Competition: Interference-Aware Multi-Channel Cross Layer Protocol for Energy-Efficient and Low-Delay Networking

Guillermo Sierra Aiello, Ilker Demirkol, Anna Calveras, Carles Gomez, Eduard Garcia
Department of Network Engineering
Universitat Politecnica de Catalunya, Spain
guillermo.sierra@alu-etsetb.upc.edu, {ilker.demirkol, anna.calveras, carlesgo, eduardg}@entel.upc.edu

August Betzler
Mobile and Wireless Internet Unit
i2CAT Foundation, Spain
august.betzler@i2cat.net

Abstract

This document aims to present the general ideas and methodology that are planned to be used for the EWSN 2016 Dependability Competition by the Wireless Networks Group of Universitat Politecnica de Catalunya. We designed, developed and conducted performance tests based on the approach detailed in this extended abstract, with the target of optimizing the metrics of each of the evaluation criteria (i.e. reliability of the sent data, end-to-end latency and energy consumption). The current approach will be evaluated and improved further until the competition day.

Categories and Subject Descriptors

C.2.1 [**Computer-communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Measurement, Performance

Keywords

Energy-efficient Networking, Internet of Things, Cross Layer Protocol, Interference-Resilient

1 Introduction

The EWSN 2016 Dependability Competition defines three types of nodes: The sensing node in proximity of a light source trying to disseminate the sudden variations in the lighting condition to the sink, through forwarder nodes. RF interference will be generated in the competition area using sensor nodes running JamLab [2].

According to EWSN 2016 Dependability Competition specifications, we evaluated several solution candidates to be resilient to the high-interference wireless sensor network environments, to achieve high packet delivery ratio, low energy consumption and low latency, which are the three performance metrics of the competition. The challenge uses off-the-shelf Maxfor MTM-5000 sensor nodes, which bring some computational and memory constraints.

IETF has developed solutions for each layer of the OSI protocol stack considering the limitations of IoT devices, resulting in the first possible solution of the use of CoAP/UDP/IPv6-RPL/6LoWPAN/IEEE 802.15.4 communication stack. Although optimizing the stack through tweaking the parameters of protocols within stack as done in [1] would provide some improved performance on default parameter settings, the specific interference-rich nature of the EWSN Dependability Competition requires exploiting the adaptive channel selection functionality, which has been also targeted by IEEE 802.15.4e through its frequency hopping feature. However, the Contiki implementations of IEEE 802.15.4e along with the IETF IoT stack do not fit the MTM-5000 node's program flash of 48KB, requiring more simpler solutions.

2 Proposed Solution

Our approach is split in two clearly disrupted parts, each with different algorithms, with different scheduled times of occurrence: i) Setup phase (Interference, network discovery, forwarding decisions), ii) a cross-layer approach as in [3] that uses the previously mapped routing. In the following subsections we detail these two approaches.

2.1 Setup Time-lapse

As revealed by challenge organizers, there will be a 20 seconds time slot available during which there will be no changes in lighting condition to be reported. Our solution takes advantage of that setup time in order to perform channel scanning for possible jammers and in a later stage this information will be exchanged between its neighbors in order to assign best channel per link basis.

A C-struct has been defined to serve as a wrapper of the info a certain node's neighbor and channel information: node ID, neighbor list, mean values of the RSSI in each of the scanned frequencies and an array of free channels in the surroundings of each node determined as detailed in the following section.

2.1.1 Interference Discovery

Using the proto-thread approach of Contiki, a scheduler thread process designates a 10 seconds time slot and dele-

International Conference on Embedded Wireless Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria

ISBN: 978-0-9949886-0-7

gates the work to another thread able to perform recursively an RSSI scan in each of the channels, in all of the nodes regardless of their role, until the timer expired. In order to save some computational resources, but above all energy and time, in each of the loops the frequency according to the channels defined by IEEE 802.15.4 (i.e. 16 different channels equally 5 MHz spaced from 2405 MHz to 2485 MHz) ended up set by using a function which uses some register variables, which writes faster on CC2420 RF chip, instead of *CC240_set_channel* function, which was found to allow 450% faster scanning.

