Sādhanā Vol. 42, No. 6, June 2017, pp. 865–878
DOI 10.1007/s12046-017-0648-2

Lifetime and latency analysis of IEEE 802.15.6 WBAN with interrupted sleep mechanism

ANIL K JACOB*, GEETHU M KISHORE and LILLYKUTTY JACOB

National Institute of Technology, Kozhikode 673601, India
e-mail: akanjama@hotmail.com; geethu_m130434ec@nitc.ac.in; lilly@nitc.ac.in

MS received 2 February 2016; revised 7 October 2016; accepted 27 November 2016

Abstract. It is of utmost importance in a wireless body area network (WBAN) to improve the lifetimes of devices, while restricting latencies within allowable limits. These two demands are often conflicting, and a method to ensure fairly good values for these parameters with a view to satisfying the requirements of the WBAN application would be highly desirable. We consider CSMA/CA option of the medium access in 802.15.6 standard, and propose a sleep mechanism for the devices. An M/G/1 queue with repeated inhomogeneous vacations model is used for the medium access in a typical WBAN network in hospital environments to see how the requirements of lifetimes and delays are taken care of. An analytical method for finding the probability generating function of the contention delay for medium access is developed first using Markovian techniques. The results obtained are then used in the queueing model. Comparison of theoretical values with simulations results shows a fairly close match and defines the conditions that affect the interplay of lifetimes and latencies.

Keywords. WBAN; IEEE 802.15.6; latency; sleep schedule; lifetime; M/G/1 queue with inhomogeneous vacations.

1. Introduction

A collection of medical devices with a coordinator, for gathering information from the devices and then sending it to a remote unit for monitoring the health conditions of patients, constitutes a wirelss body area network (WBAN). It supports the provision of health care for patients in hospital or at home by facilitating the diagnosis of diseases, by providing prompt responses to critical health conditions and by continuously gathering and disseminating information of patients to the health personnel [1].

Medical devices have two important constraints. They should run for as long as possible with a given battery capacity. Also, medical devices have strict latency requirements that are characteristic of applications they are serving. Some classes of medical applications and their required data rates and latencies are specified in ISO/IEEE 11073 specification [2]. Most of the medical applications like determination of blood saturation, blood pressure, heart rate, temperature, etc. are of low data rate (less than 10 kbps), while applications like EEG, motion sensor, video, etc. are of higher data rate. The quality of service (QoS) requirements of these applications, namely reliability, energy efficiency, and latency [3] that the network should provide, are influenced to a great extent by the type of medium access method that is used. Contention-based schemes such as CSMA/CA are more appropriate for low data rate since channel resources can be effectively utilized, whereas polling or scheduled access is suitable for applications with higher data rate as fair sharing of resources can be ensured.

In the 802.15.6 standard, medium access is provided using one of the following three modes. Mode 1 access has beacons with superframes, Mode 2 access has superframes without beacons and Mode 3 access has neither frames nor beacons [4]. There are three categories of medium access control mechanisms: (i) contention access that uses either CSMA/CA or S-Aloha; (ii) improvised and unscheduled access (connectionless, contention-free), which uses polling/posting and (iii) scheduled access (connection oriented, contention-free), also called 1-periodic or *m*-periodic allocations.

Several works on medium access for WBAN have been reported recently. Analytical work for throughput and delay limits of IEEE 802.15.6 without considering any specific MAC schemes is given in [5].

Simulation studies are described in [6], which investigate the impact of the unique WBAN channel characteristics on the trade-offs in the packet delivery vs latency vs consumed energy. However, the study does not address the effectiveness of these access methods in meeting the QoS requirements like lifetimes or latencies of heterogeneous traffic.

*For correspondence

Tachtatzis *et al* [7] give an analytical model for finding out the device lifetime when IEEE 802.15.6 scheduled mode is used for medium access. Tachtatzis *et al* [8] have used integer programming to find out the lifetime of applications mentioned in ISO/IEEE 11073 using Type I scheduled access. In [9], the authors give an analytical model for finding performance of WBAN network in CSMA mode in saturation condition. Markovian techniques have been used in their analysis. Their results showed that in saturation condition the highest priority nodes occupy the medium most of the time. The performance of IEEE 802.15.6 in non-saturation condition is studied in [10]. The average delay is found out.

Motoyama [11] proposes a polling scheme for WBAN with QoS capability.

A combination of polling and probabilistic contention is used for random access in [12], which uses energy harvesting to soften the problems arising from limited battery capacities of body sensor nodes. The work, however, does not focus on the performance of the medium access protocol. In [13], we have investigated CSMA/CA and polling, and evaluated their effectiveness for QoS support in WBAN with multipriority traffic. Analytical models for access delay and lifetime were developed. Also, we proposed a sleeping schedule for polling access scheme to extend device lifetime. Our analysis of priority contention scheme and priority polling scheme with and without sleeping revealed superior lifetime performance in polling access with proposed sleeping mechanism.

This paper is an extension of our work in [13]. Here we propose a sleeping mechanism for CSMA/CA access.

In order to increase the energy efficiency in CSMA access, nodes are put to sleep when there are no packets to be transmitted. But there is a problem with long sleep times because packet latencies get affected. In this paper we look into a sleeping mechanism where a node goes into sleep for a fixed time when there are no packets to be transmitted. The node then wakes up and checks the state of the transmit queue. If it is empty, the node continues with its sleep, else, it begins transmitting packets. Once the packets are transmitted completely, the node again sleeps and the process is repeated. An M/G/1 queue with repeated inhomogeneous vacations [14] is used as the queueing model. The analytical values are then validated using a Castalia simulator on a typical configuration of medical devices with different priority data, as found in a hospital setting. The focus of the work is on the analysis of this network with the proposed sleeping mechanism and its effect on the lifetimes and latencies of the devices.

The rest of the paper is organized as follows. Section 2 describes a Markov model for CSMA/CA access based on 802.15.6 standard. Analytical expressions for the mean and generating function of service times of packets are developed. The proposed power saving mechanism for the CSMA/CA access of 802.15.6 is described in section 3. Section 4 derives analytical expressions for the lifetimes and latencies of devices in the network using the sleeping mechanism proposed. Section 5 gives results of the simulations performed and the comparison done with the analytical results. Section 6 concludes the paper.

2. MAC layer service time

The IEEE 802.15.6 standard specifies one-hop star and two-hop restricted tree topologies. In the one-hop topology, frames are exchanged between nodes and hub, while in the two-hop restricted tree, hub and nodes may use a relay node to exchange frames. In this paper we consider the one-hop topology. Each node stands for a medical device, which can be a sensor device that transmits the measured data to the hub.

The structure of 802.15.6 standard superframe is as shown in figure 1. The superframe is divided into Exclusive Access Phase (EAP), Random Access Phase (RAP), Managed Access Phase (MAP) and Contention Access Phase (CAP). The standard allows setting of access phases other than RAP1 to zero. A beacon is broadcasted by the hub to all nodes at the beginning of each superframe. In this paper, we consider the superframe as comprising EAP1 and RAP1 with all other phases set to zero. The standard allows different options for medium access. As per the standard, medium access during EAP, RAP and CAP phases is contention-based (via S-Aloha or CSMA/CA) and during the Managed phase it is contention-free (via polling or scheduled access). Since we consider contention-based medium access option, we set the managed phase to zero. EAP2, RAP2 and CAP are also set to zero for ease of analysis. The standard allows eight user priorities (UPs) with UP_0 as the lowest and UP_7 as the highest priority. Each UP is associated with a particular contention window range. The basic access mode of CSMA/CA (i.e., no RTS-CTS) is used.

