



Impact of Collisions at Radio Receiver on IEEE 802.15.4 for Wireless Sensor Networks

Sukhvinder S Bamber*

Computer Science & Engineering, Panjab University
SSG Regional Centre, Hoshiarpur,
Punjab, India

Abstract — *This paper simulates the collisions at the radio receiver of IEEE 802.15.4 Wireless Sensor Networks (WSNs) for different current draw parameters: transmit, receive, sleep and idle modes, keeping other parameters like: initial energy and power supply same for all motes. Finally, this paper concludes that that Z1 mote should be implemented in IEEE 802.15.4 WSNs as it produces minimum collisions as compared to other motes.*

Keywords — *WSN, Collisions, Radio Receiver, Telos, MICAz, Z1, Epic Core, Guaranteed Time Slot (GTS) End Device, Contention Access Period (CAP) End Device, Personal Area Network (PAN) Coordinator.*

I. INTRODUCTION

When operating in beacon-enabled mode, i.e. beacon frames are transmitted periodically by a central node called PAN (Personal Area Network) Coordinator for synchronizing the network, The IEEE 802.15.4 protocol allows the allocation/deallocation of GTSs in a superframe for nodes that require real-time guarantees. Hence, the GTS mechanism provides a minimum service guarantee for the corresponding nodes and enables the prediction of the worst-case performance for each node's application. The IEEE 802.15.4 / ZigBee are designed for low-rate and small size Wireless Personal Area Networks (WPANs). The IEEE 802.15.4 Medium Access Control (MAC) protocol has the ability to provide very low duty cycles (from 100% to 0.1%), which is particularly interesting for WSN applications where energy consumption and network lifetime are main concerns [2]. In WSN deployments, reliably reporting data while consuming the least amount of power is the ultimate goal and the traditional IEEE 802.11 standard is developed with no energy minimization mechanisms which are necessary for those 802.15.4, designed for low-rate wireless applications [7].

IEEE 802.15.4 permits up to 10 meter communications with a transfer rate of 250 kbps, although this parameter can be decreased even more (down to 20 kbps in the 868/915 MHz band) to enable a lower power consumption in the ZigBee nodes. IEEE 802.15.4 – compliant transceivers, which operate in the Industrial, Scientific and Medical (ISM) radio bands are designed to be simpler and more economical than the modules from other WPAN standards like: Bluetooth. The main attractiveness and also the main challenge of IEEE 802.15.4 WSN is its potentiality to set up self-organizing networks capable of adapting to diverse topologies, node connectivity and traffic conditions. Typical applications of 802.15.4 WSN usually consists of tens or hundreds of simple battery powered sensor nodes which periodically transmit their sensed data to one or several data sinks (PAN Coordinator).

IEEE 802.15.4 technology was conceived to minimize the power consumption of these sensor nodes. For this purpose, the activity of the nodes must be reduced up to a minimum so that they can remain most of the time in a sleep (low-power) state. Therefore, a node just has to be active in order to sense and transmit data for a small fraction of time. The general objective is to maximize the lifetime of the battery in nodes and consequently the lifetime of the sensor network. In order to predict the battery lifetime of the devices in a practical implementation of IEEE 802.15.4 WSN, we must characterize the current which is drained (consumed) from the battery during the different operations imposed by the dynamics of IEEE 802.15.4 communications, especially those which relates to the activation of radio transceiver. In this paper we have simulated and presented the effects of varying the current consumption in WSN motes keeping all other parameters same in all scenarios except the current draw in a mote in each scenario. Comparing the results of different scenarios for different type of devices conclude that if PLR at any type of device is to taken into consideration for the performance improvement then Z1 mote must be implemented.

This paper is organized as follows: Section II reviews the existing literature of IEEE 802.15.4. Section III gives the brief system description. Section IV presents and discusses the results. Finally, the Section V summarizes the main conclusions of the paper.

