Improving the Dependability of Sensornets

Mark Louis Fairbairn[1], Iain Bate[1], John A. Stankovic[2]
[1]Department of Computer Science, University of York, York, YO10 5GH, UK
[2]Department of Computer Science, University of Virginia, Charlottesville, USA
{mark.fairbairn, iain.bate}@cs.york.ac.uk, stankovic@cs.virginia.edu

***Abstract*—Wireless Sensor Networks (WSNs) are being developed and deployed in a wide range of Cyber-Physical systems, some of which must be dependable, e.g. in assisted living facilities where their failure could lead to an accident. In this paper, it is shown that the state of the art approaches do not meet the needs of dependability that these applications require. The main reason is an issue is the unpredictable physical environment in which they operate. Currently there is little emphasis on how these systems behave when failures occur, instead authors emphasise average case performance. Consequently there is little understanding of how and why systems fail and the possible consequences e.g. a system hazard. In this paper simulated tests are used at run-time to check key dependability properties of the system. The results of these tests are used to plan maintenance, thus ensuring available and reliable operation, and determining when the system is at risk of subjecting people to unacceptable hazards such that appropriate steps can be taken. Our approach has been show to perform with 15% less time at risk than the current state-of-the art.**

I. Introduction

Wireless Sensor Networks (WSNs) are being developed to monitor the physical environment around us upon which we regularly rely upon, from monitoring the condition of cables within suspension bridges [15], the size of cracks within concrete bridges [6] and expansion and contraction of tunnels [15]. This not only provides a comprehensive log of information over time, but also allows the system to directly alert the users should a major change in the physical state be detected. With the increased use of these systems however, there is a tendency for the users of these systems to become more reliant upon the data they provide. This means that should the systems fail, e.g. due to random or systematic events, a hazard may occur that could ultimately caused injury or loss of life. Furthermore there is significant evidence that current WSN deployments have dependability issues [4], [18].

For typical sense-and-send applications node failures would be easily detected by the lack of information from specific sensors. However WSNs are also being proposed for use within event driven systems such as fire detection systems [8]. Typically in these systems, only upon an event occurring do sensors generate data and alert the operator. Therefore it is hard to distinguish between no events happening and the system failing silent.

Whilst systems have been proposed to either check for failed devices or that the network is functionally correct with respect to the application, to the best of our knowledge no publication exists that addresses the end-to-end dependability of a WSN with respect to its specific application. For this work the definition of dependability we use is the alternative proposed by Avizienis et al [1], "the ability of a system to avoid service failures that are more frequent or more severe than is acceptable". This alternative definition differs from the traditional view as it acknowledges that most systems cannot be guaranteed never to fail, however if the severity of the failure falls within the specified acceptable bounds then it can still be viewed as dependable. Dependability can be decomposed into a number of specific attributes [1]. The dependability attributes that we address within this work are listed below.

- Availability - Readiness for correct service
- Reliability - Continuity of correct service
- Safety - Absence of catastrophic consequences on the user(s) and the environment
- Integrity - Absence of improper system alterations.
- Maintainability - Ability to undergo modifications and repairs.

To address these attributes we do not focus on eliminating the causes. Even for simple systems safety analysis is hard and for WSNs it is more difficult as the external factors vary dramatically with the deployment scenario. This difficulty also increases for WSNs due to their dependency on the physical environment which commonly changes during their operational lifetime. Instead we will discuss and demonstrate how the environment in combination with the software can affect the dependability attributes of the WSN system, and to what level these effects can be reasoned about and mitigated through the WSN design.

The method in this paper provides the following contributions:

- Detecting when the system is approaching a state when the applications semantics may no longer be met. This allows for maintenance to be performed before there is a failure to provide the expected service (reliability) and before there is a service outage (availability). The current state-of-the-art approaches only react once the system enters the time at-risk-state.
- Reducing the likelihood of hazards (safety) and showing how additional Derived Safety Requirements (DSRs) can be supported. This results in 15% less time at risk that the current state-of-the-art.
- Reduction in the number of maintenance requests needed (maintainability). There were 97% fewer maintenance

requests for the same time at risk as the traditional approach (the lowest time at risk of all approaches).
- Tuning the parameters of Dependability Assurance (DA) to get an appropriate balance between the dependability attributes above and the detection rate of events specifically associated with DA. We show that for lower node failure rates our system performs close to the best possible time at risk.

The structure of the paper is as follows: Section II provides a review of the current state-of-the-art on dependability, the DSRs, attributes of dependability, existing Health Monitoring (HM) systems, finishing with our proposed approach and how it compares to the current systems. Section III provides a worked Case Study of our method, showing how the DA consists of Dependability Tests (DTs) and how these meet the DSRs. Simulations and a physical experiments are performed and the results analysed. Section IV provides an overall conclusion of the DA system.

II. Components of Dependability

The purpose of this section is to present the current state of dependability within WSNs. An overview of the DSRs to be used in this paper will follow, along with discussion as to how these relate to the attributes of dependability. Finally a review of existing related work is provided on the HM systems to be compared against our proposed approach.

A. Dependability Methods

Previous works related to dependability of WSN's [9] [3] have focused heavily on fault injection into specific parts of the underlying software stack. They have however failed to focus on the specific application. There are only a few works within the literature that specifically target dependability, with the majority by Sailhan et al [13] [12]. Sailhan demonstrates the use of fault injection in order to measure the impact of changes within the physical environment upon the dependability of a WSN. This work however fails to use any formal derivation for the requirements, leading to issues such as timing of events being omitted. Previous works have also lacked of any concrete method of generating, performing and evaluating tests. This work in this paper addresses these issues.

