A Handheld Diagnostic System for 6LoWPAN Networks

David Rodenas-Herráiz
Department of Engineering
University of Cambridge, UK
david.rodenas@eng.cam.ac.uk

Tao Feng
Department of Engineering
University of Cambridge, UK
tf292@cam.ac.uk

Sarfraz Nawaz
Computer Laboratory
University of Cambridge, UK
sarfraz.nawaz@cl.cam.ac.uk

Paul R. A. Fidler
Department of Engineering
University of Cambridge, UK
praf1@cam.ac.uk

Xiaomin Xu
Department of Engineering
University of Cambridge, UK
xx787@cam.ac.uk

Kenichi Soga
Department of Civil and
Environmental Engineering
University of California, Berkeley, USA
soga@berkeley.edu

***Abstract*—The successful deployment of low-power wireless sensor networks (WSNs) in real application environments is a much broader exercise than just the simple instrumentation of the intended monitoring site. Many problems, from node malfunctions to connectivity issues, may arise during commissioning of these networks. These need to be corrected on the spot, often within limited time, to avoid undesired delays in commissioning and yet a fully functional system does not guarantee that no new problems will occur after leaving the site. In this paper we present the first ever (to our knowledge) implementation of a handheld diagnostic system for fast on-site commissioning of low-power IPv6 (6LoWPAN) WSNs as well as troubleshooting of network problems during and after deployment. This system can be used where traditional solutions are insufficient to ascertain the root causes of any problems encountered at no additional complexity in the implementation of the WSN. The embedded diagnosis capability in our system is based on a lightweight decision tree that distills the functioning of communication protocols in use by the network, with a major focus on interoperable IPv6 standards and protocols for low-power WSNs. To show the applicability of our system, we present a set of experiments based on results from a real deployment in a large construction site. Through these experiments, important performance insights are gained that can be used as guidelines for improvement of operation and maintenance of 6LoWPAN networks.**

I. Introduction

After years of active research and engineering effort, wireless sensor networks (WSNs) have become a mature monitoring technology and their adoption is rapidly growing in various fields including agriculture, environmental and infrastructure monitoring. A WSN consists of a number of spatially distributed low-power devices (referred to as WSN nodes) with embedded processing and wireless communication capabilities. Power is commonly provided by way of batteries, although the use of energy harvesting technologies (e.g., solar panels) is increasingly gaining momentum [1], [2]. Nodes deployed at specific measurement locations are interfaced with sensors for measuring changes in parameters such as temperature, humidity, strain and acceleration. Acquired data can be processed locally in the sensor nodes and transmitted to a data sink either directly or though intermediate nodes with routing capabilities.

Much of the success of WSNs is derived from their ability to provide faster installation at a lower cost than traditional wired monitoring systems. However, commissioning and operation of these networks remain a challenge, as this is most often the time when unexpected problems, such as communication loss or node malfunctions, arise from unanticipated or underestimated issues [3]. These issues may be caused by factors such as harsh environmental conditions, wireless interference and changes in the layout of the deployment site. Even though an adequate design of the system, including network topology and robust communication protocol design, is crucial for anticipating and mitigating any potential issues, failures in the system during and after commissioning may still occur. Diagnostic systems are therefore essential to avoid delays in commissioning as well as troubleshoot any failures during operation of these networks.

Diagnostic systems have been proposed for WSNs [4]–[11], but they are designed to run on devices with no resource constraints, such as desktop computers or laptops. Much of the work in this context is constrained by the requirement to access the gateway, either remotely from some centralized location (through the Internet), or on site. Although this method may often be sufficient, an alternative solution is necessary when part of the network has lost communication with the gateway and an experienced technician is consequently required to walk around the deployment site to find what is causing such communication loss. Our experience with WSN deployments in the civil engineering field [3], [12], [13] suggests that there is a pressing interest for easy-to-use diagnostic systems that allow engineers to quickly target any problems or issues that the network may be experiencing while on site. Particularly for civil infrastructure monitoring, such systems would be instrumental in improving maintenance practices of sensor networks while minimizing labor costs.

Designing network diagnostic systems is however a complicated task due to the complex challenges arising from the

ISBN 978-3-901882-88-3

energy-constrained nature of WSNs, the variety of communication protocols that may be used in such networks, the requirements of the monitoring application, the conditions at the deployment environment, and the level of functionality and accuracy that diagnostic systems can provide.

Motivated by the preceding challenges, we have designed and developed a handheld diagnostic system capable of mapping problems with potential root causes in standards-based low-power IPv6 wireless sensor networks, also referred to as IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN). By way of real-time capture and analysis of network traffic, our system is capable of distinguishing between failure localized on a node, on the path between a node (or a group of nodes) and the data sink (gateway), and on the data sink itself. To the best of our knowledge this is the first comprehensive work that provides an easy-to-use diagnostic system for commissioning and on-site maintenance of 6LoWPAN networks. The key features of this work are:

