A Random Priority based Scheduling Strategy for Wireless Sensor Networks using Contiki

Sajid M. Sheikh, Riaan Wolhuter and Herman A. Engelbrecht
Department of Electrical and Electronic Engineering, University of Stellenbosch,
Private Bag X1, Matieland, 7602, South Africa

Keywords: Contiki, MAC, IEEE802.15.4, Priority Scheduling, Sensor Networks.

Abstract: In recent years, wireless sensor networks (WSNs) have experienced a number of implementations in various implementations which include smart home networks, smart grids, smart medical monitoring, telemetry networks and many more. The Contiki operating system for wireless sensor networks which utilises carrier sense multiple access with collision avoidance (CSMA/CA) does not provide differentiated services to data of different priorities and treats all data with equal priority. Many sensor nodes in a network are responsible not only for sending their sensed data, but also forwarding data from other nodes to the destination. In this paper we propose a novel priority data differentiation medium access control (MAC) strategy to provide differentiated services called Random Weighted Scheduling (RWS). The strategy was implemented and tested on the FIT IoT-lab testbed. The strategy shows a reduction in packet loss compared to the default CSMA/CA scheduling strategy in IEEE802.15.4 WSNs when carrying data of different priority levels.

1 INTRODUCTION

Wireless Sensor Networking (WSN) is one of the main driving forces of the Internet of Things (IoT). WSN have been deployed in a number of different environments which include smart home networks, smart health, smart transport, smart educations and other IoT applications. All these networks carry heterogeneous data with different levels of priority (Glaropoulos et al., 2014). A WSN typically consists of a large number of low cost and low power, multifunctional sensor nodes. Sensor nodes are normally equipment with different types of sensors depending on the parameter they will be measuring, different embedded microprocessors, different operating systems, and different radio transceivers (Jun Zheng, 2009).

WSN embedded operating systems include among others TinyOS, Contiki, MANTIS, T-Kernal, LiteOS and Nano-RK. The Contiki operating system is one of the most popular operating systems for embedded systems and IoT applications (Glaropoulos et al., 2014). Contiki utilises the CSMA/CA scheduling strategy. The standard CSMA/CA mechanism does not provide any data differentiation services to improve the quality of service for time critical events such as alarms that have a higher priority than normal data in a network (Koubaa et al., 2006). CSMA/CA treats all data with equal priority in a first in first out (FIFO) manner.

In this paper we propose a novel scheduling strategy that has been developed under the Contiki operating system and implemented and tested on the FIT IoT-lab testbed. Our proposed scheme has multiple queues for data of different priority and smaller values of back-off exponent (BE) and contention window (CW) are assigned to higher priority data queues to gain access to the channel faster than the lower priority queues. The data from the different queues gain access to the channel by randomly selecting a queue for transmission based on weights assigned to the different queues.

The rest of this paper is organised as follows. Section 2 presents the related work. Section 3 presents an overview of CSMA/CA in the IEEE802.15.4 standard. In section 4, we present the proposed RWS scheduling strategy. Section 5 presents an overview of the Contiki operating system. Section 6 gives a brief overview of the Testbed implementation. Section 7 presents the results and in section 8 we conclude the paper.

Sheikh, S., Wolhuter, R. and Engelbrecht, H.
A Random Priority based Scheduling Strategy for Wireless Sensor Networks using Contiki.
DOI: 10.5220/0005949301210128
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 6: WINSYS*, pages 121-128
ISBN: 978-989-758-196-0

2 RELATED WORK

A large amount of work has been carried out to optimize energy usage in WSN communications. Limited work has been done to develop priority based scheduling strategies in WSNs. A priority scheduling scheme is proposed in (Sun and Xu, 2010) which is based on queue management and MAC layer back off. When a packet arrives at the node, it gets placed at the end of the queue in a FIFO queue that does not differentiate the priorities of packets. DRAG (Sun and Xu, 2010) is a priority-based queue management policy to provide priority guarantee. The packet gets placed in the queue in an appropriate place relative to the previously sorted packets instead of placing the packet at the end of the queue. Furthermore, the strategy selects a high priority packet to send.

