Failures from the Environment, a Report on the First FAILSAFE workshop

Michael Breza
AESE, Imperial College, London
mjb04@doc.ic.ac.uk

Ivana Tomic
AESE, Imperial College, London
i.tomic@imperial.ac.uk

Julie McCann
AESE, Imperial College, London
jamm@doc.ic.ac.uk

ABSTRACT

This document presents the views expressed in the submissions and discussions at the FAILSAFE workshop about the common problems that plague embedded sensor system deployments in the wild. We present analysis gathered from the submissions and the panel session of the FAILSAFE 2017 workshop held at the SenSys 2017 conference. The FAILSAFE call for papers specifically asked for descriptions of wireless sensor network (WSN) deployments and their problems and failures. The submissions, the questions raised at the presentations, and the panel discussion give us a sufficient body of work to review, and draw conclusions regarding the effect that the environment has as the most common cause of embedded sensor system failures.

CCS CONCEPTS

• **Networks** → **Routing protocols**; • **Security and privacy** → **Security protocols**;

KEYWORDS

Taxonomy, Adversarial Models

1 INTRODUCTION

The most exciting aspect of embedded sensing systems is that they take state of the art MEMS sensors out of the lab into real environments. The most difficult aspect of embedded sensing systems is that they take fragile microelectronic components out of the lab into real environments. The lab is a comfortable, controlled environment. Real environments beyond the lab are full of chance and uncertainty. It is our inability to engineer for all of the possible ramifications of the environment on our embedded sensing systems that make deployments unreliable and error prone.

This document presents the views expressed in the submissions and discussions at the FAILSAFE workshop about the common problems that plague embedded sensor system deployments in the wild. This analysis is gathered from the submissions and the panel session of the FAILSAFE 2017 workshop held at SenSys 2017 conference. The FAILSAFE call for papers specifically asked for descriptions of wireless sensor network (WSN) deployments and their problems and failures. The submissions, the questions raised at the presentations, and the panel discussion together provide a substantial contribution to the body of WSN deployment reliability literature. We feel that the addition of this information enables us to draw conclusions connecting environmental effects like weather and radio interference to the most common cause of embedded sensor system failures.

We will first present our initial goals for the workshop, then discuss the received submissions, which will be followed by an identification of the major themes arising from the submissions. Then, we will introduce the scope of the panel session that was held as a part of the workshop, as well as the main outcomes of the session. Finally, we will use both, findings from the submissions and findings from the panel session, to point towards important areas of research aimed at making WSN reliable and usable.

ACKNOWLEDGMENTS

This paper reports the outcomes from the first *FAILSAFE* workshop held at SenSyS 2017 on the campus of TU Delft in Holland. We would like to thank all of our reviewers, authors, panellists, and attendees for their contributions and helping to make the workshop interesting and enjoyable.

2 FAILSAFE CALL FOR PAPERS

There is a pre-existing body of literature on failures in WSN that was used as the inspiration for FAILSAFE. The literature is roughly divided into taxonomy papers, and lessons learned type papers. Taxonomy papers [1, 6, 10, 21, 25–27] discuss an abstract notion of WSN failures, and create taxonomies of the causes of WSN failure. Lessons learned papers [2, 7, 11, 16, 18, 19, 22, 30] document the failures encountered, and the lessons learned from actual WSN deployments in the wild (not in a lab or WSN testbed). We felt that there were not enough lessons learned papers which documented actual failures during WSN deployments. The addition of more lessons learned papers would increase the body of knowledge of real failures in the wild, and give more information for the study

of causes of failures in WSNs, and how to engineer systems to be robust to these failures. To this end, we organised the FAILSAFE workshop.

Unique to this workshop, we asked for submissions reporting the failure of actual embedded sensing system deployments complete with data. We wanted the failures, problems and glitches that wasted hours of researcher time; head scratching whilst trying to diagnose and fix subtle or unexpected bugs etc. This workshop focuses on the point at which the simulations and tests done in the lab are taken out into the field, and the previous assumptions fail.

The workshop scope included:

- Real WSN/IOT etc. deployments that either experienced some, or ended in, failure, complete with data to show what failed and a detailed analysis of the causes of the failure and the lessons learned.
- Examples of a WSN or other sensor deployments that where either hacked or attacked in a nefarious way, with as much data and details as can be provided.
- Any practices or tools, along with methodology and description, that have evolved out of sensor system deployment failures that are used to prevent further problems and that can be shown to improve the success rate of Sensor System deployments.

