International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

Reinforcement Learning based Node Sleep or Wake-up Time Scheduling Algorithm for Wireless Sensor Network

Parag Verma
Department of Computer Science and Engineering,
Uttaranchal University, Dehradun, Uttarakhand, India.
E-mail: parag_verma@yahoo.com

Ankur Dumka
Department of Computer Science and Engineering,
Graphic Era Deemed to be University, Dehradun, Uttarkahnd, India.
E-mail: ankurdumka2@gmail.com

Dhawal Vyas
Department of Computer Science and Engineering,
Government Engineering College, Bharatpur, Rajasthan, India.
Corresponding author: d2vyas@gmail.com

Anuj Bhardwaj
Department of Computer Science and Engineering,
Chandigarh University, Chandigarh, Punjab, India.
E-mail: anuj2k3@gmail.com

(Received December 30, 2019; Accepted March 25, 2020)

Abstract
A wireless sensor network is a collection of small sensor nodes that have limited energy and are usually not rechargeable. Because of this, the lifetime of wireless sensor networks has always been a challenging area. One of the basic problems of the network has been the ability of the nodes to effectively schedule the sleep and wake-up time to overcome this problem. The motivation behind node sleep or wake-up time scheduling is to take care of nodes in sleep mode for as long as possible (without losing data packet transfer efficiency) and thus extend their useful life. This research going to propose scheduling of nodes sleeps and wake-up time through reinforcement learning. This research is not based on the nodes' duty cycle strategy (which creates a compromise between data packet delivery and nodes energy saving delay) like other existing researches. It is based on the research of reinforcement learning which gives independence to each node to choose its own activity from the transmission of packets, tuning or sleep node in each time band which works in a decentralized way. The simulation results show the qualified performance of the proposed algorithm under different conditions.

Keywords- Sleep or wake-up scheduling, Wireless sensor network, Sensor node energy.

1. Introduction

Due to subsequent technological advancement, sensors nodes (SNs) integrated with tiny sized, low power consumption, the minimal cost with high efficiency have become possible technically and monetary. These SNs are usually equipped with detection, information handling and correspondence components. These sensors can be used to determine the state of the environment that cover by SNs and then transform these estimated data into signals. The signal can be managed to search for properties on target features located in the area of the SNs. The SNs at that point send this data, mostly through a radio transmitter to a centralized system known as a 'sink

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

node' or 'base station', directly or through some delivery sensors (Xiao et al., 2011). A huge quantity of such sensors arranged in a network form for many applications that require inaccessible activity, thus creating a wireless sensor network (WSN). In present technological world WSNs have different usages, including objective detecting and monitoring (Zhu et al., 2013), social security health care monitoring (Acampora et al., 2013), classification of data (Yao et al., 2015), distributing computing (Li et al., 2013; Zhao et al., 2016) and security observations (Fu et al., 2014; Semnani and Basir, 2014).

Usually, a WSN contains multiple of hundreds or thousands of SNs that can communicate with each other (Ye et al., 2002). The SN has limited energy resource and generally, they are not rechargeable, so the use of the energy of each SN must be minimized to extend the survival time of the WSNs. Key sources of wastage of energy are passive tuning, collision, overhearing and overhead control (Ye et al., 2006). Among these, passively tuning is a major factor in most sensor network applications (Liu, 2015). There are a few different methods to extend the survival time of WSNs, for example, effectively organize of SNs (Chen et al., 2014), rationalization of inclusion of WSNs (Guo et al., 2011) and sleep or wake-up time scheduling methodology (Wei et al., 2011). Through this research work, our attention is to schedule the sleep or wake-up time of SNs. Sleep or wake-up time scheduling, which hopes to minimize the passive listening time, is one of the central research topics in WSN (Ye et al., 2004).

Especially, research on sleep or wake-up time scheduling shows how to change the ratio between the sleep time and the conscious time of each SN in each predefined time slot. At the moment a SN is awake; it is in a passive listening state and can be received and transfer data packets. In any case, if the data packet is not received or transferred during the passive listening time slot, the energy used during passive listening is misused. Such misused energy should be completely minimized by changing the SN conscious time, which is the sleep or wake-up time scheduling. More recently, several sleep or wake-up time scheduling methods have been created (Sun et al., 2008; Lai et al., 2010; Wei et al., 2011). These methodologies are usually divided into three classifications:

- an on-demand awaking approach,
- synchronous awaking approach and,
- the unconventional approach to awaking.

On-demand awaking approaches (Cao et al., 2005), out-of-band signalling is used to activate sleeping SNs upon request. For example, a SN activates on listening to a page channel with the help of a paging signal. Since page radio can operate with less power usage, this method is very effective for energy. However, it experiences the unpredictability of extended use. In the synchronous activation method (Zheng et al., 2003; Keshavarzian et al., 2006), the sleeping SNs sometimes arise together to communicate with each other. Such a method should synchronize neighbouring nodes to accommodate their conscious or sleep time. Neighbouring nodes begin exchanging data packets within regular active times, which allows a node to sleep most of the time within the operating cycle without losing any incoming data packet. The synchronous activation method can completely reduce passive listening time; however, the synchronization required presents an additional overhead and a multidimensional nature. Also, a SN may need to be awake at different times during the complete sleep or wake-up time slot, if its neighbours are on multiple schedules.

In the asynchronous activation approach (Polastre et al., 2004; Jang et al., 2013), each sensor node tracks its activation scheme for the passive state. This entails that waking provisions be covered between neighbours. To achieve this prerequisite, nodes have to be awake more often than most synchronous activation approaches. The focal points offered by asynchronous activation approaches also confirm ease of execution, low message load for correspondence, and deep assertion of system availability.

Recent research work uses the duty cycling method to periodically switch between sleep and awaking state (Lai et al., 2010; Wei et al., 2011). Here, the duty cycle is the ratio between the duration of activation time in a predefined epoch and the total duration of that epoch (Ye et al., 2006). For example, suppose an epoch is 1 second and the connecting node remains alert for 0.45 seconds and sleeps for 0.55 seconds in duration. At that point, the duty cycle is 45% (or 0.45). The use of duty cycle increases compensation between energy savings and delayed data packet transport (Kim et al., 2009): a longer activation time can waste energy, while a shorter activation time delay may cause data package transport. In wireless sensor network both energy savings and delayed data packet transport are vital. Since each node in a wireless sensor network is generally equipped with a battery that does not work as a rechargeable source of energy, life-saving power is important to boost the life of the WSN. Since delays are not satisfactory in some uses of WSNs, for example, fire location, earthquake and tsunami precaution, a reduction in delay in the transmission of data packets is critical to the adequacy of WSNs.

An instinctive response to this compensation is to powerfully decide the duration of wake time. The provisions proposed in (Tang et al., 2011) set the duration of activation time by transmitting all messages in explosions of different lengths and sleep between explosions. This provision may save energy, however as it may amplify a delay in data packet transport, in light of the fact that each node has the energy to connect the data package on its own line before transmitting these data packets in an explosion. Another provision of this problem is proposed in (Tang et al., 2011), it empowers senders to estimate receivers' activation time using a pseudo-arbitrary activation approach. Later, if senders have data packets to transmit, senders can quickly wake up before receivers' initial wake-up time, so the life force energy, which senders use for passive tuning or idle listening, can be saved. For this situation, senders are not required to compensate, on the basis that their activation program is solely based on the receivers' activation times. Receivers still face a commitment, in any case, since the receiver's activation time depends on a pseudo-arbitrary activation scheduling function and many parameter determinations in this function will generate multiple activation provisions. Furthermore, before a sender can expect about the activation time of the receiver, the sender must demand parameters in the receiver's activation scheduling function. This request generates the use of an additional vitality.

Through this research paper, a reinforcement learning-based node sleep or wake-up time scheduling method is proposed, which reflects both the saving of SNs energy and delayed data packet transmission. This scheduling is covered under an asynchronous method and does not use the method of the duty cycle. In this way, compensation between significant energy savings and delayed data package transmission can be avoided. In most node sleep or wake time scheduling methods that are based on the duty cycling methods, the node access time is separated into time periods, each of which has few slots.

In each respectively time period, the nodes modify their sleep and awakening times, that is, they change the duty cycle where each node maintains awaking state in some defined time slots and

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

sleep state in rest time slots. In this proposed node sleep or wake-up time scheduling method, the node access time is validly divided into time slots. In each of the time slot, each node chooses to sleep or wake up independently. In this way, in the proposed methodology, there is no scope of 'duty cycle' and each time slot schedule is autonomous. Figure 1 usually shows how the proposed methodology works.

