IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 10, 2020, accepted August 24, 2020, date of publication September 1, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021016

Reinforcement Learning Based Adaptive Duty Cycling in LR-WPANs

SHAHZAD SARWAR[1], RABIA SIRHINDI[1], LAEEQ ASLAM[1], GHULAM MUSTAFA[2], MUHAMMAD MURTAZA YOUSAF[1], AND SYED WAQAR UL QOUNAIN JAFFRY[1], (Senior Member, IEEE)

[1]National Centre of Artificial Intelligence, Punjab University College of Information Technology, University of the Punjab, Lahore 54000, Pakistan
[2]Department of Informatics and Systems, School of Systems and Technology, University of Management and Technology, Lahore 54782, Pakistan

Corresponding author: Laeeq Aslam (laeeq.aslam@pucit.edu.pk)

ABSTRACT For conserving energy, duty cycle is defined by setting up the active and sleep periods of network nodes. In beacon enabled networks, to provide support for duty cycle, the IEEE 802.15.4 standard uses optional super-frame structure. This duty cycle is usually fixed and does not consider the topology changes that often occur in dynamic sensor networks. In this paper, existing energy conserving duty cycling approaches for 802.15.4 networks especially the adaptive duty cycling techniques for wireless sensor networks are summed up. Also, this paper highlights the shortcomings of the proposals in the literature, such as induced additional latency, so that they may not support the practical scenarios of Internet of Things (IoT). Further, this study highlights a gross shortcoming that relative performance comparison of RL-based proposals cannot be performed without using a benchmarking framework and real test-bed environment. In this paper, we have presented the future research directions that would lay the foundation for successful development of energy efficient RL-based duty-cycling techniques.

INDEX TERMS Duty cycling, IEEE 802.15.4, reinforcement learning, super frame parameters.

I. INTRODUCTION

A wireless sensor network (WSN) comprises of a typically scattered set of sensor nodes in an environment. Sensor nodes send their data to a centralized base station (BS) at regular intervals. Such nodes are typically constrained by the available resources like battery power and capacity of communication channels. On the other hand, the base station usually does not face restrictions in data processing, power consumption, and communication. The former set of nodes usually comprises the constrained network and the later forms the unconstrained network. In the Internet of Things (IoT), we are interested in the point-to-point, broadcast, anycast and point-to-multipoint types of communication between nodes of the two types of networks [1].

The limited resources of sensor nodes introduce challenges in the design of IoT applications and related protocols. Most of standards developed for IoT protocol stack pay special attention to battery and processing power of sensor nodes. Examples include the routing protocol for low power and lossy networks (RPL) [2], constrained application protocol (CoAP) [3], and IPv6 over low power wireless personal area networks (6LowPAN) [4]. Similarly, energy conservation approaches are needed at the medium access control (MAC) layer where IEEE 802.15.4, defined in [5], is used for contention based random access of physical medium. If a sensor node remains active all the time, it is likely to lose battery rapidly and becomes dead. To achieve maximum network throughput and lifetime, we need to alternate nodes between sleep and active cycles. However, current designation of IEEE 802.15.4 standard does not specify the procedure for configuring duty cycle of sensor nodes to conserve energy under varying traffic load, without significantly compromising the network efficiency. Thus, we need to do wake-up scheduling of nodes and it involves switching of the sleep and active states of sensor nodes. In active state, the CPU, sensors, and radio of a sensor node remains ON. Similarly, a node can sense, process, and communicate the environmental information to other nodes, when it is in active state. On the other hand, CPU, sensors, and radio remains OFF when the node is in sleep state and it can run with a negligibly small amount of energy. However, nodes in sleep mode cannot transmit or receive data. Different approaches can be adopted to manage the cycle between active and sleep period.

The associate editor coordinating the review of this manuscript and approving it for publication was Rakesh Matam.

This paper provides an overview of different approaches used for scheduling of duty cycles of nodes, with a particular focus on reinforcement learning (RL) based approaches [6]. RL-based approach enables the learning of optimal duty cycle of nodes, so that it could be used during the life-span of the network.

The paper is organized as follows. Section II provides an overview of medium access mechanism in WSNs. Section III provides the details of IEEE 802.15.4 standard. Intelligent approaches for adaptive medium access control are discussed in Section IV. The Section V discusses RL-based approaches and their learning challenges in constrained environments. In the Section VI performance of different techniques is compared and discussed. The future research directions are presented in the Section VII and finally the paper is concluded in the Section VIII.

II. MEDIUM ACCESS IN WSN

MAC protocols in wireless sensor networks have the following types: random-based, slot (schedule)-based, hybrids (random/slot based), and low-power listening (LPL). LPL protocols are currently used for duty cycling. LPL uses scheduled transmissions to ensure low power consumption in WSNs. Taxonomy of LPL protocols for machine-to-machine (M2M) networks is presented in [7]. MAC protocols are significantly good in dealing with broadcast, interference, and packet collision issues in a wireless network. Traditionally, such issues are resolved by packets (re)transmission approaches, modulation schemes, using packet length, and transmission control [8]. However, such techniques do not work well in WSN due to decentralized control, highly directed data traffic (towards few sink units), volatile links, and large number of energy limited nodes [8]. MAC protocol minimizes energy consumption by scheduling the wake-up and sleep periods. Reliability, scalability, longevity, latency, and fairness are the core factors relevant in MAC protocol design, in addition to the network throughput.

Traditionally, the access to the shared medium in WSNs is regulated by two approaches: (a) reservation or schedule based and (b) contention based. In reservation-based protocols, schedule of node communication is decided on the basis of network topology. Time division multiple access (TDMA) is an example of this approach. In TDMA, a unique time-slot is assigned to each node for transmission of data. This approach avoids collisions during transmissions. However, throughput is limited by the slot-length. In WSNs, scheduling is a vital aspect of TDMA based protocols. Complexity in infrastructure-less networks, scalability of slot assignment schedule, broadcast communication, reduced flexibility, and maintenance of memory status are the central issues that should be addressed by effective scheduling protocols. Many canonical protocols are designed to address these issues. There are various MAC based scheduling protocols in literature based on centralized and distributed scheduling functions. A summary of these protocols is presented in the Table 1.

TABLE 1. Examples of different MAC protocols.

Protocol Type	Examples
Contention-based	CSMA/CA [10], MACA [12], Sift [13], CSMA/ARC [14], PAMAS [15], PSM [16]
Slotted (TDMA-based)	TSMP [19], Arisha [20], PEDAMACS [21], BitMAC [22], G-MAC [23], SMACS [24], TRAMA [25], FLAMA [26], μMAC [27], EMACS [28], PMAC [29], PACT [30], BMA [31], MMAC [32], FlexiMAC [33], PMAC [34], O-MAC [35], LMAC [36], RMAC [37]
Preamble-based	Preamble sampling ALOHA [38], preamble sampling CSMA [39], Cycled Receiver [17], Channel Polling [18], BMAC [40], EL-ALPL [41], X-MAC [42], CMAC [43], WiseMAC [44], LPL [7]
Hybrid	IEEE 802.15.4 [5], Z-MAC [45], Funneling MAC [46], Scheduled Channel Polling [18], Crankshaft [48]

In contention-based protocols (CBPs), WSN nodes compete with each other to gain access to the wireless medium, in the absence of a central scheduling authority. However, CBPs are agnostics of network topology and clock synchronization. The carrier sense multiple access (CSMA) and ALOHA [9] are popular examples of contention-based random-access protocol. However, performance of these protocols degrades under high load of traffic [10], [11]. For energy optimization, various canonical approaches have been introduced based on CSMA that reduce collisions, overhead, overhearing and idle listening [8]. Carrier sense multiple access with collision avoidance (CSMA/CA) [10], multiple access collision avoidance (MACA) [12], and Sift [13] are recognized contributions that are used to reduce collisions. To reduce protocol overhead, various optimizations have been proposed such as adaptive rate control in CSMA (CSMA/ARC) [14]. It introduces a back-off that is relaxed according to the frequency of transmission of the application. Overhead is further reduced by eliminating the use of explicit acknowledgment (ACK) messages, request-to-send (RTS) and clear-to-send (CTS) intervals, as well. Power-aware multi access with signaling (PAMAS) [15] is yet another improvement that deals with the over hearing issue in CSMA-based protocols. MAC protocols in WSNs IEEE 802.11 power save mode (PSM) focuses on reducing the idle listening issue by offering sleep and active modes [16]. However, it does not support multi-hop networks as well as increases network latency.

In contrast to protocols that use common active periods, preamble sampling-based MAC protocols do not have a common active/sleep schedule for each node. Rather, each node decides about its active period scheduling independently from other nodes [8]. A preamble precedes every data frame that ensures that all recipients detect the preamble before getting data frame. A node samples the medium according to the duty cycle parameters and goes to sleep mode if the channel is found to be idle. Conversely, if a node senses a preamble

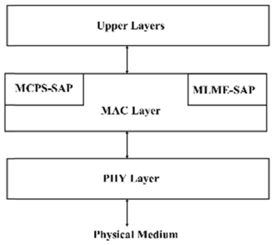


FIGURE 1. MAC sub-layers in IEEE standard.

being transmitted, it remains active until the successive data frame is also received.

For successful reception of data by a receiving node, the length of the preamble should be as long as the check interval (CI). The CI ensures that all nodes are awake during the preamble and subsequent data frame transmission. LPL [7], cycled receiver [17], and channel polling [18] are examples of preamble sampling protocols. These protocols have a benefit of less energy consumption at the cost of longer preambles. Long preambles cause collisions and limit the duty cycle of nodes.

Some hybrid alternatives also exist at MAC layer to provide the combined features of contention-based, slotted, and preamble sampling-based protocols. Hybrid protocols focus on the assurance of flexible MAC frame structure, CSMA inside TDMA slots for dynamic traffic, hybrid sampling and slotted to minimize preamble cost, minimization of convergence cost effect, and receiver-based scheduling. Examples of different MAC protocols, used in wireless sensor networks, are given in Table 1.

III. IEEE 802.15.4 STANDARD

MAC and physical sub-layers for low-rate wireless personal area networks (LR-WPANs) are detailed in IEEE 802.15.4 standard [5]. The standard uses CSMA/CA and offers contention based medium access. However, schedule-based access can also be provided by using time slots for the devices that deal with delay sensitive data. The network can operate in star or peer-to-peer topologies as per the standard. The standard consists of specifications for physical and MAC sub-layers. MAC layer interacts with higher layers using two service access points, known as MAC common part sub-layer (MCPS) and MAC sub-layer management entity (MLME), as shown in Fig. 1. Functions of physical layer include detection of energy consumption, activation/deactivation of radio transceiver, selection of channel, indication of link quality, clear channel assessment (CCA), and transmission of data units of physical layer protocols.

The MAC sub-layer helps in beacon management, time slot management, frame validation, channel access, association and dissociation, frame delivery (with acknowledgements), and providing support for implementing application specific security techniques.

