

Modelling and Optimization of DRX in Cellular IoT Networks: an MDP Approach¹

Nicholas Accurso*, Nicholas Mastronarde[†], and Filippo Malandra[‡]

Department of Electrical Engineering, University at Buffalo

Email: {naccurso*, nmastron[†], filippom[‡]}@buffalo.edu

Abstract—Due to the exponential growth of endpoints in the Internet of Things (IoT), new protocols have been proposed to utilize cellular infrastructures, allowing a large amount of IoT devices to communicate through them. These novel protocols make up the Cellular IoT (C-IoT). In C-IoT, the energy efficiency of endpoints is essential in order to reduce both operational cost and required maintenance. One method of energy reduction is Discontinuous Reception (DRX). DRX allows a device's Radio Frequency (RF) circuitry to turn off for brief periods of time. While off, the device experiences a tradeoff between saving energy and an increase in expected latency, which can be tuned by how long the device spends asleep. In this paper, we model DRX as a Markov Decision Process (MDP). This MDP is solved using a dynamic programming approach and verified through simulation. Further, the energy-latency tradeoff is explored by varying the device's priority on either energy or network performance in addition to varying the traffic intensity.

Index Terms—Energy Efficient Devices, Markov Decision Processes, Discontinuous reception (DRX), Cellular IoT, Constrained Devices, Device Management, Efficient Communications and Networking.

I. INTRODUCTION

TO help account for the massive growth of the Internet of Things (IoT), Cellular IoT (C-IoT) networking protocols have been proposed. These C-IoT protocols allow IoT traffic to communicate using existing cellular infrastructures. Two popular novel C-IoT protocols are NarrowBand IoT (NB-IoT) and LTE Cat-M. These protocols allow User Equipments (UEs) to communicate using a more narrow bandwidth when compared with legacy cellular protocols such as LTE. This, in turn, allows more users to coexist in the same cell.

Compared to its competitors, such as Long Range Wide Area Network (LoRaWAN) and Sigfox, C-IoT protocols can offer better performance in many areas, including energy consumption of UEs [1]. This can be done primarily in three ways: i) improving the scheduling and routing of information through the network [2] [3], ii) processing data using more energy efficient methods (e.g. cloud computing) [4] [5], and iii) introducing sleep modes for nodes in the network [6] [7]. There are three direct consequences of improving energy efficiency in such networks: the amount of waste generated as a byproduct of the device's operation is reduced, maintenance of devices is decreased, and the cost of operation is reduced.

However, it is rarely the case that a reduction in energy consumption does not come at a cost. Two prime examples of this are Discontinuous Reception (DRX) and Power Save Mode (PSM), which were introduced in LTE to extend the battery life of end devices. In [9], the authors introduce DRX and PSM, provide an analytical model for both, and evaluate the performance of both mechanisms through their implementation in Network Simulator 3 (NS3) using the NB-IoT protocol. In essence, DRX and PSM allow devices to turn off their Radio Frequency (RF) circuitry, which would otherwise consume considerable energy while on. At the same time, however, the device is not reachable by the network. If a packet is sent to the device while it is *off*, significant delays can be incurred since any Downlink (DL) traffic will need to be buffered at the base station. Thus, DRX and PSM have an inherent energy-latency tradeoff. In essence, by tuning the various timers that facilitate DRX and PSM operation, we also tune this tradeoff. This tradeoff is also affected by the traffic conditions in the network. In this paper, we formulate the problem of DRX *off* duration optimization considering traffic conditions as a Markov Decision Process (MDP). An MDP was selected because it allows the modelling of a time varying environment in which an agent makes decisions that will impact both immediate and future network performance.

In [10], the DRX mechanism is evaluated through a cross-layer analytical model with traffic distributed according to a Poisson process. Results show that the introduction of the DRX mechanism results in a considerable improvement (up to three times) in the energy efficiency of the device. Further, results show that, for given DRX timers, there is a certain traffic load at which the energy efficiency improvement of the mechanism is optimum. This illustrates the importance of choosing DRX timers according to traffic load to achieve the best energy efficiency and delay results.

