SAGE: Transmission Power Management for Deadline-aware IoT Edge Networks

1st Shanti Chilukuri
Dept. of Computer Science and Engineering
GITAM (deemed to be university)
chilukurishanti@gmail.com

2nd Vinayaka Shashank Varanasi
StoryXpress
Gurgaon, India
sasi8998vv@gmail.com

3rd Kalyana Chakravarthy C.
MVGR College of Engineering
Vizianagaram, India
kch.chilukuri@gmail.com

***Abstract*—Wireless networks with resource-constrained end devices are useful for a diverse range of applications, but create a need for reduction of the energy consumed by the devices. In addition, such networks have to meet application-imposed Quality of Service requirements such as low, bounded delay and reliable communication. Edge networks with Time Division Multiple Access (TDMA) reduce collisions and provide bounded end-to-end packet delay, but finding deadline-aware TDMA schedules is an NP-hard problem, even without focusing on the energy expenditure. In this paper, we propose SAGE, a Deep Reinforcement Learning-based strategy for learning transmiSsion power management for deAdline-aware multihop IoT edGE networks. The transmission power plan created by SAGE determines the time slot and adaptive transmission power to be used by each end device to transmit/relay data, such that the number of packets missing their deadlines is minimized, while also minimizing the total energy spent in packet transmission and reception. This is achieved by a carefully crafted reward function of the DRL agent that leads to the sequential optimization of these two goals. Extensive evaluations of SAGE for a number of network scenarios show that it can result in up to 48% energy savings compared to baseline and other recent deadline-aware scheduling and routing schemes, while performing equally well or better, in terms of timely packet delivery.**

***Index Terms*—adaptive power control, deep reinforcement learning, deadline-aware networks, green networking, IoT edge**

I. Introduction

In recent times, the availability of low-cost electronic devices has triggered their application in a variety of domains, leading to the Internet of Things (IoT). Some important IoT applications are process control and predictive maintenance in industries, applications for smart cities and environment monitoring. Such applications pose a new set of challenges as far as data communication and network management are concerned, as they use end devices which typically have limited processing power and memory, with limited battery power. In addition, each application may have a specific set of Quality of Service (QoS) requirements. For example, applications like pollution / shoreline monitoring may tolerate reasonable delays, while others (e.g., gas leak detection, process control in industrial IoT (IIoT)) may have strict delay constraints.

Satisfying the QoS requirements of an application, *while* honoring the resource constraints, is a complex task. This is especially difficult in cases where there is a trade-off between the QoS parameter of interest and the constrained resource. To make such problems manageable, constrained nodes are often organized as networks where the producer (typically a sensor) and consumer (an actuator or data processing node) of data belong to the same *edge* network, which is centrally controlled by a network controller (sometimes co-located with the access point or base station). Edge networks help in limiting the scale and heterogeneity for ease of network management.

In multihop edge networks, the Medium Access Control (MAC) delay is a major component of the end-to-end delay. Near-deterministic MAC delay can be obtained by using a schedule-based MAC such as Time Division Multiple Access (TDMA), which helps the nodes conserve energy due to minimized collisions and also by coordinated sleep-wakeup cycles. However, finding schedules that meet packet deadlines (called *feasible schedules*) for a given network is an NP-hard problem [1] and hence, several heuristics have been proposed for it in the past [2], [3].

The main goal of most delay-aware TDMA scheduling heuristics (e.g., [4]–[6]) is to find feasible schedules that meet the packet deadline requirement without attention to the energy spent by the nodes. Transmission power is a major component of the overall energy consumed in the network and hence impacts the network lifetime. Encouraged by the success of Machine Learning (ML) in solving optimization problems in wireless networks [7], we study the efficacy of Deep Reinforcement Learning (DRL) for minimizing energy consumption due to wireless communication of data with adaptive transmission power control, while meeting packet deadlines. While *adaptive power control* is a well-studied problem in the context of cellular networks and wireless local area networks, it is relatively unexplored in the context of deadline-aware, multihop, constrained edge networks.

We propose SAGE, a scheme for learning transmiSsion power management for deAdline-aware multihop IoT edGE networks. SAGE uses DRL to learn to create a transmission power schedule, which gives the time slot in and corresponding power with which each node in the edge network should transmit/receive data. Our main contributions in this paper are:

- design of a DRL agent's state and action spaces to find transmission power and time slot schedule for all the network nodes,
- design of a carefully crafted reward function that sequentially optimizes the number of packets missing their

ISBN 978-3-903176-72-0 © 2025 IFIP

deadlines **and** the energy needed for transmission and reception across the network and
- extensive evaluation and analysis of the proposed scheme for a wide variety of network scenarios with different features such as the number of nodes, flows, channels, node density and tightness of deadlines.

Simulation results show that depending on the network scenario, SAGE can result in up to 48% energy savings compared to the baseline, while minimizing the number of packets missing their deadlines.

II. The Motivation for SAGE

Most standards for low power networks allow nodes to change their transmission power [8]. For example, IEEE 802.15.4 specifies the minimum transmission power for an end device (1milliWatt) and devices may transmit using more power than this value. Transmission power control is done for one of the three reasons - to reduce interference in the network and increase its capacity, to conserve energy and prolong the battery life of network nodes or to adapt the network to changing channel conditions and provide a desired Quality of Service [9]. For given channel conditions and receiver sensitivity, increasing the transmission power results in better signal quality at the receiver and hence, lower Bit Error Rate (BER). However, this also results in quick drainage of device battery power and increased interference with other signals. Reducing the transmission power beyond a threshold, on the other hand, may result in high BER and hence, low Packet Delivery Rate (PDR). This may result in retransmissions, requiring the usage of more energy to meet the application PDR requirements. This power-BER trade-off has been widely studied by researchers [10].

