusion model equations are $f1,\ldots, fp$ are typically non-linear functions, and have the following forms: $fj(x1, x2, \ldots, xn)=0, (0\le j\le p)$. The system of equations is over-constraint. They solve the system n+1 times. First, they solve all the equations in the original format and then they ignore each variable and solve a least constrained system with n-1 variables (n times). They compare the values for each variable xn in all n+1 scenarios. In order to improve accuracy of fault detection, the system can be solved for m measurements by each sensor. At last, they conduct statistical analysis on the data for each sensor. If the obtained values for a sensor are not consistent within a confidence interval calculated by the percentile method, that sensor is considered faulty.

VIII. Conclusion

Fault tolerance is not only an availability feature but also a reliability feature. Due to the potential deployment in uncontrolled and harsh environments and due to the complex arch, wireless sensor networks are and will be prone to a variety of malfunctioning. Our goal was to identify the most important types of faults, techniques for their detection and diagnosis and to summarize the techniques for ensuring efficiency of fault resiliency mechanisms. All techniques and methods were briefly discussed in this paper for self-diagnosis of the sensors in various applications and situations.

References

[1] Luciana Souza, Harald Vogt, Michael Beigl, "A Survey on fault tolerance in Wireless Sensor Networks", Universitat Karlsruhe, 2007.
[2] Jonathan L. Bredin, Erik D. Demaine, Mohammad Taghi Hajiyaghayi, Daniel Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance", MobiHoc '05, May 25-27, 2005, Illinois USA.
[3] Farinaz Koushanfar, Miodrag Potkonjak, Alberto Sangiovanni-Vincentelli, "Fault tolerance in wireless sensor networks", Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, CRC Press, 2004.
[4] Hai Liu, Amiya Nayak, Ivan Stojmenovik, "Fault tolerant algorithms/protocols in wireless sensor networks", Springer-Verlag London Limited, 2009.
[5] Xin-Ming Huang, Jing Deng, Jing Ma, Zeyu Wu, "fault tolerance for wireless sensor grid networks", 2005.
[6] J.V. Capella, R. Ors, J.J. Serrano, "New challenges in wireless sensor networks: fault tolerance and real time", IEEE, 2005.
[7] Irfan Al-Anbagi, Melike Erol-Kantarchi, Hussein T.Mougtah, "Priority and Delay aware Medium access for Wireless Sensor Network in the Smart Grid", IEEE Systems Journal, 2013.
[8] Jamal N. Al-Karaki, Ahmed E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", ICUBE Initiative of lowa State University, May 2003.
[9] Erdal Cayirci, Chunming Rong, "Security in Wireless Ad hoc and Sensor Networks", John Wiley & Sons Ltd., 2009.
[10] Michael Grottke, Hairong Sun, Ricardo M. Fricks, Kishor S. Trivedi, "Ten fallacies of availability and reliability analysis", Springer-Verlag Berling Heidelberg, 2008.
[11] Holger Karl, Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", John Wiley & Sons, 2005.
[12] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey", Elseiver Science B.V., 2002.