In [11], the authors propose an actor-critic algorithm to improve the latency-energy tradeoff that exists in DRX. The authors consider a modified DRX mechanism consisting of four states: continuous reception, *on* duration of DRX cycle, *off* duration of DRX cycle, and Radio Resource Control (RRC) Idle. The algorithm learns over time through the modification of the timers that facilitate state transitions (e.g., *on* duration of DRX cycle). The authors evaluate the proposed algorithm using MATLAB, and find that the proposed algorithm outperforms standard extended DRX (eDRX) in terms of energy efficiency by approximately 300%. However, the average delay

¹This material is based upon work supported by the National Science Foundation under Grant No. 2105230

of the actor-critic algorithm is much greater at approximately 280 ms, compared with conventional DRX at 50 ms.

In contrast to previous work, this work formulates the DRX mechanism in full as a single MDP. In doing so, we are able to directly solve the MDP through dynamic programming. In much of the available literature, the energy-delay tradeoff in DRX is examined through setting a delay constraint and attempting to minimize energy consumption subject to that constraint. However, in this work, we define a single continuous variable that can be used to tune this tradeoff in either direction, i.e., varying emphasis can be placed on either energy or delay, which allows for a much wider range of operating points. Finally, we validate our results through simulation.

The remainder of the paper is organized as follows. In Section II, we formulate the problem as an MDP and solve it using value iteration. In Section III, we present and analyze our results. Finally, in Section IV, we conclude the work.

II. PROBLEM FORMULATION

In what follows, we provide an overview of the DRX technique and of MDPs. Then, we describe our DRX timer optimization as an MDP. Specifically, we first introduce the state space of the MDP, then define the action space and when an action is taken. Next we define how the state evolves over time via the transition probability function. After that, the cost function is introduced. Finally, we describe the method used to solve the MDP, namely, value iteration.

A. DRX Overview

A timing diagram of the DRX mechanism is illustrated in Fig. 1. In DRX, if the device has gone a certain period of time without having received a packet, it will enter *DRX cycles*. Each of these DRX cycles consists of an *off* and an *on* period. When the device is *off*, it will minimize the activity of its RF circuitry to not waste energy monitoring channels. During this period, the device is saving energy, but it is unable to receive DL packets. During the *on* period, the device will consume energy to wake up and check the radio control channel, to see if there are any incoming DL packets. If there are none, the device will go back into the *off* mode, and these cycles will continue. However, if there are any packets, the device will wake up fully, and exit these DRX cycles.

PSM is an additional sleep mode, which allows the device to sleep for much longer periods. PSM is triggered by the device going through m consecutive DRX cycles without any DL packets. In PSM, the device saves energy by turning off its RF circuitry for an extended period of time, but is unreachable by the network. Eventually, the device will wake up from PSM and go back to regular operation.

B. Markov Decision Processes

To model the DRX mechanism, an MDP is introduced. An MDP is used to model an agent making decisions in a stochastic environment in which immediate decisions impact the current and future costs. We will consider a discrete-time MDP with uniform time steps Δt . In each time step, the

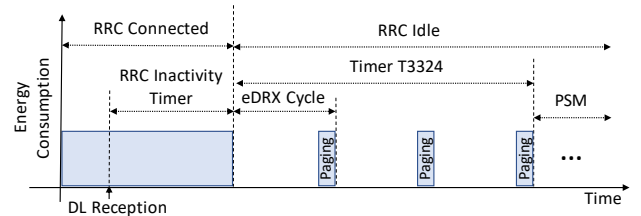


Fig. 1. DRX timing diagram.

agent first observes the current state $s \in \mathcal{S}$. The agent then takes action $a \in \mathcal{A}(s)$ accordingly, where $\mathcal{A}(s)$ denotes the set of available actions in state s . Finally, the environment stochastically transitions to state $s' \in \mathcal{S}$. The probabilities of transitions between states are defined by the following Transition Probability Function (TPF) P :

$$P(s', a, s) = \Pr[s'|s, a], \quad s, s' \in \mathcal{S}, \quad a \in \mathcal{A}(s). \quad (1)$$

The fourth component of an MDP is the cost function $C(s, a)$. This cost function measures how “expensive” the action a was in state s . The fifth and final part of an MDP is the discount factor $\gamma \in [0, 1)$. γ defines how much the model cares about future costs. When γ is zero, all the weight is placed on immediate cost while as γ approaches one, more emphasis is placed on anticipated future costs. Details about the states, actions, and TPF in the proposed MDP are provided in the subsections below. Overall, in an MDP, we look to minimize the infinite horizon discounted sum of costs, specifically

$$\min_{\pi} \sum_{t=0}^{\infty} \gamma^t C(s_t, \pi(s_t)), \quad (2)$$

where $\pi : \mathcal{S} \rightarrow \mathcal{A}$ denotes the decision policy, which maps states to actions.

