

On Energy and Data Delivery in Wireless Local Area Networks with RF Charging Nodes

Kwan-Wu Chin

School of Electrical, Computer and Telecommunications Engineering

University of Wollongong, NSW, Australia

Email: kwanwu@uow.edu.au

Abstract

Wireless charging is now a reality. Low-power devices with sensing capabilities deployed within a building for example can now be powered wirelessly via Radio Frequency (RF) transmissions from existing Access Points (APs) that form a Wireless Local Area Network (WLAN). However, an AP cannot transmit frequently to charge devices as it may starve other nearby APs operating on the same channel. Consequently, there is a need to schedule the transmissions of APs to ensure their data queues remain short whilst charging energy-harvesting devices. We present a finite-horizon Markov Decision Process (MDP) to capture the queue states at APs and also channel conditions to nodes. The reward to be optimized is the amount of delivered energy and data. We investigate the following queue selection rules: max weight, max queue, best channel state and random. Our results show that APs that select the best queue in each time slot according to the max weight rule yields a transmission schedule that has the best reward; i.e., highest delivered packets and energy. Moreover, the obtained reward has the smallest gap to the optimal/exact reward.

Index Terms

Wireless Power Transfer, Markov Decision Process, RF Energy Harvesting, Transmission Policies

1. Introduction

Currently, Access Points (APs) in Wireless Local Area Networks (WLANs) are tasked with delivering data to/from devices. In the foreseeable future, APs will also charge low-power devices via Radio Frequency (RF) [1]. This is because works such as [2] and [3] have demonstrated low-power sensor nodes with a

camera that can harvest energy from APs. One can thus imagine sensor nodes being used in buildings and factories to monitor one or more targets [4]. Moreover, as APs are likely to be deployed densely in the future, see [5] and [6], they are ideal energy transmitters. Consequently, we believe APs will help proliferate low-power devices that form the emerging Internet of Things (IoTs), which will play a pivotal role in improving the efficiency of existing industries.

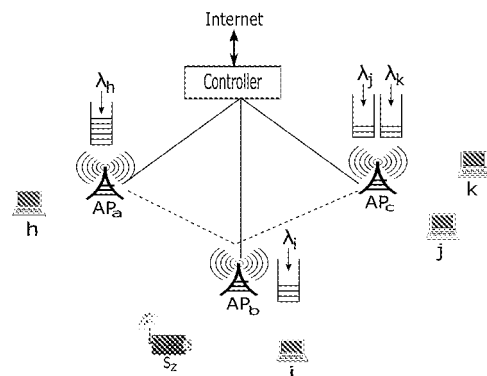


Figure 1. An example dense WLAN with an RF-charging sensor node S_z . Dotted lines indicate interference between APs.

Figure 1 shows a WLAN. In addition to delivering data to their respective station(s), the APs are responsible for charging nearby sensor nodes; e.g., S_z . Also shown is a controller, e.g., a Cisco 8540 wireless controller. It is responsible for both data and control plane operation [7]. Critically, it is responsible for determining APs that interfere with one another, as noted by the dotted lines, and scheduling the transmissions of APs; i.e., the controller ensures APs that interfere do not transmit in the same time slot. Note that link or transmission scheduling will be critical given that APs are likely to interfere when they are densely deployed. This is because of the limited number of

non-overlapping channels that can be assigned to APs. Lastly, as shown by the authors of [7], a controller is necessary to schedule the transmissions of hidden and exposed terminals.