The CSMA/CA access of 802.15.6 standard is as follows. $CW_{k,min}$ is the minimum contention window size of node with priority k. $CW_{k,max}$ is the maximum contention window size of node with priority k. $W_{k,i}$ is the maximum window size of the priority-k node at i^{th} backoff stage, $i \geq 0$; m_k is the backoff stage for priority-k node such that $2^{m_k} = CW_{k,max}$. R is the maximum backoff stage possible, after which the frame is dropped. During the start of a frame transmission by a node, backoff counter of the node is set with a value that is randomly chosen from the contention window [1, $CW_{k,min}$] depending on the priority of the node. When the counter value becomes zero, frame is transmitted. If a collision occurs, the node goes to the next backoff stage with a new contention window, where the upper limit of the contention window is chosen as per the following rule: for i_{th} backoff stage of k^{th} priority node

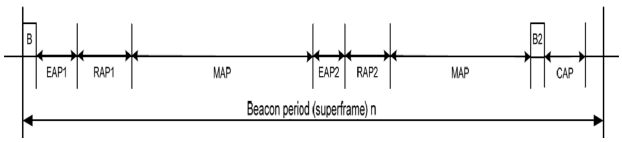

Figure 1. Superframe structure.

$$W_{k,i} = \begin{cases} CW_{k,min}, i = 0, \\ min(2W_{k,i-1}, CW_{k,max}), 2 \le i \le m_k, i \text{ even}, \\ W_{k,i-1}, 1 \le i \le m_k, \text{ i odd}, \\ W_{k,max}, m_k < i \le R. \end{cases} \quad (1)$$

During each backoff stage, the medium is sensed and the counter is decremented if the medium is idle. The counter is frozen if the medium is busy due to transmission, or if there is no sufficient time for the current frame transmission to finish before the end of the current access phase.

2.1 *Discrete time Markov chain (DTMC) model*

The access probabilities of UP_k nodes for $k \epsilon \{0, 1, \ldots, 7\}$ are computed by solving a set of eight DTMCs. The DTMC for priority k is shown in figure 2. The notations used are given in table 1.

The DTMC represents the backoff process of UP_k node and has stationary distribution $\{b_{k,i,j}\}$. The first index k shows the priority of the user, while the second index $i(i = 0, \ldots,$ R indicates the backoff stage the process is currently operating and the last index $j(j = 0, \ldots, W_{k,i})$ indicates the backoff counter value. We make the assumption that initiation of transmission of a packet and its subsequent completion does not extent over consecutive superframes. The channel is assumed to be ideal with no frame error because of forward error correction. Hence δ has a value equal to 1.

Since we consider non-saturation condition, it is possible that the queue becomes empty after successful transmission or dropping of a frame. This condition is given by $\pi_{k,0}$, which is the probability that the queue of the node is empty when a data frame is either successfully transmitted or dropped.

EAP is accessed by priority-7 frames only, while RAP can be accessed by frames of all priorities. EAP1 time slots are included in the Markov chain only for the highest priority nodes. Nodes with priorities 0,...,6 can access the medium only during RAP1 time slots.

Hence, probability that the medium remains idle during any slot in RAP1 is given by $\prod_{i=0}^{7}(1-\tau_i)^{n_i}$ and in EAP1 by $(1-\tau_7)^{n_7}$. The probability that the medium remains idle during the backoff countdown of a node with priority k where $k = 0, \ldots, 6$ is given by

$$q_k = \frac{\prod_{i=0}^{7}(1-\tau_i)^{n_i}}{(1-\tau_k)}. \quad (2)$$

Similarly, for a UP_7 node the corresponding probability is

$$q_7 = \frac{rap1}{rap1 + eap1} \frac{\prod_{i=0}^{7}(1-\tau_i)^{n_i}}{(1-\tau_7)} + \frac{eap1}{rap1 + eap1}(1-\tau_7)^{n_7-1} \quad (3)$$

where $rap1$ is the duration of RAP1 phase and $eap1$ is the duration of EAP1 phase.

$P_{k,idle}$ is the probability that a node is in idle state and the probability with which such a node in idle state starts a backoff is given by $\beta_k = (1-\pi_{k,0})q_k$.

Solving the Markov chain gives the steady-state probabilities for all possible states. They are given by the following equations. For $i = 0, \ldots, R$ and $j = 0, \ldots, W_{k,i}$

$$b_{k,i,j} = \frac{(W_{k,i} - j + 1)(1-q_k)^i}{W_{k,i} q_k} b_{k,0,0}, \quad (4)$$

$$b_{k,i,0} = (1-q_k)^i b_{k,0,0}, \quad (5)$$

$$b_{k,0,j} = \frac{W_{k,0} - j + 1}{W_{k,0} q_k} b_{k,0,0}, \quad (6)$$

$$P_{k,idle} = \frac{\pi_{k,0}}{(1-\pi_{k,0})q_k} b_{k,0,0}. \quad (7)$$

The stationary-state probabilities of the Markov chain must obviously add up to unity:

$$\sum_{i=0}^{R} \sum_{j=0}^{W_{k,i}} b_{k,i,j} + P_{k,idle} = 1, \quad (8)$$

$$\sum_{i=1}^{R} \sum_{j=1}^{W_{k,i}} b_{k,i,j} + \sum_{i=0}^{R} b_{k,i,0} + \sum_{j=1}^{W_{k,0}} b_{k,0,j} + P_{k,idle} = 1. \quad (9)$$

Using Eqs. (4)–(7) in the normalization equation (9) gives

$$\left\{ \frac{1}{2q_k} \sum_{i=1}^{R} (W_{k,i}+1)(1-q_k)^i + \frac{1-(1-q_k)^{R+1}}{q_k} + \frac{W_{k,0}+1}{2q_k} + \frac{\pi_{k,0}}{(1-\pi_{k,0})q_k} \right\} b_{k,0,0} = 1. \quad (10)$$

The channel access probability of a node of priority k is given by

$$\tau_k = \sum_{i=0}^{R} b_{k,i,0}. \quad (11)$$

Combining (5) and (11), we get

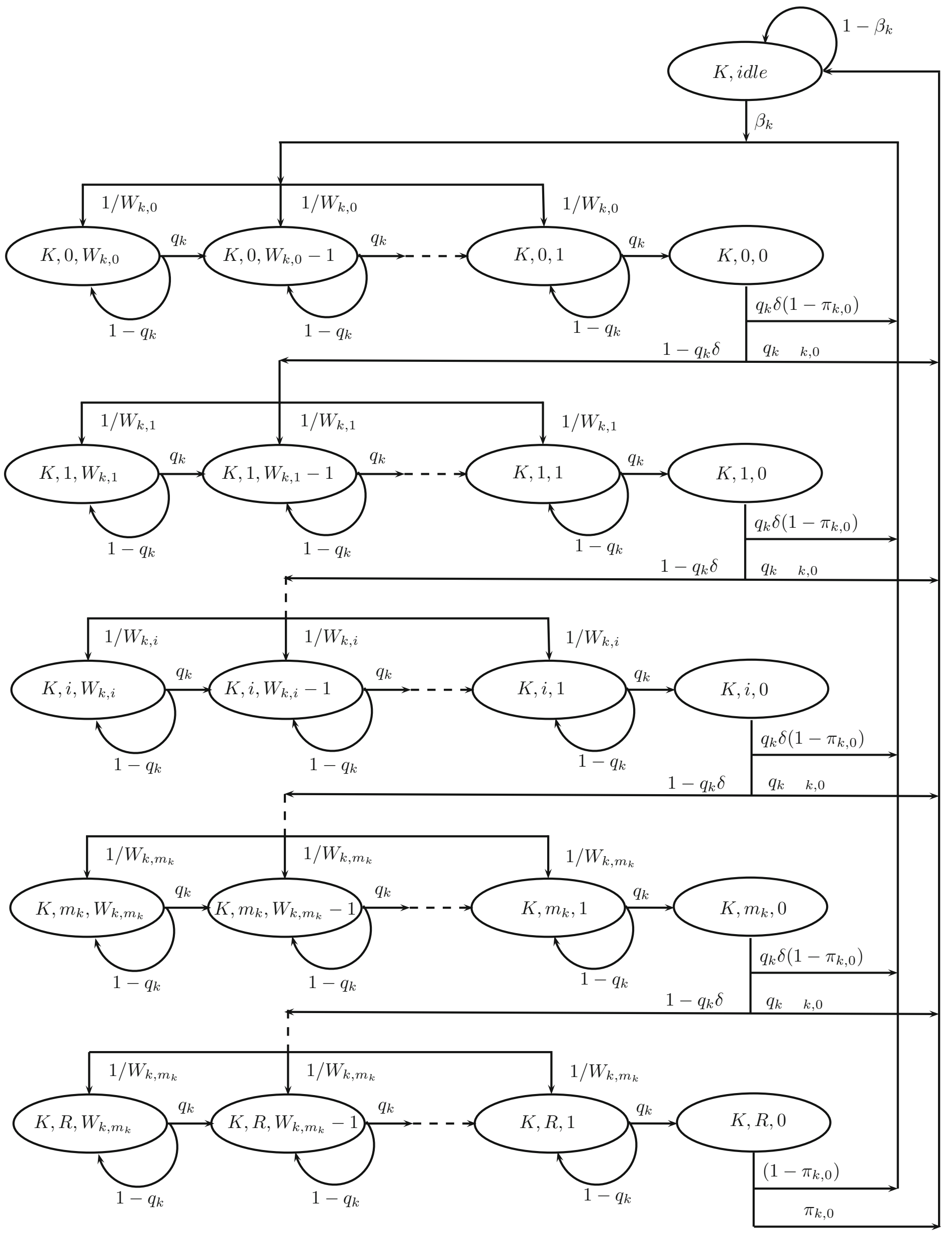

Figure 2. Markov chain for UP_k [13].

