Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

A Reinforcement Learning-based Duty Cycle Adjustment Technique in Wireless Multimedia Sensor Networks

BAO TRINH[1] (Student Member, IEEE), LIAM MURPHY[2] (Member, IEEE), and GABRIEL-MIRO MUNTEAN[3] (Senior Member, IEEE)

[1]School of Computer Science, University College Dublin, Belfield, Dublin 4, Dublin, Republic of Ireland (e-mail: nguyen.trinh@ucdconnect.ie)
[2]School of Computer Science, University College Dublin, Belfield, Dublin 4, Dublin (e-mail: liam.murphy@ucd.ie)
[3]School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Dublin, Republic of Ireland, (e-mail: gabriel.muntean@dcu.ie)

This work was supported by the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (http://www.lero.ie).

ABSTRACT Multimedia delivery support has recently been added to Wireless Sensor Networks (WSN) and has led to increased interest in Wireless Multimedia Sensor Networks (WMSN). WMSNs are expected to be crucial to the success of applications related to the Internet of Things (IoT), such as smart health, smart surveillance, smart homes, etc. Alongside their improved multimedia capabilities, WMSNs inherit WSN limitations such as energy and processing constraints. Additionally, WMSNs have significant Quality of Service (QoS) requirements, since multimedia delivery requires increased network performance in terms of bandwidth, latency, etc. Balancing energy efficiency and QoS is a fundamental challenge for WMSN users and operators alike. This paper proposes Reinforcement Learning based Duty Cycle ($rlDC$), an innovative learning-based scheme to adjust the duty cycle and contention window of WMSN nodes in order to meet energy efficiency and QoS targets. By employing $rlDC$, WMSN sensor nodes intelligently adapt their operation according to network delivery performance and application requirements. The proposed $rlDC$ scheme was evaluated under different use cases in a simulation environment, and testing results show it outperforms other state-of-the-art duty-cycle-based protocols for WMSNs.

INDEX TERMS Wireless Multimedia Sensor Networks, Energy Efficiency, Quality of Service, Duty Cycle, Reinforcement Learning, Medium Access Control, Internet of Things.

I. INTRODUCTION

THE Internet of Things (IoT) is set to influence significantly people lives, including via services which depend on interconnecting smart devices, sensors, actuators, etc. It is estimated that in 2021, the number of devices that are connected to the Internet will be three times higher than the global population [1]. Increasing number of innovative services will be enabled by IoT, including those related to rich media streaming [2], smart surveillance [3], [4], smart home applications [5], etc.

Highly important for supporting IoT applications are Wireless Sensor Networks (WSN) and lately Wireless Multimedia Sensor Networks (WMSN). In general WMSNs incorporate a number of multimedia sensor nodes deployed in an area to acquire video/audio data from the surrounding environment and deliver it to remote servers for further processing, as illustrated in Fig. 1. Currently multimedia applications such as video conferencing, video on demand (VoD), real-time content delivery dominate Internet communications. They are expected to generate traffic which should account for approximately 75% of the overall traffic in 2020, as estimated in a Cisco report [7]. The same report states that the Internet traffic generated by video surveillance, one prominent application of WMSNs, will increase seven fold between 2017 and 2022. For instance, globally, 3.4% of all Internet video traffic is expected to be related to video surveillance in 2022, up from 1.8% in 2017.

In order to efficiently deploy WMSN applications and enable good performance, some critical aspects must be addressed. First, augmenting IoT systems with multimedia capabilities is not straightforward and requires introduction of additional functionalities and revision of existing ones. Multimedia transmission is more bandwidth hungry than the conventional data exchange in WSNs. Furthermore, WMSN traffic is bursty and has real-time delivery constraints. Secondly, multimedia sensor devices have limited resources in

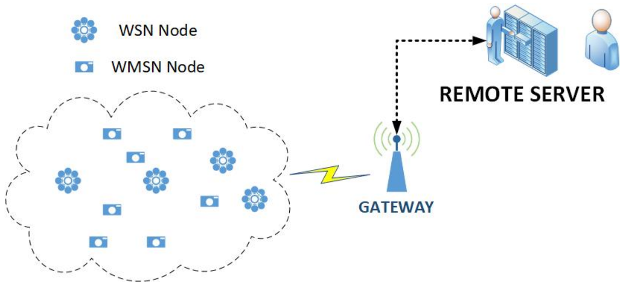

FIGURE 1. A generic Wireless Multimedia Sensor Network.

terms of processing power, memory capacity, and especially energy, whereas multimedia data requires high processing capability and timely delivery. Moreover, in many cases, WMSN nodes are usually powered by battery packs for a long time without any human intervention. Therefore, achieving energy efficiency and QoS-awareness are two critical objectives for WMSN users and operators.

Another concern of WMSN network design is related to supporting application requirements, e.g.: network lifetime, throughput, delay, reliability. For example, in surveillance or target tracking systems, sensor nodes are set up to deliver data each time an abnormal event occurs. In such a case, the quality of video streaming is the highest priority and imposes high throughput requirements. In terms of energy efficiency awareness, applications in which sensor nodes are required to perform monitoring and transmit data periodically are an example where the high priority is saving energy. Such diversity of application requirements is challenging to address in WMSN design and operation.

An efficient approach to tackle these challenges involves parameter tuning, that is adaptation of network parameter values according to application requirements. Despite the high benefits, such methods suffer from some shortcomings: *i)* Due to the unpredictable characteristics of the WMSN environment, network parameter tuning is both complex and time-consuming. *ii)* In many cases, the derived tuning parameters are not optimal. For example, given a highly configurable sensor node with many parameters and each of them can have a range of values, choosing the optimal combination is highly complex. Additionally, the dynamic variation of the sensor node's environment also contributes to the increased complexity of finding optimal parameters.

In order to facilitate decision making in relation to finding optimum network parameters, methods based on dynamic optimization [6] can be used so that sensor networks can adapt their operations according to application requirements and environment. Such methods ensure that the sensor networks execute the assigned tasks optimally, and their sensor nodes perform efficiently in the highly dynamic network environment. Among the dynamic optimization methods, Markov Decision Process (MDP) is an appropriate solution for WMSNs [33], where there is a need for optimum decision making in a highly dynamic environment with fluctuating wireless channel conditions, variable traffic and important energy constraints.

Typically, the energy consumption associated with WMSNs is dominated by node radio transmission [9]. In wireless communications, the Medium Access Control (MAC) layer is responsible for coordinating the radio network access. So, in order to optimize network lifetime for WMSNs, an effective way is to focus on energy efficiency at the MAC layer. *Duty Cycle* management techniques are among the most efficient methods to control the operation of radio transmission. Basically, duty cycle methods periodically turn ON/OFF the radio transmission of sensor nodes with the aim to conserve energy. Duty cycle methods are among the "greenest" techniques [10] and are currently used thoroughly in WSNs, including in two major operating systems especially designed for WSNs: Contiki[1] and Tiny OS[2].

A major issue when employing a Duty Cycle technique is the potential degradation of QoS, especially in terms of delay and throughput [10]. Therefore, when integrating Duty Cycle techniques at MAC layer for energy-aware data delivery over WMSNs, many research solutions have been proposed to increase QoS performance [11]- [29]. These solutions differ in terms of focus and design and are discussed in detail in section II. One of the critical points for any solution design is consideration of application diversity and associated traffic types. For example, streamed multimedia data, node location or temperature information can be transmitted at the same time for a specific application. Such applications which handle traffic of different classes, with diverse requirements, make very challenging providing QoS support. This is particularly difficult for real-time high bitrate data exchange associated with multimedia delivery.

In this paper, we study the problem of efficient content delivery in WMSNs and we solve it from the perspective of machine learning, which enable network nodes to learn how to perform the best from their own experience (e.g. runtime statistics data). We formulate the delivery problem by making use of the Markov Decision Process (MDP) framework as a decision maker. The problem is then solved by using Q-Learning [31], one of the best-known model-free reinforcement learning technique.

This paper introduces $rlDC$, an innovative machine learning-based scheme to adjust the duty cycle and transmission contention window of sensor nodes in order to balance energy efficiency and QoS. $rlDC$ acts at MAC layer in the context of WMSNs and focuses on issues related to energy consumption, and performance-aware MAC-layer parameter tuning for sensor nodes in order to meet the input requirements of applications.

The contributions of this paper are as follows:

- An overall system architecture design for WMSNs that handles a wide range of application requirements in terms of traffic types is introduced. WMSN users or operators can manage the performance by setting the

[1] contiki, http://www.contiki-os.org/
[2] tinyos, http://www.tinyos.net/

priority weighting factors for system requirements (in terms of QoS-focus or energy-related metrics).

- A MAC-based solution for sensor nodes is introduced with the aim to optimize the energy usage and achieve QoS targets. The use of a duty cycle-based method and MAC-related parameter tuning is combined in order to meet this objective.
- A reinforcement learning-based algorithm deployed at sensor node level which is at the core of $rlDC$. By employing such a model-free solution, the sensor node chooses suitable actions in terms of its duty cycle and transmission contention window values in order to meet system requirements and optimize its long term reward.
- Evaluation of the proposed solution in a NS-3 simulation environment under a wide range of traffic types and its bench-mark against other similar novel solutions.

The rest of this paper is organized as follows: section II discusses some notable works in the research literature. Section III presents the technical background including MDP and Reinforcement Learning (RL) techniques used in this work. The proposed solution is described in detail in section IV. Simulation-based testing setup and scenarios are provided and testing results are discussed in section V and section VI, respectively. Finally, the paper is concluded and future works are mentioned in section VII.

II. RELATED WORKS

This section discusses some important works that target energy efficiency and QoS at MAC layer for wireless communications in general and WSNs/WMSNs in particular.

