

A Traffic Differentiation Add-On to the IEEE 802.15.4 Protocol: implementation and experimental validation over a real-time operating system

Ricardo Severino¹, Manish Batsa¹, Mário Alves¹, Anis Koubâa^{1,2},

¹ IPP-HURRAY! Research Group, Polytechnic Institute of Porto, School of Engineering (ISEP-IPP)
Rua António Bernardino de Almeida, 431, 4200-072 Porto, Portugal

² Al-Imam Muhammad Ibn Saud University, Computer Science Dept., 11681 Riyadh, Saudi Arabia
{rars, maba, mjf, aska}@isep.ipp.pt

Abstract

The IEEE 802.15.4 is the most widespread used protocol for Wireless Sensor Networks (WSNs) and it is being used as a baseline for several higher layer protocols such as ZigBee, 6LoWPAN or WirelessHART. Its MAC (Medium Access Control) supports both contention-free (CFP, based on the reservation of guaranteed time-slots GTS) and contention based (CAP, ruled by CSMA/CA) access, when operating in beacon-enabled mode. Thus, it enables the differentiation between real-time and best-effort traffic. However, some WSN applications and higher layer protocols may strongly benefit from the possibility of supporting more traffic classes. This happens, for instance, for dense WSNs used in time-sensitive industrial applications. In this context, we propose to differentiate traffic classes within the CAP, enabling lower transmission delays and higher success probability to time-critical messages, such as for event detection, GTS reservation and network management. Building upon a previously proposed methodology (TRADIF), in this paper we outline its implementation and experimental validation over a real-time operating system. Importantly, TRADIF is fully backward compatible with the IEEE 802.15.4 standard, enabling to create different traffic classes just by tuning some MAC parameters.

1. Introduction

In the last few years, wireless networking communities have been directing increasing efforts in pushing forward *anywhere and anytime* distributed computing systems. These efforts have lead to the emergence of smart device networking, including Wireless Sensor Networks (WSNs), which represent enabling infrastructures for these new classes of large-scale networked embedded systems. However, WSNs system designers must fulfill the Quality-of-Service (QoS) requirements imposed by the applications (and users) for these to become a reality.

Although some WSN applications do not impose stringent timing requirements on data delivery, like environmental monitoring or precision agriculture, there are a number of other applications in which timeliness is of great importance. It is the case of most industrial automation and process control applications, in which

computations and communications must not only be logically correct but also be produced on time.

In this line, the standardization efforts of the IEEE Task Group 15.4 have contributed to solve this problem by the definition of the IEEE 802.15.4 protocol for Low-Rate, Low-Power Wireless Personal Area Networks (WPANs) [1], which is being used as an enabling technology to support other protocols such as ZigBee [1], 6LoWPAN [3], or WirelessHART [4]. This is partially due to the great potential of this protocol for flexibly fitting the different requirements of many WSN applications by adequately setting its parameters.

In beacon-enabled mode, this standard provides two mechanisms (Figure 1): (1) slotted CSMA/CA as a Medium Access Protocol in the Contention Access Period (CAP) and (2) Guaranteed Time Slots (GTS) in the Contention Free Period. The GTS mechanism enables a deterministic access to the medium but it has some limitations.

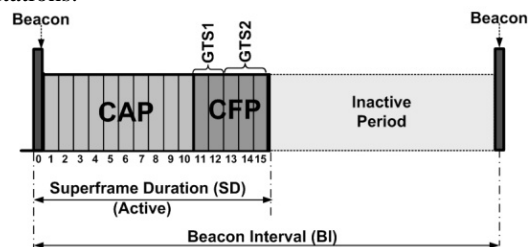


Figure 1 - IEEE 802.15.4 Superframe

The first limitation concerns the restriction on the distribution and amount of traffic that can avail this service. In a superframe, a maximum of seven GTS slots can be allocated, implying that in each cluster (PAN) a maximum of seven nodes can have guaranteed slots in any superframe. The remaining nodes may only transmit in the CAP, without any QoS support.

Second, GTS can only provide guaranteed services in bursts, limiting any node to the length of the slot allocated to it. This does not provide an optimum solution if the messages requiring QoS support are evenly distributed over time.

Third, even in applications where the limited number of GTS slots can be considered sufficient, the allocation must be preceded by an allocation request message transmitted in the CAP, and since collisions may occur, the request

may fail, delaying its service. The same problem applies to other protocol command units.

Therefore, network management (e.g. GTS allocation requests, alarms, network management commands, association commands), are more critical than regular data frames. Failing to cope with this may result in unfairness and degradation of the network performance, particularly for high traffic loads. In this line, these critical messages, require that QoS support be extended to the CAP.

Moreover, the GTS mechanism may also face coexistence problems since other wireless networks operating in the same frequency range (Bluetooth or IEEE 802.11) are completely unaware of the time slot allocations made at the IEEE 802.15.4 superframe. This turns the GTS approach worthless in the presence of collisions.

Therefore, while GTS is considered a good solution for the QoS requirement of the low-rate WPAN applications (for which IEEE 802.15.4 was originally designed), the requirements of dense sensor networks (especially at high and distributed loads) demand a more flexible mechanism.

This paper builds upon a previously proposed [5] set of mechanisms to provide QoS to the CAP (demonstrated through simulation), and describes its implementation and experimental validation.

We show that these mechanisms can easily provide increased QoS to higher priority messages, requiring only minor add-ons and ensuring backward compatibility with the IEEE 802.15.4 standard protocol.

The integration of these mechanisms in IEEE 802.15.4 is relevant for leveraging its use in time-sensitive WSN applications. TRADIF can also enrich future versions and amendments to the standard (e.g. IEEE 802.15.4e [6]), which aims at enhancing and add functionalities to the 802.15.4-2006 MAC for industrial applications.

