Energy Efficient MAC Protocol With Traffic-Aware Wakeup Scheduling For Iot-WSN

Dr.G.Srinivasan

Alva's Institute of Engineering and Technology, Solapur - Mangalore Highway, Shobhavana Campus MIJAR, Moodbidri, Mangaluru, Karnataka 574225

srinivasanphd123@gmail.com

ABSTRACT

The Internet of Things (IoT) is a network concept that is critical to the development of smart environments. Wireless Sensor Networks (WSNs) are poised to revolutionise the Internet of Things (IoT). IoT-WSN data transmission schedules and channel access remain a major issue for effective system performance in IoT networks, though. The physical and medium access control (MAC) layer functions of an IoT device are the ones that consume the most energy. MAC protocols' energy efficiency has been extensively studied in the literature because of their fundamental importance. In this paper, Energy Efficient MAC protocol with Traffic-aware Wakeup Scheduling (TWS-EEMAC) is proposed for IoT-WSN. In this protocol, the lengths of duty-cycle of the IoT sensor nodes are determined based on traffic aware duty-cycle adjustment technique. In this technique, the awake period of the nodes are adjusted based on the number of contenders and the traffic load at the node. In order to determine the optimal number of contenders, Q-learning algorithm is applied. The proposed TWS-EEMAC protocol is implemented in NS2 and it shows superior performance in terms of end-to-end delay, packet delivery ratio, throughput and energy consumption.

Keywords: Energy; IoT; WSN; Traffic; Scheduling

DOI: 10.48047/ecb/2023.12.si12.010

1. INTRODUCTION

There are a number of ways that the Internet of Things (IoT) can be used in the creation of smart environments, such as by enhancing everyday objects with computing capabilities and Internet connectivity so that they can communicate with one another to share information about the environment, coordinate activities autonomously and fulfill other specific goals such as target tracking and medical supervision of patients. [1]

Any device in an IoT network can communicate with other devices, and the communication is not affected by the location, network, or internet service provider. The communication of applications in IoT can be attained through wireless technology, thereby achieving ubiquitous and seamless communication access. However, the schedule and channel access of data transmission remain a critical problem for efficient system performance in IoT networks. Therefore, the implementation of an efficient MAC protocol is important to achieve high channel utilization [2].

Wireless Sensor Networks and the Internet of Things are two sides of the same coin (WSNs). Small, low-powered, durable smart items including sensors, actuators, and processors with numerous wireless restrictions enable the IoT to deploy the tiniest objects in any location at the lowest cost. The advancement of WSN is facilitated by the incorporation of such devices into the Internet of Things (IoT). Several critical performance needs, including as energy efficiency, infinite scalability, low latency, and high concurrency are required concurrently as the modern world moves towards this revolutionary era of WSNs in the IoT scenario [3].

IoT devices may be made more power efficient and ecologically friendly by implementing energy-efficiency (EE) paradigms. An IoT device's physical and medium access control (MAC) layer processes use the most energy. Transmission accounts for the majority of energy usage in WSN networks. For example, latency, throughput, packet delivery ratio (PDR), and energy consumption are all dependent on the design of the MAC layer, which implies that the MAC is considered to be crucial to WSN application design.

The energy efficiency of MAC protocols has been extensively researched in the literature because of its critical importance [6].

In this paper, Energy Efficient MAC protocol With Traffic-aware Wakeup Scheduling is proposed for IoT-WSN.

2. RELATED WORKS

Large discrete-event simulations back up an analysis based on discrete-time Markov chains, according to Iclia Villordo-Jimenez et al [1]. SA-MAC outperforms previous ideas in terms of power consumption, throughput, and the probability of packet loss. Because of the expected high node density and high traffic loads of IoT applications, this is an impacting.

Chien-Min Wu et al [2] distributed .'s queuing medium access control (QL-based DQMAC) protocol may now be used for Internet of Things (IoT) networks It is through the suggested QL-based DQMAC that we identify the right number of IoT nodes to compete against. The Q-learning technique is used by each node to establish its own active rate. To select whether to be active or asleep in the upcoming contention period, each node takes into account its active rate. Choosing the best IoT nodes in each contention period reduces the likelihood of collision. Because there are fewer MAC contentions, less energy is used and delays for MAC contention are decreased. The proposed QL-based DQMAC protocol outperforms existing DQMAC protocols in IoT networks, according to a protocol comparison.