B. Derived Safety Requirements

For the purposes of this paper we will analyse a fire detection system (further details in section III) similar to that proposed by Wu et al [16] based on the DSRs presented in Bate et al [2]. The DSRs in Table I represent the requirements for the monitoring system to ensure that the underlying hazards of the system are mitigated.

The precise nature of the DSRs and the case study are not the main emphasis of this work. The key point is how the DSRs relate to dependability, and how the methods proposed efficiently achieve these dependability goals.

Availability - This attribute is covered by DSR1. The requirement to have a minimum number of nodes (Y) within an area in order to detect the event; and DSR2, the timely reporting of events. These two DSRs are directly defined by the physical number of nodes that are required within an area to detect the event, and the longest possible delay the operators are willing to accept between an event occurring and an event being reported.

Reliability - The DSRs cannot directly alter the reliability of the underlying technologies, e.g. the nodes, the communications system etc. Instead these factors are a manifestation of the physical environment in which the WSN operates. The DSRs however directly affect the ability of the system to meet the application semantics, tolerating the effects of the physical environment, assuming the appropriate maintenance steps are taken.

Safety - The WSN is not directly safety-related as it cannot cause direct harm to people as it has no physical effectors. However DSRs 1-4 are concerned with correctly detecting a fire and DSR5 with informing the operator when the WSN no longer has this capability. As such all these DSRs all contribute to the overall safety by reducing the likelihood of hazards.

Integrity - Integrity of the HM system can be compromised by the physical environment affecting two major parts of WSN operation.

1) Data being sampled - DSR3 ensures errors with the sensors on the devices are detected and reported to the operators.
2) Communications between nodes - DSR4 ensures anomalies such as bit errors in communications are accounted for.

Maintainability - All the DSRs help inform when and what types of maintenance are required. In section III-E how the information is used to plan maintenance schedules and maintenance procedures is discussed.

C. Health monitoring

In order to ensure that the above DSRs are accounted for at run-time it is necessary to perform monitoring of the systems health. Full assessment of the events cannot typically be performed ahead of time as the exact positioning of nodes is not known. This is due to WSNs being commonly deployed in an ad-hoc fashion to provide additional functionality, without major modification to existing infrastructure. More importantly the exact characteristics of the environmental noise, due to the building or other factors, is commonly unknown and changes continuously during the run time of the system. There are two types of approaches for HM, passive approaches and active approaches.

DSR	Description
1	Detect event when fewer than Y nodes are operational
2	An event is reported within X seconds
3	Larger errors and implausible values from sensors are detected
4	Network is tolerant to anomalies
5	Monitoring failure is detected within W seconds

TABLE I
Summary of DSRs

Dependability Attribute	HB	RTA	DA
Availability	Ensures all nodes are available	Ensures one node per area is available	Configurable number of available nodes per area
Reliability	High due to redundancy	Low due to minimal redundancy	Varied depending on configured number of alive nodes
Safety	No Check	No Check	Checks executed to ensure DA system status
Integrity	No Check	No Check	Checks that erroneous output with be reported to the sink node
Maintainability	Constantly requires maintenance	Assumes instant repair, unrealistic maintenance	Varied depending on configured number of alive nodes

TABLE II
PRELIMINARY ASSESSMENT

Passive approaches rely upon the normal communications within the network to obtain information, i.e. deciding if communication looks likely based upon the monitoring of Link Quality Interval (LQI) values [5]. As our application typically does not transmit information until an event occurs, this approach is unsuitable, and so active monitoring must be used. The most common approach to active monitoring is periodic heartbeats (HB) to detect failures [10], however these only detect failed nodes and not the WSN's ability to deliver the expected application semantics. A consequence of this is that maintenance may be ordered before it is needed, breaking one of our main attributes of dependability, maintainability. Within our fire example this issue would manifest itself through maintenance being scheduled within a room despite sufficient nodes to detect a fire, as any failed nodes present within the same room would trigger the HM system.

Wu et al. [16] proposed RTA, which aims to 'provide confidence in application-level requirements for WSNs at run time', as opposed to simply checking if all nodes are alive. The approach is motivated by a WSN often being reactive which means for long periods the WSN may be quiet indicating there is no sensed data worth reporting. However it could also mean that the WSN has failed. Wu et al 's approach proposes that periodically the expected event is *simulated*, which allows the sink node, or device, to determine whether the WSN still has enough nodes to successfully meet the application requirements. We performed a preliminary analysis of Wu et al's approach and found that despite it using more messages than a traditional HB system, it did not raise false positives causing nodes to be replaced unnecessarily.

Given the dependability attributes proposed in section II-B, Table II contains an initial assessment of the existing HM's ability to satisfy these attributes. The table clearly shows that both approaches do not meet all our requirements. Across all the HM systems it was clear that no approach takes into account the overall dependability of the system. The HMs focus only upon availability, most commonly indicated by an unrealistic assumption that maintenance, once ordered, is immediate. Our approach needs to take into account other factors not addressed within these methods, such as the time taken between a failure being detected and the maintenance occurring. Not handling this could lead to a 'period at risk' when the system does not meet the application requirements. The table demonstrates that all the objectives are met by the DA method presented in this paper.