Table 1. Some variables used in Markov model.

Notation	Explanation
n_k	Number of UP_k nodes
τ_k	Access probability of UP_k
R	Maximum retry limit
$W_{k,i}$	Contention window of UP_k, backoff stage i
$1-\delta$	Packet error rate

$$\tau_k = b_{k,0,0}\left[\frac{1-(1-q_k)^{R+1}}{q_k}\right]. \tag{12}$$

Substituting Eq. (12) in Eq. (10), we get

$$\left\{\frac{1}{2q_k}\sum_{i=1}^{R}(W_{k,i}+1)(1-q_k)^i + \frac{1-(1-q_k)^{R+1}}{q_k} + \frac{W_{k,0}+1}{2q_k} + \frac{\pi_{k,0}}{(1-\pi_{k,0})q_k}\right\}\tau_k = \frac{1-(1-q_k)^{R+1}}{q_k}. \tag{13}$$

The probability that a queue would be empty after a successful data frame transmission or a data frame drop is given by

$$\pi_{k,0} = 1 - \lambda_k E[S_k] \tag{14}$$

where $E[S_k]$ is the average MAC layer frame service time (mean contention delay) and λ_k is the arrival rate of packets at node k. From Eqs. (13) and (14), we obtain 16 equations, which can then be solved to obtain the 16 unknown variables τ_k and $\pi_{k,0}$, for $k = 0, \ldots, 7$.

2.2 *Mean contention delay of a UP_k node*

The service time S_k for a data frame is the time elapsed from the instant the frame is put into service until the successfull delivery or drop due to the exceeding of the retry limit. It is the MAC layer contention delay. Its mean value is

$$E[S_k] = (1 - \pi_{k,0})E[S_{k,1}] + \pi_{k,0}E[S_{k,2}] \tag{15}$$

where $S_{k,1}$ and $S_{k,2}$ are the conditional service times, conditioned on the queue being non-empty and empty, respectively.

$$E[S_{k,1}] = p_s E[S_{k,S}] + p_d E[S_{k,D}] \tag{16}$$

where $S_{k,S}$ and $S_{k,D}$ are, respectively, the service times of successfully delivered frame and dropped frame; p_d and p_s are, respectively, the probability that a data frame is dropped due to an exceeding of the retry limit and the probability that a data frame is successfully delivered: $p_d = (1 - q_k)^{R+1}$ and $p_s = 1 - p_d$.

$$E[S_{k,S}] = E[B_{k,S}] + E[C_{k,S}] + T_s \tag{17}$$

where $E[B_{k,S}]$ is the mean backoff duration, $E[C_{k,S}]$ is the average time wasted in collision and T_s is the successful transmission time. Average backoff delay depends upon the number of successive retransmission attempts, the initial backoff value at each stage and the duration for which the backoff counter freezes due to the medium being busy. Let σ represent the time duration between successive counter decrements. Between two successive counter decrements the channel can be idle or it can be busy due to packet transmissions. Packet transmission can be successful or can result in collisions. We are considering the basic access mechanism and therefore we can assume collision time $T_c \approx T_s$. The average value of σ can then be expressed by the following equation:

$$\begin{aligned} E[\sigma] &= q_k + (1 - q_k)q_k(T_s + 1) + (1 - q_k)^2 q_k(2T_s + 1) + \cdots \\ &= 1 + \frac{1 - q_k}{q_k} T_s. \end{aligned} \tag{18}$$

The average number of counter decrements for the i^{th} backoff stage is $\frac{W_{k,i} + 1}{2}$. The probability that the frame is successfully transmitted after l^{th} retry is $(1 - q_k)^l q_k$. Hence, the average backoff delay is

$$E[B_{k,S}] = \sum_{l=0}^{R} (1 - q_k)^l q_k \sum_{i=0}^{l} \frac{W_{k,i} + 1}{2} \left(1 + \frac{1 - q_k}{q_k} T_s\right). \tag{19}$$

Average time wasted by the frame due to its collision in all the successive retransmission attempts is given by

$$\begin{aligned} E[C_{k,S}] &= \sum_{l=0}^{R} l(1 - q_k)^l q_k T_c \\ &= (1 - q_k)(1 - (1 - q_k)^R)T_c. \end{aligned} \tag{20}$$

The average time till the data frame is dropped is

$$E[S_{k,D}] = E[B_{k,D}] + E[C_{k,D}] \tag{21}$$

where $E[B_{k,D}]$ is the backoff that has occurred before the packet is dropped:

$$E[B_{k,D}] = \sum_{i=0}^{R} \frac{W_{k,i} + 1}{2} \left(1 + \frac{1 - q_k}{q_k} T_s\right). \tag{22}$$

$E[C_{k,D}]$ is the total collision time that has occurred before the packet dropping:

$$E[C_{k,D}] = (R + 1)T_c. \tag{23}$$

The successful transmission time $T_s = T_{data} + T_{SIFS} + T_{ACK} + 2\varphi$ and the collision time $T_c = T_{data} + T_{SIFS} + T_{DIFS} + \varphi$, where φ is the propagation delay.

If the data frame arrives when the node is in idle state, the node enters the zeroth backoff stage in the next CSMA slot. The residual time of the data frame in the idle slot will be at most the duration of a CSMA slot. Hence $E[S_{k,2}] \approx E[S_{k,1}]$. Substituting Eqs. (17) and (21) in Eq. (16) we get the mean MAC layer service time of a frame.

2.3 *Probability generating function of contention delay*

The probability generating function (pgf) of the service time of a packet, neglecting the residual slot time a packet sees on entering an idle queue, is given by

$$G_{S_k}(z) = p_s G_{S_{k,S}}(z) + p_d G_{S_{k,D}}(z). \tag{24}$$

$G_{S_{k,S}}(z)$ and $G_{S_{k,D}}(z)$ are, respectively, the pgf of service time of a successfully delivered frame and that of a dropped frame.

Let B_k be the total backoff time of a frame before it is successfully delivered or dropped, $N_{c,k}$ be the number of successive collisions that the frame has experienced, T_c the time wasted in each collision and T_s the transmission time. The pgf of service time of a frame when successfully transmitted is given by

$$G_{S_{k,S}}(z) := E[z^{B_k+N_{c,k}T_c+T_s}]. \tag{25}$$

Conditioning on $N_{c,k}$:

$$\begin{aligned} G_{S_{k,S}}(z) &= z^{T_s}E[E[z^{B_k+N_{c,k}T_c}/N_{c,k}]] \\ &= z^{T_s}\sum_{l=0}^{R} z^{lT_c}E[z^{B_k}/N_{c,k}=l]P(N_{c,k}=l). \end{aligned} \tag{26}$$

Let $G_{B_{k,l}} := E[z^{B_k}/N_{c,k}=l]$ be the pgf of backoff time with l successive collisions. Let D_i represent the backoff time during the i^{th} backoff stage. The conditional backoff time is

$$B_k/(N_{c,k}=l) = D_0 + D_1 + \cdots + D_l.$$

Since $D_0, \ldots, D_l$ are independent random variables, we have

$$G_{B_{k,l}} = G_{D_0}(z)G_{D_1}(z)\ldots G_{D_l}(z). \tag{27}$$

Let the backoff time at i^{th} stage be the sum of intervals of X_i successive backoff counter decrements. The conditional backoff time for the i^{th} stage is

$$D_i/(X_i = m) = \sigma_{1,i} + \sigma_{2,i} + \cdots \sigma_{m,i} \tag{28}$$

$\sigma_{1,i}$, $\sigma_{2,i}, \ldots, \sigma_{m,i}$ are iid random variables and are independent of the number of counter decrements m for backoff stage i. Hence

$$G_{D_i}(z) = G_{X_i}(G_\sigma(z)) \tag{29}$$

where $G_{X_i}(z)$ is the pgf of the number of counter decrements X_i for backoff stage i.