C. States

Similar to the model in [11], our base model of DRX comprises four “macro” states, i.e., RRC Connected (S_{RRC}), DRX on (S_{ON}), DRX off (S_{OFF}), and PSM (S_{PSM}), as illustrated in Fig. 2 where each of these four states are color coded. In S_{RRC} , the device is fully awake and can transmit or receive packets at any time. In the second state, S_{ON} , the device is in the awake part of its DRX cycles, and is able to receive a packet at any time during this state. In the third state, S_{OFF} , the device is in its *off* period of the DRX cycles. The device is consuming a reduced amount of energy, but it cannot be reached by the network, so any DL packet that arrives in this state will have an added delay. In the final state, S_{PSM} , the device sleeps for a long period of time.

We define $\mathcal{S}_m$ to be the set of all possible “macro” states, i.e. $\mathcal{S}_m = \{S_{RRC}, S_{ON}, S_{OFF}, S_{PSM}\}$. Each of these “macro” states is composed of a number of sub-states. To define these sub-states, a couple of additional variables must be considered. The first addition is a timer state that will help facilitate the

transitions between these main states. The set of possible timer values $t \in \mathcal{T}$ depends on the current DRX state as follows:

$$t \in \begin{cases} [0, T_{RRC}), & \text{if } s_m = S_{RRC} \\ [0, T_{ON}), & \text{if } s_m = S_{ON} \\ [0, T_{OFF}^{max}), & \text{if } s_m = S_{OFF} \\ [0, T_{PSM}), & \text{if } s_m = S_{PSM}. \end{cases} \quad (3)$$

It is worth noting that all timer values can be only integers.

The second addition is a Boolean packet indicator state that indicates the existence of a packet, i.e., this indicator will be 1 if there is a packet waiting and 0 otherwise. Note that this indicator can only be 1 in states where immediate reception of the packet is not possible (S_{OFF} and S_{PSM}); hence,

$$S_{pkt} \in \begin{cases} \{0, 1\}, & \text{if } s_m \in \{S_{OFF}, S_{PSM}\} \\ \{0\}, & \text{if } s_m \in \{S_{RRC}, S_{ON}\}. \end{cases} \quad (4)$$

The resulting state space $\mathcal{S}$ is then defined as a subset of the Cartesian product of the macro state, timer state, and packet indicator state:

$$\mathcal{S} \subset S_m \times \mathcal{T} \times S_{pkt}. \quad (5)$$

It is important to note that not all elements resulting from this Cartesian product are actually possible. For instance, assuming $T_{ON} < T_{RRC}$, then the state $s = (S_{ON}, T_{RRC} - 1, 0)$ is a state within this Cartesian product, but is not reachable.

The size of the state space is given as follows:

$$|\mathcal{S}| = T_{RRC} + m(T_{ON} + 2T_{OFF}) + 2T_{PSM}. \quad (6)$$

D. Actions

The action considered in this model is the length of time the device spends in the *off* period of its DRX cycles, i.e., T_{OFF} . We define this action space $\mathcal{A}(s)$ to be a discrete set of pre-determined timer values whose entries depend on the current state. Recall that this action is only taken immediately prior to switching to S_{OFF} . Thus, the action space only contains possible selections at this specific state, i.e., $s = (S_{ON}, 0, 0)$. For all other states, $\mathcal{A}(s)$ is the empty set:

$$\mathcal{A}(s) \in \begin{cases} (0, T_{OFF}^{max}), & \text{if } s = (S_{ON}, 0, 0) \\ \emptyset, & \text{otherwise,} \end{cases} \quad (7)$$

where T_{OFF}^{max} is the largest possible DRX off timer.