II. RELATED WORK

P. Jurcik et al. [1] have proposed an accurate simulation model with focus on the implementation of GTS mechanism. Additionally and most importantly the authors have proposed a novel methodology to tune the protocol parameters so

that better performance of the protocol can be guaranteed, both concerning maximizing the throughput of the allocated GTS as well as minimizing frame delay.

E. Casilari et al. [2] presents an empirical characterization of battery consumption in commercial IEEE 802.15.4/ZigBee motes. This characterization is based on the measurement of the current that is drained from the power source under different 802.15.4 communication operations. The measurement permits the definition of an analytical model to predict the maximum, minimum and mean expected battery lifetime of a sensor networking application.

In [3] O. Landsiedel et al. predicts the accurate power consumption in wireless sensor networks. The authors [4] have empirically characterized the battery consumption in commercial 802.15.4/ZigBee and this characterization is based on the measurement of current that is drained out from the power source under different operations of 802.15.4 communications. In [5] authors have defined a duty cycle in order to allow the devices to achieve efficient energy consumption. The behaviour of 802.15.4 MAC, especially the performance of CSMA/CA algorithm, has been analytically modelled in different papers such as [6] – [7] for beacon – enabled and/or beaconless 802.15.4 networks. The accuracy of all these models, normally based on two – dimensional Markov chains, is evaluated by simulations. Authors [8] have implemented a decentralized power aware approach for data fusion application to increase the WSN lifetime. In [9] R. K. Panta et al. have presented a detailed study of the relationship caused by low power link layer duty cycling mechanism used in WSNs, additionally QuickMAC – a novel duty cycling protocol for WSNs has been implemented. The consumption in beacons networks is also characterized in [10]; in this paper authors present their own measurements of power consumption of a CC2420 transceiver. The authors of [11] propose a method to tune the contention control of slotted CSMA/CA aiming at maximizing power saving and throughput; The study, which is evaluated by simulations utilizing the battery model of a commercial radio module, defines a specific metric to calibrate the battery efficiency; However, the model neglects the energy consumption that takes place for specific operations of radio module (e.g. in the backoff intervals). J.M. Cano-Garcia & E. Casilari have focused on the current demanded by a sensor node in a simple beaconless star topology when the CSMA contention algorithm introduces idle times in the activity of radio transceiver in [12]. The study in [13] suggests the use of battery state in the 802.15.4/ZigBee nodes as a metric for AODV (Ad Hoc on Demand Distance Vector) routing algorithm typically employed in ZigBee mesh topologies. The paper [14] investigates the effects of employing a cryptographic mechanism on the power consumption of beacon-enabled 802.15.4 networks. The mean energy consumption per transmitted byte is computed assuming that a battery mode of radio module [15] is not compatible with 802.15.4 standard. In [16] W. Du et al. have implemented an energy model for WSNs which estimates the energy both for the hardware components of the individual nodes and whole of the sensor network. In [17] authors have proposed the comprehensive simulation study by addressing the impact of IEEE 802.15.4 MAC attributes (BO, SO and BE) on the performance of slotted CSMA/CA in terms of throughput, average delay and success probability. Here the concept of utility, which is defined as a combination of two or more metrics, enables to determine the optimal offered load for achieving the best trade-off between all combined metrics. Koubaa et al. [18] have explored the most relevant characteristics of IEEE 802.15.4 protocol for WSNs and have presented the most important challenges regarding the time-sensitive applications and have also provided some timing performance analysis of the IEEE 802.15.4 that unveils some directions for resolving the previously mentioned paradoxes including power efficiency. Authors of [19] have presented a methodology that provides a Time Division Cluster Scheduling (TDCS) mechanism based on the cyclic extension of RCPS/TC (Resource Constrained Project Scheduling with Temporal Constraints) problem for a cluster-tree WSN, assuming bounded communication errors. Authors of [20] have proposed a power efficient superframe selection method that simultaneously reduces power consumption and enables to meet the delay requirements of real-time flows allocating GTSs. In [22] K. Withephanich et al. have developed an explicit Generalized Predictive Control (GPC) strategy for WSN power control that addresses practical constraints typically posed by health care problems. In [23] – [26] datasheets of various motes have been accessed to compare their performances. S S Bamber et al. [27] proved that there is trade-off for the use of motes in IEEE 802.15.4 WSNs if battery energy consumed is to be taken into consideration.