Atul Kumar Pandey et al [3] have developed a new protocol called CLA-MAC for WSNs in the IoT scenario, which focuses on traffic load balancing within each cluster in order to offer a tradeoff between energy consumption, latency, and throughput that can be tuned to meet the needs of the IoT. Using a clustering method and traffic distribution to limit rivalry between sensor nodes, the proposed solution makes the MAC protocol more energy efficient by reducing collisions and idle listening.

According to Rahul Jashvantbhai Pandya et al. [7], in indoor situations, the Internet of things (IoT) may benefit from bandwidth and power cost-optimized Medium Access Control (MAC) protocols via VLC. There is a considerable improvement in spectrum efficiency, throughput, and power efficiency with the suggested procedures They came up with four ways to simultaneously maximise bandwidth and power efficiency. The Restricted Client Position Aware Single Point of Communication Oriented Intelligent Clustering (RCPA-SPC-O-IC) had the lowest power consumption at 68 Watts, the highest bandwidth utilization of 98 percent, the lowest spare bandwidth of 12 percent, and the longest life expectancy of 165x104 hours.

Using a Harvested Energy-adaptive Medium Access Control (HEMAC) protocol, proposed by Hyeong-Kyu Lee et al. [8], an IoT Access Point (AP) can assign slots based on the number of IoT devices attempting to transmit data in a frame of time. Resource efficiency in the F-ALOHA MAC protocol is improved by the HE-MAC protocol. Through simulations, we demonstrate that the HE-MAC protocol is more efficient than the F-ALOHA MAC protocol in terms of resource use.

LCX-MAC (local coordination X-MAC) has been suggested by Arshad Ahmad et al [9] as an extension of X-MAC. A MAC protocol with an asynchronous duty cycle, X-MAC is a popular choice. Short preambles allowed X-MAC nodes to swiftly deliver their real data to receivers when they woke up, which is an essential feature of the X-MAC protocol. When the X-MAC node sends brief preambles to wake up its receiver node, it consumes energy, increases transmission latency, and fills up the channel because of the large number of short preambles that must be discarded in today's IoT healthcare applications, which need fast reaction times. X-MAC might be improved by allowing each node to exchange its local information with its neighbours. Compared to the nodes that are synchronised, the percentage of local information transmitted will have a significantly less effect on the overall performance.

3. PROPOSED SOLUTION

3.1 Overview

In this paper, Energy Efficient MAC protocol With Traffic-aware Wakeup Scheduling is proposed for IoT-WSN. In this protocol, the lengths of duty-cycle of the IoT sensor nodes are determined based on traffic aware duty-cycle adjustment technique. In this technique, the awake period of the nodes are adjusted based on the number of contenders and the traffic load at the node. In order to determine the optimal number of contenders, Q-learning algorithm is applied.

3.1 Sequential Scheduling

We assume that the network has just one sink node, which is located in one of the network's corners. Sensors in remaining nodes allow them to act as relays and send packets. Nodes are divided into numerous categories based on their distance from the sink node.

Figure 1 shows the network model. The type of a node is denoted by i, and the total number of types in the network is denoted by N.

The sensed data is delivered sequentially from a source node of type I to a source node of type $i - 1$, and so on, until they reach the destination node. Scheduler nodes function in accordance with the format of the frames they have received. The MAC protocol is used by Type I(i + 1) nodes that are both awake and sending packets at the same moment. At this time slot, all nodes of type (i + 1) simultaneously awake to receive the packet from type (I). Nodes of type I are transmitters, while nodes of type I – 1) are receivers in the following slot. As the process progresses, grade 1 nodes become transmitters and grade 1 sink nodes become receivers.