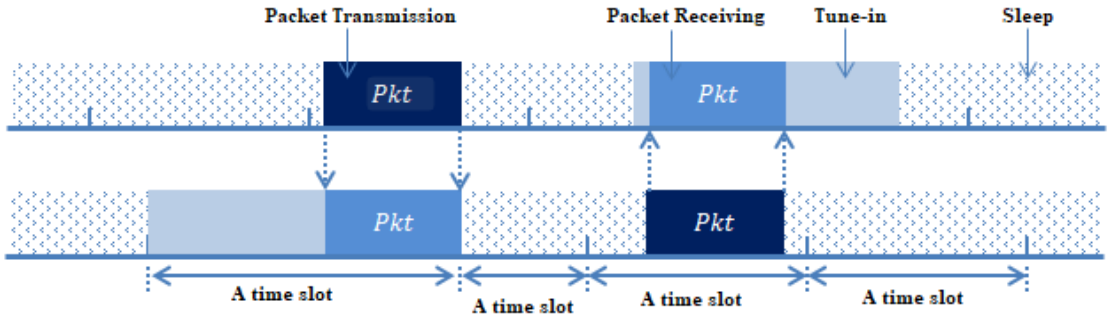


Figure 1. Overview of the proposed methods

From, Figure 1, A and B are two neighbouring nodes whose timekeeper cannot be synchronized. They decide from the beginning of each time slot independent and autonomous manner without the exchange of data. There are two facts in the figure that should be taken into consideration. For the recipient to begin with, if the scheduled time slot to receive a data package is not long, the period of availability will be extended until the package is effectively received (from the node B considering first time slot). Second, when a node chooses to transfer a packet in the current available time slot and the scheduled time slot exceeds the time required to transmit a packet, the node will also choose when to transfer the packet in the current scheduling time slot (from the node B considering third time slot).

This proposed methodology is not structured specifically for the packet routing protocol. It assumes that the node sleep or wake-up time scheduling approach is structured by linking a particular packet routing protocol, this scheduling method can only work with that routing protocol but it may not work as efficient with other routing protocols. For example, the nodes sleep or wake time scheduling approach in (Wei et al., 2011) is a strategic fusion with a packet routing protocol. Its scheduling methodology uses amazing awaking scheduling to create unidirectional forms of data transmission for the dissemination of information to essentially reduce the inactivity of the data gathering process. Its methodology works quite well if packets are moved in the prescribed manner, however, this is not enabled when packets are moved in different ways.

The objectives and contributions of this research are as follows;

(i) Duty cycling free method that compensates between node energy saving and packet transport delays.

(ii) The method that can achieve a high proportion of packet transport under different circumstances as opposed to reference conditions.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

(iii) Unlike methods based on previous researches (Niyato et al., 2007; Tang et al., 2011), where nodes were required to exchange data with each other. Is, this method allows nodes to be referenced. From these neighbourhood figures without estimating the position of its neighbours. In this way, the heavy ration of energy used for data exchange (Wei et al., 2011) can be saved.

Besides this paper is prepared as follows; nest segment covers the proposed reinforcement learning-based node sleep or wake-up time scheduling methodology in detail. Segment III covers the simulation and examines the proposed methodology. Finally, the research work is concluding in the next Segment IV.

2. Designed Approach

In this segment of paper, we begin with a description of the research work of the node sleep or wake time scheduling problem. At that point, the subtleties of the calculation are given, which are associated with the proposed methodology.

2.1 Problem Description

As describe in Segment I, the research for node sleep or wake-up time scheduling considers how to change the ratio between the sleeping and waking time of each sensor at each period of time, as represented in Figure 2.

Figure 2. Description of the problem

With considering the Figure 2, we have some descriptions are as followed;

Description 1: Sleeping state: A sensor cannot transmit or receive any packet when it is in a state of sleep. A sensor in the sleep state does not consume energy or a negligible amount of energy.

Description 2: Awaking state: A sensor can transmit and receive packets when it is in a state of awaking or activation state. When a sensor is in the activation state it spends significantly more energy as opposed to the sleep state.

Description 3: Nodes sleep or wake time scheduling: Sensors can adjust the time consumed by the sleep or wake-up state respectively to preserve energy and meanwhile ensure effective transmission of packets.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

In general, there are three modes of activity in radio transceivers in SN:

(i) Transmission mode of data, radio transceivers can transmit and receive data packets. Very high power required in this mode.

(ii) Tune-in mode, transmitter hardware turns off and the transceiver only allows packets to be received and the medium power required in this mode.

(iii) Sleep mode, both receiver and transmitter are switched off and the power required in this mode is much less than in both transmit and tune-in modes.

The model presented in (Cao et al., 2005) shows the power consumption as levels: 81 mW for transmission, 30 mW for tuning and 0.003 mW for sleep.

2.2 Model Description

Cooperation between two neighbouring nodes is demonstrated as a game of two players and three activities, where two players represent the row and column players respectively and represent with two neighbouring nodes and three activities mean Transmission, Tune-in and Sleep. Three terms, player, node and sensor are used in this research work. The game theory is a statistical method that can be used to manage decision-making problems in multiplayer mode. Through the process of decision making, there may be confrontation or participation between several players. Such confrontation or participation can be demonstrated effectively through game theory through the proper establishment of outcome outlines and utility capabilities. In WSNs, confrontation and participation occur between sensors during many processes, for example, data packet routing and node sleep or wake-up time scheduling. Consequently, in this research paper, game theory is used to manage the node sleep or wake-up time scheduling problem amongst sensors in WSN.

The game is characterized by a couple of payoff matrices:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{23} & r_{33} \end{pmatrix} \quad (1)$$

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{23} & c_{33} \end{pmatrix} \quad (2)$$

Here, R and C matrices represent the payoff for row and column players respectively. Each of the two players chooses one of the three accessible activities. The joint activity of the players decides their payoff as indicated by their payoff matrices. If the row and column player individually select the movements of i and j respectively, the row player receives the payoff r_{ij} and the player of the column receives the result c_{ij}. Players can choose their activities that are stochastically dependent on their distribution of probability over available activities. Let us consider $\alpha_1 to\ \alpha_3$ indicate the probability that the row player chooses activities individually from 1to3, where $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Let us consider $\beta_1 to\ \beta_3$ indicate the probability that the column player chooses activities individually from 1 to 3, where $\beta_1 + \beta_2 + \beta_3 = 1$. The expected payoff value of the row player and column player are P_r and P_c respectively;

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

$$P_r = \sum_{1\le i\le 3}\left(\sum_{1\le j\le 3} r_{ij}\alpha_i\beta_j\right) \tag{3}$$

$$P_c = \sum_{1\le i\le 3}\left(\sum_{1\le j\le 3} c_{ij}\alpha_i\beta_j\right) \tag{4}$$

Considered activities transmitted, tune in and sleep to be represented by 1 to 3 respectively. The result of these payoffs matrices in the network may be characterized by the node used energy (that will be a non-positive payoff). Also, if during a successful transmission of a packet, the result of the transceivers is the energy, which is used to transmit and receive the packet, except at a steady positive state X $(where\ X = 98)$. X is included in the accompaniment of energy, if and only if a packet is effectively transmitted. The result for the rest of the activity is -0.003 (the energy consumed during the sleep period) is independent of the opponent's activity, where a non-positive sign means that the energy is expended. The estimate of constant X is greater than the energy used by the packet to transmit or receive. For example, if a packet has been transmitted by a raw player and it chooses the transmission as an activity state and the column player chooses to tune in, the package can be effectively transmitted. Payoffs are secured for both players, which can be determined using the energy used to transmit/receive the package in addition to the constant value X. At that point, the row player receives the result $-81 + 98 = 17$ and the column player receives the result $-30 + 98 = 68$, where 81 and 30 are the used energy by the individual to transmit and receive a packet, a non-positive sign means energy is expended. As it may be, if the player wishes to sleep in the segment, the packet cannot be transmitted effectively. At that point, the row player receives the result -81 (energy consumed during transmit a packet) and the column player receives the result -0.003 (used energy for sleep). It must be known that if a packet does not have to be transmitted by a node, it will not choose to transmit.

During a defined time slot, every node must have one of the few states that display its buffer status. For example, if three packets are stored by node buffer store, there are four number of possible states for the node that is from s_0 to s_3, which indicates that the node buffer has individual 0 – 3 packets. The objective of each node is to find a strategy π, which maps out activities, which may enhance the node payoff for some time. Specifically, $\pi(s,a)$ would be a probability for a node based on selected activity $'a'$ at the state $'s'$, and $\pi(s)$ is a vector that is a probability distribution of accessible activities in the current state. Hence the structure of the policy $'\pi'$ will be a matrix. For example, a node buffer can accumulate three packets, so the node has four states from s_0 to s_3, as described already.