In this paper, we have compared and characterized the PLR in IEEE 802.15.4 using different motes (like: Z1, Epic Core, MICAz and Telos) under the same set of operations. The ultimate goal is to prove simulatively that certain motes are better as compared to others when PLR is to be taken into consideration.

III. SYSTEM DESCRIPTION

OPNET® Modeler has been used for developing four variants of 802.15.4 i.e. Epic Core, MICAz, Telos and Z1. Each variant (scenario) contains ten GTS enabled nodes and ten non-GTS nodes. GTS nodes can handle only the acknowledged GTS traffic while the non-GTS nodes can handle unacknowledged non-GTS traffic. All four scenarios are same in each and every respect except for the battery parameters like: current draw, initial energy and power supply.

A. Battery Process Model

Fig. 1 explains the process model for the 802.15.4 battery and it consists of init and dissipation states. The state ‘init’ initializes the node ID and the parameters like: power supply, initial energy, receive mode, transmission mode, idle mode and sleep mode. The ‘dissipation’ state gets the information associated with the remote interrupt, computes packet size, energy consumed when transmitting/receiving a packet, computes the time spent and energy consumed by the node in idle state and finally updates the current energy level in transmit, receive, sleep and active periods.

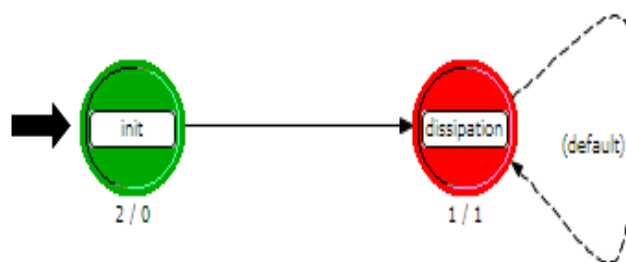


Fig. 1: Battery Process Model

Above mentioned init and dissipation states of battery have been coded as follows:

```

/* init state */
static void wpan_battery_init()
{
  Objid current_draw_comp_id;
  Objid current_draw_id;
  FIN(wpan_battery_init);
  battery.own_id = op_id_self ();
  battery.parent_id = op_topo_parent (battery.own_id);
  op_ima_obj_attr_get (battery.parent_id, "Device Mode", &battery.Device_Mode);
  op_ima_obj_attr_get (battery.own_id, "Power Supply", &battery.power_supply);
  op_ima_obj_attr_get (battery.own_id, "Initial Energy", &battery.initial_energy);
  op_ima_obj_attr_get (battery.own_id, "Current Draw", &current_draw_id);
  current_draw_comp_id = op_topo_child (current_draw_id, OPC_OBJTYPE_GENERIC, 0);
  op_ima_obj_attr_get (current_draw_comp_id, "Receive Mode", &battery.current_rx_mA);
  op_ima_obj_attr_get (current_draw_comp_id, "Transmission Mode", &battery.current_tx_mA);
  op_ima_obj_attr_get (current_draw_comp_id, "Idle Mode", &battery.current_idle_microA);
  op_ima_obj_attr_get (current_draw_comp_id, "Sleep Mode", &battery.current_sleep_microA);
  battery.current_energy = battery.initial_energy;
  statistics.remaining_energy = op_stat_reg ("Battery.Remaining Energy (Joule)", OPC_STAT_INDEX_NONE,
  OPC_STAT_LOCAL);
  statistics.consumed_energy = op_stat_reg ("Battery.Consumed Energy (Joule)", OPC_STAT_INDEX_NONE,
  OPC_STAT_LOCAL);
  statisticsG.consumed_energy = op_stat_reg ("Battery.Consumed Energy (Joule)", OPC_STAT_INDEX_NONE,
  OPC_STAT_GLOBAL);
  op_stat_write(statistics.remaining_energy,battery.current_energy);
  op_stat_write(statistics.consumed_energy,0.0);
  op_stat_write(statisticsG.consumed_energy,0.0);
  activity.is_idle = OPC_TRUE;
  activity.is_sleep = OPC_FALSE;
  activity.last_idle_time = 0.0;
  activity.sleeping_time = 0.0;
  FOUT;
}

/* dissipation state */

static void wpan_battery_update()
{
  Ici * iciptr;
  double tx_time;
  double rx_time;
  double pksize;
  double wpan_data_rate;
  double consumed_energy;
  double idle_duration;
  double sleep_duration;
  FIN(wpan_battery_update);
  if (op_intrpt_type() == OPC_INTRPT_REMOTE) {
  switch (op_intrpt_code()) {
  case PACKET_TX_CODE :
  {
  iciptr = op_intrpt_ici();
  op_ici_attr_get(iciptr, "Packet Size", &pksize);
  }
  }
  }
  }
  