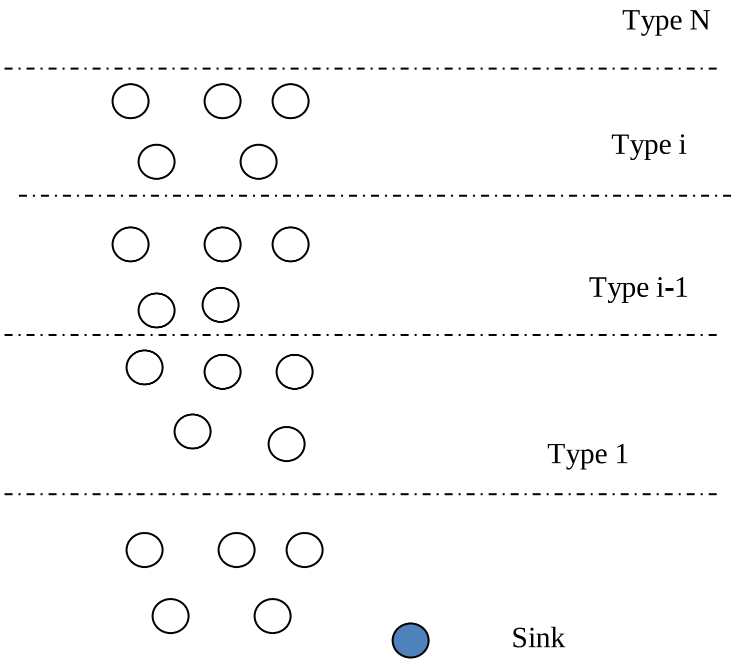

Figure 1 Network model

After receiving and sending data, all nodes of a specific kind are put into a sleep state for k slots to save energy. If tS is (k + 2) times the slot length, then tS is (k + 2) times the cycle length. If a data packet is successfully sent, as demonstrated in Figure 2, the handshake process is shown.

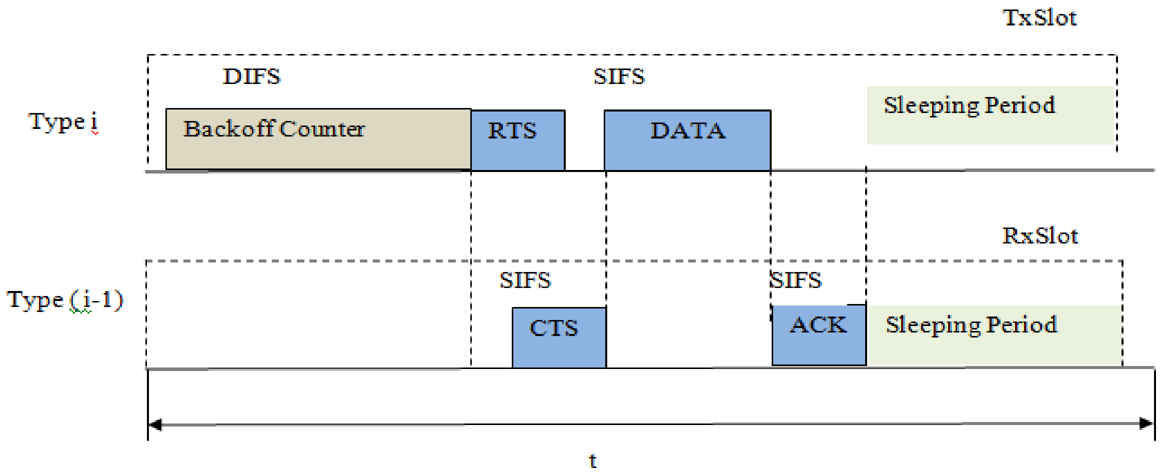

Figure 2 Transmission/reception model

3.2 Traffic Aware Wakeup Cycle

Let $AP_n(i)$ be the awakening probability of nodes at type i when the buffer is nonempty and let $EP(i)$ be the probability of having an empty buffer at type i. Hence the awakening probability for each node is defined as

$$AP(i) = AP_n(i)(1 - EP(i)) \tag{1}$$

This means that nodes at type i will wake up only if they have packets to transmit.

The probability of having n contenders besides the node of interest at type i is given by

$$C_n(i) = \binom{N-1}{n} AP(i)(1 - AP(i))^{N-1-n} \tag{2}$$

In the proposed protocol, nodes contend for the channel by generating a backoff counter w, which is a uniform random variable in $[0, -1]$.

The probability of winning the contention is

$$P_t(i) = \sum_{n=0}^{N-1} C_n(i) t_n \tag{3}$$

The probability of having a successful packet transmission as

$$P_s(i) = \sum_{n=0}^{N-1} C_n(i) S_n \tag{4}$$

3.3 Optimal Number of Contenders using Q-Learning

In this section the optimal number of contention nodes in the next contention period is determined using the Queue learning (QL) mechanism, based on the dynamic traffic rate.