E. Transition Probability Function

Now that the state space and the transitions between states have been modelled, all that is needed before we arrive at the transition probability function is a model of the incoming traffic. To this end, we use a Bernoulli-distributed traffic model [12], [13]. In each time step, there is a probability p of there being an incoming packet. This distribution keeps the model simplest, as the probability of a packet arrival in a given time slot does not vary with time. This results in a transition probability function that also does not vary with time.

With this traffic distribution defined, the transition probability function can be constructed. The high level view is illustrated through the “macro” state transition diagram in Fig.

2. Note that in these state transition diagrams, states with a dashed border describe a general macro state, while a solid border indicates a specific state.

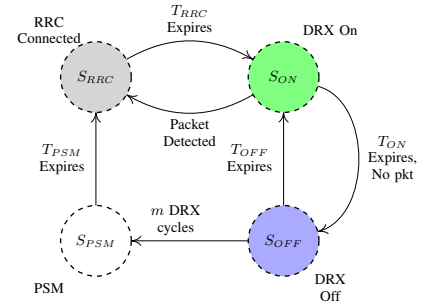


Fig. 2. DRX state transitions overview.

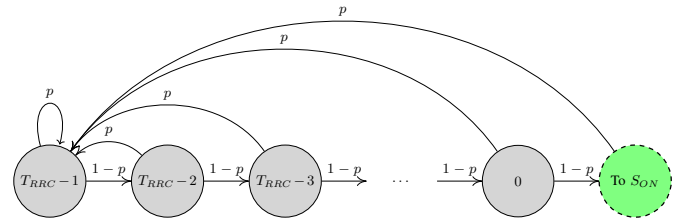


Fig. 3. RRC connected state transitions.

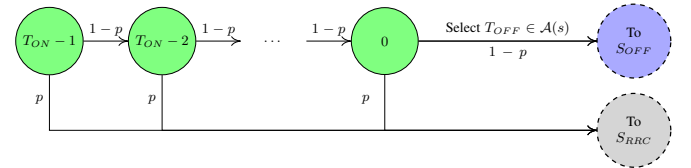


Fig. 4. DRX on state transitions.

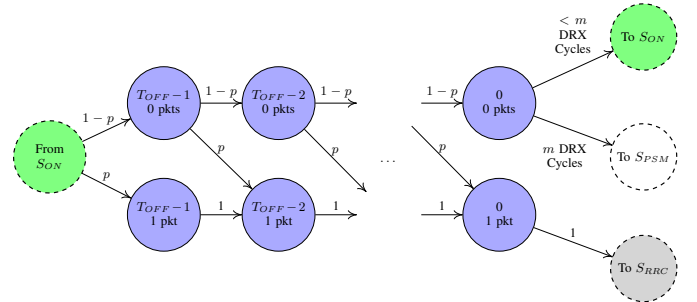


Fig. 5. DRX off state transitions.

Next, we will go through the transition probabilities within each of these high level states. The system is initialized in S_{RRC} . With each time step, the timer state is decremented with probability $1 - p$, and gets reset to timer state $T_{RRC} - 1$ with probability p . Once the timer state reaches 0, it will instead transition to S_{ON} with probability $1 - p$ and go back to the start with probability p . These stochastic timer transitions occur similarly in the state S_{ON} , as illustrated in Fig. 4. The only difference is that with probability p the state will

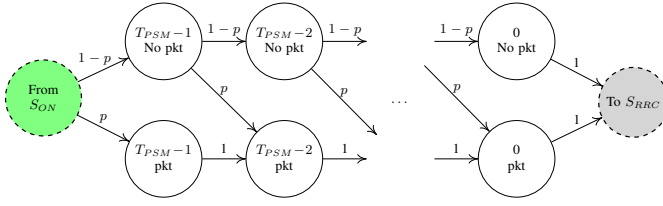


Fig. 6. PSM state transitions

transition back to S_{RRC} in each time step. At timer state 0, with probability $1 - p$, T_{OFF} will be selected and the state will transition to S_{OFF} . It will instead transition to S_{RRC} with probability p . While in S_{OFF} , the timer state is decremented with each time step. In each time step, if the current value of s_{pkt} is 0, there is a probability p that s_{pkt} in the next time step is 1 and a probability $1 - p$ that this packet indicator remains the same. At the final time step in S_{OFF} , if there is a packet, then the state will transition back to S_{RRC} . Otherwise, the state will transition to S_{PSM} if m DRX cycles have elapsed and back to S_{ON} otherwise. The PSM state transitions are illustrated in Fig. 6. In S_{PSM} , the timer state is decremented with each time step until timer state zero is reached, at which point the state is transitioned back to S_{RRC} .