Due to the associated benefit in terms of conserving energy, duty cycle techniques have received much attention from the research community. Duty cycle adjustment is considered one of the most effective solutions for "green" communications and was already deployed in some operating systems designed for WSNs such as Contiki and Tiny OS.

In principle, based on synchronization between nodes, the duty cycle-based approaches can be classified into three main types: *i)* synchronous, *ii)* semi-synchronous, and *iii)* asynchronous.

In the **synchronous** category, WSNs must maintain a common time reference and sensor nodes are required to exchange synchronization information to achieve and keep the necessary degree of synchronization throughout the network. Synchronization method classification includes: *i) rendezvous methods*, where all nodes turn ON/OFF their radio at the same time, or *ii) skewed/staggered*, where sensor nodes schedule their wake up in a ladder pattern according to their depth in a tree-like topology. The first type of synchronization is usually employed in Time-Division Multiple Access (TDMA)-based MAC schemes, such as: RT-ink [19] and Traffic Adaptive MAC protocol (TRAMA) [20]. Both RT-ink and TRAMA incorporate a Global Positioning System (GPS) receiver for clock synchronization. For these two schemes, the energy efficient objective can be achieved by eliminating collisions and putting nodes that do not participate in the communications into a sleep mode. The main drawback of synchronous methods is the high cost of maintaining global clock synchronization and overhead messaging for control.

Semi-synchronous-based schemes overcome such disadvantages by grouping nodes into synchronized clusters. Among these, Sensor-MAC (SMAC) [11] is among the most important duty cycle adjustment schemes for WSNs. Sensor nodes with SMAC form loosely synchronized virtual clusters that are created spontaneously as each node broadcasts its schedule to the neighbors. SMAC provides high energy efficient improvements in comparison to the classic IEEE Power Save Mode (PSM). However, the critical drawback of SMAC is the sacrifice of QoS due to its high duty cycle (of around 20%), and fixed and long sleep/active periods that lead to high latency. TMAC [12] improves SMAC's drawback by using adaptive radio turning ON/OFF and shows better performance in terms of energy saving and delay decrease.

Asynchronous approaches have been proposed in order to reduce the relative high cost of keeping synchronization in multi-hop wireless networks, so the nodes do not need to agree on a time reference. This category of duty cycle solutions makes use of preamble sampling or Low Power Listening (LPL) with the aim to reduce idle listening by transferring the energy consumption cost to the single sender from the potentially many receivers. The two examples of such a method are BMAC [21] and Wise MAC [22]. These schemes allow every node to switch to the sleep mode asynchronously and wake up periodically to check for channel activity. Since every frame is preceded by a long preamble - longer than the duration of active and sleep times combined - any node will have time to wake up, detect the preamble transmission, and stay awake to receive the incoming frame, if necessary. Alternatively, the wake-up/sleep time of the sensor nodes can be dynamically adjusted according to the network load conditions, i.e., number of active neighbor nodes, as in [23]. To reduce the energy waste due to idle listening, the On-Demand wake-up [24] leverages a low power radio (called "wake-up radio") that listens to the wake-up signal and sends an interruption to the CPU. In response, CPU activates the primary radio. However, such a method substantially increases the design complexity.

The authors of this paper have also introduced the Uplink Adaptive Multimedia Delivery (UAMD) solution [25], which makes use of a utility function and takes into account both energy consumption and throughput requirements for video streaming services. By dynamically adapting the duty cycle of sensor nodes, both energy saving and throughput oriented objectives are balanced in a better way than other state-of-the-art solutions. Another approach for implementing duty cycle technique for wireless mesh network is proposed in [26]. The duty cycle based solution is combined with an energy-aware routing protocol at network layer with the goal of balancing between energy consumption and QoS for mobile devices. The simulation results showed that such cross-layer design decrease energy consumption whereas also improve content delivery in comparison to IEEE 802.11s MAC protocol.

Another approach to improve the performance of sensor networks is to exploit historical data by using machine learning algorithms. In [27], Zhenzhen *et. al.* proposed RL-MAC which uses RL in the adaptive adjustment of the duty cycle in WSNs. RL-MAC reduces energy usage and increases throughput by optimizing the duty cycle of the network nodes. Similar to SMAC [11] and TMAC [12], RL-MAC synchronizes nodes' transmission on a common schedule in a frame-based structure. RL-MAC adapts the slot length, duty cycle, and transmission active time according to the traffic load and transmission channel bandwidth. In [28], Chu *et. al.* introduces a novel MAC protocol for WSNs named ALOHA-QIR that combines slotted ALOHA and a model-free reinforcement learning technique, Q-Learning.

ALOHA-QIR inherits the features of both ALOHA and Q-Learning and benefits from a simple design, low resource requirements and low collision probability. During transmission process, nodes broadcast their future transmission allocation such that the nodes not involved directly in data exchange can sleep during the reserved frame transmission period. The authors of this paper have proposed previously eAMD [29] which also employs Q-Learning in an application-layer systematic approach to improve the trade-off between video streaming quality and energy efficiency. eAMD performs duty cycle adjustment only.

By conducting a survey of many state-of-the-art MAC designs for WSNs, we note a lack of solutions that address specific issues related to multimedia delivery. Most proposed schemes focus on small data packets and have low bandwidth requirements for scalar sensor types. Besides, the breakthrough in hardware industry in recent years opens new opportunities to employ highly required processing power machine learning algorithms in small and low cost sensor motes. In this paper, we aim to bridge the gap between these fields in order to propose a novel adaptive duty cycle design for WMSNs.

III. MARKOV DECISION PROCESS - BACKGROUND

This section presents MDP background information and describes how we employ the MDP framework in our solution.

A. MARKOV DECISION PROCESS FRAMEWORK

MDP refers to a classical formalization of sequential decision making in terms of a number of *episodes*, where chosen actions influence not only immediate rewards, but also subsequent situations, or states, through future rewards [30]. A decision maker in MDP is defined as an *agent*. In a sensor network, MDP is used to model the interaction between a sensor node (i.e., an agent) and its surrounding environment in order to achieve some objectives e.g. data aggregation and routing, sensing coverage, target tracking, etc. [33].

In general, MDP relies on a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where:

- $\mathcal{S}$ denotes *State Space*, a finite set of all possible states of the system.
- $\mathcal{A}$ denotes *Action Space* that is a finite set of all possible actions that the agent can choose from.
- $\mathcal{P}$ refers to transition probability matrix, that includes entries such as the probability that the agent to move from state s to $s^{'}$ after choosing action a, normally abbreviated as $\mathcal{P}^{s^{'}}_{s,a} = \mathbf{P}[S_{t+1} = s^{'} | S_t = s, A_t = a]$.
- $\mathcal{R}$ is a reward function, $\mathcal{R}^a_s = \mathrm{E}[R_{t+1} | St = s, A_t = a]$
- γ denotes the discount factor, $\gamma \in [0, 1]$.

A *policy* π is a distribution of actions given the states:

$$\pi(a|s) = \mathbf{P}[\mathbf{A_t} = \mathbf{a} | \mathbf{S_t} = \mathbf{s}] \tag{1}$$

The *return* G_t is the total discounted reward in time-step t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2}$$

The *state value function* $v_\pi(s)$ gives the long-term value of state s, is the expected return starting from state s:

$$v_\pi(s) = \mathbf{E}_\pi[G_t | S_t = s] \tag{3}$$

The *action-state value function* $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following the policy π:

$$Q_\pi(s, a) = \mathbf{E}_\pi[G_t | S_t = s, A_t = a] \tag{4}$$

The optimal state-value function $v_*(s)$ is defined as the maximum value function over all policies:

$$V_*(s) = \max_\pi v_\pi(s) \tag{5}$$

The optimal action-value function $q_*(s, a)$ is the maximum action-value function over all policies:

$$Q_*(s, a) = \max_\pi Q_\pi(s, a) \tag{6}$$

The optimal value function specifies the best possible performance in the MDP. An MDP is "solved" when we know the optimal value function.

B. MODEL-FREE REINFORCEMENT LEARNING

In many cases, an MDP is considered "unknown" or *model-free* due to the unavailability of the probability transition matrix or lack of a system transition model. In such a case, the agent "learns" or optimizes the value function through episodes of experience. Such a method is called "Reinforcement Learning" (RL) and enables the agent (e.g., a sensor node) to learn by interacting with its environment. The agent learns and decides to take the best actions that maximize its long-term rewards through its gathered experience.

Q-Learning [31] is one of the best-known model-free reinforcement learning technique and is widely used in wireless communications [16] [17]. Fig. 2 illustrates the interaction between the agent and the environment. The figure shows how an agent updates its state to a new state s_{t+1} and receives a reward $r(a_t, s_t)$ following action a_t taken in current state s_t. The action-value function in Q-Learning is updated iteratively as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma Q(s_{t+1}, a^{'}) - Q(s_t, a_t)] \tag{7}$$

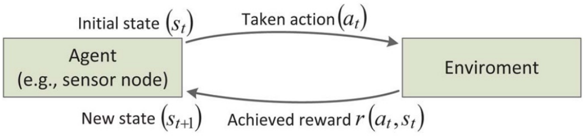

FIGURE 2. Q-Learning method principle

IV. REINFORCEMENT LEARNING-BASED DUTY CYCLE ADJUSTMENT MAC LAYER TECHNIQUE FOR WMSNS

This section describes the details of the proposed RL-based duty cycle adjustment MAC layer technique, **rlDC**. First, the system architecture that shows the general design of our solution in terms of block diagram is presented. Then, the design objectives are discussed and the problem and solution are formulated using MDP and Q-Learning as decision maker for sensor nodes.