Other MAC layer priority based scheduling strategies have also been proposed in literature such as Q-MAC and PRIMA (Barua et al., 2014). With Q-MAC, a queuing model with multiple queues for different priority levels packets is proposed. The strategy tries to minimize energy consumption and provides QoS for intra-node and inter-node scheduling. The intra-node scheduling places data into the different priority queues, while the inter-node scheduling provides channel access to reduce energy consumption by reducing collisions. Five bits of information are added to the existing packets of which 2 are for identification of the packet type and the other 3 are for sensing data. Packets that have travelled more hops have a higher priority. In Q-MAC, the queue architecture consists of five queues with one specified as an instant queue. PRIMA is also an energy efficient MAC protocol which minimizes the idle listening periods by making nodes that have no data to send to go to a sleeping state. PRIMA also employs multiple queues where data is classified and placed in the respective queues. Queues with higher priority are given first access to the channel compared to the low priority data.

To provide service differentiation to rate sensitive applications, (Na, 2011) proposes a Multi-rate Service Differentiation (MSD) model. This model is implemented with two components namely a Virtual Medium Access Control (VMAC) and the Adaptive Back-off Window Control (ABWC). The VMAC is the priority queue mechanism that adapts its back-off based on the conditions of the network. With VMAC, it is possible that more than one data packet collided with each other if they finish the back-off period at the same time. A Virtual Collision Avoidance Control (VCAC) is designed to address this situation which adjusts the back-off times of data frames. In (Koubaa et al., 2006), a mechanism that tunes the existing parameters of CSMA/CA for data of different priority is proposed. These include BE and CW values. This strategy differentiates between data traffic and command traffic in a network. Command traffic are given higher priority by assigning smaller BE and CW values.

The above priority strategies are implemented in sensor networks working on the IEEE802.15.4 standard. In the IEEE 802.11e standard, a contention based strategy called enhanced distributed channel access (EDCA) is used to provide differentiated services for data of different priority levels. With the default EDCA strategy, an unfairness problem is known to exist between higher and lower priority data classes as higher priority data can starve lower priority data (Choi et al., 2008; Kuppa and Prakash, 2004; Tseng et al., 2007). EDCA introduces the concept of access categories (AC) for data types and consists of four ACs. Each AC has specific parameters associated to its priority class such that the higher probabilities ACs have a higher probability of medium access (Poonguzhali, 2014).

Q-MAC is a complex strategy that introduces extra overhead in the network by the introduction of 5 extra bits added to every message designed for energy conservations. There are many applications such as smart home networks, smart grids, smart medical monitoring and telemetry networks where some nodes can be designed to act as backbone nodes and relay information from these clusters to the destination. In our proposed scheme, we use the same concept as Q-MAC by having multiple queues for data of different priority and we assign smaller values of BE and CW for higher priority data. However, the design does is not concerned with energy conservation and scheduling to reduce energy wastage and therefore, we do not compare its performance to Q-MAC.

3 OVERVIEW OF CSMA/CA IN THE IEEE 802.15.4 STANDARD

The popularity and features of machine-to-machine communications and the Internet of things (IoT), have resulted in a wide areas of research leading to development a low-power, low-rate, and low-cost wireless system. The IEEE 802.15.4 standard has become a standard for low rate wireless personal area network (LR-WPAN) communications (Hwang and Nam, 2014). The IEEE 802.15.4 standard which operates at the link and physical layers is designed for simple, low data rate, low-power and low-cost wireless communication with wireless personal area networks (WPANs). It is implemented in wireless sensor networks. The unlicensed Industrial, Scientific and Medical (ISM) band that operates worldwide with this technology is the 2.4 GHz ISM band (Petrova et al., 2006). In this band of 2.4 GHz, the ISM offers 16 channels with a data rate of 250 kbps (Collotta et al., 2013). Wireless data exchange is done through the direct sequence spread spectrum (DSSS) modulation scheme (Petrova et al., 2006). In our study we use a radio model that also uses the 2.4 GHz ISM band.

According to this standard, a node can optionally operate in beacon-enabled mode or non beacon enabled mode (Collotta et al., 2013). In this section, we present a brief overview of the beacon enabled mode which is based on the slotted mode; and the non-beacon enabled mode CSMA/CA mechanism of the IEEE 802.15.4 standard which is based on the un-slotted mode. Our work is based on the un-slotted, non-beacon enabled mode CSMA/CA mechanism to access the channel and transmit data.