F. Immediate Cost

The immediate cost $C(s)$ is defined by considering a weighted sum of delay and energy costs as follows:

$$C(s) = D(s) + \lambda E(s). \quad (8)$$

Note that this particular cost function does not depend on the action a , so it will simply be denoted as $C(s)$. Here, $D(s)$ is the delay cost in state s , $E(s)$ is the energy cost in state s , and λ is a coefficient that adjusts the weight placed on energy as opposed to delay. For example, when $\lambda = 0$ the UE places all priority on reducing latency no matter the cost in terms of energy. The values of $D(s)$, $E(s)$, and $C(s)$ for each state are given in Table I. In S_{RRC} and S_{ON} , the delay cost is always 0 and the energy cost is always ϵ_0 . This is because in these states, the UE is consuming maximum energy to stay awake and minimize delay. In S_{OFF} , the energy cost is always

$$E(s) = \alpha \epsilon_0, \quad (9)$$

where ϵ_0 is the energy consumed in S_{RRC} and α is the fraction of the energy ϵ_0 consumed in S_{OFF} . The delay cost in this state is 0 when there is no packet waiting and 1 when there is a packet waiting. Similarly, in S_{PSM} , there is no cost associated with energy loss, and a delay cost of 0 when there is no packet waiting and 1 when there is a packet waiting.

G. Value Iteration

After constructing all the necessary MDP components and completing the DRX model, the optimal actions need to be found. To do this, value iteration is employed. The process of value iteration can be found in [15].

The value iteration algorithm given in Algorithm 1 takes as an input the MDP, i.e., $S, \mathcal{A}, P(s', a, s), C(s)$, and γ . As

TABLE I
IMMEDIATE COST TABLE.

State	Packet Waiting			No Packet Waiting		
	$D(s)$	$E(s)$	$C(s)$	$D(s)$	$E(s)$	$C(s)$
S_{RRC}	0	ϵ_0	$\lambda \epsilon_0$	0	ϵ_0	$\lambda \epsilon_0$
S_{ON}	0	ϵ_0	$\lambda \epsilon_0$	0	ϵ_0	$\lambda \epsilon_0$
S_{OFF}	1	$\alpha \epsilon_0$	$1 + \alpha \lambda \epsilon_0$	0	$\alpha \epsilon_0$	$\alpha \lambda \epsilon_0$
S_{PSM}	1	0	1	0	0	0

Algorithm 1 Value Iteration Algorithm.

Input: $S, \mathcal{A}, P, C, \gamma$

Output: Q, V, π

Initialization:

```

1:  $\gamma, \delta, Q(s, a), V(s)$ 
2:  $V_{old}$  s.t.  $\max_s |V(s) - V_{old}(s)| > \delta$ 
3: while  $\max_s |V(s) - V_{old}(s)| > \delta$  do
4:   for  $s \in \mathcal{S}$  do
5:     for  $a \in \mathcal{A}(s)$  do
6:        $Q(s, a) \leftarrow C(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s', a, s) V(s')$ 
7:     end for
8:      $V_{old}(s) \leftarrow V(s)$ 
9:      $V(s) \leftarrow \min_a Q(s, a)$ 
10:   end for
11: end while
12: return  $Q, V, \pi$ 
```

an output, the algorithm provides two functions: the action-value function $Q(s, a)$, which tells us how good or bad it is to take action a in state s and then follow the optimal policy π^* thereafter; and the value function $V(s)$, which tells us how good or bad being in state s is assuming the optimal policy π^* is followed. The final output is the optimal policy π^* , which indicates the action with the lowest associated value in the state $s = (S_{ON}, 0, 0)$.

