IPP HURRAY!

www.hurray.isep.ipp.pt

# Technical Report

## Fault-Tolerance Mechanisms for Zigbee Wireless Sensor Networks

**Skender Ben Attia**

**André Cunha**

**Anis Koubâa**

**Mário Alves**

TR-070701

Version: 1.0

Date: July 2007

# Fault-Tolerance Mechanisms for Zigbee Wireless Sensor Networks

Skender Ben Attia, André Cunha, Anis Koubâa, Mário Alves

IPP-HURRAY!
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340509, Fax: +351.22.8340509
E-mail: {sbat@, arec@, akoubaa@dei., mjf@}isep.ipp.pt
http://www.hurray.isep.ipp.pt

## Abstract

The IEEE 802.15.4/Zigbee protocol stack has been considered as a promising technology for Wireless Sensor Networks (WSNs). Fault-tolerance is one of the main issues in WSNs since it becomes critical in real deployment environments where reliability and reduced inaccessibility times are important. However, the Zigbee protocol is currently lacking efficient faulttolerance mechanisms for supporting reliability for real-time applications. This paper analyzes common problems associated to ZigBee cluster-tree networks and proposes fault-tolerance mechanisms for those topologies. In this work in progress, we introduce two different fault-tolerance approaches, a reactive and a proactive mechanism and present some implementation guidelines for integrating these add-ons to Zigbee.

# Fault-Tolerance Mechanisms for Zigbee Wireless Sensor Networks

Skender Ben Attia[1], André Cunha[1], Anis Koubâa[1,2], Mário Alves[1]

[1] IPP-HURRAY! Research Group, Polytechnic Institute of Porto, Rua António Bernardino de Almeida, 431, 4200-072 Porto, Portugal

[2] Al-Imam Muhammad Ibn Saud University, Computer Science Dept., 11681 Riyadh, Saudi Arabia

sbat@isep.ipp.pt, arec@isep.ipp.pt, akoubaa@dei.isep.ipp.pt, mjf@isep.ipp.pt

**Abstract.** *The IEEE 802.15.4/Zigbee protocol stack has been considered as a promising technology for Wireless Sensor Networks (WSNs). Fault-tolerance is one of the main issues in WSNs since it becomes critical in real deployment environments where reliability and reduced inaccessibility times are important. However, the Zigbee protocol is currently lacking efficient fault-tolerance mechanisms for supporting reliability for real-time applications. This paper analyzes common problems associated to ZigBee cluster-tree networks and proposes fault-tolerance mechanisms for those topologies. In this work in progress, we introduce two different fault-tolerance approaches, a reactive and a proactive mechanism and present some implementation guidelines for integrating these add-ons to Zigbee.*

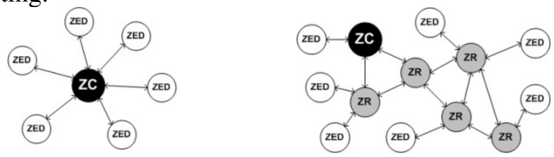
## 1. Introduction

Wireless Sensor Networks are inherently unpredictable and are very prone to failures. These failures can create blind spots in the network by isolating some of the devices or can introduce large inaccessibility times in communications that can lead to abnormal behaviours of the applications. Furthermore, in case of large scale WSNs, these failures can lead to the collapse of the entire network. The current specifications of the IEEE 802.15.4/Zigbee protocol stack, which has been considered as a promising technology for WSNs, is lacking efficient fault-tolerance mechanisms. This paper proposes fault-tolerance mechanisms for IEEE 802.15.4/ZigBee cluster-tree networks since they are sensitive to the single points of failure problem in Zigbee Routers (ZR). ZigBee lacks fault-tolerance mechanisms for ZR failures, since the basic orphan realignment mechanism is not sufficient as it imposes large inaccessibility times that may not be acceptable for certain time critical applications. In this paper, we propose two fault-tolerance mechanisms based on *reactive* and *proactive* approaches. The former improves on the default ZigBee protocol behaviour by shortening the time to associate to a new parent in reaction to a ZR failure. The latter goes beyond that, by proactively finding a new parent upon detection that the current parent has degraded quality under a certain threshold. Both mechanisms use a quality indicator (PAI - Parent Adoption Indicator) to choose a new parent, as described in section 3.1. These mechanisms are backward compatible with the IEEE 802.15.4/ZigBee standards.

