RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks

Zhenzhen Liu* and Itamar Elhanany

Department of Electrical and Computer Engineering,
University of Tennessee,
Konxville, TN 37996-2100, USA
E-mail: itamar@ece.utk.edu E-mail: zliu4@utk.edu
*Corresponding author

Abstract: This paper introduces RL-MAC, a novel adaptive Media Access Control (MAC) protocol for Wireless Sensor Networks (WSN) that employs a reinforcement learning framework. Existing schemes centre around scheduling the nodes' sleep and active periods as means of minimising the energy consumption. Recent protocols employ adaptive duty cycles as means of further optimising the energy utilisation. However, in most cases each node determines the duty cycle as a function of its own traffic load. In this paper, nodes actively infer the state of other nodes, using a reinforcement learning based control mechanism, thereby achieving high thoughput and low power consumption for a wide range of traffic conditions. Moreover, the computational complexity of the proposed scheme is moderate rendering it pragmatic for practical deployments.

Keywords: Media Access Control (MAC); energy-efficient protocols; Wireless Sensor Networks (WSN); reinforcement learning.

Reference to this paper should be made as follows: Liu, Z. and Elhanany, I. (XXXX) 'RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks', *Int. J. Sensor Networks*, Vol. X, No. Y, pp.XXX–XXX.

Biographical notes: Zhenzhen Liu received a BSc in Telecommunications at Nanjing University of Post and Telecommunications. Currently, she is pursuing a PhD in Computer Engineering at the University of Tennessee, Knoxville. Her research interests include wireless sensor networks, reinforcement learning and high-performance machine learning architectures.

Itamar Elhanany received a BSc, an MSc and a PhD in Electrical and Computer Engineering and an MBA, all from the Ben-Gurion University in Israel. Currently, he is an Assistant Professor of Electrical and Computer Engineering at the University of Tennessee, where he directs the networking research group. His research interests include packet switching architectures, scheduling algorithms, wireless sensor networks, machine learning and performance analysis.

1 Introduction

Recent advancements in the design and fabrication of low-power VLSI circuitry, as well as wireless communications, have broadened the applications prospect for Wireless Sensor Networks (WSN). The latter promise to revolutionise our ability to sense and control diverse physical environments using large numbers of small, inexpensive devices that integrate sensing, computation and communication. These sensors can collaborate with each other and achieve complex information gathering and dissemination tasks such as infrastructure security, environment and habitat monitoring, industrial sensing and traffic control.

In addition to the many unique characteristics of WSN that stem from the resource-constrained environments in which they operate, many applications, whereby collaborative processing is carried out, necessitate the ad hoc formation of node clusters. These clusters of nodes typically form in the proximity of an event and since the location and extent of the event are often unknown a priori, cluster members are decided upon in an ad hoc manner.

WSN nodes are battery-powered and cannot typically be recharged after deployment. Due to the inherent limitation of using batteries as power sources, the only way to extend the life span of a sensor node is by using energy efficient protocols and mechanisms that make judicious use of the energy resources. One such key component is the Media Access Control (MAC) protocol employed, which is responsible for coordinating the nodes' access to the medium – an essential function for all shared-medium networks. Consequently, energy efficiency is the most important attribute of a good MAC protocol for sensor networks. In addition to energy efficiency, throughput and latency are also viewed as primary performance metrics attributed to MAC protocols designed for WSN.

It is well acknowledged that wireless communications is the most energy consuming component in a sensor node. A radio can be in one of four modes: transmit, receive, idle listening (in which the radio is on but idle) and sleep. Energy cost in idle listening is almost identical to that in the receive mode, while the consumption in sleep mode is significantly lower than that of receiving.

It is also known that the largest contribution to energy waste in MAC protocols is uneventful idle listening. Since a node has no explicit knowledge of when packets are sent for it from one of its neighbours, it must consistently keep its radio in listening mode. To address this challenge and reduce the energy waste due to idle listening, several MAC protocols suited for sensor networks have emerged, including S-MAC (Ye et al., 2004) and T-MAC (Dam and Langendoen, 2003). These protocols incorporate some form of duty-cycle management that periodically sets each of the nodes in sleep mode so as to minimise the power consumption. However, in most protocols each node determines the duty cycle as a function of its own traffic load, thereby inherently limiting the overall performance of the network.