After the two output functions are initialized to arbitrary values, the value iteration algorithm consists of two steps that are repeated for all possible states until an exit condition is met. In the first step in line 7 of Algorithm 1, a form of Bellman's Equation is used to update the action-value function for every possible action a . This equation consists of the summation of two parts. The first part is simply the immediate cost from the MDP model. The second part is a measure of expected future costs. This part is multiplied by a discount factor, $\gamma \in [0, 1]$, which quantifies how much the algorithm should care about the future. In the second step of the algorithm given in line 9 of Algorithm 1, the value function is updated based on the current best action to take in each state. These two steps are repeated until the value function is relatively static for all states. This is checked after step 2 using the old and new value functions and a threshold δ . In the case of this problem, an action is only taken in the final timer state of DRX on, $s^* = (S_{ON}, 0, 0)$. So, we only need to look at the optimal action in this state to determine

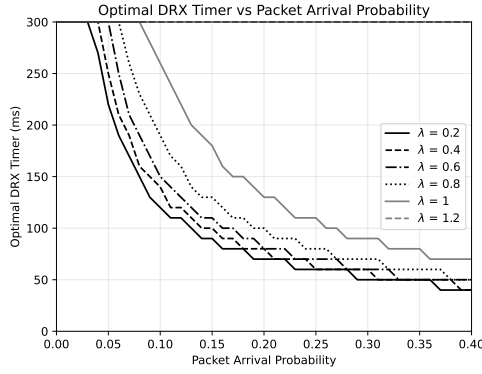


Fig. 7. Optimal DRX off timer vs packet arrival probability

the optimal timer: i.e.,

$$T_{OFF}^* = \operatorname{argmin}_a Q(s^*, a). \quad (10)$$

III. RESULTS

This section contains: i) results obtained through value iteration for a range of traffic intensities and energy-latency priorities (III-A), ii) our simulation setup (III-B), and iii) simulation results for model validation (III-C).

A. Value Iteration Results

The list of values used in the generation of results unless otherwise specified is provided in Table II. First, we varied the values of p and λ and observed the resulting optimal DRX timer by using the value iteration algorithm. More specifically, for each value of λ , we plotted a curve where the packet arrival probability is on the x-axis and the optimal DRX Off timer T_{OFF}^* is on the y-axis. These results are shown in Fig. 7. It should be noted that the curve corresponding to $\lambda = 1.2$ lies at $T_{OFF} = 300$.

These results show exactly what was to be expected. For very low traffic rates (very small p), T_{OFF}^* becomes very large, tending toward the maximum allowed T_{OFF} at $p = 0$. This was expected because at very low traffic rates, the device can be in a sleep mode more often without risking too much network performance degradation. The opposite is also true: as p increases, the optimal DRX off timer becomes shorter. In this case, the system realizes that the probability of missing a packet when sleeping increases with increasing traffic rate, so it decides to stay awake more often. It is important to note that when λ exceeds a certain threshold, the cost of consuming energy becomes greater than any possible delay incurred, so for λ larger than this threshold it is always more beneficial to sleep as long as possible.

B. Simulation Setup

In order to test our model, a simulation scenario was set up using Python. In this scenario, we consider a discrete time simulation in which there exists one base station and one UE that is using the DRX mechanism and employing a policy $\pi(s)$. In each discrete time step of the simulation, the

TABLE II
LIST OF SIMULATION PARAMETERS.

Parameter	Value
T_{OFF}	{10, 20, ... 300} ms
T_{RRC}, T_{ON}, T_{PSM}	100, 30, 450 ms
Number of DRX Cycles m	3
Number of States, actions	103, 30
Δt	10 ms
Energy consumed in $S_{RRC}, S_{ON} \in 0$	1
α, p, λ	0.1, 0.05, 0.6

base station first observes the current state consisting of the “macro” state, the current timer values, and whether or not there is a downlink packet. Next, the cost is calculated from this observed state. Then, the base station will take action $a \in \mathcal{A}$ if the current state is the final timer state of S_{ON} . Finally, the state transitions based on the current state and the existence of a DL packet arrival.