The rest of the paper is organized as follows. Section 2 presents an overview on the related work concerning traffic differentiation, focusing on IEEE 802.15.4 beacon-enabled networks. Section 3 highlights the IEEE 802.15.4 features and its slotted CSMA/CA mechanism. Section 4 presents the proposed differentiation service strategies. Section 5 outlines the implementation, and Section 6 elaborates on the experimental performance evaluation. Section 7 concludes the paper.

2. Related Work

The improvement of the IEEE 802.15.4 Slotted CSMA/CA MAC mechanisms to achieve reduced (soft) delay guarantees and better reliability of time-critical events on Wireless Sensor Networks has drawn a few research works.

In [7], the authors modified the slotted CSMA/CA algorithm to enable fast delivery of high priority frames in emergency situations, using a priority toning strategy. Nodes that have high priority frames to be transmitted must send a tone signal just before the beacon transmission. If the tone signal is detected by the PAN Coordinator, an emergency notification is conveyed in the beacon frame, which alerts other nodes with no urgent

messages to defer their transmissions by some amount of time, in order to privilege high priority frame transmissions at the beginning of the contention access period.

In [8], the authors extend the previous schemes by allowing high priority frames to perform only one Clear Channel Assessment (CCA) operation instead of two, using a frame tailoring strategy, which aims to avoid collisions between data frames and acknowledgment frames when only one CCA is performed. This approach of CCA reduction requires Frame Tailoring, i.e. adjusting data packet length in such a way that one CCA becomes sufficient to detect any acknowledgement frame transmission. While this method reduces the CCA overhead by half, problem of backward incompatibility remains.

PECAP [9] presented yet another solution based on a toning signal. Here, the main idea was to use the inactive portion of the superframe to carry out the transmission of high priority packets. The beginning of this portion is signaled by a jamming signal at the end of the CAP. This approach does not tolerate the use of the CFP for transmitting guaranteed traffic.

Although these solutions seem to improve the responsiveness of high priority frames in IEEE 802.15.4 slotted CSMA/CA, they require a non-negligible change to the IEEE 802.15.4 MAC protocol, thus turning them non-compatible with the standard. The toning mechanism imposes some changes to the hardware (using a tone signal transmitter) and also to the protocol itself, due to the frame tailoring strategy. This represents a major drawback for these proposals since they contradict the ongoing 15.4 working groups standardizations efforts.

Other approaches that do not present such an inconvenient have been proposed in the meanwhile to support service differentiation. These are usually similar to the strategy implemented in [10].

The IEEE 802.11e [10] specified a Hybrid Coordination Function (HCF) by defining variable parameters such as Arbitrary Interframe Space (AIFS), CW_{min} and CW_{max} . This amendment was approved and incorporated in IEEE 802.11-2007 [11] specification.

Recent research works in IEEE 802.15.4 have presented priority-based service differentiation models similar to HCF, by tuning of some of the MAC parameters as the Backoff Exponent (BE) and Contention Window size. In what follows, we enumerate some of those proposals that are focused on the slotted CSMA/CA.

So far, most of the work concerning traffic differentiation either relies on Markov Chain models or on simulation work. In fact, to our best knowledge, besides our work, there is only one proposal that presents an experimental validation in a real WSN platform [16].

Concerning analytical and simulation work, [12] presented a Markov chain model and analysed the impact of changing the backoff and contention window concerning delay and throughput. More recently in [13] the authors modeled a differentiation scheme based in two

priority classes. The differentiation was achieved by changing the CW_{init} value between one and two. Although results were interesting, changing the contention window to one may cause collisions with ACK frames.

This strategy of tuning a set of MAC parameters to improve the performance of a traffic category has been used by other recent works.

In [14] the authors introduced a backoff parameter change to improve the responsiveness of a network control system. The authors used Matlab/Simulink to simulate the control system and evaluate its response. In DBP [15] the authors introduced a (m,k) -firm deadline task model to assign priorities to messages. The Backoff parameters were changed according to the proximity to lose m deadlines within a window of k service requests, and implemented the model in MICAz platforms. However, no thorough evaluation of the effects of the parameter change was carry out in any of these studies.

ANGEL [16], presents, the only implementation and performance evaluation in a WSN platform (Tmote Sky) of a traffic differentiation mechanism, so far. Their approach is based on a multi-queue service implemented in a layer above the IEEE 802.15.4 MAC sub-layer. Traffic differentiation is achieved by tuning some MAC parameters.

However, in their work, the effect of each parameter was not studied separately, and the performance evaluation was only focused on changing the $macMinBE$ and $macMaxBE$ parameters, although it was stated that it was possible to change others.

Moreover, the implementation was built over TinyOS [17] which we find unreliable [20] when facing large amounts of traffic due to its lack of preemption and its FIFO-based task management approach, making it difficult to precisely identify the impact of the parameter variations at heavier traffic loads. Also, in [16], if a lower priority message is already being transmitted by the slotted CSMA-CA algorithm and a higher priority message arrives at the higher priority queue, the transmission is aborted so that the higher priority message can be transmitted. This preemptive approach may lead to the starvation of lower priority traffic under certain conditions.

In this paper, we carry out a thorough experimental validation of a set of traffic differentiation mechanisms, previously presented in [5] which are completely backward compatible with the standard protocol. This work proposed two mechanisms to achieve traffic differentiation in IEEE 802.15.4 beacon-enabled networks: (1) a single FIFO queue supporting different traffic priorities by tuning the $macMinBE$, $aMaxBE$ and CW_{init} MAC parameter, and (2) a multi-queue strategy in which different parameter values were assigned to the different queues. Its improvement was verified by simulation with the OPNET [18] Open-ZB IEEE 802.15.4/ZigBee simulation model [19]. Now we implemented it over a real-time operating system.