The critical features of $rlDC$ are summarized next:

- Propose an overall system design for WMSNs that includes essential functions for feeding input system requirements.
- Calculate optimal duty cycle or wake-up/sleep duration for sensor node in order to prolong network lifetime.
- Due to the nature of CSMA/CA based protocol, $rlDC$ also derives the optimal transmission contention window for sensor nodes so that high QoS-related performance is achieved.

A. SYSTEM ARCHITECTURE

Fig. 3 presents the block diagram of the proposed system architecture. The solution design has the following three major blocks and components:

1) **WMSN APPLICATION** is located at the remote server. The **WMSN Application** module relates to the use cases by building specific profiles for each of them in **Application Profile** module. Then, a set of *application metrics* and tunable parameters (i.e. *weighting factors*) denoting the priority of system performance (that could be energy efficiency or QoS)) are derived in the **Parameters Tuning** module.
2) **WMSN GATEWAY** is responsible for receiving the requirements from network user/operator via the remote server. The **Application Profile** keeps such information and exchanges it with the **QoS Monitor** module in order to guarantee they match. **Data Aggregation** is responsible for collecting and aggregating all data received from sensor nodes in the gateway's neighbourhood.
3) **WMSN NODE** is associated with a sensor node, responsible for collecting data according to system requirements. Sensor nodes communicate with the Gateway through the **Wireless Communication Interface** module. They are also responsible for monitoring two key parameters: *i) energy*, and *ii) QoS*. These are then fed into the **Reward Function** to calculate the reward value associated with the chosen action. At the heart of a sensor node is the proposed $rlDC$ scheme that is built based on the RL technique. The **Decision Making** $rlDC$ calculates iteratively the optimal action in terms of duty cycle and contention window.

TABLE 1. Notations & Definitions

Parameter	Meaning
MDP	Markov Decision Process
RL	Reinforcement Learning
S	State Space
s_k	state s at episode k
A	Action Space
cw	Contention Window
τ	Duty Cycle
w_E	Weighting factor for Energy
w_D	Weighting factor for Delay
w_T	Weighting factor for Throughput
γ	Discount Factor
α	Learning Rate

B. PROBLEM AND SOLUTION FORMULATION

In this paper, we assume that WMSNs are deployed in an ad-hoc mode with CSMA/CA scheme as the Medium Access Control (MAC) protocol. This assumption is made in order to reduce the complexity of synchronization (e.g., in a TDMA-based system) and decrease the message overhead required to maintain time synchronization. Additionally, estimated information about the state of the network is assumed to be available at the sensor node at any time.

We denote s and a as the network state and the corresponding action, respectively, at the sensor node. The state s is comprised of the triplet composed of estimated Energy-Throughput-Delay (E, T, D) and action a is a combination between the decision of turning ON/OFF the radio transmission during duty cycle adjustment and setting the optimal contention window for accessing radio channel. Fig. 4 shows an illustration of the operation of a sensor node, combining tuning of the duty cycle and setting the contention window. In this example, a sensor node aiming to transmit data (denoted as TX) calculates its wake-up duration and optimal contention window based on the QoS requirements of the traffic. A sink/gateway node receives data (denoted as RX) and acknowledges it with an ACK frame. Other sensor nodes (Non-RX) in "active" states are listening to the channel until it is free. When the "active" time ends, these Non-RX sensor nodes switch to a "sleep" state and turn off their radios.

The main objective of our paper is to determine an optimal decision policy that tackles the energy-throughput-delay trade-off at the sensor node level.

Next, we present how the proposed solution $rlDC$ is formulated by using the MDP framework. The notations used in this section are summarized in Table 1.

1) State Space

The state space S is modeled as a triplet $\langle E, T, D \rangle$, where:

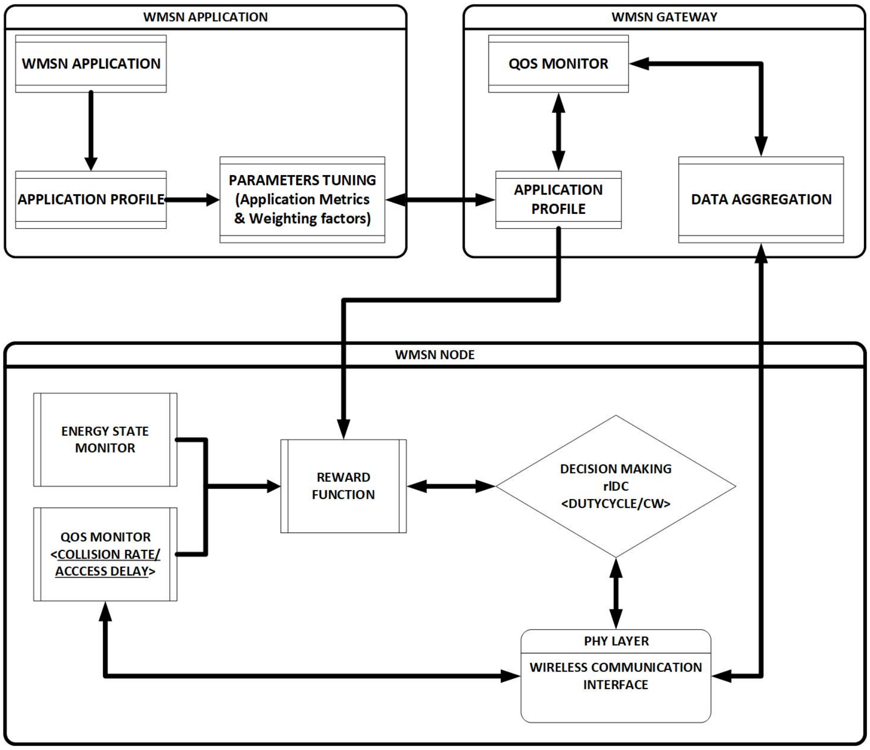
WMSN APPLICATION
WMSN APPLICATION
APPLICATION PROFILE
PARAMETERS TUNING (Application Metrics & Weighting factors)
WMSN GATEWAY
QOS MONITOR
APPLICATION PROFILE
DATA AGGREGATION
WMSN NODE
ENERGY STATE MONITOR
QOS MONITOR <COLLISION RATE/ ACCCESS DELAY>
REWARD FUNCTION
DECISION MAKING rlDC <DUTYCYCLE/CW>
PHY LAYER
WIRELESS COMMUNICATION INTERFACE

FIGURE 3. $rlDC$ Block Diagram design

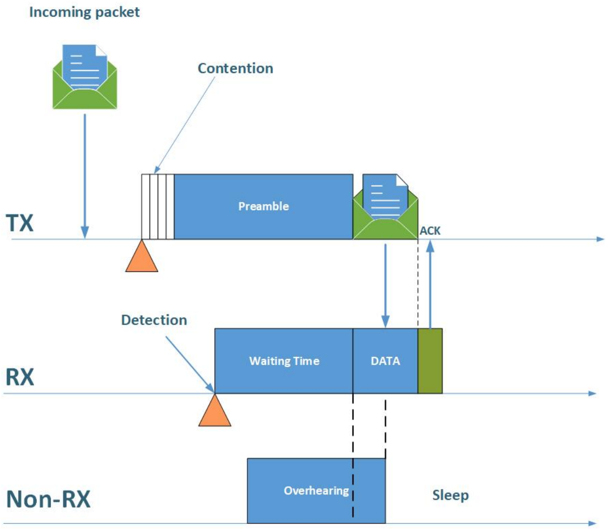

FIGURE 4. An example of using Duty Cycle in sensor networks

- E refers to energy consumption in terms of depletion rate (Joules/s)
- T denotes the estimated throughput (Mbps)
- L denotes the estimated latency/delay (seconds)

For example, in each episode k, the state s_k specifies the energy consumption, estimated throughput and delay in k.

2) Action Space

The action space A represents all possible actions that the agent (i.e. sensor node) can take in a specific state. In $rlDC$, A consists of a tuple $\langle \tau, cw \rangle$, where:

- τ denotes the value of duty cycle.
- cw refers the contention window value.

It is assumed that S and A spaces do not vary in time.

3) Reward Function

The reward function is used to calculate the reward value which evaluates how Good/Bad an action is in a specific state. Equation (8) shows the formula for calculating the reward value when sensor node performs action a_k in state s_k in episode k.

$$U(s_k, a_k) = w_E U_E(s_k, a_k) + w_T U_T(s_k, a_k) + w_L U_L(s_k, a_k) \tag{8}$$

where:

- $U_E(s_k, a_k)$, $U_T(s_k, a_k)$, and $U_L(s_k, a_k)$ denote utility functions for energy consumption, throughput, and delay, respectively.
- w_E, w_T, and w_L are weight factors (i.e. refer to the relative importance of diverse metrics) for energy, throughput, and latency, respectively. Note $w_E + w_T + w_L = 1$.

The reward function is stored at the level of application block and can be changed over time due to changes in application requirements or changes in the network. Such changes can be made by varying the weight factors w_E, w_T, and w_L, or the utility function for each component. Next, each utility function used in equation (8) is presented.

If E denotes the depletion rate of a sensor node, the Energy utility function is formalized by employing a min-max normalization as follows:

$$U_E(s_k, a_k) = \begin{cases} 0 & \text{if } E > E_{Max} \\ \frac{E_{Max}-E}{E_{Max}-E_{Min}} & \text{if } E_{Min} \leq E < E_{Max} \\ 1 & otherwise \end{cases} \tag{9}$$

In equation (9), E_{Max} and E_{Min} are two constants defining upper and lower bound depletion rate of sensor node.