## 2. IEEE 802.15.4/ZigBee relevant aspects

### 2.1. Device types and network topologies

In ZigBee networks, there are three types of devices: (1) *ZigBee Coordinator* (ZC): one for each PAN, initiates and configures the network formation; (2) *ZigBee Router* (ZR): associated with the ZC or with a previously associated ZR that participates in multi-hop message routing; (3) *ZigBee End Device* (ZED): a simple device that has sensing capabilities and does not allow other devices to associate with it and does not participate in routing.



a) star topology b) mesh topology

**Figure 1**. IEEE 802.15.4/ZigBee network topologies

The IEEE 802.15.4/ZigBee standard enables three network topologies – star, mesh and cluster-tree. In the star topology (Fig. 1a) the communication paradigm is centralized i.e. communications are always relayed through the ZC. The mesh topology (Fig. 1b) the communication paradigm is decentralized i.e. each node can directly communicate with any other node within its radio range or through multi-hop. The cluster-tree network topology (Fig. 2) is a special case of a mesh network where there is a single routing path between any pair of nodes and a distributed synchronization mechanism (beacon-enabled mode).

### 2.2 Cluster-Tree Network Model

In this paper, we consider the case of a ZigBee cluster tree topology as the one exemplified in Fig. 2. One ZC identifies the entire network and each ZR assumes the role of cluster-head allowing the association of other ZRs and ZEDs in a *parent-child* relationship. There can be multiple clusters in a network, as depicted in Fig. 2. When the association process is successful, we say that the child device has joined (or associated) the network through its parent (ZR). Inside a cluster, the communication is established via the cluster-head i.e. direct communication between two children in the same cluster is not possible.

IEEE 802.15.4/Zigbee supports a native fault-tolerance mechanism denominated as the *orphaned device realignment*. This recovery/repair procedure occurs when there are repeated communication failures in the requests for data transmissions (e.g. data frames sent without receiving the requested acknowledgment) between the device and its parent or when the device loses synchronization with its parent. The MAC layer defines the constant *aMaxLostBeacons* to specify the maximum allowed beacon frame losses and the *aMaxFrameRetries* as the maximum number of retries after a transmission failure.

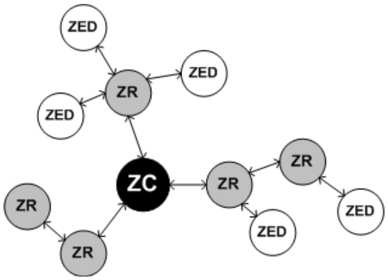


**Figure 2.** The cluster-tree topology model

The MAC sublayer can have two different behaviours upon the conclusion that the device is orphaned. It can either perform the orphaned realignment procedure or reset the MAC sublayer leading to a new association procedure to the network. The orphan realignment procedure relies on two command frames, the *orphan notification frame*, which is broadcast by the orphan device including its extended address, and the *coordinator realignment frame* sent in response by the parent containing the information about the device (e.g. short address allocated) and about the network. The orphan association starts with an orphan scan procedure where the orphan device performs a physical channel scan on all available (or pre-defined) radio channels and sends orphan notifications, as depicted in Fig 3.

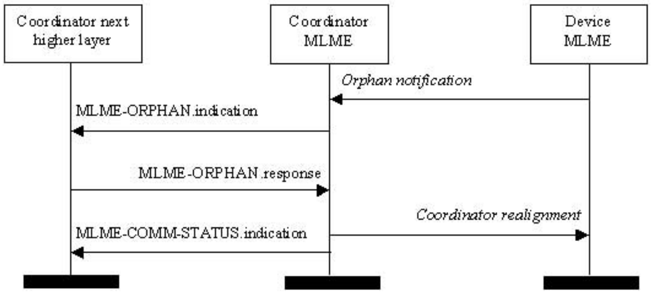


**Figure 3.** Message sequence chart for orphan notification[1]

If the parent device receives the orphan notification command, it will reply with a coordinator realignment frame, after a search in its neighbour table (this table contains information of the neighbours including the associated devices) verifying if the command was sent by one of its child devices. The orphan device stops the channel scan procedure upon reception of the realignment frame and then updates its PAN information. If the orphan device completes the channel scan without finding its parent, it must start a new association to the network. In the association mechanism, the device performs a channel scan searching for a suitable parent. After the synchronization with the new parent, the device starts the association procedures. During the time to scan the channels, synchronize with the chosen parent and associate with it, the device cannot transmit nor receive any messages.