A. SUPER FRAME STRUCTURE

IEEE 802.15.4 optionally supports the use of super-frame structure. The format of super frames is defined by the PAN coordinator. It is restricted by the transmission of a beacon frame at the start and end of super frame. Beacons are used in synchronization of devices, identification of the network and definition of super frame structure [5]. Each super frame is partitioned into 16 equal slots. Also, it can have active and inactive portions. A coordinator can switch to less-power mode during the inactive duration.

If the coordinator does not want to utilize super frame structure, it can break off beacon's transmission at the initiation of first slot of the frame. Two PAN attributes namely *macBeaconOrder* (BO) and *macSuperframeOrder* (SO) are used to describe the structure of the super frame.

Let us assume D_{BS} denotes *aBaseSuperframeDuraion*. D_{BS} is a MAC sublayer constant and is equal to the number of symbols in a superframe *of* order zero. BO refers to the time duration at which PAN coordinator broadcasts beacon frames. Beacon interval (BI) and BO are related to each other as shown in Equation 1.

$$BI = D_{BS} \times 2^{BO} \tag{1}$$

where, $0 \leq BO \leq 14$.

For BO = 15, beacon frames are not transmitted except if explicitly requested. The attribute SO defines the length of the active period of the super frame including the beacon frame. The super frame duration (SD) is calculated from SO using the Equation 2.

$$SD = D_{BS} \times 2^{SO} \tag{2}$$

where, $0 \leq SO \leq BO \leq 14$.

A device that wishes to communicate, has to contend with other devices by employing a mechanism of ALOHA or CSMA/CA during the contention access period (CAP). Some portions of the active super frame are reserved by the PAN coordinator for sensitive applications or applications that require specific bandwidth. These reserved parts of the active period are termed as guaranteed time slots (GTSs). These GTSs collectively make the contention-free period (CFP) which begins immediately after the CAP, usually at the end of the active period, as shown in Fig. 2. A GTS can constitute more than one slot and seven GTSs can be allocated. However, an ample portion of the CAP is still left for random access by other existing nodes in the network, or new nodes that want to join the network. The CFP begins when all contention-based transactions have been completed. Also, each device, using a GTS for transmission, makes sure that its transaction is finished before the next upcoming GTS or the end of the CFP.

An un-slotted channel access mechanism is used by non-beacon-enabled networks. A device has to wait for a random period of time whenever it has to transmit data frames or MAC commands. This is called the random back off time. Data can only be transmitted by the device if the channel is inactive after the random back off. If the channel is busy then the device will wait for another unplanned period before

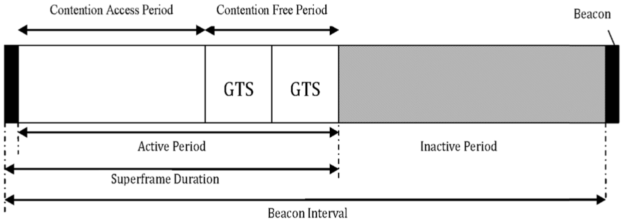


FIGURE 2. IEEE 802.15.4 superframe structure.

the next random channel access. Transmission of acknowledgment frames does not use the CSMA/CA mechanism. Beacon-enabled networks employs a slotted CSMA/CA channel access. Here, back off slots are synchronized with the beginning of beacon frame. To find the start of the upcoming back off slot, a device has to wait for some random amount of back off slots for transmission during the CAP. If the medium is busy then the device has to wait for another random number of back off slots before accessing the channel again. If the medium is unused, the device can start to transmit data on the upcoming back off slot boundary available. Acknowledgment and beacon frames are transmitted without a carrier sensing mechanism.

B. DUTY CYCLING

The IEEE 802.15.4 standard does not specify how to configure activation time of wireless nodes for optimal network performance under varying traffic load. Moreover, appropriate SO and BO parameters need to be set during the initialization of network, in beacon enabled mode. Non-optimal BO and SO values may lead to wastage of power and channel capacity. The four major factors responsible for energy wastage are the following.

1) COLLISIONS

When a transmitted frame collides with other frames, it becomes corrupted and has to be neglected. This leads to retransmissions and increased energy consumption. Collisions result in high latency as well.

2) OVERHEARING

This happens when a node receives frames that are bound for other nodes.

3) CONTROL FRAME OVERHEAD

This refers to sending and receiving frames containing control information, and it consumes energy too.

4) IDLE LISTENING

This refers to listening to the channel in anticipation of data that has not yet been sent, and occurs commonly in WSNs. If nothing is being sensed then this implies that nodes are in an idle mode for maximum number of times. Measurements indicate that idle listening energy can match the energy required to receive a message [47].

IV. MAC PROTOCOLS WITH ADAPTIVE DUTY CYCLING

Duty cycling is a mechanism to manage major energy wastages factors. Also, packet transmission scheduling helps reduce collisions. However, such scheduling techniques may incur non-deterministic transmission latency due to the longer sleep times of nodes. The contention-based approaches that implement duty cycling can be categorized into two main classes: synchronous and asynchronous. The sender sends a preamble frame which is received by the receiver once it wakes up, followed by data frames. Asynchronous scheduling introduces longer delays than the synchronous approach. Also, the preamble frame incurs overhead and consumes extra energy. The synchronous approaches such as C-MAC, S-MAC, DW-MAC, and T-MAC synchronize the transmission schedules and duty cycles among neighboring nodes. Energy consumption due to idle listening is reduced by sending data packets when the neighboring nodes are active. However, extra energy is used by control frames sent to synchronize duty cycling among neighboring nodes.

In [49], a dynamic duty cycle (DDC) scheme has been proposed for minimizing the delay in WSNs. The DDC scheme suggests prolonging the active period of nodes proportional to the residual energy, in non-hotspots areas of the network. The residual energy is used to increase the listening time of node and its availability which enhances data forwarding but reduces transmission delay. Similarly, an adaptive duty cycle control–based opportunistic routing (ADCCOR) scheme has been proposed to reduce the end-to-end delay. In ADCCOR, duty cycle of a node is adaptively set in direct proportion to residual energy of node. The study [50] shows the end-to-end network delay can be reduced, up to 70%, without reducing the network lifetime. By adding hardware of wake-up radio (WuR) in nodes, a node can wake up only on-demand, at any time even in the sleep state, when there is data to be transmitted [51]. Whereas, in duty cycling, a node wakes up regularly according to its duty cycle. A WuR-based scheme for relay selection consecutive packet routing (RS-CPR) has been proposed in [52]. In RS-CPR, relaying node is selected with much residual energy, a large number of packets, and a short distance from sink by other nodes.

The following sections provide an overview of synchronous MAC protocols that use common active periods for duty cycling.

A. S-MAC

Ye *et al.* [47], [53] propose a medium access control protocol for wireless sensor networks that considers the constrained nature of sensor nodes that remain inactive most of the time. The primary goal of sensor MAC (S-MAC) is to provide self-configuration and energy conservation while latency and node fairness are secondary goals. It allows low duty cycle in multi-hop networks and uses the concept of virtual clusters where nodes are grouped according to their sleep schedules. The protocol also claims to achieve collision avoidance and valuable scalability using a mutual contention and scheduling scheme.

S-MAC aims to minimize energy consumption overhead from all major energy wastages factors. It puts nodes to sleep periodically, as opposed in IEEE 802.11 networks where nodes always remain in active mode, thus saving battery and network life. Low-duty-cycle is the default mode in S-MAC, where nodes wake up only during the existence of traffic in the network. The authors have implemented low-duty-cycle scheme in multi-hop networks with periodic listen and sleep modes that greatly reduces energy consumption by avoidance of idle listening. Also, it presents a technique of adaptive listening that minimizes the delay occurred due to periodic sleeping. In-channel signaling is used to avoid energy consumption on overhearing and message passing is introduced to minimize control overhead and application-perceived latency. S-MAC is evaluated for energy, latency and throughput on Berkeley Motes.

Each node sleeps and hibernates for some random time period and sets a timer to awake itself later. After the sleep periods, it wakes up to check if any other node wants to communicate to it. During inactive time the radio of the node is turned off. The active interval of node is fixed based on the physical and MAC layer parameters; like contention window size and radio bandwidth. The sleep interval can be modified with respect to specific application requirements thus changing the duty cycle. The sleep and awake time values are consistent for all nodes. Moreover, all nodes can select their own schedules for sleep/listen. However, virtual clusters of nodes are created, that share the same neighborhood, to minimize control overhead. For this, the nodes need to synchronize with each other so that the sleep/listen schedule start at the same time and they listen for same duration and sleep for the same duration. It should be observed that all neighboring nodes cannot synchronize in a multi-hop network. The node schedules are broadcasted using a SYNC packet. An advantage of S-MAC is that it uses a peer-to-peer topology instead of using only cluster heads to communicate with each other.

Each transmitted frame has a field that defines a duration for which the remaining transmission will continue. If a node gets a frame meant for another node, it can determine how long to remain in sleep mode using this field. The value of this field is stored in a variable called the network allocation vector (NAV) and a timer is set. As soon as the timer expires, NAV is decremented until it reaches zero. Before starting a transmission, a node first looks at its NAV. A non-zero NAV value indicates that the medium is busy. This is referred to as virtual carrier sense. It is performed by all nodes before starting a transmission. A node goes to sleep if it finds the medium busy and wakes up only when the receiver starts listening again. Broadcast frames are transmitted without using CSMA request-to-send (RTS) and clear-to-send (CTS) packets. However, unicast packets follow the normal sequence of RTS/CTS/DATA/ACK between the sender and receiver. Data transmission is done following the successful communication of RTS/CT between two nodes.

Sleep schedules are coordinated among nodes in S-MAC rather than randomly sleeping. When an interfering node overhears an RTS or CTS message meant for another node, it goes to sleep. This is how S-MAC avoid overhearing. Since DATA frames are usually long in length than control frames, this approach prevents overhearing of long frames by neighboring nodes.

B. T-MAC

Timeout-MAC (T-MAC) [54] transmits all messages in bursts of variable sizes and sleeps in between the bursts. The duration of active time (TA) is defined dynamically and it ends if nothing is heard on the channel for a specific period of time.

The protocol works as follows. Each node wakes up occasionally to communicate with its peers, sleeps again until the arrival of the next frame. Queue of upcoming messages is maintained. Nodes communicate with each other using RTS/CTS/Data/ACK scheme, which ensures collision avoidance and reliable transmission. An active period is triggered by an activation event (discussed below) and terminates when no such event has happened for a time. The activation timer gives an upper bound on the duration for idle listening time of a node at the end of the active period. The following five events can trigger activation of a node.

- Expiration of periodic frame
- Message reception
- Collision sensing
- Acknowledged transmission of data
- Data transmission of neighbor node is finished

A node will sleep if it is not in an active period or is idle. This timeout scheme moves all communication as a burst to the start of the frame. The buffer capacity defines an upper bound on the maximum frame time because messages during the sleep time have to be buffered.