- Development of a handheld diagnostic system: A diagnostic system has been developed for small handheld devices, such as a tablet. It can be employed in WSN deployments where walking around with a laptop [9]–[11] may be inconvenient and even risky, such as in construction sites. Furthermore, this diagnostic system adds no complexity in the implementation of the WSN [4], [7], [8] and has no reliance on diagnostics available at the gateway [5], [6].
- Integrability with 6LoWPAN networks: Our diagnostic system design is compatible with standards-based communication protocols at each layer of the network protocol stack. Particularly, the embedded diagnosis functionality in our system is accomplished by constructing a lightweight decision tree based on RPL [14], the routing protocol standardized by the Internet Engineering Task Force (IETF) to provide any-to-any data routing in 6LoWPAN networks. To provide deeper insight into problems arising from communication issues, we have also devised a straightforward method to estimate the quality of the wireless links between nodes.
- Real-world applicability: We present a use case based on previously obtained results from a six-month-long deployment undertaken on a large construction site [13]. The performance of the as-installed WSN at this site was not satisfactory due to continuous connectivity problems during much of the deployment duration. The lack of a suitable diagnostic system at this deployment eventually became the main driving force for this work.

The paper is organized as follows. Section II introduces IPv6-based networking protocols for low-power WSNs. The decision tree-based methodology to infer network problems and associated potential root causes is presented in Section III. Section IV describes the implementation of our diagnostic system. The use of the system is shown in Section V. Section VI presents a review of alternative diagnostic solutions. Section VII offers concluding remarks and directions for future research in this area.

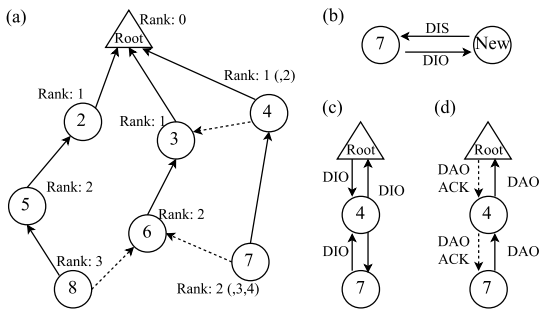


Fig. 1. (a) RPL tree-like topology. Nodes use only one parent for routing messages, although it may change over time depending on the link quality. (b) Joining procedure; (c) Establishment of up routes; (d) Establishment of down routes in non-storing mode with optional DAO-ACK transmission.

II. Overview of 6LoWPAN/RPL-based Networks

A core function of any wireless sensor network is to provide some way of routing information between nodes and the data sink, even if the nodes are deployed multiple hops away. Routing in WSNs is nonetheless challenging, posing a trade-off between energy consumption and overall network performance. Much of the energy consumed by a node is due to transmission, reception and idle listening, and so a routing protocol that minimizes the use of these operations while permitting adequate network performance metrics, such as throughput or latency, is paramount.

With these challenges in mind, and motivated by the likelihood that IP-based networking would become the leading solution for enabling connection of WSNs to the Internet, the IETF standardized the IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [14]. Its design is largely based on the Collection Tree Protocol (CTP) [15], the reference data collection protocol for WSNs, and it is currently present in most 6LoWPAN networks.

RPL is a distance vector protocol that builds upon a hierarchical tree topological configuration or DODAG (Destination-Oriented Directed Acyclic Graph), depicted in Fig. 1a. A DODAG consists of a DODAG root, typically the data sink/ gateway, and a collection of subtrees of child nodes each with a parent node (i.e., directed acyclic graphs or DAGs). The DODAG is constructed based on an objective function in use by all the nodes. More specifically, the objective function is used to compute the rank for a node (i.e., its distance from the root) based on the evaluation of a specific cost metric, such as hop count, the expected transmission count (ETX) or latency [16]. The rank can then be employed to select a potential preferred parent from any candidate neighbors. The parent may change over time, which allows for local or global re-arrangements of the network that aim for continuous acceptable network performance.

The DODAG formation is initialized by the network root by broadcasting DODAG Information Object (DIO) messages. DIO messages, transmitted by all the network nodes, are intended to advertise the DODAG as well as create and maintain routes for upward traffic (i.e., DAGs) (Fig. 1c). Among

other relevant information, a DIO message contains the node's current rank and DODAG-related information including the objective function in use. DIO transmission is driven by a Trickle timer [17], enabling nodes to control routing overhead and to react promptly to network inconsistencies. A DIO transmission is also triggered in response to a DODAG Information Solicitation (DIS) message from a potential new node willing to join the network. DIS messages are intended to pro-actively solicit DODAG-related information (Fig. 1b) from nearby DODAGs. Any network nodes within a DODAG receiving the DIS message may send a DIO message back to the new node, which it then uses to finalize the joining process by selecting a preferred parent.

RPL enables the optional establishment of routes for downward traffic. Simply by using unicast transmission over multiple hops (where needed), every node sends a Destination Advertisement Object (DAO) message directly to the selected parent (if using the storing mode of operation, where every node maintains a routing table) or toward the DODAG root (when non-storing mode, suitable for memory-constrained devices as no routing table is maintained, is used). A Destination Advertisement Object Acknowledgement (DAO-ACK) message may optionally, upon explicit request or error, be sent back as a unicast transmission by its recipient in response to a DAO message (Fig. 1d). The mechanism for transmitting DAO messages is not specified in the RPL RFC [14] (it is left to the developer), but implementations such as the used in Contiki OS follow a similar approach based on DIO transmission.