The feasibility of RF-energy harvesting and dense deployment of APs have spurred the development of joint charging and data transmission schemes. These schemes aim to supply energy to devices in order to maximize their throughput. Examples include “harvest-then-transmit” approaches, where an AP first charges devices wirelessly. These devices then set their transmission rate to the AP based on the amount of harvested energy. In [8], the authors determine the energy harvesting time that maximizes the minimum transmission rate. The same problem is then extended to cases where the AP has Multiple-Input Multiple-Output (MIMO) capability [9]; for other extensions, please see [1] and references therein. In [10] the authors propose to adapt an AP’s energy beamforming vector, transmission time and power of devices to ensure the queue at these devices remain stable. We emphasize that these prior works on wireless powered communication networks assume one AP and do not consider *multiple* interfering APs scenarios; e.g., in Figure 1, all APs are on the same channel, meaning AP_a and AP_b cannot transmit simultaneously. Also, these works do not consider queue dynamics at APs. This is important as APs are responsible for delivering data and the controller or link scheduler must ensure the queues at APs remain short. In a different example, the authors of [11] design a signaling protocol that pairs sinks and sensor nodes, either for charging or data collection; both of which are carried out simultaneously. They, however, did not consider data delivery from APs. In [2], the authors prototype sensor nodes that harvest energy from APs. They observe that APs traffic load is too low to charge sensor nodes. A solution is to have APs transmit *power* or dummy User Datagram Protocol (UDP) packets when their queue occupancy is low. The authors of [12] jointly optimize routing and link schedule over a multi-hop wireless network to ensure flow demands and energy requirement of energy harvesting sensor nodes are met. Lastly, in [13] and [14], sensor nodes send charging requests to APs or Energy Transmitters (ETs). The key problem addressed in both works is to pick the set of ETs/APs responsible for charging sensor nodes and investigating the impact charging operation has on data transmissions. The foregone works, however, did not consider the problem of scheduling the *data* transmissions of APs in order to maximize delivered energy and data.

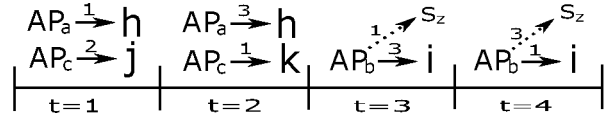


Figure 2. An example schedule for the WLAN shown in Figure 1 with $T = 4$. The number next to each arrow denotes the maximum number of packets or energy transfer afforded by the channel at a given time.

Different from existing works, we consider the following *novel* problem. Referring to Figure 1, each AP has random exogenous data packet arrivals. Sensor node S_z relies on AP_b for energy, where its energy level is directly linked to AP_b ’s transmission duration, path loss and energy conversion efficiency α . In this example, we have the following transmission or independent sets $a_i := [AP_a, AP_b, AP_c]$: $a_1 := [1, 0, 1]$ or $a_2 = [0, 1, 0]$, where a ‘1’ indicates a transmitting AP. Assume time is slotted, where each slot lasts for 100ms. One possible schedule is to activate a_1 and a_2 for 500ms (five slots) each. This means for a received power P_{bz} , sensor node S_z will harvest $\frac{1}{2}\alpha P_{bz}$ of energy. Although this schedule activates APs equally, it may cause long queues; this occurs if the active time or service rate of an AP, say a , is smaller than λ_h . Conversely, a schedule that favors a_1 may starve S_z of energy.

Henceforth, the controller must determine an appropriate transmission schedule for APs. The schedule derived by the controller determines the amount of energy delivered to sensor nodes around APs. A key challenge, however, is that packet arrivals to each AP and the channel condition to each device is random. Another challenge is that the delay between the controller and APs cause sub-optimal scheduling; see [7]. Consequently, as advocated by the authors of [7], the scheduler/controller must operate in terms of *epoch* or a time interval with $1, \dots, T$ slots. That is, the controller computes and sends a link schedule every *epoch*. The schedule specifies the set of transmitting APs in each time slot. One example schedule is shown in Figure 2. Also shown is the maximum number of data packets or energy that can be transferred in each time slot to each device. In practice, these values are random. Our *problem* is thus to determine a transmission schedule over T time slots subject to random packet arrivals and channel gains that optimizes the following metrics: (i) data packets transferred, and (ii) energy harvested by sensor or energy harvesting nodes. For example, if we assume all APs are saturated, i.e., more packets than they can transmit over a channel, and define a so called *reward* to be the sum of packets

and energy transmitted, then the schedule depicted in Figure 2 with length $T = 4$ has reward 15. Note, as the channel conditions and packet arrivals are random, we will need to maximize the *expected* reward.