Since the initial value of backoff counter at backoff stage i is uniformly distributed in $[1, W_{k,i}]$,

$$G_{X_i}(z) = \frac{1}{W_{k,i}}\sum_{j=1}^{W_{k,i}} z^j = \frac{z}{W_{k,i}}\frac{(1-z^{W_{k,i}})}{(1-z)}. \tag{30}$$

The time interval between two successive counter decrements has the pgf given by

$$G_\sigma(z) = zq_k + z^{1+T_s}q_k(1-q_k) + \cdots = \frac{zq_k}{1-(1-q_k)z^{T_s}}, \tag{31}$$

$$G_{D_i(z)} = \frac{G_\sigma(z) - (G_\sigma(z))^{W_{k,i}+1}}{(1-G_\sigma(z))W_{k,i}}. \tag{32}$$

From Eqs. (27) and (32)

$$G_{B_{k,l}}(z) = \prod_{j=0}^{l}\frac{G_\sigma(z) - (G_\sigma(z))^{W_{k,j}+1}}{(1-G_\sigma(z))W_{k,j}}. \tag{33}$$

From Eqs. (26) and (33) the pgf of service time for a successfully transmitted data frame is given by

$$G_{S_{k,S}}(z) = z^{T_s}\sum_{l=0}^{R} z^{lT_c}\prod_{j=0}^{l}\frac{G_\sigma(z) - (G_\sigma(z))^{W_{k,j}+1}}{(1-G_\sigma(z))W_{k,j}}(1-q_k)^l q_k. \tag{34}$$

Similarly, the pgf of the service time of a dropped packet is

$$G_{S_{k,D}}(z) = z^{(R+1)T_c}\prod_{i=0}^{R}\frac{G_\sigma(z) - (G_\sigma(z))^{W_{k,i}+1}}{(1-G_\sigma(z))W_{k,i}}. \tag{35}$$

Substituting eqs. (34) and (35) in Eq. (24), we get the pgf of MAC layer service time of a frame. From the pgf, we obtain the mean and second moment as follows:

$$E[S_k] = \frac{\partial}{\partial z}G_{S_k(z)}|_{z=1}; E[S_k^2] = \frac{\partial^2}{\partial z^2}G_{S_k(z)}|_{z=1} + \frac{\partial}{\partial z}G_{S_k(z)}|_{z=1}.$$

3. CSMA/CA with sleep mechanism

To improve the energy efficiency and hence the device lifetime, the nodes can be made to sleep when there are no packets to be transmitted. However, the packet latency will increase if the node does not wake up promptly on the arrival of a packet to the empty queue. We consider a sleeping mechanism similar to the power saving mechanism of IEEE 802.16e [14]. A node goes to sleep when the queue becomes empty. After a fixed time, the node wakes up and checks if the queue is still empty. If so, it goes again to sleep; else, it begins transmitting packets until the queue becomes empty. This process is then repeated. This sleeping mechanism is followed by nodes of all priorities during the RAP and by nodes of UP_7 in both EAP and RAP. However, since nodes of $UP_k, k = 0, \ldots, 6$, cannot be transmited during EAP, they can be made to sleep for the entire EAP. A superframe with the sleeping periods is shown in figure 3, in which the checking of queue for packets and sleeping for a fixed duration is shown as T_W. The sleep time T_{eap} is not applicable for priority 7 nodes.

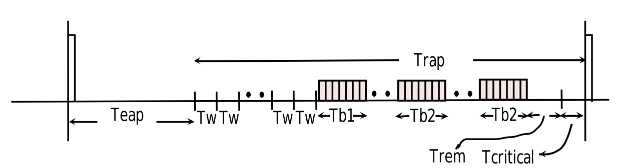

Figure 3. Superframe structure with sleep.

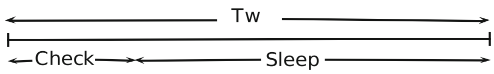

Figure 4. A sleep period.

T_W periods start from the beginning of the superframe itself for priority 7 nodes.

Each such period of duration T_W is made up of two parts (see figure 4). The first part represents the period during which the node wakes up and checks whether there are any packets in the buffer. If it does not find any packet in the buffer, the node goes to sleep immediately. The second part represents the sleep duration. On the other hand, if a packet is present in the buffer, the node enters a busy period. The node goes into backoff and when conditions become conducive the node transmits.

For the sake of analysis, we consider the transmission of packets as occurring in two phases. The first phase occurs when the node finds for the first time in a superframe that the buffer has at least one packet to be transmitted. The beginning of the first phase can occur at the end of EAP period or during the beginning of one of the T_W periods that follow EAP (as shown in figure 3) for nodes with priorities 0–6. In the case of priority 7 nodes, the first phase can begin at the end of first T_W period in a superframe or during the beginning of one of the subsequent T_W periods. The busy period represents the time during which it transmits packets collected in its buffer and the packets that arrive during the transmission of the packets stored in the buffer. The end of a busy period is the time at which the transmit buffer becomes empty. The sleeping of node, characterized by the T_W periods, then resume. The second phase begins when the node sees again a non-empty buffer on waking up at the beginning of a T_W period. The node transmits packets as in the first phase till the buffer becomes empty. This process is repeated until either enough time is not available in the current superframe period to transmit the packets accumulated in the buffer or the end of the superframe has reached. There can be multiple busy periods T_{b2} during the second phase of transmission. $T_{critical}$ is the fixed duration of time in a superframe where no transmission is possible. This time takes into account the guard time and some extra precautionary time. We make the assumption that once a busy period starts, its completion should occur within the same superframe. Busy periods therefore should not cross into the critical period nor span successive superframes. This requirement can result in situations where packets may be present in the buffer but the node is unable to transmit. Or there can be situations where during the current superframe there may be no more packets arriving after the last busy period. We define such time intervals by T_{rem}. T_{rem} is therefore the time left in the current superframe period apart from $T_{critical}$, during which time the node does not transmit and therefore can sleep. The value of T_{rem} is variable and is determined by the arrival rate and the congestion in the medium. Packets arriving during T_{rem} and $T_{critical}$ periods are transmitted in the next superframe.

The superframe duration is assumed to be long enough for a node to transmit packets accrued from the T_{rem} and $T_{critical}$ periods of the previous superframe and the packets arriving in the EAP period of the current superframe. If this were not the case, we would have packets piling up in the buffer over successive superframes.

4. Lifetime and latency analysis

We use an M/G/1 queue with repeated inhomogeneous server vacations [14] to model a node, in which the sleep durations of the node correspond to the vacations of the server. As mentioned earlier, in the first phase of transmission, the first sleep period, which comprises T_{rem} and $T_{critical}$ of the previous superframe and T_{eap} of the current superframe for nodes with priority 0–6, is different from the subsequent sleep durations, and hence we have inhomogeneous vacations. This is also the case with priority-7 nodes, where the first sleep period is composed of T_{rem} and $T_{critical}$ of the previous superframe, and the first T_W of the current superframe. In order to facilitate the analysis, we ignore the state transitions and beacon reception time at the beginning of each superframe. It is also to be noted that the first sleep period is defined as the time for which a node sleeps after the last busy period and the first time the node wakes up to look for packets in the buffer.

