Reinforcement Learning for Joint Transmit-Sleep Scheduling in Energy-harvesting Wireless Sensor Networks

Hrishikesh Dutta, Amit Kumar Bhuyan, and Subir Biswas
Michigan State University, East Lansing, USA
duttahr1@msu.edu, bhuyanam@msu.edu, sbiswas@egr.msu.edu

***Abstract* — This paper proposes an interactive multi-agent Reinforcement Learning (RL) framework for joint transmit-sleep scheduling in energy-harvesting wireless sensor networks. The scheduling problem is modeled as a Markov Decision Process (MDP) and solved using temporal difference Reinforcement Learning approach. The online learning abilities of RL make the nodes learn a scheduling policy for transmission and sleep so as to minimize the MAC layer packet loss, while maintaining a stable packet queue, in the presence of limited energy budget and heterogeneous traffic patterns. This is accomplished by the joint coordination of two interactive RL agents launched per node to make scheduling decisions. Each node learns the scheduling policy independently and without explicit information sharing. The decentralized nature of the proposed architecture makes the model computationally efficient, scalable with network size, and suitable for resource constrained Sensor and IoT networks. With simulation experiments, the proposed approach is validated for different traffic and network conditions and compared against an existing hybrid sleep-scheduling mechanism.**

Index Terms* — *Medium Access Control, Energy Harvesting, Reinforcement Learning, Sleep Scheduling

I. INTRODUCTION

This paper presents a Reinforcement Learning-framework for joint transmit-sleep scheduling in energy-harvesting wireless networks with ultra-thin energy budgets. Efficient decisions on transmit-sleep scheduling in energy-constrained wireless networks is important from two perspectives. First, to reduce the network service disruption duration because of energy shortage and second, to maintain a reliable network performance in terms of throughput and delay. Traditionally, such schedules are often pre-programmed in the wireless nodes and as such they often fail to deliver desired performance in a situation-specific manner. For example, these sleep scheduling policies are oblivious to network traffic patterns and heterogeneities which lead to wastage of precious networking resources, including energy. In addition, for networks with energy harvesting capabilities, where the energy availability depends on temporal and geographical characteristics, these policies do not allow nodes to react according to the spatiotemporal energy profile. Shortcomings are usually manifested in the form of not being able to maintain the desired balance between network performance and network lifetime. Hence, in this work, an online learning-based paradigm is proposed that allows the nodes to learn a scheduling policy so as to overcome the above limitations.

There are works [1-4] that deal with sleep scheduling in networks with energy constraints. The authors of [1] use a game-theoretic approach to find the sleep-scheduling policy for solar powered sensor networks. In addition to the fact that these policies are static with respect to network traffic and topologies, they also do not consider transmission scheduling decisions. As will be shown in this work, transmission scheduling plays a significant role in maintaining network performance in energy-harvested networks. There are RL-based approaches [3-8] for sleep scheduling in energy harvested networks. Besides the above limitations arising from not considering policies for transmission strategies, these often rely on a centralized arbitrator for learning the scheduling policies. Centralized learning, apart from being computationally inefficient and creating burden on the central server, it also comes with an additional cost of requirement of extra bandwidth and energy consumption for downloading the learnt policies from the server. Moreover, performance of the learnt policies heavily depend on the reliability of information collected from the sensor nodes over a possibly error-prone channel.

In this work, we provide a decentralized RL-based approach for joint sleep-transmission scheduling in wireless networks. The scheduling problem is modeled as a Markov Decision Process (MDP) and then solved using temporal difference learning. Each energy-harvesting node learns on-the-fly to take judicious transmit-sleep decisions so that the available resources can be efficiently utilized to improve network performance. This is accomplished by the joint coordination of two interactive RL agents launched per node to make scheduling decisions. The interactive cooperative behavior of the learning agents helps the network achieve throughput higher than the existing sleep schedulers. To be noted that the proposed framework is distributed in that each node learns its scheduling policy independently and without explicit communication with other nodes. In addition to the benefits of decentralized learning mentioned above, this makes the model scalable with network size. From an application standpoint, the framework is suited to resource-constrained embedded networks of IoTs and sensors that are powered from ambient harvested energy and have ultra-low energy budgets.

The paper has the following scopes and contributions. First, an RL-based framework is developed with two interactive agents for making joint transmit-sleep scheduling decisions in energy-harvested wireless networks. The two agents cooperatively learn policies for judicious energy management for sustainable communication in wireless networks. Second, the proposed approach is decentralized such that each node learns its scheduling policy independently. Third, the proposed framework is shown to be scalable with network size. Fourth, with extensive simulations, a detailed characterization of the proposed architecture is done for different traffic patterns and compared against known scheduling policies.

