Fuzzy Duty Cycle Adaption Algorithm for IEEE 802.15.4 Star Topology Networks

Hayder Ahmed Abdulmohsin

Computer Systems Department, Technical Institute / Qurna, Basra, Iraq

email: hayderaam@gmail.com

Received: 4/10 /2012

Accepted: 22/7 /2013

Abstract – IEEE 802.15.4 is a standard designed for low data rate, low power wireless personal area networks (WPANs) intended to provide interconnection of nodes via radio communication. Such nodes have very limited computation, energy, and communication capabilities. In order to extend the lifetime of the nodes, we propose a fuzzy logic controller to adapt a duty cycle of each node by changing superframe order (SO) based upon the network traffic and node's buffer occupation. Simulating the proposed algorithm in Castalia simulator and comparing it with constant duty cycle schemes are carried out in this paper. Simulation results show that the proposed algorithm decreases drop packets in node's buffer, decreases packet latency and increases energy efficiency.

Keywords – Fuzzy logic, IEEE 802.15.4, Duty cycle, Energy efficiency.

1. Introduction

Low rate-wireless personal area network (LR-WPAN) is a simple, low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. The main objectives of LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol [1]. IEEE 802.15.4 standard defines the physical layer (PHY) and medium access control (MAC) sublayer specifications for such LR-WPANs.

The IEEE 802.15.4 standard defines two medium-access modes: the nonbeacon-enabled mode and beacon-enabled mode. In the nonbeacon-enabled mode, arbitration of medium access is only distributed among wireless nodes based on carrier sense multiple access/collision avoidance (CSMA/CA). In addition to CSMA/CA-based transmissions, the beacon-enabled mode provides a contention-free guaranteed time slot (GTS) mechanism that supports time-critical data delivery [2]. In beacon-enabled mode, an energy saving mechanism can be provided by adapting duty cycle and specify the amount of active period by setting two parameters, *macBeaconOrder* (*BO*) and *macSuperFrameOrder* (*SO*). But the current specification of IEEE 802.15.4 beacon-enabled mode does not define how active and sleep schedules can be configured. This work focuses on the IEEE 802.15.4 beacon-enabled mode.

In IEEE 802.15.4, a large duty cycle with a long active period provides increased bandwidth resources for devices, such that throughput is increased. Since the system stays in wakeup mode for a long time, power consumption is increased. Conversely, a system that operates in a relatively lower duty cycle consumes less power. However, transmission latency increases for the packets generated in the extended inactive period. When the duty cycle is extremely small, the throughput could be reduced significantly because of insufficient bandwidth. Therefore, a trade-off exists for duty-cycle setting between system throughput and device energy consumption [2].

In this paper, we design a fuzzy logic controller to dynamically adjust duty cycle of the IEEE 802.15.4 protocol in star topology LR-WPAN based on network traffic and buffer occupation of nodes. Fuzzy logic is capable of making real time decisions, even incomplete information. Since fuzzy logic systems can manipulate the linguistic rules in a natural way. Moreover, they are simple and can be executed on limited hardware, and unreliable data.

The reset of the paper is organized as follows. In section 2, a general description of IEEE 802.15.4 standard is provided. In section 3, the design of fuzzy logic controller is described. The proposed algorithm is evaluated through simulation in section 4. In section 5, conclusions and future work are given.

2. IEEE 802.15.4

Two different device types can participate in IEEE 802.15.4 network; a full-function device (FFD) and a reduced-function device (RFD). The FFD can operate in three modes serving as a PAN coordinator, a coordinator, or a node. An FFD can talk to RFDs or other FFDs, while an RFD can talk only to an FFD [3]. An RFD is intended for applications that are extremely simple; they do not have the need to send large amounts of data and may only associate with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

The IEEE 802.15.4 protocol supports several network types such as: star, and cluster-tree. The network consists of FFD that perform network control tasks and RFD that perform data sensing and control tasks [1]. Communication is facilitated by the superframe structure as shown in Fig. 1 that is determined by the PAN coordinator. The superframe is activated by a beacon issued by a PAN coordinator, and has an active and inactive portion. The coordinator and nodes can communicate during active periods and enter the sleep mode during inactive periods. The duration of the superframe which called beacon interval (BI) is determined by the parameter BO as follow

$$BI= aBaseSuperFrameDuration \times 2^{BO}, \ 0 \leq BO \leq 14$$

where *aBaseSuperFrameDuration* is the number of symbols forming the superframe. Furthermore, the parameter SO determines the length of an active period which called *SuperframeDuration* (SD); SD is be derived as follows.