We focus on an important side-effect of changing the transmission power – the resulting change in the number of neighbors of a transmitting node. This, in turn, affects the routes possible between a source and a destination node. In TDMA systems, the route affects the length of packet queues at each node and hence, the end-to-end packet delays. For example, in IoT edge networks where all data converges to an edge node, a greedy (low) transmission power choice may result in more hops between the source and the destination, leading to long packet queues at nodes close to the edge node, besides the delay due to possible retransmissions resulting from low PDR.

Finding a feasible schedule of transmission slots (and channels, in case of multi-channel radios) in TDMA networks with packet deadlines is NP-hard [2]. Without a proper schedule, packets may have to wait to be relayed at a node for subsequent super frames, leading to missed deadlines. In [1], [3], [5], heuristics or ML-based solutions had been proposed to find feasible schedules.

The takeaway from the above discussion is that *the choice of the transmission power used impacts both the energy consumption and the possible routes and the choice of the TDMA schedule for a given route affects the end-to-end packet delay*. In constrained networks such as those at the IoT edge, minimizing the overall energy consumption and packets missing their deadlines are both important. *SAGE uses DRL for combining adaptive transmission power and time slot scheduling to meet both the above goals in the best possible manner.*

III. Reinforcement Learning – Basics

Reinforcement Learning is a kind of ML strategy that has been gathering a lot of attention for decision-making in systems that are difficult to model. In RL, the agent (e.g., a network controller) learns an *optimal policy* $\pi^*(a|s)$, which gives the most optimal action a to take when in a particular state s at time t. The goal of the optimal policy is to optimize the long term reward R. During the learning phase, the RL agent observes the state of the system, tries an action (either by exploring the action space or exploiting the previously known optimal policy) and gets a scalar reward. With enough training, the agent learns the best action to be taken when in any state. A detailed discussion of RL is beyond the scope of this paper and interested readers may please refer to [11].

For large and/or continuous state and/or action spaces, it may not be possible for the agent to see all state-action combinations during training. In this case, a deep neural network can be used to approximate the relation between the policy and the reward. This is called Deep Reinforcement Learning (DRL). Actor-critic methods are a type of DRL algorithms that have been proven to give good results. SAGE uses Proximal Policy Optimization (PPO) [12] which is an actor-critic method. PPO uses first-order derivatives with soft constraints and hence, has better scalability than natural policy gradient methods. In addition, PPO with a clipped objective function can avoid large changes to the policy and ensure better and faster convergence. PPO is one of the most efficient DRL algorithms as it is both scalable and converges fast and is hence our choice for SAGE. Please see [12], for more details about PPO.

IV. Related Work

RL and DRL have been successfully applied to find optimal actions in several systems (including task and time-slot scheduling) that are difficult to model, but can be represented as Markov Decision Processes [4], [13]. Research in ML (including RL) for IoT applications such as building energy management systems and smart energy grids has been surveyed in [14]–[16]. From these surveys, it can be observed that RL can result in better performance when used for network management in IoT edge networks.

Energy efficiency for constrained networks using ML has been a topic of interest for sometime now. In [17], the authors propose an Enhanced Energy Optimization Model for Industrial Wireless Sensor Networks which uses knowledge-based learning to identify and optimize the energy consumption of nodes. In [18], the effectiveness of Non-Orthogonal and Orthogonal Multiple Access for improving energy efficiency, content distribution, latency, and transmission speeds using RL was explored.

Transmission power control is a well-studied problem for minimizing interference and/or energy consumption. In [19], the authors propose a distributed, joint power and data rate control mechanism that takes into cognizance the congestion levels in a wireless network. The aim of this work is to achieve a desired Signal to Interference Ratio (SIR). The work presented in [20] proposes Adaptive Transmission Power Control, where each node creates a model for each of its neighbors. This model gives the correlation between transmission power and link quality for each $pair$ of nodes. This is better than power control done across the network or at a node (irrespective of the neighbor). [21] proposes a proportional-integral-derivative (PID) kind of data-driven algorithm for power control. In this approach, the channel conditions are taken as feedback to set the gain for multiple PID controllers, to achieve good SINR values in a cell with multiple users. More recently, the SmartAPM framework has been proposed for adaptive power management in wearable devices using deep reinforcement learning [22]. However, none of this literature considers packet deadlines or scheduling.

Resource scheduling in edge networks [23], especially deadline-aware scheduling (and routing), have been studied by many researchers in the past. In [1], the authors propose a scheduling heuristic for real time, centralized scheduling of transmissions in a WirelessHART (TDMA) network, while [6] proposes a distributed scheduling scheme for the same. DRL has also been studied by some researchers for deadline-aware scheduling, both with joint routing [4] and without routing [5]. More recently, a joint routing and scheduling scheme using DRL for prioritized data in WirelessHart, Industrial IoT networks has been proposed in [24]. This work focuses on reducing the buffer space requirement, but energy expenditure has not been considered by any of these authors. In [25], an adaptive transmission power selection scheme with deadline-constrained channel scheduling has been proposed for body area networks without multi hop communication.

Unlike the existing work that focuses either on packet deadlines or on energy savings using transmission power control, SAGE aims to minimize packets missing their deadlines (primary goal) *while also minimizing the overall energy expenditure* (secondary goal) in a multi hop edge network, by making use of DRL with a unique reward function. This is crucial in edge networks in which both time and energy are constrained and missing packet deadlines has undesirable consequences. Recently, a cross-layer protocol for joint power control, link scheduling and routing has been proposed for underwater wireless sensor networks in [26]. The authors formulate the problem as a mixed-integer linear programming (MILP) problem which is NP-hard and use a bio-inspired heuristic to minimize the energy expenditure and the end-to-end delay, without packet deadlines being considered. *To the best of our knowledge, no previous approach can give a schedule of both transmission power and time slots to sequentially optimize packets missing their deadlines and the overall energy expenditure for communication in multihop edge networks.*

V. Network Model

In this section, we explain the model of the network considered by us. As in [2], we represent an energy-constrained wireless network as an undirected graph *G=(V,E)*, where each graph vertex v_i represents a network node n_i and each edge e_{ij} represents the wireless link between two nodes n_i and n_j. The notation used by us is given in Table I.