The main contributions of this paper are as follows. We propose an optimisation framework that generally captures several parameters pertaining to the dynamics of the MAC layer and develop a practical algorithm, based on Reinforcement Learning (RL) formalism, to learn a near-optimal MAC protocol policy. A key component lies in the fact that nodes, in addition to taking into consideration their individual traffic load, actively infer the states of other nodes. The broad adaptability exhibited by the scheme is coherent with the nature of WSN deployments, in which topology and location are dynamic. The proposed optimisation scheme is simple, inherently distributed and self-organised. It may be noted that this work complements other work recently carried out which exploits RL techniques in wireless sensor network platforms, such as the work by Pandana and Liu (2005) on physical-layer power optimisation using RL.

The rest of this paper is organised as follows. In Section 2, we review the Markov Decision Process (MDP) and its optimal solution, followed by an overview of the RL algorithm employed. In Section 3, we formulate the MAC layer protocol described along with its objective function in the context of throughput maximisation and energy consumption minimisation. Section 4 presents simulation results that accentuate the distinct advantages of the proposed approach, in particular under high traffic load conditions. Finally, in Section 5, the conclusions are drawn.

2 Related work

Recently, several adaptive MAC protocols have been proposed for WSN, including the Timeout-MAC (T-MAC) (Dam and Langendoen, 2003) and Pattern-MAC (P-MAC) (Zheng et al., 2005). These protocols aim at expending the main theme introduced by S-MAC in order to better utilise the available node resource, while optimising channel usage. However, coarsely speaking, in most protocols a node aims to perform better by predominantly considering its own state information. Here, we propose to have nodes indirectly infer the state of other nodes, as an inherent part of their decision-making process. By doing so, the overall network media access efficiency increases.

As means of reducing the energy wasted during periods of idle listening, several MAC protocols have emerged, which can be categorised into either being contention-free based or contention-based. A typical contention-free approach is TDMA-based protocols, which are naturally energy preserving as they prescribe a deterministic duty cycle and do not suffer from collisions (Havinga and Smit, 2000). However, allocating TDMA slots is a complex problem and maintaining a TDMA schedule in an ad hoc environment is an intricate task that necessitates complex, resource-intense logic in the nodes. Since TDMA schemes divide time into very small slots, the effect of clock drift is fatal, yielding a need for exact timing. Furthermore, the idle slots directly result in waste of channel bandwidth and energy.

S-MAC is a single-frequency contention-based MAC protocol that was specifically designed for wireless sensor network. It divides time into relatively large frames. Every frame has two phases consisting of an active state (Listen state) and a sleeping state. A node turns off its radio in the sleep state so as to preserve energy and exchanges data packets with its neighbours in the active states. The main drawback of S-MAC is that it uses preset fixed duty cycles for all the sensor nodes. As such, it cannot adapt well to changes in the network load and topology dynamics. Moreover, it induces substantial energy waste due to instances whereby several nodes require significantly higher duty cycle than others, such as those that are located near the sink node in data gathering applications.

The T-MAC protocol borrows the virtual clustering method of S-MAC to synchronise nodes. In contrast to S-MAC, it operates with fixed length slots (615 ms) and uses a time-out mechanism to dynamically determine the end of the active period. The time-out value (15 ms) is set to span a small contention period and an RTS/CTS exchange. If a node does not detect any activity (an incoming message or a collision) within the time-out interval, it can safely assume that no neighbour wants to communicate with it and goes to sleep. On the other hand, if the node engages or overhears a communication, it simply starts a new time-out after that communication finishes.

To save energy, a node turns off its radio while waiting for other communications to finish (overhearing avoidance). Transmission occurs at the beginning of each frame, while transitioning into sleep mode occurs if no communication are sensed for a duration of time denoted by T_A. The adaptability attributed to T-MAC allow it to improve performance when compared to S-MAC. However, two main drawbacks characterise the T-MAC protocol. Firstly, due to the aggressive concentration of traffic transmission attempts at the beginning of each frame, a node that loses contention is forced to sleep and prohibited from transmitting for the entire frame. Notably, at high traffic loads the forced sleep problem has a high probability of occurring, thus significantly limiting the data throughput and increasing latency when the traffic load is heavy. A second drawback, pertaining to both S-MAC and T-MAC, is that they group communications during relatively short periods of activity, causing performance to significantly degrade under high traffic conditions.