```

```
op_ici_attr_get(iciptr, "WPAN DATA RATE", &wpan_data_rate);
op_ici_destroy(iciptr);
tx_time = pksize/wpan_data_rate;
consumed_energy= (battery.current_tx_mA * milli) * tx_time *
battery.power_supply;
idle_duration = op_sim_time()-activity.last_idle_time;
consumed_energy= consumed_energy +(battery.current_idle_microA * micro) * idle_duration *
battery.power_supply;
battery.current_energy = battery.current_energy - consumed_energy;
activity.last_idle_time = op_sim_time()+tx_time;
op_stat_write(statistics.remaining_energy,battery.current_energy);
op_stat_write(statistics.consumed_energy,battery.initial_energy-
battery.current_energy);
op_stat_write(statisticsG.consumed_energy,consumed_energy);
break;
}
case PACKET_RX_CODE :
{
iciptr=op_intrpt_ici();
op_ici_attr_get(iciptr, "Packet Size",&pksize);
op_ici_attr_get(iciptr, "WPAN DATA RATE",&wpan_data_rate);
op_ici_destroy(iciptr);
rx_time = pksize/wpan_data_rate;
consumed_energy= (battery.current_rx_mA * milli) * rx_time *
battery.power_supply;
idle_duration = op_sim_time()-activity.last_idle_time;
consumed_energy= consumed_energy +(battery.current_idle_microA * micro) * idle_duration *
battery.power_supply;
battery.current_energy = battery.current_energy - consumed_energy;
activity.last_idle_time = op_sim_time();
op_stat_write(statistics.remaining_energy, battery.current_energy);
op_stat_write(statistics.consumed_energy, battery.initial_energy-
battery.current_energy);
op_stat_write(statisticsG.consumed_energy, consumed_energy);
break;
}
case END_OF_SLEEP_PERIOD :
{
sleep_duration = op_sim_time()-activity.sleeping_time;
consumed_energy= (battery.current_sleep_microA * micro) * sleep_duration * battery.power_supply;
printf ("END_OF_SLEEP_PERIOD: current_sleep_microA = %f, time in the sleep period = %f ,
consumed_energy = %f mJoule\n", battery.current_sleep_microA, sleep_duration, consumed_energy*1000);
battery.current_energy = battery.current_energy - consumed_energy;
op_stat_write(statistics.remaining_energy, battery.current_energy);
op_stat_write(statistics.consumed_energy, battery.initial_energy-
battery.current_energy);
op_stat_write(statisticsG.consumed_energy, consumed_energy);
activity.last_idle_time = op_sim_time();
activity.is_idle = OPC_TRUE;
activity.is_sleep = OPC_FALSE;
break;
}
case END_OF_ACTIVE_PERIOD_CODE :
{
idle_duration = op_sim_time()-activity.last_idle_time;
consumed_energy= (battery.current_idle_microA * micro) * idle_duration *
battery.power_supply;
battery.current_energy = battery.current_energy - consumed_energy;
op_stat_write(statistics.remaining_energy,battery.current_energy);
op_stat_write(statistics.consumed_energy,battery.initial_energy-
battery.current_energy);
op_stat_write(statisticsG.consumed_energy,consumed_energy);
activity.sleeping_time = op_sim_time();
```

```

activity.is_idle = OPC_FALSE;
activity.is_sleep = OPC_TRUE;
break;
}
default :
{
};
}
}
FOUT;
}