Next, Section 2 presents key notations. Then in Section 3, we present a finite-horizon Markov Decision Process (MDP) that models the novel problem at hand. Its states capture the random queue lengths and channel conditions over horizon T . Using the MDP model, our goal is to determine a rule, and in turn the schedule in each time slot, that maximizes the total *expected* reward in a given epoch with T slots. We study the following queue selection rules: (i) *max weight*, where APs select the queue with the biggest product value between queue length and channel condition, (ii) *max queue*, where APs select the station with the longest queue, (iii) *max channel state information (CSI)*, where APs select the station with the best channel, and (iv) *random*, where APs randomly return a queue to service. Once APs have chosen a queue as per one of the above rules, we then employ a *policy* that selects the transmission set that maximizes the expected reward in each state. The reward is proportional to the total number of packets and energy that can be delivered in each time slot. Our results show that in comparison to other rules, the schedule, i.e., set of transmitting APs in each epoch, determined by the max weight rule yields high packets and energy delivery rate. Moreover, the max weight rule yields the smallest gap to the optimal *expected* reward.

2. Preliminaries

We denote the set containing APs as \mathcal{A} ; all of which operate on the same channel. A controller connects all APs and is aware of stations associated to each AP, and nearby sensor or energy-harvesting nodes. It also computes the schedule of each epoch, using our approach as detailed in Section 4, and informs all APs their transmission schedule every T slots. Each slot is indexed by t . APs serve two types of nodes: (i) *stations*, which we record in the set \mathcal{S}_d . Stations transmit and receive data to/from their associated AP. They do not rely on the APs for power. Those associated to AP a are placed in set $\mathcal{S}_d(a)$, and (ii) the set of *sensor nodes* \mathcal{S}_e . Those “near” AP a are denoted as $\mathcal{S}_e(a)$, meaning when AP a transmits, these sensor nodes are able to harvest energy because the received power is higher than a given received sensitivity; e.g., for the energy harvesters reported in [1], an RF input power in the range -14 to -22 dBm is sufficient to produce 1V DC output. Note, sensor nodes are able to harvest energy from one AP only. This is reasonable given the

low receiver sensitivity of current energy harvesting implementation that restricts the charging distance to no more than five meters [2]. We do not consider uplink traffic from stations and sensor nodes, and leave it as a future work.

APs may interfere due to the limited number of orthogonal channels in IEEE 802.11-based WLANs. To represent the *interference* between APs, we use a conflict graph, whereby its vertices correspond to APs [15]. Note that a controller can use IEEE 802.11k to determine which APs interfere with one another. If two APs interfere, i.e., they hear each other’s transmissions or an associated station hears transmissions from the other AP, then there is an edge in the conflict graph between the two APs. Note that the conflict graph of a WLAN is fixed because APs are static. With the conflict graph in hand, we can then derive the collection of transmissions or independent sets; see Section 5 for an example heuristic. Each of these sets forms the columns of a matrix \mathbf{A} with elements taking a value of zero or one. For example, for the conflict graph of Figure 1, namely AP_a - AP_b - AP_c , two possible transmission sets are $[1\ 0\ 1]^T$ and $[0\ 1\ 0]^T$; i.e., either AP_a and AP_c transmit together or AP_b transmits by itself. Note, there can be up to $2^{|\mathcal{A}|}$ independent sets and finding the maximum independent set is an NP-hard problem. In the sequel, $\tau(t)$ refers to the transmission set activated at time t . Also, with a slight abuse of notation we will also treat \mathbf{A} as a set.

Each AP has a First-In-First-Out (FIFO) queue for each associated station. Formally, at AP a , the queue that stores packets headed to station j at time t is $Q_{aj}(t)$. Let $A_{aj}(t)$ be the packet arrivals at time t for station j . Packets arrive at the start of each slot as per an i.i.d process with mean λ_j .

We model the amount of harvested energy and the number of packets that can be transmitted to a node at a given time as follows. The channel state evolves according to a finite-state Markov chain and is static for the duration of a time slot. Moreover, the channel state to each station is independent and is known to the controller. We write $\alpha_{az}(t)$ to represent the number of packets that can be received by station z from AP a at time t . On the other hand, $\varepsilon_{az}(t)$ refers to the potential energy (in Joules) that can be harvested by sensor node z at time t for each packet transmitted by AP a . As explained in Section 5, the amount of energy (Joules), which is a random value, a sensor node can harvest within five meters is drawn from the measured data obtained in [2]. This random value or a state of the Markov chain thus represents the variable received power and the resulting energy harvested after accounting for conversion efficiency.