II. RELATED WORK

There are works that develops sleep scheduling and resource allocation policies for establishing efficient communication in power-constrained energy harvesting wireless networks. In [1], the authors develop scheduling policies for solar powered wireless networks. The proposed approach considers node battery status, harvested energy, queue status, and channel characteristics for sleep-awake decisions. However, it does not

consider transmission-scheduling decisions into account. This would require the network designer to adjust policy parameters depending on network traffic conditions.

The paper [2] focuses on helping sensor nodes and a network coordinator to save energy using transmission power control, learnt using Q-learning. Learning here is solely executed by the centralized coordinator and is validated for single-hop networks, thus limiting its applicability. The works in [3-6] develop learning-driven approaches for sleep scheduling in sensor networks. These mechanisms mainly deal with finding a suitable sleep schedule for sensor nodes using centralized learning techniques. These mechanisms are mainly suitable for star topology networks where the coordinator evaluates the strategy for the nodes. Also, there is a significant control packet overhead associated with these techniques for periodically transmitting node information to the coordinator. Similar limitations of centralized computation exist in the RL-based energy management solutions in [7, 8] for rechargeable and energy-harvested networks.

In [9], the authors propose a resource scheduling approach to improve transmission reliability of emergency-critical sensor networks. Here, to reduce complexity, the authors employed deep RL to solve an optimization problem at the node level. However, this mechanism too appears to be restricted only to single-hop networks. The authors in [10] propose an RL-based solution that helps resource-constrained nodes to enhance their performance by saving battery power and maintaining the quality of transmitted data. Similar to the previous scheduling-based methods, this paper does not consider optimizing the transmission scheduling decisions, which play a significant role in efficient energy management.

In the work presented in this paper, a decentralized RL-based solution is proposed for energy management in resource-constrained wireless networks without the need for a central server. This solution uses an interactive two-agent system that jointly takes decisions on transmission and sleep scheduling for reliable communication in energy-harvesting networks.

III. System Model

A. Network and Traffic Model

In this work, we consider multi point-to-point energy-harvesting networks. As shown in the network in Fig. 1, the wireless sensor nodes send data to a wirelessly connected base station using fixed size packets. Time is slotted and the MAC frames are of fixed size, which is dimensioned *a priori* based in the degree of the network topology. MAC slot allocation is done based on TDMA, since the network is time synchronized.

Application layer packet generation at source node follows a Poisson distribution with packet generation rate λ packet per frame (ppf). Each node maintains an $\mathrm{M/G/1/K}$ queue, where the Poisson distributed queue arrival rate is governed by λ, and the queue service rate is determined by the actuated transmission-sleep policy. The latter will be learnt using the learning mechanism presented in Section IV.

B. Energy Harvesting and Consumption Model

With a long-term goal of making the framework generalized by considering different kinds of energy harvesting sources, in this work, we start with solar energy harvesting. A 2-state Markov Model is used for simulating solar energy harvesting. The two states of the Markov model are (1) low radiation state, where sunlight is blocked by clouds and hence, radiation is not enough to charge the battery and (2) high radiation state, where there is direct sunlight and is sufficient to charge the battery. This is represented by the transition probability matrix **R**, where state 1, 2 represent high and low radiation states respectively:

$$\mathbf{R} = \begin{bmatrix} R_{1,1} & R_{1,2} \\ R_{2,1} & R_{2,2} \end{bmatrix} \quad (1)$$

Assuming that the cloud size is exponentially distributed with mean 'C' m and the wind speed is w_s m/s, the elements of matrix **R** can be obtained using the analytical model in [4]:

$$R_{1,2} \approx \left(\frac{1}{\mu_s}\right) t \times \mathrm{e}^{\left(\frac{1}{\mu_s}\right)t},\ R_{2,1} \approx \left(\frac{1}{\mu_c}\right) t \times \mathrm{e}^{\left(\frac{1}{\mu_c}\right)t},$$

$$R_{1,1} = 1 - R_{1,2} \text{ and } R_{2,2} = 1 - R_{2,1}$$

Here, $\mu_c = \frac{C}{w_s}$, $\mu_s = \frac{C \times (1-P_c)}{w_s(P_c)}$ are the average time for which the radiation is low and high respectively; P_c is the probability of solar radiation in low state and t is the length of a time frame. Each node has a battery capacity of B units, where a packet transmission consumes one unit of battery. Thus, the battery status at time t is given by $b \in \{0,1,2,\dots,B\}$, with battery charging probability P_{charge}. When $b = 0$, battery is completely depleted and needs recharging. If $b = B$, the battery is fully charged and, the recharging circuitry is turned off.