```

B. Parametric Description

Parametric values for the different types of devices in all scenarios are same except for the battery parameters (as shown in the TABLE I). E.g. parametric values of the PAN Coordinator acknowledged traffic like: MSDU Interarrival time, MSDU size, start time, stop time etc. are same in all four scenarios and the battery parameters like: current draw in 'Idle mode' (1.0, 20, 545 and 426) μ A, is different for each scenario.

TABLE I: PARAMETRIC VALUES OF PAN COORDINATOR, GTS & CAP DEVICES IN DIFFERENT SCENARIOS

Scenario Device Type / Parameter	Epic Core			Micaz			Telos			Z1		
	PAN Coord	GT S	CAP	PAN Coor d	GTS	CAP	PAN Coord	GTS	CA P	PAN Coord	GT S	CA P
<i>Acknowledged Traffic Parameters</i>												
MSDU Interarrival Time (sec)	Exponential (1.0)											
MSDU Size (bits)	Constant (912)											
Start Time (sec)	0.1											
Stop Time (sec)	180											
Destination MAC Address	Broadca st	PAN Coord		Broadca st	PAN Coord		Broadcast	PAN Coord		Broadcast	PAN Coord	
<i>Unacknowledged Traffic Parameters</i>												
MSDU Interarrival Time (sec)	Exponential (1.0)											
MSDU Size (bits)	Constant (912)											
Start Time (sec)	0.1											
Stop Time (sec)	180											
<i>CSMA Parameters</i>												
Maximum Backoff Number	4											
Minimum Backoff Exponent	3											
<i>Battery</i>												
Current Draw Receive Mode (mA)	19.7						21.8			18.8		
Current	17.4						19.5			17.4		

Draw Transmit Mode (mA)									
Current Draw Idle Mode (µA)	1.0	20	54.5	426					
Current Draw Sleep Mode (µA)	9.0	1.0	5.1	20					
Initial Energy	2 AA Batteries (1.5 V, 2300 mAh)								
Power Supply	2 AA Batteries (3V)								
IEEE 802.15.4									
Device Mode	PAN Coord	End Device	PAN Coord	End Device	PAN Coord	End Device	PAN Coord	End Device	
MAC Address	Auto Assigned								
WPAN Settings									
Beacon Order	7								
Superframe Order	3								
PAN ID	0								
Logging									
Enable Logging	Enabled								
GTS Settings									
GTS Permit	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled	
Start Time (sec)	0.1	Infinity	0.1	Infinity	0.1	Infinity	0.1	Infinity	
Stop Time (sec)	180	Infinity	180	Infinity	180	Infinity	180	Infinity	
Length (slots)	2		2		2		2		
Direction	Receive	Transmit	Receive	Transmit	Receive	Transmit	Receive	Transmit	
Buffer Capacity (bits)	10,000	1000	10,000	1000	10,000	1000	10,000	1000	
GTS Traffic Parameters									
MSDU Interarrival Time (sec)	Exponential (1.0)								
MSDU Size (bits)	Constant (912)								
Acknowledgment	Enabled								

IV. RESULTS AND DISCUSSIONS

This section presents the obtained results by varying the sensor notes in IEEE 802.15.4 four different scenarios and keeping all other required parameters same as mentioned in TABLE I same in all the scenarios. In this section the results for Fully Functional Device (FFD) PAN Coordinator and Reduced Functional Devices (RFD) GTS and CAP are presented and analyzed.

A. Radio Receiver Collision Status at Fully Functional Device: PAN Coordinator

Fig. 2 below shows that the collision status at the radio receiver of PAN Coordinator is: 0.970531, 0.778813, 0.585916 and 0.004032 for Epic Core, Telos, MICAz and Z1 motes respectively.

