ISSN 2277-3061

Performance Effects of Current Consumption on Radio Receiver Bit Error Rate in IEEE 802.15.4 for Wireless Sensor Networks

Sukhvinder S Bamber
Computer Science & Engineering, Punjab University SSG Regional Centre, Punjab, India

ABSTRACT

This paper investigates the radio receiver Bit Error Rate (BER) at different types of devices in IEEE 802.15.4 Wireless Sensor Networks (WSNs) for the different current draw parameters: transmit mode, receive mode, sleep mode and idle mode keeping other parameters like: initial energy and power supply same for all motes; Clearly proving that if BER is to be taken into consideration for the performance enhancement then Z1 mote should be implemented in IEEE 802.15.4 WSNs as they produce minimal BER.

KEYWORDS:

IEEE 802.15.4; WSN; BER; Radio Receiver; Telos; MICAz; Z1; Epic Core; Guaranteed Time Slot (GTS) End Device; Contention Access Period (CAP) End Device; Personal Area Network (PAN) Coordinator.

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS AND TECHNOLOGY
Vol. 13, No. 9
editorijctonline@gmail.com
www.ijctonline.com, www.cirworld.com

1. INTRODUCTION

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]. In fact, 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.

IEEE 802.15.4 protocol provides real-time guarantees by using the GTS mechanism, which is quite attractive for WSNs [1]. 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].

Basic framework of IEEE 802.15.4 permits up to 10 meters 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 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 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 types of devices concludes that if BER is to be taken into consideration in IEEE 802.15.4 for WSNs then Z1 mote should be preferred.

This paper is organized as follows: Section 2 reviews the existing literature on the characterization of IEEE 802.15.4. Section 3 gives the brief system description. Section 4 presents and discusses the results. Finally, the Section 5 summarizes the main conclusions of the paper.

2. RELATED WORK

Ever since the release of IEEE 802.15.4 in 2003, many researches have been done to evaluate its performance in different environments, including software, hardware and analytical analysis. Initially in [1] authors 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 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 commericial 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 modeled 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 beaconed 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. Witheephanich 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 current consumption in IEEE 802.1.5.4 using different motes (like: Z1, Epic Core, MICAz and Telos) under the same set of operations. The ultimate goal is to prove simulatively that how the BER affects the performance of IEEE 802.15.4 WSNs.

3. SYSTEM DESCRIPTION

Simulative model of IEEE 802.15.4/ZigBee implements physical and medium access layer defined in IEEE 802.15.4 standard and application layer defined by ZigBee. The OPNET® Modeler is 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.