2.1 The Received Submissions

We received 9 submissions that ranged from mostly deployment reports, with an end section discussing some encountered problems, to a mix of a deployment description, a list of the problems encountered, and a list of lessons learned, to in-depth discussions around the diagnosis of a problem that originated in a deployment.

There were three general themes that arose from the contributions that we received. The first concerned the unreliability of networking, and the importance of knowing the network environment.

Paper 1. M. Bezunartea, B. Sartori, J. Tiberghien and K. Steenhaut. *Tackling malfunctions caused by Radio Duty Cycling protocols that do not appear in simulation studies.* [3] - The paper reports the experiences from testing a multi-hop WSN for environmental control and security in buildings and the differences observed between the behaviour in simulations and in the real world. The results showed that the main source of issues was the use of radio duty cycling protocol. This observation led to an in-depth experimental study of the different asynchronous RDC protocols available in Contiki to improve the overall reliability. This paper demonstrates an interesting experimental approach to debug problems with the radio environment.

Paper 2. R. Marfievici, P. Corbalan, D. Rojas, A. McGibney, S. Rea and D. Pesch. *Tales from the C130 horror room: A wireless sensor network story in a data center.* [23] - The paper reports a 17-month long deployment of 30 wireless sensor nodes in a small data center room. The most difficult challenge reported was how to understand the impact of the environment on the network and application which emphasized the necessity for a WSN design and deployment methodology and the need for dependable protocols. This paper is another example of an investigation into a problematic radio environment.

Paper 3. L. Tian, S. Santi, L. Steven and J. Famaey. *Accurate sensor traffic estimation for station grouping in highly dense IEEE 802.11ah networks.* [31] - The paper presents a novel traffic estimation method for RAW optimization in highly dense IEEE 802.11ah networks. This improved traffic interval estimation allows more accurate selection of station grouping parameters based on real-time dynamic traffic conditions which in turn results in highly improved throughput and latency in dense networks with a difficult radio environment.

Paper 4. I. M. Runge and R. Kolla. *MCGC: A network coding approach for reliable large-scale wireless networks.* [28] - The paper presents the use of Network Coding based Multicast Growth Codes (MCGC) for reliable data maintenance in large-scale WSNs with challenging radio environments with high fault rates. MCGC are able to tolerate high fault rates and reconstruct data in a shorter period of time than other similar approaches. This addresses radio environments with high fault probabilities.

Paper 5. S. Little, D. Zhang, C. Ballas, N. E. O'Connor, D. Prendergast, K. Nolan, B. Quinn, N. Moran, M. Myers, C. Dillon and T. Meehan. - *Understanding packet loss for sound monitoring in a smart stadium IoT testbed.* [20] - The paper presents the practical evaluation of an end-to-end testbed for IoT innovation at Croke Park Stadium in Dublin. The Ḧow loud is the 16th player?s̈tudy reported significant packet loss from sensor to cloud that occurred due to an extremely crowded radio environment. The test environment was a crowded stadium with roughly 70,000 cell phone users accessing the WiFi and the cellular network at the same time. This paper showed clearly how the radio environment caused the sensor network to fail.

These five papers looked at the importance of knowing the radio environment, and how the radio environment can affect the reliability of a WSN deployment.

The next two papers highlighted problems caused by the actual environment, rain, foliage, and high temperature.

Paper 6. R. Hartung, U. Kulau, B. Gernert, S. Rottmann and L. Wolf. *On the Experiences with Testbeds and Applications in Precision Farming.* [15] - In this paper the authors present experiences and findings from a testbed and two WSN deployments on an agricultural area which were used to measure the stress of potato crops. Throughout both of the deployments a number of problems and failures due to

environmental factors, farming activities, use of third-party components etc. was reported. This paper had a very good lessons learned section, and was a good example of a paper recording very practical problems related to the overall complexity of an embedded sensor deployment. The paper also had a good list of the many possible environmental affects that can cause failures.