The energy consumption model in [11] is considered for the radio hardware energy dissipation. In this model, power consumed while transmitting and receiving a packet are given by Eqs. (2) and (3), where P_{tx} represents the transmission power of the transmitter with amplifier inefficiency factor α_T and P_{ct} is the circuitry power consumption, which is a constant depending on specific transmitter.

$$P_T = (1 + \alpha_T) \times P_{tx} + P_{ct} \quad (2)$$

$$P_R = P_{ct} \quad (3)$$

IV. Reinforcement Learning for Joint Sleep-Transmission Scheduling

The high-level objective of the proposed framework is to make the energy-harvesting wireless nodes learn an efficient transmit/sleep/listen scheduling policy. The desired behavior of the learnt schedule is that the available energy is judiciously managed so as to minimize the service disruption resulting from energy shortage. To be noted that there has to be a right balance between sleep and awake decisions. Sleeping, on one hand, can reduce a node's energy consumption, and hence increase the network lifetime. On the other hand, excessive sleeping can lead to missed packet receptions. Similarly, an efficient transmission scheduling is important to strike the right balance between minimizing energy usage and reducing the packet drops resulting from an unstable queue.

This joint scheduling behavior is achieved by the coordination of two RL agents per node, namely, Sleep Scheduling agent and Transmission Scheduling agent. Fig. 1 shows a high-level working model of the RL-based architecture. The role of the Sleep Scheduling agent is to learn policies for efficient decisions for turning the transceiver on and off. The role of the Transmission Scheduling agent is to make decisions on transmission scheduling when the sleep scheduling agent decides to turn the transceiver on. Note that both these agents share the same reward function which is computed from the RL observable variables including the harvested energy and network performance parameters. The state definitions for these

two agents, however, are different and independent. As indicated by the dotted arrows in the figure, the state for the transmission scheduling agent is perceived directly from the network observables, whereas the state for the sleep scheduling agent is dependent on the RL-policies of the transmission scheduler. The common aim of both these agents is to judiciously manage the available limited energy resources for maintaining sustainable communication in the network. Details of these learning agents are given below.

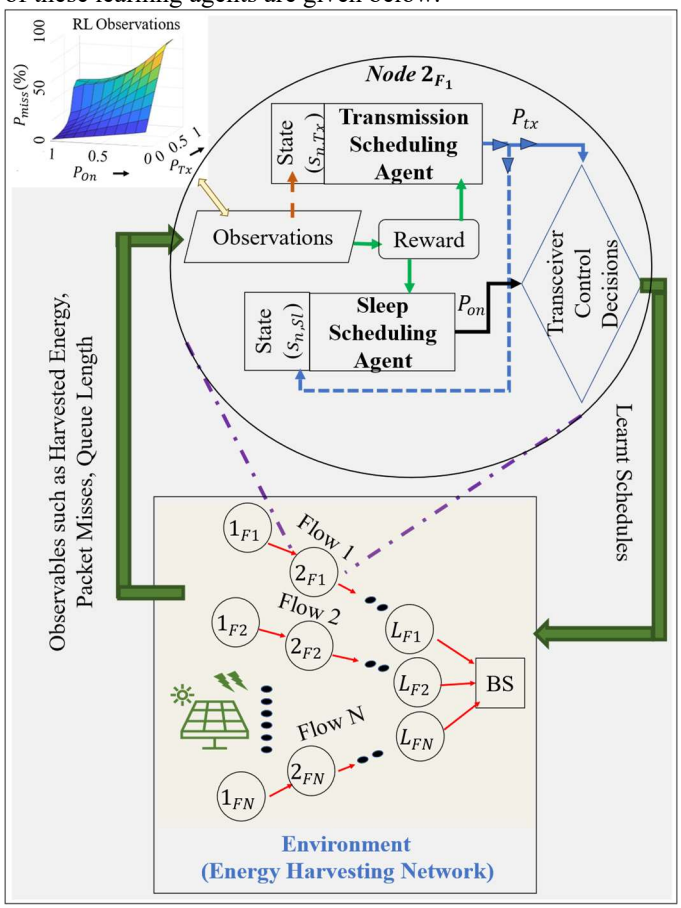


Fig.1 Proposed Framework and the network model: Node N_{F_K} is indexed such that N is the hop count of the node from source 1_{F_K} in flow K

Transmission Scheduling Agent: Suitable transmission decisions are important, because, if the transmission rate is low, it leads to building up queues, which will eventually manifest as an increase in packet drops due to full MAC packet queue. On the other hand, high transmission rates affect energy consumption which gives rise to high missed packet receptions due to energy shortage. As a result, an efficient transmission scheduling policy is required, which will depend on the temporal dynamics and characteristics (depending on the geography, time of the day etc.) of energy harvesting. This is achieved by a transmission scheduling RL agent, that schedules packet transmission so that available harvested energy is efficiently used while maintaining reliable communication.