One additional consideration must be made prior to comparing simulation results with our model. In the model, we use a discount factor γ to calculate the infinite horizon discounted sum of costs, while this process is not done in the simulation. Simply averaging the observed simulated cost would therefore introduce a mismatch. To overcome this, the first-visit Monte Carlo method given in Algorithm 2 is used [14]. First, a simulation of n time steps was conducted, and the state visited at each time step $s(t)$ was recorded. Through this entire simulation, the action a is fixed. After the simulation, the discounted future costs of the first visit of each state was calculated. This process is shown in lines 8 through 13 of Algorithm 2. First, the first visit of state s is located, and the time at which this occurred is marked as time t . Next, for all times after t until the end of the simulation $t' \leq n$, the value of state s is updated according to the following equation:

$$V(s) \leftarrow V(s) + \gamma^{t'-t} C(s(t')), \quad (11)$$

where γ is the discount factor and $C(s(t'))$ is the cost of the state visited at time t' . Note that after this value is computed, it will need to be normalized by a factor of $1 - \gamma$ so we can directly compare values for different discount factors. After repeating this for all states, $V(s)$ is returned. This process was repeated for all $a \in \mathcal{A}$.

C. Model Validation

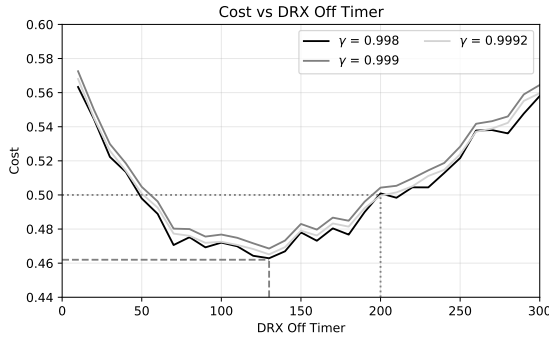
The results of this simulation are provided in Fig. 8. Here, the value function approximation algorithm was conducted for various discount factors γ and the average cost was recorded. Here, it is clear that there exists an optimal timer (indicated by the dashed line) at which the cost is at a minimum, occurring at $T_{OFF} = 130$ ms, which is in agreement with the value iteration results in 7. An example of a suboptimal selection is indicated by the dotted line. It is also clear that as γ approaches 1, the resulting curve approaches a single curve which is the theoretical average cost per time step we would observe in an infinitely long simulation.

Algorithm 2 Value Function Approximation Algorithm.**Input:** S, A, γ, C, π **Output:** V *Initialization:*

```

1:  $V(s)$  arbitrarily,  $t = 0$ ,  $s_0 = (S_{RRC}, T_{RRC}, 0)$ 
2: while  $t < n$  do
3:    $a_t = \pi(s_t)$ 
4:    $s(t) = s_t$ 
5:    $s_t \leftarrow s_{t+1}$ 
6: end while
7: for  $s \in S$  do
8:    $t = \arg \min_t (s(t) == s)$ 
9:   for  $t \leq t' \leq n$  do
10:     $V(s) += \gamma^{t'-t} C(s(t'))$ 
11:   end for
12: end for
13: return  $(1 - \gamma)V(s)$ 

```

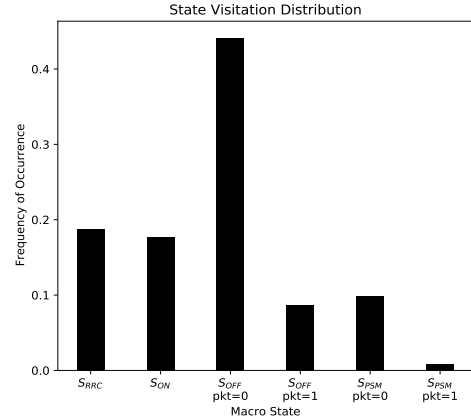
Fig. 8. Observed cost vs DRX off timer for varying γ . $p = 0.05$, $\lambda = 0.6$

Further, from this simulation we were able to gather the experimental steady state distribution, shown in Fig. 9. Here, the value of p was set to 0.02, and T_{OFF} was set statically at a duration of 100ms. It can be seen that under this very small traffic arrival probability, the device spends most of its time in S_{OFF} and S_{PSM} . This occurs because there is a low traffic arrival probability triggering the transition back to S_{RRC} .