3.3.1 Number of contention Nodes

The number of contenders during the contention period ct_i is derived as follows:

$$N_{con}^{i} = \begin{cases} CN_{avg} - \sum_{j=1}^{i-1} N_{suc}^{j} & 2 \le i \le n \\ CN_{avg} & i = 1 \end{cases} \tag{5}$$

Where CN_{avg} denotes the average number of contention nodes in one beacon interval, N_{suc}^{j} denotes the number of successful contention nodes in the contention period j.

The active rate of the number of contention nodes in contention period is defined as follows:

$$Rate_{active} = 1 - \frac{N_{sleep}}{N^{i}_{con}} \tag{6}$$

Where N_{sleep} denotes the number of sleeping nodes during the contention period.

All contention nodes in the next contention period include the fail contention node and the sleeping nodes in the previous contention period.

3.3.2 Basics of Q-Learning

With Reinforcement Learning (RL), the goal is clearly defined and the learning process is directed by this direction. Agents are often taught by trial and error when they interact with an unfamiliar environment. Agents are always learning and improving their performance based on feedback they get from their environments, which might be in the form of rewards (or penalties).

If you use the simplest form of Q-learning, each state-action combination has its own entry in the lookup table Q(s, a). The Q-learning algorithm uses the Bellman equation for the Q-value function to learn the best Q-value function. Here are a few definitions of some of the key terms in Q-learning:

A. Environment

Every real-life situation necessitates the creation of a specific environment. The process of defining an environment involves determining the rules that govern it, such as which activities an agent is permitted to perform, which states the environment has, and which rewards or penalties are associated with each condition. This state of things is a result of what an agent has accomplished, which is reflected in the environment's present input.

B. States [S=s0; s1,….., sn]

Agents can travel to or visit any number of conceivable states, locations, and positions within a given global environment under the overarching term "state." Using coordinates, alphabets, or integers, a state may be depicted. A visit is a transition from one state to another, and several visits, i.e. from the beginning state to the end state, is known as an episode.

C. Actions [A = a_0, a_1,...., a_n]

Anything an agent can think of or is permitted to perform in an environment is an action or collection of actions. In every given situation, an agent can take any action he or she wants.

D. Transition Model

In this setting, the rules, dynamics, and physics of a specific environment are explained. a method of establishing a new state The symbol T represents the current state, the intended action, and the new state. The probability of moving from state s (current state) to state s' (new state) by completing action a and obtaining reward R is given.

E. Reward

A Reward function R(s) returns a numerical value that an agent may receive for being in a specific condition following a specific action. Agents are paid more heavily when they move through useful states, and less heavily when they move through undesirable ones. As a kind of feedback, rewards may be either positive or negative for a user agent.

. The cumulative future reward also called return is given as:

$$R_t = r_{t+1}, r_{t+2}, r_{t+3} +,,,,,,+ r_{\infty} \quad (7)$$

Where subscript t denotes a specific time step.

3.3.3 Determining optimal number of contenders

A $Q(S_t, At)$ table is maintained by the agent in QL. The state S_t is observed by the agent in an IoT network for t = 1, 2, 3, ... From a list of available options, the agent will select one to execute (A). In response to action At, the agent receives R(t) and observes St+1. The order in which events occur (S_t,A_t,R(t), S_{t+1}) determines the agent's learning experience.The following QL function will be used to update the Q-sequence table's of events under the (St,At) heading.

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha.[R_{t+1} + \gamma.\max Q(S_{t+1,}, a) - Q(S_t, A_t)] \quad (8)$$

The agent will decide on a course of action in light of the existing state of affairs in S_t. The highest Q-value can be computed in the following state S_{t+1} depending on the current situation. The existing Q-value is then replaced with the maximum possible Q-value.

Prior to each beacon period, the number of contention nodes is adjusted by one of the following actions:

$$X_{t+1} \leftarrow X_t, a \in (X_t - d, X_t, X_t + d) \quad (9)$$

The ideal number of contention nodes at time step *t* is X_t, where d indicates the variance in the number of contention nodes.

A simple action selection rule is used to select the one with the highest estimated value from a variety of potential actions. If more than one "greedy" action is present, a random action is selected. Using the greedy-action selection strategy, we may say

$$\pi(s) = \arg\max_a Q(s,a) \quad (10)$$

Where $\arg\max_a$ denotes the action a for which the expression is maximized.

Hence, the interactions between the IoT node (agent) and the environment at time step t are as follows:

Q-Learning Algorithm

---

1 The IoT node monitors the IoT network and estimates the current number of contention nodes (S_t) after one beacon interval.