T-MAC also provides clustering and synchronization, as similar to S-MAC. In addition to this, it uses a fixed contention interval which means that the contention interval remains the same under high and low traffic loads. Contention time is used even if there is no collision. T-MAC also uses RTS retries to avoid the early sleeping problem. If a node sends an RTS and does not receive a reply in TA time, it retries by sending a subsequent RTS. If there is still no reply from the receiver, the sending node changes its mode to sleep. This prevents the sending node from sleeping early in case of a collision or overhearing RTS/CTS. Moreover, the activation timer should be long enough to receive the start of a CTS message in response to an RTS. As soon as the CTS is received, the TA is also renewed. The lower limit on the length of this interval can be given as,

$$TA > C + R + T \tag{3}$$

where, C is the duration of the RTS contention interval, R is the RTS packet length and T is the turnaround time between the end of an RTS and start of CTS packet. In [54] TA is set to be $1.5 \times (C + R + T)$. A large value of activation interval can lead to more energy consumption. The protocol also supports overhearing avoidance as proposed in [53]. However, it is noted that collisions increase as a result because a node may miss hearing other nodes exchanging RTS/CTS messages when it wakes up and starts sending right away.

It can be concluded that time-outs are an effective way to resolve the idle listening issue under variable load of traffic. The implementation of T-MAC has shown energy conservation both in simulation in OMNET++ [55] and on real hardware (EYES nodes) [56]. It is shown that during a high load, nodes communicate without sleeping however, during low traffic load the nodes will use their radios as little as 2.5% of the time, resulting in 96% of energy conservation compared to a traditional MAC protocol.

C. Z-MAC

A hybrid MAC protocol, called Zebra-MAC (Z-MAC), has been presented for WSNs in [45]. The Z-MAC protocol merges the benefits and balances the drawbacks of TDMA and CSMA. Its main feature is adaptability to network contention levels, acting like CSMA under low contention level while attaining low latency and high channel utilization. Under high contention levels, similar to TDMA, Z-MAC achieves high channel utilization and reduces collisions between two-hop neighbors with minimal control overhead. A unique characteristic of Z-MAC is that it performs well despite slot allocation failures, synchronization errors, and dynamic channel conditions.

Z-MAC has a setup phase in which the following operations are performed in sequence: (i) discovery of neighbor, (ii) assignment of slot, (iii) exchange of local frame, and (iv) synchronization of global time. These operations are performed exactly once during the setup phase and are repeated only if a major change occurs in the network topology. The initial high overhead, caused by setting up the network, is compensated by energy efficiency and improved throughput over a longer network operation. As soon as a node starts up, neighbor discovery protocol is run where a ping is broadcasted to its one-hop neighbors periodically. This is done to create a one-hop neighbor list of this node. Each node transmits one ping message per second at a random time and does so for 30 seconds. In this way, each node aggregates the information it receives from its one-hop neighbors using ping operations, thus gathering its two-hop neighbor information. The list of two-hop neighbor is then used for time-slot allocation. Z-MAC uses DRAND [57] algorithm to assign time slots to each node of network using the two-hop neighbor list of each node. DRAND ensures that nodes lying within a neighborhood of two-hop communication are not assigned same slot in a broadcast schedule. This guarantees the minimization of interferences in two-hop neighbor communication. It should be noted that broadcast schedule is used to cater the routing changes occurring among neighbors at one-hop.

After the time slot assignment, each node defines the period which is the time interval that can be used for transmission. This period is referred to as time frame of the node. A time frame rule helps nodes to pick their own time frame lengths, based on their local neighborhood information. At the termination of DRAND phase, each node sends its time frame and slot number to its neighborhood at two-hop distance. This is followed by the transmission phase of Z-MAC. A node can function in one of two modes in Z-MAC: (i) high contention level (HCL) and (ii) low contention level (LCL). If a node gets an explicit contention notification (ECN) from any two-hop neighbor with in a contention time interval, then node is in HCL. Otherwise, the node is in LCL. In LCL a node can transmit in any slot where as in HCL only owner and their one-hop neighbors can contend for a slot for transmission. Owners of a slot are given higher priority to send data over non-owners in both HCL and LCL modes. But if the owner does not have data to send, then non-owners can take the slot for transmission. This allows for high utilization of channel, even in LCL mode. Node always performs carrier-sensing and transmission of a packet occurs when the channel is clear.

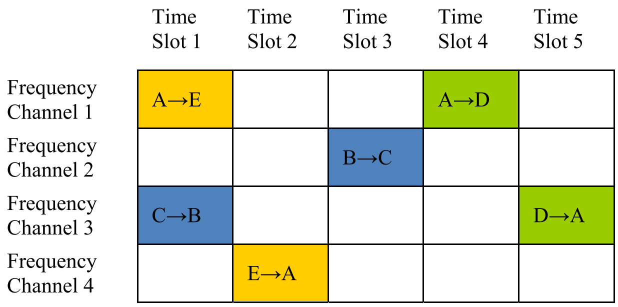
The Z-MAC protocol deals with the loosely synchronized clocks and knowledge of topology to enhance MAC performance in scenarios involving high contention. Under low contention the protocol resorts to acting like CSMA. Z-MAC fits best for applications where two-hop contention and expected data rates are medium or too high.

D. DCA

A new algorithm for duty cycling in IEEE 802.15.4 networks has been proposed in [58]. The idea is to use a parameter called MAC status index (MSID) in addition to the existing MAC parameters of the standard. It represents the buffer occupancy and queuing delay at the MAC layer and is used to control the duty cycle by the PAN coordinator so that packet loss is minimized and energy efficiency is maximized. It is assumed that the BO is constant and the SO is set by the PAN coordinator adaptively. The MSID is used to show the MAC status of the end device such as queue sizes and queue wait times to the PAN coordinator, which only know the number of transmitted packets and end devices during the active period. Queuing delay tells whether the end device in the network has been a part of the contention in a given active period. Buffer occupancy and queuing delay are combined to express the MSID in 8 levels. The buffer occupancy is categorized into four levels and the queuing delay into three levels. Three bits (7-9) of the reserved field in the MAC control frame are used to represent MSID. To reflect the queuing delay, the MSID is modified at every last back off count slot. The coordinator finds the SO using parameters collected during an active duration and MSID value. The number of sending nodes and number of received packets are also measured. The previous SO value and number of end devices are known.

E. TSCH

The scheduled nature of Time synchronized channel hopping (TSCH) makes it different from traditional low power MAC protocols [59]. TSCH divides time into timeslots grouped to form a slot frame. However, no slot frame size



	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 4	Time Slot 5
Frequency Channel 1	A→E			A→D	
Frequency Channel 2			B→C		
Frequency Channel 3	C→B				D→A
Frequency Channel 4		E→A			

FIGURE 3. A typical TSCH schedule.

is imposed by TSCH. It can range from 10s to 1000s of timeslots. Smaller slot frame provides increased availability at the cost of higher energy consumption. In TSCH network, every node follows a schedule of transmission. This schedule looks like a matrix. The width of this matrix is equal to slot frame size. The height of this matrix is equal to the number of available frequencies. A typical TSCH schedule is shown in Fig. 3. Each cell of the schedule matrix is comprised of two components a slot offset and a channel offset. A cell can be allocated to only a single pair of communicating nodes. A node can be allowed to use same frequency channel in two different time-slots, for its communication with two different nodes.

However, a node must be using at most one frequency channel for a given time-slot. Similarly, two communicating nodes may use different frequency channel at different time-slots. The schedule matrix must also ensure that the scheduled communications are conflict free. Such that a node may either transmit or receive for a given time-slot. Similarly, a wireless sensor node can use only one frequency channel for its communication, for a given time-slot. In this manner, a conflict free schedule matrix relies upon temporal and spatial spacing resulting in collision free communication. Energy conservation can be ensured by using adaptive schedules especially for applications that require low power consumption, at the cost of low bandwidth.

TSCH has the benefits of providing collision free communication along with channel hopping enabling multiple pairs of nodes to exchange data during the same slot but at different frequencies. In the schedule matrix, if a cell is allocated for communication between nodes A and B, where A is sending and B is receiving data, this cell is called transmission cell and receiving cell for nodes A and B, respectively. During a transmission cell, a node matches the destination address of the packet with the address of receiving cells in the current time slot. If there is no match found the node keeps the radio off and sleeps for this time slot. If a match is found, the node transmits and start waiting for the acknowledgment from the legitimate recipient. Similarly, in a receive cell, a node listens for incoming packets. It goes to sleep if none to be received in the current time slot. Otherwise, for the received packet an acknowledgement is sent. If a time slot is shared for a particular frequency, the standard implements a back off algorithm for contention resolution [59]. In [60] an adaptive channel selection (ACS) mechanism has been proposed for TSCH networks. It is reported that ACS reduces the number of retransmissions using a channel assessment and channel selection before updating a hopping schedule, hence, conserving the energy.

F. PASAGA

A priority-based algorithm for super frame adjustment has been proposed in [61]. It partitions the guaranteed time slots (GTS) in IEEE 802.15.4 standard to a new length that is nearly half in size of the original length of time slot. This GTS length is calculated based on the size of the high priority packets. Also, BO and SO parameters are adjusted to alter the duty cycle after a predefined time period, which is defined according to the network load in contention period. This leads to an increased throughput, better bandwidth allocation, and energy conservation.

G. BARBEI

A battery aware and reliable beacon enabled technique is proposed in [62] which considers the non-linear process that takes place whenever batteries diffuse a charge. The existing IEEE 802.15.4 standard does not take this into account and assumes linear consumption of power. This approach is unique in the sense that it exploits battery status as well as network latency to reduce energy consumption and delay.

H. ZISENSE

ZiSense [63] is an asynchronous low duty cycling mechanism which is robust to interference. It detects the presence of ZigBee [64] signals and wakes up nodes when the signal is sensed. This is contrary to the method of checking signal strength or probe packets which might be susceptible to interference. However, the detection time should be very small to avoid unnecessary energy consumption. Similarly, there are compute and storage constraints on ZigBee devices. To solve such problems, it is proposed in [63] to use temporal feature vector from the samples of received signal strength indicator (RSSI). This helps to identify ZigBee signals from other interfering signals. Resultantly, ZiSense achieves significant improvement in energy saving by avoiding false wakeups caused by interference.

V. REINFORCEMENT LEARNING BASED MAC PROTOCOLS FOR DUTY CYCLING

Reinforcement learning belongs to a class of machine learning algorithms aiming at goal-directed learning using the concept of actions and corresponding rewards. The learning agent does not know which actions will maximize rewards to start with, but only learns the best set of actions by interacting with the environment. Delayed reward and trial-and-error search are two core features of reinforcement learning [6]. A learning agent must be able to sense its environment and be able to take actions to affect its state. It must also have a goal relating to the state of the environment. The interaction of an agent with the environment is depicted in the Fig. 4.

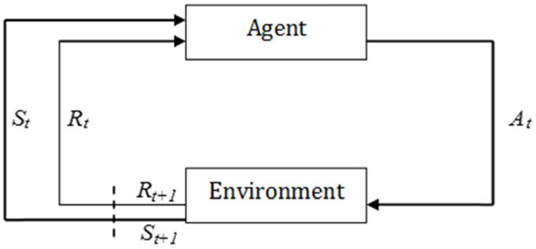


FIGURE 4. Interaction of agent with environment.