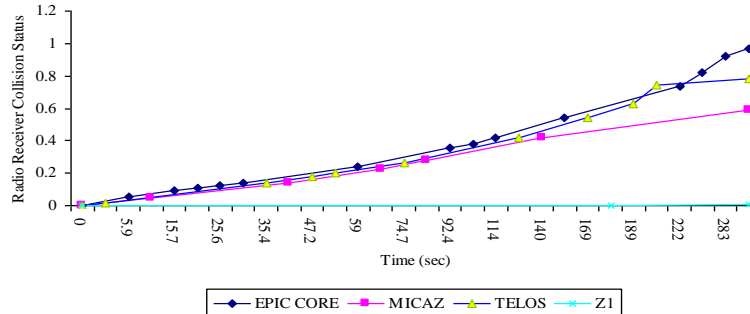


Fig. 2: Radio Receiver Collision Status at PAN Coordinator

It is observed that collisions are minimum in case of Z1 mote because it consumes maximum current in idle mode (TABLE I) and the mean current that must be supplied to a mote so that it can transmit/receive a packet successfully without collisions is [2]:

$$I = [t_{onoff}I_{onoff} + t_{idle}I_{idle} + t_{tx}(n)I_{tx}] / t_{act}(n). \tag{1}$$

Where:

t_{onoff} : Total Time wake up and turn off transceiver as well as to transmit the data.

I_{onoff} : Current necessary to wake up and turn off transceiver as well as to transmit the data.

t_{idle} : Time required to access the radio channel and to receive the acknowledgement from the Coordinator

I_{idle} : Current required to access the radio channel and to receive the acknowledgement from the Coordinator

t_{tx} : Time taken by the mote for transmission

I_{tx} : Current absorbed by the mote during transmission

t_{act} : Time taken to complete the activity period.

n : No. of bytes

Additionally as per implementation:

/ Idle Mode*/*

$$\text{consumed_energy} = (\text{battery.current_idle_microA} * \text{micro}) * \text{idle_duration} * \text{battery.power_supply};$$

$$\text{consumed energy} \propto \text{battery.current_idle_microA} \tag{2}$$

where:

battery.current_idle_microA is the current consumed by battery in idle mode (μA).

Since, Z1 mote consumes more current in idle mode and has maximum energy as compared to other motes (2, TABLE I) therefore, it will consume maximum current while shifting from idle to transmit/receive mode, as a result of which its power level increases the power/bit and most of the bits of data is received successfully and the collisions are reduced to minimum (1). It has also been observed that collisions are maximum in case of Epic Core mote because it consumes least current in the idle mode (TABLE I) as compared to other motes and therefore mean current required to successfully transmit a packet without collisions is minimum in case of Epic Core (1) as a result of which collision status is maximum in case of Epic Core mote.

B. Radio Receiver Collision Status at Reduced Functional Device: GTS End Device

Figure 3 below indicates that the collision status at the radio receiver of GTS end device is: 0.97054, 0.778818, 0.585919 and 0.004032 for Epic core, Telos, MICAz and Z1 motes respectively.

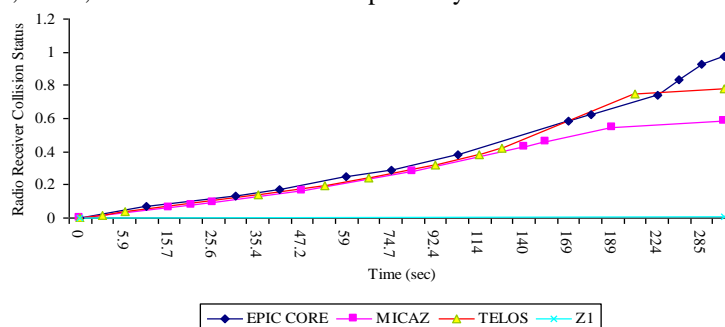


Fig. 3: Radio Receiver Collision Status at GTS End Device

It is observed that collision status is minimum in case of Z1 mote and maximum in case of Epic Core mote for the same reasons as explained in the Section IV-A for the fully functional device – PAN Coordinator.