D. Proposed Method for Dependable HM

As the approach used by Wu et al is the only method to correctly identify failures without false positives, we use the same idea of simulating events to measure the impact upon the system within our work. This approach is used in combination with the idea of fault injection identified within the previous works on dependability. Fault injection is used not only to test for functional correctness of the system but also to test correctness under set failure conditions. This allows us to cover all the DSRs in Table I and identify the failures correctly.

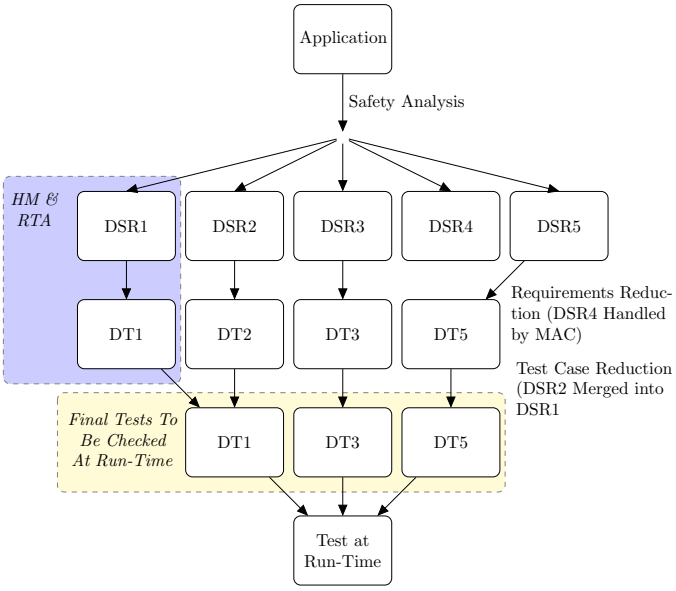

Fig. 1. Figure showing the generation of the Dependability Tests

Figure 1 shows the method for deriving the Dependability Tests (DT). The first stage, which has been covered in a previous paper [2], is the derivation of the DSR's from the application. The derived DSR's are shown above in Table I. Secondly is the reduction stage where we reason about failures that invalidate the DSR's and how they might occur. Within this application DSR4 (handling of anomalies) is handled by the lower aspects of the protocol stack, and as such no tests are needed. The second reduction is the observation that DSR2 (packet delay) equates to packet loss, and thus is covered by DSR1. This transformation of delay into packet loss is due to the time scales involved in the application being in seconds. At the level of seconds the routing layers and MAC layers

will have timed out delayed messages, dropping the packet. This provides the final set of tests for the DA system. The area currently covered by HB and RTA are indicated.

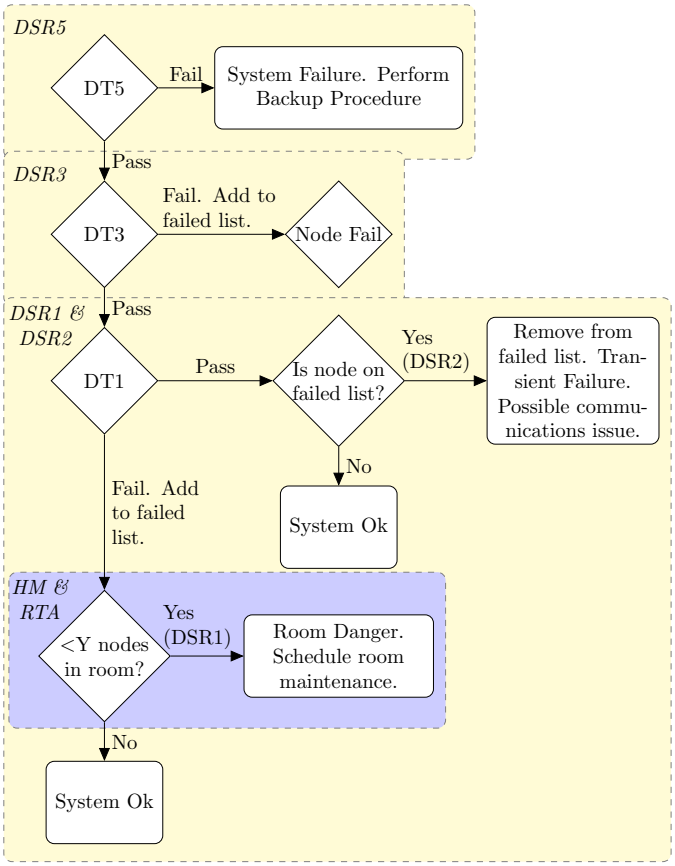

Fig. 2. Figure showing the actions that can be taken within DA

Figure 2 shows the three remaining DTs after the reduction from the 5 DSRs and how these cover DSRs 1-4 and DSR 5. These DTs are checked at run-time as the core part of the DA. DSR2, DSR3 and DSR5 are not covered by any other HM approach and DSR1 that both HB and RTA cover is indicated.

III. Case Study

Within this section we will describe the DA system, how it comprises of DTs and the maintenance that may be performed. The subsystems required from the nodes such as time synchronisation will be described and reasoned about, followed by the main experimental parameters for the simulation environment and the physical system.