Reinforcement learning differs from both supervised learning and unsupervised learning. One of the challenges faced in a reinforcement learning problem is balancing the compensation between exploration and exploitation. The learning agent has to exploit its existing knowledge of the environment based on experiences it has accumulated so far. It also needs to explore the environment in search of better actions, so that the reward could be maximized.

A reinforcement learning system has four main components: (i) policy, (ii) reward signal, (iii) value function and optionally, (iv) model of the environment. A policy defines a mapping from environment states to actions to be taken in those states, and it can be stochastic. Reward is the goal in the reinforcement learning system. It defines the best and worst events for an agent. Rewards can alter policies and actions so that long term goals are achieved. The value function gives the value of a state which is the total reward an agent can expect to gain over the future if it starts from that state. Value of a state is different from a reward in that it gives the long term rewards that a state can offer by taking into account the reward of all subsequent states from that state. Rewards on the other hand are immediate or short term signals that a state can offer. The value of a state can be larger as compared to the reward that it has to offer. Finally, a model of the environment tells about the behavior of the environment in response to actions of learning agents. Thus, for a given action and state the model may predict the next sate and reward.

FINITE Markov Decision Processes: In mathematics, finite Markov decision processes (MDPs) [19] are used to formulate the reinforcement leaning problem. The key elements of the problem's mathematical structure can be defined using MDPs. The agent interacts with the environment at discrete time steps t $= 0, 1, 2, \cdots$. Environment states are represented as S_t at any given time and an agent can take action A_t from this state. As a consequence of the action the agent receives a numerical reward R_{t+1} and transitions to state S_{t+1}. This process continues until maximum reward has been achieved. It should be noted that in a finite MDP, the set of states, S, actions, A, and rewards, R, all have finite number of elements. The goal of all reinforcement learning algorithms is to estimate value functions of states that tell how much reward an agent gets from a given state against a given action. This not only involves current or immediate rewards but also future rewards, more specifically the expected value of future rewards. This in turn depends on what actions an agent will take. As earlier stated, actions are defined in policies, therefore the maximum expected return depends on an optimal policy that the agent pursues. A policy can formally be defined as a mapping from states to probabilities of selecting each possible action. If the agent is following policy π at time s, then $\pi(a|s)$ is the probability that $A_t = a$ if $S_t = s$. Reinforcement learning methods specify how policies are altered after agent gets new experiences from the environment. The value of a state, s, under a policy π is denoted by $v_\pi(s)$, and defined as expected return when agent starts in s and follows actions in policy π afterwards. For MDPs, v_π is formally defined by

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi\left[G_t \,|S_t = s\right] \\ &= \mathbb{E}_\pi\left[\sum\nolimits_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t = s\right], \end{aligned} \tag{4}$$

For all $s \in S$, where $\mathbb{E}_\pi[.]$ represents the expected return given an agent follows policy π at time step t. The function v_π is called the state-value function for policy π. Similarly, the value of an action, a, in state, s, is represented as $q_\pi(s, a)$ and described as,

$$\begin{aligned} q_\pi(s, a) &\doteq \mathbb{E}_\pi\left[G_t \,|S_t = s, A_t = a\right] \\ &= \mathbb{E}_\pi\left[\sum\nolimits_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t = s, A_t = a\right]. \end{aligned} \tag{5}$$

where the function $q_\pi(s, a)$ is called the action-value function for policy π.

Q-learning is a model-free algorithm of reinforcement learning. It does not depend on a state-transition function to predict transition probabilities from one state to another. The optimal policy is estimated using interactions between agent and the environment. This is done using a value function, $Q(S_t, A_t)$, that evaluates an action for all states, produces a 'quality' value and obtains a policy from this value.

Q-learning [6] is a type of temporal difference learning which combines Monte Carlo methods and dynamic programming to solve a Markov decision process. It can learn from raw experiences without a model of the environment and is suited to problems where the complete state space of the environment is not known beforehand. Agents can learn effectively by interacting with the environment and converge to an effective and optimal policy under certain conditions with a probability of one. The Q-values of any <state, action> pair can be represented as,

$$\begin{aligned} Q(S_t, A_t) \leftarrow \alpha \times \left[R_{t+1} + \gamma max_a Q(S_{t+1}, \alpha)\right] + (1 - \alpha) \\ \times Q(S_t, A_t) \end{aligned} \tag{6}$$

where,Q is the new q-value learnt, γ is the learning rate and α is the rate of exploration / exploitation. The Q matrix is initialized to random values and updated for each state and action after each iteration according to (6). The reward matrix R is usually given. When the Q matrix represents maximum reward for all states and actions, the problem has been converged. We have now identified the state-action pairs that can give us the maximum reward. Note that the value of learning parameter α has to be carefully chosen, to keep a balance between exploration and exploitation.

TABLE 2. RL parameters for duty cycling in wpans.

RL Agent	- Constrained Devices e.g., sensor nodes - PAN coordinators - Gateways
Reward Function	- Simple: $R(t) = +1,-1$ - Complex: involves multiple traffic parameters
State Space	- Can be single state/stateless as in Q-learning - Can have multiple states e.g., number of slots in a frame
Actions	- Selecting a time slot as active window - Remaining in sleep mode
Loss Function	- Used in deep Q-learning models - Usually squared error loss between temporal difference target and current Q-value

Reinforcement learning can be used to provide an efficient solution to the duty cycling problem, as will be discussed in the section to follow. However, a few considerations need to be made here.

Effective learning depends on reward function formulation. For duty cycling, the reward function can be as simple as giving a positive value to encourage or a negative value to penalize a certain action, or it can be a complex function of different system variables. Few examples of RL parameters for duty cycling are given in Table 2.

Ideally, an efficient reward function is easy to converge, less complex and computationally less intensive. This is especially important in the context of LR-WPANs because most RL agents are sensor nodes and the complex reward computations may lead to excessive battery consumption. Efficient reward function formulation is an open research issue in reinforcement learning and may vary according to the problem at hand. Reward functions of a few RL approaches used in duty cycling are given in Table 3.

Reinforcement learning is a technique which can be used for effective duty cycling. The following subsections elaborate the use of various reinforcement learning based duty cycling approaches to achieve optimal energy consumption and better network lifetime.

A. USE OF REINFORCEMENT LEARNING APPROACHES

A study of Q-learning approaches for medium access in wireless sensor networks, named ALOHA-Q, is presented in [65]. The implementation of ALOHA-Q is extended to grid, linear chain, and random topologies. Hardware limitations and other practical limitations in ALOHA-Q are studied. The performance of the ALOHA-Q is compared with a Z-MAC [45] and SSA [66] through simulation. It is concluded that ALOHA-Q performs better than SSA and Z-MAC protocols in multi-hop networks. However, a limitation of ALOHA-Q, is to trade-off exploration and exploitation by setting parameters accordingly. To solve this problem, ALOHA-Q-DEPS [65] is proposed. It is similar in working to ALOHA-Q but uses a decreasing ε -greedy policy in which a transmission slot with highest q-value is selected by a node with probability $1-\varepsilon$ and random slot with probability ε. The channel performance of ALOHA-Q-DEPS and ALOHA-Q are evaluated with respect to two practical events: addition of new nodes to the network and packet losses introduced due to sensor node hardware. ALOHA-Q-DEPS is better than ALOHA-Q in terms of robustness. It allows protection of the channel performance in dynamic environments.

B. DECENTRALIZED RL BASED DUTY CYCLING

The work presented in [67] is focused on energy efficient wake-up schedule in wireless networks using a decentralized RL technique. Nodes interact with each other locally and communicate their wake-up cycles with each other. By this, synchronization and de-synchronization of active periods in an independent way is learned by sensor nodes. Synchronization helps in improving message throughput and de-synchronization aims at reducing communication interference. Thus, the self-organizing RL approach helps in reducing the duty cycle of a system. The performance of a self-organizing RL approach is compared with a standard synchronized network. Three topologies are evaluated. It is shown that each node can adapt its duty cycle to network's routing tree of network in an independent manner. As compared to full synchronized technique in which each node wakes up simultaneously, the proposed adaptive behavior enhances the lifetime of the system and increases the throughput for high data rates. No explicit coordination mechanism is needed for converging to synchronization and de-synchronization. Initial randomized wake-up schedules achieve successful convergence. The object is to minimize the amount of active time slots in a frame in a particular contiguous period.

Size of action space of an agent creates a tradeoff between energy conservation and optimality of the solution. Consequently, it is evident that, for duty cycling, the action space is limited to only select an active period in a frame. Thus, the solution can be sub-optimal creating energy burden on the system. Each node holds a q-value for every slot in a frame. This value shows how good it is for a node to be awake during a given slot. It depicts the efficiency of an active time pattern considering the duty cycle and history of communication. The q-value is updated for a node whenever idle listening or a communication event (send/ receive packet) occurs.

Initialization is done using a series of q-values drawn using a uniform random distribution. The number of q-values updated by a node is defined by the number of events occurred during active period. This approach differs from the traditional q-learning approach in which q-value of a selected action is updated only. The time slots which have highest sum of q-values define the wake-up time for nodes. For example, if a user defines the duty cycle to 10%, then each node will be active (or stay awake) for those 10 consecutive time slots in a frame that holds maximum sum of q-values. The agents will stay asleep for the rest of the time slots as it is not beneficial to stay awake. Thus, during transmission of messages to sink

TABLE 3. Reward functions of a few reinforcement learning approaches used in duty cycling.

Scheme / Year	RL agent	Reward function*	State space	Actions	Action space	Q-value initialization
RL-MAC [73] / 2006	Sensor nodes	$R(x, y)$ For a node, x is the no. of successfully transmitted / received packets, y is no. of failed transmission from its neighbors.	Number of packets queued for transmission at the beginning of the frame.	Reserved active time (Turning radio on/off).	Related to the number of packets queued for transmission.	All zero.
DCLA [71] / 2012	Coordinator	$R(x, y)$ x is the queue occupancy, and y is the idle listening time.	Single-state	Proportional to the energy saving of the sensor (sleep schedules of protocol).	Possible energy saving of a sensor (difference of BO and SO, hence 0 <= a <= 14).	Null value/ Maximum value for all actions.
Decentralized Reinforcement Learning [67] / 2012	Sensor nodes	$R(x, y)$ x is the successful transmission and y is successful reception of packet /acknowledgment	State-less because frame and slot sizes do not change.	Selecting active time slot with in a frame (turning radio on/off)	Depends on number of slots in a frame.	All values from uniform random distribution.
SSA [66] / 2013	Sensor nodes	$R(x, y, z)$ x is energy consumption, y is FIFO queue length component and z is a constant value.	S={0,1,2}, 0 for an empty, 1 for decreasing and 2 for increasing queue length.	Choose to send a packet or turn on sleep mode	Discrete values in [0, 1] corresponding to the continuous values of the transmission parameter.	Random value between [0, 1] for all actions.
ALOHA-Q [69] / 2015	Sensor nodes	$R(x)$ = +1,-1 x represents transmission status (+1 if successful, -1 if failed)	N states, each corresponding to a slot in the frame.	Choosing an active slot (turning radio on/off).	Number of slots per frame (N slots corresponding to every node).	All zero
Smart Duty Cycling for M2M Networks [68] / 2015	M2M gateway routers	$R(x, y, z)$ x is the queue length, y is count of packets successfully received in a given time, z is average energy cost of transmission, idle listening and delay.	Possible queue lengths of router which are strictly less than max-queue length.	Choosing an optimal SO value (to calculate duty cycle) for the outgoing frame.	Possible number of packets received and corresponding SO values.	All zero
CR-MAC [74] / 2018	Primary User devices	$R(x, y)$ = {+i, 0}, i is the no. of successful transmissions. x represents successful transmissions, y is channel occupancy	N states, each corresponding to a slot in the frame.	Choosing an optimal channel to transmit packets.	Number of slots in a frame (N slots each corresponding to a node).	Channel occupancy + successful transmissions in initial slot.