The MDP action space ($\mathcal{A}_T$) for this agent is defined by the probability of transmitting a packet (P_{tx}) in the queue. To keep the action space discrete, this probability is quantized into $|\mathcal{A}_T|$ discrete values in the range $[0, 1]$, where $|\mathcal{A}_T|$ denotes the cardinality of the action space. These actions are selected following a learning policy in an RL decision epoch which is set to a duration of h frames.

The state space ($\mathcal{S}_T$) as perceived by the agent is represented by the energy influx to the node resulting from harvesting. The RL state at a decision epoch is given by the energy harvested at that epoch. Similar to the packet transmission probability, the harvested energy is also discretized into $|\mathcal{S}_T|$ distinct ranges. Formally, the state as perceived by an agent for node n at an epoch t is defined as $s_{n,Tx}(t) = g(\frac{1}{h}\sum_{\tau=0}^{h} E_{in,n}(t-\tau))$, where $E_{in,n}(k)$ is the energy harvested in time frame k by node n and $g(x)$ is the function for quantization as defined by:

$$g(x) = \begin{cases} \mathcal{s}_T, & if -0.5 \leq \mathcal{s}_T - 0.5 \leq 10x < \mathcal{s}_T + 0.5 \leq 8.5 \\ 9, & if\ x \geq 0.85 \end{cases} \quad (4)$$

Here $\mathcal{s}_T \in I$ and $\mathcal{s}_T \geq 0$. A point to note here is that unlike the classical MDP, in this case, the state transition is oblivious to the action taken at the epoch and is totally controlled by the environment, which is the wireless network in this case.

In this work, a tabular RL-technique, known as Q-learning [12], is used. It is chosen due to its low computational complexity which is suitable for embedded sensor nodes with inherent energy and computational resource limitations. This approach is computationally less complex than other RL-based approaches such as the policy gradient-based RL method. The use of Q-learning ensures that the objective of efficient sleep scheduling is achieved with a low computational burden on the sensor nodes.

The reward for the agent in node n at an epoch t in this setting is given by Eq. 5.

$$r_n(t) = \begin{cases} \frac{\tau_1}{P_{miss}^{rx}(t)+\delta}, & if Ql(t) < Ql_{max} \times \nu \\ -\tau_2, & otherwise \end{cases} \quad (5)$$

Here, $P_{miss}^{rx}(t)$ and $Ql(t)$ denote the missed packet receptions and queue length at epoch t; Ql_{max} is the MAC packet queue size (that is, $Ql_{max} = K$, for an M/G/1/K queue); and τ_1, τ_2, ν and δ are hyperparameters that are chosen empirically, as detailed in Section V. The physical interpretation of the reward function is that if the queue length is higher than the discounted value of maximum possible queue length, then the action should be penalized, since it will lead to packet drops resulting from unstable queue. Else, the action is rewarded for low missed packet receptions. In other words, an action is rewarded more if it reduces the packet drops resulting from energy shortage, bad sleeps, or an unstable queue.

Thus, the transmission scheduling agent learns the policy for packet transmission on-the-fly to reduce the packet drops resulting from shortage of energy while maintaining a stable queue. However, to be noted that the transmission strategy learnt by this agent is contingent upon the transceiver on/off policy of the node. To exemplify, if the packet transmission probability of a node is P_{tx} and the node off probability is P_{off}, then the effective transmission rate of the node is given by $P_{tx} \times (1 - P_{off})$. Thus, the node's sleep decisions indirectly affects the learning behavior of the transmission scheduling. In addition, efficient sleep decisions also play a role in reducing the missed receptions of packets from upstream nodes. This calls for the requirement of a sleep scheduling agent that can cooperate with the transmission scheduling agent to achieve the above goals.

Sleep Scheduling Agent: The goal of this agent is to find a transceiver '*On/Off*' schedule for the energy harvesting node it is part of, so that the limited energy budget of the node can be efficiently managed while maintaining reliable communication by reducing packet loss.

The action space ($\mathcal{A}_S$) of the sleep scheduling agent is given by the probability of keeping the radio transceiver on (P_{on}) in an RL decision epoch of h frames. This probability is discretized into $|\mathcal{A}_S|$ discrete values in the range $[0, 1]$, where $|\mathcal{A}_S|$ denotes the cardinality of the action space of the sleep scheduling agent. The actions are selected using ϵ-greedy policy to maximize the expected long-term expected reward using Q-learning approach mentioned above.