A. Scenario / Application Specification

There are a number of possible scenarios that could be used to analyse typical WSN deployments, ranging from environmental controls [17] to logistics [11]. Any event-driven scenarios would be suitable for analysis, however we have chosen fire detection as it is one of the less trivial scenarios, as the failure of the system could directly lead to loss of life. Within the fire detection system there are two types of WSN nodes, temperature detection nodes and the operator node. These node types were chosen as they are the same used by Wu et al. It is easy to envisage how a real application may require multiple sensor types per node, however these would simply be included within the reported readings and as such do not affect the analysis of the DA system. In the context of this paper it is assumed that temperature detecting nodes broadcast a warning to the operator if they detect a possible fire. If the operator node receives one or more warnings about a fire then the room is flagged as containing a fire, with the fire service being alerted in a real deployment.

An important issue for WSNs is when and how the system is deployed. The WSN may have been statically deployed before the event (specifically to monitor the building), or perhaps having been dynamically deployed (e.g. as breadcrumbs for communication around the site [7]) as the event is ongoing. For the purposes of this paper a static, pre-defined deployment of nodes is assumed which monitors the building constantly. By assuming this static scenario we can assess the likelihood of fires occurring and the effect they have. We can also assess what effect the parameters of the monitoring system have upon the dependability of the WSN. Throughout this paper it is assumed that there is only one operator node, to which all monitoring applications report, to more closely match the previous literature. However in practice, especially in large buildings, there would be multiple spatially separated operator nodes physically networked together to avoid a single point of failure, e.g. a fire breaking out where the operator's console is situated.

There are a number of assumptions within previous literature that are relaxed within this paper, primarily that maintenance occurs instantaneously. Secondly we relax the criteria that a fire must be detected immediately or is otherwise a false negative (unreported fire). Instead we opt for fire detection within a fixed period. This more accurately reflects the physical environment where detection within some specified time of the fire starting may be satisfactory. An arbitrary duration of 5 minutes is chosen for this work and is discussed further in section III-D2.

B. Application of DTs

The purpose of this section is to describe the tests that form the DA (DTs), and how these meet the DSRs, with specific parameters being investigated within section III-D. It is assumed that the operator node is initially unaware of the number of rooms, the number of nodes or the location of any nodes. This information is built up as nodes report their DT messages at runtime, with their unique identifier and their location (in this case which room they are within) to the operator node. Once information from a specific node has been added it is continuously monitored, including after failure, and it is assumed that any replacement nodes have the same room designation. Whenever these DTs are performed a DT test flag is included within all outgoing messages to indicate that they have been generated during this time period and thus can be distinguished from a real message.

DSR1 is checked by DT1, an on-line simulation of fires within a specific room performed every HM period. These simulations are constructed by returning emulated values instead of the real values for the temperature sensors. Emulated values are set such that the temperature is sufficiently high to indicate a fire. Should any node from the room being tested not alert the operator node to the presence of the test fire then the node in question is flagged as having a DT1, and thus DSR1 failure.

DSR2 raises a DSR failure if any nodes DT message does not arrive at the operator node after X amount of time. Within the fire scenario X is assumed to be 1 minute, as this is an early indication of either a node failure, node communications being disrupted, or messages taking too long to reach the operator node. All of these issues should be handled by the maintenance procedures. No extra DT messages are generated for this DSR, instead it relies on the messages produced from DT1. The test for DSR2 is performed on the operator(s) consoles.

DSR3 is another on-line simulated test, using DT3, with the node simulating the presence of a temperature sensor fault by injecting temperatures well outside normal operating range for non-fire and fire scenarios. These clearly incorrect readings should be caught by the nodes internal logic, and alert the operator node to a DSR3 failure. Should a DSR3 failure not be reported to the operator node, the node is marked as failed. The raw sensor values are not transmitted to the operator as these may be large depending on the application. For example if the alarm is raised based upon statistical evaluation of the sensor readings every T ms, then we would not want all the readings being transmitted to the operator.

DSR4 is not tested by any of the HM systems as it is assumed low level errors such as bit-errors or routing path issues are handled by the appropriate protocol. Any issues involving local denial of service, such as physical destruction of nodes or wireless jamming are identified with DSRs 1 and 2, in combination with the locality of the errors. Global denial of service is identified by DSR5. The security of the WSN, in the form of intrusion into the software on the nodes is outside the scope of this paper.

DSR5 is the total failure of the HM system and is the last on-line test to be run. Periodically all DTs are suspended for a set period of time. As with DSR1, it is expected that this test is performed on the operator(s) consoles, and that when all DTs are suspended the operators console raises a DSR5 failure, the absence of which indicates a failed test.

If problems are detected then the maintenance procedures from Figure 2 is used.

C. Node Simulation and Physical Implementation

For the experiments we used both simulation and a physical implementation of the application. The simulation allows us to generate a large number of results within a reasonable time frame due to the ability to run numerous tests in parallel and without the need to change any batteries. The physical tests allow us to verify that the simulated results are an accurate representation of the real application behaviour, in addition to verifying that the additional noise from the physical environment does not adversely effect the application.

HM periods and Maintenance periods aim reflect to the context in which the system is deployed, with HM being performed every 6 hours and maintenance every week. Failure times are also picked to be as accurate to real-world situations, with values being picked randomly from the inverse survival function, commonly used in survival analysis to define failure times, with a mean of 30 days and a standard deviation of 3.