In addition to RPL, the IETF provided support for route discovery by way of a lightweight modified version of the IPv6 Neighbor Discovery (ND) protocol [18]. RPL may disseminate ND information (essentially route-request and route-reply Internet Control Message Protocol (ICMPv6) messages) when the wireless link with a parent node is detected to be broken (typically when the information stored in the routing table – if the storing mode is enabled – is obsolete) and the source node engages in discovering a path to its intended destination. While this route discovery is performed, any data messages to be sent are buffered in the source node. When a route is established, these messages are then transmitted; however, if no route can be found, the data messages are discarded.

III. Decision Tree for Diagnosis of 6LoWPAN Networks

In this section we present a high-level overview of our diagnosis strategy for low-power standards-based IPv6 WSNs. The intuition underlying our solution is that a WSN works well under specific network and link-layer assumptions concerning (i) the proper functioning of the network protocols, and (ii) the expected successful transmission/reception ratio between two nodes. If either of these assumptions is broken then the network is considered to have a problem. Each problem instance has a particular "signature", which can be attributed to a number of potential primary causes. The goal of our solution is to automatically identify and locate the most likely primary causes for any encountered problems (i.e., "signatures").

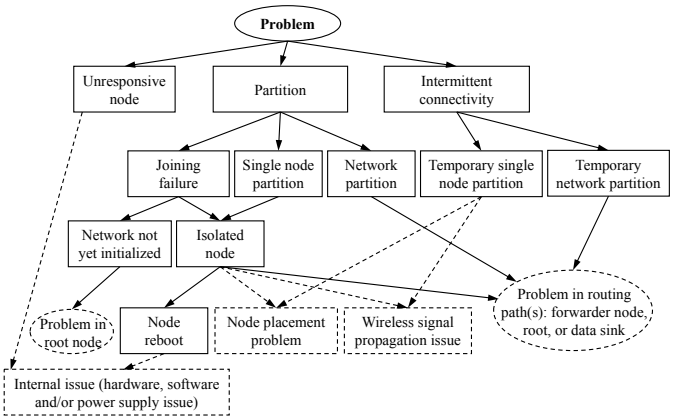


Fig. 2. Simplified decision tree for root cause analysis. Dashed rectangles represent potential root causes that cannot be verified with the current implementation of the system. Dashed ovals represent potential root causes, which cannot be assessed with currently collected network traffic.

Inspired by problem diagnosis approaches such as Sympathy [4] and SNIF [7], we have devised a decision tree that infers the state of each node from network traffic collected and interpreted with a built-in traffic sniffer. Our work departs from the above in that it can be used without the need for additional software installed in the network nodes [4] or requiring several traffic sniffers to be deployed alongside the deployment [7]. Our decision tree, depicted in Fig. 2, is computationally lightweight, fast, and easy to implement in inexpensive tablet-like devices. It breaks down into the following decisions:

Unresponsive node. Most likely causes for a node (including the root node) to become unresponsive include software errors (which may cause the node to enter a blocking state, e.g., due to an infinite loop), faulty hardware, or problems in the power supply (e.g., exhausted batteries). Since diagnostics are performed in a passive way, i.e., without physically accessing those nodes suspected to be faulty, the decision tree deems a node unresponsive if no outgoing traffic is captured from it.

Partition. This decision examines whether a given node (or group of nodes) fails to connect to its intended destination, leading to a complete loss of data messages from the given node, and of data transmitted toward it by other nodes. This problem may arise from:

- *Joining failure*. When a node attempts to join a DODAG, it may stay silent, waiting to receive DIO messages sent by nodes from within the DODAG of interest. In the absence of DIO messages, a node may decide to send DIS messages periodically after some configurable period of time. This may be an indicator of a potential problem, possibly due to the network not being initialized (perhaps because of a failure in the network root) or the node being isolated. The former possibly may be discounted after capturing traffic from the network root or from those nodes which succeed in joining the network. The latter cause may be more difficult to diagnose, as a node may be isolated from the network due to its positioning and radio coverage, possibly as a result

of wireless propagation issues (e.g., wireless interference, obstructions from nearby objects, and so on). Another reason may be an internal issue causing the node to reboot after one or many outgoing transmissions. This possibility can be confirmed by examining the frame sequence number field in the link-layer header of the transmitted messages.

- *Single node partition*. A node that was previously connected to the network but has lost connectivity with its parent including other neighbors. In this case, if the IPv6 ND protocol is enabled, a node automatically sets its rank to a maximum value (e.g., 0xFFFF in Contiki's implementation). Otherwise, the rank computation is conducted as specified on the IETF RFC6719 [16] (see Fig. 4a).
- *Network partition*. This decision describes a group of nodes which are connected to one another, but which are disconnected (i.e., isolated) from the network root and/or data sink, thus forming a network partition. Similarly, such nodes will set their rank to a maximum value if the IPv6 ND protocol is enabled.

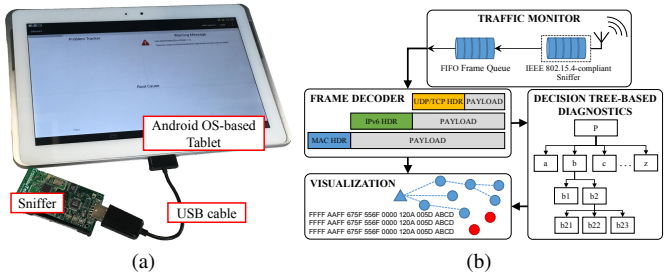
Intermittent connectivity Similar to the above, this decision is intended to find those nodes which intermittently connect to the data sink. This may span from a single node to several nodes which temporarily disconnect from the network, leading to irregular or low data message reception rate at the data sink.