A superframe T_{sf} is made up of a sequence of subcycles $T_1, T_2, T_3, ..., T_N$. A typical subcycle has the representation shown in figure 5. During the beginning of periods T_{W1}, $T_{W2}, ...$ the node checks whether there are packets to be transmitted and goes to sleep if it does not find any. When the node finds packets at the beginning of $(V+1)^{th}$ T_W period in figure 5, it wakes up in $T_{wake-\mathrm{up}}$ period. Packets are transmitted during the busy period T_{b1}–T_{b2}. T_1 is the first subcycle, also called the primary subcycle, while T_2, $T_3, ..., T_N$ are secondary subcycles, which are iid random variables with the mean value $E[T']$.

$$E[T_2] = E[T_3] = \cdots = E[T_N] = E[T'], \tag{36}$$

$$T_{sf} = E[T_1] + (N-1)E[T'] + E[T_{rem}] + T_{critical}, N \geq 1. \tag{37}$$

4.1 *Primary subcycle T_1*

Primary subcycle T_1 is made up of sleep, wake-up and busy periods given by $\mathrm{E}[T_1] = \mathrm{E}[T_{sleep_1}] + T_{wake-\mathrm{up}} + \mathrm{E}[T_{busy_1}]$.

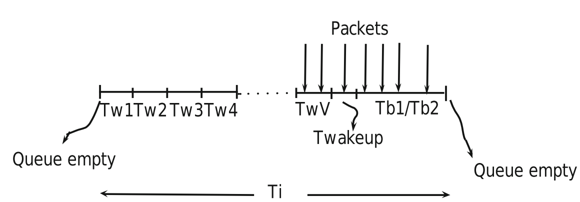

Figure 5. A typical subcycle.

$$\mathrm{E}[T_{sleep_1}] = \mathrm{E}[T_{W1} + T_{W2} + \cdots + T_{WV}].$$

We need to first compute the distribution of V, the number of successive vacations. It is observed that the event $V \geq i$ is equivalent to the event of no arrivals during $\sum_{k=1}^{i-1} T_{Wk}$. Since we assume Poisson arrivals with rate λ, $P(V \geq i) = \exp(-\lambda \sum_{k=1}^{i-1} T_{Wk})$. Denoting by $L_k(s) := E[\exp(-sT_{Wk})]$, the Laplace–Stieltjes transform (LST) of T_{Wk}, we have $P(V \geq i) = \prod_{k=1}^{i-1} L_k(\lambda)$ and we have from [14]:

$$E[T_{sleep_1}] = \sum_{i=1}^{\infty} E[T_{Wi}] \prod_{k=1}^{i-1} L_k(\lambda). \tag{38}$$

The first sleep period for the primary subcycle is given by

$$E[T_{W_1}] = T_{critical} + E[T_{rem}] + T_{eap} \tag{39}$$

for nodes with priorities 0–6 and

$$E[T_{W_1}] = T_{critical} + E[T_{rem}] + T_W \tag{40}$$

for nodes with priority 7. The subsequent sleep periods of primary subcycle is given by

$$E[T_{W_i}] = T_W, i = 2, 3, \ldots \tag{41}$$

The mean sleep time for the primary subcycle of UP_0–UP_6 is then obtained as

$$E[T_{sleep_1}] = (T_{critical} + E[T_{rem}] + T_{eap}) + e^{-(T_{critical}+E[T_{rem}]+T_{eap})\lambda} T_W \frac{1}{1 - e^{-T_W\lambda}}. \tag{42}$$

The busy period when packets are transmitted during the primary cycle is given by

$$E[T_{busy_1}] = \frac{\rho}{1-\rho}(E[T_{sleep_1}] + T_{wake-\mathrm{up}}), \tag{43}$$

where $\rho = \lambda E[S_k]$ and $E[S_k]$ is the mean service time of a packet obtained in section 3.

4.2 *Secondary subcycles* $T_2, T_3, \ldots, T_N$

The secondary subcycles are also made up of sleep, wakeup and busy periods:

$$E[T'] = E[T_{sleep_2}] + T_{wake-\mathrm{up}} + E[T_{busy_2}]. \tag{44}$$

Similar to the primary subcycle

$$E[T_{sleep_2}] = \sum_{i=1}^{\infty} E[T_{Wi}] \prod_{k=1}^{i-1} L_k(\lambda). \tag{45}$$

However, the secondary subcycles have the same fixed sleep period T_W, including the first sleep period.

$$E[T_{Wi}] = T_W, i = 1, 2, \ldots \tag{46}$$

The mean sleep time of the secondary subcycle is then obtained as

$$E[T_{sleep_2}] = T_W + T_W e^{-T_W\lambda} \frac{1}{1 - e^{-T_W\lambda}}. \tag{47}$$

The busy period of the secondary subcycle is given by

$$E[T_{busy_2}] = \frac{\rho}{1-\rho}(E[T_{sleep_2}] + T_{wake-\mathrm{up}}). \tag{48}$$

4.3 *Lifetime calculation of nodes with priorities 0–7*

Lifetime determination of nodes entails the calculation of several parameters, which are defined in table 2. We assume the system to be ergodic and the mean values are obtained as long-term averages values. The long-term time averages of T_{rem}, N_{sc}, N_{v1}, N_{v2}, T_{onIdle_1}, $T_{onActive_1}$, T_{onIdle_2}, $T_{onActive_2}$ and N_{frames} are found out using an iterative algorithm (see Algorithm 1). During each iteration, the values of these parameters are updated. N_{v1} and N_{v2} are necessary to find the energy consumed in the sleep–wakeup transitions. They are found out as in [14]:

$$N_{v1} = 1 + \frac{e^{-S_{time}\lambda}}{1 - e^{-T_W\lambda}} \tag{49}$$

$$N_{v2} = 1 + \frac{e^{-T_W\lambda}}{1 - e^{-T_W\lambda}} \tag{50}$$

where S_{time} stands for mean first sleep period of the primary subcycle. It is defined as

$$\begin{aligned} S_{time} &= T_{critical} + E[T_{rem}] + T_{eap}, k = 0, \ldots, 6 \\ &= T_{critical} + E[T_{rem}] + T_W, k = 7 \end{aligned} \tag{51}$$

$$T_{onIdle_1} = \frac{Nq_1}{1-\rho}\left[p_s E[B_{k,S}] + p_d E[B_{k,D}]\right] \tag{52}$$

$$T_{onIdle_2} = \frac{Nq_2}{1-\rho}\left[p_s E[B_{k,S}] + p_d E[B_{k,D}]\right] \tag{53}$$

$$T_{onActive_1} = \frac{Nq_1}{1-\rho}\left[p_s(E[C_{k,S}] + T_s) + p_d E[C_{k,D}]\right] \tag{54}$$

$$T_{onActive_2} = \frac{Nq_2}{1-\rho}\left[p_s(E[C_{k,S}] + T_s) + p_d E[C_{k,D}]\right] \tag{55}$$

where N_{q1} and N_{q2} are given by

$$N_{q1} = \lambda(T_{sleep1} + T_{wake-\mathrm{up}}) \tag{56}$$

$$N_{q2} = \lambda(T_{sleep2} + T_{wake-\mathrm{up}}). \tag{57}$$

Algorithm 1 Algorithm for computing the long-time averages of necessary parameters

Initialize:

$N_{frames} = 1$

$E[T_{rem}] = 0$

$T_{remPrev} = T_{rap}$

$S_{time1} = S_{time} - E[T_{rem}]$

if ($k = 7$)

$T_{eap} = 0$

end

$Compute\ E[T_{sleep_1}],\ E[T_{sleep_2}]$

$Compute\ N_{v2},\ N_{q2}$

$Compute\ T_{onIdle_2},\ T_{onActive_2}$

while abs($T_{remPrev} - E[T_{rem}]) > \varepsilon$

$Compute : N_{v1}\ N_{q1}$

$Compute : T_{onIdle_1},\ T_{onActive_1}$

$T_{remPrev} = E[T_{rem}]$

$T_{sleepCurrent} = E[T_{sleep_1}] - E[T_{rem}] - T_{critical}$

if $(T_{sf} < T_{sleepCurrent} + T_{wakeup} + T_{onIdle_1} + T_{onActive_1} + T_{critical})$

if $(T_{sleepCurrent} > T_{sf})$

$T_l = mod\ (T_{sleepCurrent}\ ,\ T_{sf})$

$N_{frames} = \frac{T_{sleepCurrent} - T_l}{Tsf}$

if $(T_l < T_{eap})$

$T_{sleepCurrent} = T_{sleepCurrent} - T_l + T_{eap}$

end

$S_{time1} = T_{critical} + N_{frames}\ T_{sf} + T_{eap}$

if $(T_l \geq T_{eap})$

if $((N_{frames} + 1)\ T_{sf} < T_{sleepCurrent} + T_{wakeup} + T_{onIdle_1} + T_{onActive_1} + T_{critical})$

$T_{sleepCurrent} = T_{sleepCurrent} - T_l + T_{sf} + T_{eap}$

$S_{time1} = T_{critical} + (N_{frames} + 1)\ T_{sf} + T_{eap}$

$N_{frames} = N_{frames} + 1$

end

end

$N_{frames} = N_{frames} + 1$

else

$S_{time1} = T_{sf} + T_{eap} + T_{critical}$

$T_{sleepCurrent} = T_{sf} + T_{eap}$

$N_{frames} = 2$

end

end

$Compute : T_{rem},\ N_{sc}$

$Compute : S_{time},\ T_{sleep1}$

end

Table 2. Parameters used in the algorithm.