Using a similar method, the delay utility function [35] is derived as in equation (10), where D_{Max} and D_{Min} refer to maximum and minimum delay requirement for the application (they are different for various use cases).

$$U_D(s_k, a_k) = \begin{cases} 0 & \text{if } D \geq D_{Max} \\ \frac{D_{Max}-D}{D_{Max}-D_{Min}} & \text{if } D_{Min} \leq D < D_{Max} \\ 1 & \text{if } 0 < D \leq D_{Min} \end{cases} \tag{10}$$

Finally, the utility function for throughput [36] is described in equation (11), where Th is the estimated throughput.

$$U_T(s_k, a_k) = \begin{cases} 0 & \text{if } Th < Th_{Min} \\ 1 - e^{\frac{-\alpha \times Th^2}{\beta + Th}} & \text{if } Th_{Min} \leq Th < Th_{Max} \\ 1 & otherwise \end{cases} \tag{11}$$

In equation (11):

- Th_{Min} and Th_{Min} refer to the minimum and maximum throughput requirements.
- α and β are two positive constants that determine the shape of the utility function.

4) Optimality Equation

Denote $\pi(a|s) = \mathcal{P}\big[A_t = a|S_t = s\big]$ as the policy or a distribution over actions given state, the Bellman expectation equation for action value function can be derived as follows:

$$Q_\pi(s,a) = \mathbf{E}\big[R_{t+1} + \gamma Q_\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a\big] \tag{12}$$

where R_{t+1} refers to the reward value achieved at the next time step $t+1$.

The ultimate goal of an MDP is to find the *optimal action-value function* $Q_*(s,a)$ that is the maximum action-value function over all policies:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a) \tag{13}$$

The optimal action is chosen accordingly:

$$a = \arg\max_{a \in A} Q_*(s,a) \tag{14}$$

C. Q-LEARNING BASED DUTY CYCLE AND CONTENTION WINDOW ADAPTATION ALGORITHM

This section describes $rlDC$, the proposed algorithm for adjustment of duty cycle and contention window for a sensor node. We employ Q-Learning, a model-free RL technique, to find the optimal action value function in an iterative way. Based on this, the sensor node chooses the optimal action.

Algorithm 1 Q-Learning based Duty Cycle and Contention Window Adaptation

```
procedure $rlDC$
Input $\langle U_E, U_D, U_T\rangle$ & $\langle w_E, w_D, w_T\rangle$ & $\langle E_{Max}, E_{Min}\rangle$ & $\langle D_{Max}, D_{Min}\rangle$ & $\langle T_{Max}, T_{Min}\rangle$
Output Duty cycle $\tau$ and contention window $cw$ values $\langle \tau, cw\rangle$
Initialize $Q\langle s, a\rangle$ to 0 for all $\forall s \in S$, $a \in A(s)$ and $Q\langle S^*, .\rangle = 0$
  for each episode $k$ do
Initialize $s = \langle E, T, D\rangle$
    for each step of episode do
Choose action $a = \langle \tau, cw\rangle$ from $s$ using $\epsilon$-greedy policy
Take action $a$
Observe reward value feedback $r$ and next state $s^{'}$
Update $Q\langle s, a\rangle \leftarrow Q\langle s, a\rangle + \alpha[r + \gamma max_a Q(s^{'}, a) - Q(s, a)]$
$s \leftarrow s^{'}$
Until no further improvement
```

Algorithm 1 is initialized by setting the input requirements in terms of utility function used for energy/throughput/delay, weight factors which specify the priority of each parameter, and threshold (i.e., Min and Max values). These depend on the applications targeted, however default values can also be used. A Q table that specifies the Q values for each pair (State, Action) is initialized with 0.

In each episode, the state of sensor node is specified by the currently estimated throughput, delay and energy. An action (i.e. duty cycle and contention window settings) associated to the state is chosen accordingly. In order to keep the trade-off between the exploration and exploitation problem of Reinforcement Learning, we employ the ϵ-greedy strategy as follows: in each step, the best action of the current state is chosen with probability of (1-ϵ) (exploitation), or else any other action is implemented with probability ϵ (exploration). The ϵ-Greedy exploration policy [30] can be summarized as follows:

$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a^* = \arg\max_{a \in A} Q(s,a) \\ \epsilon/m & otherwise \end{cases} \tag{15}$$

The utility function in equation (8) is used to calculate the feedback reward after each action is chosen and implemented. The Q-table for the (State, Action) pair is then updated by using equation (7). An episode is considered as terminal when no signification improvement in the total reward is achieved.

V. SIMULATION-BASED TESTING SETUP

This section describes how the performance of the proposed scheme is assessed via simulation-based testing. The simulation setup and test use cases are described next, as well as the QoS metrics employed for evaluation.

The WMSNs considered in the simulation-based testing using Network Simulator 3 is illustrated in Fig. 5. A number of both multimedia and scalar sensor nodes are deployed randomly surrounding the Gateway; they communicate via WiFi. The effect of the routing protocol is not considered in this paper being left for future work. Table 3 summarizes the simulation configuration for the tests.

The performance of $rlDC$ is evaluated in four **use cases** with different traffic types as follows [34]:

- **Event-driven (E)** The sensor nodes report and deliver data to the Gateway only if an event occurs. In general, when an event occurs, a large number of packets are generated and need to be delivered reliably. For such applications (e.g., surveillance, target tracking), the priority of throughput is set highest, so that content delivery is most important.
- **Query-driven (Q)** Query-driven applications share some similarities to event-driven model, except one critical point: data is requested by the sink/gateway node. Besides, data delivery in this model is two-way traffic that consists of requests from the sink/gateway and responses from sensor nodes. Low latency and high throughput must be guaranteed in order to achieve QoS performance. Examples of query-driven model are habitat monitoring and environmental control.
- **Continuous (C)** In this type of application, data originated from sensor nodes is collected and delivered at periodic intervals. Continuous class can be seen as the basic model for traditional monitoring applications. The energy conservation is considered as the highest priority whereas real-time data delay and loss can be tolerated. Examples of continuous traffic types include surveillance, and reconnaissance.
- **Hybrid (H)** This use case considers a mixture of applications and therefore the priority of all utilities is considered equal. Examples of hybrid model include surveillance application that senses and delivers both periodical temperature/humidity and event-triggered video streaming.

The parameters in the four **use cases** used in this simulation-based study are summarized in Table 2.

Modeling and simulation-based testing was performed in Network Simulator NS-3 [38] and rtDC performance was benchmarked against other three solutions: AWP [23], SMAC [11], and eAMD [29].

Note that the key point in the proposed solution is the capability of dynamic adaptation according to various application requirements in comparison to other non-learning solutions like AWP or SMAC. The results of these two schemes only change slightly when switching between various configurations.

TABLE 2. Parameters setting for the different use cases

Test cases	Notation	w_E	w_T	w_D
Event-driven	E.1	0.05	0.9	0.05
	E.2	0.25	0.5	0.25
	E.3	0.25	0.75	0.0
	E.4	0.0	0.75	0.25
Continuous	C.1	0.9	0.05	0.05
	C.2	0.5	0.25	0.25
	C.3	0.75	0.25	0.0
	C.4	0.75	0.0	0.25
Query-driven	Q.1	0.05	0.05	0.9
	Q.2	0.25	0.25	0.5
	Q.3	0.25	0.0	0.75
	Q.4	0.0	0.25	0.75
Hybrid	H.1	0.33	0.33	0.33
	H.2	0.5	0.5	0.0
	H.3	0.5	0.0	0.5
	H.4	0.0	0.5	0.5

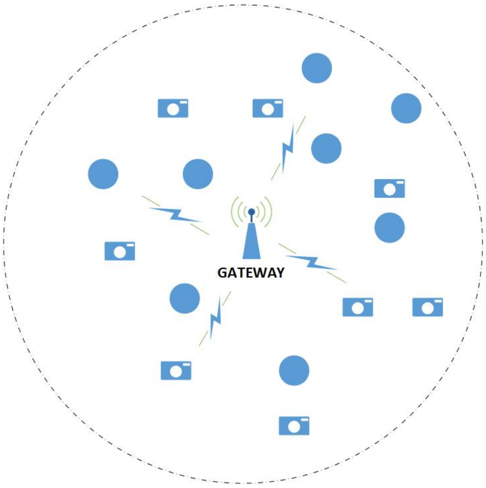

FIGURE 5. Network topology for testing

TABLE 3. Simulation setup

Parameter	Value
Simulation Length	10,000 seconds
No. of WMSN sensor nodes	40
No. of scalar sensor nodes	10
Cell layout	Single cell; Radius - 50 meters
WiFi Mode	IEEE 802.11n 2.4 GHz
Antenna Model	Isotropic Antenna Model
Initial Energy	1,000 (Joules)
Data rate for WMSN Nodes	2.0 $Mbps$
Data rate for Scalar sensor Nodes	150 $Kbps$
Learning Rate	α = 0.5
Discount Factor	γ = 0.5

SMAC and AWP schemes are designed to operate with scalar sensor type (such as humidity, temperature, etc.) with low network resource requirements. In particular SMAC results are of interest. Although the remaining battery levels are always high after simulations (approximately 44%), it shows poor performance in terms of throughput and delay. Related to AWP, sensor nodes adapt the duty cycle according to the number of active neighbors. In a densely deployed

TABLE 4. Parameter values for different application types

Notation	Description	Event-driven	Query-driven	Continuous	Hybrid
L_E	Minimum depletion rate (Joules/s)	0.10	0.10	0.10	0.10
H_E	Maximum depletion rate (Joules/s)	1.14	1.14	0.50	1.14
L_T	Minimum Throughput (Mbps)	0.8	0.12	0.12	0.12
H_T	Maximum Throughput (Mbps)	2.0	1.0	1.0	1.0
L_D	Minimum Delay (ms)	10	100	100	100
H_D	Maximum Delay (ms)	50	200	200	200

network, this causes high latency as sensor nodes always reduce their duty cycle due to a large number of neighbors. eAMD has improved performance in comparison to both AWP and SMAC as it employs a learning-based scheme for sensor node. By making decision to adjust the on/off sensor node time, eAMD shows better adaptation capability for different application requirements. By adding contention window adjustment, the scheme proposed in this paper, rlDC, shows further improvement in terms of QoS, even in comparison to eAMD.