3.1 Beacon Enabled Mode

For the slotted mode, the slots are aligned with the beacon frames sent periodically by the Personal Area Network (PAN) coordinator. With the un-slotted mode, there are no beacon frames (Kim et al., 2007). The principle of operation of this standard depends on beacon messages in the form of superframes regularly sent from a PAN coordinator. The MAC superframe structure is shown in fig 1. The time between the beacons is split into 16 slots. The superframe consists of an active period with the Contention Access Period (CAP) and CFP (Contention Free Period), and an inactive period. The CFP consists of Guaranteed Time Slots (GTS) which are allocated to support QoS such as real-time applications (Youn et al., 2007). In the CFP region, nodes can obtain access to the medium without collisions. In the inactive period the radio interface can be put in a low energy consumption status in order to improve energy savings (Collotta et al., 2013). GTS are provided by the PAN coordinator for nodes that need to transmit data within a certain time. Nodes access the CAP using CSMA/CA. In this mechanism, a node that wants to send data first senses the medium after a random number of back-off periods. If the medium is free the data is transmitted, otherwise a back-off is performed. There are seven GTS slots that can be accommodated in one frame. There are limitations

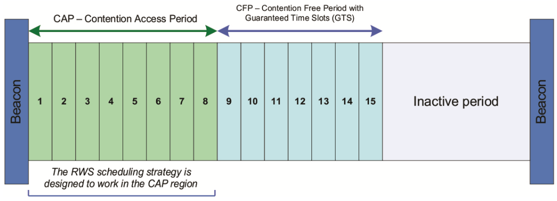


Figure 1: The MAC super frame structure in the IEEE 802.15.4 standard.

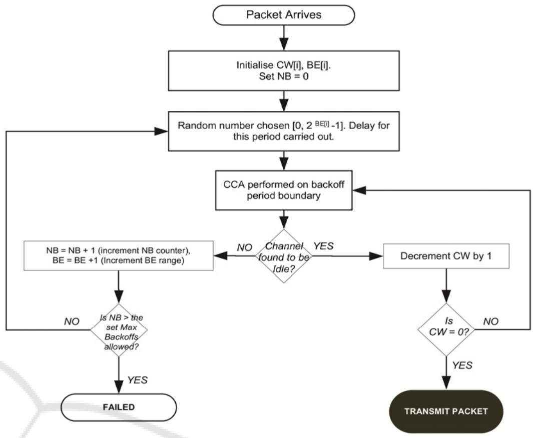


Figure 2: The slotted CSMA/CA mechanism in IEEE 802.15.4.

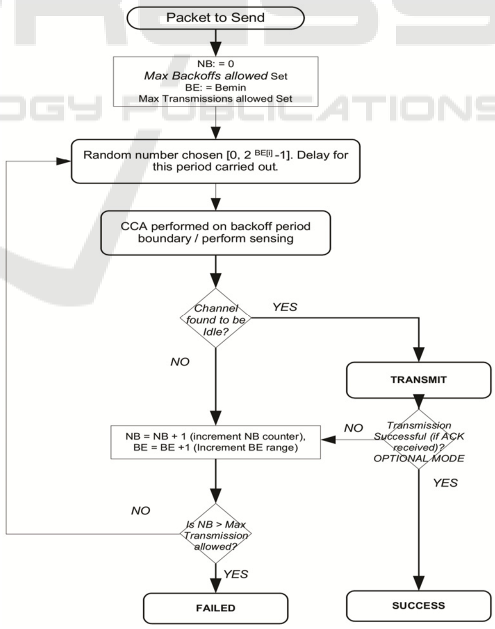


Figure 3: Flow chart for the non-beacon enabled unslottted mode for the CSMA/CA mechanism with ACK in the IEEE 802.15.4 standard.

with GTS as it can only support a limited number of nodes and does not provide any method to support QoS in the CSMA/CA mode (Youn et al., 2007).