* $R(.)$, where R is the reward function at time t, computed based on its parameters. It is specific to each RL scheme.

node, sufficient knowledge regarding slot quality is acquired by nodes to assess best period to stay active. This results in neighboring nodes to desynchronize their actions. By this, fast message delivery and low latency is achieved.

Active time slots are updated in individual manner irrespective of node's wake-up schedule due to three reasons. First, this approach results in exploration of each time slot in a frequent way. Second, altering q-values individually makes it possible to dynamically update the duty cycle of nodes. Third, the proposed approach not only updates wake-up slots but every slot in an active period. Also, updating individual time slots does not require an explicit exploration scheme. The greedy strategy ensures that each slot is explored and updated at application initiation. This continues until the sum of q-values of some group of slots gets larger as compared to the rest, in which case the policy converges and further exploration is stopped. Reward based greedy policy will not get stuck in local optima as alteration of slots is done regularly. When a node fails and messages are not delivered, the goodness of wake-up slot will start to decrease till nodes relearn and re-wake-up using a different scheme. Duty cycle schedule fixed by the user has an impact on the convergence speed and re-learning. Learning rate is usually in the range of 0.1 to 0.2. In non-stationary environment, it is desirable to have a constant learning rate to make sure that the policies change with regard to the rewards received most recently.

C. SMART DUTY CYCLE CONTROL

A duty cycle control scheme based on reinforcement learning that ensures reliable and efficient M2M communication is proposed in [68]. The scheme is reinforcement learning based duty cycle control for IEEE 802.15.4. A multi-hop based M2M communication network is modeled that considers various dynamics of network. A distributed and optimal duty cycle control scheme is derived mathematically to ensure reliability of transmission and optimizing end-to-end delays and energy. This scheme helps in learning optimal policy without having initial information about network. It helps in smart control of duty cycle with respect to various dynamics of network.

D. ALOHA-Q

A scheme reported in [69] applies Q-learning for intelligent slot selection mechanism to frame based ALOHA. It starts with random access and moving towards complete slot scheduling. It presents reinforcement learning algorithm called Q-learning for an adaptation of slotted ALOHA with reduced collisions and retransmissions in single-hop networks. Experimental results show that perfect scheduling in steady states can be achieved. The only overhead incurred is acknowledgement packet. It uses a Markov model to analyze convergence and to validate the simulation model. The performance of a steady state is evaluated against S-MAC and Z-MAC through simulation.

E. DYNAMIC ADAPTATION OF DUTY CYCLING USING MAC PARAMETERS

Mihaylov, M. et al. investigated the previous schemes of IEEE 802.15.4 duty cycle management, by using MAC layer parameters like SO and BO, and highlighted their limitations [66]. It is reported that the use of buffer occupancy and super frame occupation ratio as parameters can lead to less than optimal BO and SO settings. Therefore, a dynamic and adaptive duty-cycling scheme for cluster tree networks using MAC parameters is suggested. This scheme determines channel traffic while incurring low control overhead. A Markov model to approximate power consumption and delay in transmitting frames is also suggested in [70].

The algorithm approximates channel conditions by taking three parameters as input which are *macMinBE* (BEB), *macMaxFrameRetries* (NR), and *macMaxCSMABackoffs* (NB). The BO and SO parameters are then adjusted for subsequent super frames. The default values of BEB, NB, and NR are 3, 4, and 3 respectively, as specified in the standard IEEE 802.15.4. The algorithm can be started after a set number of super frames have been received or after every super frame cycle.

The PAN coordinator initiates the algorithm. The coordinator fetches the values of the three parameters from received super frames and calculates the average. At termination of the active period, the coordinator determines the average of BEB, NB and NR, and computes SO and BO, as given in Table 4. Under low channel contention, SO and BO parameters are set in a way that the nodes sleep for longer periods of time. If this is not done; it will lead to inefficient duty cycles due to frequent beacon transmissions. In case of a high retransmission rate of frames in the network only SO is decremented by 1. This is because back offs occur due to high contention in the network and packets are being retransmitted due to collisions. Collisions could be either a result of synchronization issues between nodes, or hidden terminal problem. This directs to having a shorter active period, without altering the frequency of beacon transmissions. However, if the number of back offs increase, it shows more frames have sensed a busy channel and therefore transmission in contention access period of the coordinator cannot happen. SO is incremented leading to a longer active window for transmission of frames. This lowers the contention. New SO and BO parameters are communicated to connected devices which now follow the new duty cycle. Experimental results using the proposed approach show that this duty cycling approach leads to the best use of the PAN coordinator's active period and good energy conservation. This ensures a longer network lifetime as compared to the IEEE 802.15.4 standard.

TABLE 4. Use of MAC parameters for dynamic adaptation of duty cycling [70].

Values of BEB, NB, and NR are retrieved from received frames	
Condition	Action
NB ≤ (macMaxCSMABackoffs / 2)	Decrease value of SO by 1.
NB > (macMaxCSMABackoffs / 2)	Increase value of SO by 1.
NB ≤ (macMaxCSMABackoffs / 2) AND NR ≤ (macMaxFrameRetrieves / 2) AND BEB ≤ (macMaxBE / 2)	Increase value of BO by 1.

F. DCLA

To estimate incoming traffic, DCLA [71] captures network statistics during active period. To learn best duty cycle, reinforcement learning based framework is used at every beacon interval. This approach eradicates the need to manually reconfigure the duty cycle for nodes for specific network deployment. To achieve this, traffic estimation performed by FFD devices in setting duty cycle is reviewed and q-learning based method is used to design a DCLA based algorithm, policy functions, and reward which is employed by the agent. The DCLA based algorithm is designed as a software component that dynamically configures duty cycle at run time without human involvement. This aims at providing energy efficiency. The algorithm is designed such that a PAN coordinator executes it to find the optimal duty cycle schedule without any initial knowledge about network. This is achieved by q-learning in which the RL agent takes only one state and takes one of the many actions. Exploration and selection are done in an optimal way. For static environment, best actions in the q-matrix are exploited as the training process starts progressing. For dynamic environment, new actions are explored on the basis of rewards received. Thus, to ensure fast convergence in static environments and adaption to dynamic environment, DCLA manages the exploration to exploitation ratio with biasness of a policy using knowledge obtained from rewards.

A number of parameters that are used to measure the overall network performance are also mentioned in [71]. Examples of such parameters include, network offered load (NOL), which refers to the amount of data a wireless sensor node generates at a specific time and is measured in bps (bits per second). The energy efficiency (E_{eff}) is the total amount of information sent / received per Joule of utilized energy by the sensor node. End-to-end delay (D) is the latency measured in seconds, experienced by a datagram from its generation to reception. Probability of success (PS) refers to the number of times a packet is successfully transmitted in the network.

Comparison of DCLA with different duty cycle approaches for IEEE 802.15.4 is also reported with respect to event based and periodic scenarios. This technique is more effective and ensures a long network lifetime as compared to contemporary duty cycling techniques [71].

G. SSA

To lower the consumption of energy and achieve low latency in wireless sensor network a distributed and self-learning scheduling approach (SSA) is presented in [66]. It extends the q-learning method to make nodes learn sleep and continuous transmission parameters. SSA incrementally learns an optimal scheduling policy without initial knowledge of network. The learning process of all nodes is periodic and simultaneous. Transmission parameter is taken as an action for a node. Sleep schedule parameter is found using parameter for transmission scheduling. There are two stages in SSA: scheduling update (SU) and data delivery (DD). In DD, all nodes perform their tasks according to sleep and packet transmission scheduling. In SU, nodes update their q-values after they receive a reward in DD stage. Nodes then use ε-greedy scheme for selection of next action. This is done iteratively until the q-values of the states reach the expected optimal value. Transmission of packets takes place in DD stage. The nodes sleep or wake-up according to the scheduling parameters. Reward is calculated in SU stage and q-values are updated. All nodes interact simultaneously with a wireless sensor network to learn the optimal scheduling parameters. Learning these parameters is vital as it will have an impact on behavior of a node and wireless network as a whole. The aim is to achieve a long-term optimal sum of rewards by a series of exploration and exploitation.

In order to verify the correctness, SSA is implemented in MAC layer. ACK frame helps in acknowledgment of data received by a recipient. In case of packet transmission failure, sender has to re-transmit the data packet. Under various work load, simulation study reveals that performance of SSA is better as compared to S-MAC [53] and DW-MAC [72], in terms of throughput, latency, energy consumption, and maximum length of queue. In scenarios of heavy workload and serious collisions, SSA performs better than the other two methods.

H. RL-MAC

An adaptive RL based MAC protocol for WSNs is presented in [73]. Other existing approaches focus on active and sleep period scheduling of nodes to minimize consumption of energy. Recent protocols that use adaptive duty cycles are employed to optimize energy consumption. However, in many cases, duty cycle is determined on the basis of traffic load of node itself. In adaptive RL-based MAC protocol [73], nodes make inference about their state using a reinforcement learning approach. This results in low power utilization and high throughput for a series of traffic conditions. This approach is good for practical deployments as it has moderate computational complexity. It is simple, self-organized, and distributed in nature. RL based MAC protocol uses a frame-based structure similar to T-MAC and S-MAC. Active time and duty cycle of the frame are dynamically updated with respect to node's traffic load characteristics. The length of a slot is a function of packet length and bandwidth. Initially at the start of the frame, RL agent acts as a MAC protocol engine. It reserves the active time slots dynamically. Node exchanges packets with its peers and listen to the channel in its active time. After expiration of active time, the node does not send or receive any data packet and goes to the sleep state. Two goals of RL agent are to maximize the energy efficiency (as ratio of transmission/receive time and total reserved time for active period) and to maximize the throughput of data.

I. CR-MAC

Carie *et al.* [74] present a cognitive radio-based approach for medium access in such a way that the nodes learn about the channel by using Q-learning. Cognitive radio-based MAC employs channel switching to improve achievable throughput. Existing CR-MAC protocol selects channels randomly which leads to packet collisions and latency in the network. Carie *et al.* [74] presents a channel selection scheme based on Q-learning. In this scheme, time is divided into fixed intervals of size t and the number of successfully transmitted packets (ND) are recorded. At the beginning of each interval the node updates its q-value and broadcasts this value to its neighbors along with the primary user free channel list (PCL). The neighbors update their PCL table after receiving this broadcast message. Also, the occupation state of each channel is calculated, which shows the number of users that have selected each channel. Nodes prefer channels with low occupation rate. Q-values (0, 1) of each available channel are maintained in a table. A node takes a greedy approach to select a channel with the highest q-value. For every successful transmission, reward R of the channel is increased by the number of packets transmitted on it, and zero otherwise. A variable discount factor is used which is dependent on the number of competing nodes and the bandwidth. Channel selection is based on the activity of other nodes. Channel selection and channel re-use are exploited in order to optimize power consumption.