VI. TESTING RESULTS AND DISCUSSION

A. EVENT-DRIVEN USE CASE

In the event-driven use case, sensor nodes report and deliver data only if some events occur. Surveillance and target tracking are two examples of this application type. The application performance is dependent on the quality of the observation and reliability of the information about the detected event. In such a case, for instance video streaming should be captured at acceptable level, so the weight factor for throughput (w_T) is set higher than that for energy (w_E) and delay (w_D). Fig 6 and Table 5 summarize the results for the event-driven use cases.

Average throughput of $rlDC$ achieves the highest value of 0.91 $Mbps$ when w_T is set to 0.9. This result outperforms the throughput results of 0.52 $Mbps$, 0.49 $Mbps$, and 0.16 $Mbps$ of eAMD, AWP, and SMAC, respectively. Another interesting observation is related to the effect of weight w_T on the average throughput result. Throughput of $rlDC$ in usecases E.3 and E.4 when w_T is set to 0.75, of about 0.85 $Mbps$, is higher than the 0.36 $Mbps$ obtained when w_T is set to 0.5. $eAMD$ scheme achieves a similar result decreasing throughput depending on w_T value, i.e., 0.6 $Mbps$ and 0.75 $Mbps$ in E.3 and E.4 (with w_T is set to 0.75), in comparison to 0.37 $Mbps$ in E.2 (with w_T equal to 0.5). Throughput of AWP only adapts slightly when varying w_T whereas in a static scheme like SMAC, no change in the result for different weight factors is noted.

However, improvements noted determine increases in energy consumption with an average of 0.68 $Joules/s$ of $rlDC$ (E.1) in comparison with 0.44, 0.43, and 0.27 $Joules/s$ of eAMD, AWP, and SMAC, respectively. In the remaining cases, $rLDC$ also suffers from high energy depletion rate that leads to lower remaining battery levels than when its counterparts are used. This is considered as a sacrifice of network lifetime in order to achieve higher QoS. Finally, the overall performance in terms of the achieved total reward according to equation (8) is also evaluated. The first four graphs in Fig. 6 show how $rlDC$ outperforms the other schemes in terms of the total reward.

B. CONTINUOUS USE CASE

In contrast to the Event-driven use case, in the Continuous use case, sensor networks are designed to operate for long time. This is the reason the network lifetime is most important. Figure 7 summarizes simulation results with w_E set to 0.9, 0.5, 0.75 and 0.75 for continuous use cases C.1, C.2, C.3 and C.4 when testing $rlDC$.

In the first case with w_E set to 0.9, the average depletion rate of $rlDC$ is 0.22 $Joules/s$ that is lower than that of SMAC (with 0.27 $Joules/s$), eAMD (with 0.25 $Joules/s$), and much better than that of AWP (with 0.64 $Joules/s$). A significant result of $rlDC$ is maintaining acceptable QoS performance under strict energy consumption requirements. The average throughput of $rlDC$ in case C.1 is 0.25 $Mbps$, better than that of eAMD (0.23 $Mbps$) and static scheme SMAC (0.16 $Mbps$), although lower than that of AWP (0.48 $Mbps$). In the cases with w_E set to 0.5, 0.75 and 0.75 of C.2, C.3 and C.4, $rlDC$ still shows better performance in terms of depletion rate in comparison to those of its alternative solutions. SMAC achieves a low depletion rate (approximately 0.27 $Joules/s$) at the expense of low throughput, that is around 0.16 $Mbps$ for all cases. The AWP scheme, with no concern about energy saving, achieves an average 0.49 $Mbps$ in all cases, but suffers from extremely high depletion rate (with more than 0.50 $Joules/s$) in all cases. $rlDC$ improves the idea of dynamic duty cycle of eAMD by adapting contention window based on input requirements. This leads to $rlDC$ outperforming eAMD in all four cases.

To conclude the discussion in this type of use case, the total reward illustrated in Fig. 7 is analyzed. $rlDC$ shows slightly higher reward than eAMD and AWP, and much higher than that of SMAC. Such observation illustrates $rlDC$'s ability to operate and dynamically adapt under different circumstances.

C. QUERY-DRIVEN USE CASE

In this use case, low delay is considered most important to guarantee. Fig. 8 summarizes performance results when weight factors for delay w_D are set to the highest values. In order to achieve such an objective, we design the solution with adaptive contention window to access the channel for sensor node. By varying contention window intelligently,

TABLE 5. Comparative performance results for Event-driven Use case

Case +	w_E	w_T	w_D	Schemes	Throughput (Mbps)			Depletion rate (Joules/s)			Delay (s)			%Energy
					Mean	Std. Dev	95%CI	Mean	Std. Dev	95% CI	Mean	Std. Dev	95% CI	
E.1	0.05	0.9	0.05	rlDC	0.91	0.47	(0.81; 1.0)	0.68	0.32	(0.63; 0.76)	0.4	0.32	(0.33; 0.46)	29.12
				eAMD	0.52	0.01	(0.47; 0.58)	0.44	0.06	(0.22; 0.25)	0.24	0.06	(0.22; 0.25)	32.88
				AWP	0.49	0.02	(0.48; 0.50)	0.43	0.14	(0.41; 0.45)	0.84	0.64	(0.77; 0.85)	25.25
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
E.2	0.25	0.5	0.25	rlDC	0.36	0.17	(0.31; 0.41)	0.32	0.12	(0.27; 0.39)	0.86	0.18	(0.81; 0.89)	45.19
				eAMD	0.37	0.11	(0.28; 0.40)	0.33	0.19	(0.26; 0.35)	0.90	0.21	(0.83; 1.03)	44.66
				AWP	0.5	0.02	(0.49; 0.51)	0.47	0.08	(0.45; 0.48)	0.74	0.51	(0.69; 0.66)	24.36
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
E.3	0.25	0.75	0.0	rlDC	0.86	0.75	(0.79; 0.87)	0.66	0.53	(0.60; 0.72)	0.44	0.38	(0.38; 0.46)	34.97
				eAMD	0.60	0.41	(0.57; 0.62)	0.67	0.66	(0.62; 0.72)	0.45	0.31	(0.39; 0.47)	32.62
				AWP	0.49	0.04	(0.48; 0.50)	0.46	0.09	(0.44; 0.48)	0.90	0.81	(0.86; 0.94)	40.38
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
E.4	0.0	0.75	0.25	rlDC	0.85	0.47	(0.82; 0.91)	0.44	0.33	(0.38; 0.46)	0.45	0.38	(0.41; 0.49)	41.23
				eAMD	0.75	0.46	(0.71; 0.78)	0.40	0.06	(0.21; 0.24)	0.50	0.41	(0.43; 0.52)	40.91
				AWP	0.49	0.04	(0.48; 0.50)	0.46	0.09	(0.44; 0.48)	0.79	0.61	(0.74; 0.84)	30.35
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01

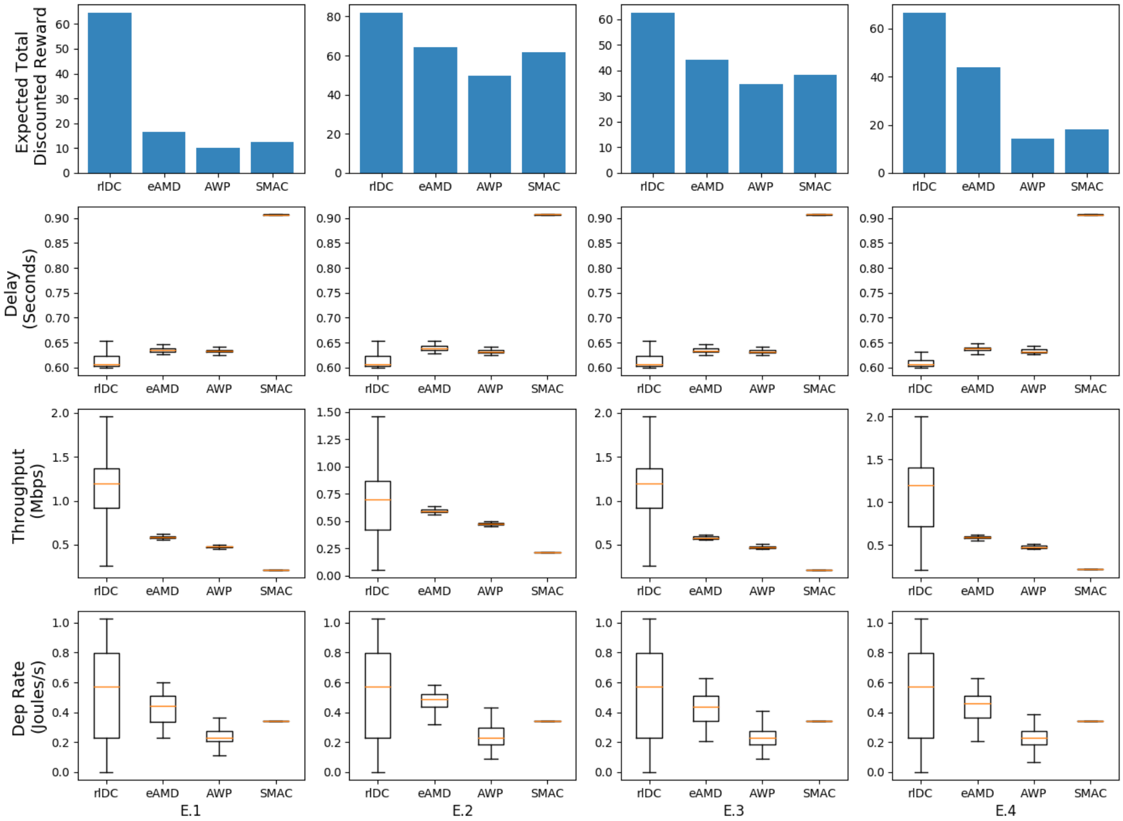

FIGURE 6. **Event-driven** use case with setting values of (w_E, w_T, w_D) (columns from left to right): **E.1)** (0.05; 0.9; 0.05); **E.2)** (0.25; 0.5; 0.25); **E.3)** (0.25; 0.75; 0.0); **E.4)** (0.0; 0.75; 0.25). The first row: Total reward gained after simulation. The second, third rows: Statistics of network performance in terms of delay (seconds) and throughput (Mbps). The fourth row: depletion rate (Joules/s) of sensor node.