TABLE I
Notation Used

Notation	Meaning
N	No. of nodes in the network
M	No. of flows in the network
C	No. of frequency channels
α	deadline to period ratio
ρ	density (no. of nodes per coverage area) of nodes in the deployment area
$\mathbf{F}$	set of flows in the network
$\mathbf{R_i}$	ordered set of routes of the i^{th} flow
$\overrightarrow{R_i^j}$	j^{th} shortest route of the i^{th} flow
τ_{ij}^k	transmission of the packet generated by n_k from n_i to n_j
T	super frame length (slots)
$\mathcal{T}_t$	set of transmissions queued at time slot T
$\mathcal{T}_t^*$	set of transmissions to be advanced at T
$(\overrightarrow{\mathcal{T}^*}, \overrightarrow{\mathcal{P}})$	transmission power plan for a super frame
$\overrightarrow{s_t}$	state of the system at time slot T
$\pi^*(a\|s)$	optimal policy given by the DRL agent
ϵ_t^i	energy spent by node i by time slot t for data transmission/reception
p_g	No. of packets generated in a super frame
p_d	No. of packets delivered in a super frame
$\widehat{p_T}$	total no. of packets missing their deadlines in a super frame
p_d^t	No. of packets delivered in time slot t

SAGE starts by assuming a TDMA-based MAC scheme with T time slots forming a *super frame*. Our aim is to build conflict-free schedules. Though the radio (e.g., CC2420) may support multiple channels, frequency reuse across the network is not considered by us, since the interference graph is dynamic and hence difficult to build in real-time for most constrained deployments [2]. In the absence of conflicts, the bit error rate (BER) between any two nodes depends on the signal to noise ratio between the nodes. For a given modulation scheme, noise spectral density, bandwidth and receiver sensitivity, the BER increases as the distance between two nodes increases.

For SAGE, we consider that an edge exists between n_i and n_j if it possible to transmit from n_i to n_j with a BER that results in a packet delivery rate of $\approx$100% between two nodes, using transmission power between $\mathcal{P}_{min}$ and $\mathcal{P}_{max} dBm$ (based on the physical and MAC layer standards used). Such an edge is associated with a cost e_{ij}, which is the minimum possible energy to achieve the desired PDR, for given physical and MAC layer parameters. All scenarios consider the same white noise and interference is considered negligible, as TDMA is used without frequency reuse across the network.

Any of the network nodes (except the network controller) may generate data packets. In addition, all nodes are also capable of acting as relay nodes between the source and destination of a data packet. Each packet is sent to its destination in one or more hops. The packet traversal is represented by a series of transmissions τ_{ij}^{k}, where k is the node generating the packet and the packet hop is from n_i to n_j. The network may have up to M data flows in each super frame period. While each transmission is essentially a packet in a particular hop, a flow $\overrightarrow{f_i}$ consists of a series of transmissions of a given data packet from its source to its destination. The aim of SAGE is to give a transmission power plan (the slot and channel in which and power with which each node should transmit). Nodes use this plan to wake up to transmit/receive with the recommended power in their assigned slots. To represent a typical constrained edge network consisting of *things*, we consider both periodic and aperiodic data generation. Nodes may generate data periodically with random, harmonic periods of $2^i, i = \beta_{min}...\beta_{max}$ as suggested in [2]. Each flow is also associated with a constant deadline, which is the maximum number of time slots before which a packet of the flow has to reach its destination after being generated. The time remaining till the deadline is called the relative deadline.The ratio of the deadline of a flow to its period is denoted by $\alpha \in (0, 1)$ and the minimum length of a super frame is T slots, where T is the least common denominator of all the flow periods.

For a given network graph G, $\mathbf{R_i}, i \in 1..M$ is an ordered set of routes from the source to the destination of the i^{th} flow. To calculate $\mathbf{R_i}$, all the routes from the source to the destination of $\overrightarrow{f_i}$ are calculated using a shortest path algorithm and are sorted in the increasing order of path weights. Hence, $\overrightarrow{R_i^j}, j \in 1..\gamma$ is the j^{th} shortest route for $\overrightarrow{f_i}$. Since the edge weights are in terms of the minimum energy required, following the shortest path results in low energy expenditure. However, if not combined with a proper transmission schedule, it may result in large packet delays and missed deadlines. We turn to DRL to solve this problem.

VI. Transmission Power Management with SAGE

Since the savings in energy can come at the cost of missed packets in deadline-aware networks, SAGE's *primary* aim is to maximize the number of packets being delivered within their deadline in a given network, while the *secondary* goal is to minimize the overall energy consumption. Given a network graph G and a set of flows $\mathbf{F}$, the primary objective can be written as:

$$\min_{\pi} \widehat{p_T}, \tag{1}$$

where $\widehat{p_T}$ is the total number of packets (of all flows) missing their deadlines, in a super frame of T slots and π is the policy of the DRL agent. Since this is NP-hard, the DRL agent tries to find the best possible (possibly sub-optimal) solution. When there are multiple such solutions, the agent further picks one of them to achieve the following objective:

$$\min_{\pi} \sum_{i=0}^{N-1} (\epsilon_T^i - \epsilon_0^i), \tag{2}$$

where ϵ_t^i is the energy spent by node i by slot t for data transmission and reception. SAGE uses DRL with a carefully crafted reward function for meeting the above two goals in that order. The resulting model can then be deployed at the central network controller for optimal network control.