IV. CONCLUSION

In this work, the energy-latency tradeoff inherent to DRX was closely examined. First, the problem of optimizing DRX sleep duration was formulated as a MDP where the action taken is the selection of a DRX *off* duration from a discrete set of possible timers. The state of the device evolved according to a discrete Markov chain realistically simulating DRX operation. A single parameter λ was introduced to facilitate the tradeoff between energy and latency in the cost function of the MDP. This MDP was solved using value iteration.

The results of value iteration were analyzed by examining the effects of λ and the incoming traffic intensity p on the optimal timer selection. These results were verified through a simulation during which all possible DRX *off* timers were selected and the average cost was observed. As predicted by

Fig. 9. Steady state distribution over the “macro” states obtained via simulation. $p = 0.02$, $T_{OFF} = 100\text{ms}$.

the value iteration results, there exists a timer at which the observed cost is at a minimum.

REFERENCES

- [1] N. Accurso, N. Mastronarde, and F. Malandra. Exploring Tradeoffs between Energy Consumption and Network Performance in Cellular-IoT: a Survey. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pp 01–06, 2021.
- [2] P. Liu, K. Wu, J. Liang, J. Chen, and Y. Tseng. Energy-Efficient Uplink Scheduling for Ultra-Reliable Communications in NB-IoT Networks. In *2018 IEEE 29th Annual Int. Symp. on Perso., Indoor and Mobile Radio Commun. (PIMRC)*, pp.1–5, 2018.
- [3] N. H. Bui, C. Pham, K. K. Nguyen, and M. Cheriet. Energy Efficient Scheduling for Networked IoT Device Software Update. In *2019 15th Int. Conf. on Netw. and Service Manage. (CNSM)*, pp.1–5.
- [4] B. R. Stojkoska and Z. Nikolovski. Data Compression for Energy Efficient IoT Solutions. In *2017 25th Telecom. Forum*, pp.1–4.
- [5] W. S. Ravindra and P. Devi. Energy Balancing Between Computation and Communication in IoT Edge Devices Using Data Offloading Schemes. In *2018 Second Int. Conf. on Inventive Commun. and Comput. Technol. (ICICCT)*, pp.22–25.
- [6] N. M. Balasubramanya, L. Lampe, G. Vos, and S. Bennett. DRX With Quick Sleeping: A Novel Mechanism for Energy-Efficient IoT Using LTE/LTE-A. *IEEE IoT Journal*, 3(3):398–407, 2016.
- [7] M. S. A. Shuvo, A. R. Munna, T. Adhikary, and M. A. Razzaque. An Energy-Efficient Scheduling of Heterogeneous Network Cells in 5G. In *2019 Int. Conf. on Sustain. Technol. for Industry 4.0 (STI)*, pp.1–6.
- [8] J. Kim, H. Lee, and S. Chong. Traffic-Aware Energy-Saving Base Station Sleeping and Clustering in Cooperative Networks. *IEEE Trans. Wireless Commun.*, 17(2):1173–1186, 2018.
- [9] A. K. Sultania, P. Zand, C. Blondia, and J. Famaey. Energy Modeling and Evaluation of NB-IoT with PSM and eDRX. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pp.1–7, 2018.
- [10] B. Cai, Y. Chen, and I. Darwazeh. Analyzing Energy Efficiency for IoT Devices with DRX Capability and Poisson Arrivals. In *2019 26th Int. Conf. on Telecommun. (ICT)*, pp.254–259.
- [11] J. Zhou, G. Feng, T. P. Yum, M. Yan, and S. Qin. Online Learning-Based Discontinuous Reception (DRX) for Machine-Type Communications. *IEEE IoT Journal*, 6(3):5550–5561, June 2019.
- [12] N. Apthorpe, D. Huang, D. Reisman, A. Narayanan, and N. Feamster. Keeping the Smart Home Private with Smart(er) IoT Traffic Shaping. *Proc. on Privacy Enhancing Technologies 2019*, pp 128–148, 2019.
- [13] N. Akar and O. Dogan. Discrete-Time Queueing Model of Age of Information With Multiple Information Sources. *IEEE Internet of Things Journal*, 8(19):14531–14542, 2021.
- [14] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.
- [15] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.