Paper 7. G. Jackson, S. Gallacher, D. Wilson and J. McCann. *Tales from the wild: Lessons learned from creating a living lab.* [17] - The paper explored multiple case studies undertaken in the London Living Lab, namely air quality, microclimate and urban bat monitoring. This paper provided an interesting lessons learned section which included advice from providing adequate time to plan, debugging as you would in the wild, all the way to protecting sensor systems from unexpected events such as a squirrel attacks. The affects of the environment once again played a major role in WSN failures in this paper. There was a recorded sensor failure due to a storm felling the tree that a sensor was deployed in.

The final papers address the interesting issue of data failures in WSN, and methods for data robustness. This exposes the problem that the nodes may be working correctly, but the data that is collected is not fit for purpose.

Paper 8. X. Fang and I. Bate. *Issues of using wireless sensor network to monitor urban air quality.* [14] - The paper reports the lessons learned and issues identified from three deployments in an uncontrolled environment for monitoring urban air quality. These mostly relate to the practical limitations in locating the nodes, the issues arising from data acquisition and data processing. This paper was interesting as it was the only paper that dealt with failures from a purely data oriented point of view, and was an example of a deployment where the nodes and radio links did not fail, but the data was of low quality due to the difficulty of sensor calibration.

Paper 9. M. Cvjetkovic and V. Rakocevic. *Relative localisation algorithm for neighbour classification in Ad Hoc networks of moving robots.* [9] - The paper presents a solution for mitigating mobility induced challenges in networks of moving robots operating in challenging environments. The robots use radio signal strength measurements to coordinate their position and to communicate with each other. Noisy radio environments cause the signal strength measurements to be inaccurate, and cause problems with communication. An algorithm is presented that allows a robot to track the changes in the location of their neighbours and increases the efficiency of data transfer by improving the selection process of which neighbour to communicate with, and use for location.

The papers that were selected for the FAILSAFE workshop illustrated three rough areas of potential failures for WSN. Failures to the hardware or sensors caused by the physical environment, communication failures caused by the physical, or radio environment, and data failures caused by physical affects to the sensors, or difficulty getting the required data precision.

2.2 Common Failures and Questions Raised

We saw three major types of failures in all of the papers that were submitted to FAILSAFE which are:

(1) Hardware failures - caused by physical damage to the hardware resulting in hardware malfunction or complete failure. These failures prevent a node from performing as designed and required.
(2) Radio failures - caused either by physical damage to the radio transceiver, or by other radio interference, such as other transmitters causing radio interference, or a physical object causing an obstruction of the radio signal. In all cases this failure prevents a node from communicating.
(3) Data failures - caused by physical damage to a sensor or other part of the node, or caused by environmental conditions that affect the correct reading of the sensor. Can also be a simple sensor calibration issue, or a failure of the systems designers to properly understand the phenomenon being measured, and so collect measurements in a way that is difficult to process.

These failures can be caused by the environment. In some cases these failures overlapped, for instance when the leaves cover the antenna of a sensor node and caused a communication failure.

This analysis of the failures caused by the environment is similar to general computer systems dependability analysis in that they both have similar goals, usable data for successful applications. The difference here is that general computer systems assume dependable environments, and so focus on mean times to failure of individual hardware components and software coding standards. The components are tested by the manufacturers which produce mean times to failure that are fairly reliable. Software coding and testing is then done to assure that the software behaves as expected for a range of inputs and uses. The difference that we have with embedded systems is that the environment is not controlled, and so can cause failures more frequently that the manufacturers component lifetime estimates.

3 THE PANEL SESSION

3.1 Setup and Attendance

The accepted papers widely recognised the effect that environment has on WSN deployments. We also wanted to address what can be done to mitigate failures, and how a clear, scientific methodology for the design, development, and deployment of WSN systems could be defined.

We invited Alberto Boano, Michael Fisher, Josiah Hester, and Ramona Marfievici, four experts from WSN community, to comment onto the issues that were highlighted during the day and set up the stage for what has to be done next.

Both the paper sessions and the panel were well attended with 22-26 attendees throughout the whole day.