In each time slot t , we assume APs select the *best* station according to some rule; see Section 4. Let this station be j^* . Assume AP $k \in \tau(t)$ is scheduled to transmit. Its total number of transmitted packets is,

$$T_k(t) = \text{MIN}(Q_{kj^*}(t), \alpha_{kj^*}(t)) \quad (1)$$

The queue for each station j at AP k evolves as,

$$Q_{kj}(t+1) = Q_{kj}(t) + A_{kj}(t) - T_k(t)\mathbb{1}_{j=j^*} \quad (2)$$

In words, for each station j , we sum its current queue length plus any arrivals. For station j^* , we subtract transmitted packets as the indicator function $\mathbb{1}_x$ returns a value of one when j equals j^* .

Lastly, let $X_a(t)$ be the total queue length of AP a . Formally, it evolves as per,

$$X_a(t+1) = \sum_{j \in S_d(a)} Q_{aj}(t+1) \quad (3)$$

We ignore battery capacity because the energy delivered by an AP is a few orders of magnitude smaller than the battery capacity of sensor nodes; e.g., two AA batteries, as commonly used by sensor nodes, is capable of storing tens of kilo-joules of energy versus a recharging rate of micro-joules.

3. A Markov Decision Process Model

Briefly, an MDP [16] is specified by the tuple $(\mathbf{Y}, \mathbf{A}, P(\cdot|\cdot, \cdot), r(\cdot, \cdot))$, where \mathbf{Y} denotes the state space and \mathbf{A} is the action space. In state $\mathbf{y} \in \mathbf{Y}$, if we take action $a \in \mathbf{A}$, then we earn a reward $r(\mathbf{y}, a)$. After that, we move to a new state $\mathbf{y}' \in \mathbf{Y}$ with probability $P(\mathbf{y}'|\mathbf{y}, a)$, where $P(\mathbf{y}'|\mathbf{y}, a) \geq 0$ and $\sum_{\mathbf{y}' \in \mathbf{Y}} P(\mathbf{y}'|\mathbf{y}, a) = 1$.

We are now ready to instantiate a MDP for the problem at hand. At time t , the queue length at all APs is recorded as $Q_t = (|Q_{aj}(t)|)_{a \in \mathcal{A}, j \in S_d}$. The channel states, from an AP to sensor nodes and stations at time t , are stored in $C_t = (\alpha_{aj})_{a \in \mathcal{A}, j \in S_d} \cup (\alpha_{aj})_{a \in \mathcal{A}, j \in S_e}$. The current state at time t is denoted as $\mathbf{y}_t = (Q_t, C_t)$. Formally, we have a discrete-time Markov chain $\{\mathbf{y}_t\}_{t=0}^{\infty}$. Lastly, we refer to $\mathbf{Y} = (Q, C)$ as the state space, where Q and C are the possible queue lengths and channel states, respectively. Observe that the state space has a high dimensionality. For example, if the maximum queue length at each AP is 100 packets and there are 10 possible channel states, then in total we have $100^{|\mathcal{A}|} \times 10^{|\mathcal{S}_d|}$ states.

The action space corresponds to all the transmission sets in matrix \mathbf{A} . The transition probability to a new state \mathbf{y}_{t+1} , i.e., $P(\mathbf{y}_{t+1} | \mathbf{y}_t, \tau(t))$, given state \mathbf{y}_t and action $\tau(t)$, is determined by both the Markov model

that dictates the channel condition of each link and also the packet arrival process.

We now define the reward $r_t(\mathbf{y}_t, \tau(t))$. First, let the total number of packets transmitted by APs at time t be,

$$T(t) = \sum_{k \in \tau(t)} T_k(t) \quad (4)$$

At time t , the total amount of energy delivered is

$$E(t) = \sum_{k \in \tau(t)} \sum_{j \in S_e(k)} \varepsilon_{kj}(t) T_k(t) \quad (5)$$