S_{time}	Mean first sleep period of the primary subcycle
S_{time1}	First sleep period of primary subcycle excluding T_{rem}
T_{sleep_1}	Total sleep time for the primary subcycle
T_{sleep_2}	Total sleep time for a secondary subcycle
N_{v1}	Mean number of sleep periods in the primary subcycle with fixed duration (T_W)
N_{v2}	Mean number of sleep periods in a secondary subcycle
N_{q1}	Mean number of packets arriving during sleep time and wake-up time of primary subcycle
N_{q2}	Mean number of packets arriving during the sleep and wake-up time of secondary subcycles
T_{onIdle_1}	Mean backoff time during the busy period of primary subcycle
$T_{onActive_1}$	Mean transmission time during the busy period of primary subcycle
T_{onIdle_2}	Mean backoff time during the busy period of secondary subcycle
$T_{onActive_2}$	Mean transmission time during the busy period of secondary subcycle
$T_{sleepCurrent}$	Time slept in the current superframe during the primary subcycle
N_{sc}	Number of secondary subcycles
N_{frames}	Mean number of consecutive superframes through which it sleeps + 1
$T_{remPrev}$	Mean value of T_{rem} in the previous iteration

$T_{sleepCurrent}$ is a parameter used within the algorithm to find out the number of superframes a node sleeps continuously. The algorithm converges when the updated value of T_{rem} during successive iterations becomes invariant. During each iteration T_{rem} is computed as follows:

$$T_{rem} = mod((N_{frames}T_{sf} - T_{sleepCurrent} - T_{wake-\mathrm{up}} - T_{onIdle_1} - T_{onActive_1} - T_{critical}), (T_{sleep_2} + T_{wake-\mathrm{up}} + T_{onIdle_2} + T_{onActive_2})). \quad (58)$$

N_{sc} is also updated in each loop using the expression given below.

$$N_{sc} = ((N_{frames}T_{sf} - T_{sleepCurrent} - T_{wakeup} - T_{onIdle_1} - T_{onActive_1} - T_{critical}) - T_{rem})/(T_{sleep_2} + T_{wakeup} + T_{onIdle_2} + T_{onActive_2}). \quad (59)$$

If sleep time ends within the EAP period, the sleep time gets extended to the end of EAP period. The transmission in the primary subcycle then starts at the beginning of the RAP period. On the other hand, if the sleep ends inside the RAP period, we see whether the transmission can complete within the current RAP period. If not, the sleep period gets extended to the end of the EAP period of the next superframe, and busy period of the primary subcycle starts when it wakes up. We then find out T_{rem} and N_{sc}. We update the values of the total sleep time of the primary subcycle and the cycle repeats. The same procedure is valid for priority-7 nodes except that checking of packets starts in the EAP period. Consequently T_{eap} is made zero; ε is taken as 0.000001.

4.4 *Energy consumption by nodes*

$$Esleep_1 = (T_{eap}(N_{frames}) + ((T_{sf} - T_{eap})\frac{(N_{frames}-1)}{T_W} + (N_{v1}-1))(T_W - T_h) + T_{rem} + T_{critical})P_{sleep}, \tag{60}$$

$$Elisten1_1 = ((T_{sf} - T_{eap})(N_{frames}-1)(T_{sleepRx}/T_W) + (N_{v1}-1)T_{sleepRx})P_{sleepRx}, \tag{61}$$

$$Elisten1_2 = ((T_{sf} - T_{eap})(N_{frames}-1)(T_{rxSleep}/T_W) + (N_{v1}-1)T_{rxSleep})P_{rxSleep}. \tag{62}$$

$Elisten_1$ is the total energy consumed while checking the presence of packets in buffer in primary subcycle (table 3). Hence

$$Elisten_1 = Elisten1_1 + Elisten1_2, \tag{63}$$

$$Eidle_1 = (T_{onIdle_1})Pidle, \tag{64}$$

$$Eactive_1 = (T_{active_1})Pactive. \tag{65}$$

Total energy consumed by the node during primary subcycle:

$$Etot_1 = (Esleep_1 + Elisten_1 + Eidle_1 + Eactive_1). \tag{66}$$

Similarly for the secondary subcycles

$$Esleep_2 = N_{sc}(Tsleep_2 - (N_{v2}-1)T_h)P_{sleep}, \tag{67}$$

$$Elisten_2 = N_{sc}((N_{v2}-1)(T_{sleepRx}P_{sleepRx} + T_{rxSleep}P_{rxSleep}) + T_{sleepRx}P_{sleepRx}), \tag{68}$$

$$Eidle_2 = (T_{onIdle_2})Pidle, \tag{69}$$

$$Eactive_2 = (T_{active_2})Pactive. \tag{70}$$

Total energy consumed by the node during secondary subcycles:

$$Etot_2 = (Esleep_2 + Elisten_2 + Eidle_2 + Eactive_2). \tag{71}$$

Lifetime of a node (in days) is then calculated as follows:

$$Lifetime = \frac{Eb \times N_{frames} \times T_{sf}}{(Etot_1 + Etot_2) \times 60 \times 60 \times 24}. \tag{72}$$

4.5 *Mean packet delay*

The mean delay for packets transmitted during the primary and secondary subcycles are computed using equations in [14]. In using those equations in [14], we first derived corresponding entities in our model, the details of which are skipped.

Thus, we get the mean delay for a packet transmitted in the primary subcycle follows:

$$D_1 = \frac{\lambda}{2}\frac{\frac{1}{\lambda} - E[S_k]}{(S_{time} + T_{wakeup}) + e^{-S_{time}\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}} \times \left(S_{time}^2 + \frac{T_W^2 e^{-S_{time}\lambda}}{1-e^{-T_W\lambda}}\right) + T_{wakeup}\frac{\left(S_{time} + \frac{T_{wake-\mathrm{up}}}{2}\right) + e^{-S_{time}\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}}{(S_{time} + T_{wake-\mathrm{up}}) + e^{-S_{time}\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}} + \rho\frac{\left(S_{time}^2 + \frac{T_W^2 e^{-S_{time}\lambda}}{1-e^{-T_W\lambda}}\right)}{2\left((S_{time} + T_{wake-\mathrm{up}}) + e^{-S_{time}\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}\right)} + E[S_k] + \frac{\lambda E[S_k^2]}{2(1-\rho)}. \tag{73}$$

Similarly, mean packet delay for secondary subcycle is obtained as follows:

$$D_2 = \frac{\lambda}{2}\frac{\frac{1}{\lambda} - E[S_k]}{(T_W + T_{wakeup}) + e^{-T_W\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}}\left(\frac{T_W^2}{1-e^{-T_W\lambda}}\right) + T_{wakeup}\frac{T_W + \frac{T_{wakeup}}{2} + e^{-T_W\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}}{(T_W + T_{wakeup}) + e^{-T_h\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}} + \frac{\rho\frac{T_W^2}{1-e^{-T_W\lambda}})}{2\left(T_W + T_{wakeup} + e^{-T_W\lambda}T_W\frac{e^{T_W\lambda}}{e^{T_W\lambda}-1}\right)} + E[S_k] + \frac{\lambda E[S_k^2]}{2(1-\rho)}. \tag{74}$$

Finally, mean packet delay averaged over both primary and secondary subcycles is obtained as follows:

Table 3. Parameters used for energy calculation.