The state space ($\mathcal{S}_S$) for this agent is defined by the quantized probability of packet transmission (P_{tx}) in a learning epoch. The probability P_{tx} and hence the state of the sleep scheduling agent is directly controlled by the learning policy of the transmission scheduling agent. The logic behind using P_{tx} as the state for this agent is because the suitable value of transceiver on probability for minimum packet misses is dependent on the transmission probability (see the inset plot in Fig. 1). Now, for a given P_{tx} governed by the action selected by the transmission scheduling agent, and that determines the state for the sleep scheduler, this RL agent decides the sleeping strategy based on the learnt Q-table. Formally, the state perceived by the agent for node n at a decision epoch t is given by Eq. 6 ($s_S \in I$ and $s_S \geq 0$).

$$s_{n,Sl}(t) = f\big(P_{Tx,n}(t)\big) = s_s \; if \; s_s \leq 10 \times P_{Tx}(t) \leq s_s + 1 \quad (6)$$

Here, $P_{Tx,n}(t)$ is the probability of transmission by node n at epoch t. Similar to the other agent, the state transition probability in this case is also totally controlled by the environment, unlike classical MDP problem, where the transition is dependent on the agent's policy as well. The actions taken by the sleep scheduling agent are evaluated using the same reward function given in Eq. (5). The same reward is used for both the agents since they share the common objective of reducing missed packet receptions and queue drops.

Using the above two-stage interactive RL model, the wireless nodes learn a joint transmit-sleep policy so as to manage available harvested energy efficiently to maintain a reliable communication in a resource-constrained network.

V. EXPERIMENTATION

The experiments are performed to analyze the performance of the proposed scheduling protocol using a MAC layer simulator with embedded learning components. The time-driven simulation kernel performs event scheduling in terms of packet generation, transmissions, and receptions. To implement the proposed protocol, the RL model is embedded on top of the MAC layer functions. The baseline experimental parameters are tabulated in Table I. From the transmitter amplifier and radio energy consumption details tabulated in Table I, the reception to transmission power consumption becomes $E_R : E_T \approx 0.04$. The performance of the learning-based MAC protocol is evaluated on the following metrics.

Missed Packet Reception Rate (P_{miss}) represents the rate of packet misses due to oversleeping and shortage of energy. Queue drop rate (Q_{drop}) indicates the rate of packet drops resulting from full MAC packet queue. The combined packet loss (P_{loss}) can be expressed as sum of P_{miss} and Q_{drop}. Queueing delay is computed from queue length using Little's Law [1].

TABLE I: BASELINE EXPERIMENTAL PARAMETERS

Parameter	Value
P_C	0.2
w_s	33.33 m/s
C	50 m
α	0.99
γ	0.1
h	200
B	150
P_{charge}	0.9
Ql_{max}	1000
$\lvert\mathcal{A}_T\rvert = \lvert\mathcal{A}_S\rvert$	10
τ_1	100
τ_2	1
ν	0.90
δ	0.001
α_T	2.4
P_{tx}	0.353 mW
P_{ct}	50μW

We consider two scheduling policies for comparing the proposed approach. First, a naïve scheduling policy is considered to understand the environment and to create a benchmark for the RL-based scheduling mechanism. The naïve policy here is that the node remains awake in a frame with probability P_{on} and transmits with probability P_{tx} given the node is on. We experiment with different combinations of these probabilities to understand the variation of the objective space in response to these sleep-transmit decisions and to see where the learning-based solutions lie in that space. The proposed approach is also compared against the decentralized hybrid scheduling policy (battery and queue-based) proposed in [1], where, a sensor node goes from active to sleep mode depending on the queue length and battery status.

VI. RESULTS AND ANALYSIS

In order to test feasibility and gain insights of the proposed mechanism, we first analyze it on a single flow, three nodes network as shown in Fig. 2 (h), where the node H is the energy harvesting node, receiving packets from source S with flow data rate $\lambda = 0.75$ packet per frame (ppf) to be forwarded to destination D. Performance is analyzed first using the naïve scheduling policy explained in Section V and then using the RL-based scheduling approach detailed in Section IV. A three-dimensional surface plot shown in Fig. 2 (a)-(d) summarizes the performance of the naïve policy for different sleep and transmit probabilities. The following observations can be made for the naïve scheduling policy. For low transceiver on probability (P_{on}), missed packet reception rate (P_{miss}) first increases with increase in transmit probability (P_{tx}) and then saturates. This is because, with increase in P_{tx}, harvested energy drains out more that leads to high percentage of packet misses (P_{miss}). However, for very high value of P_{tx}, missed packet reception (P_{miss}) becomes very less sensitive to P_{tx}, because of very low effective data rate λ^{eff} due to missed receptions. Note that the effective data rate $\lambda^{eff} (< \lambda)$ is determined by λ, P_{on}, P_{tx} and other energy harvesting parameters (Section V). With increase in P_{tx}, packet missed receptions increase and there are less packets received, and more packets transmitted by the node because of high P_{tx}. This causes the queue length and hence queuing delay to decrease with P_{tx} (Fig. 2 (d)). With increase in queue length, queue drop rate (Q_{drop}) also increases with P_{tx} in scenarios when $P_{tx} < \lambda^{eff}$, owing to unstable queue (Fig. 2 (b)). On the other side, with increase in ON probability (P_{on}), P_{miss} decreases. Reduction in packet loss is because the node is ON for more duration and hence allowing more packets to be received. Note that, for high P_{tx}, the gradient of decrease in