The reward is then defined as,

$$r_t(\mathbf{y}_t, \tau(t)) = \gamma_d T(t) + \gamma_e E(t) \quad (6)$$

where γ_d and γ_e are scalar factors to normalize the value of $T(t)$ and $E(t)$. Observe that the reward is tied closely to both the queue and channel state of the station chosen by each AP as well as the number of transmitting APs in $\tau(t)$. Given T time slots, our problem is to determine a *policy* π that returns the best transmission set from \mathbf{A} for each of the next T slots such that the expected reward is maximized. Formally, we have,

$$\max_{\pi} \mathbb{E} \left[\sum_{t=0}^{T-1} r_t(\mathbf{y}_t, \pi(\mathbf{y}_t, \mathbf{A})) \right] \quad (7)$$

where the expectation \mathbb{E} is taken with respect to random channel conditions and queue arrivals. Note, in practice, in addition to the issues highlighted in [7], the exact value of T needs to be balanced against signaling overheads and computation time. If T is small, then frequent commands to APs may cause congestion. On the other hand, if T is big, then the computation time will be long as the problem has high dimensionality.

4. Approximate DP and Queue Selection Rules

We employ approximate DP given the high dimensionality of the problem at hand. Specifically, we use forward dynamic programming, whereby for a given starting state $\mathbf{y}_0 \in \mathbf{Y}$ and discount factor γ , the problem as formulated by (7) is equivalently to computing the following value function [16],

$$V_t(\mathbf{y}_t) = \max_{\tau(t) \in \mathbf{A}} \left[r_t(\mathbf{y}_t, \tau(t)) + \gamma \sum_{\mathbf{y}_{t+1} \in \mathbf{Y}} P(\mathbf{y}_{t+1} | \mathbf{y}_t, \tau(t)) V_{t+1}(\mathbf{y}_{t+1}) \right] \quad (8)$$

A key problem is computing V_t due the size of \mathbf{Y} . To this end, we use approximate value iteration; see [16]. Specifically, we randomly generate a sample of N outcomes from \mathbf{Y} . Let the set Ω stores these N outcomes, and $p(\omega)$ be the probability of outcome $\omega \in \Omega$. We then rewrite Equ. (8) as,

$$\bar{V}_t(\mathbf{y}_t) = \max_{\tau(t) \in \mathbf{A}} \left[r_t(y_t, \tau(t)) + \gamma \sum_{\omega \in \Omega} P(\omega) \bar{V}_{t+1}(\omega) \right] \quad (9)$$

Implicitly in the calculation of V_t (or \bar{V}_t) is that each AP has chosen the best station j^* . In this paper, we study the following rules. The first is called *Max Weight*. At each time t , AP a selects station j^* as follows,

$$j^* = \arg \max_{j \in S_a(a)} \alpha_{aj}(t) Q_{aj}(t) \quad (10)$$

That is, pick the station that has the longest queue and best channel. The second rule called *Max Queue* corresponds to APs picking only the longest queue, i.e., identify the station j with the biggest $Q_{aj}(t)$ value. The third rule, aka *Max CSI*, requires APs pick the station with the biggest $\alpha_{aj}(t)$ value. Lastly, the *Random* rule returns a data queue randomly.

For each of the above rules, at each time t , we employ the following *policy* π : in each state \mathbf{y}_t , select the transmission set $\tau(t)$ from \mathbf{A} that maximizes Equ. (9). That is, select a column from \mathbf{A} that maximizes $\bar{V}_t(\mathbf{y}_t)$.

We remark that for a given matrix \mathbf{A} , which contains a finite set of actions, and queue selection rule, the said policy is optimal. In particular, as there are finite number of transmission sets, i.e., columns of matrix \mathbf{A} , there is an optimal action that maximizes expression $\bar{V}_t(\mathbf{y}_t)$ for each t in each epoch [16]. A key issue, however, is the gap between $\bar{V}_t(\mathbf{y}_t)$ and $V_t(\mathbf{y}_t)$. In Section 5, we show for T values less than 10, the value of $\bar{V}_t(\mathbf{y}_t)$ is at most 25% higher than $V_t(\mathbf{y}_t)$.

We note that constructing matrix \mathbf{A} involves finding maximum independent sets; an NP-hard optimization problem. Thus we use a heuristic, see Section 5, when generating transmission sets. Note, other heuristics can be used to construct maximal independent sets, which may yield a bigger or smaller WLAN capacity region; i.e., transmission sets with more or fewer active APs. In other words, the new heuristic only scales our results and does not alter our conclusions.