An overview of CSMA/CA in slotted mode is presented in fig 2. NB denotes the number of times that the algorithm is required to back-off due to the medium being busy during channel assessment. CW is the contention window which is the number of back-off periods that need to be clear of channel activity before the packet transmission can be started. BE is the back-off exponent which is the number of back-off periods that a device should wait before attempting to assess the channel.

When a packet arrives, NB, BE and CW are initialized (ie. $NB = 0$, $CW = CWinit = 2$, $BE = 2$ or $BE = min(2, macMinBE)$ where *macMinBE* is the default minimum BE value). After initializing of the variables, the back-off period is started which is chosen by a random number generated in the range of $[0, 2^{BE} - 1]$. When this back-off has expired, the algorithm then performs one Clear Channel Assessment (*CCA)* to verify if the channel is busy or free. If the channel is found to be busy, the *CW* is again initialized to *CWinit* = 2, the *NB* and *BE variables* are incremented by one. If the channel is found to be free (idle), the *CW* is decremented by one. The CCA process is than repeated until the *CW* value becomes 0. After this the data is transmitted. This mechanism ensures that at least two *CCA* operations are performed to prevent potential collisions (Youn et al., 2007). If the channel is busy, both values of NB and BE are increased by one. BE cannot exceed the set macMaxBE having the default value 5 and CW is reset to 2. If NB becomes greater than the set maximum back-offs allowed, the algorithm terminates with a channel access failure status. This failure will be reported to the higher protocol layers, which decide whether or not to attempt the transmission as a new packet again (Kim et al., 2007).

3.2 Non-Beacon Enabled Mode

Just like in the beacon enable mode, when a packet arrives, the number of back-offs (NB) and the back-off exponent (BE) are initialized. After this initialization of the variables, the back-off period is started which is chosen by a random number generated in the range of $[0, 2^{BE} - 1]$. Initially, BE is initialized to BEmin which is 3 by default. BEmax is 5 by default. When this back-off has expired, the algorithm then performs one *CCA* to verify if the channel is busy or free. If the channel is found to be busy, the *NB* and *BE variables* are incremented by one. The procedure is repeated until NB is less than the set maximum allowed transmissions. After the maximum Transmissions allowed set + 1 unsuccessful attempts to access the channel, the packet is dropped. If the channel is found to be free (idle), a transmission takes place.

In the IEEE802.15.4 standard, the acknowledgement (ACK) mode to transmitted packets is optional unlike in the IEEE 802.11 standard. It is an optional feature as it can increase network overhead and have an effect on the achievable throughput of the network. If ACK mode is enabled, for any transmission that does not receive an acknowledgment, the NB and BE values are increased. If NB becomes greater than the set Max Transmissions allowed, the algorithm terminates with a channel access failure status (Kim et al., 2007). Fig 3 presents the flowchart for the operation of CSMA/CA in non-beaconed un-slotted mode.

4 THE PROPOSED RANDOM WEIGHTED SCHEDULING (RWS) STRATEGY

This section presents an overview of the proposed strategy. In each node three queues are created. These are for high, medium and low priority data. In the packet, a data field is created of 2 bits which carries information on the priority of the packet. Using this information, data is placed in either one of the 3 queues depending on the priority set in the packet header. The priority field in the packet is shown in fig 4. When a node has data to transmit and more than one queue has data, a selection process is followed. If only one queue has data, then the packet from that queue is selected for transmission without the need to follow a selection process. However, the BE, CW and NB processes are carried out after the selection process as is in the un-slotted non-beaconed based CSMA/CA. With our strategy we assign smaller values of BE and CW for higher priority data to allow the higher priority data to gain access to the channel faster than the lower priority data. The queue selection followed when two or more queues have data is as follows:

1. Probability weights are initially assigned to each data priority queue.
2. The strategy determines the size of the individual queues. If all of the queues have data, the original assigned weights in stage 1 are used. If all the queues do not have data, then the weight of the queues with data are added and then the weights of the queues with data are normalised and assigned new weights. The queues with no data are assigned a weight of zero i.e. 0.

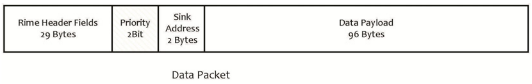


Figure 4: Integration of priority information in the packet.