Algorithm 1 SAGE

Input: G**,** F **and** $\overrightarrow{\mathcal{R}}$
Output: The transmission power plan $(\overrightarrow{\mathcal{T}^*}, \overrightarrow{\mathcal{P}})$

Compute the optimal policy $\pi^*(a|s)$ using DRL
$t \leftarrow 0$
for $t < T$ **do**
  Using $\pi^*(a|s)$, determine $\mathcal{T}_t^*$ (the transmissions to be advanced)
  $\mathcal{T}^*[t] \leftarrow \mathcal{T}_t^*$
  Using $\pi^*(a|s)$, determine $\mathcal{P}_\tau$, $\forall \tau \in \mathcal{T}_t^*$ (the transmission power for each transmission)
  $\mathcal{P}[t] \leftarrow \mathcal{P}_\tau, \quad \forall \tau \in \mathcal{T}_t^*$
  t++
end for
return $(\overrightarrow{\mathcal{T}^*}, \overrightarrow{\mathcal{P}})$

The transmission power to be used by each network node at each time slot in a super frame (for a given network scenario with the network graph G and the set of flows $\mathbf{F}$) is determined using the following steps:

- *Step 1:* A list $\overrightarrow{\mathcal{R}}$ of the top γ routes (determined greedily in terms of power needed to transmit a packet at each hop) for all flows (each i^{th} flow's routes are denoted by $\mathbf{R_i}$) is determined. This can be done using any shortest path algorithm such as Dijkstra's algorithm.
- *Step 2:* Given G, F and $\overrightarrow{\mathcal{R}}$, the DRL agent uses the PPO algorithm as shown in Figure 1, to find the optimal policy for a super frame. This is used to determine the power and time slot plan (as shown in Algorithm 1), which is sent by the edge network controller to the edge network nodes.
- *Step 3:* Steps 1 and 2 are repeated periodically (e.g., whenever the network controller detects performance deterioration in terms of the packets missing their deadlines with the current transmission plan), to cater to changes in the network conditions.

Since the most important part of SAGE is the learning of the optimal policy by the DRL agent, we describe the DRL agent design below.

A. State Space of the DRL Agent

SAGE uses a DRL agent (a program at the network controller) that observes the current state $\overrightarrow{s_t}$ of the network at each time step (in our case, this corresponds to a time slot in a super frame) t, takes an action a as per the policy, gets a reward r and moves to a new state $\overrightarrow{s_{t+1}}$. What constitutes the state for a DRL agent depends on several factors including

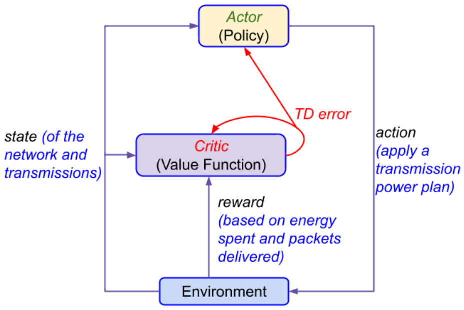

Fig. 1. Actor-critic PPO for SAGE

the agent's environment and goal. For the agent in SAGE, we consider a network with a constant (for the duration of a super frame) network topology. Based on its goals, SAGE considers the $\overrightarrow{s_t}$ at time slot t to be a vector consisting of the following:

- The (flattened) adjacency matrix (NxN) of a given network graph,
- a vector of length M consisting of the nodes at which the $\mathcal{T}_t$ transmissions (one per flow) are queued,
- a vector of length M with the relative deadlines of each of the M flows at time t and
- a vector of length N consisting of the energy spent by each of the N nodes in the current super frame, up to time slot t.

The total length of $\overrightarrow{s_t}$ is $(2 * M) + ((N + 1) * N)$.

B. Action Space of the DRL Agent

At each time step t, the DRL agent in SAGE takes an action a from an action space $\mathbf{A}$ and observes the reward obtained. The action space $\mathbf{A}$ should be chosen such that the agent explores enough of the search space to find an optimal solution, with minimum possible convergence time. After training using PPO, the policy gives the optimal action for that network at each time slot. An action a in SAGE encodes two choices – the transmissions to advance in that time slot and the power to be used by each transmitting node.

1) Choosing the transmissions to advance in slot t: Let $\mathcal{T}_t$ be the set of transmissions belonging to various flows queued at different nodes in the network at time slot t. Of these, C (which is the number of channels) $non-conflicting$ transmissions may be picked to be forwarded along each channel. Two transmissions τ_{ij} and τ_{uv} are non-conflicting $\iff$ {i,j} $\cap$ {u,v}=ϕ. Since deadline-aware scheduling is NP-hard, we leverage on two widely-used scheduling heuristics for picking these transmissions:

- transmissions with the minimum remaining time (in slots) to deadline are picked (the Earliest Deadline First or EDF heuristic) or
- transmissions with the minimum average laxity are picked. The average laxity is defined as the minimum remaining time in slots minus the average number of hops (along different routes) to the destination.

During training, the DRL agent uses one of these heuristics (based on its policy) and picks $\mathcal{T}_t^*$ transmissions to advance at time slot t.

2) Choosing the transmission power for slot t: The DRL agent should also choose the transmission power to be used by each transmitting node. During training, the DRL agent tries one of the top γ routes (in terms of overall energy spent along the path) for each flow. Hence, the agent first picks a transmission to be scheduled and then chooses among the γ neighboring nodes (for that flow) to forward it. This choice of next hop, in turn, determines the transmission power to be used by the node, which is the minimum transmission power to achieve the desired PDR at the next hop node. Hence, the DRL agent can decide the power to be used by each transmitting node $\mathcal{P}_\tau \forall \tau \in \mathcal{T}_t^*$.

C. Reward Function of the DRL Agent

The design of the reward obtained at each step is crucial for achieving the desired goal in DRL. As discussed in Section I, SAGE aims at minimizing the number of packets missing their deadlines and the total energy spent in a TDMA super frame. To achieve this, the reward rew_t obtained at each time step t is split into three parts –

1) rew_1^t, which is the reward obtained at each time step t for delivering packets to their destinations,
2) rew_2, which is a negative reward obtained at the end of the super frame (at slot $(T-1)$) for each packet that is delivered to its destination, but has missed its deadlines in a super frame and
3) rew_3, which is also a negative reward obtained at the end of the super frame for each packet that is not delivered to its destination within the super frame in which it is generated.