2 The IoT node obtains the active rate in the next contention period and then determines the next action A_t.

3 The IoT node applies the selected action At in the MAC protocol.

4. After one time step, the IoT node obtains the feedback reward R_{t+1} of the IoT network.

4 The IoT node moves from the state S_t to the new state S_{t+1}.

3.4 Traffic Estimation

The traffic load at a node N_i is given by

$$TL_i = \frac{.(ar_i + fr_i + cr_i)}{c.\eta.} \qquad (11)$$

Where c is the capacity of the radio, η is the maximum expected utilization of capacity, ar and fr are the packet arrival and forwarding rate and cr is the collision rate. Using Eq. (12), we can determine the total traffic load at node N_i during the superframe period T_c.

3.5 Traffic-aware Duty-Cycle Adjustment

The awake period of a node is then adjusted based on the traffic load TL and the number of contenders C as follows:

If TL_i< LB or C < X then
$T_{awake} = T_{awake} - \omega_1$
If TL_i > UB or C > X then
$T_{awake} = T_{awake} + \omega_2$

In duty-cycle adjustment, two thresholds LB and UB are maintained for the TL. If either the estimated traffic (TL) is more than UB or the contenders C of the device is higher than the optimal contenders X, then T_{awake} can be increased by a huge increment ω_2. On the other hand, either TL is less than than LB or the contenders C is less than X, T_{awake} can be decreased with a small increment of ω_1.

4 EXPERIMENTAL RESULTS

4.1 Simulation settings

The proposed TWS-EEMAC protocol is implemented in NS2 and compared with Selective-Awakening MAC (SA-MAC) protocol [1]. The performance is evaluated in terms of the metrics end-to-end delay, packet delivery ratio, throughput and energy consumption.

The experimental settings a are shown in Table 1

No. of Nodes	21,41,61,81 and 101
Area	50 X 50
MAC	802.11
Simulation Time	50 sec
Traffic Source	Exponential
Flows	4
Propagation	TwoRayGround
Antenna	OmniAntenna
Initial Energy	10.1J
Transmission Power	0.3
Receiving Power	0.3

Table 1: Simulation parameters

4.2 Results

In this section, the results of varying the number of nodes from 20 to 100 are presented.

Nodes	**TWS-EEMAC (ms)**	**SA-MAC (ms)**
20	1.40	2.01
40	1.72	2.49
60	2.61	3.45
80	2.71	3.69
100	3.52	4.38

Table 2 Results of end-to-end Delay

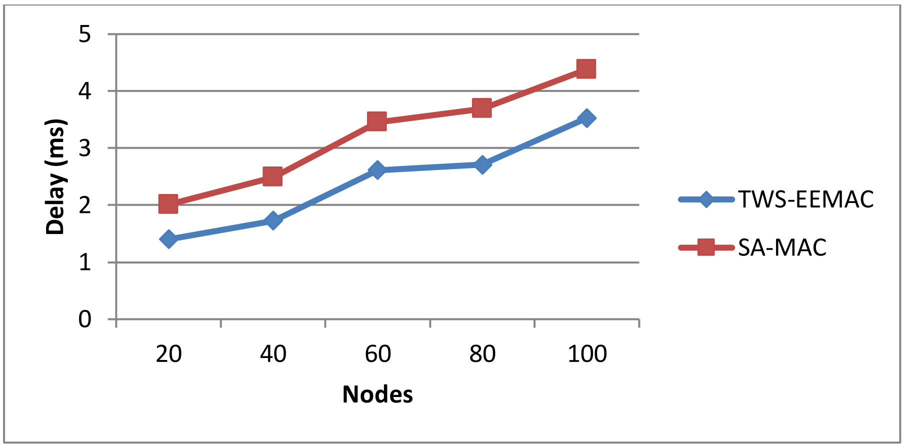

Figure 3 Delay for varying nodes

Table 2 and Figure 3 present the end-to-end delay values obtained by increasing the number of nodes. The figure shows the delay of TWS-EEMAC is 26% lower than SA-MAC.