Moreover, we would like to assess if such a simple approach is sufficient to satisfy the requirements of time-critical messages and can provide interesting results with

current WSN technology. We believe, this proposal can be easily adopted in the IEEE 802.15.4e extension [6] of the standard.

3. IEEE 802.15.4 Slotted CSMA/CA MAC

In beacon-enabled mode, beacon frames are periodically sent by a central device, referred to as *PAN coordinator*, to identify its PAN and synchronize nodes that are associated with it. The PAN coordinator defines a superframe structure characterized by a *Beacon Interval (BI)* specifying the time between two consecutive beacons, and a *Superframe Duration (SD)* corresponding to the active period, defined as:

$$BI = aBaseSuperframeDuration \cdot 2^{BO} \quad (1)$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO}$$

for $0 \leq SO \leq BO \leq 14$

BO and SO are called *Beacon Order* and *Superframe Order*, respectively. The Beacon Interval may optionally include an inactive period (for $SO < BO$), in which all nodes may enter into a sleep mode, thus saving energy. More details can be found in [4].

By default, nodes compete for medium access using slotted CSMA/CA during the *Contention Access Period (CAP)*. The IEEE 802.15.4 protocol also provides a *Contention-Free Period (CFP)* within the superframe, in which a node may request the PAN coordinator to allocate Guaranteed Time Slots (GTS). In this paper, we consider the physical layer operating in the 2.4 GHz frequency band and with a 250 kbps data rate. The slotted CSMA/CA algorithm is based on a basic time unit called *Backoff Period (BP)*, which is equal to $aUnitBackoffPeriod = 80$ bits (0.32 ms). The slotted CSMA/CA backoff algorithm mainly depends on three variables: (1) the *Backoff Exponent (BE)* enables the computation of the backoff delay, (2) the *Contention Window (CW)* represents the number of BPs during which the channel must be sensed idle before channel access, (3) the *Number of Backoffs (NB)* represents the number of times the CSMA/CA algorithm was required to backoff while attempting to access the channel. Fig. 2 presents the slotted CSMA/CA algorithm [4].

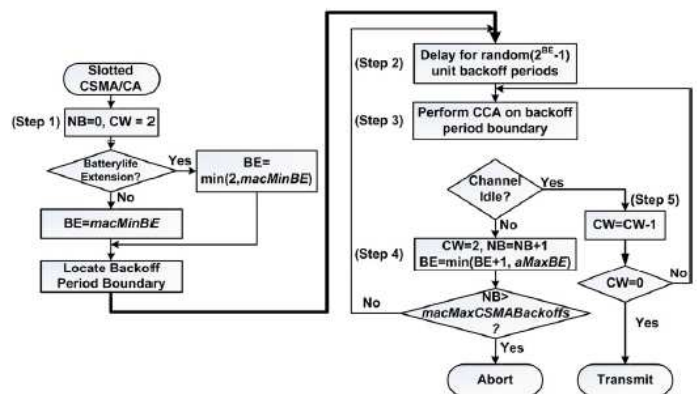


Figure 2 - The IEEE802.15.4 Slotted CSMA/CA Algorithm

First, the number of backoffs and the contention window are initialized ($NB = 0$ and $CW = CW_{init} = 2$) (Step 1). The backoff exponent is also initialized to $BE = 2$ or $BE = \min(2, macMinBE)$, depending on the value of the *Battery Life Extension* MAC attribute. $macMinBE$ is a constant, which is by default equal to 3.

Then, the algorithm starts counting down a random number of BPs uniformly generated within $[0, 2^{BE}-1]$ (Step 2). The count down must start at the boundary of a BP. When the timer expires, the algorithm then performs one CCA operation at the BP boundary to assess channel activity (Step 3). If the channel is busy (Step 4), CW is re-initialized to $CW_{init} = 2$, NB and BE are incremented. BE must not exceed $aMaxBE$ (default value fixed to 5). Incrementing BE increases the probability for having greater backoff delays. If the maximum number of backoffs ($NB = macMaxCSMABackoffs = 5$) is reached, the algorithm reports a failure to the higher layer; otherwise, it goes back to (Step 2) and the backoff operation is restarted. The protocol allows $aMaxFrameRetries = 3$ after each failure. If the channel is sensed as idle, CW is decremented (Step 5). The CCA is repeated if $CW \neq 0$. This ensures performing two CCA operations to prevent potential collisions of acknowledgement frames. If the channel is again sensed as idle, the node attempts to transmit, provided that the remaining BPs in the current CAP are sufficient to transmit the frame and the subsequent acknowledgement. If not, the CCAs and the frame transmission are both deferred to the next superframe. This is referred to as *CCA deferral*.

4. Traffic Differentiation Strategy

As shown in [5], the behavior of slotted CSMA/CA is mostly affected by four initialization parameters, which are: (1) the minimum backoff exponent ($macMinBE$), (2) the maximum backoff exponent ($aMaxBE$), (3) the initial value of the CW (CW_{init}) and (4) the maximum number of backoffs ($macMaxCSMABackoffs$).

Changing the value of any of these parameters will have an impact on the performance. For instance, a performance valuation study in [21] has shown that the average delay of broadcast frames increases with $macMinBE$, whereas the probability of success remains independent of $macMinBE$ in large-scale WSNs. However, the probability of success increases for high $macMinBE$ values, in small-scale WSNs. Based on those observations, we propose to offer differentiated services for time-critical messages. In this paper, our service differentiation mechanisms are particularly based on the $macMinBE$, $aMaxBE$ and CW_{init} parameters.