3.2 Points of Discussion

The main point that was raised during the discussion related to the path or workflow that each deployment should go through in order to ensure success. The current standard workflow seems to be: develop on a simulator, then test on a WSN testbed, finally do a real deployment. This workflow seems to be lacking as evident from the persistence of deployment failures. The main points of the discussion are as follows:

- The models used should suit the purpose and be robust enough. A good example are the models of failures and robustness used more generally in engineering. This includes Robustness Block Diagrams and Failure Trees [4]. Another aspect that needs to be included into WSN simulation are environmental affects including those that affect radio communication and those that affect hardware.
- There is a need for improving the current simulators (for example Cooja) by introducing more realistic simulations parameters to test the system thoroughly before moving onto real hardware. At the same time, going down this path might not be of interest for the academic community as simulation-based papers are usually "not good enough" for being published in top conferences/journals.
- Testbeds should have pre-defined test vectors/benchmark for the evaluation of proposed solutions. A good example would be Dependability competition (EWSN) [5] where a set of parameters and classes of applications are pre-defined and solutions are build around these. Such approach could enable the continuous integration of testing methodology, but there is a danger of not being able to cover all different classes of IoT applications.
- A choice of environment might be crucial for the success of deployment. Unless there is a need for an extreme environment, this should be chosen to be easily accessible and easily deployable. Also, the fact that there might be more than one network operating in the same environment should be taken into account and an adequate orchestration of these is necessary.
- Finally, most of the problems are usually known. How do we avoid making same mistakes again? Is there a need for a way to organise and share existing issues and knowledge in embedded sensor deployments? This could take the form of an embedded sensor system design and deployment methodology similar to the current software methodologies.

The panel session was concluded with an open question of what should be done next. Is there a need for running FAILSAFE with the same scope again or shall we call for the success stories which success was built on the "lessons learned" that were reported in the literature? One suggestion that we received was to organise a Dagstual.

4 DISCUSSION

The environment is the single most common cause of failure. This fact was re-iterated in the background literature, the submissions, discussions during FAILSAFE presentations, and the panel discussion. The environment poses a class of failures that are both unbounded in type (anything can happen) and difficult to predict (if anything can happen, how do we know when it will happen?). Other forms of failures, such as programming or packaging, need to be informed by this fact, and take it into account.

The question is, how do we deal with unbounded causes of failure? The first approach is to know the physical and radio environment as well as possible. This includes seasonal variations in temperature, rainfall and vegetation, as well as the chance of intervention by animals or people. A complementary approach that comes from the literature and discussions is to focus on the affects, not the causes, of failures. This reduces the problem to, failures at the node level, failures in the communication between nodes, and the resultant failures caused by not providing good data to the application, and the subsequent failure of the application.

There are three related directions that we can pursue. The first is to create improved failure models to be able to reason about what part of our systems need to be redundant, and understand the affects of failures on our application goals. Related to models, but different in scope, is the need to be able to know what guarantees we can make about our software systems that run on these embedded sensor systems. This includes a focus on the parts of the system that have been identified by our failure model and can not be guaranteed by static analysis at development time. That which can not be guaranteed needs to be monitored for upstream notification or local adaptation at runtime. Finally, we need some sort of design and deployment methodology similar to software development methodologies, so that past deployment experience can be re-used, and prevent future deployment engineers from 're-inventing the wheel'.

We need improved models and related simulations that help us to inject the results of environmental failure into an application. This will help us to understand its affects, and

find any ways to improve its robustness. Interesting models from the reliability engineering community exist, such as Reliability Block Diagrams and Failure Tree Models [12]. Neither of these models map well into the software side of embedded sensor development, nor deal with the distributed, redundant aspects of embedded sensor systems. We would like to see the development of a model that would show the relationship between node and network failures, and the resultant effect on the application (or applications) using them. This model could then be implemented on a simulator such as Cooja [13, 24], and used to measure the affects of failures. This would help with the identification of parts of the system most affected by environmental failures.

The use of models mentioned above assumes that the software was programmed correctly in the first instance. It would be good to have a static analysis tools that would ensure that the code itself is robust to coding errors, and to identify the parts of the software which are most vulnerable to environment induced failures, like sensor readings, or radio communication. These parts could be identified and flagged up to the developer for either runtime verification, or runtime adaption to enable to node to continue to function. This tool could take the form of a current static analysis tool, like Frama-C [8] or Klee [29], which has been modified to deal with the many linked libraries that compose an embedded system operating system. Another approach would be to use a domain specific language based on a commonly used language (like C) but which constrains it enough to be able to provide some code quality and robustness guarantees. Either approach needs to be able to identify code that it can not make guarantees about, and flag them up for further attention.