Nodes	TWS-EEMAC	SA-MAC
20	0.9522	0.9285
40	0.9336	0.9025
60	0.9243	0.8733
80	0.9149	0.8602
100	0.9097	0.8475

Table 3 Results of Packet Delivery Ratio

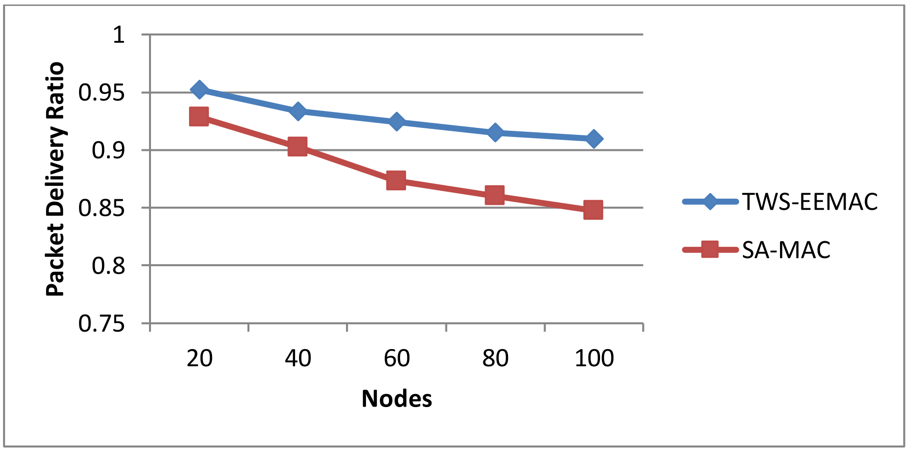

Figure 4 Packet Delivery Ratio for varying nodes

Table 3 and Figure 4 present the packet delivery ratio values obtained by increasing the number of nodes. The figure shows the delivery ratio of TWS-EEMAC is 5% higher than SA-MAC.

Nodes	TWS-EEMAC (Joules)	SA-MAC (Joules)
20	6.13	6.88
40	6.28	7.26
60	6.45	7.35
80	6.74	7.54
100	7.13	7.72

Table 4 Results of Energy Consumption

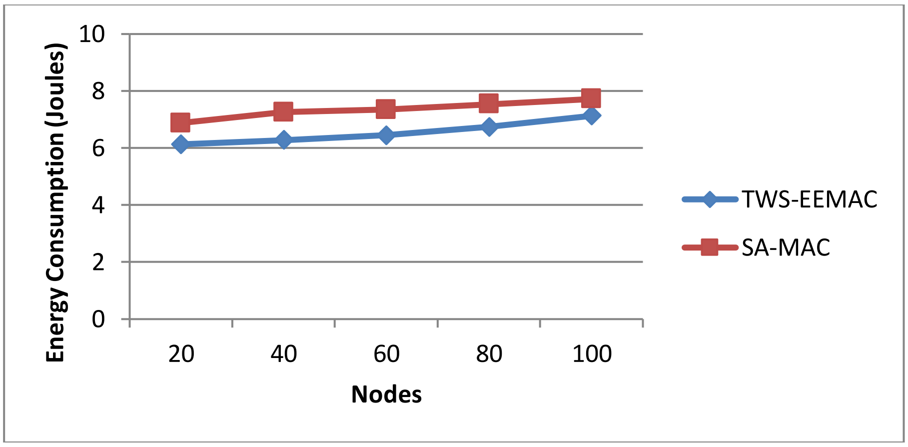

Figure 5 Energy Consumption for varying nodes

Table 4 and Figure 5 present the energy consumption values obtained by increasing the number of nodes. The figure shows the energy consumption of TWS-EEMAC is 11% lower than SA-MAC.

Nodes	TWS-EEMAC (Mb/s)	SA-MAC (Mb/s)
20	1.44	1.31
40	1.35	1.28
60	1.31	1.26
80	1.30	1.20
100	1.24	1.17