Note that IEEE 802.15.4 defines two frame types: (1) data traffic, which typically represents sensory data broadcasted to the network (without using acknowledgments), (2) and command traffic, which comprises critical messages (such as alarm reports, PAN management messages and GTS allocation requests) sent by sensor nodes to the PAN Coordinator. Due to their importance, command frames are sent using acknowledged

transmissions and require a particular QoS support to be delivered to their destination in a bounded time interval. In this paper, we consider command frames as the high priority service class and data frames as the low priority service class.

The differentiated service strategies are presented in Fig. 3.

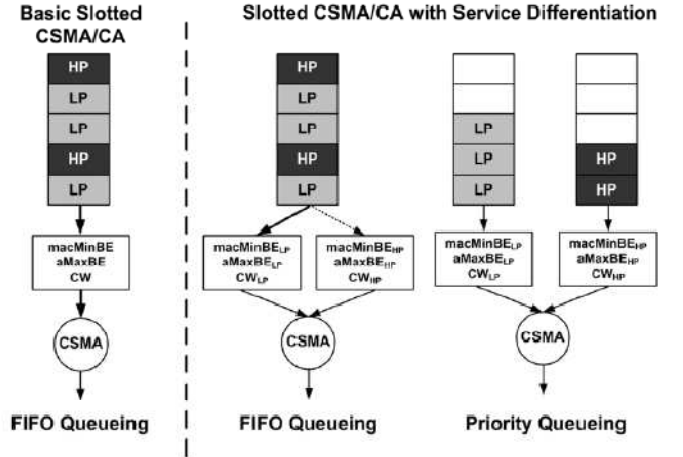


Figure 3 - Differentiated service strategies

The idea is simple. Instead of having the same CSMA/CA parameters for both traffic types, we assign each class its own attributes. We denote $[macMinBE_{HP}, aMaxBE_{HP}]$ and CW_{HP} the backoff interval and the contention window initial values for high priority traffic related to command frames, and $[macMinBE_{LP}, aMaxBE_{LP}]$ and CW_{LP} the initial values for low priority traffic related to data frames. While, the slotted CSMA/CA described in Section 2 remain unchanged, the adequate initial parameters that correspond to each service class must be applied.

In addition to the specification of different CSMA/CA parameters, Priority Queueing can be applied to reduce queuing delays of high priority traffic (Fig. 3). In this case, slotted CSMA/CA uses priority scheduling to select frames from queues, and then applies the adequate parameters corresponding to each service class. Note that if a low priority frame is selected, i.e. the high priority queue is empty, then the backoff process corresponding to this frame will not be preempted by a high priority frame arriving during that service time. It will have to wait until the low priority frame is sent, or rejected if the maximum number of backoff is reached. The heuristics for adequately setting the CSMA/CA parameters are the following. Intuitively, a first differentiation consists in setting CW_{HP} lower than CW_{LP} . It results that low priority traffic has to assess the channel to be idle for a longer time before transmission. A second differentiation is related to the backoff interval. Providing lower backoff delay values for high priority traffic by setting $macMinBE_{HP}$ lower than $macMinBE_{LP}$ would improve its responsiveness without degrading its throughput, as it has been observed in [21] where these intuitive heuristics were previously evaluated.

5. Implementation Approach

The mechanism was implemented over the open-ZB [1] IEEE 802.15.4 stack implementation in ERIKA [23]. ERIKA RTOS is a multi-processor real-time operating system kernel for embedded devices, which implements a set of Application Programming Interfaces (APIs) similar to those of OSEK/VDX [24] standard for automotive embedded controllers.

This version of the open-ZB protocol stack implementation was specially designed to cope with the stringent timing requirements imposed by the IEEE 802.15.4 operating in beacon-enabled mode.

As shown in a previous work [20], fulfilling these requirements can become quite challenging at high duty-cycles or if the network traffic increases considerable, when relying on other operating systems like TinyOS, which do not provide any kind of real-time guarantees.

Because of this fact and since the performance assessment of the proposed mechanism involves a significant stress on the network, and consequently in the OS and protocol stack, we have chosen this platform to assess and validate the traffic differentiation strategies.

5.1. System Overview

The implementation of the IEEE 802.15.4 protocols over ERIKA is organized in a layered architecture, in which the *HW* layer abstracts the current selection of hardware components described in section 6.1. Figure 4 presents an overview of the system architecture [25].

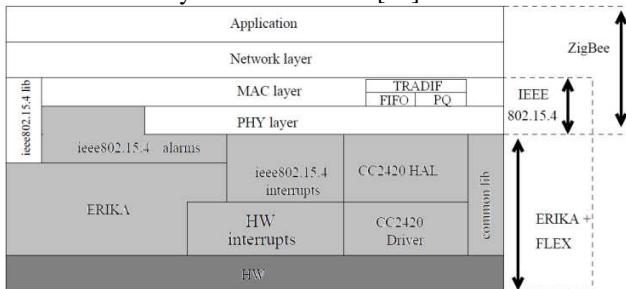


Figure 4 - System Architecture

The *HW interrupts* layer holds the ERIKA Interrupt Service Routines (ISRs), handling all hardware interrupts. Above it, the *ieee802.15.4 interrupts* layer implements the support for the IEEE 802.15.4 communication protocol. This layer contains the code to initialize the hardware timers, to initialize the communication between the radio transceiver and the microcontroller unit (MCU), and to handle timer and transceiver interrupts.

The *CC2420 driver* implements the communication with the radio transceiver and exports to the *CC2420-HAL* all the primitives standardized in IEEE 802.15.4 PHY. The *Transceiver-HAL* was designed to extend the radio transceiver support to different hardware solutions.