In the absence of rew_3, the DRL agent chooses a transmission power schedule that minimizes the overall energy spent by not transmitting any packets at all. The goal of minimizing the number of packets missing their deadlines is achieved by penalizing the agent with a negative reward rew_2 for every packet that misses its deadline.

For fixing rew_1^t, which is the reward for delivering a packet within its deadline to the destination, we first give an expression for the maximum total energy that may be spent to deliver C packets (along each of the C channels) to the destination in the network. This is used for **normalizing** the reward and denoted by E_{max}, where –

$$E_{max} = max_{i,j \in 0..(N-1)}(e_{ij}) * hops_{max} * C, \tag{3}$$

where $hops_{max}$ is the maximum number of hops from the source to the destination for any flow along any route. The number of packets that reach their destination in a given time slot is p_d^t, which may actually be less than or equal to C. The reward at a time step t is:

$$rew_1^t = \frac{E_{max}}{C} * p_d^t - \sum_{i=0}^{N-1} (\epsilon_{t-1}^i - \epsilon_t^i)/E_{max} \tag{4}$$

This helps *minimize* the overall energy spent. Hence, if no packets are delivered to their destination in time slot t, the agent gets a negative reward proportional to the the energy spent by all the nodes in advancing transmissions towards their destinations. When some packets reach their destinations in t, the agent gets a positive reward.

The total reward obtained at each step is:

$$rew_t = \begin{cases} rew_1^t, & \text{if } t \in 0..T-2 \\ rew_1^t + rew_2 * \widehat{p} + rew_3 * (p_g - p_d), & \text{if } t = T-1 \end{cases}$$

Here, p_g and p_d are the total number of packets generated and delivered during a super frame respectively. While $\text{rew}_2 * \widehat{p}$ penalizes the agent for packets delivered beyond their deadlines, $\text{rew}_3 * (p_g - p_d)$ penalizes the agent for packets that not delivered in the super frame in which they are generated.

This reward design is crucial for SAGE's DRL agent to meet its goals, which are to minimize the number of packets missing their deadlines and the total energy consumption in a super frame, *in that order*.

VII. Results and Discussion

For evaluating the performance of SAGE, we built a custom network simulator based on NS3, integrated with a DRL agent that runs the PPO algorithm [12] using OpenAI Gym [27]. The MAC layer model considered is IEEE 802.15.4 (a popular choice for TDMA-based low rate, wireless personal area networks) with log-distance propagation model and white noise. Further, the receiver sensitivity was set to -106.58 dBm (the maximum value specified by the standard [28], with O-QPSK and 250kps) and a packet size of 20 bytes was considered. The PPO algorithm had an actor and a critic, each with two hidden layers. The hidden layers for both the actor and the critic networks were of 256x256 neurons with the *tanh* activation function. The minimum transmission power to be used by a node was based on the distance to the nearest neighbor such that a PDR of 100% can be obtained with a confidence of 95%. The rest of the parameters (defined in [12]) used for training the PPO agent are given in Table II.

TABLE II
DRL Agent Training Parameters

Parameter	Value Used
entropy-coefficient	0.015
value-loss-coefficient	0.5
reward discount factor	0.9999
mini-batch size	8
PPO epochs	4
clip parameter	0.2
learning rate	$1 * 10^{-5}$ to $7 * 10^{-5}$

Several network scenarios with different parameters (typical of IEEE 802.15.4 applications such as equipment monitoring, smart grid, smart homes etc.) were generated for the evaluation. By scenario, we refer to a particular network topology G, with a particular set of flows $\mathbf{F}$. For the sake of evaluation, we consider that flows are generated by random nodes and terminate at the base station, which is typically how most TDMA-based standards for IoT edge networks work. However, SAGE can be used for communication between any two devices in the network. A set of scenarios consists of different network topologies (G), random flows and the same values of the number of nodes N, the number of flows M, the number of channels C, the super frame length T, the node density ρ and the deadline to period ratio range α. For each set of (8 to 10 different) scenarios with the same parameters, we trained the DRL agent to give the packet transmission schedule and the power to be used at each step, using the top 5 routes ($\gamma = 5$) during training. Two other schemes were considered for comparison:

1) a simple baseline scheme which uses the earliest deadline first scheduling heuristic and chooses minimum transmission power at each hop from the source to the destination (the "Greedy" strategy)
2) RECCE [4], which is a DRL-based joint routing and scheduling scheme that focuses on deadline-aware packet delivery and uses fixed transmission power.

A. Effect of the Number of Nodes

To study the effect of different number of nodes, we evaluated SAGE for three different sets of scenarios. Each set had the same number of channels ($C = 2$), node density ($\rho = 4$), super frame length ($T = 64$), deadline to period ratio range ($\alpha \in (0.2, 1.0)$) and fraction of nodes generating data flows ($M/N = 0.2$). The number of nodes N considered was 20, 40 and 60. Since the deadlines are quite lenient, all three schemes considered deliver all packets within their deadlines (*0% packets miss their deadlines*).

Figure 2 depicts the Cumulative Distribution Function (CDF, across the different random scenarios in a set) of the percentage energy saving compared to the Greedy and RECCE schemes. It can be seen that while RECCE consumes slightly more average energy per super frame than the Greedy scheme, SAGE saves up to 26.3% energy per super frame on the average, with the energy savings going up to 48% for some scenarios (for 20 nodes). SAGE saves more energy compared to the baseline schemes when the number of nodes and flows is lesser. This is because with lesser number of nodes, the average number of hops on each path is lesser, leading to lesser average delay. This gives better scope for SAGE to explore alternate routes with lesser energy consumption, while delivering the packets within their deadlines.