5. Evaluation

Using the formulated MDP, we now study the foregone rules as follows. We first study the following scenario. There are $|\mathcal{A}| = 10$ APs; each of which

is randomly assigned up to five sensor nodes and five stations. In each simulation run, a new topology is generated; i.e., one with a different conflict graph, and APs have a new set of stations and sensor nodes. The channel state at each time t is driven by a Markov model. For sensor nodes, each state is the amount of harvested energy (in μJ) and is drawn from $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. These values are based on the measurement data reported in [2]. As for data packets, each state corresponds to the data rates of IEEE 802.11g. Each state has a uniform probability to enter another state. At each time t , packet arrivals follow a Bernoulli process. When approximating the value of a given state, i.e., $\bar{V}_t()$, we sample $N = 100$ states. We set $T = 10000$ and calculate the average queue length and harvested energy over 20 runs. The discount factor γ is set to one. We recorded the average queue length of APs and also the energy harvested by all sensor nodes.

We derive matrix \mathbf{A} as follows. Each AP has a unique identity (ID). For a given transmission set τ , we start with an AP a with the lowest ID. We then greedily add another AP into the set τ if it does not interfere with APs in τ . This continues until no more APs can be added. We then add τ into \mathbf{A} if it is new. Otherwise, we construct a new τ and repeat the process by starting from the AP with the next highest ID.

Figure 3 shows that the max weight rule is able to support the highest arrival rates. This is because APs pick the longest queue that can fully take advantage of the channel condition at each time t . In contrast, rules such as random and max queue may choose a station with few packets or poor channel. In terms of harvested energy, see Figure 4, when the load is high, max CSI allows APs to transmit the highest number of packets, and thereby, allow sensor nodes to harvest more energy. At lower loads, the max weight rule may cause APs to choose a station with few packets despite having good channel conditions. Hence, the actual number of transmitted packets is low. Lastly, we observe that larger transmission sets, those with more transmitting APs, are preferred because they yield higher total expected reward. Hence, some APs may receive fewer transmission opportunities in the short term.

Next, we consider inter-APs interference. The traffic load is fixed at 0.5. When APs are less likely to interfere, more APs transmit simultaneously. With increasing interference, only one AP out of $|\mathcal{A}|$ may transmit. Figure 5 shows that the max weight rule has the best performance due to the previously discussed reasons. The amount of harvested energy decreases when more APs interfere with one another; see Figure

6. This is because each AP transmits infrequently. At a traffic load of 0.5, APs usually have packets awaiting transmission. Thus, both the max weight and CSI rules are able to fully exploit good channel conditions. In contrast, the random and max queue rules may select a station with poor channel condition.

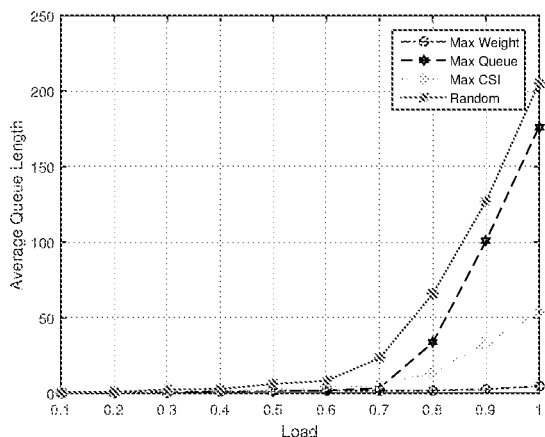


Figure 3. Average queue length versus probability of arrivals or load

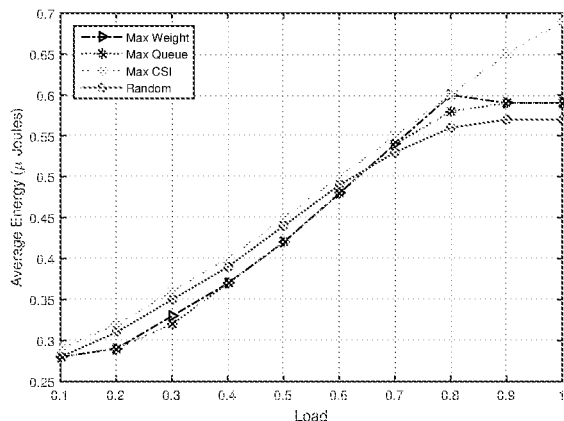


Figure 4. Average harvested energy versus probability of arrivals or load

Lastly, we study the case where Equ. (8) can be solved exactly as opposed to being approximated; see Equ. (9). Note that finding a good method that has minimal or no gap between the approximate and exact value remains an open research question; see [17]. In our case, we also have the added complexity of deriving the set of actions or transmission sets; see Section 4. We consider a small scenario with two interfering APs. Hence, there are only two transmission sets, each with one transmitting AP. Each AP has two