In the decision tree we distinguish two types of communication issues: partition and intermittent connectivity issues. Although these issues may arise from similar root causes, the potential corrective actions to resolve these issues may be completely different. Solving a partition problem may require servicing a faulty intermediate forwarder node which was used to connect the partitioned node(s) to the rest of the network, while an intermittent connectivity problem may just require re-positioning the currently deployed forwarder nodes or deploying new ones to provide for greater path diversity and communication reliability.

IV. Implementation of Handheld Diagnostic System

Our diagnostic system is implemented in Android OS version 4.4 and runs on a Samsung Galaxy Tab 2 tablet. A Crossbow/Berkeley TelosB mote, programmed with a generic Contiki-based sniffer application, is connected to the tablet's built-in USB/UART port to passively capture WSN traffic (see Fig. 3a). The TelosB platform, which is based on the TI CC2420 radio transceiver, was chosen because of its compatibility and interoperability with low-power IEEE 802.15.4 standard-compliant radios, typically encountered in low-power IPv6 WSNs.

The software architecture of the diagnostic system is illustrated in Fig. 3b. It consists of four main components, where the fourth component is intended solely for visualization purposes. The first component is responsible for logging the captured network traffic, consisting of IEEE 802.15.4-compliant data and acknowledgment (ACK) frames. The received frames are buffered in a frame queue allocated in SRAM memory



(a) (b)

Fig. 3. Handheld diagnostic system: (a) Hardware; (b) Software architecture.

in the tablet for further processing, allowing for sufficient memory space to gracefully handle the existing traffic. The received frames are also time stamped with their time of arrival at the system and saved to a log file on an external SD card for future off-line processing if necessary.

The second component decodes and transforms the frames from the frame queue into meaningful information for analysis. It does so by extracting layer-specific header and footer information, i.e., IEEE 802.15.4 medium access control (MAC) layer, IPv6 network-layer (i.e., 6LoWPAN [19], RPL and ND protocol standards), and TCP/UDP (where applicable) transport-layer information. Relevant information includes frame/message type (i.e., whether it is a data or an ACK frame; and if the former, whether it is a RPL/ND message or a data message), frame sequence number, source-destination MAC and network addresses, and upper-layer header checksum. The latter is necessary to determine whether a data message to be sent over multiple hops is forwarded or dropped by a forwarder node in the routing path. Frames are processed in order of arrival and without making any decision about frame priority according to a first-in-first-out (FIFO) queuing policy. This second component is implemented in a modular way to allow for new communication protocols to be added.

The third component is the core of our handheld system design and implements the decision tree-based diagnostic method presented in the previous section. Progressively as new traffic is collected and analyzed, the system learns about which nodes are operational, and to which other nodes each node connects with (i.e., RPL parent-child relationships) and how frequently, such that the network topology formed by all of the inspected nodes can be reconstructed. The system informs the user of any identified problems and associated root causes only when the amount of network traffic collected is sufficient for inferring the state of the inspected nodes, including their wireless links. The network traffic necessary to provide a reliable result will depend on the type, location and extent of the problem being diagnosed. For example, a node which is suspected to be either unresponsive or isolated from the network can be easily diagnosed by physically approaching the node and then waiting for any outgoing transmissions, particularly for those sent on a regular basis, such as DIO messages. However, finding the node(s) which potentially may be causing a network partition involving several nodes may

often be more difficult because it requires more network traffic, possibly collected at different locations of the deployment site (depending on the radio coverage range of the diagnostic system) and in some cases for a prolonged period of time.

A. Estimation of link quality

We have devised a straightforward method to determine under which link quality conditions a node or group of nodes comprises a fully or partially connected (i.e., nodes that intermittently connect to the network root) network partition. This information is relevant to those tasked with commissioning and maintenance, as it permits adopting more appropriate decisions to correct communication issues (e.g., whether re-positioning or deployment of nodes is necessary).

More specifically, inspected nodes are classified according to the quality of their wireless connectivity. This is determined by calculating an estimate of the reception success ratio (RSR), i.e., the ratio of the number of messages correctly received by a node, for any existing links between nodes within communication range. To establish the relationship that allows us to estimate the RSR, we first study the performance of the wireless link between two nodes when such link is subjected to different RSRs. In order to do this effectively, we run a set of experiments using Contiki's Cooja simulation. We consider a simple network setup, as depicted in Fig. 4a, which consists of a root node and two nodes arranged on a multi-hop linear topology. Nodes use Contiki's standards-based IPv6 stack (6LoWPAN/RPL) at the network layer and ContikiMAC [20] at the link layer. For these experiments, the IPv6 ND protocol is disabled. This has been found to make little or no difference to the results, excepting when RSR approaches 0% and consequently the partitioned node adopts maximum rank and stops retransmitting until it finds a suitable parent. In this case, the use of the ND protocol greatly facilitates the diagnosis of partition problems by simply checking whether the rank of those potentially partitioned nodes is maximum. The link between nodes with identifiers 2 and 3 is configured with RSRs between 0% and 100% (e.g., 10% yields 80% unsuccessful message transmissions). All other parameter settings are set to their default values.