B. Effect of the Number of Flows

We next explore the effect of the number of flows (M — 8, 16 and 24) for scenarios which are similar in terms of the rest of the parameters ($N = 40$, $C = 2$, $T = 64$, node density $\rho = 4$ and deadline to period ratio $\alpha \in (0.2, 1.0)$). From the results depicted in Figure 3, it can be seen that the savings in energy are better with lesser number of flows (an average of 18% for 8 flows and 8% for 24 flows, compared to the Greedy strategy) and all packets are delivered within their deadlines in all the three schemes. This is so because as the number of

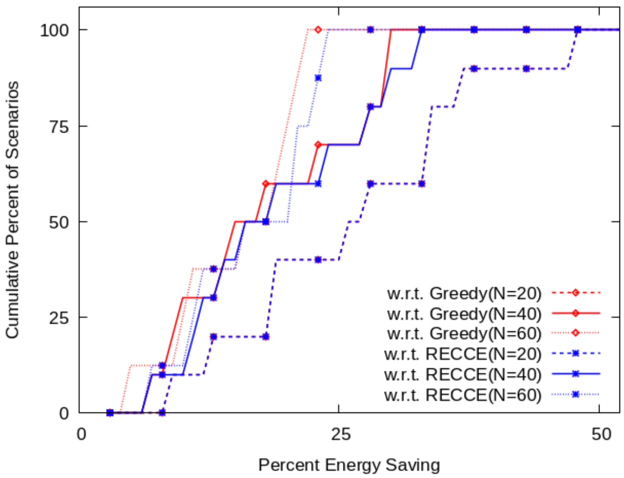

Fig. 2. Effect of the Number of Nodes(N) on the Energy Spent; No packets miss their deadlines in any case considered

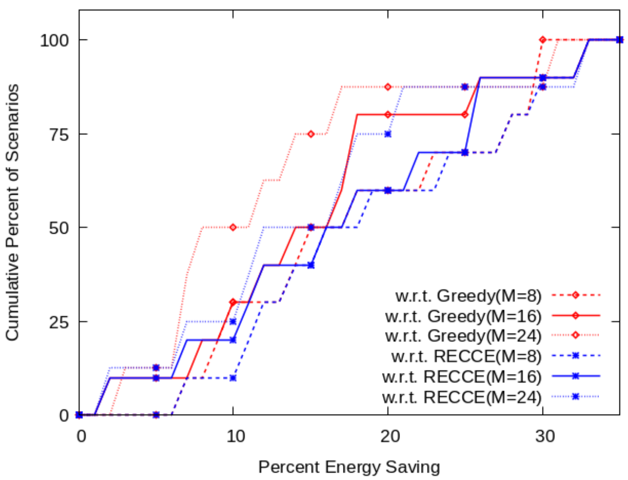

Fig. 3. Effect of the Number of Flows(M) on the Energy Spent; No packets miss their deadlines in any case considered

flows is lesser, the queuing delay is small and there is more scope to take routes that are longer in terms of the number of hops, but are better in terms of the energy savings, without missing the deadline.

C. Effect of the Node Density

For a given number of flows M, the energy savings depend on two factors related to the network topology:

- the number of alternate paths available from the source to the destination and
- the variance in distance between pairs of nodes (variance of the edge weights e_{ij}) along these paths.

These two factors, in turn, depend on the node density. As the node density increases, so does the path diversity, which may lead to better energy conservation. For evaluating the effect of the node density on the performance of SAGE, we generated different network scenarios with $N = 40$, $M = 16$, $T = 64$ slots, $C = 2$ channels and a deadline to period ratio $\alpha \in (0.2, 1.0)$. The node density ρ was varied between 2 and 4. From Figure 4 it can be observed that as the node density increases, the energy savings increase compared to the Greedy scheme(6%, 11% and 16% for ρ=2,3 and 4 respectively). This is because with denser networks, it is possible to find alternate routes that consume lesser energy, while satisfying the deadline constraints. Since this is a set of scenarios with longer deadlines, both RECCE and SAGE deliver all packets within their deadlines.

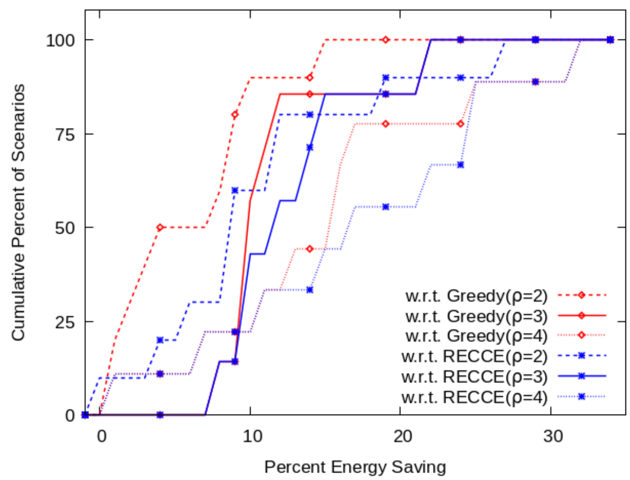

Fig. 4. Effect of the Node Density(ρ) on the Energy Spent; No packets miss their deadlines in any case considered

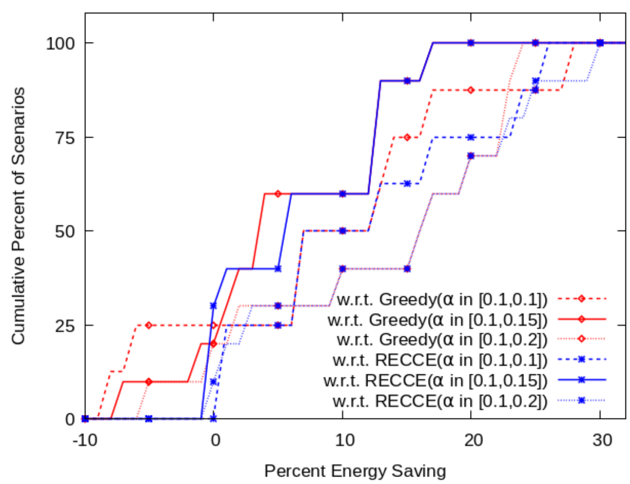

Fig. 5. Effect of the Deadlines on the Energy Spent; Greedy has 40%, 25% and 3.75% packets missing deadlines for α = $[0.1, 0.1]$, $[0.1, 0.15]$ and $[0.1, 0.2]$ respectively. RECCE and SAGE reduce this to 39%, 20% and 2.5% respectively at the cost of more energy expenditure (-ve savings).