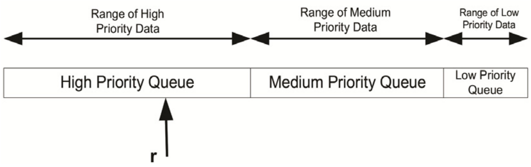


Figure 5: Queue Selection process in RWS.

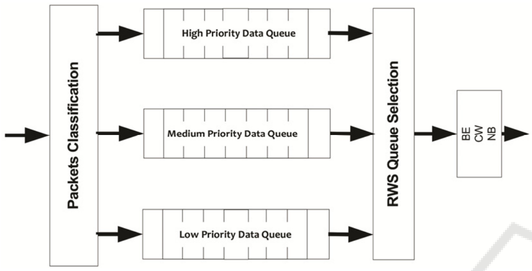


Figure 6: Proposed RWS strategy mechanisms.

3. The range values for each of the priority data classes are assigned over a scale with the high priority data being first on the scale, then medium priority data and lastly low priority data.
4. A random number is then generated in the range of 0 to the maximum scale value. A packet is chosen for transmission from a queue from which the number generated falls in its range as shown in fig 5.
5. After the packet is selected, the BE, CW and NB processes are followed as stated earlier. The complete scheduling strategy is shown in fig 6.

5 CONTIKI

The scheduling scheme proposed in this paper was implemented in Contiki as Contiki does not have a priority based scheduling mechanism for data of different priority. In this section we give a brief overview of this Contiki operating system which is an open source operating system.

Contiki is implemented in the C language developed at the Swedish Institute of Computer Science (SICS) (Dunkels, 2004); (Networks, 2011). Contiki has an event-driven kernel and follows a linear programming style which was used for the programming in this work. The Contiki protocol stack is designed for resource-constrained devices with constraints on memory and processing power

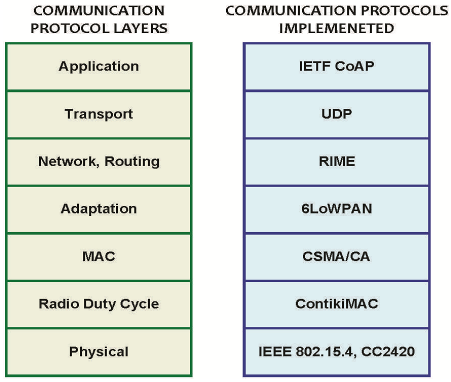


Figure 7: Contiki Layer Model.

(Colitti et al., 2009). It supports IPv6, RPL routing protocol for low-power and lossy networks, Rime and the Constrained Application Protocol (CoAP), making it suitable to develop IoT applications (Glaropoulos et al., 2014). Compared to many other closed source firmware operating systems implemented in hardware, Contiki is open source as stated earlier. We therefore, used Contiki in our test bed implementation as it allows us to use, modify and make additions to this operation system.

The Contiki OS provides two communication stacks namely uIP and Rime. uIP is a TCP/IP stack that makes it possible for Contiki to communicate over the Internet. Rime is a lightweight communication stack designed for low-power radio communication. Rime is a custom lightweight networking stack. It provides primitives for single-hop and multi-hop (mesh) communication (Networks, 2011). In our study, the Rime communication stack for multi-hop communication was used as the other layers are less detailed. Fig 7 presents the communication protocol stack used in our study. Our scheduling strategy is implemented at the MAC layer as an enhancement to CSMA/CA.

6 TESTBED IMPLEMENTATION

The FIT IoT-Lab testbed was used to implement and test the RWS scheduling strategy. The FIT IoT-Lab is a very large scale open WSN test bed at INRIA, France. This test bed allows for testing of scalable protocols and applications on this large scale test bed (Rosiers et al., 2011); (Inria, 2016).The implementation and testing of the scheduling strategy was done on nodes in Lille and Grenoble. The strategy was implemented on M3 nodes which has a 32-bit ARM cortex micro-controller, high performance and uses a 2.4GHz radio interface.

Table 1: RWS parameters.

Traffic Type	BE	CW_{min}	CW_{max}
High Priority Data	4	7	15
Medium Priority Data	5	15	31
Low Priority Data	10	31	1023

Table 2: Data Transmission Test Cases.