ISSN 2277-3061

3.1 Scenarios

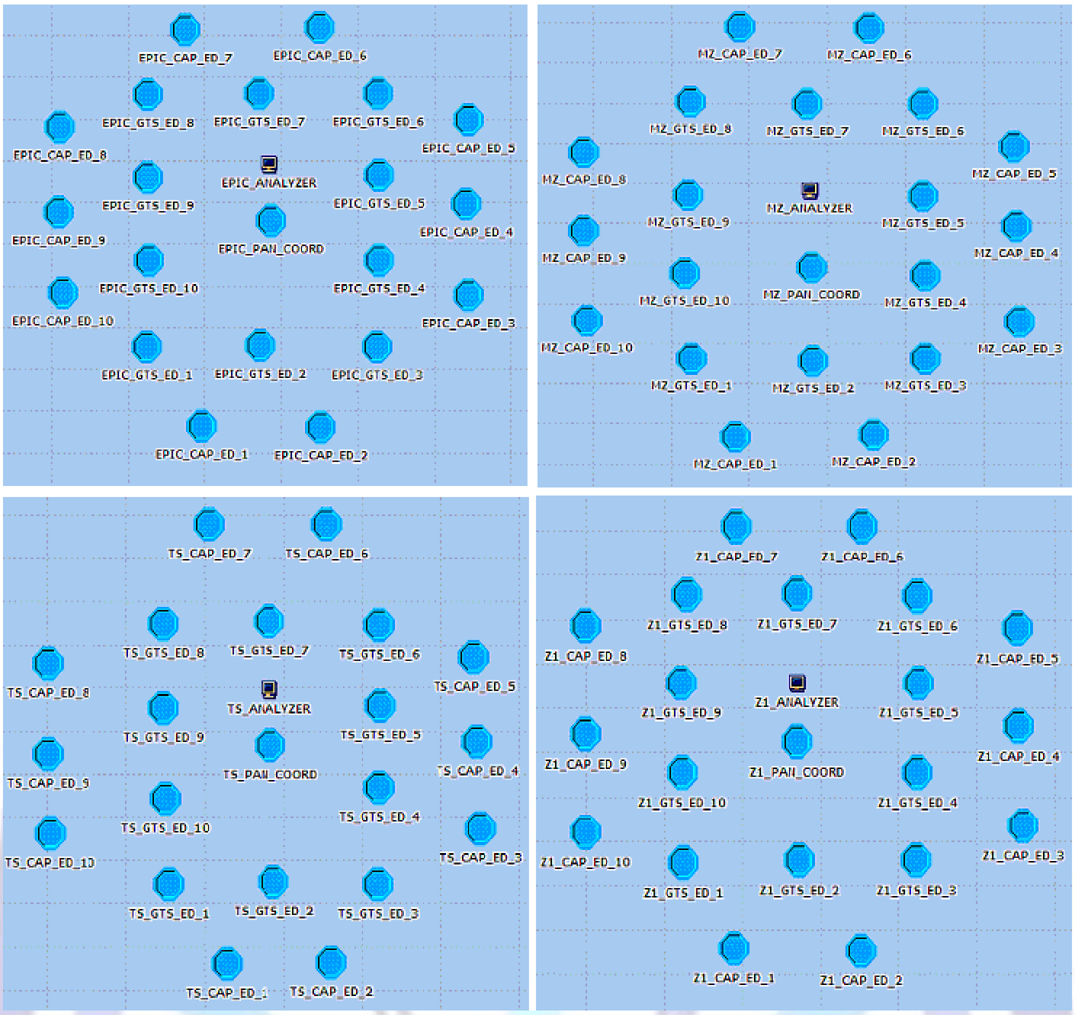

Fig 1: Network Scenarios (a) Epic Core (b) MICAz (c) Telos (d) Z1

Fig. 1(a) shows the Epic Core scenario which contains one PAN Coordinator, one Analyzer and twenty end devices (ten GTS enabled and ten non-GTS enabled), similarly Fig. 1(b) shows MICAz scenario, Fig. 1(c) shows Telos scenario and Fig. 1(d) shows the Z1 scenario. PAN Coordinator is a Fully Functional Device (FFD) that can support three operation modes, serving as:

- A *Personal Area Network (PAN) Coordinator*: the principal controller of the PAN. This device identifies its own network, to which other devices may be associated.
- A *Coordinator*: provides synchronization services through the transmission of beacons. Such a coordinator must be associated to a PAN coordinator and does not create its own network.
- A simple *Device*: a device which does not implement the previous functionalities.

End device is a Reduced Functional Device (RFD) operating with minimal implementation of IEEE 802.15.4 protocol. They do not need to send large amounts of data and associate with a single FFD at a time.

3.2 Battery Process Model

Figure 2 shows 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.