Implementation: NS-2 is used to simulate the network level for the nodes, allowing for high fidelity simulation. NS-2 is configured for a typical WSN application, AODV for the transport layer with 802.15.4 for the MAC protocol used in combination with the two-ray ground model for the physical radio model. To verify that the behaviour observed in the simulation of the DA is an accurate representation of the systems behaviour a smaller scale physical deployment was conducted. For this 12 TelosB nodes were used which have been tested to ensure suitable clock drift tolerance. The application logic is implemented on top of TinyOS 2.1.1, with the NST-AODV routing protocol, along with the use of the standard 802.15.4 MAC implementation within TinyOS. The majority of the node logic was automatically generated from the simulation logic adding to the confidence in its correctness. Transmission power levels were adjusted before the experiment so that only communication between rooms is reliable.

D. Deployment and Evaluation Metrics

This deployment was conducted with 12 nodes evenly deployed over 4 rooms, with communications range being limited to neighbouring rooms. As the experiment was to demonstrate the inter-room connectivity, not connectivity to the sink, the room containing the sink node does not experience any node failures. This ensures that the last hop to the sink has a greater chance of success. Figure 3 shows the layout of the nodes.

There are three possible criteria that can be used to measure the cost of running the HM system, the one used by most literature is the number of packets used by the network, however we also suggest two additional metrics; number of maintenance requests and time at risk. Maintenance requests are used as these can be directly associated with calling in contractors to replace broken nodes, which incurs large costs in addition to being one of the important dependability attributes as stated in section II-B. Another metric is the notion of time at risk, an important aspect of safety maintainability, which is how often and how long a room has no coverage, possibly missing the initial stages of a fire. DSR1 is the only DSR common to all the HM systems, however our system improves on the current state of the art by allowing the operator to tune the network to either of the above metrics to obtain the desired availability (how low are we willing to let the node population go), maintainability (how often are we willing to call out maintenance) and battery life (directly detecting the maintenance schedule) trade-off.

Fig. 3. Layout of the physical nodes within the rooms.

1) DSR1: DSR1 establishes that there should never be less than Y fully working nodes per room, and as such is concerned with detecting node failures so that maintenance can be scheduled.

Simulated Deployment

Initial testing was done within the NS-2 simulator, with simulations running for 1/2 a year of simulation time which equates to 34 minutes of elapsed computing time. Tests were performed 10 times for each of the experiments, with an experiment run for each of the node types. The results are shown in Table III.

Node Type	Time At Risk	Maintenance requests	Number Of Messages
HB	15	58	61275
RTA	7955	16	43916
DA	2198	23	46700

TABLE III
OVERVIEW OF THE THREE HM SYSTEMS IN THE SIMULATED DEPLOYMENT SHOWING THE TIME ROOMS WERE AT RISK, THE AMOUNT OF MAINTENANCE AND THE NUMBER OF FAILURES.

Here time at risk is the amount of time where there is no ability to detect a fire should one occur. It can be clearly seen how DA incurs much less time at risk than RTA, with a small increase in maintenance requests, whereas HB has the lowest time at risk, but incurs large numbers of maintenance requests and a large increase in the number of messages.

Physical Deployment

To validate the results from the simulation it was decided to run a physical deployment of the same fire detection scenario described in section II-B.

So that the experiments could produce results within a reasonable time frame the tests were accelerated by 800 times, allowing 1/2 a year to pass in 5.5 hours, causing a HM test to run every 45 seconds. With this level of acceleration collisions between packets within the same test may become more frequent, which may prove pessimistic. Pessimism is still safe for our dependability testing as it assumes the worst case. However as identified earlier even a small level of desynchronization will alleviate this issue.

	Time at risk	Maintenance requests	Total Failures
HB	23806	72	47
RTA	63996	52	45
DA	31532	65	45

TABLE IV
OVERVIEW OF THE THREE HM SYSTEMS IN THE PHYSICAL DEPLOYMENT SHOWING THE TIME ROOMS WERE AT RISK, THE AMOUNT OF MAINTENANCE AND THE NUMBER OF FAILURES.

Table IV shows the results of the experiments. It clearly shows that RTA allows a large time-at-risk, which is unacceptable for a fire detection system. 170% more time at risk in RTA for a 28% reduction in maintenance requests when compared to 32% increase in time at risk for DA for a 10% reduction in maintenance requests.

2) DSR2: DSR2 is the timely delivery of alerts to the operator node. Within our fire detection system we assume that timely equates to less than 5 minutes between detection and the operator being alerted. Tests were undertaken to assess how the worst case time for the message to reach the destination is affected by the number of nodes. This showed that as the number of nodes approaches 128 the time increases up to 22 seconds, however at 256 nodes it decreases to 13 seconds. The number of dropped packets however increases with the number of nodes which is consistent with the literature [14]. Further experiments showing that with a small spread of 1.8 seconds between the nodes the number of missed messages drops back down to 0% for 256 nodes. From this experiment we can deduce that as long as all the nodes are roughly synchronised (i.e. to within the same 10 seconds) then there should be no substantial packet loss. The time it takes for packets to travel does not need to be monitored, instead monitoring the loss of packets is enough to ensure dependability. For this reason tests still need to be executed to check that DSR2 holds.

3) DSR3: The third DSR exclusively deals with nonsensical output from the nodes, as it is important that erroneous sensor output is detected as soon as possible after it has occurred. This error should also be reported to the operator immediately so that maintenance can be scheduled accordingly. Only large errors that are clearly outside the expected range of values can be detected confidently, however sensors are typically designed to fail with Full Scale Deflection. Tests were conducted showing that these were correctly handled by DA, with HB and RTA raising many false positives.