C. Radio Receiver Collision Status at Reduced Functional Device: CAP End Device

Fig. 4 below shows that the collision status at the radio receiver at the CAP end device is: 0.970537, 0.778816, 0.58592 and 0.004032 for Epic core, Telos, MICAz and Z1 motes respectively.

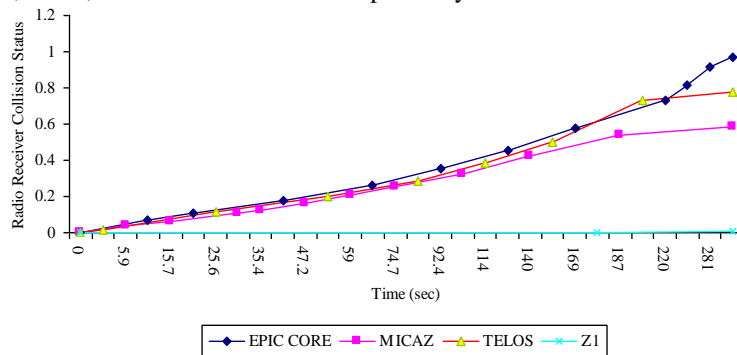


Fig. 4: Radio Receiver Collision Status at CAP End Device

It is observed that collision status is minimum in case of Z1 mote and maximum in case of Epic Core mote for the same reasons as explained in Section IV-A for the fully functional device – PAN Coordinator.

V. CONCLUSIONS

This paper concludes that if collisions at the radio receiver of IEEE 802.15.4 for WSNs are to be used for the performance improvement then Z1 mote at all types of devices in IEEE 802.15.4 WSNs should be implemented as it produces minimum collisions by consuming the maximum current in the Idle mode. Also, concluded that Epic Core mote should not be used at any type of device IEEE 802.15.4 WSNs as it produces maximum collision at all types of devices by consuming the least current in the Idle mode as simulatively proved and statistically shown in the following TABLE II:

TABLE II: STATISTICAL COMPARISON OF COLLISIONS AT FFD & RFD's

Mote(s)	Z1	Epic Core	Telos	MICAz
Radio Receiver Collision Status				
FFD (PAN Coordinator)	0.004032	0.970531	0.778813	0.585916
RFD (GTS)	0.004032	0.97054	0.778818	0.585919
RFD (CAP)	0.004032	0.970537	0.778816	0.58592

REFERENCES

- [1] P. Jurcik, A. Koubaa, M. Alves, E. Tovar and Z. Hanzalek, "A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism", *MASCOTS 2007, 15th International Symposium*, 2007.
- [2] E. Casilari, J. M. Cano-Garcia and G. Campos-Garrido, "Modeling of Current Consumption in 802.15.4/Zigbee Sensor Motes", *Sensors (ISSN 1424 - 8220)*, www.mdpi.com/journal/sensors, Basel, Switzerland, 2010.
- [3] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate Prediction of Power Consumption in Sensor Networks", *Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, Sydney, Australia, 2005.
- [4] E. Casilari, G. Campos-Garrido, and J.M. Cano-Garcia, "Characterization of Battery Consumption in 802.15.4/ZigBee Sensor Motes", *IEEE*, 2010.
- [5] F. Despaux, Y. Song and A. Lahmadi, "Measurement based Analysis of the effect of duty cycle in IEEE 802.15.4 MAC performance", *IEEE International Conference on mobile Ad-hoc and Sensor Systems*, 2013.
- [6] C. Buratti and R. Verdone, "A Mathematical Model for Performance Analysis of IEEE 802.15.4 Non-beacon Enabled Mode", *14th European Wireless Conference (EW 2008)*, Prague, Czech Republic, 2008.
- [7] Z. Chen, C. Lin, H. Wen and H. Yin, "An Analytical Model for Evaluating IEEE 802.15.4 CSMA/CA Protocol for Low-Rate Wireless Application", *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, Niagara Falls, Canada, 2007.
- [8] A. R. Pinto and C. Montez, "Decentralized Power Aware Approach for Data Fusion in IEEE 802.15.4 Wireless Sensor Networks", *Second Brazilian Conference on Critical Embedded Systems, IEEE*, 2012.
- [9] R. K. Panta, J. A. Pelletier and G. Vesonder, "Efficient Asynchronous Low Power Listening for Wireless Sensor Networks", *IEEE International Symposium on Reliable Distributed Systems*, 2012.