$P_{rxSleep}$	Power consumed while changing from receive to sleep state
$P_{sleepRx}$	Power consumed while changing from sleep to receive state
P_{idle}	Power consumed during backoff/receive state
P_{active}	Power consumed during transmit state
$T_{sleepRx}$	Transition time from sleep to receive state
$T_{rxSleep}$	Transition time from receive to sleep state
T_h	$T_{sleepRx} + T_{rxSleep}$
$Esleep_1$	Energy spent during sleep for primary subcycle
$Elisten1_1$	Energy spent for sleep-to-receive transitions for primary subcycle
$Elisten1_2$	Energy spent for receive-to-sleep transitions for primary subcycle
$Eidle_1$	Energy spent on idle listening during backoffs for primary subcycle
$Eactive_1$	Energy spent on transmissions for primary subcycle
$Esleep_2$	Energy spent on sleep by the node for secondary subcycle
$Elisten_2$	Energy spent on wake-up's for secondary subcycle
$Eidle_2$	Energy spent on idle listening during backoffs for secondary subcycle
$Eactive_2$	Energy spent on transmissions for secondary subcycle

Table 4. Nodes and their parameters.

UP	Node	NN	PR	PS
0	ECG	1	2p/s	100B
0	EEG	1	2p/s	100B
1	ECG	1	2p/s	100B
1	Blood pressure	1	2p/s	100B
2	EEG	1	2p/s	100B
2	EEG	1	2p/s	100B
3	Glucose	1	1p/s	100B
3	Oxygen saturation	1	1p/s	100B
4	EMG	1	1p/s	100B
4	EMG	1	1p/s	100B
5	Temperature	1	1p/s	100B
5	Respiration rate	1	1p/s	100B
6	ECG	1	0.25p/s	100B
6	ECG	1	0.25p/s	100B
7	ECG	1	0.5p/s	100B
7	ECG	1	0.5p/s	100B

UP: user priority, NN: number of nodes, PR: packet rate, PS: payload size (byte).

$$D_{avg} = \frac{N_1^{'} D_1 + N_{sc} N_2^{'} D_2}{N_1^{'} + N_{sc} N_2^{'}} \tag{75}$$

where $N_1^{'}$ is the mean number of packets arriving during the sleep and busy time of the primary subcycle and $N_2^{'}$ is the mean number of packets arriving during the sleep and busy time of the secondary subcycle.

Table 5. WBAN traffic and priorities.

UP	CW_{min}	CW_{max}	Traffic
0	16	64	Background
1	16	32	Best effort
2	8	32	Excellent
3	8	16	Video
4	4	16	Voice
5	4	8	Medical data
6	2	8	High-priority medical data
7	1	4	Emergency report

Table 6. Path loss parameters.

d_0	1.0 m
$PL(d_0)$	55 dBm
η	2.4
sd	4 dB

5. Simulation results

The analytical results are validated via simulation studies using the Castalia simulator. Variations in the results are then explained. The results reported are for a WBAN with 16 nodes and Poisson packet arrival process at each node with rate λ that has values as specified in table 4.

The nodes are given user UPs ranging from 7 to 0, where 7 denotes the highest UP and 0 the lowest UP. Each priority is associated with a characteristic contention window range shown in table 5.

Apart from the 16 nodes used, there is also a hub whose function is to collect information transmitted by the nodes. The nodes are placed on different parts of the body. The hub is assumed to be awake all the time. Hub can have energy that can be replenished without much difficulty.

IEEE 802.15.6 specifies three different physical (PHY) layers, namely narrow band (NB), ultra-wide band (UWB) and human body communications (HBC). For this study, the NB PHY is considered, specifically, ISM 2.4 GHz.

The model CM3 A, for WBAN, is taken as the channel model. The path loss is given by

$$PL(d) = PL(d_0) + 10\eta \log\frac{d}{d_0} + X_{sd} \tag{76}$$

where d is the distance between a node and the hub in metres. $PL(d_0)$ is known path loss at a reference distance d_0, η is path loss coefficient and X_{sd} is a Gaussian random variable, with zero mean and standard deviation equal to sd. The values for different parameters of the path loss model are given in table 6. The modulation scheme used in the transceiver is DQPSK with noise floor equal to −87 dBm. The buffer size for the packets is 1000 byte, which means it can hold at the most 10 packets at any given time.

Table 7. Simulation parameters.

CSMA slot	125 µs
$T_{critical}$	1 ms
Application payload per packet	100 byte
Allocation slot length	1 ms
Superframe length	150 ms
Battery capacity	560 mAh

Table 8. Transceiver characteristics.

Tx–Rx , Rx–Tx (transition time)	0.02 ms
Rx–Sleep, Tx–sleep (transition time)	0.194 ms
Sleep–Rx, sleep–Tx (transition time)	0.05 ms
Transmit power level	−10 dBm
Tx (power consumed)	3 mW
Rx (power consumed)	3.1 mW
Tx–Rx, Rx–Tx (power consumed)	3 mW
Sleep–Rx, sleep–Tx (power consumed)	1.5 mW
Rx–sleep, TX–sleep (power consumed)	1.5 mW
Sleep power level	0.05 mW

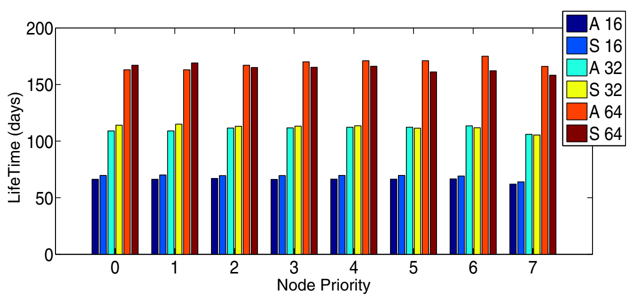

Figure 6. Lifetime of UP_0–UP_7 nodes for EAP/T_{SF} = 0.1; A:= analytical, S:= simulation.

Simulation parameters chosen for the study are as shown in table 7 and the parameters assumed for the transceiver are listed in table 8.

The focus of the simulation studies is to find the performance of the network with respect to lifetimes and latencies and to compare it with the analytical results. For most WBAN applications, emergency nodes generate information occasionally, which in turn justifies a small EAP fraction compared with RAP. Increasing the EAP time interval in a superframe will only serve to increase the average delay of the lower priority nodes.

Latency of a packet is the time from the moment it arrives at the node's input buffer to the time when it is successfully received at the hub. Lifetime of a node is determined by finding the energy spent on a superframe, and then finding the number of superframes it can live through for a given battery capacity.

The simulation study is conducted by assigning different fixed values for the time interval, T_w, during which a node sleeps before examination of the presence of packets. The sleep intervals used are 16, 32 and 64 CSMA slots.

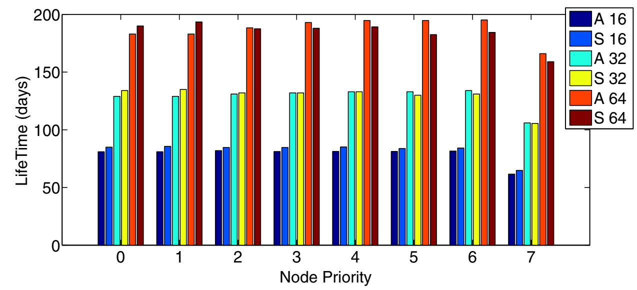

Figure 7. Lifetime of UP_0–UP_7 nodes for EAP/T_{SF} = 0.3; A:= analytical, S:= simulation.

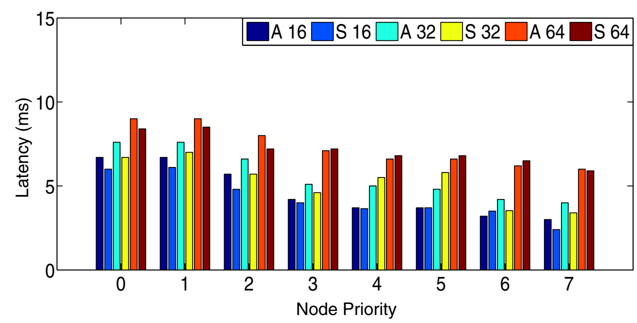

Figure 8. Latencies of UP_0–UP_7 nodes for EAP/T_{SF} = 0.1; A:= analytical, S:= simulation.