A first approach involved the most intuitive focus which aims to obtain the mean value of the RSSI among all of the loops in each of the channels so that it can be easily determine which channels are less interfered. However both physical and simulated tests revealed unambiguously that by performing the count of the amount of times the RSSI had raised over a certain RSSI threshold, and by therefore setting it as a busy channel, is a safer, more realistic approach in order to decide if a channel experiences interference and hence should be excluded from communication.

After the first 10 loops, to safely discard the unusual behaviors at the beginning of the scan, if at least 10% of the scans for certain channel exceed the threshold, the channel will be labeled as busy channel in its link struct and ignored in the further loops, which also increases the amount of loops that may be performed within the 10 seconds reserved time slot. The mean RSSI value is also stored for each of the channels, which can be used in the following phases.

2.1.2 Joint Network Discovery and Forwarding Decisions

The scheduler thread starts another 10 seconds time slot so the *network_init* thread process is able to set the previously found least interfered channel the radio will be listening to. Then, the destination (sink) node starts propagating the observed channel interference information to its neighbors along with the rank information in all the channels, so it can assure all neighbors receive the information.

Once a node receives such packet, it evaluates the best channel for the link communication with the sender of the incoming packet by evaluating the overlapping low-interference channel lists. The node sets its rank by incrementing the minimum rank it receives from its neighbors. It also calculates the Expected Transmission Count (ETX), using the Link Quality Indicator (LQI) of the received packet (assuming similar bidirectional link qualities), and updates the ETX value received in the packet. All this information is propagated to the rest of neighbors up to the source node. Thanks to that procedure, each of the nodes will be clearly aware of the optimal route towards the destination in terms of ETX and also the channel to be used in each of the links for further transmission.

2.2 Sensing and reporting light changes

There will be three different node roles: Source, Destination and Forwarder. The source node will only send a packet, whereas forwarders only will ensure that they relay the packet towards the destination through the best route. As there will be at least 2 seconds between each change in the lighting condition, each of the nodes can sleep for 2 seconds after transmitting the packet in order to save energy.

3 Methodology

3.1 Algorithm Implementation

As previously introduced, Contiki offers events and protothreads oriented programming functions in order to manage the main execution thread either synchronously or asynchronously. Our algorithm implements in each of the nodes a scheduler thread which targets to schedule the different time slots for each of the stages and roles and also manages the events that should wake up the nodes from sleep mode with a synchronous event such as “light_change”, “PROCESS_EVENT_MSG” or different timeouts besides setting the different components in sleep mode, so that energy consumption may be kept as low as possible.

3.2 Evaluation Methodology

In order to devise and evaluate the solutions involving different network topologies and different strategies, there are several possible evaluation methodologies.

3.2.1 Simulation

An available tool is Cooja simulator, as part of the Contiki toolset. This software allows the user to establish the desired network topology, with different number of devices and surrounding objects, obstacles or interferences. It also allows collecting the performance information such as delays, errors and energy consumption, which are the target to beat within the challenge context. However simulations employ predictions and estimations on the channel behavior, the timing emulation, etc., resulting in possibly skewed results from physical executions.

3.2.2 Testbed Experimentation

Another approach to perform tests and check possible programming issues involves the use of a group of physical nodes in which each of them are separately programmed, including at least one source and one sink node and few other intermediate node in order to test the behavior of the developed algorithms. However, a larger size TelosB test bed is available in our university, consisting on a large grid topology of 6x10 TelosB boards, already used for other research purposes [1]. This test bed can be used for physical environment tests and validation of simulation results.

4 Acknowledgments

This work was supported in part by the Spanish Government's ”Ministerio de Economia y Competitividad” through project TEC2012-32531, RYC-2013-13029 and FEDER.

5 References

[1] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells. A holistic approach to ZigBee performance enhancement for home automation networks. *Sensors*, 14(8):14932, 2014.

[2] C. Boano, T. Voigt, C. Noda, K. Romer, and M. Zuniga. Jamlab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 175–186, April 2011.

[3] C.-H. Feng, Y. Zhang, I. Demirkol, and W. Heinzelman. Stateless multicast protocol for ad hoc networks. *Mobile Computing, IEEE Transactions on*, 11(2):240–253, Feb 2012.