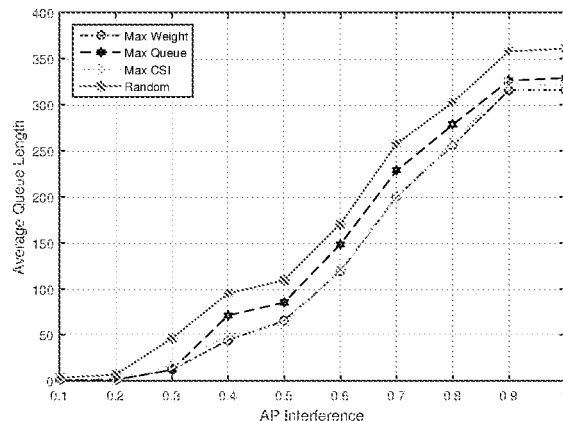


Figure 5. Average queue length versus probability of interference between APs

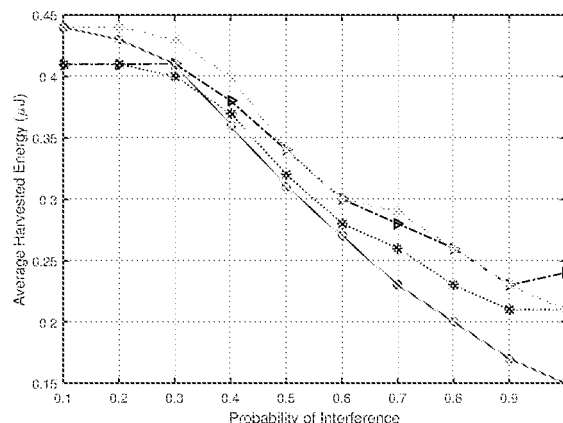


Figure 6. Average harvested energy versus interference between APs

stations and one sensor node. The queue length takes on three states: $\{0,1,2\}$. There are two channel states, namely $\{1,2\}$ and in each state, there are either zero, one or two packet arrivals. In total we have 5184 states. When computing Eq. (9), we set $N = 1000$. Figure 7 shows the gap from the exact expected reward value. With increasing T , the gap increases and it is advantageous to keep T small; this ensures a smaller gap and also fast computation time. Advantageously, we see that the max weight rule has the smallest gap. Note that for a given T , the gap is a constant. Note, we have tried higher values of N , at the expense of longer computation time, but the results remain similar. Figure 8 shows the exact reward value obtained by the tested rules. The trend is consistent with earlier results whereby the max weight rule has the best expected reward.