$$SD= aBaseSuperFrameDuration \times 2^{SO}, \ 0 \leq SO \leq BO$$

The active portion with 16 time slots has three parts: a beacon, a contention access period (CAP), and a contention free period (CFP). The beacon is transmitted by the coordinator at the start of slot 0, and the CAP follows immediately after. The CAP, uses a slotted CSMA/CA approach for devices accessing the channel. The CFP, uses the standard protocol of GTS for devices to exclusively occupying the transmission time slots.

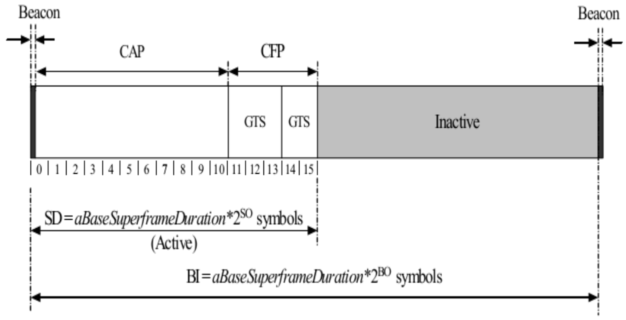


Figure 1. The superframe structure of IEEE 802.15.4 [1].

3. Proposed Fuzzy Duty Cycle Adaption Algorithm

3.1. Principle

In IEEE 802.15.4, the duty cycle is determined by PAN coordinator by means of adapting two system parameters BO and SO. In this paper, we set the parameter BO to the default value of 6 without change. We use fuzzy logic controller to adapt SO depends on network traffic which is the number of packets received by coordinator from nodes during the active period of the superframe and buffer occupation of nodes with number of packets that are waiting in node's buffer to transmit. The star topology network is used with one coordinator placed at the center and each node in the network can communicate directly with the coordinator.

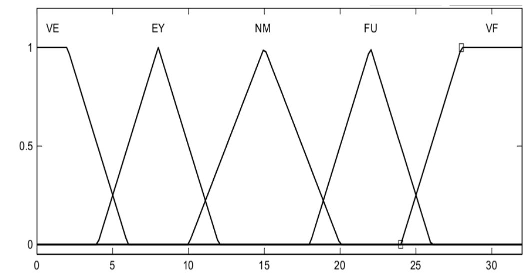
The fuzzy logic controller is located at the coordinator. The coordinator can counts the number of packets received from nodes during the active period of the superframe and each node send the queue occupation to the coordinator in the first data packet sent within the superframe. Based on these two variables, network traffic and buffer occupation, the fuzzy logic controller determines the duty cycle of the superframe by selecting SO parameter.

3.2. Fuzzy Logic Controller

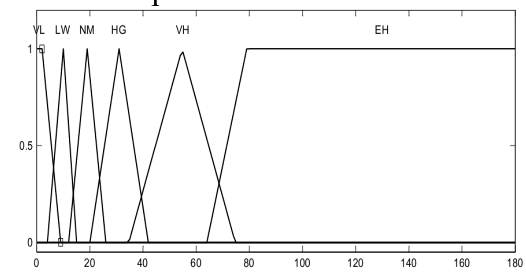
The fuzzy inference system is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. The basic structure of a fuzzy inference system consists of three components [4]: (1) a fuzzification, which is the mapping of the crisp values into a fuzzy sets represented by membership functions, (2) The fuzzy logic rule base, and (3) defuzzification, which converts fuzzy conclusions into crisp control actions. There are many approaches to achieve defuzzification like center of area [5].