Figs. 4b and 4c show the number of link-layer retransmissions per unicast data message (which includes retransmissions of application-layer messages and network-layer messages such as DAO) sent by node 3 and the rank values against RSR computed for node 3, respectively. As observed, the figures also imply a linear relationship between the average number of retransmissions and the rank, which is reasonable because Contiki's implementation of RPL uses ETX as cost metric (see Fig. 4a). Results show that a low RSR causes node 3 to perform a high number of retransmissions, which give as a result a high rank value. Furthermore, both figures fluctuate significantly, becoming smoother as the RSR increases (as the link quality between nodes 2 and 3 improves).

From the above results, we have obtained and incorporated into our diagnostic system polynomial approximations that generate an estimate of the RSR by using as input parameters

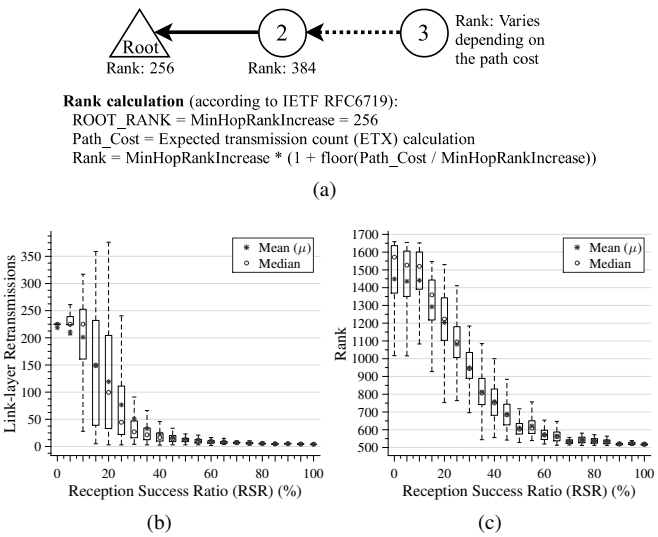


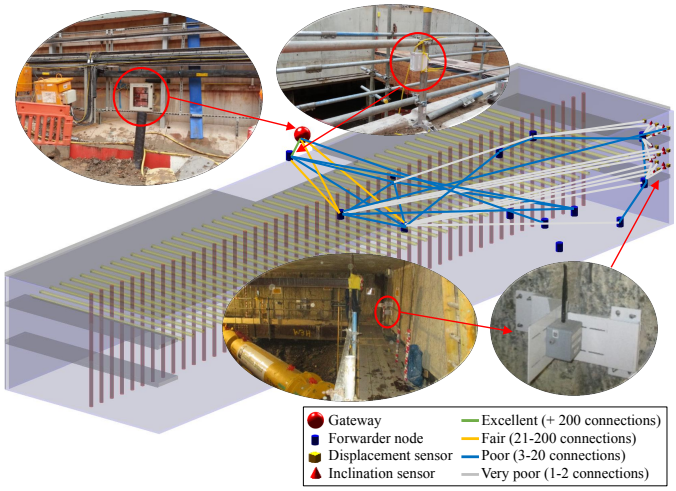
Fig. 4. (a) Three-node linear network simulated in Contiki OS/Cooja. Box plots showing (b) the number of link-layer retransmissions per unicast message (e.g., application data and DAO messages) sent by node 3, and the (c) rank values announced by node 3 through DIO messages. Different reception success ratios (RSRs) for the link between nodes 2 and 3 are considered. Experiments are repeated 10 times.

either a moving average of the captured link-layer retransmissions or the rank (announced through DIO messages). The downside of using the approximation based on link-layer retransmissions lies in its dependence on the protocol used at the link layer of the protocol stack (e.g., sender-initiated protocols such as ContikiMAC [20] send multiple copies of the message for each (re-)transmission) and also on the maximum number of retransmissions per unicast message that are configured at both the link and network layers, which may be application-specific. Studying the quality of a link using different link-layer protocols while varying the maximum number of retransmissions is left as future work. The second method of obtaining RSR, using the rank, may be used in networks where the protocol at the link layer is unknown, but this requires knowing the RPL path cost metric in use as well as obtaining an estimate of both the hop distance and path cost between the inspected node and the root node. The cost metric can be obtained from DIO messages, while the later may be derived after reconstructing the network topology. In our implementation, we have calculated different approximations to estimate RSR from the rank of a node located at different hops. Finally, where both methods to obtain RSR can be applied, we use the one which minimizes the error in the estimation.