D. Effect of the Deadlines

Since the goal of SAGE is to minimize the number of packets missing their deadlines and then the average energy consumed in a super frame, the deadlines of the flows have an impact on the performance. To illustrate this, we considered a set of network scenarios with with $N = 40$ nodes with node density $\rho = 4$, $M = 8$ flows, $T = 64$ slots and

$C = 2$ channels. We considered three different ranges of $\alpha = [0.1, 0.1], [0.1, 0.15]$ and $[0.1, 0.2]$. These scenarios are representative of applications with very low delay requirement. Both RECCE and SAGE do equally well in terms of missed packets (39%, 20% and 2.5% respectively for $\alpha \in [0.1, 0.1], [0.1, 0.15]$ and $[0.1, 0.2]$ respectively) compared to the Greedy scheme (40%, 25% and 3.75% respectively). Further, Figure 5 shows that SAGE can result in better energy savings (upto 12%), while RECCE actually consumes slightly more energy compared to the Greedy scheme.

E. Effect of the Number of Channels

Most of current day's radios support multiple channels for communication. This allows multiple, non-interfering, concurrent transmissions in a network. One way to use multiple channels is reuse the channel at non-interfering nodes, but most standards for wireless, constrained networks such as WirelessHART and 6TiSCH do not encourage frequency reuse. A different way to make use of multichannel radios is Time-Slotted Channel Hopping as in 6TiSCH.

To study the effect of multiple channels on the performance of SAGE, we changed the number of channels from 1 to 4, without channel reuse across the network or channel hopping, but SAGE can easily be extended to support this. For this evaluation, we considered a set of scenarios with $N = 40$, $M = 8$, node density $\rho = 4$ and the deadline to period ratio α between 0.1 and 0.2, which is representative of applications with very low delay requirements.

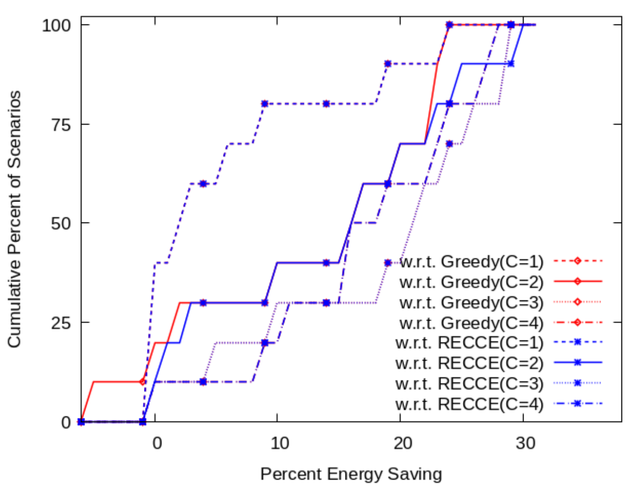

Fig. 6. Effect of the Number of Channels(C) on the Energy Spent; For one channel, Greedy and RECCE both have 49% packets missing deadlines. SAGE reduces this to 46% at the cost of more energy expenditure (-ve savings). For two channels, Greedy has 3.75% and both RECCE and SAGE have 2% packets missing their deadlines. For 3 and 4 channels, all schemes deliver all packets within their deadlines.

For a single channel, 49% of the packets miss their deadlines with both the Greedy and RECCE schemes while SAGE brings this down to 46%. In addition, SAGE saves 5% of the energy consumed (on the average) when compared to the Greedy strategy. As the number of channels increases to 2, both RECCE and SAGE reduce the percentage of packets missing their deadlines to 2%, as compared to the Greedy strategy (3.75%). In this case, SAGE results in average energy savings of nearly 16% over the Greedy strategy. This trend continues for 3 channels, where the number of packets missing their deadlines reduces further to zero with all three schemes and SAGE results in much better energy savings.

F. Convergence of the Training Phase

The state space of the DRL agent in SAGE is (2 * M) + ((N + 1) * N), while SAGE's action space is γ^C, which can be quite large and may result in slow convergence of the learning process. However, the number of channels for most radios does not exceed sixteen. In addition, IoT standards such as 6TiSCH use channel hopping and the number of channels where *simultaneous* transmissions can happen is even lower than the number of channels supported by the radio. These facts make SAGE feasible in practice.

Moreover, our evaluation shows that as the number of channels increases, the time taken to converge increases due to increase in the action space (≈3000 training episodes for 2 channels vs ≈4200 episodes for 5 channels). The convergence time does not increase exponentially (even if the action space does) because of SAGE's usage of the PPO algorithm (which is well-suited for large state and action spaces) and a careful choice of the hyper parameters used therein.

VIII. Conclusions and Future Directions

We propose SAGE, a DRL-based transmission power planning strategy for deadline-aware, multihop IoT edge networks. The DRL agent design used in SAGE reduces the number of packets missing their deadlines compared to the baseline strategies, while minimizing the overall energy spent in the network in a super frame duration. Evaluation of SAGE for different network parameters shows that up to 48% energy savings are possible (for $N = 20$), depending on the network scenario.

Our evaluation also leads to some interesting findings. Exploring different transmission power settings and time schedules can reduce the percent of packets missing the deadlines, but node conflicts impose a limit on this. For any TDMA scheme aiming at near-deterministic end-to-end packet delay, node conflicts must be avoided by scheduling transmissions in conflicting nodes at different time slots and/or channels and SAGE does the same. SAGE can easily be extended to use a MAC scheme such as Code Division Multiple Access (CDMA), but this may incur more bandwidth and equipment cost.