The *ERIKA* layer is responsible for managing the system hardware resources and providing OS services such as Task management, resource access control, interrupt and timer management. Software timer abstractions are provided by means of software counters and alarms.

Alarms are software abstractions for timers. The *ieee802.15.4 alarms* are used in this context to activate periodic tasks.

The *common lib* is a generic library providing some software utilities to the upper layers. More specifically, this layer provides: (1) basic data structures used in memory buffer management; and (2) debugging support, e.g. utilities for printing data on the console using serial communication with the MCU UART port.

The *ieee802.15.4 Lib* supports the PHY and MAC layers of IEEE 802.15.4 standard by controlling the timing and memory management services provided by the underlying layers.

5.2. TRADIF implementation

Implementing these mechanisms represented a minor modification to a few MAC layer functions that were in charge of queuing/dequeuing messages and initializing the slotted CSMA/CA parameters. Everything else remained unchanged. A thorough description of the implementation is carried out in [26].

A new mode of operation (*TRADIF*) was implemented in addition to the standard IEEE 802.15.4 implementation, in such a way that it could be enabled or disabled simply by setting a variable in the protocol stack configuration file, in the same way it was possible to set other MAC parameters like *BO* or *SO*. In *TRADIF* mode, support was provided for the two queuing strategies: FIFO and PQ. Since in the proposed mechanism only two priority levels are assumed, Priority Queuing mode support has been provided by maintaining two transmission queues: High Priority (HP) queue and Low Priority (LP) queue.

In the standard mode, when a message is to be sent, it is enqueued in the send buffer and its transmission is triggered. This is unchanged for the FIFO mode of *TRADIF*. In Priority Queuing mode, when a message is to be sent, it is enqueued in the High Priority (HP) or Low Priority (LP) Queue, depending on the priority of the message. In our implementation, command frames have been treated as high priority traffic and data frames as low priority, by default. However, this can be easily modified to support prioritization of traffic generated at application level (which was done for the performance evaluation, as discussed in the next section).

6. Performance Evaluation

We carried out a thorough experimental analysis of TRADIF to understand the impact of these mechanisms on the network performance, namely in terms of *network throughput* (S) and *probability of successful transmissions* (P_s), for different *offered loads* (G), in one cluster with a star-based topology. Both metrics (S , P_s) have been also used to evaluate the performance of the Slotted CSMA/CA MAC protocol [21] in previous works. The network throughput (S) represents the fraction of traffic correctly received normalized to the overall capacity of the network (250 kbps). The success probability (P_s) reflects the degree of reliability achieved by the network for successful

transmissions. This metric is computed as the throughput S divided by G , representing the amount of traffic sent from the application layer to the MAC sub-layer, also normalized to the overall network capacity.

6.1. Testbed Setup

The experimental setup consisted of five FLEX boards [27] programmed with the open-ZB [25] IEEE 802.15.4 implementation over the ERIKA operating system with the traffic differentiation add-on.

The FLEX consists of an embedded board for the development of embedded real-time applications. It features a DsPIC33FJ256MC710 Microcontroller [28] at 40 MHz, 256 Kb of Flash memory and 32 Kb of RAM.

It is also equipped with a Flexipanel EASYBEE IEEE 802.15.4 Transceiver module [29] to enable communications. Figure 5 presents a picture of the setup.

One of these devices was programmed as Coordinator and the others as End Devices. The End Devices were used to generate traffic, both high and low priority, while the Coordinator, apart from synchronizing the devices through beacon transmission, was also used to manage the experiment by transmitting control information included in its beacon payload.

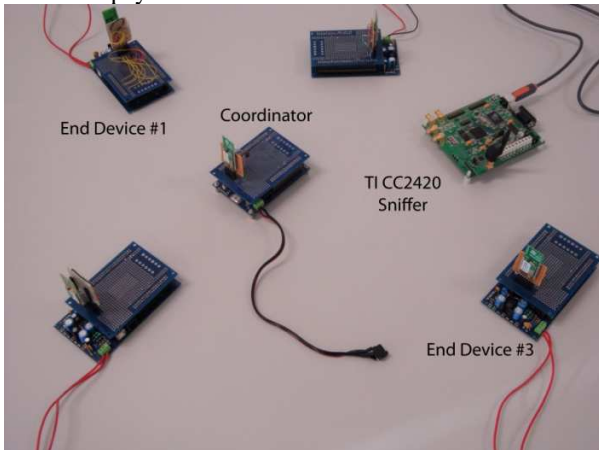


Figure 5 - Testbed Setup

The payload included information about the amount and type of traffic to be generated by the end devices and signals to start and end the experiment (Figure 6). This information was sent to the Coordinator device through a serial port connection. The end devices, upon receiving the beacon, would set the traffic generator alarms (of both high and low priority), with intervals as specified in the beacon payload.

Although the traffic differentiation mechanism considers by default command frames as high priority and data frames as low priority traffic, this was modified to carry out the performance evaluation, and only data frames were used for both high and low priority traffic, not to interfere with the protocol stack. The first byte of the application payload of the packet was used to differentiate both traffic classes.

To measure output parameters such as throughput, delays and queue overflows, the same strategy was used.