In the design process, we apply Mamdani fuzzy system [6] to design of fuzzy logic controller due to its simplicity and easily to be implemented in LR-WPAN. Based on previous assumptions, fuzzy logic controller controls duty cycle based on network traffic and node's buffer occupation. Thus there are two inputs and one output for the Mamdani fuzzy system. The first input is the network traffic which is the number of received packets during the active part of superframe, and represented as *traffic*. The second input is buffer occupation in the nodes which is the average number of packets that are waiting to be transmitted, and represented as *occupation*. The output is the value of SO, and represented as *SO*.

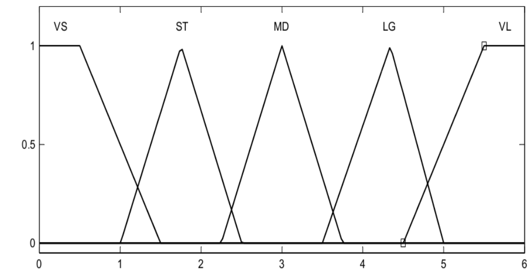
In this simulation, we use chipcon CC2420 as a radio [6]. In CC2420 radio, data rate is 250 kbps and the number of bits per symbols is 4. Thus *aBaseSuperFrameDuration* is 960 symbols or 15.36ms and BI = 15.36ms x 2^6 = 938.04ms. We set universes of *traffic* to [0, 180]. We also set the buffer size in each node to 32 packets and the universes of *occupation* is set to [0, 32]. Furthermore, the universes of SO is set to [0,6] according to 0≤SO≤BO.



(a) Membership functions and universes of Occupation



(b) Membership functions and universes of Traffic



(c) Membership functions and universes of SO

Figure 2. Membership functions of fuzzy logic controller inputs and output

The linguistic variables to represent the *traffic* input are divided into six variables which are Very Low (VL), Low (LW), Normal (NM), High (HG), Very High (VH) and Extremely High (EH) as shown in Fig. 2(a). Furthermore, five linguistic variables are used to describe both *occupation* input and *SO* output. They are Very Empty (VE), Empty (EY), Normal

(NM), Full (FU) and Very Full (VF) for occupation as shown in Fig. 2(b) and Very Short (VS), Short (ST), Medium (MD), Long (LG) and Very Long (VL) for SO as shown in Fig. 2(c). The triangle and trapezoidal membership functions is used to represent the linguistic variables. Because the traffic has seven membership functions and occupation has five membership functions, there are 30 fuzzy rules. Table 1 shows these fuzzy rules.

Fuzzy logic controller can be solved by using lookup table (LUT) [7]. Furthermore, because the inputs and output is discrete, it is easy to use LUT to implement fuzzy logic controller. The relation between the two inputs (*traffic* and *occupation*) and the output (*SO*) can arrange the LUT. We have two inputs and one output, therefore LUT becomes two-dimensional table. When PAN coordinator wants to adjust duty cycle, it just finds a correct entry in the LUT which contains the proper SO value.

Table 1. Fuzzy logic rule base

SO		Occupation				
		VE	EY	NM	FU	VF
Traffic	VL	VS	VS	ST	MD	LG
	LW	VS	ST	MD	LG	VL
	NM	ST	MD	LG	VL	VL
	HG	MD	LG	VL	VL	VL
	VH	LG	VL	VL	VL	VL
	EH	VL	VL	VL	VL	VL

4. Simulation Results

The simulation of the proposed algorithm is tested using Castalia version 3.2 [8] and OMNeT++ version 4.1 [9]. Castalia is a simulator for wireless sensor networks (WSN), body area networks (BAN) and generally networks of low-power embedded devices and it is built on top of OMNeT++. The parameters values used in this simulation are summarized in table 2.

The network used for our simulation is star topology with one PAN coordinator and eight nodes as in Fig. 3. The coordinator determines SO by employing fuzzy logic controller and periodically broadcasts SO inside the beacon. Each node sends packets to the coordinator with transmission interval of 0.1, 0.5, 1, 5 and 10 seconds whereas each node sends 1000 packets at each transmission interval. Initially we set SO to 4.

Fig. 4 shows the average number of dropped packets. The packets drop occurs when buffer is full and there is no space to hold packets arrived from network layer. We can see that as transmission intervals are decreased the numbers of dropped packets are increased. Fig. 4 shows that the dropped packets in fuzzy duty cycle adaption algorithm (FDCA) is low compared with SO equal to 0 and 1 because FDCA always adjust duty cycle according to network traffic and buffer occupation of the node unlike schemes with constant duty cycle.