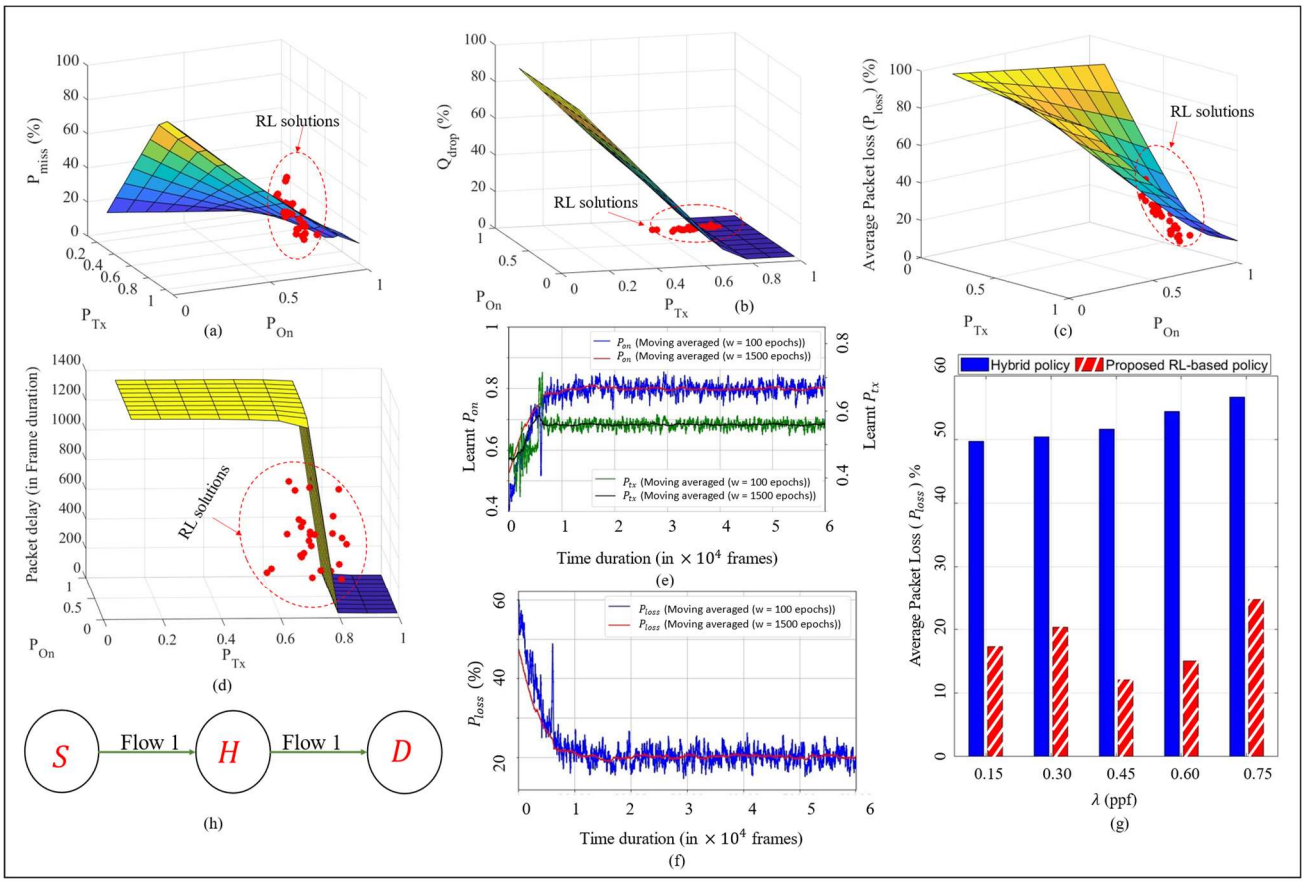


Fig. 2:(a)-(d): Network performance with naïve policy and proposed RL-based policy, (e)-(f): RL convergence behavior, (g) Comparison of proposed policy with existing hybrid policy, (h) Simple one-flow network

P_{miss} with P_{on} reduces. This can be explained by the fact that since the node is ON with high probability for high P_{on}, this leads to more energy consumption (which is high for high P_{tx}), so more packet loss. However, effect of P_{on} on queuing delay and hence queue drop rate (Q_{drop}) is not significant. This is because, expected queue length ($\mathbb{E}[N_Q]$) as defined by Eq. (7) does not get affected by P_{on}, since flow rate and service rate both are affected by P_{on} the same amount.

$$\mathbb{E}[N_Q] \propto \frac{\rho^2}{1-\rho}, \text{ where } \rho = \frac{p_{on} \times \lambda}{p_{on} \times \mu} = \frac{\lambda}{\mu} \quad (7)$$

Thus, the desired scheduling policy should be such that the packet drops resulting from missed receptions and queue drops should be minimized while still maintaining a stable queue. In other words, the aim here is to find transmit and sleep policies that can find the minimum in the surface of P_{loss} in Fig. 2 (c). To be noted that the variation of P_{loss} with P_{tx} and P_{on} is dependent on the energy harvesting parameters and network traffic, and hence the static policies cannot find the right balance among all the above-mentioned performance metrics.