Monitoring Type	Real Fires	Detected Instances	Maintenance Requests	False Positives
HB	75	286	374	134005
RTA	84	278	418	122233
DA	77	156	159	7

TABLE V
THE NUMBER OF FALSE POSITIVES (FIRES DETECTED THAT DO NOT ACTUALLY OCCUR) REPORTED BY DIFFERENT HM SYSTEMS WHEN PRESENTED WITH REAL FIRES AND FAULTY SENSORS

Table V clearly shows the significantly increased number of false positives reported by HB and RTA, leading to more maintenance requests. It should be noted that looking at false negatives (unreported fires), all three of the systems report fires within the 5 minute window. This is due to the high statistical chance that a faulty node is reported as a fire at the correct times, due to it randomly receiving readings from the sensor.

4) DSR4: The fourth DSR states that the nodes should be tolerant to anomalies, and as such should deal with node and communications failures. Failure to deal with these anomalies can lead to a failure of the system to report a fire. From experiments the fire detection application showed tolerance to transient failures as fires were still successfully detected. The HB based system however scheduled an unnecessary large amount of maintenance. Both RTA and DA are comparable in the number of missed fires and the number of maintenance requests, as the majority of nodes have to simultaneously be in the failed state to raise a maintenance request.

5) DSR5: Finally the fifth DSR requires that a monitoring failure be reported to the operator nodes. This is extremely important as any monitoring failure must be handled immediately as the application may not be fulfilling its requirements, which in the case of the fire detection system could lead to loss of life. Should the monitoring system itself fail on either the HB system or the RTA system then the end user is none-the-wiser as both systems are fully autonomous and never inform the user of anything apart from if maintenance is required. DA detects the failure of the DTs and then alert the operator that the system needs to be repaired and any backup systems should be used.

E. Parameter Tuning

Throughout the evaluation it became clear that there are a number of different factors that may affect the HM systems, and the network as a whole. This section aims to investigate a number of these factors with a larger number of nodes and a larger number of rooms.

Nodes are deployed in a series of rooms containing varying numbers of nodes. There were 20 nodes in DSRs 2-5, with either 1,2,4,5,10 or 20 nodes per room. The default scenario used is 4 nodes in 5 rooms. All nodes within a room are within the full transmission range of the neighbouring rooms only, with rooms arrange in a simple column, thus causing a pessimistic case where the loss of a single room between the source and the sink can cause an impassable network void. This allows for single-hop communication amongst neighbouring rooms, but requires multi-hop communication as provided by AODV for communicating further.

Within each simulation run, there was a series of simulated fires in a random room, which should be reported to the operator node. All nodes constantly monitor their room for fires, checking the value of the sensors every 5 minutes. Fires last for a duration of 10 minutes, after which the fire is removed as detection after this time interval is unacceptable. All experiments within this section have an average of 590 fires. All experiments were run for 7 simulation days, and 50 trials were run. The data points in the resulting figures and tables are the mean values over all trials.

1) Effect of Number of failures: The initial experiment looks into how the number of failures affects the fire detection rate of the system. Nodes have a maintenance period of 5 minutes, with the failure rate being increased from zero failures per week to 100 failures per week. This is especially large number of failures with a small maintenance period, and as such presents an extreme scenario when compared to a real world example. The number of unreported fires is recorded, i.e. the false negative rate.

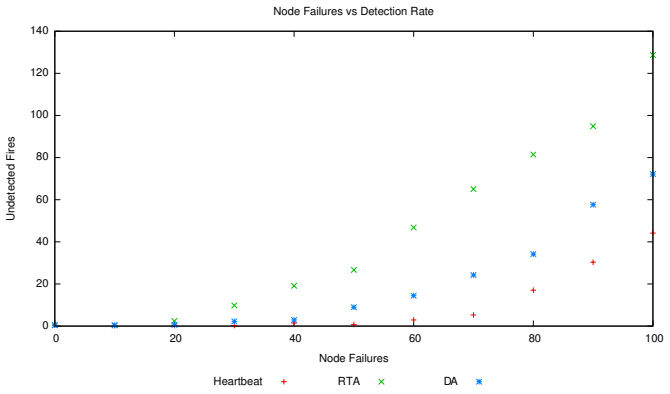

Fig. 4. Variations in the HM period affects the number of detected fires.

Figure 4 shows how as the number of failures is increased, more fires go undetected. As the number of failures reaches 50, the HM period of 1 hour becomes too small to repair the nodes in time, thus HM starts missing more fires.

Failures	HB	RTA	DA
0	0	0	0
10	7.9	0.0	0.2
20	13.1	0.2	1.2
30	17.1	1.2	3.0
40	19.6	2.0	4.6
50	21.1	2.7	5.6
60	22.5	3.9	7.0
70	22.8	4.9	8.0
80	23.2	5.8	9.1
90	23.7	6.0	9.8
100	24.2	7.0	11.0

TABLE VI
NUMBER OF MAINTENANCE DISPATCHES FOR DIFFERENT NUMBERS OF FAILURES

HB appears to perform the best in these scenarios due to it requesting maintenance when any nodes have failed, followed by DA and then RTA. Table VI shows that the number of

maintenance however tells a different story. HB needs a huge number of maintenance requests. In contrast when there are small number of failures, DA detects similar numbers of fires, but with much lower maintenance requests.