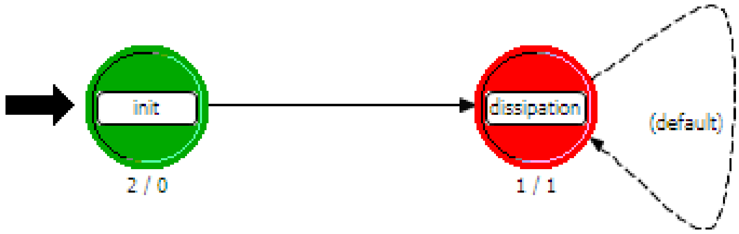

Fig 2: Battery Process Model

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;
}
```

3.3 Parametric Description

Different types of devices in all scenarios have same parametric values except for the battery parameters (as shown in the Table 1). 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) µA, is different for each scenario.

Table 1. Parametric values of PAN Coordinator, GTS and CAP devices in different scenarios.

Scenario	**Epic Core**			**Micaz**			**Telos**			**Z1**		
Device Type / Parameter	**PAN Coord**	**GTS**	**CAP**	**PAN Coord**	**GTS**	**CAP**	**PAN Coord**	**GTS**	**CAP**	**PAN Coord**	**GTS**	**CAP**
Acknowledged Traffic Parameters												
MSDU Interarrival Time (sec)	Exponential (1.0)											
MSDU Size (bits)	Constant (912)											
Start Time (sec)	0.1											
Stop Time (sec)	180											
Destination MAC Address	Broadcast	PAN Coord		Broadcast	PAN Coord		Broadcast	PAN Coord		Broadcast	PAN Coord	
Unacknowledged Traffic Parameters												
MSDU Interarrival Time (sec)	Exponential (1.0)											
MSDU Size (bits)	Constant (912)											
Start Time (sec)	0.1											
Stop Time (sec)	180											

CSMA Parameters								
Maximum Backoff Number	4							
Minimum Backoff Exponent	3							
Battery								
Current Draw Receive Mode (mA)	**19.7**				**21.8**		**18.8**	
Current Draw Transmit Mode (mA)	**17.4**				**19.5**		**17.4**	
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)							
Acknowledgement	Enabled							

4. RESULTS AND DISCUSSIONS

This section presents the obtained results by varying the sensor motes in IEEE 802.15.4 four different scenarios and keeping all other required parameters same as mentioned in table 1 same in all scenarios. In this section the results for Fully Functional Device (FFD) PAN Coordinator and Reduced Functional Devices (RFD) GTS and CAP have been presented.

4.1 Radio Receiver Bit Error Rate at Fully Functional Device (PAN Coordinator)

Fig. 3: below represents that the BER at the radio receiver of a PAN Coordinator is: 0.496495, 0.062314, 0.057142 and 1.62E-07 for Epic Core, Telos, MICAz and Z1 motes respectively.

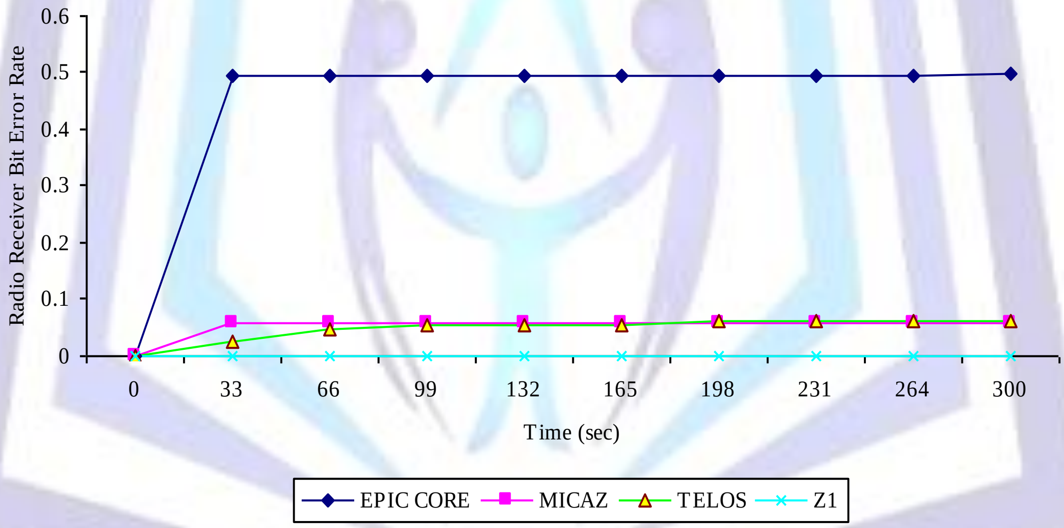

Fig 3: Radio Receiver Bit Error Rate at PAN Coordinator

It is observed that BER is minimum in case of Z1 mote as compared to the other motes because as per implementation:

/* Idle Mode*/

consumed_energy= (battery.current_idle_microA * micro) * idle_duration * battery.power_supply;

consumed energy α battery.current_idle_microA **(Eq. 1)**

where

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