Now, experimenting with the proposed RL-based architecture, the solutions obtained are indicated by red points on the surfaces as shown in Fig. 2 (a)-(d). The RL-solutions obtained are concentrated in the region of low values of P_{loss} (Fig. 2 (c)). To be noted that the RL-based architecture finds solutions that are better than the naïve scheduling policy in terms of P_{loss} for the same P_{tx}, P_{on} values, most of the times. This is because the RL approach allows the nodes to learn a dynamic sleep-transmit scheduling. In other words, the learnt scheduling probabilities oscillate epoch by epoch. To exemplify, a transmit probability of 0.8 can indicate refraining from transmission deterministically for two consecutive epochs (400 frames), thus recharging its battery, and then transmitting for next eight epochs (1600 frames), thus, utilizing its recharged battery. It is observed that such dynamic policies learnt by the RL agents help the node to obtain a packet loss (P_{loss}) rate lower than the static naïve policies for the same $< P_{tx}, P_{on} >$ tuple. Also, it is observed that the queueing delays of the RL-based solutions lie in the narrow region of transition from stable to unstable queue. This is because, missed packet reception rate (P_{miss}) increases with P_{tx} and queue drop rate, delay decreases with P_{tx} (Fig. 2 (a), (b)). The RL-based framework makes the nodes learn policies so that there is the right balance between these two opposing performance parameters.

Fig. 2 (e), (f) demonstrates the convergence behavior of the learning framework in terms of average P_{tx}, P_{on} and P_{loss}. The plots show both the long-term average as well as the transient behavior to capture the learning progression. Over time, the nodes learns transmission and sleep scheduling policy using the above-discussed framework, so that MAC packet loss reduces. Another observation is that on an average, the learning convergence time remains in the vicinity of 10^4 frame durations. Thus, for a typical MAC frame duration (for a degree-10 network) of 3-4 ms [13], the convergence happens within 30-40 sec. This makes the approach highly practical in that it can cope with network condition changes with a time constant larger than roughly a minute. For many static (i.e.,

non-mobile) sensor networks, the time constants for network topology and traffic condition changes can be much larger – often up to hours.

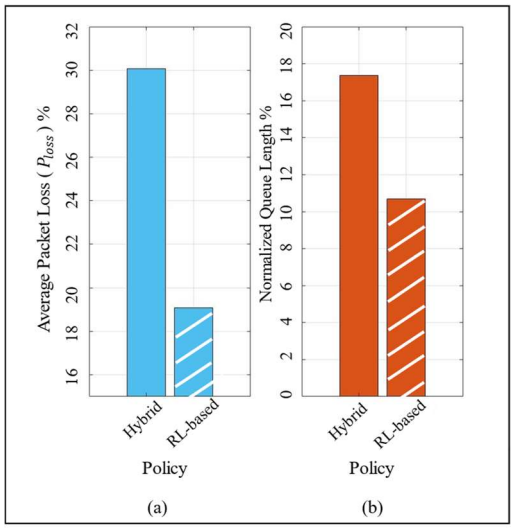
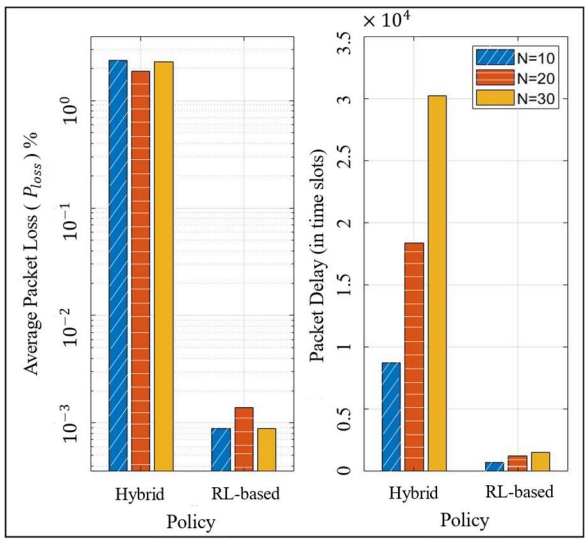


Fig. 3: Performance Comparison in heterogeneous traffic conditions

The proposed RL-based scheduling logic is experimented for different flow data rate λ and then compared with the existing hybrid sleep scheduling policy mentioned earlier in Section V. As depicted in Fig. 2 (g), the RL-based policy outperforms the hybrid sleep scheduling policy in terms of packet loss for different traffic patterns. The mean MAC queueing delay for the range of flow rates experimented is 126.64 frame duration which is when the MAC packet queue is 9.5% full.



