Competition: ContikiMAC with Differentiating Clear Channel Assessment

Alex King
School of Computing and Communications
Lancaster University, United Kingdom
a.king4@lancaster.ac.uk

James Hadley
School of Computing and Communications
Lancaster University, United Kingdom
j.hadley1@lancaster.ac.uk

Utz Roedig
School of Computing and Communications
Lancaster University, United Kingdom
u.roedig@lancaster.ac.uk

Abstract

This challenge presents hazardous conditions for common Wireless Sensor Network (WSN) MAC protocols to operate in. ContikiMAC uses Clear Channel Assessments (CCA) for transmitting and receiving packets. Since CCA checks only measure the channel energy, nodes cannot differentiate a ContikiMAC sender from other interference. Coupled with the design of ContikiMAC, this ambiguity causes reduced energy efficiency, and link performance, in busy environments.

Here, we propose an extension to ContikiMAC to prevent *false wakeups*, which are the result of ambiguous CCA results. By implementing a Differentiating CCA (DCCA), which can tell apart ContikiMAC packets from other interference, false wakeups are prevented. As well, specific responses to sources of interference can be made following a channel collision, thereby potentially achieving better link performance.

1 Introduction

Wireless Sensor Network (WSN) MAC protocol designers have confronted a tradeoff, between energy efficiency - by keeping the radio powered down as much as possible, and communication reliability. Low Power Listening (LPL) MAC protocols have emerged from this, achieving a low duty cycle while still providing reliable communication. In LPL, incoming transmissions are detected by sampling the channel energy, keeping the radio otherwise powered down to conserve energy and battery life.

ContikiMAC [1] is one example of LPL, which employs Clear Channel Assessment (CCA) checks to listen for incoming packets. A CCA check cannot decode any incoming data, thus these checks can not differentiate between valid, incoming IEEE 802.15.4 packets, and other interference, such as from WiFi, Bluetooth, and microwave oven devices. A CCA check detecting such interference will indicate affirmatively to ContikiMAC, which will leave the radio powered on to receive a non-existent packet. This is known as a *false wakeup*, depicted in Figure 1a. False wakeups cause the radio to be unnecessarily powered on, listening to the channel. Since the radio is typically the greatest power-drain on sensor node hardware, this inefficiency can greatly reduce battery life [2].

As well, ContikiMAC sends packets to receivers by repeatedly sending probes during the receivers duty cycle until an acknowledgement is received. CCA checks are used between probes to detect possible collisions. Since CCA checks can not differentiate another co-located IEEE 802.15.4 node from interference, the same collision policy is applied in all cases: abort the transmission and execute a random back off. This is depicted in Figure 1b.

Solutions to the false wakeup problem have included AEDP: which calibrates the CCA energy threshold to reduce false wakeups while still supporting reliable communication [2]. ZiSense collects a signal strength trace of the channel, and searches for characteristics indicative of IEEE 802.15.4 data, which is readily distinguishable from other sources of interference [3]. Our solution incurs little additional overhead, achieves high detection accuracy, and is compatible with any MAC protocol that employs CCA checks.

2 ContikiMAC with DCCA

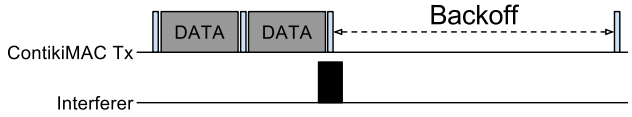
Here, we augment ContikiMAC with support for Differentiating CCA (DCCA), which allows each node to discern not only the availability of the channel, but also the nature of any current occupier of the channel. DCCA operates similarly to standard CCA, however three responses can be returned: *CHANNEL_CLEAR* - indicating a clear channel, *CHANNEL_BUSY_CMAC* - indicating another ContikiMAC sender, and *CHANNEL_BUSY_UNKNOWN* - indicating an unknown interference source. The DCCA implementation used incurs little overhead compared to standard CCA checks, and can be easily implemented on any IEEE 802.15.4-compliant radio.

ContikiMAC Receivers use DCCA to listen for incoming data. After detecting ContikiMAC data, the normal operation of ContikiMAC is preserved. Non-ContikiMAC signals, originating from interference sources nearby, are ignored. Thus false wakeups are prevented, maintaining the energy

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

ISBN: 978-0-9949886-0-7

(a) False Wakeups in ContikiMAC, caused by interference



(b) ContikiMAC response to interference, during packet transmission

Figure 1: Operation of ContikiMAC under Interference

efficient operation of ContikiMAC.

ContikiMAC Transmitters use DCCA between packet transmissions to detect collisions. Source-specific responses are employed to handle collisions. Collisions with other ContikiMAC devices result in the standard collision resolution mechanism. Other sources of interference are ignored, and ContikiMAC continues packet transmission as normal. This will result in better link performance under interference, as fewer transmission attempts are aborted.

We expect these modifications to standard ContikiMAC will result in greater link reliability between nodes in the network, which in turn will reduce latency from the source to the sink. Similarly, by reducing false wakeups, ContikiMAC should retain it's energy consumption characteristics, even in such a busy environment.

3 References

[1] A. Dunkels. The contikimac radio duty cycling protocol. 2011.

[2] M. Sha, G. Hackmann, and C. Lu. Energy-efficient low power listening for wireless sensor networks in noisy environments. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 277–288. ACM, 2013.

[3] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu. Zisense: towards interference resilient duty cycling in wireless sensor networks. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 119–133. ACM, 2014.