## 3. Fault-Tolerance Mechanisms for ZigBee Cluster-Tree Networks

There are two reasons for a child and its parent to loose connectivity: (1) Wireless link problems induced by electro magnetic interference (EMI), the presence of obstacles between the nodes or to node's mobility; (2) Device failure namely hardware, battery, software (or other) problems that prevent it from performing normally. In this paper, we present two alternatives to mitigate the impact of the aforementioned problems with the purpose of reducing the inaccessibility times that may be experienced by orphan nodes.

### 3.1. The Parent Adoption Indicator (PAI)

In order to be able to choose a new parent, it is mandatory to assess the adoption potential of the set of available new parents. Thus, we have defined the PAI indicator that depends on several important metrics, namely the Link Quality Indicator (*LQI*), the depth of the candidate parent (*Dp*) in the tree, traffic load (*Tl*) and the energy indicator (*Ei*). The Parent Adoption Quality Indicator (PAI) is expressed as:

$$PAI = LQI \cdot (a \cdot Ei) \cdot \left(\frac{b}{Dp}\right) \cdot \left(\frac{c}{Tl}\right) \qquad \textbf{(1)}$$

In (1), coefficients *a*, *b* and *c* are integer weighting factors that can take different values depending on the importance given to each quality parameter. Since higher values of *Dp* and *Tl* parameters indicate less potential of the candidate parent, we choose to compute their inverse to reflect the degradation they introduce. The most suitable parent will have the highest PAI value.

### 3.2. Reactive Re-association Mechanism

The reactive re-association mechanism is triggered when a device looses connectivity with its parent. This approach can be considered as an extension to the IEEE 802.15.4 orphan realignment mechanism introducing a new functionality in the orphan scan procedure.

The reactive re-association mechanism (Fig. 4) begins with the occurrence of a fault event (e.g. wireless link or parent hardware failures). At a first step (Fig. 4A), the device initiates the orphan scan procedure in order to search for his parent and simultaneously searches for other possible parents. The scan starts in the currently used channel and goes through all the logical channels defined for the network. In each logical channel, an orphan notification is sent and the device listens to the wireless medium (the channel assessment duration is equal). While waiting for a realignment command, it also listens for other possible parent devices. At a second step (Fig 4B), if the node receives its parent realignment command, the orphan device re-associates and resumes normal operation.

If the previous parent is not found and the child device discovers possible parents, it calculates the PAI indicator expressed in Eq. (1) for each candidate parent and stores the results in the neighbour table. If all the channels have been scanned and the association to the previous parent was not possible, the child device initiates a re-association procedure with the best possible parent with regard to the PAI indicator results. If no suitable parent is found, the procedure ends with an error indicating that the device is not in range with any possible parent. In this case the orphan device can eventually perform point-to-point data transfers with another device in order to establish a communication flow.

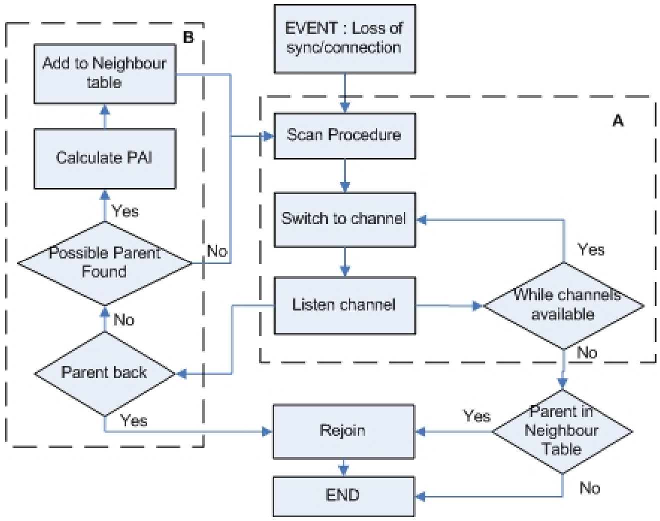


**Figure 4.** The reactive re-association mechanism

This approach introduces two major contributions to the standard. Firstly, it creates a real fault-tolerance mechanism since the ZigBee orphaned device realignment only allows the association of an orphan device to its parent. In our approach, however, if the parent fails, the child device does not need to perform a second channel scan to find a new suitable parent because it already collected that information during the scan for its parent. This will reduce the inaccessibility times and allow a quicker re-association. Secondly, the usage of a Parent Adoption Indicator (PAI) presented in Eq. (1), encompassing several important metrics for measuring the quality of candidate parent in a more efficient manner.