	High Priority Data	Medium Priority Data	Low Priority Data
Case 1	60 packets/sec	60 packets/sec	60 packets/sec
Case 2	60 packets/sec	120 packets/sec	120 packets/sec
Case 3	120 packets/sec	60 packets/sec	120 packets/sec

The scheduling strategy is mainly developed for backhaul nodes that will carry data of different priority in a multi-hop fashion until it reaches its destination. The M3 nodes were chosen over other available test bed nodes such as the WSN430 and A8 nodes as they have much higher processing power, which is needed in backbone WMN nodes. The nodes that were chosen as one hop away were spaced 4.8m apart. The mesh network was setup so that communication with the receiver takes place in multiple hop fashion by limiting the transmission range of the nodes. There are two ways of limiting the transmission range. These are being either decreasing the transmission power, or by setting a minimal energy level for the packet reception. The range of the nodes was limited so that it can only communicate with its 1 hop neighbour.

The default CSMA/CA scheduling strategy in Contiki works in a FIFO fashion and does not differentiated packets of different priority. Our RWS strategy was developed on top of this by introducing 3 queues in the nodes, one for each priority level. An application was written that generates packets of 100byte of different priority levels at the transmission rates of the different test cases. Packets with the fields as shown in fig 5 were created. The application at each node records the number of packets sent and number of packets received. Before implementing the scheduling strategy and application to generate packets of different priority level, the codes written in C were tested in the Cooja simulator on Tmote Sky nodes. It was compiled for the actual test bed nodes and then implemented on the test bed.

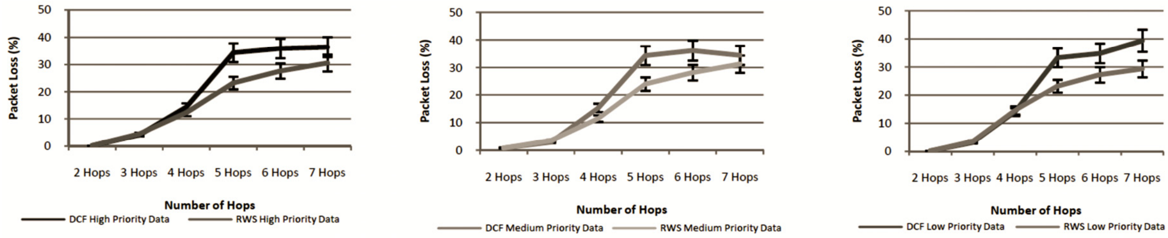
Since the strategy is intended to be used in backhaul networks receiving a high number of packets from the different network domains, the parameters for BE, NB and CW were adjusted to match those of the EDCA strategy in the IEEE802.11 standard. The modifications therefore made to the CSMA/CA in the IEEE802.15.4 were as follows:

1. The acknowledgment mechanism was activated to receive acknowledgment messages for any successful transmission as in the IEEE 802.11 CSMA/CA; 2. The maximum transmission allowed value was set to 7 as is the case with IEEE802.11g; 3. The values of BEmin and BEmax were changed such that the CW size will be the same as in the EDCA based on the CWmin and CWmax values. Table 1 presents the default parameters used for high, medium and low priority data to match those used in EDCA. In the IEE 802.11 standard, the CW length is basically the number of back-off countdown periods that need to sense the channel idle before a transmission attempt can be made while in the un-slotted IEEE802.15.4, it is the duration that the back-off waits before performing CCA. Channel activity is only performed at the CCA stage; 4. For Distributed Coordination Function (DCF), BE is set to 10 as BE of 10 equals a CW size of 1023. The retransmission limit was set to 7.

The proposed RWS strategy was tested against CSMA/CA in many test cases. Three test cases are presented in this paper as shown in table 2. The packets are transmitted at different rates of high and low combinations for number of hops from 1 to 7 hops. Each intermediate hop forwards data as well as transmits its own data generated at the rates stated in table 2. The chain topology was used for the implementations.