2) Effect of Length of HM period: To ascertain the effects that the HM period has upon the number of detected fires more experiments were performed. Within these experiments there is a fixed rate of 100 failures per week and the HM period is varied from 1 to 24 hours in increments of 4 hours and the number of undetected fires is reported.

Figure 5 shows the results of the second experiment showing that as the period of HM is increased the number of correctly reported fires decreases. This is due to the HM systems being unable to repair failed rooms in time for the next fire to be detected. This demonstrates that for a particular deployment the HM period would need to be chosen carefully, based upon the failure rate and the number of maintenance dispatches acceptable to get the desired level of assurance. DA also allows the number of nodes that are required before maintenance is scheduled (node threshold) to be adjusted, thus allowing DA to have the same behaviour as HB, or at the other extreme RTA. In this example the node threshold for DA was set to two.

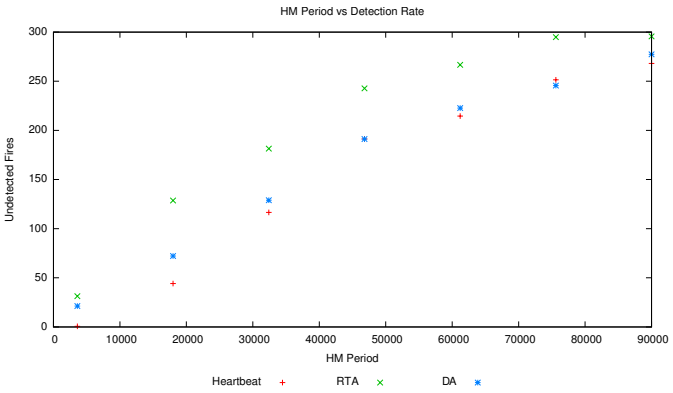

Fig. 5. Variations in the HM period showing changes in fire detection rates

The main outcome from these experiments is that the DA system is highly parametrisable based upon the application requirements. For example if DA is required to behave like HB then the number of allowed node failures within each room can be set to zero. However an appropriate balance needs to be reached between availability, and the number of maintenance dispatches. The precise values would need to be tuned based on specific safety requirements, i.e. tolerable risk of hazards and then monetary cost of maintenance. An example physical deployment of DSR2 comparing the three HM systems is given in section III-D1.

Using the results from this section it can be seen that within a more representative real-world scenario, with the HM test running every 5 hours and 10 node failures per day, this tuning can be beneficial. In this scenario DA achieves the same level of fire detection as HB (0.31 missed fires out of 672), whereas RTA misses 18% more fires. From these experiments it can be seen that the preliminary assessment of HB and RTA with respect to the 5 DSRs as shown in Table II was partially correct. A revised version of the Table, with reasoning is given in Table VII. It is noted the other HM systems were designed with only availability in mind and thus omitted most of the other attributes.

F. Comparison with State-of-the-art

DSR1 is covered by all three HM systems and so Figure 6 shows DSR1 in more detail and highlights the difference between the three systems. Within Figure 6 the red dashed arcs show where the occurrence of node failures causes a change in system state. Red dashed nodes are critical states where the application requirements are not being met. In the case of the fire detection system, this state means fires cannot be detected as there are not enough nodes left alive. Black solid arcs are unconditional jumps and blue dotted arcs are labelled with the particular HM system or condition required to take the transition. Blue dotted arcs with simply HB, RTA or DA with no probabilities are always taken. Alternatively some of the transitions have probabilities labelled, with the values based upon number of failures, failure rates, and maintenance rates.

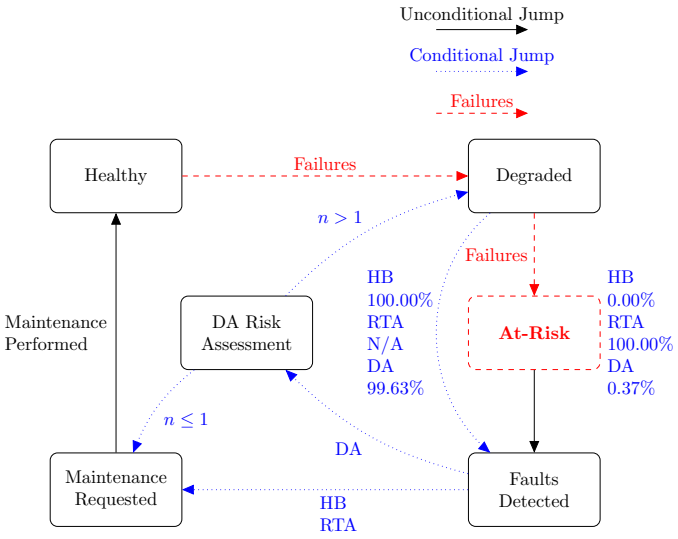

Fig. 6. Figure showing the flow of actions that are taken for DSR1

The probabilities are derived from a simulation of all three HM approaches, with communications being assumed to be reliable as this does not effect the overview drastically. here the probability is N/A within the figure it indicates that the arc can never be taken, whereas 0.00% shows that it is unlikely to take the transition. In this case not a single transition of this type over a 100,000 year simulation occurred. The HM is run every day, and the maintenance occurs 24 hours later. Nodes have a mean lifetime of 3 months, with a deviation of 2 months, based on the inverse survival function. These failures values are assumed to be pessimistic, with higher times between failures simply reducing the probability of DA getting to the at-risk state.