## 3.2. Proactive Re-association Mechanism

A proactive re-association approach proposes a preventive change of the parent in order to avoid the loss or extreme degradation of the current parent-child link. Fig. 5 depicts the proactive re-association mechanism procedure.

The proactive approach must guarantee that certain conditions are verified before switching to a new parent. The estimation of the parent link degradation must be as accurate as possible. Using weak switching conditions would cause children to change parents too often without real need, thus inducing more frequent inaccessibility periods and increased energy consumption.

During normal operation, a node samples one out of every N messages in order to assess the quality of its current parent using the PAI formula. If the PAI is under the minimum quality threshold (S) (Fig. 5A), the child will trigger the confirmation phase (Fig. 5B) during which it will process the PAI for a given number of consecutive received packets. If all packets are below S, the mechanism is sure that the parent is degrading; otherwise it goes back to the normal sampling phase. If the child detects a parent failure, it scans its current channel (Fig. 5C) during its superframe inactive period looking for alternative parent devices within its own PAN. If candidate parents are found within its PAN, the device calculates their PAI.

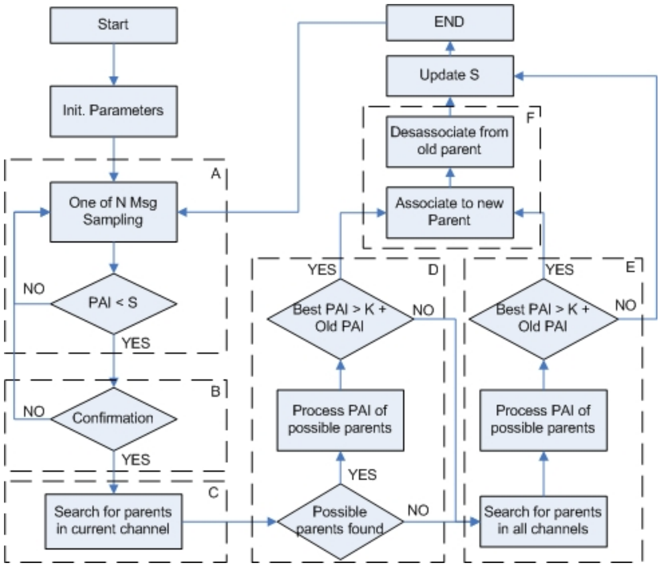


**Figure 5.** The proactive re-association mechanism

The candidate parent with the best PAI must also fulfil the condition (Fig. 5D) in Eq. (2).

$$\text{Best PAI} > \text{old PAI} + K \qquad \textbf{(2)}$$

*where K is the expected quality improvement gained by changing parents*

If such a parent device is found, the device will associate to it and disassociate from its former parent. In this case there is no inaccessibility time since the device is always connected to the PAN (Fig. 5F). If there is no device fulfilling the condition in Eq. (2), the child device will launch a channel scan on all the available pre-defined channels searching for possible parents during the inactive period of its superframe (Fig. 5E). After calculating the PAI of the different devices found during the scan, the device searches for a candidate parent that fulfils the condition in Eq. (2). If such ZR is found the device will associate to it and break the link with its old parent (Fig. 5F), otherwise it will stay connected to its current parent. The last step of the mechanism is the dynamic adaptation of the S threshold. If activated, the child device updates the value of the S threshold using Eq. (3).

$$S = \text{new PAI} - Tu \qquad \textbf{(3)}$$

*where Tu is the deterioration factor expressed in %*

The proactive re-association approach introduces interesting advantages that improves energy balancing by ordering the candidate parents based on energy information, traffic load, number of associated nodes and link quality information. Ultimately, the proactive re-association mechanism leads to an establishment of connections that offer the best transmission conditions between all the nodes of the network. However, this mechanism will not eliminate blind spots in the networks when there is a complete failure of the parent node and there are no alternative parents in the vicinity.

### 3.3. Implementation guidelines

We are currently working on the implementation of the proposed reactive and proactive fault-tolerance mechanisms and their integration as a module in Open-ZB [3], an open-source implementation of the IEEE 802.15.4/ZigBee standard protocol stack for TinyOS [4].