Fig. 5 shows the average data rate with different transmission interval, we can see that with transmission interval of 0.1 second and 0.5 second the data rate for schemes with SO of 0, 1 and is very low because there is no enough time to send packets (low duty cycle) and the number of dropped packets are increased while in FDCA data rate is very high for the same reasons stated above.

Fig. 6 shows the average energy efficiency which is measured as the total consumed energy (in joule) to total successful delivered packets. We can see that FDCA always achieves better energy efficiency then other constant duty cycle schemes. In the case of high transmission interval with SO of 0 and 1, the number of dropped packets is increased and energy efficiency become with transmission interval of low value with SO of 4 and 5, the idle listening is high and the consumed energy is very high.

Table 2. Values of simulation parameters

Parameter	Value
Data packet size	70 bytes
Tx power consumption	62 mW
Rx power consumption	57.24 mW
Sleep power consumption	1.4 mW
Data rate	250 kbps
Bits per symbol	4 bits

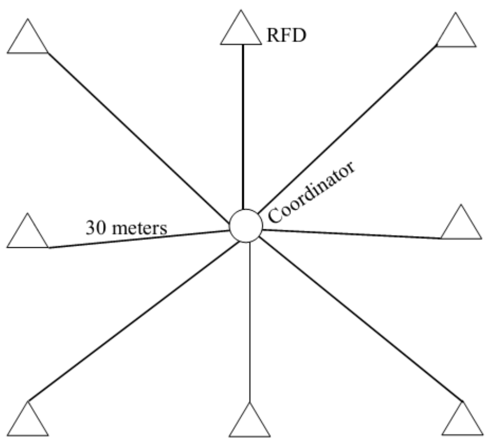


Figure 3. The network used in simulation

Fig. 7 shows the application-level latency histograms with 6 time buckets which is measured as delay in seconds since the packet generated by node's application layer until it is received by the application layer at coordinator. As we explained that FDCA considers the packets waiting in node's buffer to transmitting unlike other constant duty cycle schemes which increases their latency as traffic in the network is increased. Therefore, FDCA have less latency than others. We can see that most of the packets are received under (0,2) seconds bucket latency with FDCA and SO=5, which means that the packets do not wait along in node's buffer while with SO=0, 1 and 2 most of the packets are received with under [10,inf) seconds bucket latency because packets are waiting along in the node's buffer.