Similarly, the node has three activities: transmission, tune-in and sleep are represented by 1 to 3 respectively. So the policy of the node is

$$\pi = \begin{pmatrix} \pi(s_0,1) & \pi(s_0,2) & \pi(s_0,3) \\ \pi(s_1,1) & \pi(s_1,2) & \pi(s_1,3) \\ \pi(s_2,1) & \pi(s_2,2) & \pi(s_2,3) \\ \pi(s_3,1) & \pi(s_3,2) & \pi(s_3,3) \end{pmatrix} \tag{5}$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

In the beginning, since the node has no information, each activity is considered equally important in each state. For each state, $s, \sum_{1\leq i\leq 3}\pi(s,i)$ must be 1, each component in π is set from 1 to 3. At that point, through learning, the node will modify the estimated value of each component in π. The next segment of this paper covers it in detail.

Here, the terms $\pi(s,a)$ and α,β indicate the probability of choosing an activity. The value of α and β does not act as states while states are chosen thought $\pi(s,a)$, where α and β representation are used for model representation and calculation.

2.3 Proposed Algorithm

In view of the proposed model, we present an algorithm of reinforcement learning, which is used by a player to learn himself with his ideal activities through associations of experiments within a powerful domain. The algorithm is termed the Q-learning algorithm (presented in algorithm-1). Q-learning is one of the least complex algorithms of reinforcement learning. Reinforcement learning and evolutionary computation are both subfields of machine learning (ML). Reinforcement learning hopes to explain sequential decision assignment through experimentation with nature (Gong et al., 2015) and error connections with the environment. In a sequential decision assignment, a member interacts with a unique framework by choosing activities that affect change in status to update certain reward functions. Evolutionary computations are global search systems technique derived from Darwin's hypothesis of evolution by natural selection. Evolutionary computation iterations update a population of possible solutions, often encoded in structures called chromosomes. In this way, the key difference between reinforcement learning and evolutionary computation is that members use reinforcement learning to increase their individual rewards, while evolutionary computation is used to achieve global optimization. Furthermore, the calculation of reinforcement learning is mostly decentralized and requires only close data to members, although evolutionary computation is mainly integrated or require global data (Glavic et al., 2017). In this document, WSNs are in a distributed environment and each sensor only has data about itself, so reinforcement learning is more appropriate than evolutionary computation for the node sleep or wake-up time scheduling problem.

The advantage of reinforcement learning is that a player does not have to bother with a teacher to discover how to solve a problem. The main signal used by the player to make a profit from his activities in powerful situations is the result (sometimes called a reward), a number that tells the player whether his last activity was excellent or not. Q-learning makes simplest reinforcement learning without a model, meaning that players using Q-learning can function ideally in Markovian domain without building large maps of regions. During the learning process, a player performs an activity in a specific state that based on the probability distribution over accessible activities. The higher probability of an activity is the more possibility of taking more activities. At that point, the player estimates the outcome of the activity that the player has just taken, based on the quick reward or penalty that receives when accepting an activity and also depends on the indicator of the estimate of that state. It is the movement in which all activities are tried more than once in all states, the player finds which activity in a particular state is the best decision.

Algorithm 1 shows how the proposed methodology works in a one-time slot. The methodology is described in the perception of an individual node. As indicated in Figure 3, it can be very well seen that the proposed methodology consists of three phases. Initially, a node selects an activity that depends on the probability distribution over three activities: to transmit, to tune-in, or to sleep in the current state s. Secondly, the node performs the chosen activity and increasingly grasps

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

results and new states. Finally, the node modulates the probability distribution and probability of mutual neighbour policy over three activities depends on the results of states. The detailed description of Algorithm 1 is as follow;

Algorithm 1: RL based node sleep or wake-up time scheduling

1. Let us assume γ denotes the discount factor; ξ and ∂ are denotes the learning rates of nodes
2. ***for*** all activities,
   initialise a value function $Q \rightarrow 0$ and policy function $\pi \rightarrow \frac{1}{a}$ [1]
3. ***repeat***
4. choose an activity a from the current state s based on the policy function $\pi(s,\ a)$;
5. ***if*** $a = tranmit$
6. the node is responsible to determines throughput time of transmission of a packet in the predefined time slot; **/* Algorithm 2 */**
7. perceive the payoff p value and
   the next state s', then
   update Q-value
8. $Q(s,a) \leftarrow (1-\xi)Q(s,a) + \xi(p + \gamma max_{a'} Q(s',a'))$
9. ***if*** $a = tune - in$
10. on behalf of the modified Q-value, estimated the policy function π of the neighbour which interrelated with the node occurred in the current time slot;
    on behalf of the estimation, for each activity $a \epsilon A$, update the node's policy function $\pi(s,a)$
11. ***Else***
12. estimate the value of average payoff
13. $\bar{P}(s) \leftarrow \sum_{a \in A} \pi(s,a)\ Q(s,a)$
    for all activities $a \in A$ do
14. $\pi(s,a) \leftarrow \pi(s,a) + \delta(Q(s,a) - \bar{P}(s))$
15. $\pi(s) \leftarrow Normalise(\pi(s))$; **/* Algorithm 3 */**
16. $\xi \leftarrow \frac{k}{k+1}.\xi$
17. $s \leftarrow s'$
18. until the process is terminate

In the initial of each time slot, the algorithm 1 is repeated from row 3, with the exception of the first time slot where the algorithm starts on row 1. On row 1, the extent to which a learning rate is determined will replace recently obtained data with older data. The learning rate is estimated in the range [0, 1]. A factor of 0 means that the node is not habituated to anything; whereas a factor of 1 means that the node only considers the latest data (Zhang et al., 2018). A discount factor determines the importance of future compensation. The discount factor is estimated in the range [0, 1]. A factor of 0 means that the node is short-sighted only by considering about current rewards, whereas a factor that moves toward 1 means that the node is moving toward long-sighted high rewards (Zhang et al., 2018). For the start of each time slot, a node has to choose in

[1] a is the number of available activities

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

which mode in this current time slot. Thus the node chooses an activity that relies on the probability distribution over its accessible activities in its current state from row 4. The underlying probability distribution can be similarly established on accessible activities. For example, considering this paper has three activities. First, the probability of choosing each activity can be set from 1 to 3. Subsequently, during the learning process, the probability distribution on the activities will be updated based on the outcome of each activity. In the event that the chosen activity is transmitted, the node must choose to transmit the packet in the time slot (rows 5 and 6, where the description will be indicated in Algorithm 2). At that point, the node achieves one result and reaches another state. Update the Q-value of the chosen activity in its current state based on the obtained result and the most extreme Q-value in the new state (row 7). Here, Q-value, $Q(s,a)$, would be a reinforcement for creating an activity a in state s. These data are used to strengthen the learning process. The procedure on row 7 is a significant iteration update. First, the Q-value is wisely given by the designer. At that point, the Q-value is updated using the current estimate Q-value, $(1-\xi)\,Q(s,a)$, plus the learning information, $\xi\left(p+\gamma\ \max_{a'} Q(s',a')\right)$.

Algorithm 2 Calculation of selecting node during transmit packets in a defined time slot

1. Assume $\epsilon\ and\ \zeta$ are the learning rates
2. ***for*** all sub-slots available in the current time slot
   initialise the Q-value i.e. $Q \to 0$ and the probability of selecting all sub-slots is $x_i = \frac{1}{t_s}$[2];
   $1 \le i \le t_s$
3. based on the probability distribution chosen a sub-slot from the current time slot over the sub-slots $x = \langle x_1,\ x_2, \ldots\ldots, x_{t_s} \rangle$
4. observe the value of payoff p and
   update Q-value for every sub-slot
   $$Q_i \leftarrow Q_i + x_i\ .\ \zeta\ .\ (p - \sum_{1 \le i \le t_s} x_i Q_i)$$
5. update x_i value for every sub-slot
   $$x_i = \begin{Bmatrix} (1-\epsilon) + (\epsilon/t_s) & if\ Q_i\ is\ the\ maximum \\ \epsilon/m & otherwise \end{Bmatrix}$$
6. $x \leftarrow normalise(x)$

Learned information is composed of the result obtained by the node after creating an activity plus the estimation of ideal future value: $p + \gamma\ \max_{a'} Q(s',a')$. On the rows of 8 to 10, if the chosen activity is not in the sleeping state, the node will estimate the probability distribution over the accessible activities of the neighbour. This neighbour is the one who has interrelated with the node, which is either transmit or receive a packet. At that point, based on the approximation, the node updates its policy $\pi(s,a)$ to each accessible activity. On the rows of 11 to 14, if the chosen activity is to be sleeping, means that in the current time slot the node does not interrelate with any other node, then at that point the node updates its policy $\pi(s,\ a)$ for each available activity depends on the outcome. In row 12, the determination of the average payoff depends on the

[2] t_s is the total number of sub-slots in the network

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

estimation Q-value of an activity multiplied by the probability of selecting the activity.