Different counters were inserted at various stages of the transmission procedure, starting from the traffic generation at application layer to transmission from the physical layer. For instance, a high priority packet counter at the application level (*hp_app_counter*), was used to count the number of high priority packets generated by an end device from the beginning of the experiment to the instant of that packet creation. The counter was incremented with every call to *generate_hp_traffic()* and inserted into the application payload of the high priority messages. Other counters were used: *lp/hp_queued*, counters representing the number of high and low priority packets successfully enqueued; *lp/hp_mac_sent*, counters representing the number of packets transmitted after completing the CSMA/CA procedure; *lp/hp_csma_fail*, counters representing failed slotted CSMA/CA transmissions; *lp/hp_last_csma_delay_backoff_period*, counters about the CSMA delay in the last transmission of respective priority classes, in terms of the number of backoffs. A Chipcon CC2420 packet sniffer [1] was used to capture the traffic for processing and analysis.

Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Source PAN	Superspan specification	CTS fields	Beacon payload
22	Type Sec Pnd Ack seq Intrta PAN	0x03	0x1234	0xFFFF	0x0000	0x0000	06 06 15 0 1 1	0	01 01 00 3C 00 3C 00
28	Type Sec Pnd Ack seq Intrta PAN	0x05	0x1234	0xFFFF	0x1234	0x0001	11 00 00 00 03 05 02 11	216	OK
28	Type Sec Pnd Ack seq Intrta PAN	0x07	0x1234	0xFFFF	0x1234	0x0001	02 00 01 00 01 00 00 00	216	OK
28	Type Sec Pnd Ack seq Intrta PAN	0x08	0x1234	0xFFFF	0x1234	0x0001	02 00 02 00 02 00 00 00	216	OK
28	Type Sec Pnd Ack seq Intrta PAN	0x09	0x1234	0xFFFF	0x1234	0x0001	02 00 04 02 05 02 12	216	OK
28	Type Sec Pnd Ack seq Intrta PAN	0x0A	0x1234	0xFFFF	0x1234	0x0001	11 00 02 00 02 00 00 00	216	OK
28	Type Sec Pnd Ack seq Intrta PAN	0x0B	0x1234	0xFFFF	0x1234	0x0001	02 00 02 00 02 00 00 00	184	OK
28	Type Sec Pnd Ack seq Intrta PAN	0x0C	0x1234	0xFFFF	0x1234	0x0001	02 00 14 02 05 02 12	184	OK

Figure 6 - A view of the TI CC2420 Sniffer output

The packet analyzer generates a log file containing all the received packets and the corresponding timestamps (Figure 6), enabling to retrieve all the necessary data embedded in the packets payload. A parser application was developed to carry out that task.

6.2. Experimental Evaluation

The set of experiments consisted of varying low priority traffic while keeping high priority traffic constant, and measuring the throughput of the high priority traffic for the different scenarios. The values of CSMA parameters used for each of these scenarios are listed in Table 1.

Scenario	[macMinBE _{HP} , aMaxBE _{HP}]	[macMinBE _{LP} , aMaxBE _{LP}]	CW _{HP}	CW _{LP}
Sc1	[2,5]	[2,5]	2	2
Sc2	[2,5]	[2,5]	2	3
Sc3	[0,5]	[2,5]	2	2
Sc4	[0,5]	[2,5]	2	3

Table 1 - Test Scenarios

Although, the IEEE 802.15.4-2006 standard allows a higher setting of $aMaxBE$, (up to 8), we used the same scenarios to enable a fair comparison with the simulation results in [5]. Each case was examined for FIFO as well as Priority Queuing scheduling policies.

The network was set to work in full duty cycle with $BO=SO=6$, with no-hidden nodes, and the traffic generation was controlled using timers, generating high priority frames at a rate of 40 frames/second and low priority frames ranging from 3 frames/second up to 600 frames/second.

Several runs were carried out for each traffic interval stopping the experiment every time the number of high priority packets received reached 1000.

In the following discussions, Application layer traffic is denoted by $Gapp$ and the MAC layer traffic by $Gmac$. Similarly, $Gapp_{hp}$ and $Gapp_{lp}$ are used to denote Application layer high priority and low priority traffics, and $Gmac_{hp}$, $Gmac_{lp}$ used for MAC layer high and low priority traffic, respectively.

Figure 7 shows the comparison of the success rates of the high priority application traffic of the four scenarios of Table 1, for both FIFO and Priority Queuing mode. These results are analogous to the ones obtained through simulation in [4], illustrated in figure 3 in section 4.2.

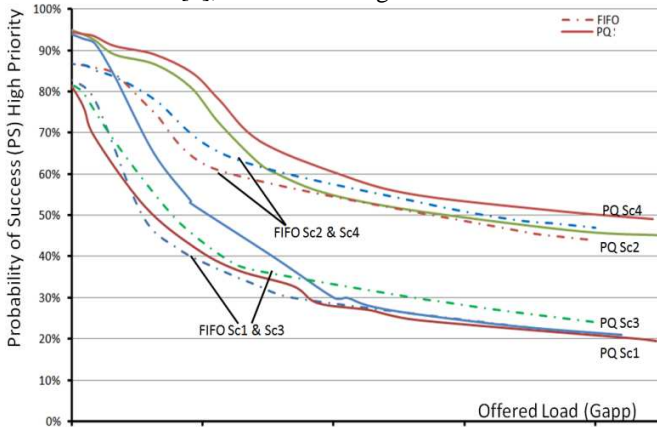


Figure 7 - Probability of Success for FIFO and PQ mode

The contention windows size for high priority frames is kept 2 (standard value) in all cases, while it is increased to 3 for low priority frames in Sc2 and Sc4. On the other hand, the value of $macMinBE$ is kept constant (2, standard value) for low priority traffic in all cases, whereas it is set to 0 for high priority traffic in Sc3 and Sc4.

