Demo: Deploying a Drone to Restore Connectivity in a WSN

Thuy T. Truong, Kenneth N. Brown and Cormac J. Sreenan
CTVR, Department of Computer Science, University College Cork, Ireland
Email: {t.truong, k.brown, cjs}@cs.ucc.ie

Abstract

This paper describes our demonstration of a network repair problem where a drone places a new sensor node to replace a failed node in order to heal the connectivity for a Wireless Sensor Network (WSN). It serves to show the potential of our published solutions for automated network repair when the repairing agent is a drone.

Keywords

Drone, Wireless Sensor Network, Network Repair

1 Introduction

Many applications for wireless networks will be in settings where network damage can be expected to occur, e.g. battlefield sensing, fire detection, etc. In addition, many sensor nodes are powered by batteries, and so they may fail as batteries deplete. The loss of nodes might cause network partitioning, thus leading to longer delivery delays and/or lost packets. To overcome node failure and to restore network connectivity, network repair should be initiated where we must place new nodes in the environment to restore network connectivity. Our prior published work [2] presented solutions for automated repair of damaged wireless sensor networks. In this demo we use a simple star topology to show the potential for using a drone as the WSN repair agent.

The goal of robotic network repair is to restore connectivity for a partitioned wireless network. A repairing agent is deployed to discover network damage, RF environment characteristics, and physical access constraints, and run algorithms for optimisation of radio equipment use, network service quality, and physical deployment plans. Recent research shows the potential of using drones in many areas, e.g. journalism, traffic surveillance, delivery, agriculture, etc. [1] . In this paper, we demonstrate a scenario where a drone, controlled by a Raspberry Pi, carries a new sensor node and flies to the area where an existing node has failed and tries to find a location in that area to replace that failed node.

2 System Description

The scenario is based on a star topology WSN which consists of a Personal Area Network (PAN) coordinator (PC) forming a network and four end devices (ED) associating

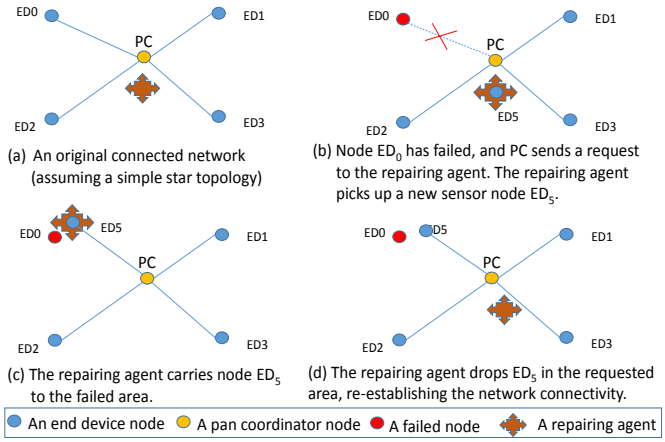

Figure 1. Scenario

with the PC to join the network (Figure 1). The communication is based on IEEE 804.15.4 standard. The EDs monitor the regions they are in, and send the detected data to the PC. The PC will show the received data on the screen. Incidents (e.g. physical breakdown or batteries depleted, etc.) might occur which cause any of the EDs fails to perform the sensing task. Because of that, the PC no longer receives data reports from those nodes, and thus the corresponding data streams disappear on the screen. At this time, the PC will send a REQUEST to the repairing agent which will place a new ED node in the failed region. In this demonstration, our repairing agent is capable of carrying only one node at a time, therefore, the repairing process will be sequential.

We build our prototype for the repairing agent combining a Raspberry Pi model B+, a Raspbee radio module, an EPM device (electromagnet permanent device) and an Iris+ drone.

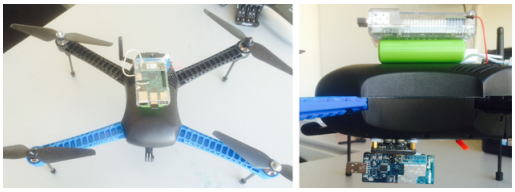

Figure 2. The repairing agent: the Iris+ drone carries the Pi on top and the EPM (with a sensor node) at bottom.

We have a Raspberry Pi attached with a Raspbee radio module to communicate with the network using IEEE 802.15.4 (Figure 3-Left). We implement new firmware to the Raspbee, forcing the Raspbee to follow IEEE 804.15.4 standard. The Raspbee communicates with the Pi via a seria UART port. With these features, the Pi can receive requests

for repairing a failed node from the sensor network and communicate with the network when repairing. The Pi is also responsible for controlling the movement of the drone.

For picking up, carrying and dropping a new node in repairing the network, we use the EPM device (Figure 3). The EPM device, which connects to the PI via the GPIO on the PI, draws the power from there and also receives on/off commands from the Pi for magnetising/demagnetising its magnets to hold/release a new sensor sensor node (for picking up/dropping down a sensor sensor node).