Z1 mote has maximum current consumption in idle mode [table 1] and it will consume maximum current while shifting from idle to transmit/receive mode [Eq. 1] as a result of which its power level increases which in turn increases the power/bit as a result of which its BER decreases as:

BER α 1 / power level. **(Eq. 2)**

It has also been observed that BER is maximum in case of Epic Core mote because it consumes the least amount of current in idle mode (Table 1), therefore its power level is lowest while shifting from idle to transmit/receive mode [Eq. 1] and the BER is maximum [Eq. 2].

4.2 Radio Receiver Bit Error Rate at Reduced Functional Device (GTS End Device)

Fig. 4: below indicates that the BER at the radio receiver of the GTS end device is: 0.031986, 0.017335, 0.01416 and 1.8038E-06 for Telos, MICAz, Epic Core and Z1 motes respectively.

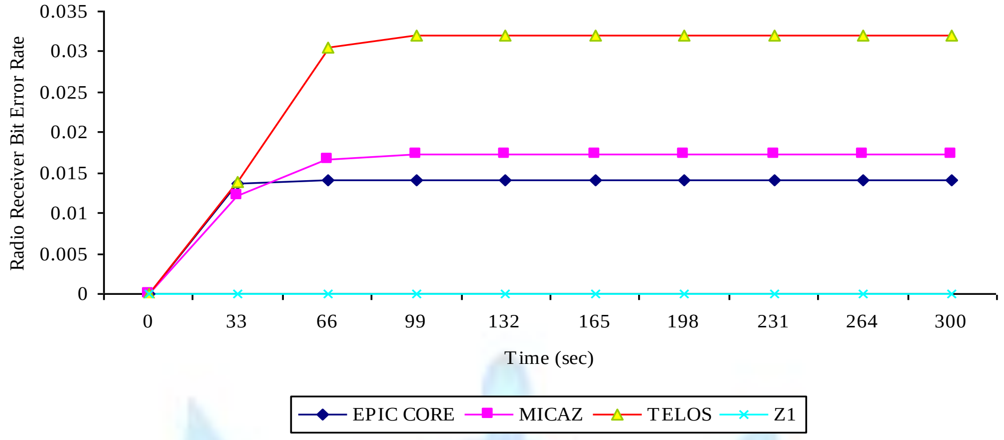

Fig 4: Radio Receiver Bit Error Rate at GTS end device

It is observed that BER is minimum in case of Z1 mote as compared to the other motes for the same reason as cited in section [4.1] for the PAN Coordinator. It has also been observed that BER is maximum in case of Telos mote because GTS end device reserves the bandwidth in advance to provide guarantee of service to a particular application, therefore long queues are formed at the GTS end device as the channel is occupied and also because of higher current consumption in transmit/receive mode as compared to other motes [Table 1] which increases the power/bit [27], data rate is more thus forming longer queues at the transmitter/receiver because of which BER increases in case of Telos mote.

4.3 Radio Receiver Bit Error Rate at Reduced Functional Device (CAP End Device)

Fig. 5: below represents that the BER at the radio receiver of CAP end device is: 0.035997, 0.004442, 4.6E-05 and 1.21E-07 for Epic Core, MICAz, Telos and Z1 motes respectively.

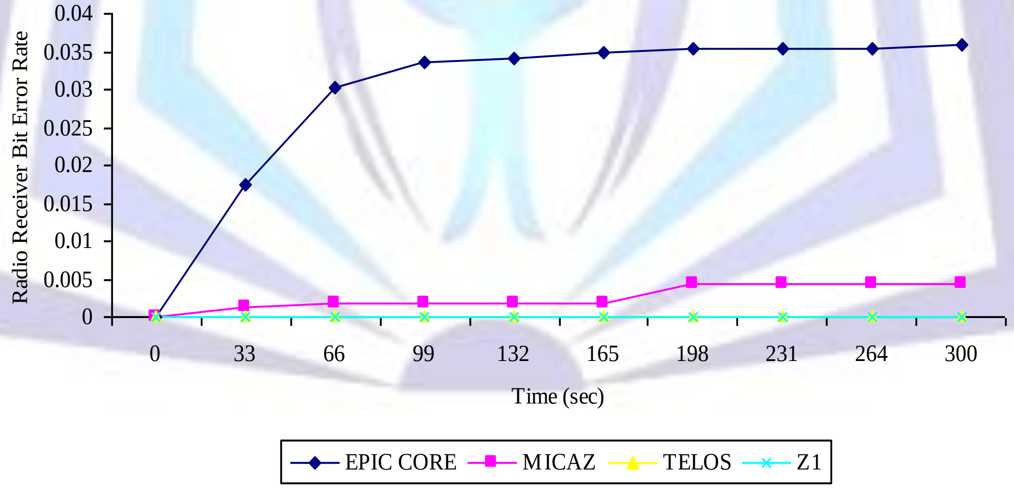

Fig 5: Radio Receiver Bit Error Rate at CAP end device

It is observed that BER is minimum in case of Z1 mote as compared to the other motes for the same reason as cited in section [4.1] for the PAN Coordinator. It has also been observed that BER is maximum in case of Epic Core mote because it consumes the least amount of current in idle mode (Table 1), therefore its power level is lowest while shifting from idle to transmit/receive mode [Eq. 1] and the BER is maximum [Eq. 2].

5. CONCLUSION