Once the risks are understood and the code trusted, there needs to be a deployment methodology that takes into account both the environment of deployment, and the monitoring and adaptation built into the software to cope with environmental uncertainty. The methodology has to include directions for deployment, and run time monitoring of the sensor system. The idea of a deployment methodology can be related at first to similar software development methodologies, and other workflow methodologies that help practitioners deal with large amounts of complexity and uncertainty in a work environment.

The results of the endeavours mentioned above would be to produce tools that could be used during embedded sensor system design time, implementation time, and during deployment. There are further questions that will be raised with the realisation of these tools. Can we accurately recognise a fault? How do we respond to the presence of non-fault causing errors? These are data and information related questions that are an intrinsic part of sensing systems.

There is no clear, scientific methodology for the design, development, and deployment of embedded sensing systems that takes into account the fact that the hardware will fail at very unpredictable rates. These added assumptions of unreliability means that extra considerations need to be taken to assure that a computing system that uses information from embedded sensor systems, or that perform computation on the embedded sensors themselves, can produce the required results under the affects of failure. At the very least, a defensive approach to programming and building embedded sensor systems can reduce the affects of failure on the collection of data. It will take further engineering to be able to assure that a system or computation can make progress and deliver the required performance in the face of highly probable failures.

5 CONCLUSION

The FAILSAFE workshop sought to increase the body of literature regarding WSN deployments and their failures. The aim was to provide more knowledge to deepen our understanding of the causes of failures in WSN deployments. The submitted papers and the panel session discussion pointed clearly the need to understand and manage the effects of the WSN deployment environment. It became clear to us that we need to consider the deployment environment as part of our system. This should be done by creating a models of the effects of environment such as random node death from rain or heat, and bad communication links from radio noise. These models can be built into our simulators, and we can consider their effects when we plan a WSN deployment. We can plan for node losses that may seem excessive at the start of the deployment. Most importantly, we need a way to learn from the mistakes and failures of others so that we do not repeat them ourselves!

REFERENCES

[1] Nancy Alrajei and Huirong Fu. 2014. A survey on fault tolerance in wireless sensor networks. In *Proceedings of the ASEE North Central Section Conference. American Society for Engineering Education.*

[2] Guillermo Barrenetxea, François Ingelrest, Gunnar Schaefer, and Martin Vetterli. 2008. The hitchhiker's guide to successful wireless sensor network deployments. In *Proceedings of the 6th ACM conference on Embedded network sensor systems.* ACM, 43–56.

[3] Maite Bezunartea, Benjamin Sartori, Jacques Tiberghien, and Kris Steenhaut. 2017. Tackling malfunctions caused by Radio Duty Cycling protocols that do not appear in simulation studies. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems.* ACM, 10–15.

[4] Roy Billinton and Ronald Norman Allan. 1992. *Reliability evaluation of engineering systems.* Springer.

[5] Carlo Alberto Boano, Markus Schuß, and Kay Römer. 2014. EWSN Dependability Competition: Experiences and Lessons Learned. *Newsletter* 2014 (2014).

[6] Samira Chouikhi, Inès El Korbi, Yacine Ghamri-Doudane, and Leila Azouz Saidane. 2015. A survey on fault tolerance in small and large scale wireless sensor networks. *Computer Communications* 69 (2015), 22–37.

[7] Peter Corke, Tim Wark, Raja Jurdak, Wen Hu, Philip Valencia, and Darren Moore. 2010. Environmental wireless sensor networks. *Proc. IEEE* 98, 11 (2010), 1903–1917.

[8] Pascal Cuoq, Florent Kirchner, Nikolai Kosmatov, Virgile Prevosto, Julien Signoles, and Boris Yakobowski. 2012. Frama-c. In *International Conference on Software Engineering and Formal Methods*. Springer, 233–247.

[9] Milan Cvjetkovic and Veselin Rakocevic. 2017. Relative Localisation Algorithm for Neighbour Classification in Ad Hoc Networks of Moving Robots. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*. ACM, 46–53.

[10] Luciana Moreira Sá De Souza, Harald Vogt, and Michael Beigl. 2007. A survey on fault tolerance in wireless sensor networks. *Sap research, braunschweig, germany* (2007).

[11] Ahmad H Dehwah, Mustafa Mousa, and Christian G Claudel. 2015. Lessons learned on solar powered wireless sensor network deployments in urban, desert environments. *Ad Hoc Networks* 28 (2015), 52–67.