Clearly, the average payoff can also be determined using the summation of the payoff of each activity dividing by the number of total activities (Renold and Chandrakala, 2017). However, the prior calculation technique is more productive and is more widely used than the previous one (Foerster et al., 2017). On row 13, the probability of choosing each activity is updated. The probability of choosing an activity is estimated using the current probability of choosing an activity in addition to the difference between the Q-value of the activity and the average payoff. In the event that the predicted Q-value of an activity exceeds the average payoff, the probability of choosing the activity will increase; otherwise, the probability will decrease. At row 15, the probability distribution $\pi(s)$ is standardized as an adequate distribution, where $a \in A, \pi(s,a) = 1$ and each $\pi(s,a)$ is within the range (0, 1). The subtleties of standardization will be illustrated in Algorithm 3. Finally, on row 16, the learning rate ξ is rotten, where k implies the k^{th} time slot. The rot technique is not a unique one. In fact, any dynamic rot technique can be used here (Leibo et al., 2017). At that point, towards the beginning of each time slot, the node re-implements algorithm 1 from row 3.

Algorithm 3 Normalize()

1. let in state s, there are b available activities, $a_1, a_1, \ldots\ldots, a_b$
2. let $d = \min_{1 \le k \le b} \pi(s, a_k)$, mapping center $c_0 = 0.5$ and mapping lower bound $\Delta = 0.001$
3. ***if*** $d < \Delta$
4. $\rho \leftarrow \dfrac{c_0 - \Delta}{c_0 - d}$
5. ***for*** $k = 1$ to m do
6. $\pi(s, a_k) \leftarrow c_0 - \rho \,.\, (c_0 - \pi(s, a_k))$
7. ***for*** $k = 1\ to\ b$
8. $r \leftarrow \sum_{1 \le k \le b} \pi(s, a_k)$
9. $\pi(s, a_k) \leftarrow \dfrac{\pi(s, a_k)}{r}$
10. return $\pi(s)$

Algorithm 2 and 3 are the intermediate parts of algorithm 1, where algorithm 2 and 3 are the evolution of row 6 and row and row 15 respectively. As it may be, the representation of algorithm 2 and 3 is very important and complex to join with algorithm 1. In this way, for reasons of clarity, algorithms 2 and 3 are separated from algorithm 1 to become autonomous algorithms.

2.4 Detail Description of Algorithm

In Algorithm 1, there are four problems that must be addressed.

(i) If a node chooses to transmit the packet at the current time slot, then how does the node decide to transmit the packet on availability of time slot (from algorithm 1 row 6).

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

(ii) How does the node approximate consider its opponent's probability distribution without examining the opponent for any information (from algorithm 1 row 9).

(iii) How a node updates its policy based on the approximation (from algorithm 1 row 10).

(iv) How is the invalid probability distribution standardized for a valid one (from algorithm 1 row 15).

The keys of the four problems are presented as:

2.4.1 Problem 1

As discussed in Segment I, if there is more time slot than the scheduled time for transmitting a packet, a node must decide when to transmit the packet in the current time slot. For each packet transmission, the length of a slotting time and the time period are known quickly. Assume that the start time of the schedule is long enough to transmitting the m packets. A time slot is divided into m sub-slots. The length of each sub-slot is equal to the time required to transmit a packet. Now a node needs to select a sub-slot from m sub-slots to transmit the packet. The two natural solutions are that:

- the node receives a random sub-slot for the packet and,
- from the first starting point of the slotting of time, the node sends the packet.

If the packet is not sent effectively, the node sends it once again. This process continues until the packet is sent effectively or the current time slot is terminated. In any case, the two solutions can cause a great deal of wastage of energy, especially the latter solution. To solve this problem, another learning algorithm is invented.

In algorithm 2, if a node chooses to transmit a packet in a slotted time, it chooses a sub-slot dependent on the probability distribution x on the sub-slots (row 3). At that point, the node looks at the result p gained to select the sub-slot and update the Q-value for each sub-slots dependent on the result p and the current probability distribution x (row 4). The probability distribution x is balanced based on the updated Q-value of the sub-slot (row 5) and is standardized to have sufficient distribution (row 6). The technique for updating the probability distribution (row 5) is $\epsilon - greedy$ exploration. $\epsilon - greedy$ exploration is characterized by a semi-uniform probability distribution. The best current sub-slot is chosen with probability $1 - \epsilon + (\epsilon/m)$ and one of the remaining sub-slots is chosen with probability (ϵ/m), where m is the number of sub-slots. The reason for the investigation is to organize compensation between misuse and examination with the ultimate goal being to reinforce the evaluation of node sub-slots that it knows are certainly excellent, but also investigate new sub-slots. The importance of obtaining a Q-learning algorithm with $\epsilon - greedy$ exploration is also supported through a large number of uses (Singh et al., 2000; Liu and Yoo, 2017).

It should be noted that the use of the intuitive solution described above would be a waste of energy, as there is no trace of these solutions. The proposed algorithm, named, algorithm 2, in any case, allows nodes to correctly select suitable sub-slots for effective packet transmission. In this way, in compared with the intuitive solution, the wasted energy in the transmission of failed packets can be saved by using algorithm 2. Algorithm 2 takes time for assembly. As it may, during the assembly process, the performance of the algorithm 2 is extended step by step, while intuitive solution performance does not improve as time progresses, as they do not have the any adaptive ability. This means that algorithm 2 exceeds the intuitive method from the beginning of

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

execution. At that time, the performance breach increases w.r.t. time.

2.4.2 Problem 2

For example, taking the player from the row (review segment II - B), the general payoff of the row player's activities is 1 – 3 as follows;

$$P_r^{(1)} = \lim_{\alpha_1 \to 1} \sum_{1 \le j \le 3} \left(\sum_{1 \le j \le 3} r_{ij} \alpha_i \beta_j \right) = \sum_{1 \le j \le 3} r_{1j} \beta_j \tag{6}$$

$$P_r^{(2)} = \lim_{\alpha_2 \to 1} \sum_{1 \le j \le 3} \left(\sum_{1 \le j \le 3} r_{ij} \alpha_i \beta_j \right) = \sum_{1 \le j \le 3} r_{2j} \beta_j \tag{7}$$

$$P_r^{(3)} = \lim_{\alpha_3 \to 1} \sum_{1 \le j \le 3} \left(\sum_{1 \le j \le 3} r_{ij} \alpha_i \beta_j \right) = \sum_{1 \le j \le 3} r_{3j} \beta_j \tag{8}$$

It shows that if each activity is expressed in an endless number of times in an endless run and the learning rate ξ is rotten correctly, and then the estimate of that activity is determined using Q-value (the calculation from row 5 of the algorithm 1) will be associated with the probability 1 of Q* where Q* is the normal result of the player playing that activity. As indicated in equation $(vi) - (viii)$, it can be ascertained that the general result of a row player executing an activity depends on the column player's chances to execute each activity. On behalf of this conclusion, the row player can use the normal Q-value of activity to estimate the general result of that activity.

$$Q(s,1) = P_r^{(1)} = \sum_{1 \le j \le 3} r_{1j} \beta_j \tag{9}$$

$$Q(s,2) = P_r^{(2)} = \sum_{1 \le j \le 3} r_{2j} \beta_j \tag{10}$$

$$Q(s,3) = P_r^{(3)} = \sum_{1 \le j \le 3} r_{3j} \beta_j \tag{11}$$

By using equation $(ix) - (xi)$, the row player can compute $\beta_1 - \beta_2$, that is the column player's probability to perform different activities from 1-3 respectively. After computing, the column player modifies his own probabilities to perform each activity, which will be described in the next segments of the paper. Positively, it is not possible to execute the same activity multiple times. In this way, the general result should be approximate; however, it cannot be estimated at all. As it may be the learning progresses, the algorithm of Q-learning will gradually add to the ideal Q-value, Q* and in this way, the accuracy of perception will be increased. At that point, the probability distribution of the column player can be calculated more definitively by the player. Consequently, it is seen that by using the assumption, the row player predict on activities without correspondence with the column player can provide for the possibility distribution of the column player.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

2.4.3 Problem 3

By using a gradient ascent algorithm (Abdul-Salaam et al., 2017), a player can extend his normal result by moving his progress towards the current inclination with a fixed progression size. The gradient is calculated as a partial derivative to the player's required payoff w.r.t. their probability of choosing each activity. We take the row player, for example, and set $\alpha_3 = 1 - \alpha_1 - \alpha_2$. At that point, we have

$$\frac{\partial P_r}{\partial \alpha_1} = \sum_{1 \le j \le 3} r_{1j}\beta_j - \sum_{1 \le j \le 3} r_{3j}\beta_j \tag{12}$$

$$\frac{\partial P_r}{\partial \alpha_2} = \sum_{1 \le j \le 3} r_{2j}\beta_j - \sum_{1 \le j \le 3} r_{3j}\beta_j \tag{13}$$

In case if the k^{th} time slot, ${\alpha_1}^{(k)}\ to\ {\alpha_2}^{(k)}$ are the row player probabilities for choosing from 1 to 3 activities respectively, then new possibilities for the next time slot are

$$\alpha_1^{(k+1)} = \alpha_1^{(k)} + \eta \frac{\partial P_r}{\partial \alpha_1^{(k)}} \tag{14}$$

$$\alpha_2^{(k+1)} = \alpha_2^{(k)} + \eta \frac{\partial P_r}{\partial \alpha_2^{(k)}} \tag{15}$$

and $$\alpha_3^{(k+1)} = 1 - \alpha_1^{(k+1)} - \alpha_2^{(k+1)} \tag{16}$$

where η is the gradient step size, it is quite possible to realize that if the η is sufficiently low, the technique will be involved, based on mathematical information.