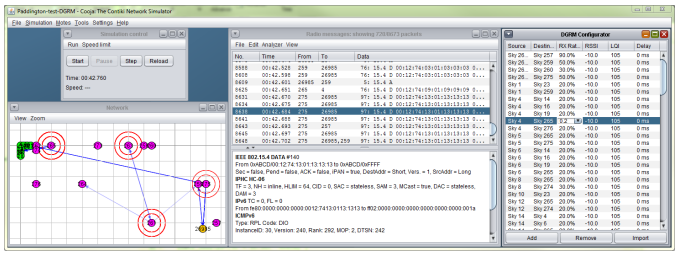
V. Diagnostics Use Case

A. Methodology

We have conducted a series of experiments in the laboratory based on previously obtained results from a six-month-long deployment undertaken in a large construction site. The site was an excavation for a new Crossrail station at Paddington, London, which took the form of an underground box (260m



(a)



(b)

Fig. 5. Deployment at Paddington station box: (a) Model of the site and WSN layout. The figure also shows the average number of connections made by each node to the gateway per day during a 5-day period six months after initial installation [13]; (b) Simulation of Paddington deployment in Contiki/Cooja.

long, 25m wide and 23m deep). The main aim of the monitoring, initiated in February 2014, was to assess deformation of three diaphragm wall panels on one of the corners of this underground box during excavation.

The WSN layout, shown in Fig. 5a, is composed of fifteen displacement sensors, twelve inclination sensors and thirteen forwarder nodes, all of them battery powered and positioned within the excavation. The WSN also contained a gateway/data sink placed outside, where a permanent power supply and good 3G signal coverage were available. After installation, the WSN experienced continuous connectivity problems that resulted in data message delivery ratios of below 10% during much of the deployment duration. Whilst the WSN performance was not satisfactory, the received data was sufficient to understand the performance of the monitored wall panels [13], [21].

Because our original deployment site was no longer available, we have designed our experiments based on extensive Contiki/Cooja simulations of a WSN as deployed in Paddington. These are intended to produce network traffic traces that can be uploaded to and analyzed in our system. In order for our simulations to be as much realistic as possible, we use real diagnostic data collected from the original deployment to simulate the actual network topology in Paddington six months after initial installation [13], as depicted in Figs. 5a and

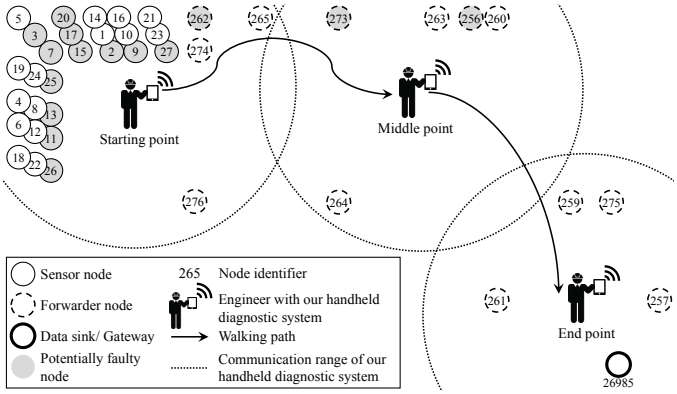


Fig. 6. Simulated use case of our handheld diagnostic system at the WSN deployment in Paddington.

6. In addition, we adopt the Directed Graph Radio Medium (DGRM) propagation model available in Cooja, which allows the simulation of networks where not all the nodes have the same communication capabilities, including transmission range, transmission success ratio and reception success ratio, as it was the case in Paddington.

The application software used in Cooja is similar to the one developed for our deployment. All nodes use Contiki's IPv6 stack (6LoWPAN/RPL/ND) at the network layer and ContikiMAC at the link layer. Each node generates non-synchronized UDP traffic flows addressed to the gateway at fifteen minutes intervals, consisting of a data message with 'sensor' measurements (where applicable), and two additional messages with network connectivity information (such as current parent node). This information is used to specify wireless links for each node in Cooja, tuning parameters such as message reception ratio and link quality indication (LQI) on a per-link basis (see Fig. 5b).

B. Diagnostic Test Results

Fig. 6 illustrates the methodology followed to examine the traffic traces from Cooja in our diagnostic system. By progressively uploading data through a Matlab script to our system, we simulate a reasonable realistic use case where one of our colleagues is dispatched to the Paddington site to investigate several potentially faulty nodes (those colored in gray) and the reasons of poor overall data reception at the gateway. For ease of simulation, we assume that the wireless reception range of the diagnostic system has a fixed range, rather than by using the DGRM propagation model employed between the nodes. In reality the diagnostic system may need to be moved closer to some nodes than to others to obtain similar results. We also choose random start times in the simulated traffic traces (e.g., after 10 minutes of simulation) to reflect the case when our colleague arrives at the site at any time of day.

The first decision of our colleague is to conduct diagnostics around the area denoted as 'starting point', where there are many sensor nodes suspected to be faulty. On average, it

is observed that the diagnostic system is able to provide an initial snapshot of the topology (just by using DIO/DAO and ND transmissions) within the first 2-3 minutes of running diagnostics, while it takes up to fifteen minutes to provide detailed information regarding the quality of each node's wireless connectivity. Although the system is agnostic to the running WSN application, the average time to obtain a complete view of the inspected nodes and their wireless connectivity is not surprising due to the configured data message transmission interval of fifteen minutes.

The results provided at the starting point indicate that sensor nodes primarily select forwarder nodes 274 or 265 as next hops, and these in turn the node 276. However, results also show that the connections between sensors and the forwarder nodes are very intermittent, exhibiting a poor RSR of below 20%, with a similar fraction of the data messages from sensors being successfully forwarded.