TABLE 6. Comparative performance results for Continuous Use case

Case	w_E	w_T	w_D	Schemes	Throughput (Mbps)			Depletion rate (Joules/s)			Delay (s)			%Energy
					Mean	Std. Dev	95%CI	Mean	Std. Dev	95% CI	Mean	Std. Dev	95% CI	
C.1	0.9	0.05	0.05	rlDC	0.25	0.17	(0.23; 0.34)	0.22	0.12	(0.18; 0.31)	0.89	0.32	(0.73; 1.06)	45.28
				eAMD	0.23	0.02	(0.22; 0.25)	0.25	0.16	(0.24; 0.29)	0.83	0.16	(0.81; 0.85)	43.84
				AWP	0.48	0.02	(0.35; 0.50)	0.64	0.11	(0.62; 0.66)	0.82	0.11	(0.78; 0.83)	22.00
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
C.2	0.5	0.25	0.25	rlDC	0.41	0.27	(0.36; 0.45)	0.30	0.22	(0.26; 0.31)	0.80	0.28	(0.75; 0.83)	43.83
				eAMD	0.38	0.21	(0.28; 0.40)	0.33	0.17	(0.27; 0.36)	0.84	0.21	(0.78; 0.85)	39.34
				AWP	0.49	0.02	(0.48; 0.49)	0.45	0.09	(0.43; 0.47)	0.86	0.01	(0.80; 0.94)	27.84
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
C.3	0.75	0.25	0.0	rlDC	0.38	0.32	(0.32; 0.45)	0.27	0.19	(0.22; 0.37)	0.83	0.08	(0.01; 0.04)	45.28
				eAMD	0.37	0.01	(0.07; 0.08)	0.22	0.06	(0.21; 0.24)	0.83	0.01	(0.03; 0.03)	42.66
				AWP	0.49	0.02	(0.48; 0.49)	0.50	0.06	(0.49; 0.52)	0.89	0.80	(0.85; 0.94)	23.63
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
C.4	0.75	0.0	0.25	rlDC	0.40	0.33	(0.33; 0.47)	0.30	0.23	(0.21; 0.34)	0.93	0.68	(0.82; 0.95)	44.99
				eAMD	0.38	0.21	(0.34; 0.41)	0.31	0.26	(0.27; 0.35)	0.95	0.56	(0.83; 0.99)	42.57
				AWP	0.49	0.04	(0.48; 0.50)	0.46	0.09	(0.44; 0.48)	0.79	0.61	(0.74; 0.84)	27.90
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.94; 0.96)	44.01

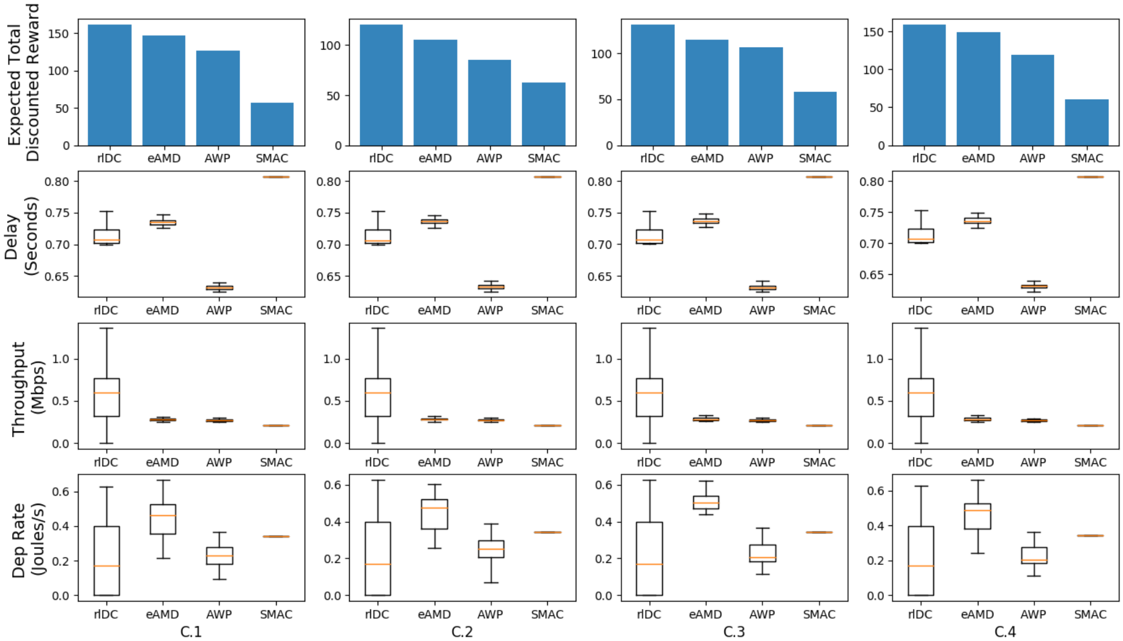

FIGURE 7. **Continuous** use case with setting values of (w_E, w_T, w_D) (columns from left to right): **C.1)** (0.9; 0.05; 0.05); **C.2)** (0.5; 0.25; 0.25); **C.3)** (0.75; 0.25; 0.0); **C.4)** (0.75; 0.0; 0.25). The first row: Total reward gained after simulation. The second, third rows: Statistics of network performance in terms of delay (seconds) and throughput (Mbps). The fourth row: depletion rate (Joules/s) of sensor node.

sensor nodes with strict delay requirements have higher chance to gain channel access in order to meet application requirements.

When w_D is set to 0.9, the average delay of $rlDC$ was about 0.30 seconds in comparison to 0.33, 0.79, and 0.95 seconds of eAMD, AWP and SMAC, respectively. When w_D is equal to 0.5, 0.75 and 0.75 as in Q.2, Q.3 and Q.4, the results for $rlDC$ vary slightly but still outperform those of its counterparts. A noticeable performance of $rlDC$ is also in terms of the energy depletion rate, with approximately 0.38, 0.40, 0.48, and 0.49 $Joules/s$ in Q.1, Q.2 Q.3, and Q.4 respectively. These results outperform those of eAMD and AWP although they are still lower than those of SMAC. The reason of this improvement is the adaptive contention

TABLE 7. Comparative performance results for Query-driven Use case

Case	w_E	w_T	w_D	Schemes	Throughput (Mbps)			Depletion rate (Joules/s)			Delay (s)			%Energy
					Mean	Std. Dev	95%CI	Mean	Std. Dev	95% CI	Mean	Std. Dev	95% CI	
Q.1	0.05	0.05	0.9	rlDC	0.82	0.47	(0.76; 0.83)	0.38	0.22	(0.33; 0.40)	0.30	0.18	(0.25; 0.32)	35.47
				eAMD	0.80	0.72	(0.77; 0.82)	0.42	0.36	(0.36; 0.44)	0.33	0.11	(0.26; 0.35)	32.19
				AWP	0.50	0.03	(0.48; 0.51)	0.46	0.09	(0.44; 0.48)	0.79	0.20	(0.74; 0.82)	22.21
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
Q.2	0.25	0.25	0.5	rlDC	0.81	0.27	(0.76; 0.90)	0.40	0.23	(0.32; 0.44)	0.34	0.29	(0.31; 0.44)	42.22
				eAMD	0.77	0.31	(0.67; 0.88)	0.41	0.27	(0.33; 0.45)	0.36	0.21	(0.33; 0.46)	39.67
				AWP	0.49	0.02	(0.48; 0.49)	0.47	0.09	(0.45; 0.49)	0.87	0.81	(0.84; 0.92)	10.9
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
Q.3	0.25	0.0	0.75	rlDC	0.45	0.35	(0.38; 0.52)	0.48	0.32	(0.73; 0.86)	0.33	0.27	(0.31; 0.38)	42.22
				eAMD	0.42	0.31	(0.37; 0.48)	0.81	0.56	(0.76; 0.84)	0.35	0.31	(0.29; 0.39)	40.37
				AWP	0.49	0.04	(0.48; 0.50)	0.46	0.09	(0.44; 0.48)	0.83	0.71	(0.79; 0.85)	28.64
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
Q.4	0.0	0.25	0.75	rlDC	0.43	0.39	(0.41; 0.49)	0.49	0.33	(0.71; 0.84)	0.35	0.35	(0.31; 0.40)	41.29
				eAMD	0.42	0.41	(0.37; 0.48)	0.59	0.76	(0.71; 0.84)	0.39	0.31	(0.33; 0.43)	39.95
				AWP	0.49	0.04	(0.48; 0.50)	0.46	0.09	(0.44; 0.48)	0.82	0.81	(0.78; 0.84)	21.29
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01

window adjustment. Sensor nodes, although with low duty cycle (illustrated by low energy depletion rate) still achieve low delay due to low adapted contention window.