Concerning the FIFO mechanism, it can be observed that all three scenarios of parameter tuning (Sc [2-4]) result in higher success rates compared to the standard case (Sc1). Sc1, presents the lowest success probability. Sc3, in which $macMinBE_{HP}$ is decreased to 0, results in improved success rates, but it is still very close to the standard case (change of 0-5%). This is so because setting $macMinBE_{HP}$ lower than $macMinBE_{LP}$ means lower backoff delays for high priority traffic (refer to slotted CSMA/CA algorithm, Figure 2), but the number of backoffs and contention

window size, which are directly related to the contention success probability, are unchanged. On the other hand, setting CW_{LP} greater than CW_{HP} means that high priority traffic need the channel to remain idle for shorter time before transmitting, which means higher probability of success in every sensing attempt. The comparatively higher success rates in Sc2 and Sc4 (improvement of 20-25%) reflect this, showing greater improvement in performance by setting $CW_{LP} > CW_{HP}$, compared to changing $macMinBE_{HP}$.

A similar behavior is observed for PQ mode. For both queuing strategies, results were very alike concerning scenarios 2 and 4, showing that the correct setting of the CW has the greatest effect in the throughput of both queuing modes. One of the noticeable changes from the FIFO cases is the fall of success probability of Sc3. Again, the effect of changing $macMinBE_{HP}$, which would decrease the backoff delay of high priority packet, does not make much difference on contention success. Therefore, Sc1 and Sc3 have approximately the same success rates for Priority Queuing at higher traffic loads.

Sc2 and Sc4, again have better success rates since setting having CW_{HP} lesser than CW_{LP} means that high priority traffic need the channel to remain idle for shorter time before transmitting and hence has more chances of success. In this case again, changing CW_{LP} to 3 improves the success rate of high probability packets by 20 to 25%.

As shown, the priority queuing mechanism slightly improves the probability of success when compared to FIFO. However, its main contribution is in reducing the queuing delay as shown in [5], since the high priority queue will always take precedence over low priority queue, thus reducing queuing delay for high priority packets.

To separately evaluate the effect of the priority queuing mechanism, a single sender was used to generate equal amount of high and low priority frames. The queue size for both high and low priority queues was set to hold 15 messages. The Application layer traffic generation rate was increased at equal rate. The number of packets enqueued of both types were calculated by parsing the output file of the sniffer used to receive packets.

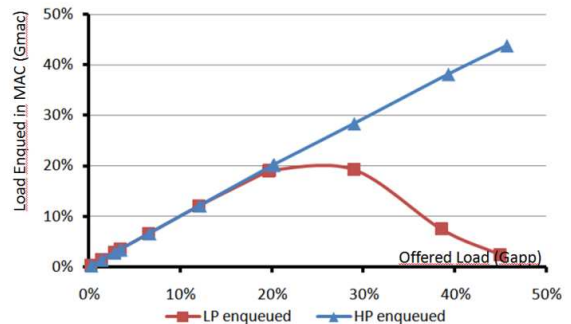


Figure 8 - Comparing queuing success in Priority Queuing

Figure 8 shows the packets enqueued against the packets generated by the application of both high and low priority. It is visible that beyond 20% of channel capacity,

while the low priority frames are dropped due to queue overflow, the high priority frames are unaffected. Moreover, it indicates that at high traffic load, priority queuing plays an important role in ensuring the precedence of high priority frames. This will result in a lower queuing delay for high priority packets.

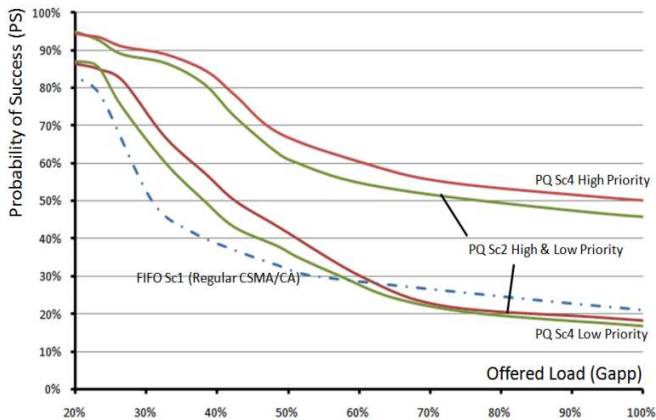


Figure 9 - Probability of Success for HP P in Priority Queuing

However, the improvement of this differentiation scheme to the throughput of high priority command frames is more significant than the degradation of the throughput of low priority data frames (Figure 9), which further demonstrates the efficiency of this differentiation mechanism. As shown, the Probability of Success of low priority frames for PQ Sc2 and Sc4 is just slightly lower (5%) at high offered loads than with the default MAC, taking advantage of lower CW at lower loads, thus increasing throughput.

7. Concluding Remarks

This paper presented the implementation of a set of traffic differentiation mechanisms for the IEEE 802.15.4 slotted CSMA/CA using the open-ZB IEEE 802.15.4/ZigBee protocol stack over the ERIKA real-time operating system. We carried out a thorough experimental analysis of the mechanisms, showing that adequately tuning the parameters of slotted CSMA/CA leads to an improved QoS for time-critical messages. This fact is especially visible by tuning the CW_{init} parameter of the IEEE 802.15.4 MAC.

This practical proposal can be easily used since it only requires a minor add-on and ensures backward compatibility with the existing standard. Thus, it can be integrated in future versions of the standard, such as the IEEE 802.15.4e amendment. Moreover, several higher layers protocols could potentially benefit from these additions, as the IEEE 802.15.4 serves as a baseline for ZigBee and 6LoWPAN, among others.

With this in mind, we are currently triggering the implementation of these mechanisms in TinyOS, both for the IEEE 802.15.4 beacon and non-beacon enabled modes, to provide an even larger WSN community with a simple set of mechanisms for supporting traffic differentiation in IEEE 802.15.4-based networks.