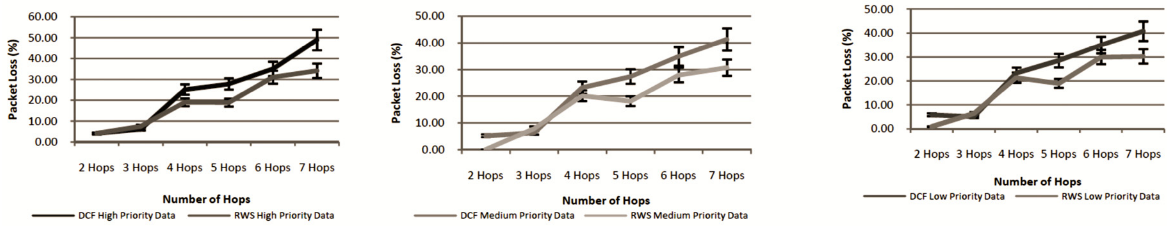
7 RESULTS

The packet loss results for the different test cases with the default CSMA/CA and the proposed scheduling strategy are shown in figs 8 to 10. In the graphs we call CSMA/CA as DCF as it operates like the DCF in the IEEE802.11 standard, with all data of any priority level treated in a single queue, FIFO manner. The results are those obtained from the real test bed implementation as such the conditions of the channel can change over time depending on the environment. The performance of CSMA/CA depends also on the value chosen for the back-off which is randomly selected. For any two test bed tests carried out, the exact conditions might not be the same as the number generated might be different which has an effect when the packets are transmitted to the next hop as well as the link conditions. The proposed scheme also largely depends on a random



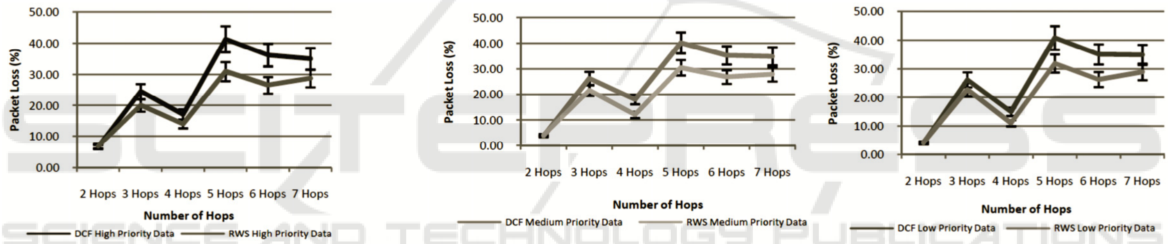
(a) High Priority Data (b) Medium Priority Data (c) Low Priority Data

Figure 8: Packet loss in test case 1.



(a) High Priority Data (b) Medium Priority Data (c) Low Priority Data

Figure 9: Packet loss in test case 2.



(a) High Priority Data (b) Medium Priority Data (c) Low Priority Data

Figure 10: Packet loss in test case 3.

number generated to choose which queue must transmit its data. To keep the conditions of the channel the same, both tests for each hop number were run immediately one after the other for CSMA/CA and RWS for the same hop number to make the comparison approximately the same.

For all the three test cases, the tests were run for five minutes each. Less packet loss can be observed for RWS over CSMA/CA in nearly all the test cases as the number of hops increases. The performance at 2 or 3 hops is approximately the same as for the original CSMA/CA technique. The performance improvement is mainly observed at hops more than 3. There is also more packet loss observed at higher loads (case 3), compared to lower loads (case 1). When the data loads are the same, a higher packet loss for the lower priority data is observed. All these test bed results presented in this paper conform to the 95% confidence levels.

8 CONCLUSIONS

A novel scheduling strategy for networks carrying data of different priority has been developed and implemented in the Contiki operating system. Contiki is an open source operating system and therefore, we modified the existing codes to implement our strategy. The Rime protocol communication stack was used as the other layers are light weight and this helps to ascertain the performance of the proposed scheme.

The proposed strategy has shown a reduction in packet loss as the number of hops is increased for most of the test cases implemented over the FIT IoT-lab test bed. The assessment of the performance of the strategy by means of a live test bed is more accurate, as opposed to testing by simulation only. This is clearly beneficial in terms of confidence in the success of any future implementations. This work is important in view of the rapid IoT and smart application implementations.