Fig, 4: Performance in one-hop networks with varying network size (N)

Performance of the proposed approach has been evaluated for a 20-nodes network, with 10 active flows ($N = 10, L = 2$ in the network in Fig. 1) and heterogeneous traffic and is shown in Fig. 3. The traffic pattern is such that flow ID 1-3; 4-6; 7-9 and 10 had data rate $\lambda = 0.25;\ 0.10;\ 0.50$ and 0.25 ppf respectively. It is observed that there is an $\approx 12\%$ decrease in P_{loss} and $\approx 6\%$ decrease in normalized queue length (and hence queueing delay), compared to the hybrid sleep scheduler.

As a special case, we experiment with the learning-based scheduler in a one-hop network ($L = 1$ in Fig. 1) with N sensor nodes, where N varies from 10 to 30. To be noted that since it is a one-hop network, the packet loss is completely due to queue drop It is observed that, as shown in Fig. 4, for all the three networks, the proposed protocol achieves a low packet loss rate as well as a low queuing delay compared to that achieved by the hybrid sleep scheduler. This is because the hybrid sleep scheduler does not consider transmission scheduling into consideration, which gives rise to both high packet delays and queue drop rate. Furthermore, the increase in delays with increase in network size is because of the increase in MAC frame size with increase in number of wireless nodes.

VII. Summary and Conclusions

In this work, a Reinforcement Learning-based joint transmit-sleep scheduling architecture is proposed for reliable communication in a solar energy-harvested wireless network. The scheduling problem is modeled as an MDP and solved using an interactive RL model. The online learning ability of the proposed approach makes the nodes learn a scheduling policy so as to minimize the MAC layer packet loss, while maintaining a stable packet queue, in the presence of limited energy budget and heterogeneous traffic patterns. Learning is decentralized in that each node learns its policy independently without explicit information sharing. With simulation studies, the proposed approach is validated and compared against a benchmark policy and an existing hybrid sleep-scheduling mechanism. Future extension of this work includes developing the framework for time-varying energy harvesting model, heterogeneous topologies, and energy profile.

VIII. References

[1] Niyato, Dusit, et al. "Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: Performance analysis and optimization." *IEEE Transactions on Mobile Computing,* 2007, 221-236.

[2] Chen, Guihong, et al. ‘‘Reinforcement learning based power control for in-body sensors in WBANs against jamming,’’ IEEE Access, 2018

[3] Chen, Guihong, et al. "Reinforcement learning-based sensor access control for WBANs." IEEE Access 7 (2018): 8483-8494.

[4] Wang, Xun et al. "A reinforcement learning-based sleep scheduling algorithm for compressive data gathering in wireless sensor networks." *EURASIP Journal on Wireless Communications and Networking* 2023.1 (2023): 28.

[5] Mohammadi, Razieh, and Zahra Shirmohammadi. "RLS2: An energy efficient reinforcement learning-based sleep scheduling for energy harvesting WBANs." *Computer Networks* 229 (2023): 109781.

[6] Sangaiah, Arun Kumar, et al. "SALA-IoT: Self-reduced internet of things with learning automaton sleep scheduling algorithm." *IEEE Sensors Journal* (2023).

[7] Cao, Xianbo, et al. "A deep reinforcement learning-based on-demand charging algorithm for wireless rechargeable sensor networks." *Ad Hoc Networks* 110, 2021

[8] Xu, Yi-Han, et al. "Reinforcement learning (RL)-based energy efficient resource allocation for energy harvesting-powered wireless body area network." *Sensors* 20.1 (2019): 44.

[9] Wang, Lili, et al. "Joint optimization of power control and time slot allocation for wireless body area networks via deep reinforcement learning." *Wireless Networks* 26 (2020): 4507-4516.

[10] Das, Sankar Narayan, et al. "Temporal-correlation-aware dynamic self-management of wireless sensor networks," *IEEE Transactions on Industrial Informatics* 12.6 (2016): 2127-2138.

[11] Liu, Zhiqiang, et al. "Joint power-rate-slot resource allocation in energy harvesting-powered wireless body area networks." IEEE Transactions on Vehicular Technology 67.12 (2018): 12152-12164

[12] Sutton, Richard S., et al. “Reinforcement learning: An introduction” MIT press, 2018.

[13] Zhang, Xinming, et al. "A black-burst based time slot acquisition scheme for the hybrid TDMA/CSMA multichannel MAC in VANETs." IEEE Wireless Communications Letters 8.1 (2018): 137-140.