2.4.4 Problem 4

Proportion-based mapping is used, to standardize invalid probability distribution to be the valid one. Invalid probability distribution contains learning information in order to preserve learned information; the transmission of standardized probability must be "closed" as unstandardized. Proportion-based mapping can modify each invalid probability into a range (0, 1) and, meanwhile, maintain the overall relative magnitude of the probabilities.

From the above details of the algorithm, it can be very well seen that, unlike the existing node sleep or wake-up time scheduling approaches, where the sleep or wake-up time arrangement of nodes is almost prefixed, this method allows nodes to be freely and powerfully selected if sleep stage using learning processes. In this way, hypothetically, nodes in this method are smarter than nodes in most standing methodologies.

3. Simulation and Analysis

In this segment, the results of simulation and comparative research are presented. The simulation was executed using the JAVA programming language and was executed within the framework of Windows 10 Professional SP1 system with Intel Core i5 3.4-GHz CPU and 8GB of RAM.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

The node is adapted as a class, and each node is an object of this class. To simulate a region, a 2-dimensional array is defined which speaks of the directions of a region. The length of the array of the row is the length of the region and the length of the array of the column is the width of the region. For the impact of two nodes, if the two nodes are in correspondence range, the two nodes are connected. For the execution of time, a Java function is used to check the system time. To measure the duration of the packet transport time, both the beginning and the end of the system time of the transmit packet are carefully examined. At that point, the duration of transport time of the packet can be obtained using the system time read towards the beginning of the packet delivery time, with the frame time read at the end of the delivery of the packet.

To evaluate the proposed approach, we created a platform to use JAVA with four different methods in the examination: 1) Two-Radio-MAC (TR-MAC) (Cao, et al., 2005) is a method of on-demand; 2) DW-MAC (Sun et al., 2008), is a synchronous method. 3) EM-MAC (Tang et al., 2011); and 4) AS-MAC (Kim et al., 2009), are asynchronous methods.

By comparing these methodologies, the performance of the proposed methodology can be similarly characterized. The purpose behind choosing these methods is that TR-MAC and DW-MAC are the most in-demand and synchronous methods respectively, while EM-MAC and AS-MAC both are the modern asynchronous methods, perhaps, as far as we know, their performance is not yet compared. Hence, we selected these for our simulation environment. These methods are detailed description as follows;

***TR-MAC*:** In these two signals are used, one for wake up the neighbour and the other to send the package. Nothing looks like traditional methods for on-demand in TR-MAC, when a node has a packet for transmission; it does not wake up its all neighbourhood, but specifically awakens some neighbours who have recently dealt with correspondence rate estimates.

***DW-MAC*:** It uses a synchronization duty cycle in MAC protocol, where each cycle is distributed into three periods: a) synchronous; b) information; and c) sleep. It needs to synchronize clocks to SNs during the synchronous period. At that point, DW-MAC establishes a consistent mapping between an information period and the accompanying sleep time period. In an information period, the sender will send a schedule frame to the receiver. In the intermediate time slot after the start of the notification deadline and the duration of the transmission of the scheduling frame, both the sender and the recipient will establish their wake-up time slot to transmit/receive packets during the sleep time frame.

***EM-MAC*:** In this, each node uses a pseudorandom number generator: $X_{n+1} = (aX_n + c)\, mod\, m$ to record the time of its activation, where $m > 0$ is the modulus, a represent the multiplier, c is constant growth function, X_n is the current seed and X_{n+1} becomes the next seed. In the simulation, $m = 65536$, each node a, c and X_n are autonomously selected following the standards suggested by Knuth (Abdul-Salaam et al., 2017). By mentioning the parameters, m, a, c and X_n, a sender can estimate the receivers' future wake-up activity time and prepare to send information on those time frames. EM-MAC does not need to bother with synchronization but expects nodes to exchange data before making predictions.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

***AS-MAC*:** In this nodes are sometimes activated to obtain packets (though asynchronously). The nodes in which the packet is targeted to be delivered are waking up at the scheduled time of the proposed target nodes. Neighbouring nodes should distribute commercial data intermittently to avoid prolonged introduction towards the start of transmission.

Similarly, we also compare the proposed SA-Mech with its synchronized interpretation, _PRO. In _PRO, it is expected that a sink node will periodically transmit an exceptional packet throughout the system to synchronize the node clocks. Point of demonstration to _PRO for comparison is to examination how synchronization will affect SA-Mech performance.

3.1 Simulation Setup

Simulation works experiment with one type of grid systems. The configuration is intended to evaluate the presentation of the systems in different scales and different types of systems. The size of the grid network is 49 nodes are organized as a 7 × 7 matrix arrangement. In this grid structures, each node is 200 meters from its neighbours and there are 5 sinks that are located in the four corners and centre points of the system.

In this organized grid structure, each node able to generate a packet toward the beginning of each time slot that based on a predetermined probability i.e. the packet generation probability. As the position of a node is controlled by the number of packets in its framed buffer, there is a possibility that the packet directly affects the position of each node. At that point, the determination of the activity of each node will be indirectly affected. The completion time of a packet depends on the exponential distribution. The typical size of a package is 100 bytes, and the actual size of a package depends on normal transmissions with a variance valued 10. In this simulation, four possibilities of packet generation can use: 0.2, 0.4, 0.6 and 0.8. This configuration is for evaluating the performance of these methods in a network transmission of a number of packets. For routing a packet, we use a fundamental routing approach, gossiping (Abdul-Salaam et al., 2017). Gossiping is an improved version of flooding approach where the receiving nodes send the packet to a randomly chosen neighbour, who chooses another random neighbour to push the packet forward, etc., until it reaches the final destination. It should be kept in mind that when the destination and some individual nodes are all within the range of the source signals, through the routing protocol the source still transfers the packet to one of the neighbours and the process continues until the destination or the extreme hop is achieved.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

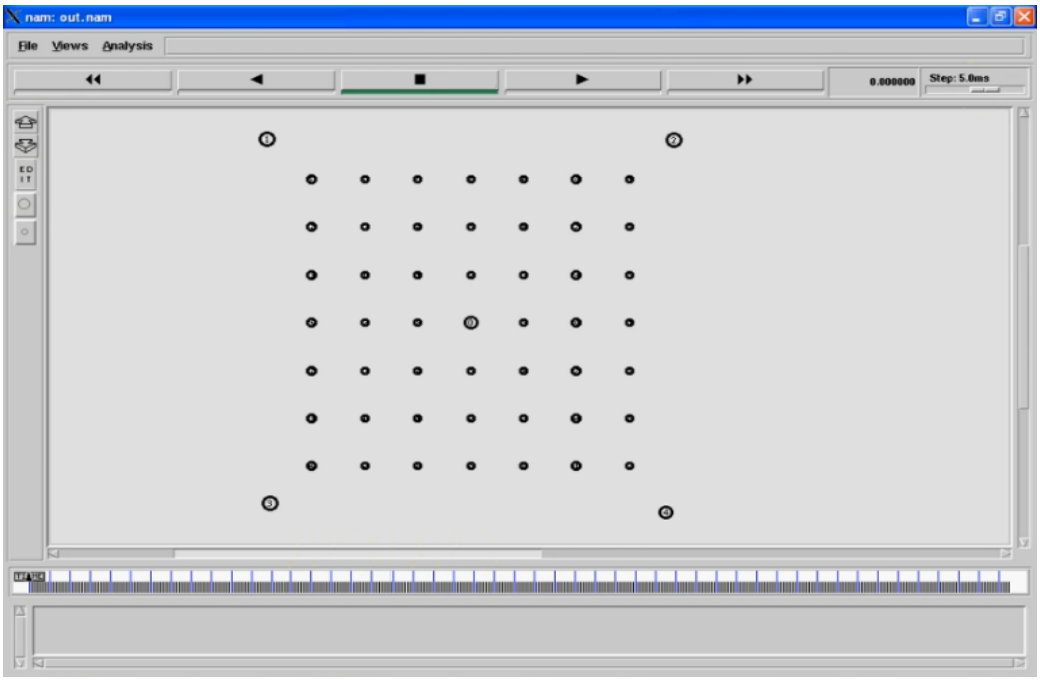


Figure 3. Network Grid with 49 nodes (7 × 7) matrix arrangement

3.2 Performance is measured by Three Quantitative Metrics

Routing process does not optimize in the simulation, as this research work only focuses on node sleep or wake-up time scheduling. This routing protocol is not capable to achieve energy efficiency, but it is not difficult to implement. Since all node sleep or wake time scheduling approaches use the same routeing protocol in simulation, examination between them remains required. Performance is estimated by three quantitative measurements:

- Packet delivery ratio
- Average energy consumption
- Average packet delivery latency

The least time required by nodes to transmit or receive a packet is about 2 ms (Wei et al., 2011), for example, using the Chipkon CC2420 radio chip. The above three estimations are detailed as follows:

3.2.1 Ratio of Package Delivery

The ratio of package delivery is estimated using the packets percentages that are effectively transported from source to sink. Each packet comes with a parameter, life-time, which is a non-negative number. When a packet is transmitted from the sender to a receiver node (whether it is transferred effectively or not), the lifetime of this packet decreases by 1. In the event that the lifetime of this packet becomes 0 and the destination is not reached, the transport of this packet is a disappointment.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

3.2.2 Average Energy Consumption

The average energy consumption is determined by the use of absolute energy consumption to divide the number of nodes in the network throughout the simulation run.

3.2.3 Average Packet Delivery Latency

The latency of packet transport is estimated from the average time it takes for each packet transported from source to sink or destination. Keep in mind that those packets, which do not effectively reach the sink node, have also been considered. Its delivery latency is the intermediate time during which they are present in the network.

In this simulation setup, we configure the evaluation cases ourselves. The method used for correlation in the simulation comes from four unique referenced that use specific evaluation cases. In this way, there are no general evaluation cases between these contexts. As with all methods evaluated, testing is done in similar cases, the examination remains reasonable and the results are motivational.

The lifetime approximation is set to 8 in the network and the value of the parameter was chosen to temporarily give the best execution of the proposed approach. For the analysed approaches, the duty cycle is set to 5%. A complete node sleep or wake-up time interval is set to 1s. As our proposed methodology does not use the delay cycle, the time length of a time slot in our methodology is set to 8 ms. As described in Segment II C, learning rates are used to update learning information. Since the observational approaches do not use reinforcement learning, they do not need to be bothered by updates and in this sense; they do not have to worry about learning rates. In each case of simulation, each of these methods has 200 iterations and each iterator contains 5000s.

3.3 Simulation Results in Grid Networks

The grid network topology is similar to a chessboard as shown in Figure 3 and 4, have three standards as follow:

(i) Four corners have two neighbours in each of the four nodes.

(ii) Each node on the edge has three neighbours.

(iii) All other nodes have four neighbours.

Figure 5 represents active mode of network grid in which activity modes of neighbour nodes of node 5 transmitting packets in a prescribed time slot.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