In networks with short packet deadlines or low node density, there may not be much scope to find alternate routes that reduce energy consumption and packets missing deadlines. In conclusion, while SAGE performs equally well or better than the baseline schemes for all sets of network parameters, *the energy savings are most substantial for networks with relatively longer deadlines, fewer nodes and flows, more channels and higher node density*. This is due to the fact that such networks offer more flexibility in terms of choosing alternate routes that save energy while meeting packet deadlines.

Since SAGE is built on DRL which requires extensive training, it is tailored for networks where there can be a short pre-deployment phase when the agent can be trained. Alternatively, the DRL agent may be trained offline and the policy can be deployed in the network controller. Since the policy may vary for different network scenarios, SAGE is most suited for applications where the scenario does not change often, such as equipment monitoring in industries, smart homes, smart grid etc.

With deadline-aware applications in view, SAGE aims to sequentially optimize the two objectives, giving secondary importance to energy. In reality, a single network may have heterogeneous application requirements, with some applications requiring strict deadline-based delivery while others having more lenient deadline requirements. Additionally, some networks may have more stringent energy requirements than others. In the future, we intend to explore an adaptive scheme that can switch between either in-time packet delivery or minimal energy expenditure being the primary goal, depending on the application requirements and/or network conditions.

REFERENCES

[1] X. Jin, A. Saifullah, C. Lu, and P. Zeng, "Real-time scheduling for event-triggered and time-triggered flows in industrial wireless sensor-actuator networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1684–1692.

[2] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for wirelesshart networks," in *2010 31st IEEE Real-Time Systems Symposium*. IEEE, 2010, pp. 150–159.

[3] S. Chilukuri and A. Sahoo, "Delay-aware tdma scheduling for multi-hop wireless networks," in *Proceedings of the 16th International Conference on Distributed Computing and Networking*, ser. ICDCN '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2684464.2684493

[4] S. Chilukuri and D. Pesch, "Recce: Deep reinforcement learning for joint routing and scheduling in time-constrained wireless networks," *IEEE Access*, vol. 9, pp. 132 053–132 063, 2021.

[5] S. Chilukuri, G. Piao, D. Lugones, and D. Pesch, "Deadline-aware tdma scheduling for multihop networks using reinforcement learning," in *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 2021, pp. 1–9.

[6] V. P. Modekurthy, A. Saifullah, and S. Madria, "Distributedhart: A distributed real-time scheduling system for wirelesshart networks," in *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2019, pp. 216–227.

[7] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal next-generation wireless networks," *arXiv preprint arXiv:1902.01946*, 2019.

[8] E. Holmstedt, *Power Effect Control for IEEE 802.15.4 Based Wireless Devices*. Uppasala University, 2012. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:564052/FULLTEXT02.pdf

[9] V. G. Douros and G. C. Polyzos, "Review of some fundamental approaches for power control in wireless networks," *Computer Communications*, vol. 34, no. 13, pp. 1580–1592, 2011.

[10] S. Alizadeh and H. Aydın, "Energy efficient communication protocol for 6tisch networks," in *2019 Seventh International Conference on Digital Information Processing and Communications (ICDIPC)*, 2019, pp. 63–67.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[13] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, Z. Meng, and M. Alizadeh, "Learning scheduling algorithms for data processing clusters," in *Proceedings of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 270–288. [Online]. Available: https://doi.org/10.1145/3341302.3342080

[14] M. Elsisi, M. Amer, A. Dababat, and C.-L. Su, "A comprehensive review of machine learning and iot solutions for demand side energy management, conservation, and resilient operation," *Energy*, vol. 281, p. 128256, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036054422301650X

[15] S. F. A. Shah, M. Iqbal, Z. Aziz, T. A. Rana, A. Khalid, Y.-N. Cheah, and M. Arif, "The role of machine learning and the internet of things in smart buildings for energy efficiency," *Applied Sciences*, vol. 12, no. 15, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7882

[16] I. Kim, B. Kim, and D. Sidorov, "Machine learning for energy systems optimization," *Energies*, vol. 15, no. 11, 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/11/4116

[17] A. Bagwari, J. Logeshwaran, K. Usha, K. Raju, M. H. Alsharif, P. Uthansakul, and M. Uthansakul, "An enhanced energy optimization model for industrial wireless sensor networks using machine learning," *IEEE Access*, vol. 11, pp. 96 343–96 362, 2023.

[18] M. M. El-Gayar and M. N. Ajour, "Resource allocation in uav-enabled noma networks for enhanced six-g communications systems," *Electronics*, vol. 12, no. 24, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/24/5033

[19] A. Subramanian and A. Sayed, "Joint rate and power control algorithms for wireless networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4204–4214, 2005.

[20] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, S. Son, and G. J. Pappas, "Atpc: Adaptive transmission power control for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 12, no. 1, mar 2016. [Online]. Available: https://doi.org/10.1145/2746342

[21] A. Paul, M. Akar, M. Safonov, and U. Mitra, "Adaptive power control for wireless networks using multiple controllers and switching," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1212–1218, 2005.

[22] R. Sunder, U. K. Lilhore, A. K. Rai, E. Ghith, M. Tlija, S. Simaiya, and A. H. Majeed, "Smartapm framework for adaptive power management in wearable devices using deep reinforcement learning," *Scientific Reports*, vol. 15, no. 1, p. 6911, 2025.

[23] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Communications Surveys Tutorials*, vol. 23, no. 4, pp. 2131–2165, 2021.

[24] S. Chilukuri, A. Gupta, and H. S. Sai Pulamolu, "Smarthart: A priority-aware scheduling and routing scheme for iiot networks using deep reinforcement learning," in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2023, pp. 452–456.

[25] A. Arghavani, H. Zhang, Z. Huang, and Y. Chen, "Power-adaptive communication with channel-aware transmission scheduling in wbans," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 087–16 102, 2024.

[26] Y. Yuan, X. Zhuo, M. Liu, Y. Wei, and F. Qu, "A centralized cross-layer protocol for joint power control, link scheduling, and routing in uwsns," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12 823–12 833, 2024.

[27] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[28] "Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011.