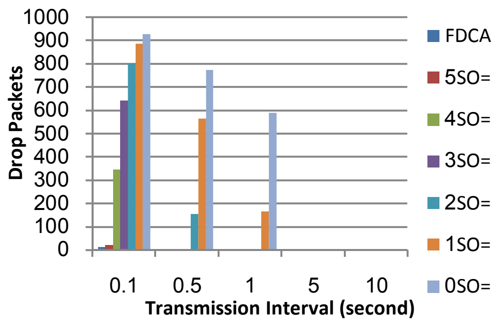


Figure 4. Drop packets with different transmission interval

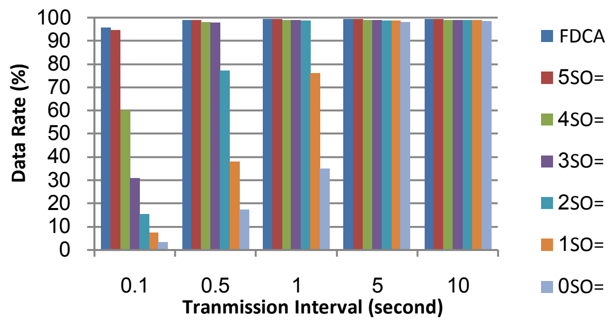


Figure 5. Data rate with different transmission interval

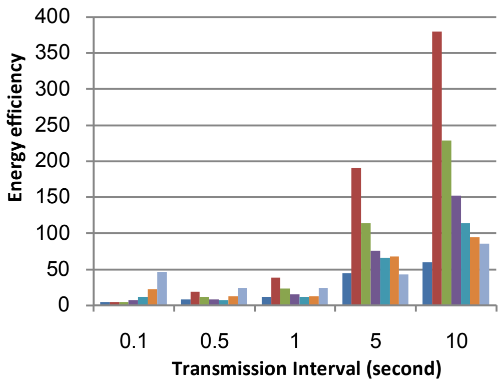
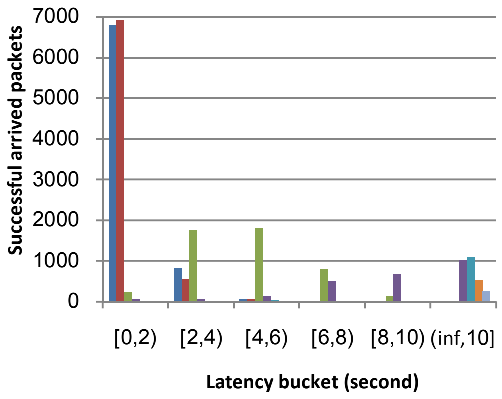
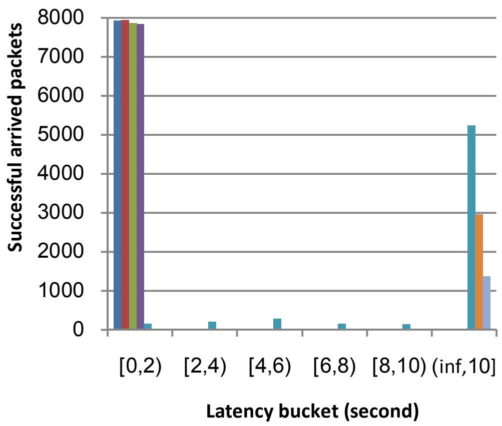


Figure 6. Energy efficiency measured as successful packets per power in joules



(a) Transmission interval 0.1 second



(b) Transmission interval 0.5 second

Figure 7. Application level latency histograms

5. Conclusions and Future Work

In this paper we explore fuzzy logic as solution to duty cycle adaption problem in IEEE 802.15.4 star topology network. We propose an algorithm called fuzzy duty cycle adaption algorithm that employs fuzzy logic controller to adapt duty cycle according to network traffic and buffer occupation of nodes. Simulating FDCA in Castalia simulator and comparing it with constant duty cycle schemes are done here. The simulation results show that FDCA decrease drop packets in node's buffer, increase data rate, decrease packet latency and increase energy efficiency.

The proposed algorithm can be implemented in star topology network. The proposed algorithm can be improved for cluster-tree topology where routing mechanism have to be considered. Such a work is our major future work.

References

[1] IEEE 802.15.4 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2006.

[2] Y. Huang, A. Pang and H. Hung, "A Comprehensive Analysis of Low-Power Operation for Beacon-Enabled IEEE 802.15.4 Wireless Networks," IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 8, NO. 11, NOVEMBER 2009.

[3] M. Valero, R. A. Beyah and A. Bourgeois, Implementing Energy-Saving Improvements to the IEEE 802.15.4 MAC Protocol, Computer sceince thesis, Department of Computer Science at Digital Archive, Georgia State University, 2009.

[4] G. Chen and T. T. Pham, Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, CRC Press LLC, 2001.

[5] A. Costa, A. De Gloria, P. Faraboschi, A. Pagni and G. Rizzotto, "Hardware solutions for fuzzy control," Proc. of the IEEE, vol. 83, Issue: 3, pp. 422 - 434, March 1995.

[6] E. Mamdan, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," Computers, IEEE Transactions on, vol. C-26, no. 12, pp. 1182–1191, Dec. 1977.

[7] CC2420 radiochip datasheet: 2.4Ghz IEEE 802.15.4 Zigbee Ready RF Transceiver, Texas Instruments, 2012, Available from http://www.ti.com.

[8] J. Jantzen. Design of Fuzzy Controllers, 1998, This paper is available on www.iau.dtu.dk/~jj/pubs/design.pdf.

[9] A. Boulis and D. Pediaditakis, Castalia A Simulator for WSNs, National ICT Australia, 2012, http://castalia.npc.nicta.com.au.

[10]A. Varga, OMNet++ Discrete Event Simulation System, 2012, www.omnetpp.org.

.