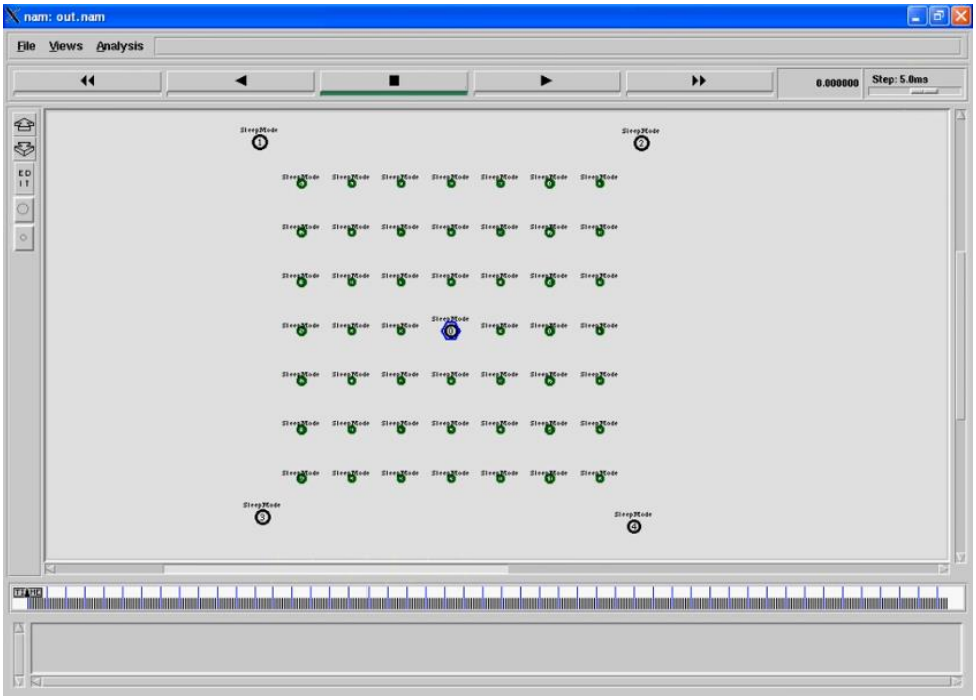


Figure 4. Network Grid active mode with nodes initial states

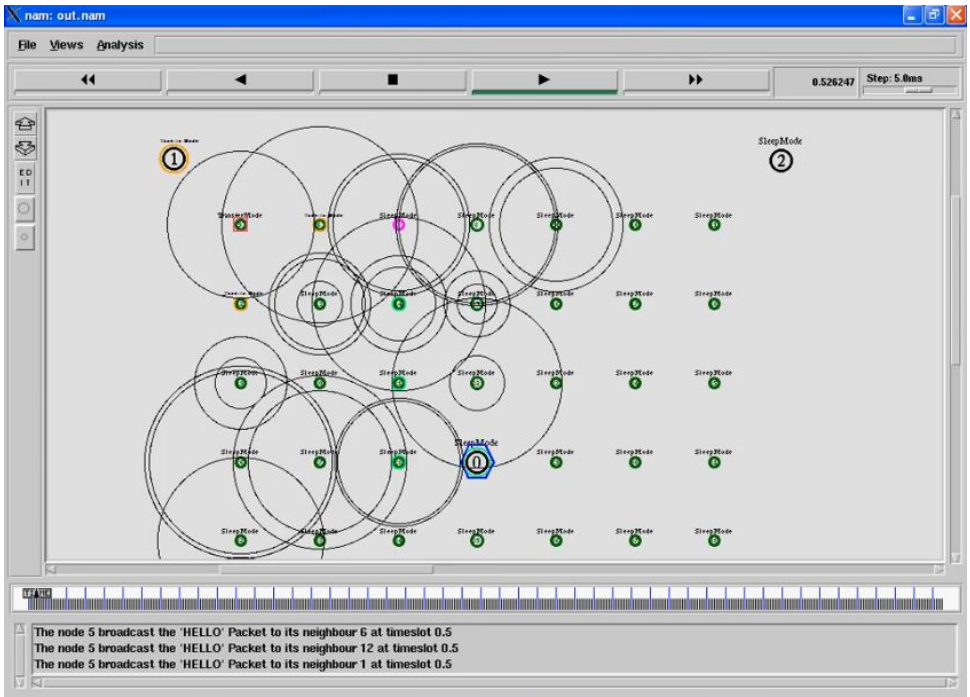
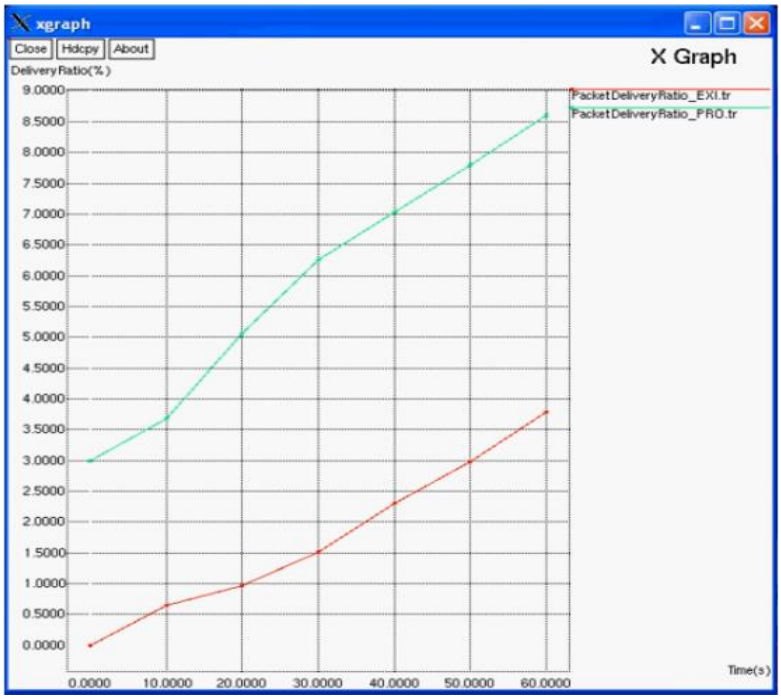
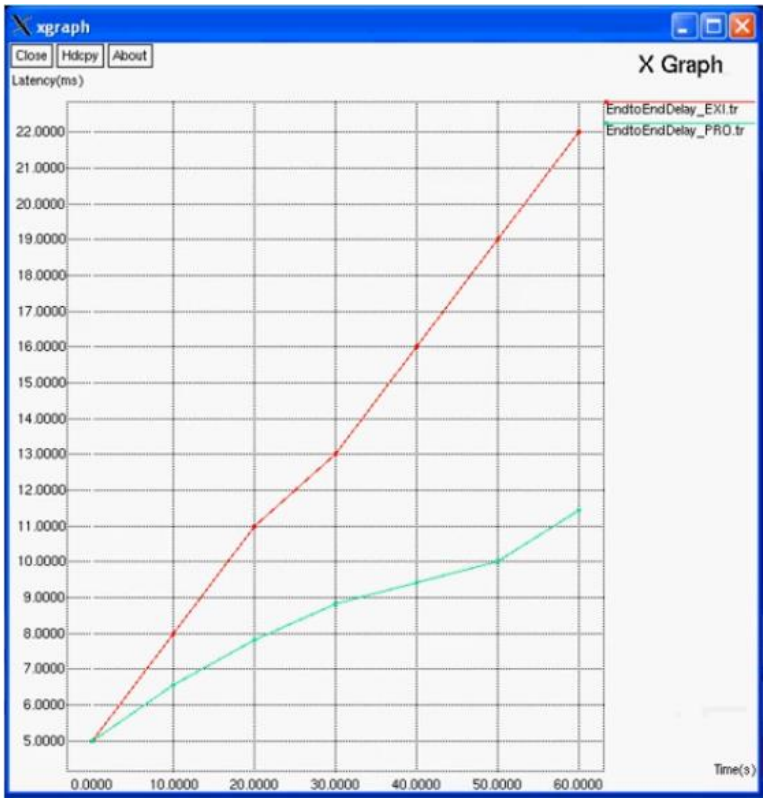


Figure 5. The activity modes of neighbour nodes of Node 5 by transmitting packets in a time slot

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057



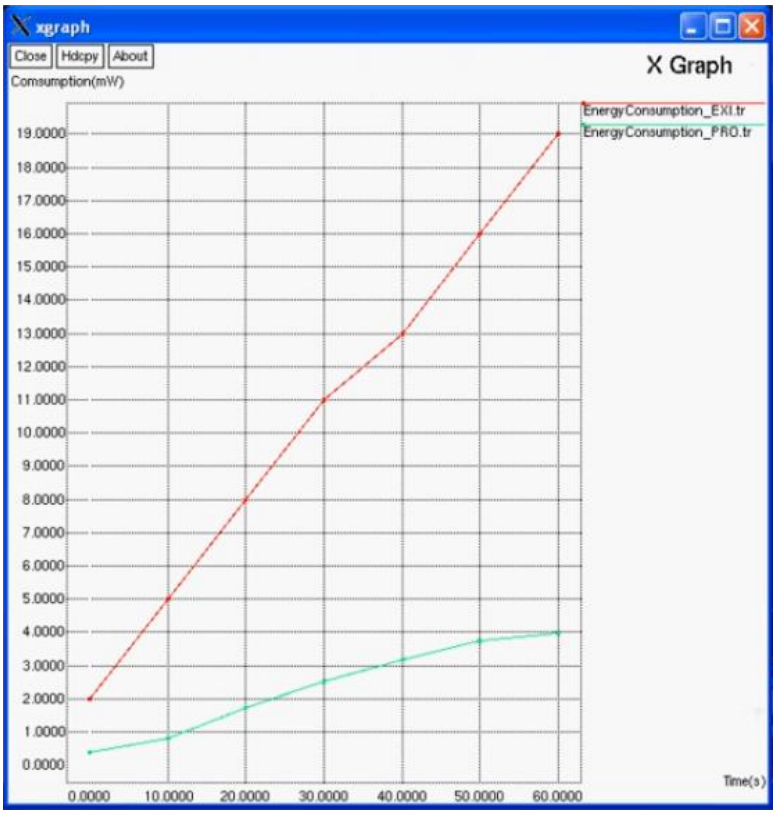
(a)



(b)

International Journal of Mathematical, Engineering and Management Sciences
Vol. 5, No. 4, 707-731, 2020
https://doi.org/10.33889/IJMEMS.2020.5.4.057

Figure 6 continued…



(c)

Figure 6. Performance evaluation based on: (a) Packet delivery ratio, (b) Average delivery latency or end to end delay, (c) Energy consumption

Furthermore, in this simulation setup, the five sinks are located separately in the four corners and the one at the centre point of the network. In this way, matrix systems are exceptionally general. Under this circumstance, the proposed methodology, SA-Mech (_PRO) may work admirably, depending on SA-Mech. It depends on reinforcement learning, and sometimes it can be learned effectively with high stability. In this grid, each packet is moved to the four corners and centre, where the sinks are placed. This generality can be effectively learned by nodes under SA-Mech.

Based on the progressive time and the learned information nodes can correctly consider the circumstances of their neighbours. Detailed analysis of the simulation result is enclosed in this segment.

3.4 Performance Evaluation

Figure 6 (a) shows the performance of the proposed methods (_PRO) with other existing methods (_EXI). The packet generation probability is set to 0.2. In Figure 6 (a), the average delivery latency in these methodologies has been presented. It is revealed that existing methodology conducts at the highest latitude approximate 36 ms at the 49 nodes.

This is because in existing methodology expects to transmit a packet to a sender at any receiver node; it needs to wake up the receiver and wait for the recipient's response. This process will continue until the packet reaches the target or the lifetime of the packet reaches 0, causing each packet to suffer major inactivity during the transmission process. The proposed methodology, SA-Mach., achieves minimum latency around 25 ms in a grid network of 49 hubs. The fact that:

- In SA-Mech periodic exchange of data is not required unlike existing methods (like in EM-MAC). It does not expect nodes to demand data from their neighbours for forecasts, so the corresponding time is saved, and
- As existing methods (like EM-MAC) is an asynchronous approach, resulting in the two neighbouring nodes are not synchronized. When a node predicts its neighbours, the wake interval uses its alarm clock, so the expectation may not be accurate enough. While in the proposed method, nodes can reliably approximate their neighbours' behaviour based on current conditions and their neighbours, hence SA-Mech achieves 10% less delivery latency inactivity than existing methods.

In Figure 6 (b), the packet delivery ratio decreases in these methodologies due to the increases in network scale. This is mainly a direct result of the routing approach. In this existing simulation setup, each packet receives the same lifetime. As the network expands, the lifetime estimate is never large enough to guarantee a successful transmission of the packet to its target. Consequently, an increasing number of packages cannot be effectively reached their destination. In this situation, existing methods achieve the best results for 95% in the network of 49 nodes. In existing methods (especially TR-MAC), the transmission is on-demand and transmission occurs after communication with the sender to receiver, so packets will likely be transmitted effectively. While in the proposed method the best results are achieved 95% in the network of 49 nodes. To save energy, nodes cannot transmit data. The nodes are based on choices that depend on an estimate that may contain errors; hence SA-Mech. is to some extent, around 2%, in this case, less productive than existing methods.