The QoS-Aware MAC is a distributed MAC and Q-learning based duty cycling technique proposed for QoS differentiation [75]. The QL-MAC presents RL-based radio scheduling schema [76]. In this decentralized on-line approach, each node determines the best suited radio schedule by dynamic

TABLE 5. Reinforcement learning for MAC protocols in WSNs.

Protocol	RL Technique Used	Features	Reward Function Parameters	Tested Topologies	Performance Metrics	Compatibility with IEEE 802.15.4
RL-MAC [73]	Q-Learning with ε-greedy policy	Adaptive MAC protocol based on Q-learning where nodes evaluate the state of other nodes to improve their MAC schedule	Receive/transmit time ratio, successful transmissions, sleep time, reserved active time	Linear, Star, Mesh	E_{eff}, D, NT	Not compatible
DCLA [71]	Q-Learning with ε-greedy, round robin and random policy	Duty cycle learning algorithm that runs on coordinator devices. Duty cycle is learnt using incoming traffic estimates during active periods and Q-learning at beacon intervals	Queue occupation threshold	Single cluster star topology	NOL, E_{eff}, D, PS*	Compatible
Decentralized Reinforcement Learning [67]	Q-Learning with ε-greedy policy	Decentralized learning algorithm used to schedule wakeup cycle based on interactions of nodes only with neighbors, desynchronizing with other nodes	Successful transmission (send, receive)	Linear, Single-hop Mesh, Grid	D	Not compatible
SSA [66]	Q-Learning with ε-greedy policy	Distributed self-learning approach that uses Q-learning to enable each node learn its transmission and sleep parameters	Energy consumption, queue length for latency	Linear, Tree	EC*, Q_{len}**, NT, D	Not compatible
Learning-based MAC [76]	Q-Learning	A Q-learning based MAC protocol for optimal energy consumption in WSN, where q-value of each slot is updated based on no. of successful transmissions	Successful transmissions (send, receive), overheard packets, expected packets	Linear, Star, Mesh	D, E_{res}, NT	Not compatible
ALOHA-Q [69]	Q-Learning	Q-learning used with frame-based ALOHA, converges to scheduled access from random access	Number of successful transmissions (send, receive)	Single-hop network of 20 nodes	D, NT, E_{eff}, T_{conv} **	Compatible
Smart Duty Cycling for M2M Networks [68]	Q-Learning with ε-greedy policy	Distributed duty cycle learning framework developed for 802.15.4 gateways and routers to find optimal policy without prior network traffic information.	Transmit, receive and idle-listening energy cost, delay cost	Two-hop cluster tree network	E_{eff}, D, PDR**	Compatible
Dynamic Adaptation of Duty Cycling using MAC Parameters [70]	Markov Model	Dynamic duty cycle adaptation approach using MAC parameters such as binary exponential back-off (BEB), no. of back-offs and no. of retransmissions to estimate channel condition. Uses a Markov model to calculate delay and energy consumption	MAC parameters	Cluster tree topology	D, E_{res}, NT**	Compatible
CR-MAC [74]	Q-Learning	A channel selection scheme based on Q-learning, where each channel maintains a q-value according to the number of successful transmissions	Successful transmissions (send, receive)	Grid	EC, NT, D	Not compatible

* NOL: Network Offered Load, E_{eff}: Energy Efficiency, D: End-to-End Delay, PS: Probability of Success, EC: Energy Consumption
** Eres – Residual Energy, NT – Network Throughput, PDR – Packet Drop Ratio, Tconv – Convergence Time, Qlen – Max. Queue Length

adaptation according to the communication activities and its own traffic load.

VI. DISCUSSION AND PERFORMANCE COMPARISON

The use of reinforcement learning based approaches present a plausible solution to adaptive duty cycling, leading to a better energy conservation in IEEE 802.15.4 low-power networks. It mainly relies on determining an optimal Q-matrix which captures the state transitions leading to a maximized reward.

In the Table 5 some RL based duty cycling solutions are presented which we have reviewed in this paper. It should be noted that all approaches discussed here need to maintain information about traffic load of sensors and occasionally their neighbors also. This leads to a couple of considerations when adopting RL based techniques for efficient duty cycling. First, defining a network-representative reward function can be challenging because the requirements of delay, throughput, battery, and network lifetime, vary from application to application. Second, the convergence time of the algorithm can induce unnecessary delays especially in applications where contention is low and data is sent after long intervals of time. Conversely, under high traffic loads

TABLE 6. RL based MAC protocols for duty cycling scheme in first column outperforms schemes in second column.

Protocol	Compared With	Experimental Setup	Topology Used
RL-MAC [73]	S-MAC [53] T-MAC [54]	Simulation using NS-2	Star, Linear, Mesh.
Decentralized RL [67]	Synchronicity De-synchronicity	Simulation using OMNET++	Linear, Mesh, Grid.
QL-MAC [76]	RL-MAC [73] S-MAC [53]	Simulation using OMNET++	Linear, Star, Mesh.
ALOHA-Q [65]	Z-MAC [45], SSA [66]	Simulation and experiments on MicaZ [82] and IRIS [83] nodes using TinyOS.	Grid, Linear, Random.
CR-MAC [74]	IEEE 802.11 DCF, Inband-CRMAC, Hybrid-CRMAC (Omni), Hybrid-CRMAC (Directional), Out-of-band-CRMAC	Simulation using NS-2.31	Grid

and high contention levels the time needed by RL based algorithms to reach the optimal duty cycle may cause extra latency. Network nodes are mostly constraint by power (computation) which limits the complexity of potential Q-learning algorithm and reward functions.

It should also be noted that Q-learning algorithms require time to converge and it might be challenging in case of delay sensitive applications and in the situations where network lifetime is less than the convergence time of the learning algorithm. Adjusting duty cycle for energy saving leads to longer end-to-end message latency. The active period is also reduced and leads to achieve contention at the start of the super frame. In some approaches, extra control overhead is incurred at nodes and the coordinator. It is difficult to compare the performance of all the techniques from the literature because most of them are tested in different simulation environments for certain topologies with particular network parameters. Due to this reason, rather than an exhaustive comparison, the available choice is to compare techniques which share the testing environment. Table 6 presents such a comparison, where, the techniques listed in first column outperform the techniques listed in second column for each row separately.

Results of RL-MAC [73] are quoted in comparison with S-MAC [53] in simulated environments configured using star, linear, and mesh topologies on NS-2 platform. For all three topologies, throughput of RL-MAC is more than S-MAC. In linear topology RL-MAC consumes 30.93% less energy as compared to S-MAC. In mesh topology, performance of RL-MAC is better than not only S-MAC but also T-MAC in both data throughput and energy efficiency, as shown in Table 7.

Decentralized reinforcement learning scheme [67] was evaluated on OMNET++ simulator by using linear, mesh, and grid topologies. The results are given for average end-to-end latency at different values of duty cycles by considering both synchronicity and de-synchronicity. It was shown that the de-synchronized approach could reduce duty cycle by around 10% without increasing the latency.

TABLE 7. Performance comparison of RL-based MAC protocols.

Protocol	Throughput (byte/sec)			Energy Efficiency (mJoule/byte)		
	Linear	Mesh	Star	Linear	Mesh	Star
QL-MAC	200*	850	200	1.5	1.5	1.8
RL-MAC	70	710	152	13	5	2.1
S-MAC	20	290	35	24	9	8
T-MAC	-	230	-	-	6.15	-

Performance comparison of QL-MAC [76] with RL-MAC [73] and S-MAC [53] carried out on OMNET++ simulator by using linear, mesh, and star topologies. QL-MAC demonstrated maximum throughput and minimum power consumption in all topologies, as shown in Table 7.

It is shown in [65] that ALOHA-Q outperforms SSA [66] and Z-MAC [45] in both simulated and real setups. In simulation for linear, grid, and random topologies, ALOHA-Q showed more channel throughput as compared to SSA and Z-MAC. The, throughput of SSA was more than Z-MAC but less than ALOHA-Q. Real experiments show a similar pattern.

CR-MAC [74] was evaluated on NS-2.31 simulation platform using grid topology. The results were compared with IEEE 802.11 DCF, Inband-CRMAC, Hybrid-CRMAC (Omni), Hybrid-CRMAC (Directional), and out-of-band-CRMAC. Proposed RL based CR-MAC outperforms others in terms of throughput, energy consumption, and elapsed time.

VII. FUTURE RESEARCH DIRECTIONS

In various studies, experimental results have highlighted that various Q learning-based duty cycling techniques are capable of achieving higher throughput (byte/second) in an energy efficient manner. Thus, RL-based techniques are considered as plausible solution for energy conservation at MAC layer. For further successful development of such techniques we need to acquire better visibility in many aspects such as estimation of convergence time, formulation of efficient reward functions, and performance evaluation on real network test-bed using standardized network configuration, and impact of cross layer design, as well.

A. CONVERGENCE TIME

The time elapsed until every node has learnt an optimal duty cycle is called convergence time after which nodes are in steady state. These RL-based algorithms may take long time before convergence depending upon several factors such as state-action space, reward function, network topology, and traffic load, a few to name.

A primary challenge is to estimate the size of the state-action space which determines how long the algorithm takes to reach a steady state. Q learning is a model-free algorithm that does not require state-action transition probabilities and prior knowledge of the environment. Mostly, state-space consists of number of slots in a frame. However, in a network determining an optimal duty cycle can be challenging. A study on determining optimal number of slots per frame can be conducted for different network configurations.

Moreover, it is also important to note that network parameters, whether set statically or learnt adaptively, should be in line with the global view of the network. It is crucial that all optimal parameter settings should account for topology changes, mobility of the nodes, and network lifetime in IoT networks. Further, most techniques [66]–[71] studied so far assume a static model of the environment, where the network topology does not change. Still, it has to be asked that how RL algorithm will perform in a dynamically changing network topology. Furthermore, what factors influence convergence time of RL-based duty cycling algorithm and what is their significance?

Notably, for the proposed algorithms, only empirical results have been presented without any support of analysis for asymptotic running-time complexities, to attain an optimal state of duty cycle. Also, on account of possible variation in the state-action space and reward function of a duty cycling algorithm, it is hard to come up with a generalized convergence theorem. A careful theoretical study on the expected number of learning iterations to ascertain convergence time can be conducted using randomized or probabilistic approaches. Further, the cost-benefit analysis should be done for employing duty cycling techniques.

B. EFFICIENT REWARD FUNCTION

Every RL algorithm employs a reward function to discourage bad actions and to encourage good actions. However, in pursuit of an optimal duty cycle, formalizing an efficient reward function is the most daunting task. Further, efficient reward function should be capable of depicting network dynamics by considering network parameters. The Table 3 summarizes reward functions that are comprised of several parameters, for different duty cycling algorithms in the literature.