Because the system shows data transmissions being addressed to nodes 259, 264 and 275, our colleague decides to walk toward the point that we have denoted as 'middle point'. In addition to the above nodes, the diagnostic system finds nodes 257, 260, 261, 263, 264, 275 and 26985 (i.e., the gateway), but only shows results of RSR for outgoing connections from nodes 260, 263 (both over 70%) and 264 (intermittent connectivity to node 275, RSR is in between 30% and 50%).

Our colleague then moves to the 'end point', where the state of nodes 257, 259, 261, 275 and 26985 including the RSR for their outgoing connections can be finally obtained. In this case, only node 261 suffers from poor connectivity to the gateway, with a RSR of below 10%, while the other nodes exhibit RSR values of over 70%, except for node 257 which has a RSR of over 90%.

As expected, the diagnostic system does not show the potentially faulty nodes as neither outgoing nor incoming traffic to or from such nodes is captured. Because these experiments are performed under controlled (simulated) conditions, these nodes can be regarded as faulty. In a real deployment, our system would be unable to provide actual verification of a node that is faulty without physical access to it.

Our findings reveal that the wireless conditions around sensors are considerably problematic as compared with the connectivity of the forwarder nodes to one another and to the gateway. The computed RSR values with our diagnostic system closely match the reception ratios per link that have been configured in Cooja. This has allowed us to verify the effectiveness of our proposed method to obtain and use an estimate of the RSR for WSN diagnostics. In addition to other lessons learned during this deployment [13], the installation of additional forwarder nodes closer to the sensors seems a convenient way forward to improve the wireless connectivity in such area.

VI. Review of alternative diagnostic systems

Network diagnostics are important to allow a monitoring system based on wireless sensor networks to be maintained and to operate effectively. Network diagnostics can be performed in a number of ways [22]. They are very often embedded into the WSN gateway and managed from a central monitoring station, ideally situated at a remote location. This is an advantage from a practical point of view, as it is easier and less costly to manage and monitor the performance of a WSN remotely via the Internet.

BeanScape from BeanAir [5] and SensorConnect from LORD Sensing MicroStrain [6] are examples of readily available systems that provide maintenance and diagnostics data through the gateway. In addition to real-time visualization of data, these systems also provide information for use by the user for in-situ operational monitoring of WSN nodes, such as battery level and radio signal quality. However, their diagnosis capability is limited leaving the interpretation of such information, as well as subsequent investigation of potential network problems, to the user. Perhaps closer to our work, Sympathy [4] has the ability to diagnose failures in a node, in a routing path and in the gateway by way of a decision tree implemented at the gateway, which can distinguish whether a given failure is due to a node crashing or rebooting, or due to a connectivity issue. The main downside of systems such as BeanScape, SensorConnect and Sympathy is the difficulty to gain some deeper insight into the nature of such connectivity issues, particularly when it involves complete disconnection to one or many nodes [13]. Another downside is related to the need to program every network node to collect and report local information about its current state (e.g., battery level) and connectivity with other nodes to the gateway. This involves an increase in programming complexity, routing overhead and subsequent power consumption of WSN nodes derived from the transmission of this diagnostic information.

An alternative approach is to install a secondary independent monitoring network intended to allow checks to be carried out where required on the primary WSN, and to provide key information in the event of malfunction of the main WSN. Although network diagnostics and control of both networks is still centralized either at the gateway or at the monitoring station, nodes from the main WSN need not be programmed with additional diagnostic software as this capability resides entirely in the secondary network. Furthermore, a solution of this type can further benefit from the secondary network being of a different technology (but still with the ability to interact with the primary network) so as to reduce the likelihood of systematic errors in both networks. Two representative examples are SNIF [7], which also adopts a decision-tree approach, and Z-monitor [8]. Through the installation of sniffers alongside the network being assessed, these solutions are capable of locating a variety of root failure causes, including the potential causes of connectivity issues. However, besides the advantages arising from having a secondary network, this approach is not adopted very extensively because of the additional difficulty and labor and equipment cost of installing and maintaining two networks.

Because of the existing limitations with centralized approaches, there has been a growing interest in systems that

allow for on-site network diagnostics. The simplest and perhaps the most commonly used solution is to carry a laptop with an installed network traffic analyzer, such as WireShark [9]. However, while this is sufficient to capture and show the content of network traffic, it often requires an experienced user with a deep knowledge of WSN communication protocols to examine the captured traffic and figure out what is going on. Specifically developed for low-power IPv6 networks, Perytons [10] and Foren6 [11] provide real-time visualization of the network topology and different traffic flows, but they have very limited capability to ascertain the potential location of root causes of problems.

VII. Conclusion and Future Work

In this paper, we have presented a handheld diagnostic system for deployment and troubleshooting of IPv6/6LoWPAN/RPL networks. Our system has the ability to locate the most likely causes of problems that the network experiences while on site. Our system departs from previous work in that it can be utilized where traditional diagnosis solutions fail to successfully capture the potential root causes for encountered network problems, and without the need for running additional software for network diagnostics on the wireless nodes or installing an independent system to monitor the main network. The diagnosis engine of our system is based on a lightweight decision tree that considers the functioning of the underlying communication protocols in use by the network and the link-layer interaction between nodes. The applicability of our solution is shown by way of a use case based on previously obtained results from a real deployment on a construction site.