Table 5 Results of Throughput

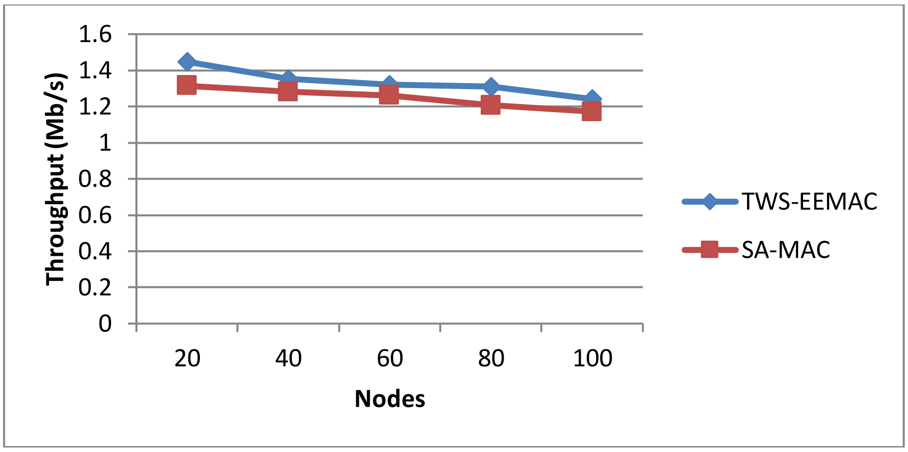

Figure 6 Throughput for varying nodes

Table 5 and Figure 6 present the throughput values obtained by increasing the number of nodes. The figure shows the throughput of TWS-EEMAC is 6% higher than SA-MAC.

5. CONCLUSION

In this protocol, the lengths of duty-cycle of the IoT sensor nodes are determined based on traffic aware duty-cycle adjustment technique. In this technique, the awake period of the nodes are adjusted based on the number of contenders and the traffic load at the node. In order to determine the optimal number of contenders, Q-learning algorithm is applied. The proposed TWS-EEMAC protocol is implemented in NS2 and compared with SA-MAC protocol. The performance is evaluated in terms of the metrics end-to-end delay, packet delivery ratio, throughput and energy consumption. Performance results show the superior performance of TWS-EEMAC in terms of end-to-end delay, packet delivery ratio, throughput and energy consumption, over the existing SA-MAC protocol.

REFERENCES

1. Iclia Villordo-Jimenez ,Noé Torres-Cruz ,Marcelo M. Carvalho , Rolando Menchaca-Mendez ,Mario E. Rivero-Angeles and Ricardo Menchaca-Mendez, "*A Selective-Awakening MAC Protocol for Energy-Efficient Data Forwarding in Linear Sensor Networks*", Hindawi Wireless Communications and Mobile Computing, Volume 2018, Article ID 6351623, 18 pages, 2018.

2. Chien-Min Wu, Yu-Hsiu Chang, Cheng-Tai Tsai and Cheng-Chun Hou, *"A Q-Learning-Based Distributed Queuing MAC Protocol for Internet-of-Things Networks"*, DOI: https://doi.org/10.21203/rs.3.rs-1554570/v1, 2022.

3. Atul Kumar Pandey and Nisha Gupta, *"An Energy Efficient Clustering-based Load Adaptive MAC (CLA-MAC) Protocol for Wireless Sensor Networks in IoT"*, I.J. Wireless and Microwave Technologies, 2019, 5, 38-55, 2019.

4. Syed Waqas Haider Shah, Adnan Noor Mian, Adnan Aijaz and Jon Crowcroft, *"Energy-Efficient MAC for Cellular IoT: State-of-the-Art, Challenges, and Standardization"*, arXiv:2103.05311v1 [cs.NI] 9 Mar 2021, 2021.

5. Abdulrahman Sameer Sadeq, Rosilah Hassan, Hasimi Sallehudin, Azana Hafizah Mohd Aman and Anwar Hassan Ibrahim, *"Conceptual Framework for Future WSN-MAC Protocol to Achieve Energy Consumption Enhancement"*, Sensors, Vol-22, 2022.

6. Daewoo Kim, Jinhwan Jung, Yoonpyo Koo, and Yung Yi, *"Bird-MAC: Energy-Efficient MAC for Quasi-Periodic IoT Applications by Avoiding Early Wake-up"*, IEEE, 2019.

7. Rahul Jashvantbhai Pandya, *"Bandwidth and Power Cost Optimized MAC Protocols for IoT over VLC"*, IEEE-HYDCON, 2020.

8. Hyeong-Kyu Lee, MinGyu Lee and Tae-Jin Lee, *"HARVESTED ENERGY-ADAPTIVE MAC PROTOCOL FOR ENERGY HARVESTING IOT NETWORKS"*, NETCOM, NCS, WiMoNe, CSEIT, SPM - 2015, 2015.

9. Arshad Ahmad ,Ayaz Ullah, Chong Feng , Muzammil Khan , Shahzad Ashraf, Muhammad Adnan, Shah Nazir and Habib Ullah Khan, *"Towards an Improved Energy Efficient and End-to-End Secure Protocol for IoT Healthcare Applications"*, Wiley, Hindawi Security and Communication Networks, Volume 2020, Article ID 8867792, 10 pages, 2020.