It is trivial to understand that generalization across short range wireless technologies for reward functions is not easy because they are specific to an environment. Therefore, research endeavors should be made to design an efficient reward function for short range wireless technologies such as IEEE 802.15.4, IEEE 802.15.1, and IEEE 802.11. For example, in IEEE 802.15.4 it would entail finding optimal values of BO and SO parameters.

It should be noted, although complex reward function may capture many network parameters for example incoming/outgoing data of a node, state of neighboring node(s), and changing network conditions, yet, complex reward function may lead to large convergence time. Here, an important question arises, how to tradeoff complexity of reward function with convergence time, while achieving high throughput and energy efficiency.

C. PERFORMANCE EVALUATION

Finally, most of the work summarized in Table 5 has been evaluated using simulation software and it is important to investigate it on a real test bed.

So far, RL-based duty cycling techniques have been evaluated for energy efficiency, network throughput, and transmission latency. But, these evaluations are made in a simulated environment under varying/non-standardized network configurations leading to obscure comparison of RL algorithms. The following two possible research directions are proposed.

(1). A proposed RL algorithm should be evaluated on real network test-bed, under varying network parameters and traffic intensity inviting a wide community to evaluate and improve. In that case, comparison of relative performance would pave the way for the integration of RL-based techniques into the current protocol stack.

(2). To make an effective performance comparison, it is required that all proposals should be evaluated under a standardized environment, offered by a benchmarking framework. We suggest that a benchmarking framework should be designed with a set of network configuration and parameters. This will provide a common ground for evaluation and comparison of current and coming techniques in future resulting in a data-driven and time-efficient decision making.

D. CROSS-LAYER DESIGN

The approach of cross layer design can be applied in the following ways.

(1). To study the adaptive duty cycling in conjunction with RPL [2] which is designed as network layer protocol for IoT protocol stack. The potential effects of duty cycling on construction and maintenance of RPL destination oriented directed acyclic graph (DODAG) needs to be investigated, in case a node fails or topology changes.

(2). It is clear, that RL engine should not be hosted at constrained nodes of IoT. Computationally extensive tasks, such as learning the optimal duty cycle, should be offloaded to unconstrained nodes of the network for instance IoT gateway. However, such a configuration may require adequate trust management [77] mechanisms between gateway and sensor nodes. Further, in this case, dissemination of duty cycle schedule information to each node would be a cumbersome task and may induce additional transmission delay and unwarranted use of radio resources.

(3). We suggest using fog computing paradigm as an answer to this issue. Duty cycle learning engine can be offloaded to a resourceful node [78] for instance PAN coordinator in IEEE 802.15.4 network. Such resourceful nodes can act as command and control nodes, in line with the ideology of software defined network (SDN), which learn through reward function and disseminate optimal duty cycle to lean nodes, while maintaining global perspective of the network.

VIII. CONCLUSION

In this paper, medium access control (MAC) protocols have been summarized. In the view of energy conservation, at MAC layer the duty cycling has been extensively studied in the last decade. Further, to find the optimal duty cycle schedule the use of reinforcement learning (RL) has been a popular technique in the literature. This papers attempts to make a performance comparison of RL-based duty cycling techniques. Also, this survey paper highlights the associated challenges that hinder a fair and meaningful comparative analysis of RL-based proposals. However, it is highlighted that lean nodes of IoT are incapable of running complex learning algorithms while paying high cost of energy. In the light of this survey, we emphasized the future research directions that would help in overcoming the hindrances in adoption of RL-based energy conserving duty cycling, in the context of Internet of Things (IoT). The recommendations have been made in four broader areas for: a) estimation of convergence time, b) formulation of efficient reward functions, c) performance evaluation on real network test-bed, and d) cross-layer design.

REFERENCES

[1] A. Serbanati, A. S. Segura, A. Oliverau, Y. B. Saied, N. Gruschka, D. Gessner, and F. Gomez-Marmol, "Internet of Things architecture, concept and solutions for privacy and security in the resolution infrastructure," Eur. Commission, Berlin, Germany, Tech. Rep. D4.2, EU Project IoT-A, Project Report, 2012.

[2] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, 2012.

[3] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, document RFC 7252, 2014.

[4] N. Kushalnagar, G. Montenegro, and C. Schumacher, *IPv6 Over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, document RFC 4919, 2007.

[5] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4, Dec. 2015. [Online]. Available: https://standards.ieee.org/standard/802_15_4-2015.html

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[7] K.-I. Hwang and S.-H. Yoon, "Taxonomy and evaluations of low-power listening protocols for machine-to-machine networks," *J. Appl. Math.*, vol. 2014, pp. 1–12, Jul. 2014.

[8] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 222–248, 2nd Quart., 2010.

[9] N. Abramson, "The ALOHA system: Another alternative for computer communications," in *Proc. Fall Joint Comput. Conf.*, Houston, TX, USA, Nov. 1970, pp. 281–285.

[10] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1400–1416, Dec. 1975.

[11] H. Takagi and L. Kleinrock, "Throughput analysis for persistent CSMA systems," *IEEE Trans. Commun.*, vol. 33, no. 7, pp. 627–638, Jul. 1985.

[12] P. Karn, "MACA—A new channel access method for packet radio," in *Proc. ARRL/CRRL Amateur Radio 9th Comput. Netw. Conf.*, Sep. 1990, pp. 134–140.

[13] Y. C. Tay, K. Jamieson, and H. Balakrishnan, "Collision-minimizing CSMA and its applications to wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1048–1057, Aug. 2004.

[14] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2001, pp. 221–235.

[15] S. Singh and C. S. Raghavendra, "PAMAS–power aware multi-access protocol with signalling for ad hoc networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 5–26, Jul. 1998.

[16] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ANSI/IEEE Standard 802.11, 1999.

[17] E.-Y.-A. Lin, J. M. Rabaey, and A. Wolisz, "Power-efficient Rendez-Vous schemes for dense wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 7, Jun. 2004, pp. 3769–3776.

[18] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2006, pp. 321–334.

[19] K. S. J. Pister and L. Doherty, "TSMP: Time synchronized mesh protocol," in *Proc. IASTED Int. Symp. Distrib. Sensor Netw.*, Orlando, FL, USA, Nov. 2008, pp. 391–398.

[20] K. Arisha, M. Youssef, and M. Younis, "Energy-aware TDMA-based MAC for sensor networks," in *System-Level Power Optimization for Wireless Multimedia Communication*, R. Karri and D. Goodman, Eds. Boston, MA, USA: Springer, 2002, pp. 21–40.

[21] S. C. Ergen and P. Varaiya, "PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 7, pp. 920–930, Jul. 2006.

[22] M. Ringwald and K. Römer, "BitMAC: A deterministic, collision-free, and robust MAC protocol for sensor networks," in *Proc. 2nd Eur. Workshop Wireless Sensor Netw.*, Istanbul, Turkey, Feb. 2005, pp. 57–69.

[23] M. I. Brownfield, K. Mehrjoo, A. S. Fayez, and N. J. Davis, "Wireless sensor network energy-adaptive MAC protocol," in *Proc. CCNC. 3rd IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 2006, pp. 778–782.

[24] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 16–27, Oct. 2000.

[25] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Netw.*, vol. 12, no. 1, pp. 63–78, 2006.

[26] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka, "Energy-efficient, application-aware medium access for sensor networks," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst. Conf.*, Washington, DC, USA, Nov. 2005, pp. 630–638.

[27] A. Barroso, U. Roedig, and C. Sreenan, "μ-MAC: An energy-efficient medium access control for wireless sensor networks," in *Proc. 2nd Eur. Workshop Wireless Sensor Netw.*, Istanbul, Turkey, Feb. 2005, pp. 70–80.

[28] L. F. W. van Hoesel, T. Nieberg, H. J. Kip, and P. J. M. Havinga, "Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs," in *Proc. IEEE 59th Veh. Technol. Conf. (VTC-Spring)*, Milan, Italy, vol. 3, May 2004, pp. 1598–1602.

[29] P. Cheong and I. Oppermann, "An energy-efficient positioning-enabled MAC protocol (PMAC) for UWB sensor networks," in *Proc. 14th IST Mobile Wireless Commun.*, Dresden, Germany, Jun. 2005, pp. 95–107.

[30] G. Pei and C. Chien, "Low power TDMA in large wireless sensor networks," in *Proc. MILCOM. Commun. Netw.-Centric Oper., Creating Inf. Force*, McLean, VA, USA, vol. 1, Oct. 2001, pp. 347–351.

[31] J. Li and G. Y. Lazarou, "A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Berkeley, CA, USA, Apr. 2004, pp. 55–60.

[32] M. Ali, T. Suleman, and Z. A. Uzmi, "MMAC: A mobility-adaptive, collision-free MAC protocol for wireless sensor networks," in *Proc. PCCC. 24th IEEE Int. Perform., Comput., Commun. Conf.*, Phoenix, AZ, USA, Apr. 2005, pp. 401–407.

[33] W. Lee, A. Datta, and R. Cardell-Oliver, "FlexiMAC: A flexible TDMA-based MAC protocol for fault-tolerant and energy-efficient wireless sensor networks," in *Proc. 14th IEEE Int. Conf. Netw.*, Singapore, Sep. 2006, pp. 1–6.

[34] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. 19th IEEE Int. Parallel Distrib. Process. Symp.*, Denver, CO, USA, Apr. 2005, pp. 1–8.

[35] H. Cao, K. W. Parker, and A. Arora, "O-MAC: A receiver centric power management protocol," in *Proc. IEEE Int. Conf. Netw. Protocols*, Santa Barbara, CA, USA, Nov. 2006, pp. 311–320.

[36] L. F. W. van Hoesel and P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," in *Proc. 1st Int. Workshop Netw. Sens. Syst. (INSS)*, Tokyo, Japan, 2004, pp. 205–208.

[37] S. Datta, "RMAC: A randomized adaptive medium access control algorithm for sensor networks," in *Proc. 2nd Workshop Sensor Actor Netw. Protocols Appl.*, Boston, MA, USA, Aug. 2004, pp. 1–11.

[38] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, New York, NY, USA, Apr. 2002, pp. 3418–3423.

[39] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov./Dec. 2002.

[40] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Baltimore, MD, USA, Nov. 2004, pp. 95–107.

[41] R. Jurdak, P. Baldi, and C. V. Lopes, "Energy-aware adaptive low power listening for sensor networks," in *Proc. Int. Workshop Netw. Sens. Syst. (INSS)*, San Diego, CA, USA, Jun. 2005, pp. 24–29.

[42] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Boulder, CO, USA, Oct. 2006, pp. 307–320.

[43] S. Liu, K. W. Fan, and P. Sinha, "CMAC: An energy-efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 4, pp. 1–34, 2009.

[44] A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux, "Poster abstract: WiseMAC, an ultra low power MAC protocol for the wiseNET wireless sensor network," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Los Angeles, CA, USA, Nov. 2003, pp. 302–303.

[45] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[46] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. SenSys*, Boulder, CO, USA, Oct. 2006, pp. 293–306.