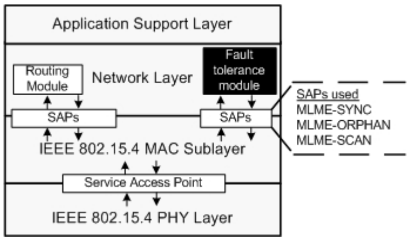


**Figure 6.** Implementation software architecture

Fig. 6 depicts the implementation architecture of the current Open-ZB stack and the localization of the fault-tolerance module. The implementation of the reactive re-association approach will also introduce minor changes in the orphan scan procedure assuring total backward compatibility, thus enabling the coexistence of both devices that do or not implement these add-ons.

## 4. Fault-Tolerance – Related Work

There are several reasons for a communication link or a device/node to fail. Fault-tolerance mechanisms tackle these abnormal situations. Generally there is a trade-off between the reliability improvement obtained by a fault-tolerance mechanism and the performance of the network.

Several works have assessed the problem of fault tolerance in classic wired networks and some proposals became popular such as the Spanning Tree Protocol [5] or IP Multicast [6].

There are several research works that analyze and propose fault-tolerance mechanisms for wireless sensor networks. A first type of faults is related to software problems (e.g. bugs in the embedded programs) in the nodes, that prevent them from functioning correctly. In Reference [7], the authors have analyzed and presented a mechanism to correct them. A second type of errors is the erroneous estimation of the sensed parameter (e.g. the node seems to work properly, but the values returned by sensor are incorrect). Reference [8] has examined this behavior and proposed a solution based on the computation of a correlation value between the different sensed values. A third type of faults are hardware faults where the node seized to neither receive nor send any information and is totally disconnected from the global sensor network (e.g. battery depletion, hardware deterioration, transmitter failure, etc.). Fault recovery procedures should allow the isolation of the faulty node and the restructuring of the network. In reference [9], the authors have proposed a re-clustering mechanism based on redundant information and link state status of every router. LEACH [10] is a re-clustering protocol that can be considered to be fault-tolerant, since it distributes the failure probabilities between all routers in the network, even though it was not designed for that specific purpose.

## 5. Conclusions and ongoing work

In this paper, we have presented two mechanisms for handling router degradation or failure in Zigbee cluster-tree Wireless Sensor Networks: a proactive approach and a reactive approach. In the reactive mechanism, which can be considered as an enhanced version of the orphaned device realignment, the device only needs to perform one scan procedure to realign itself with its parent or to associate to a new parent. The *proactive approach* has the advantage to avoid the device re-association procedures by planning in advance its re-association to a more reliable parent. These two approaches enable a faster re-association of the orphaned devices to the network, thus reducing or even eliminating inaccessibility times and improving reliability in ZigBee cluster-tree networks. Currently, we are working towards the implementation of both fault-tolerance approaches in TinyOS. We are also working to analytically evaluate the inaccessibility times and the network performance using our proposed mechanisms and to compare them with the standard mechanism of ZigBee.

## 5. Acknowledgement

This work was partially funded by FCT ("Fundação para a Ciência e Tecnologia") under the CISTER Research Unit (FCT UI 608), by the PLURALITY project (CONCREEQ/900/2001), and by the ARTIST2 NoE.

## References

[1] IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE standard for Information Technology, 2003.
[2] Zigbee-Alliance, "ZigBee specification," http://www.zigbee.org/, Dec 2006.
[3] Open-ZB, "Open-source toolset for IEEE 802.15.4 and ZigBee", www.open-zb.net
[4] TinyOS, www.tinyos.net
[5] Cisco Systems, "Understanding Spanning Tree Protocol".
[6] Samuel Tardieu, Laurent Pautet, "Building Fault Tolerant Distributed systems using IP multicast", Ecole Nationale Supérieure des Télécommunications de Paris
[7] Veljko Kurnic, Eric Trumpler, Richard Han, "NodeMD: Diagnosing Node-Level Faults in Remote Wireless Systems", University of Colorado (2006)
[8] Thomas Clouqueur, Parameswaran Ramanathan, Kewal K. Saluja , Kuang-Ching Wang, "Value Fusion Vs Decision Fusion for Fault-tolerance in collaborative Target Detection in Sensor Networks", University of Wisconsin-Madison
[9] Gaurav Gupta, Mohamed Younis, "Fault Tolerant Clustering in Wireless Sensor Networks", University of Maryland
[10] W. R. Heinzelman, A. Chandrakasan, and H.Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in Proc. of the Hawaii International Conference on Systems Sciences, Jan. 2000.