Some immediate future work that we plan to pursue is to use our system during commissioning of a real deployment. We also plan to add the ability to diagnose WSNs based on the IEEE 802.15.4e-2012 specification, with a major focus on, but not restricted to, its Time-Slotted Channel-Hopping (TSCH) MAC mode [23]. This constitutes a significant challenge in the design of portable diagnostic systems as it requires due consideration of new potential root causes of failure, such as network partitions because of nodes communicating in different channels, as well as new requirements for hardware and software. Finally, a direction of interest is to undertake an investigation on how network diagnostics and topology management techniques can be combined in order to enable optimal re-deployment of sensor nodes.

Acknowledgment

This research has been funded by the EPSRC Innovation and Knowledge Centre for Smart Infrastructure and Construction project (EP/K000314/1). The authors wish to thank Costain-Skanska Joint Venture (CSJV) and our industrial partner Crossrail for allowing access and instrumentation of the Paddington site referenced in this paper. Data supporting this paper is available from http://dx.doi.org/10.17863/CAM.4896.

References

[1] N. A. Bhatti, M. H. Alizai, A. A. Syed, and L. Mottola, "Energy Harvesting and Wireless Transfer in Sensor Network Applications: Concepts and Experiences," *ACM Transactions on Sensor Networks*, vol. 12, no. 3, pp. 24:1–24:40, 2016.

[2] A. S. M. Z. Kausar, A. W. Reza, M. U. Saleh, and H. Ramiah, "Energizing wireless sensor networks by energy harvesting systems: Scopes, challenges and approaches," *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 973–989, 2014.

[3] D. Rodenas-Herráiz, K. Soga, P. Fidler, and N. de Battista, *Wireless Sensor Networks for Civil Infrastructure Monitoring – A Best Practice Guide*. ICE Publishing, 2016.

[4] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '05. San Diego, California, USA: ACM, 2005, pp. 255–267.

[5] BeanAir. (2016) Beanscape® supervision software. [Online]. Available: http://www.beanair.com/wsn-monitoring-software-over.html

[6] LORD Sensing MicroStrain. (2016) Sensorconnect® sensing software. [Online]. Available: http://www.microstrain.com/software

[7] M. Ringwald, K. Römer, and A. Vitaletti, "Passive inspection of sensor networks," in *Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS'07. Santa Fe, NM, USA: Springer-Verlag, 2007, pp. 205–222.

[8] S. Tennina, O. Gaddour, A. Kouba, F. Royo, M. Alves, and M. Abid, "Z-Monitor: A protocol analyzer for IEEE 802.15.4-based low-power wireless networks," *Computer Networks*, vol. 95, pp. 77 – 96, 2016.

[9] Wireshark Foundation. (2015) Wireshark. [Online]. Available: http://www.wireshark.org

[10] Perytons. (2015) Protocol analyzer. [Online]. Available: http://www.perytons.com

[11] L. Deru and S. Dawans. (2015) Foren6: A 6lowpan diagnosis tool. [Online]. Available: http://cetic.github.io/foren6/

[12] X. Xu, K. Soga, S. Nawaz, N. Moss, K. Bowers, and M. Gajia, "Performance monitoring of timber structures in underground construction using wireless SmartPlank," *Smart Structures and Systems*, vol. 15, pp. 769–785, 2015.

[13] S. Nawaz, X. Xu, D. Rodenas-Herráiz, P. R. A. Fidler, K. Soga, and C. Mascolo, "Monitoring A Large Construction Site Using Wireless Sensor Networks," in *Proceedings of the 6th ACM Workshop on Real World Wireless Sensor Networks*, ser. RealWSN '15. Seoul, South Korea: ACM, 2015, pp. 27–30.

[14] *IETF RFC 6550 – RPL: IPv6 Routing Protocol for Low power and Lossy Networks*, Internet Engineering Task Force (IETF) Std., 2012.

[15] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. Berkeley, California, USA: ACM, 2009, pp. 1–14.

[16] *IETF RFC 6719 – The Minimum Rank with Hysteresis Objective Function*, Internet Engineering Task Force (IETF) Std., sep 2012.

[17] *IETF RFC 6206 – The Trickle Algorithm*, Internet Engineering Task Force (IETF) Std., 2011.

[18] *IETF RFC 6775 – Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)*, Internet Engineering Task Force (IETF) Std., nov 2012.

[19] *IETF RFC 4944 – Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, Internet Engineering Task Force (IETF) Std., 2007.

[20] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Swedish Institute of Computer Science, Tech. Rep. T2011:13, 2011.

[21] X. Xu, S. Nawaz, P. R. A. Fidler, D. Rodenas-Herráiz, J. Yan, and K. Soga, "Wireless sensor monitoring of Paddington Station Box Corner," in *Transforming the Future of Infrastructure through Smarter Information: Proceedings of the International Conference on Smart Infrastructure and Construction*, ser. ICSIC'16. Cambridge, UK: ICE Publishing, 2016, pp. 209–214.

[22] A. Rodrigues, T. Camilo, J. S. Silva, and F. Boavida, "Diagnostic tools for wireless sensor networks: A comparative survey," *Journal of Network and Systems Management*, vol. 21, no. 3, pp. 408–452, 2013.

[23] *IEEE Std 802.15.4e-2012 – IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) (Amendment 1: MAC sublayer)*, IEEE Std. 802.15.4e, 2012.