The P-MAC (Zheng et al., 2005) protocol attempts to further improve performance by employing patterns of schedules as means of minimising the idle listening periods,

which are a primary source of energy wastage. P-MAC adapts a node's sleep-wakeup schedules according to its own traffic and that of its neighbours. However, two primary drawbacks are introduced first by P-MAC: the underlying assumption is that knowledge of anticipated channel usage requirements are available in advance. This is a somewhat questionable assumption when considering highly dynamic WSN applications, such as high speed target tracking. Moreover, the computational complexity of the algorithm is quite high thereby limiting its applicability to resource-constraint sensor platforms.

3 Markov decision process and RL methods

An MDP (Bertsekas and Tsitsiklis, 1996) is defined as a (S, A, P, R) tuple, where S stands for the state space, A contains all the possible actions at each state, P is a probability transition function $S \times A \times S \rightarrow [0, 1]$ and R is the reward function $S \times A \rightarrow R$. Also, we define π as the decision policy that maps the state set to the action set: $\pi : S \rightarrow A$. Specifically, let us assume that the environment is a finite-state, discrete-time stochastic dynamic system. Let the state space S be $S = (s_1, s_2, ..., s_n)$ and, accordingly, action space A be $A = (a_1, a_2, ..., a_m)$. Suppose at episode k, the RL agent detects $s_k = s \in S$, the agent chooses an action $a_k = a \in A(s_k)$ according policy π in order to interact with its environment. Next, the environment transitions into a new state $s_{k+1} = s^{'} \in S$ with the probability $P_{ss'}(a)$ and provides the agent with a feedback reward denoted by $r_k(s, a)$. The process is then repeated. The goal for the RL agent is to maximise the expected discounted reward or state-value, which is represented as

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k} r_k(s_k, \pi(s_k))|s_0 = s\right\} \quad (1)$$

where $\gamma(0 \leq \gamma < 1)$ is the discount factor and $E_{\pi}\{\}$ denotes the expected return when starting in s and following policy π thereafter. The equation above can be rewritten as

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s^{'} \in S} P_{ss'}(\pi(s)) V^{\pi}(s^{'}) \quad (2)$$

where $R(s, \pi(s)) = E\{r(s, \pi(s))\}$ is the mean value of the reward $r(s, \pi(s))$.

However, in many practical scenarios, as in our case, the transition probability $P_{ss'}(a)$ and the reward function $R(s, \pi(s))$ are unknown, which makes it hard to evaluate the policy π. Q-learning (Sutton and Barto, 1998) is one of the most effective and popular algorithms for learning from delayed reinforcement to determine an optimal policy, in absence of the transition probability and reward function. In Q-learning, policies and the value function are represented by a two-dimensional lookup table indexed by state-action pairs. Formally, for each state s and action a, we define the Q value under policy π to be:

$$Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s^{'} \in S} P_{ss'}(a) V^{\pi}(s^{'}) \quad (3)$$

as the expected discounted reward starting from s, taking the action a and thereafter following policy π. Therefore, the value function of an optimal policy π^*, denoted by V^*, can be defined as:

$$V^*(s) = V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad (4a)$$

$$= \max_{a \in A(s)} \left(R(s, a) + \gamma \sum_{s^{'} \in S} P_{ss'}(a) V^*(s^{'}) \right) \quad (4b)$$

let $Q^*(s, a) = Q^{\pi^*}(s, a) = \max_{\pi} Q^{\pi}(s, a)$ be the optimal action function under π^*, the optimal value function can be rewritten as

$$V^*(s) = \max_{a \in A(s)} \left(Q^*(s, a)\right) \quad (5)$$

Therefore, we express the optimal policy

$$\pi^*(s) = \arg\max_{a \in A(s)} (Q^*(s, a)) \quad (6)$$

and rewrite $Q^*(s, a)$ as

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s^{'} \in S} P_{ss'}(a) \max_{a^{'} \in A(s^{'})} (Q^*(s^{'}, a^{'})) \quad (7)$$