This paper provides simulative characterization of BER at the radio receiver in IEEE 802.15.4 with different sensor motes. The characterization concludes that: Firstly, at the PAN Coordinator BER is minimum in case of Z1 Mote (1.62 E-07) while it is maximum in case of Epic Core (0.496495). Therefore at the PAN Coordinator Z1 mote should be preferred. Secondly, at the Reduced Functional Device (GTS End Device) BER is again minimum in case of Z1 mote (1.8038 E-06) while it is maximum in case of Telos (0.031986); therefore proving that at the GTS End Device if BER is to be minimized then again Z1 mote should be implemented. Finally, at the Reduced Functional Device (CAP End Device) BER is again minimum in case Z1 mote (1.21 E-07) while it is maximum in case of Epic Core mote (0.035997), concluding that even at the CAP End Device if the BER is to be minimized then Z1 mote should be preferred. This paper clearly concludes that if BER at the radio receiver in IEEE 802.15.4 WSNs is to be utilized for performance enhancement then Z1 mote should be implemented at all types of devices in IEEE 802.15.4 WSNs.

REFERENCES

[1] Jurcik P., Koubaa A., Alves M., Tovar E. and Hanzalek Z., 2007. “A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism”, MASCOTS 2007, 15th International Symposium.

[2] Casilari E., Cano-Garcia J. M. and Campos-Garrido G., 2010. “Modeling of Current Consumption in 802.15.4/Zigbee Sensor Motes”, Sensors (ISSN 1424 - 8220), www.mdpi.com/journal/sensors, Basel, Switzerland.

[3] Landsiedel O., Wehrle K., and Gotz S., 2005. “Accurate Prediction of Power Consumption in Sensor Networks”, Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II), Sydney, Australia.

[4] Casilari E., Campos-Garrido G., and Cano-Garcia J.M., 2010 “Characterization of Battery Consumption in 802.15.4/ZigBee Sensor Motes”, IEEE.

[5] Despaux F., Song Y. and Lahmadi A., 2013. “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.

[6] Buratti C. and Verdone R., 2008. “A Mathematical Model for Performance Analysis of IEEE 802.15.4 Non-beacon Enabled Mode”, 14th European Wireless Conference (EW 2008), Prague, Czech Republic.

[7] Chen Z., Lin C., Wen H. and Yin H., 2007. “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.

[8] Pinto A. R. and Montez C., 2012. “Decentralized Power Aware Approach for Data Fusion in IEEE 802.15.4 Wireless Sensor Networks”, Second Brazalian Conference on Critical Embedded Systems, IEEE.

[9] Panta R. K., Pelletier J. A. and Vesonder G., 2012. “Efficient Asychronous Low Power Listening for Wireless Sensor Networks”, IEEE International Symposium on Reliable Distributed Systems.

[10] Bougard B., Catthoor F., Daly D. C., Chandrakasan A. and Dehaene W., 2005. “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.

[11] Zhang Y., Xu P., Bi G. and Bao F. S., 2007. “Analysis of Energy Efficiency and Power Saving in IEEE 802.15.4”, Wireless Communications and Networking Conference (WCNC 2007), Hong Kong, China.