ACKNOWLEDGEMENTS

The authors will like to thank the reviewers for their comments; INRIA (France) and Nathalie Mitton for letting us use the FIT IoT testbed; and Viktor Toldov for assisting with the use of the test bed. This research was supported by the University of Botswana and the South African National Research Foundation (NRF) under the THRIP project TP13081327740.

REFERENCES

Barua, S., Afroz, F., Islam, S. S., Ahmed, A. U., Ghosal, P., Sandrasegaran, K. (2014). Comparative study on Priority Based QoS Aware MAC Protocols for WSN. *International Journal of Wireless & Mobile Networks (IJWMN)*, *6*(5), 175–181.

Choi, J., Yoo, J., Kim, C. (2008). A Distributed Fair Scheduling Scheme with a New Analysis Model in IEEE 802.11 Wireless LANs. *IEEE Transactions on Vehicular Technology*, *57*(5), 3083–3093.

Colitti, W., Steenhaut, K., Lemmens, B., Borms, J. (2009). Simulation Tool for Wireless Sensor Network Constellations in Space.

Collotta, M., Scatà, G., Pau, G. (2013). A Priority-Based CSMA/CA Mechanism to Support Deadline-Aware Scheduling in Home Automation Applications Using IEEE 802.15.4. *International Journal of Distributed Sensor Networks*, *2013*.

Dunkels, A. (2004). Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. *In Proceedings of the First IEEE Workshop on Embedded Networked Sensors, Tampa, Florida, USA*.

Glaropoulos, I., Vukadinovic, V., Mangold, S. (2014). Contiki80211: An IEEE 802.11 Radio Link Layer for the Contiki OS. *IEEE International Conference on High Performance Computing and Communications (HPCC)*, 621–624.

Hwang, K., Nam, S. (2014). Analysis and Enhancement of IEEE 802.15.4e DSME Beacon Scheduling Model. *Journal of Applied Mathematics*, *2014*.

Inria. (2016). FIT IoT-Lab. Retrieved from https://www.iot-lab.info/hardware/

Jun Zheng, A. J. (2009). *Wireless Sensor Networks, A Networking Perspective*. John Wiley & Sons.

Kim, E., Kim, M., Youm, S., Choi, S., Kang, C. (2007). Priority-based service differentiation scheme for IEEE 802.15.4 sensor networks. *International Journal of AEU of Electronics and Communications*, *61*, 69–81.

Koubaa, A., Alves, M., Nefzi, B., Song, Y.-Q. (2006). Improving the IEEE 802.15. 4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks. *Workshop on Real Time Networks RTN*, 1–6.

Kuppa, S., Prakash, R. (2004). Service differentiation mechanisms for IEEE 802.11 based wireless networks. *Wireless Communications and Networking Conference*, *4*, 796–801.

Na, C. W. (2011). *IEEE 802.15.4 Wireless Sensor Networks: GTS Scheulding and Service Differentiation*.

Networks, S. (2011). *Comparison of Simulators for Wireless Sensor Networks*.

Petrova, M., Riihij, J., Petri, M. (2006). Performance Study of IEEE 802. 15. 4 Using Measurements and Simulations. *IEEE Wireless Communications and Networking Conference (WCNC)*, *1*, 487–492.

Poonguzhali, A. (2014). Performance Evaluation of IEEE 802. 11e MAC Layer Using Cell Processor. *International Journal of Sceintific and Technology Research*, *3*(1), 255–261.

Rosiers, C. B., Chelius, G., Ducrocq, T., Fleury, E., & Vanda, J. (2011). SensLAB : a Very Large Scale Open Wireless Sensor Network Testbed, *2011*.

Sun, G., & Xu, B. (2010). Dynamic Routing Algorithm for Priority Guarantee in Low Duty-Cycled Wireless. *Springer-Verlag Berlin Heidelber LNCS*, 146–156.

Tseng, K.-N., Wang, K., Shih, H.-C. (2007). Enhanced Fair Scheduling for IEEE 802.11e Wireless LANs. *Journal of Information Science and Engineering*, *1721*, 1707–1721.

Youn, M., Lee, J., Kim, Y. (2007). IEEE 802.15.4 based QoS support Slotted CSMA / CA MAC protocol. In *International Conference on Sensor Technologies and Applications* (pp. 113–117).