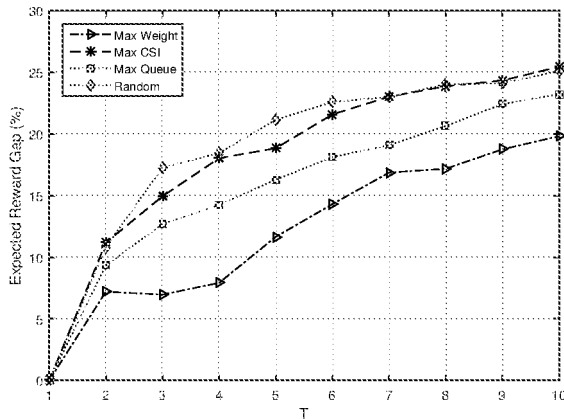


Figure 7. Average gap between approximated and exact value

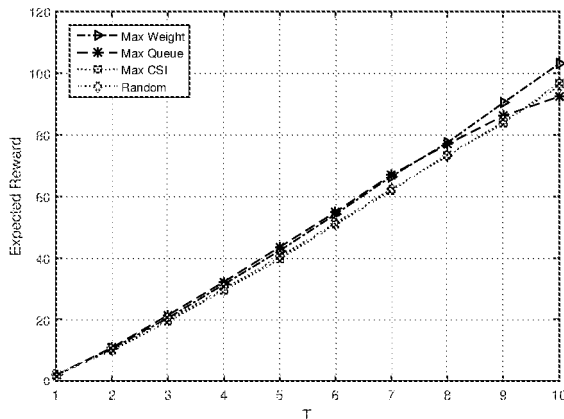


Figure 8. Exact reward attained by tested rules

6. Conclusion

Future dense deployment of APs are likely to be used to charge nearby energy-harvesting devices and also deliver data to stations simultaneously. To this end, we use a MDP to study various rules and show that the max weight rule ensures APs have short queues whilst ensuring energy-harvesting devices receive ample energy. Moreover, it yields the smallest gap between the approximate and exact value.

References

- [1] X. Lu, P. Wang, D. Niyato, D.-I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: A contemporary survey," *IEEE Communication Surveys and tutorials*, vol. 17, no. 2, pp. 757–767, 2015.
- [2] V. Talla, B. Kellogg, B. Ransford, and S. Naderiparizi, "Powering the next billion devices with Wi-Fi," in *ACM CoNEXT*, Germany, Dec. 2015.

- [3] S. Naderiparizi, Z. Kapetanovic, and J. R. Smith, "WISPCam: An RF-powered smart camera for machine vision applications," in *ACM 4th Intl Wksp on Energy Harvesting and Energy-Neutral Sensing Systems*, Stanford, CA, USA, Nov. 2016.
- [4] C. Yang and K.-W. Chin, "On nodes placement in energy harvesting wireless sensor networks for coverage and connectivity," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 27–36, Feb. 2017.
- [5] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANs," *IEEE wireless Comms*, vol. 23, no. 1, pp. 38–46, 2016.
- [6] F. Debele, M. Meo, D. Renga, M. Ricca, and Y. Zhang, "Designing resource-on-demand strategies for dense WLANs," *IEEE JSAC*, vol. 62, no. 12, pp. 2494–2509, Dec. 2014.
- [7] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "CENTAUR: Realizing the full potential of centralized hybrid WLANs through a hybrid data path," in *ACM MOBI-COMN*, Beijing, China, Sep. 2009.
- [8] H. Ju and R. Zhang, "Throughput maximization for wireless powered communications networks," *IEEE Trans. on Wireless Communications*, vol. 13, no. 1, pp. 418–428, Jan. 2014.
- [9] L. Liu, R. Zhang, and K.-C. Chua, "Multi-antenna wireless powered communication with energy beamforming," *IEEE Transactions on Communications*, vol. 62, no. 12, pp. 4349–4352, Dec. 2014.
- [10] K. Kim and K. W. Choi, "Stochastic optimal control for wireless powered communications networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 1700–1711, Jan. 2016.
- [11] C. Shao, H. Roh, T. Kim, and W. Lee, "Multisource wireless energy harvesting-based medium access control for rechargeable sensors," *IEEE Trans. on Consumer Elec.*, vol. 62, no. 2, pp. 119–127, May 2016.
- [12] K.-W. Chin, L. Wang, and S. Soh, "Joint routing, data and power links scheduling in two-tier RF-energy harvesting networks," *IEEE Communications Letters*, vol. 20, no. 9, pp. 1864–1867, Sep. 2016.
- [13] M. S. I. Khan, J. Mistic, and V. B. Mistic, "Impact of network load on the performance of a polling MAC with wireless recharging nodes," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 307–317, Sep. 2015.
- [14] M. Y. Naderi, P. Nintanavongsa, and K. Chowdhury, "RF-MAC: A medium access control protocol for rechargeable sensor networks powered by wireless energy harvesting," *IEEE Trans. on Wireless Comms*, vol. 13, no. 7, pp. 3926–3936, 2014.
- [15] X. Zhou, Z. Zhang, G. Wang, X. Yu, B. Y. Zhao, and H. Zheng, "Practical conflict graphs in the wild," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 824–835, Jun. 2015.
- [16] M. L. Putterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
- [17] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley and Sons, 2011.