Figure 3. Left: A Pi with a Raspbee. Middle: An EPM device and a Tmote sky (attached with a metal plate). Right: the EPM holding a sensor node with its magnets.

3 Implementation

We implement a star topology Wireless Sensor Network with a PAN Coordinator (PC) as a sink node and four end devices (EDs), each ED will be placed in each region. The communication between the devices is shown in Figure 4.

The PC: The PC acts as a sink node, starts a network with selected PAN ID and CHANNEL and broadcasts its beacons for association. For initial setup, it keeps track of the region of each end device. When an ED sends request for association, the PC will accept the request, allocate a new short address for the ED node and then send that information to the ED. The short address is used for communication later on. The PC will receive *DATA* messages periodically from the EDs. The PC also connects to a laptop via serial USB port. It then sends the received data to the laptop which extracts the values needed and displays them on the screen. The screen shows live streaming data for each region in the network, and thus we can easily notice if an ED is missing, e.g. no data stream for the region that the missing node is in.

If the PC does not receive update from a node in a certain time, it sends a *REQUEST* message containing the node's ID and region to the Pi. After that, it might receive the *LOCATING* message from the Pi indicating that a new node is being located in the requested region. At this time, the PC will check if it receives DATA message from that node; if so, it will send *GOODLOCATION* message to the Pi. After that, if the PC receives a *DROPPING* message from the Pi (i.e. the new node has been dropped), it associates this node with the requested region and start updating the data from the node. At this point, the data stream from the requested region should be restored back on the screen.

The ED: The end device always tries to associate with the PC in selected PANID and CHANNEL network, and if SUCCESS, it will periodically sense the phenomena (light, humidity and temperature) and pack the data into an IEEE 802.15.4 frame and send to the PC.

The Pi: The Pi is attached to the drone to control its movement. Upon receiving a REQUEST message, the Pi extracts

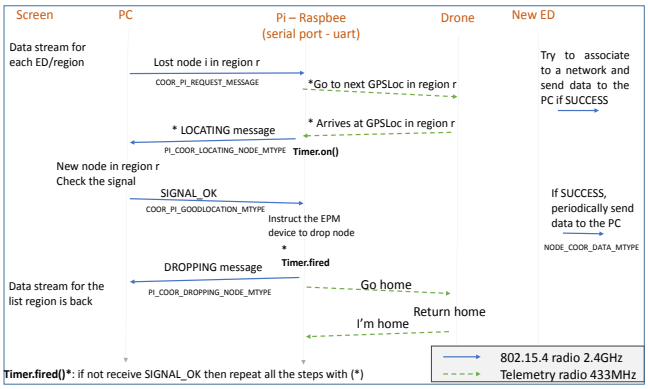

Figure 4. Communication between the devices.

the content of the message to get the requested region. It then instructs the drone to move to a GPS location in that region. For simplicity, the Pi has a list of target GPS locations in each region so that it knows exactly where to send the drone to. When the drone arrives at the first target location, the Pi will send a LOCATING message to the PC and then wait for a GOODLOCATION message from the PC. If the Pi does not receive a GOODLOCATION message after a timeout, it instructs the drone to move to the next target location, repeating the steps until it hears a GOODLOCATION. At this time, the Pi commands the EPM to demagnetise its magnets to release the carrying sensor node. It then sends a DROPPING message to the PC indicating that the node has been dropped. Finally, it instructs the drone to go home. In this topology, the Pi communicates with the PC via peer-to-peer communication using IEEE 802.15.4 technology (in a single hop) thank to the Raspbee module.

The drone: The drone always follows the instructions from the Pi. We have 6 commands for the drone: (i) *arm_and_takeoff* tells the drone to switch to GUIDED mode, do the safety check, arm the throttle and then take off to a given altitude; (ii) *moveto* moves the drone to a specific GPS location at a specific altitude; (iii) *rtl* (return to launch) commands it to return to its launch location; (iv) *setHome* tells the drone to update its home location; (v) *hold_altitude* commands the drone to hold at a given altitude; and (vi) *land* tells it to land at the current GPS location.

4 Conclusion

This demo shows a scenario in *robotic network repair* where a drone prototype system is deployed to heal connectivity in a star topology WSN for a single node failure. Future work includes a larger scale network with multiple failures including radio link and node failures.

Acknowledgment

This project is funded by the SFI Centre CTVR.

5 References

[1] M. E. Abid, T. Austin, D. Fox, and S. S. Hussain. Drones, UAVs, and RPAs. An Analysis of a Modern Technology. Technical report, Worcester Polytechnic Institue, Massachusetts, US, May 2014.

[2] T. T. Truong, K. N. Brown, and C. J. Sreenan. Multi-objective hierarchical algorithms for restoring wireless sensor network connectivity in known environments. *Ad Hoc Network*, 33(C):190–208, 2015.