For Query-driven applications, the total reward achieved by all four schemes are also compared. By achieving significant performance in terms of delay and depletion rate, $rlDC$ outperforms eAMD, AWP, and SMAC in terms of total reward as illustrated in Fig. 8.

D. HYBRID USE CASE

Fig. 9 summarizes simulation results when considering the combined requirements of multiple different applications. In hybrid use cases, the requirements can be a mixture between two (by setting two weight factors equal) or three (with all three weight factors set equal).

In the first case H.1, when all w_E, w_T, and w_D are set to 0.33, $rlDC$ achieves 0.60 $Mbps$ throughput, 0.32 $Joules/s$ energy consumption rate, and 0.34 seconds for delay. Such a balanced scheme produces an "average" result for all three metrics and $rlDC$ is slightly better than eAMD with 0.57 $Mbps$, 0.41 $Joules/s$, and 0.39 seconds in terms of throughput, energy depletion rate, and delay, respectively. AWP and SMAC achieve the same results as in the previous use cases.

In case H.2, in which energy and throughput are focused on, AWP outperforms all schemes in terms of throughput (about 0.49 $Mbps$), but at the cost of higher energy depletion rate (0.41 $Joules/s$) and higher delay (0.60 seconds) in comparison to $rlDC$. When w_T and w_E are the only two settings of interest in case H.3, $rlDC$ achieves 0.34 seconds and 0.33 Joules/s for delay and depletion rate, respectively. These results outperform those of SMAC and AWP although at the cost of decreased throughput ($rlDC$ achieves 0.38 Mbps in comparison to 0.49 Mbps of AWP).

In the last case, where throughput and delay are focused on, $rlDC$ has better results (with 0.87 $Mbps$ and 0.4 seconds) than AWP (0.5 $Mpbs$ and 0.84 seconds), and eAMD (0.78 $Mbps$ and 0.49 seconds). For SMAC, in all setting, the results do not change significantly due to its usage of fixed duty cycle and no-adaptation of other parameters.

Finally, the overall performance of our solution is evaluated in terms of the total reward and illustrated in Fig. 9. In all four cases, $rlDC$ outperforms eAMD, AWP, and SMAC in terms of total reward. This indicates the ability of our solution to perform dynamic adaptation and achieve better trade-off between QoS and energy consumption than the other schemes.

VII. CONCLUSIONS

This paper proposed a Reinforcement Learning-based framework for optimizing the performance of WMSNs. The solution model is built based on MDP and solved by employing a model free reinforcement learning technique, Q-Learning. This solution was incorporated in a novel adaptation algorithm $rlDC$ based on which sensor nodes make adjustment decisions in terms of duty cycle and transmission contention window. By deploying this algorithm, WMSN nodes can adapt dynamically their operation according the the requirements of applications. The proposed scheme was evaluated in a simulation environment and its performance was validated under different use cases that stress the priority of energy, throughput, and delay, respectively. The simulation results show how our solution outperforms three other schemes in four different use cases and diverse scenarios.

REFERENCES

[1] D. Evans, “The Internet of Things: how the next evolution of the internet is changing everything,” CISCO white paper, 2011.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015.

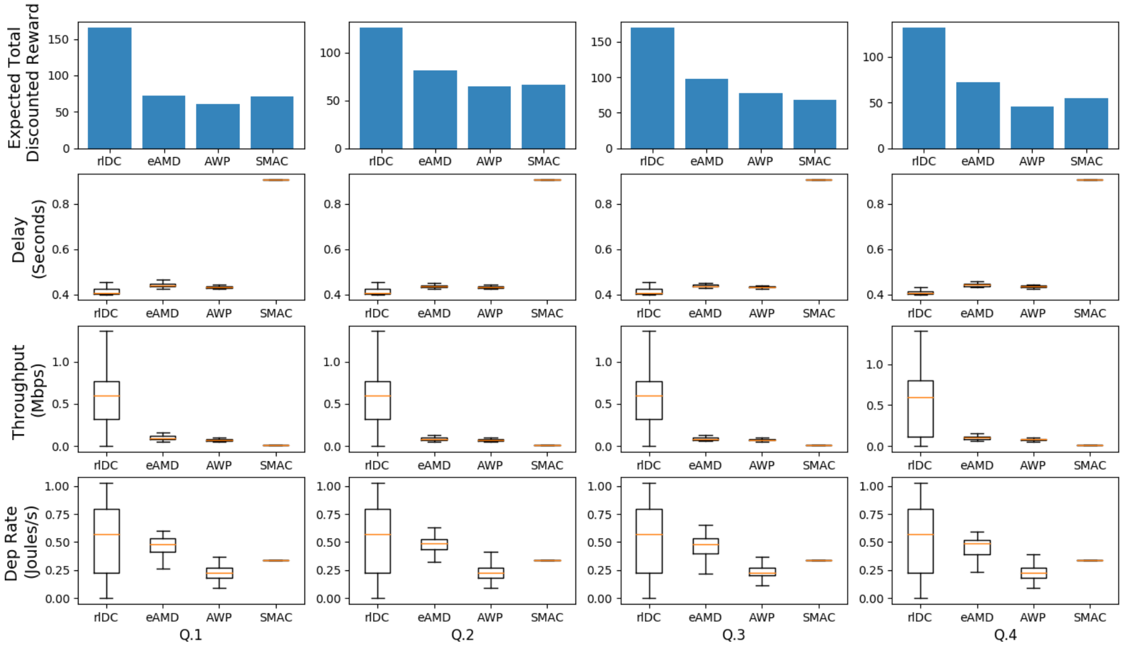

FIGURE 8. **Query-driven** use case with setting values of (w_E, w_T, w_D) (columns from left to right): **Q.1)** (0.05; 0.05; 0.9); **Q.2)** (0.25; 0.25; 0.5); **Q.3)** (0.25; 0.0; 0.75); **Q.4)** (0.0; 0.25; 0.75). The first row: Total reward gained after simulation. The second, third rows: Statistics of network performance in terms of delay (seconds) and throughput (Mbps). The fourth row: depletion rate (Joules/s) of sensor node.

TABLE 8. Comparative performance results for Hybrid Use case

Case	w_E	w_T	w_D	**Schemes**	**Throughput (Mbps)**			**Depletion rate (Joules/s)**			**Delay (s)**			**%Energy**
					Mean	**Std. Dev**	**95%CI**	**Mean**	**Std. Dev**	**95% CI**	**Mean**	**Std. Dev**	**95% CI**	
H.1	0.33	0.33	0.33	rlDC	0.60	0.27	(0.51; 0.76)	0.32	0.12	(0.28; 0.36)	0.34	0.38	(0.29; 0.38)	43.16
				eAMD	0.57	0.16	(0.51; 0.59)	0.41	0.12	(0.39; 0.44)	0.39	0.41	(0.33; 0.43)	42.26
				AWP	0.49	0.02	(0.48; 0.49)	0.48	0.08	(0.47; 0.50)	0.94	0.51	(0.90; 0.99)	21.51
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
H.2	0.5	0.5	0.0	rlDC	0.40	0.27	(0.34; 0.42)	0.32	0.22	(0.29; 0.34)	0.87	0.39	(0.81; 0.94)	44.06
				eAMD	0.38	0.11	(0.30; 0.42)	0.34	0.17	(0.29; 0.35)	0.87	0.41	(0.83; 0.93)	41.67
				AWP	0.49	0.03	(0.48; 0.49)	0.41	0.1	(0.39; 0.43)	0.78	0.60	(0.73; 0.84)	28.96
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
H.3	0.5	0.0	0.5	rlDC	0.38	0.15	(0.33; 0.42)	0.33	0.22	(0.29; 0.35)	0.34	0.32	(0.25; 0.36)	42.91
				eAMD	0.34	0.21	(0.27; 0.38)	0.31	0.26	(0.26; 0.34)	0.37	0.37	(0.33; 0.43)	41.05
				AWP	0.49	0.02	(0.49; 0.51)	0.48	0.07	(0.47; 0.51)	0.79	0.41	(0.75; 0.84)	26.75
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01
H.4	0.0	0.5	0.5	rlDC	0.87	0.38	(0.78; 1.02)	0.69	0.38	(0.61; 0.74)	0.4	0.38	(0.31; 0.44)	42.18
				eAMD	0.78	0.41	(0.67; 0.82)	0.72	0.26	(0.63; 0.77)	0.49	0.41	(0.33; 0.43)	41.85
				AWP	0.5	0.02	(0.49; 0.50)	0.44	0.09	(0.42; 0.46)	0.84	0.81	(0.74; 0.88)	25.40
				SMAC	0.16	0.01	(0.15; 0.17)	0.27	0.01	(0.26; 0.27)	0.95	0.15	(0.93; 0.96)	44.01

[3] L. Hu and Q. Ni, "IoT-Driven Automated Object Detection Algorithm for Urban Surveillance Systems in Smart Cities," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 747-754, April 2018.

[4] C. Fan, Y. Wang and C. Huang, "Heterogeneous Information Fusion and Visualization for a Large-Scale Intelligent Video Surveillance System," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 4, pp. 593-604, April 2017.

[5] M. R. Alam, M. B. I. Reaz and M. A. M. Ali, "A Review of Smart Homes—Past, Present, and Future," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 1190-1203, Nov. 2012.

[6] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen and L. Hanzo, "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 550-586, Firstquarter 2017.