[47] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Societies*, New York, NY, USA, Jun. 2002, pp. 1567–1576.

[48] G. P. Halkes and K. G. Langendoen, "Crankshaft: An energy-efficient MAC-protocol for dense wireless sensor networks," in *Proc. Eur. Conf. Wireless Sensor Netw.*, Berlin, Germany, Jan. 2007, pp. 228–244.

[49] Y. Liu, A. Liu, N. Zhang, X. Liu, M. Ma, and Y. Hu, "DDC: Dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 131, pp. 16–27, Apr. 2019.

[50] X. Xiang, W. Liu, A. Liu, N. N. Xiong, Z. Zeng, and Z. Cai, "Adaptive duty cycle control-based opportunistic routing scheme to reduce delay in cyber physical systems," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, pp. 1–21, Apr. 2019.

[51] L. Guntupalli, D. Ghose, F. Y. Li, and M. Gidlund, "Energy efficient consecutive packet transmissions in receiver-initiated wake-up radio enabled WSNs," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4733–4745, Jun. 2018.

[52] M. Peng, W. Liu, T. Wang, and Z. Zeng, "Relay selection joint consecutive packet routing scheme to improve performance for wake-up radio-enabled WSNs," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–32, Jan. 2020.

[53] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[54] T. V. Dam and K. Langendoen, "An adaptive energy efficient MAC protocol for wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, Los Angeles, CA, USA, Nov. 2003, pp. 171–180.

[55] *OMNET++, Discrete Event Simulator*. Accessed: Jun. 14, 2019. [Online]. Available: https://omnetpp.org/

[56] P. Havinga, S. Etalle, H. Karl, C. Petrioli, M. Zorzi, H. Kip, and T. Lentsch, "Eyes-energy efficient sensor networks," in *Proc. IFIP Int. Conf. Pers. Wireless Commun.*, Berlin, Germany, Sep. 2003, pp. 198–201.

[57] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1384–1396, Oct. 2009.

[58] J. Jeon, J. W. Lee, J. Y. Ha, and W. H. Kwon, "DCA: Duty-cycle adaptation algorithm for IEEE 802.15.4 beacon-enabled networks," in *Proc. IEEE 65th Veh. Technol. Conf. (VTC-Spring)*, Dublin, Ireland, Apr. 2007, pp. 110–113.

[59] T. Watteyne, M. Palattella, and L. Grieco, *Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem statement*, document RFC 7554, 2015.

[60] A. Elsts, X. Fafoutis, R. Piechocki, and I. Craddock, "Adaptive channel selection in IEEE 802.15.4 TSCH networks," in *Proc. Global Internet Things Summit (GIoTS)*, Geneva, Switzerland, Jun. 2017, pp. 1–6.

[61] B.-H. Lee, H.-K. Wu, and N.-C. Yu, "A priority based algorithm for adaptive superframe adjustment and GTS allocation (PASAGA) in IEEE 802.15.4 LR-WAN," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Tokyo, Japan, Apr. 2018, pp. 318–320.

[62] M. Salayma, A. Al-Dubai, I. Romdhani, and M. B. Yassein, "BARBEI: A new adaptive battery aware and reliable beacon enabled technique for IEEE 802.15. 4 MAC," in *Advances in Network Systems*. Cham, Switzerland: Springer, 2017, pp. 317–335.

[63] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "Interference resilient duty cycling for sensor networks under co-existing environments," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2971–2984, Jul. 2017.

[64] *ZigBee Specifications, ZigBee Standard*. Accessed: Jun. 14, 2019. [Online]. Available: http://www.zigbee.org/wp-content/uploads/2014/11/docs-05-3474-20-0csg-zigbee-specification.pdf

[65] S. Kosunalp, Y. Chu, P. D. Mitchell, D. Grace, and T. Clarke, "Use of Q-learning approaches for practical medium access control in wireless sensor networks," *Eng. Appl. Artif. Intell.*, vol. 55, pp. 146–154, Oct. 2016.

[66] J. Niu and Z. Deng, "Distributed self-learning scheduling approach for wireless sensor network," *Ad Hoc Netw.*, vol. 11, no. 4, pp. 1276–1286, Jun. 2013.

[67] M. Mihaylov, Y. A. Le Borgne, K. Tuyls, and A. Nowé, "Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 9, no. 3, pp. 207–224, 2012.

[68] Y. Li, K. K. Chai, Y. Chen, and J. Loo, "Smart duty cycle control with reinforcement learning for machine to machine communications," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, London, U.K., Jun. 2015, pp. 1458–1463.

[69] Y. Chu, S. Kosunalp, P. D. Mitchell, D. Grace, and T. Clarke, "Application of reinforcement learning to medium access control for wireless sensor networks," *Eng. Appl. Artif. Intell.*, vol. 46, pp. 23–32, Nov. 2015.

[70] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, "Dynamic adaptation of duty cycling with MAC parameters in cluster tree IEEE 802.15.4 networks," in *Proc. IECON-43rd Annu. Conf. IEEE Ind. Electron. Soc.*, Beijing, China, Oct. 2017, pp. 3449–3454.

[71] R. de Paz Alberola and D. Pesch, "Duty cycle learning algorithm (DCLA) for IEEE 802.15.4 beacon-enabled wireless sensor networks," *Ad Hoc Netw.*, vol. 10, no. 4, pp. 664–679, Jun. 2012.

[72] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "Dw-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proc. 9th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Hong Kong, May 2008, pp. 53–62.

[73] Z. Liu and I. Elhanany, "RL-MAC: A reinforcement learning based MAC protocol for wireless sensor networks," *Int. J. Sensor Netw.*, vol. 1, nos. 3–4, pp. 117–124, Sep. 2006.

[74] A. Carie, M. Li, C. Liu, P. Reddy, and W. Jamal, "Hybrid directional CR-MAC based on Q-learning with directional power control," *Future Gener. Comput. Syst.*, vol. 81, pp. 340–347, Apr. 2018.

[75] Y. Li, K. K. Chai, Y. Chen, and J. Loo, "QoS-aware joint access control and duty cycle control for machine-to-machine communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2014, pp. 1–6.

[76] S. Galzarano, G. Fortino, and A. Liotta, "A learning-based MAC for energy efficient wireless sensor networks," in *Proc. Int. Conf. Internet Distrib. Comput. Syst.*, Calabria, Italy, Sep. 2014, pp. 396–406.

[77] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Jun. 2014.

[78] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, Hangzhou, China, Jun. 2015, pp. 37–42.

[79] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[80] P. Thubert, M. R. Palattella, and T. Engel, "6TiSCH centralized scheduling: When SDN meet IoT," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Tokyo, Japan, Oct. 2015, pp. 42–47.

[81] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.

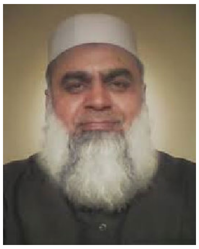
[82] *Datasheet for MicaZ Wireless Measurement System*. Accessed: Jun. 14, 2019. [Online]. Available: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[83] *Datasheet for IRIS Wireless Measurement System*. Accessed: Jun. 14, 2019. [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf

SHAHZAD SARWAR received the B.Sc. degree in civil engineering from the University of Engineering and Technology (UET) Taxila, Pakistan, in 1998, the M.S. degree in computer science from the Lahore University of Management Sciences (LUMS), Pakistan, in 2004, and the Ph.D. degree in electrical engineering and information technology from the Vienna University of Technology, Austria, in 2008. He is currently a Professor with the Punjab University College of Information Technology (PUCIT), University of the Punjab, Lahore, Pakistan. His main area of research interests are the Internet of Things (IoT), machine-to-machine (M2M) communication, optical burst switched (OBS) networks, network-on-chip (NoC), and high-speed data centers. He has made significant contributions to research. He has already published more than 40 papers in well reputed journals and conferences. His research has more than 200 citations. He has already supervised more than 10 M.Phil. theses. He has graduates two Ph.D. and is also supervising three Ph.D. students. Further, as the Principal Investigator (PI) and a Co-PI, he has secured research funding amounting around USD 1.0 million.

RABIA SIRHINDI received the B.S. degree (Hons.) in computer science from the PUCIT, University of the Punjab, Lahore, Pakistan, and the M.S. degree in information security from the College of Signals, NUST, Rawalpindi. She is currently pursuing the Ph.D. degree in computer science with the PUCIT, University of the Punjab. Her research interests include machine learning, reinforcement learning, clustering, and information security.

LAEEQ ASLAM received the B.S. degree in agricultural engineering from the University of Agriculture, Faisalabad, Pakistan, in 2000, the M.S. degree in computer science from the Lahore University of Management Sciences, Lahore, Pakistan, and the Ph.D. degree in computer science from the University of the Punjab, Lahore, Pakistan. Since 2007, he has been working as an Assistant Professor with the Punjab University College of Information Technology (PUCIT), University of the Punjab, where he is currently the Head of the Algorithms Development and Computing Laboratory (ADCL). His research interests include machine-to-machine (M2M) communication, the IoT technologies, algorithm development, graph theory, and combinatorics.



GHULAM MUSTAFA received the B.Sc. degree in civil engineering from the University of Engineering and Technology (UET), Taxila, Pakistan, the M.S. degree in computer science from the University of Management and Technology (UMT), Lahore, Pakistan, and the Ph.D. degree in computer science from UET, Lahore. He was a full-time Researcher with the High Performance Computing and Networking Laboratory (HPCNL), Al-Khwarizmi Institute of Computer Science (KICS), UET, Lahore. He joined KICS as a Senior Research Associate, in 2008, promoted to an Assistant Manager Research, in 2011, and to the Manager Research, in 2015. He has worked on various funded Research and Development projects in lead roles. He is currently working as an Assistant Professor with the School of Systems and Technology (SST), UMT. He has also organized various hands-on short courses and workshops, and published research articles. His area of research interests are high-performance computing, specifically parallel computing, parallel and distributed computing, cloud computing, scientific computing, virtualization, machine learning, and open source software development. At UMT, he has been teaching different courses to both bachelor's and master's students. He is also supervising the M.S. theses and final year projects. He also holds the position of director final year projects, with the Department of Informatics and Systems, SST.

MUHAMMAD MURTAZA YOUSAF received the Ph.D. degree from the University of Innsbruck, Austria, in 2008. He worked on networks for grid computing during his Ph.D. degree. He is currently a Professor with the PUCIT, University of the Punjab, Lahore, Pakistan. His current areas of research include parallel and distributed computing, cloud computing, data science, transport layer of networks, and interdisciplinary research.

SYED WAQAR UL QOUNAIN JAFFRY (Senior Member, IEEE) received the Ph.D. degree in computer science from VU University Amsterdam, The Netherlands. He is currently a Professor and the Director of the National Center of Artificial Intelligence (NCAI), Punjab University College of Information Technology (PUCIT), University of the Punjab, Lahore, Pakistan. He has more than 18 years of teaching and research experience at academic and research institutes. At NCAI, he is employing techniques of artificial intelligence, data mining, machine learning, and modeling and simulation to address various multidisciplinary research questions.