In Figure 6 (c), with the expansion of the network scale, the consumption of energy decreases continuously. In fact, with the expansion of the scale of the network, the energy consumption increases, as the packet must be transmitted with more intermediates toward the destination. As it may be, the packet generation probability and lifetime are likely to be fixed; the use of the total energy consumption can't be increase expanded with the expansion of the network scale. In this sense, the acceleration of the use of energy is slower than the increased number of nodes. In Figure 6 (c) we can very well see that proposed methodology uses less energy than existing methodologies, similarly, the existing methodology can achieve a less packet delivery ratio and higher average packet delivery latency. This synchronization is a direct result of the performance. To recognize synchronization, a sink needs to be broadcast to synchronize node clocks and this synchronization process certainly costs some energy. It may be that, synchronization can predict the behaviour of the node compared with the synchronized nodes. In existing approach, each node adapts its behaviour that based on the prediction approach and able to approximate the behaviour of its neighbour nodes while introducing the synchronization as in proposed method can make the behaviour of the nodes more predictable. We leave this as an investigation into our future.

4. Conclusion

In this research paper, a node sleep or wake time scheduling method based on reinforcement learning for WSN was presented. This method does not use any duty cycling technology to schedule nodes in the network. Conversely, it separates the node's axis time into multiple time slots and allows each node to transmit, tune-in, and sleep activity in predefined time slots. Each node is satisfied with a decision-based process that depends on its current state and the state of its neighbouring nodes, where such an approximation does not require communicating with neighbours. Through this technique, the proposed methodology is outperforming other existing methods. Most of the current existing methodologies are depends on duty cycles technology. Therefore, the duty cycle node is an effective strategy for sleep or wake-up scheduling. The proposed methodology may have less impact on performance with existing methods but it provides another way of sleeping or waking times of nodes in WSNs. Since the paper mainly focuses on theoretical studies, there are some assumptions. These assumptions are established to improve the interaction of our methodology.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgements

This research didn't receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors sincerely appreciate the editor and reviewers for their time and valuable comments.

References

Abdul-Salaam, G., Abdullah, A.H., & Anisi, M.H. (2017). Energy-efficient data reporting for navigation in position-free hybrid wireless sensor networks. *IEEE Sensors Journal*, *17*(7), 2289-2297.

Acampora, G., Cook, D.J., Rashidi, P., & Vasilakos, A.V. (2013). A survey on ambient intelligence in healthcare. *Proceedings of the IEEE, 101*(12), 2470-2494.

Cao, Q., Abdelzaher, T., He, T., & Stankovic, J. (2005, April). Towards optimal sleep scheduling in sensor networks for rare-event detection. In *International Processing in Sensor Network* (*IPSN*) *2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005. (pp. 20-27). IEEE*. Boise, ID, USA.

Chen, C.P., Mukhopadhyay, S.C., Chuang, C.L., Lin, T.S., Liao, M.S., Wang, Y.C., & Jiang, J.A. (2014). A hybrid memetic framework for coverage optimization in wireless sensor networks. *IEEE Transactions on Cybernetics*, *45*(10), 2309-2322.

Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P.H., Kohli, P., & Whiteson, S. (2017, August). Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1146-1155). Sydney, Australia.

Fu, B., Xiao, Y., Liang, X., & Chen, C.P. (2014). Bio-inspired group modeling and analysis for intruder detection in mobile sensor/robotic networks. *IEEE Transactions on Cybernetics*, *45*(1), 103-115.

Glavic, M., Fonteneau, R., & Ernst, D. (2017). Reinforcement learning for electric power system decision and control: Past considerations and perspectives. *IFAC-PapersOnLine, 50*(1), 6918-6927.

Gong, Y.J., Chen, W.N., Zhan, Z.H., Zhang, J., Li, Y., Zhang, Q., & Li, J.J. (2015). Distributed evolutionary algorithms and their models: a survey of the state-of-the-art. *Applied Soft Computing, 34*, 286-300.

Guo, P., Jiang, T., Zhang, Q., & Zhang, K. (2011). Sleep scheduling for critical event monitoring in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems, 23*(2), 345-352.

Jang, B., Lim, J.B., & Sichitiu, M.L. (2013). An asynchronous scheduled MAC protocol for wireless sensor networks. *Computer Networks, 57*(1), 85-98.

Keshavarzian, A., Lee, H., & Venkatraman, L. (2006, May). Wakeup scheduling in wireless sensor networks. *In Proceedings of the 7th ACM International Symposium on Mobile ad hoc networking and computing* (pp. 322-333). ACM. New York, USA.

Kim, J., Lin, X., Shroff, N.B., & Sinha, P. (2009). Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Transactions on Networking, 18*(2), 515-528.

Lai, S., Ravindran, B., & Cho, H. (2010). Heterogenous quorum-based wake-up scheduling in wireless sensor networks. *IEEE Transactions on Computers, 59*(11), 1562-1575.

Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J., & Graepel, T. (2017, May). Multi-agent reinforcement learning in sequential social dilemmas. In proceedings of the *16th Conference on Autonomous Agents and MultiAgent Systems* (pp. 464-473). International Foundation for Autonomous Agents and Multiagent Systems, Sai Paulo, Brazil.

Li, M., Li, Z., & Vasilakos, A.V. (2013). A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE, 101*(12), 2538-2557.

Liu, X. (2015). A deployment strategy for multiple types of requirements in wireless sensor networks. *IEEE Transactions on Cybernetics, 45*(10), 2364-2376.

Liu, Y.Y., & Yoo, S.J. (2017, July). Dynamic resource allocation using reinforcement learning for LTE-U and WiFi in the unlicensed spectrum. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 471-475). IEEE. Milan, Italy.

Niyato, D., Hossain, E., Rashid, M.M., & Bhargava, V.K. (2007). Wireless sensor networks with energy harvesting technologies: A game-theoretic approach to optimal energy management. *IEEE Wireless Communications, 14*(4), 90-96.

Polastre, J., Hill, J., & Culler, D. (2004, November). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 95-107). ACM. Baltimore MD, USA.

Renold, A.P., & Chandrakala, S. (2017). MRL-SCSO: multi-agent reinforcement learning-based self-configuration and self-optimization protocol for unattended wireless sensor networks. *Wireless Personal Communications, 96*(4), 5061-5079.

Semnani, S.H., & Basir, O.A. (2014). Semi-flocking algorithm for motion control of mobile sensors in large-scale surveillance systems. *IEEE Transactions on Cybernetics, 45*(1), 129-137.

Singh, S., Jaakkola, T., Littman, M.L., & Szepesvári, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning, 38*(3), 287-308.

Sun, Y., Du, S., Gurewitz, O., & Johnson, D.B. (2008, May). DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing* (pp. 53-62). ACM. New York, US.

Sun, Y., Gurewitz, O., & Johnson, D.B. (2008, November). RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems* (pp. 1-14). ACM. Raleigh NC, USA.

Tang, L., Sun, Y., Gurewitz, O., & Johnson, D.B. (2011, May). EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (pp. 1-11). ACM. Paris, France.

Tang, L., Sun, Y., Gurewitz, O., & Johnson, D.B. (2011, April). PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In *2011 Proceedings IEEE INFOCOM* (pp. 1305-1313). IEEE. Shanghai, China.

Wei, G., Ling, Y., Guo, B., Xiao, B., & Vasilakos, A.V. (2011). Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter. *Computer Communications*, *34*(6), 793-802.

Xiao, Y., Peng, M., Gibson, J., Xie, G.G., Du, D.Z., & Vasilakos, A.V. (2011). Tight performance bounds of multihop fair access for MAC protocols in wireless sensor networks and underwater sensor networks. *IEEE Transactions on Mobile Computing*, *11*(10), 1538-1554.

Yao, Y., Cao, Q., & Vasilakos, A.V. (2015). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Transactions on Networking*, *23*(3), 810-823.

Ye, W., Heidemann, J., & Estrin, D. (2002, June). An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 3, pp. 1567-1576). IEEE. New York, USA.

Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, *12*(3), 493-506.

Ye, W., Silva, F., & Heidemann, J. (2006, October). Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems* (pp. 321-334). ACM. Boulder Colorado, USA.

Zhang, D., Ma, Y., Zhang, Y., Lin, S., Hu, X.S., & Wang, D. (2018, April). A real-time and non-cooperative task allocation framework for social sensing applications in edge computing systems. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)* (pp. 316-326). IEEE.Porto, Portugal.

Zhao, Y., Liu, Y., Duan, Z., & Wen, G. (2016). Distributed average computation for multiple time-varying signals with output measurements. *International Journal of Robust and Nonlinear Control*, *26*(13), 2899-2915.

Zheng, R., Hou, J.C., & Sha, L. (2003, June). Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile ad hoc Networking & Computing* (pp. 35-45). ACM. Maryland, USA.

Zhu, S., Chen, C., Li, W., Yang, B., & Guan, X. (2013). Distributed optimal consensus filter for target tracking in heterogeneous sensor networks. *IEEE Transactions on Cybernetics*, *43*(6), 1963-1976.