- [10] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan and W. Dehaene, "Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modelling and Improvement Perspectives", *Design, Automation and Test in Europe (DATE '05)*, Munich, Germany, 2005.
- [11] Y. Zhang, P. Xu, G. Bi and F. S. Bao, "Analysis of Energy Efficiency and Power Saving in IEEE 802.15.4", *Wireless Communications and Networking Conference (WCNC 2007)*, Hong Kong, China, 2007.
- [12] J. M. Cano-Garcia and E. Casilari, "An Empirical Evaluation of the Consumption of 802.15.4/ZigBee Sensor Motes in Noisy Environments", *International Conference on Networking, Sensing and Control*; Delft, Netherlands, 2011.
- [13] F. Zhang, H. Zhou and X. Zhou, "A Routing Algorithm for Zigbee Network Based on Dynamic Energy Consumption Decisive Path", *International Conference on Computational Intelligence and Natural Computing (CINC '09)*, Wuhan, China, 2009.
- [14] J. Mistic, "Traffic and Energy Consumption of an IEEE 802.15.4 Network in the Presence of Authenticated, ECC Diffie-Hellman Ephemeral Key Exchange", *Computer Networks*, 2008.
- [15] D. Rohm, M. Goyal, H. Hosseini, A. Divjak and Y. Bahsir, "A Simulation Based Analysis of the Impact of IEEE 802.15.4 MAC Parameters on the Performance under Different Traffic Loads", *Mob. Info. Syst.*, 2009.
- [16] W. Du, F. Mieleveville and D. Navarro, "Modeling of Energy Consumption of Wireless sensor Networks by System C", *International Conference on Systems and Networks Communications, IEEE*, 2010.
- [17] A. Koubaa, M. Alves and E. Tovar, "A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks", *International Workshop on Factory Communication Systems (WFCS '06)*, IEEE, 2006.
- [18] A. Koubaa, M. Alves and E. Tovar, "IEEE 802.15.4: A Federating Communication Protocol for Time-Sensitive Wireless Sensor Networks", *TR - 060202*, www.hurray.isep.ipp.pt, 2006.
- [19] Z. Hanzalek and P. Jurcik, "Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks with Time-Bound Data Flows: Application to IEEE 802.15.4/Zigbee", *HURRAY - TR - 100501*, www.hurray.isep.ipp.pt, 2010.
- [20] A. Koubaa, M. Alves and E. Tovar, "Energy/Delay Trade-off of the GTS Allocation Mechanism in IEEE 802.15.4 for Wireless Sensor Networks", *TR - 061002*, www.hurray.isep.ipp.pt, 2006.
- [21] A. Koubaa and M. Alves, "A Two - Tired Architecture for Real - Time Communications in Large - Scale Wireless Sensor Networks: Research Challenges", *TR - 050701*, 2005.
- [22] K. Witheephanich, J.M. Escano and M.J. Hayes, "Explicitly Constrained Generalised Predictive Control Strategies for Power Management in Ambulatory Wireless Sensor Network Systems", *American Control Conference*, USA, 2010.
- [23] MICAz Datasheet: www.memsic.com.
- [24] Epic Core Datasheet: www.moteware.com.
- [25] Z1 Datasheet: www.zolertia.com.
- [26] Crossbow Technology: <http://www.xbow.com>.
- [27] S S Bamber and A K Sharma, " Impact of Current Consumption on Battery Energy in IEEE 802.15.4 for Wireless Sensor Networks with different Sensor Motes", Vol: 3(4), pp. 1-8, *IJARCSSE*, 2014.