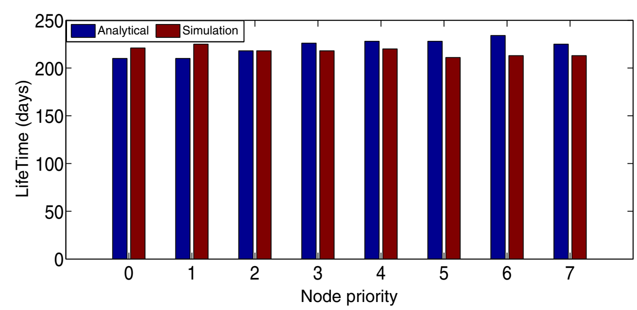

Figure 9. Lifetime of UP_0–UP_7 nodes for $EAP/T_{SF} = 0.1$ and T_W = EAP.

Figures 6 and 7 show the lifetimes of nodes for EAP/T_{SF} equal to 0.1 and 0.3, respectively. The figures show fairly good match between analytical and simulation values. In each case, it can be seen that the lifetimes are fairly same for nodes of all priorities for a particular T_w except for the highest-priority nodes, which have slightly lesser values. This is understandable, since highest-priority nodes can be active in the EAP periods, while nodes with other priorities sleep during this period and sleep times in the RAP period do not differ much because of the low arrival rate of data packets. Lifetimes show good increase as T_w increases. The nodes can sleep for longer times with lesser number of wake-ups as T_w increases. In fact increasing T_W has a greater effect on the lifetimes of nodes compared with an increase in EAP/T_{SF} ratio. The latencies of nodes also show

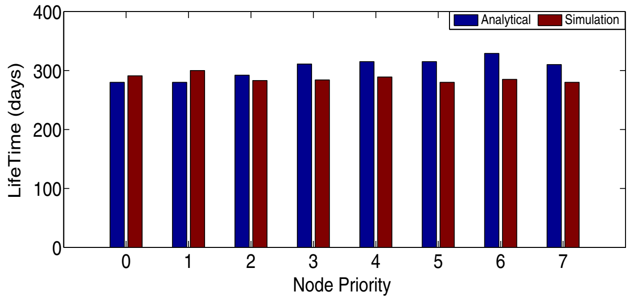

Figure 10. Lifetime of UP_0–UP_7 nodes for $EAP/T_{SF} = 0.3$ and T_W = EAP.

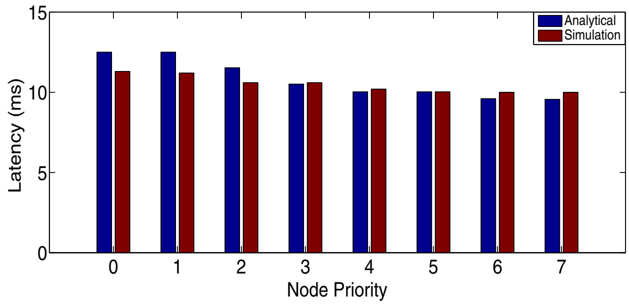

Figure 11. Latencies of UP_0–UP_7 nodes for EAP/T_{SF} = 0.1 and T_W = EAP.

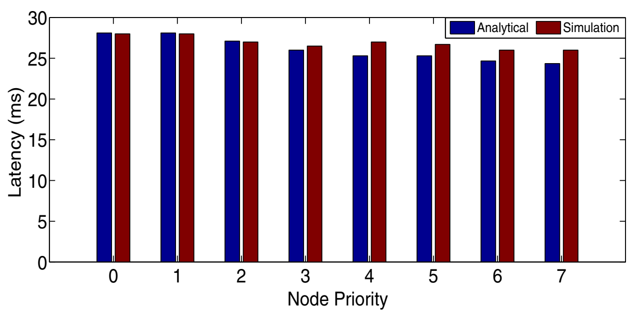

Figure 12. Latencies of UP_0–UP_7 nodes for EAP/T_{SF} = 0.3 and T_W = EAP.

a fairly good match between analytical and simulation values (see figure 8).

It would be worthwhile to see the behaviour of nodes with respect to lifetimes and latencies when T_W is set to EAP (figure 9). The lifetimes of nodes hover around the 300 days mark when EAP/T_{SF} becomes 0.3 as shown in figure 10.

The latencies increase with EAP/T_{SF}, but values are comparable, irrespective of the priorities. Obviously, as T_W increases, the latencies increase (figures 11 and 12).

It can be deduced, from the results of simulation and analytical studies performed for finding out the latencies and lifetimes of nodes, that T_W can be used as a parameter to control the lifetimes/latencies of devices according to our requirements for a given EAP/T_{SF} ratio. Furthermore, EAP/T_{SF} itself can be used as a measure to fine-tune our latency and lifetime requirements.

The net result boils down to the question of determining which parameter is more important for a particular WBAN application and choosing the appropriate sleep mechanism and parameters.

6. Conclusion

The simulation studies show that the analytical results for latencies and lifetimes match with simulation results to a good extent for EAP/T_{SF} values chosen, thereby validating the analytical model. The interrupted sleeping model produces a low latency value for nodes of all priorities. T_W can be used as a parameter to control the lifetimes and latencies of nodes according to the requirements of the applications. At the same time, nodes sleeping during the absence of packets in the transmit buffer help increase the lifetimes of devices.

References

[1] Jovanov E, Milenkovic A, Otto C and Groen P 2005 A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *J. Neuroeng. Rehabil.* 2(6): 16–23

[2] IEEE Draft 2008 Draft health informatics—point-of-care medical device communication—guidelines for the use of RF wireless technology. Technical Report, Std P11073-00101/D5

[3] Boulis A, Smith D, Miniutti D, Libman L and Tselishchev Y 2012 Challenges in body area networks for healthcare: the MAC. *IEEE Commun. Mag.* 5: 2–8

[4] IEEE 2012 IEEE Standard for Local and Metropolitan Area Networks Part 15.6-: Wireless Body Area Networks. IEEE Computer Society, New York

[5] Ullah S and Kwak K S 2011 Throughput and delay limits of IEEE 802.15.6. In: *Proceedings of IEEE WCNC 2011*

[6] Boulis A and Tselishchev Y 2010 Contention vs. polling: a study in body area networks MAC design. In: *Proceedings of ICST 2010*

[7] Tachtatzis C, Franco F D, Tracey D C, Timmons N F and Morrison J 2010 An energy analysis of IEEE 802.15.6 scheduled access modes. In: *Proceedings of IEEE Globecom 2010 Workshop on Mobile Computing and Emerging Communication Networks*

[8] Tachtatzis C, Franco F D, Tracey D C, Timmons N F and Morrison J 2011 An energy analysis of IEEE 802.15.6 scheduled access modes for medical applications. In: *Proceedings of ADHOCNETS 2011*

[9] Rashwand S, Misic J and Khazaei H 2011 IEEE 802.15. 6 under saturation: some problems to be expected. *J. Commun. Netw.* 13(2): 142–148

[10] Rashwand S and Misic J 2011 Performance evaluation of IEEE 802.15. 6 under nonsaturation condition. In:

Proceedings of the Global Telecommunications Conference (GLOBECOM 2011)

[11] Motoyama S 2012 Flexible polling-based scheduling with QoS capability for wireless body sensor network. In: *Proceedings of the 8th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks, 2012, Florida*

[12] Ibarra E, Antonopoulos A, Kartsakli E and Verikoukis C 2015 HEH-BMAC: hybrid polling MAC protocol for WBANs operated by human energy harvesting. *Telecommun. Syst*. 5(2): 111–124

[13] Jacob A K, Kishore G S and Jacob L 2015 Contention versus polling access in IEEE 802.15.6: delay and lifetime analysis. In: *Proceeding of the National Conference on Communication (NCC-2015), IIT Bombay*

[14] Alouf S, Altman E and Azad A P 2008 *Analysis of an M/G/1 queue with repeated inhomogeneous vacations with application to IEEE 802.16e power saving mechanism*. Research Report RR-6488, *INRIA*