[7] Cisco Visual Networking Index: Forecast and Trends, 2017–2022

[8] R. C. Carrano, D. Passos, L. C. S. Magalhaes and C. V. N. Albuquerque, "Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor

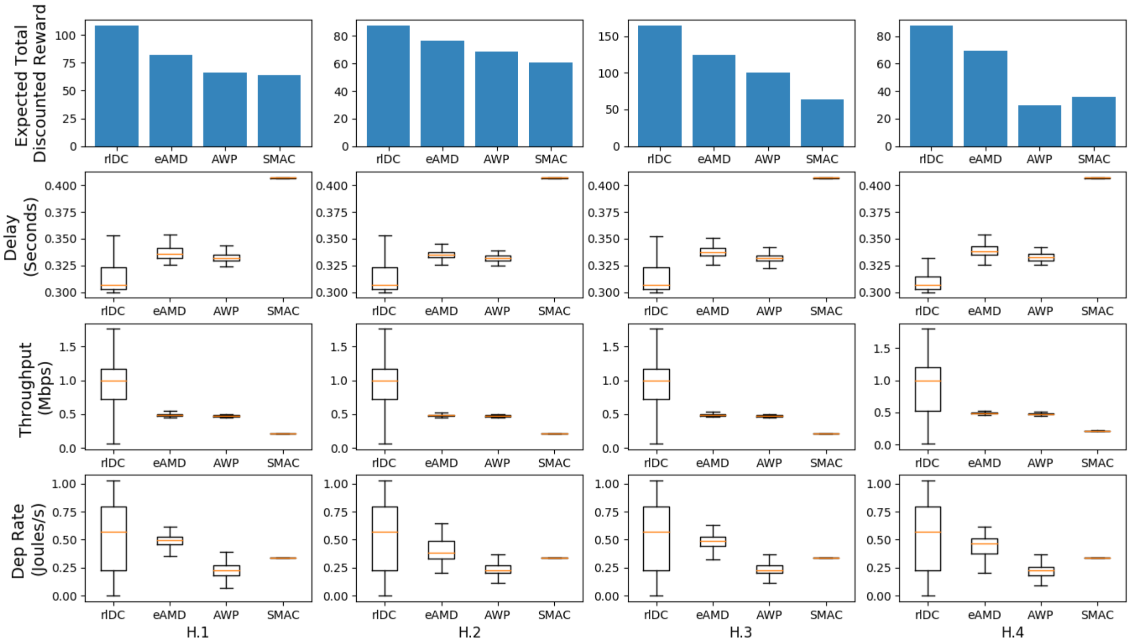

FIGURE 9. Hybrid use case with setting values of (w_E, w_T, w_D) (columns from left to right): **H.1)** (0.33; 0.33; 0.33); **H.2)** (0.5; 0.5; 0.0); **H.3)** (0.5; 0.0; 0.5); **H.4)** (0.0; 0.5; 0.5). The first row: Total reward gained after simulation. The second, third rows: Statistics of network performance in terms of delay (seconds) and throughput (Mbps). The fourth row: depletion rate (Joules/s) of sensor node.

Networks," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 181-194, First Quarter 2014.

[9] A. Bachir, M. Dohler, T. Watteyne and K. K. Leung, "MAC Essentials for Wireless Sensor Networks," in IEEE Communications Surveys & Tutorials, vol. 12, no. 2, pp. 222-248, Second Quarter 2010.

[10] R. C. Carrano, D. Passos, L. C. S. Magalhaes and C. V. N. Albuquerque, "Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 181-194, First Quarter 2014.

[11] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA, 2002, pp. 1567-1576 vol.3.

[12] T.-V. Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks”, First ACM Sensys, Nov. 2003, pp.171-180.

[13] N. Saxena, A. Roy, and J. Shin, "Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks", Computer Networks 52 (13) (2008) 2532–2542, doi:10.1016/j.comnet.2008.05.009.

[14] J. Tan, M.-C. Chan, H.-X. Tan, P.-Y. Kong, and C.-K. Tham, "A medium access control protocol for UWB sensor networks with QoS support", in: 33rd IEEE Conference on Local Computer Networks, LCN 2008, 2008, pp. 289–296. doi:10.1109/LCN.2008.4664182.

[15] I. Slama, B. Shrestha, B. Jouaber, and D. Zeghlache, "A hybrid MAC with prioritization for wireless sensor networks", in: 33rd IEEE Conference on Local Computer Networks, LCN 2008, 2008, pp. 274–281. doi:10.1109/LCN.2008.4664180.

[16] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," in IEEE Wireless Communications, vol. 24, no. 2, pp. 98-105, April 2017.

[17] M. A. Alsheikh, S. Lin, D. Niyato and H. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996-2018, Fourthquarter 2014.

[18] R. Elhabyan, W. Shi and M. St-Hilaire, "Coverage protocols for wireless sensor networks: Review and future directions," in Journal of Communications and Networks, vol. 21, no. 1, pp. 45-60, Feb. 2019.

[19] A. Rowe, R. Mangharam, and R. Rajkumar, “RT-Link: A global time-synchronized link protocol for sensor networks,” Ad Hoc Netw.,vol.6, no. 8, pp. 1201–1220, November 2008.

[20] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy- efficient collision-free medium access control for wireless sensor net- works,” in Proc. 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys ’03). New York, NY, USA: ACM, 2003, pp. 181–192.

[21] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proc. 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys ’04).New York, NY, USA: ACM, 2004, pp. 95–107.

[22] A. El-Hoiydi and J.-D. Decotignie, “WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks,” in Algorithmic Aspects of Wireless Sensor Networks - Lecture Notes in Computer Science, S. Nikoletseas and J. Rolim, Eds. Springer Berlin / Heidelberg, 2004, vol. 3121, pp. 18–31.

[23] J.-H. Lee, "A traffic-aware energy efficient scheme for WSN employing an adaptable wakeup period", Wireless Personal Communications, 2013, pp. 1879–1914.

[24] F. Z. Djiroun and D. Djenouri, "MAC Protocols With Wake-Up Radio for Wireless Sensor Networks: A Review," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 587-618, Firstquarter 2017.

[25] B. T. Nguyen, L. Murphy and G.-M. Muntean, "Uplink Adaptive Multimedia Delivery (UAMD) scheme for Video Sensor Network," 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, 2017, pp. 85-90.

[26] S. Chen, Z. Yuan and G.-M. Muntean, "Balancing Energy and Quality Awareness: A MAC-Layer Duty Cycle Management Solution for Multimedia Delivery Over Wireless Mesh Networks," in IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 1547-1560, Feb. 2017.

[27] Z. Liu, I. Elhanany, "RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks," 2006 IEEE Interna-

tional Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, 2006, pp. 768-773.

[28] Y. Chu, P. D. Mitchell and D. Grace, "ALOHA and Q-Learning based medium access control for Wireless Sensor Networks," 2012 International Symposium on Wireless Communication Systems (ISWCS), Paris, 2012, pp. 511-515.

[29] B. T. Nguyen, L. Murphy and G.-M Muntean, "Enhanced scheme for adaptive multimedia delivery over wireless video sensor networks," 2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Cagliari, 2017, pp. 1-7.

[30] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction. Second edition", A Bradford Book. The MIT Press, 2014, 2015.

[31] C. J. C. H. Watkins, P. Dayan ”Q-Learning” - Machine learning, vol. 8, issue 3, Springer, 1992, Issue 3, pp. 279–292

[32] G. Premsankar, M. Di Francesco and T. Taleb, "Edge Computing for the Internet of Things: A Case Study," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1275-1284, April 2018.

[33] M. A. Alsheikh, D. T. Hoang, D. Niyato, H. Tan and S. Lin, "Markov Decision Processes With Applications in Wireless Sensor Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1239-1267, thirdquarter 2015.

[34] M. A. Yigitel, O. D. Incel, and C. Ersoy, "QoS-aware MAC protocols for wireless sensor networks: A survey." Computer Networks 55.8 (2011)

[35] A. Munir and A. Gordon-Ross, “An mdp-based application oriented optimal policy for wireless sensor networks,” in Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis, ser. CODES+ISSS ’09, 2009, pp. 183–192.

[36] R. Trestian, O. Ormond and G.-M. Muntean, ”Energy–Quality–Cost Tradeoff in a Multimedia-Based Heterogeneous Wireless Network Environment,” in IEEE Transactions on Broadcasting, vol. 59, no. 2, pp. 340-357, June 2013.

[37] F. Al-Turjman, A. Radwan, "Data Delivery in Wireless Multimedia Sensor Networks: Challenging and Defying in the IoT Era," in IEEE Wireless Communications, vol. 24, no. 5, pp. 126-131, October 2017.

[38] Network Simulator v. 3 [Online] https://www.nsnam.org

BAO-NGUYEN TRINH received the B.Eng. degree in telecommunications from Hanoi University of Science and Technology, Hanoi, Vietnam, in 2009 and the M.Eng. degree in telecommunications from Dublin City University, Dublin, Ireland in 2014. He is currently working toward the Ph.D. degree with the Performance Engineering Laboratory, School of Computer Science, University College Dublin, Ireland. His research interests include protocol design for Internet of Things system, adaptive multimedia delivery, and resource management.

LIAM MURPHY received a B.E. in Electrical Engineering from University College Dublin in 1985, and an M.Sc. and Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1988 and 1992 respectively. He is currently a Full Professor of Computer Science and Informatics at University College Dublin. Prof. Murphy has published almost 200 refereed journal and conference papers on various topics, including multimedia transmissions, dynamic and adaptive resource allocation in computer/communication networks, and software performance. Prof. Murphy is a Member of the IEEE (Communications, Broadcasting, and Computer societies) and a Fellow of the Irish Computer Society.

GABRIEL-MIRO MUNTEAN (M’04–SM’17) is an Associate Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of the DCU Performance Engineering Laboratory. He has published over 350 papers in top-level international journals and conferences. He has authored four books and 19 book chapters and he has edited seven additional books. Dr. Muntean’s research interests include quality, performance, and energy saving issues related to multimedia and multiple sensorial media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and a Reviewer of important international journals, conferences, and funding agencies. He is the Project Coordinator of the EU-funded Project NEWTON.