[12] Cano-Garcia J. M. and Casilari E., 2011. “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.

[13] Zhang F., Zhou H. and Zhou X., 2009. “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.

[14] Misic J., 2008. “Traffic and Energy Consumption of an IEEE 802.15.4 Network in the Presence of Authenticated, ECC Diffie-Hellman Ephemeral Key Exchange”, Computer Networks.

[15] Rohm D., Goyal M., Hosseini H., Divjak A. and Bahsir Y., 2009. “A Simulation Based Analysis of the Impact of IEEE 802.15.4 MAC Parameters on the Performance under Different Traffic Loads”, Mob. Info. Syst.

[16] Du W., Mieyeville F. and Navarro D., 2010. “Modeling of Energy Consumption of Wireless sensor Networks by System C”, International Conference on Systems and Networks Communications, IEEE.

[17] Koubaa A, Alves M. and Tovar E., 2006. “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.

[18] Koubaa A, Alves M. and Tovar E., 2006. “IEEE 802.15.4: A Federating Communication Protocol for Time-Sensitive Wireless Sensor Networks”, TR – 060202, www.hurray.isep.ipp.pt.

[19] Hanzalek Z. and Jurcik P., 2010. “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.

[20] Koubaa A, Alves M. and Tovar E., 2006. “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.

[21] Koubaa A and Alves M., 2005. "A Two – Tired Architecture for Real – Time Communications in Large – Scale Wireless Sensor Networks: Research Challenges", TR – 050701.

[22] Witheephanich K., Escano J.M. and Hayes M.J., 2010. "Explicitly Constrained Generalised Predictive Control Strategies for Power Management in Ambulatory Wireless Sensor Network Systems", American Control Conference, USA.

[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] Bamber S S and Sharma A K, 2014. " 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.

Authors' Biography with Photo

Sukhvinder S Bamber completed his B.Tech (Computer Science & Engineering) in 2001, M-Tech (Computer Science & Engineering) in 2007. PhD (Computer Science & Engineering) in 2012. Presently working as an Assistant Professor in the department of Computer Science and Engineering, Panjab University SSG Regional Centre, Hoshiarpur, Punjab. Has also worked as a Lecturer at National Institute of Technology, Jalandhar, Punjab for one and a half years. Has published 07 papers in the international Journals. Area of specialization: Wireless Sensor Networks. Areas of research interest : All types of Wired and Wireless Networks. Member Board of Studies from 2013 to 2015 representing Computer Science & Engineering department of University Institute of Engineering & Technology, Hoshiarpur.