This clearly demonstrates that not only can our approach perform to the high level of reliability as HM provides, but it also does such by checking the application requirements such as RTA does enabling the number of maintenance requests to be reduced. Our approach not only checks that the system is operating correctly, but it also checks that the system, should it degrade, degrades safely with faulty nodes being reported and failure of the entire system being monitored.

IV. Conclusion

As shown within the results it is possible to design a wireless sensor network that allows the system designer to ensure dependability to a set level in the presence of numerous external physical effects. In this paper the concept of simulating events to test the normal working behaviour of the network has been extended to cover a range of error cases and built into an overall method for ensuring a dependability within a WSN application. Dependability is provided by delivering an available, reliable, safe and maintainable WSN, along with integrity in the WSN, as part of an overall Cyber-Physical system. Three DTs are the core of the DA system and allow failures to be directly identified or easily deduced. The particular benefits of the approach is that potential failures are identified earlier so that service is not lost, leading to the application being at-risk, but not that early that unnecessary maintenances are performed. A key focus of the work was also to deliver the dependability attributes of interest to the level of assurance application developers desire. At the same time the application developers must decide the level of trade-off this requires against the number of maintenance requests.

	HB	RTA	DA
DSR1	Incorrectly Scheduled Maintenance	Incorrect Scheduled Maintenance	Identifies Communications Issue
DSR2	Pessimistic	Optimistic	Configurable from HB to DA
DSR3	Incorrectly Raises Alarm	Incorrectly Raises Alarm	Correctly Schedules Maintenance
DSR4	Incorrectly Scheduled Maintenance	Tolerant	Tolerant
DSR5	System Fails Silent	System Fails Silent	Operator Alerted

TABLE VII
Overview of the three HM systems and how they react to the DSRs.

References

[1] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, pages 11–33, 2004.

[2] I. Bate, Y. Wu, and J. A. Stankovic. Developing safe and dependable sensornets. In *Proceedings of the 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 279–282, 2011.

[3] M. Cinque, D. Cotroneo, C. Di Martino, S. Russo, and A. Testa. AVR-INJECT: A tool for injecting faults in wireless sensor nodes. In *IEEE International Symposium on Parallel & Distributed Processing*, pages 1–8, 2009.

[4] J. Decotignie. Real-time and wireless sensor networks: Are they compatible? Keynote at Euromicro Conference on Real-Time Systems, 2012.

[5] M. M. Holland, R. G. Aures, and W. B. Heinzelman. Experimental investigation of radio performance in wireless sensor networks. In *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks*, pages 140–150, 2006.

[6] N. Hoult, P. Bennett, I. Stoianov, P. Fidler, C. Maksimovic, C. Middleton, N. Graham, and K. Soga. Wireless sensor networks: creating'smart infrastructure'. In *Proceedings of the Institution of Civil Engineers*, volume 162, pages 136–143. Telford, 2009.

[7] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu. Automatic and robust breadcrumb system deployment for indoor firefighter applications. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pages 21–34. ACM, 2010.

[8] J. Lloret, M. Garcia, D. Bri, and S. Sendra. A wireless sensor network deployment for rural and forest fire detection and verification. *Sensors*, 9(11):8722–8747, 2009.

[9] P. Pereira, A. Grilo, F. Rocha, M. Nunes, A. Casaca, C. Chaudet, P. Almström, and M. Johansson. End-to-end reliability in wireless sensor networks: Survey and research challenges. In *EuroFGI Workshop on IP QoS and Traffic Control*, pages 67–74, 2007.

[10] S. Rost and H. Balakrishnan. Memento: A health monitoring system for wireless sensor networks. In *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, volume 2, pages 575–584, September 2006.

[11] L. Ruiz-Garcia, P. Barreiro, and J. I. Robla. Performance of ZigBee-based wireless sensor nodes for real-time monitoring of fruit logistics. *Journal of Food Engineering*, 87(3):405–415, 2008.

[12] F. Sailhan, T. Delot, A. Pathak, A. Puech, and M. Roy. Dependable wireless sensor networks. In *3th workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in cunjunction with INFORSID*, pages 1–16, 2009.

[13] F. Sailhan, T. Delot, A. Pathak, A. Puech, and M. Roy. Fault injection and monitoring for dependability analysis of wireless sensor-actuators networks. In *4th workshop Gestion des Données dans les Systèmes d'Information Pervasifs (GEDSIP) in cunjunction with INFORSID*, 2010.

[14] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking*, volume 2, pages 1124–1130, March 2003.

[15] F. Stajano, N. Hoult, I. Wassell, P. Bennett, C. Middleton, and K. Soga. Smart bridges, smart tunnels: Transforming wireless sensor networks from research prototypes into robust engineering infrastructure. *Ad Hoc Networks*, 8(8):872–888, 2010.

[16] Y. Wu, K. Kapitanova, J. Li, J. A. Stankovic, S. H. Son, and K. Whitehouse. Run time assurance of application-level requirements in wireless sensor networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 197–208, 2010.

[17] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[18] F. Zhao. Challenge problems in sensornet research. Keynote at NSF NOSS PI meeting and Distinguished Lectures at Johns Hopkins and Princeton, 2005.