[12] Salvatore Distefano and Antonio Puliafito. 2007. Dynamic reliability block diagrams vs dynamic fault trees. In *Reliability and Maintainability Symposium, 2007. RAMS'07. Annual*. IEEE, 71–76.

[13] Joakim Eriksson, Fredrik Österlind, Niclas Finne, Nicolas Tsiftes, Adam Dunkels, Thiemo Voigt, Robert Sauter, and Pedro José Marrón. 2009. COOJA/MSPSim: interoperability testing for wireless sensor networks. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 27.

[14] Xinwei Fang and Iain John Bate. 2017. Issues of using wireless sensor network to monitor urban air quality. In *International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems (FAILSAFE)*. ACM.

[15] Robert Hartung, Ulf Kulau, Björn Gernert, Stephan Rottmann, and Lars Wolf. 2017. On the Experiences with Testbeds and Applications in Precision Farming. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*. ACM, 54–61.

[16] Timothy W Hnat, Vijay Srinivasan, Jiakang Lu, Tamim I Sookoor, Raymond Dawson, John Stankovic, and Kamin Whitehouse. 2011. The hitchhiker's guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 232–245.

[17] Greg Jackson, Sarah Gallacher, Duncan Wilson, and Julie A McCann. 2017. Tales from the Wild: Lessons Learned from Creating a Living Lab. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*. ACM, 62–68.

[18] Niina Kotamäki, Sirpa Thessler, Jari Koskiaho, Asko O Hannukkala, Hanna Huitu, Timo Huttula, Jukka Havento, and Markku Järvenpää. 2009. Wireless in-situ sensor network for agriculture and water monitoring on a river basin scale in southern Finland: Evaluation from a data userâĂŹs perspective. *Sensors* 9, 4 (2009), 2862–2883.

[19] K. Langendoen, A. Baggio, and O. Visser. 2006. Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*. IEEE, 8–pp.

[20] Suzanne Little, Dian Zhang, Camille Ballas, Noel E O'Connor, David Prendergast, Keith Nolan, Brian Quinn, Niall Moran, Mike Myers, Clare Dillon, et al. 2017. Understanding packet loss for sound monitoring in a smart stadium IoT testbed. (2017).

[21] Hai Liu, Amiya Nayak, and Ivan Stojmenović. 2009. Fault-tolerant algorithms/protocols in wireless sensor networks. In *Guide to Wireless Sensor Networks*. Springer, 261–291.

[22] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. 2002. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Acm, 88–97.

[23] Ramona Marfievici, Pablo Corbalán, David Rojas, Alan McGibney, Susan Rea, and Dirk Pesch. 2017. Tales from the C130 Horror Room: A Wireless Sensor Network Story in a Data Center. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*. ACM, 24–31.

[24] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. 2006. Cross-level sensor network simulation with cooja. In *Local computer networks, proceedings 2006 31st IEEE conference on*. IEEE, 641–648.

[25] Lilia Paradis and Qi Han. 2007. A survey of fault management in wireless sensor networks. *Journal of Network and systems management* 15, 2 (2007), 171–190.

[26] Duarte Raposo, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. 2017. A Taxonomy of Faults for Wireless Sensor Networks. *Journal of Network and Systems Management* (2017), 1–21.

[27] Matthias Ringwald and Kay Romer. 2007. Deployment of sensor networks: Problems and passive inspection. In *Intelligent Solutions in Embedded Systems, 2007 Fifth Workshop on*. IEEE, 179–192.

[28] Isabel Madeleine Runge and Reiner Kolla. 2017. MCGC: A Network Coding Approach for Reliable Large-scale Wireless Networks. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*. ACM, 16–23.

[29] Raimondas Sasnauskas, Olaf Landsiedel, Muhammad Hamad Alizai, Carsten Weise, Stefan Kowalewski, and Klaus Wehrle. 2010. KleeNet: discovering insidious interaction bugs in wireless sensor networks before deployment. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 186–196.

[30] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler. 2004. Lessons from a sensor network expedition. In *EWSN*, Vol. 2920. Springer, 307–322.

[31] Le Tian, Serena Santi, Steven Latré, and Jeroen Famaey. 2017. Accurate Sensor Traffic Estimation for Station Grouping in Highly Dense IEEE 802.11 ah Networks. In *Proceedings of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*. ACM, 1–9.