8. References

- [1] IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).", IEEE standard for Information Technology, 2003.
- [2] ZigBee-Alliance, "ZigBee specification," <http://www.ZigBee.org/>, 2010.
- [3] 6LoWPAN Working Group, "IPv6 over Low power WPAN", <http://www.ietf.org/dyn/wg/charter/6lowpan-charter.html>, 2010.
- [4] HCF - HART Communication Foundation, "HART7 Specification", September 2007.
- [5] A. Koubaa, M. Alves, B. Nefzi, and Y.-Q. Song, "Improving the IEEE 802.15.4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks," in Proc. Workshop of Real-Time Networks (RTN 2006), Satellite Workshop to (ECRTS 2006), July 2006.
- [6] IEEE 802.15 WPAN™ Task Group 4e (TG4e), <http://www.ieee802.org/15/pub/TG4e.html>, 2010.
- [7] Kim, D. Lee, J. Ahn, and S. Choi, "Priority toning strategy for fast emergency notification in IEEE 802.15.4 LR-WPAN," in Proceedings of the 15th Joint Conference on Communications & Information (JCCI), April, 2005.
- [8] T. Kim and S. Choi, "Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs," IEEE Communications Letters, Nov. 2005, pp. 213-215.
- [9] J. Jeon, J. W. Lee, et al. (2007). "PECAP: Priority-Based Delay Alleviation Algorithm for IEEE 802.15.4 Beacon-Enabled Networks." Wirel. Pers. Commun. 43(4): 1625-1631.
- [10] "IEEE Std 802.11e: Wireless LAN Medium Access Control (MAC) and Physical Layer specifications," LAN/MAN Standards Committee, 2005.
- [11] IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.
- [12] E.-J., Kim, M. Kim, et al. (2007). "Priority-based service differentiation scheme for IEEE 802.15.4 sensor networks." AEU - International Journal of Electronics and Communications 61(2): 69-81.
- [13] Ndihi, E D N ; Khaled, N ; De Micheli, G, "An Analytical Model for the Contention Access Period of the Slotted IEEE 802.15.4 with Service Differentiation", ICC 2009, International Conference on Communication, Dresden, June 14-18, 2009.
- [14] N. Boughanmi, Y.-Q. Song, E. Rondeau, "Online adaptation of the IEEE 802.15.4 parameters for wireless networked control systems", 8th IFAC International Conference on Fieldbuses and networks in industrial and embedded systems, FET2009, 20/05/2009.
- [15] T. Semperebom, C. Montez; R. Moraes, F. Vasques, R. Custodio, "Distributed DBP: A (m,k)-firm based distributed approach for QoS provision in IEEE 802.15.4 networks", IEEE ETFA 2009, Mallorca. 14th IEEE International Conference on Emerging Technologies and Factory Automation, 2009. v. 1. p. 1-8
- [16] D. Kipnis, A. Willig, J. H. Hauer, and N Karowski, "The ANGEL IEEE 802.15.4 Enhancement Layer: Coupling Priority Queueing and Service Differentiation," In Proceedings of 14th European Wireless Conference, Prague, June 2008, pp.1-7.
- [17] TinyOS, www.tinyos.net, 2010.
- [18] OPNET Technologies, Inc., "Opnet Modeler Wireless Suite - ver. 11.5A," <http://www.opnet.com>, 2010
- [19] Petr Jurčík, Anis Koubâa, The IEEE 802.15.4 OPNET Simulation Model: Reference Guide v2.0", www.open-zb.net, IPP-HURRAY Technical Report, HURRAY-TR-070509, May 2007
- [20] A.Cunha, R. Severino, N. Pereira, A. Koubâa, and M. Alves" ZigBee over TinyOS: implementation and experimental challenges," 8th Portuguese Conference on Automatic Control (CONTROLO'2008), Vila Real, Portugal, 21-23 July, 2008.
- [21] A. Koubaa, M. Alves, E. Tovar, "A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks," In IEEE WFCs 2006, Torino (Italy), June 2006, pp.183-192.
- [22] Open-ZB, "Open-ZB open-source toolset for the IEEE 802.15.4/ZigBee protocols" <http://www.open-zb.net>, 2010.
- [23] ERIKA Real-time operating system, <http://erika.sssup.it/>, 2009
- [24] OSEK, "OSEK/VDX-STANDARD," <http://portal.osek-vdx.org> July, 2009.
- [25] P. Pagano, M. Chitnis, A. Romano, G. Lipari, R. Severino, M. Alves, P. Sousa, E. Tovar, "ERIKa and OpenZB: an implementation for real-time wireless networking," in 24th ACM Symposium on Applied Computing (SAC 2009), Poster Session, March 2009, pp 1687-1688.
- [26] M.Batsa, "Supporting Different QoS Levels in Multiple-Cluster Wireless Sensor Networks", MSc Thesis in Computer Science and Engineering, Department of Electronics and Computer Engineering, Indian Institute of Technology (IIT) Roorkee, September 2009 (defended January 2010). Available at: <http://www.open-zb.net/publications/Master%20Thesis%20-%20Manish.pdf>
- [27] FLEX: Microchip dsPIC evaluation board, "FLEX Embedded Platform Reference Manual", <http://www.evidence.eu.com/content/view/114/204/>, 2010.
- [28] Microchip, "dsPIC33F Family Data Sheet," www.microchip.com, 2010
- [29] Flexipanel, "2.4GHz ZigBee ready IEEE 802.15.4 RF transceiver," www.flexipanel.com, 2010
- [30] Chipcon, Texas Instruments Incorporated, "Chipcon Packet Sniffer for IEEE 802.15.4," www.chipcon.com, 2010.