In the Q-learning process, a learned action value function Q directly approximates Q^* through value iteration. Correspondingly, the Q-value updating rule is given by

$$Q_{k+1}(s, a) = \begin{cases} Q_k(s, a) + \alpha\delta & \text{if } s_k = s, a_k = a \\ Q_k(s, a) & \text{otherwise} \end{cases} \quad (8)$$

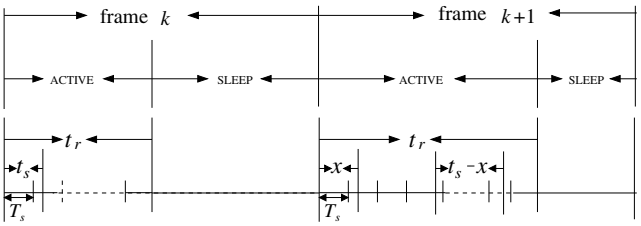
where $\delta = r_k + \gamma \max_{a^{'} \in A(s^{'})} Q_k(s^{'}, a^{'}) - Q_k(s, a)$ is called the *temporal difference* and α is the learning rate.

In Section 4, we formulate the MAC optimisation function, with respect to the energy consumed, as an MDP. We will show that by carefully defining the reward function and the RL algorithm, a near-optimal media access policy is attained.

4 RL-MAC protocol design

The RL-MAC protocol employs a similar frame-based structure to that of S-MAC and T-MAC. A distinct feature of our protocol is that both the frame active time and duty cycle are dynamically modified in accordance with a node's traffic load as well as its incoming traffic characteristics. As illustrated in Figure 1, time is divided into frames while each frame is further divided into finer time slots. The slot length is determined by the channel bandwidth and data packet length. At the beginning of each frame, the RL agent, which acts as the MAC protocol engine, dynamically reserves slots as active times. In active time, a node listens to the channel and attempts to exchange packets with its neighbours. When a reserved active time expires, the node refrains from sending or receiving any data and transitions into the sleep state. We next define the key components of our protocol, with particular emphasis on the RL context underlying its functionality.

Figure 1 Frame structure as employed by the RL-MAC protocol. Adaptive duty cycle allows the protocol to optimise throughput and energy attributes



4.1 Active time reservation

An ideal MAC protocol can derive an optimal media access policy if it has complete knowledge of the node's traffic as well as the traffic pending at neighbouring nodes. The result would be optimal throughput using minimal energy consumption. Since this information is not readily available to the nodes, some approximation should be employed. Our approach relies on this assertion and, accordingly we formulate the active time reservation problem as an MDP. The goals of our RL agent are two-fold. Firstly, it strives to maximise an energy efficiency metric defined as the ratio of effective transmit/receive time to the total reserved active time. Secondly, it is designed to maximise data throughput. We refer to the term throughput as actual payload bytes/sec and do not consider the overhead data exchanged as part of the throughput metric.

4.2 Reward function formulation

We next discuss the reward functions defined and the algorithm used to allow the agent to learn the optimal media access control policy. Energy consumption and throughput are both critical for sensor networks. It would thus not be desirable to minimise energy consumption at the cost of unacceptable throughput. Therefore, the reserved active time should be a function of the node's traffic load conditions. Since the traffic load is highly correlated within a given sensor node neighbourhood, we chose to employ the number of packets queued for transmitting at the beginning of the frame, n_{b} as the state s and to complement that, the reserved active time t_{r} represents the action a generated.

To evaluate the effective transmit/receive time ratio, we record the number of successfully transmitted packets, n_{s} and received packets, n_{r} during the reserved active period. Let this ratio, for a single packet, incorporate the Carrier Sensing/RTS/CTS/ACK elements and be denoted by T_{P}. We further note that by overhearing RTS/CTS messages or collisions, the sensor node transitions into sleep mode for a total period of t_{s}, which occurs during the reserved active time (see Figure 1). T_{s} denotes the slot time, given that the action space is discretised. Furthermore, the number of packets queued for transmission at the start of the next frame $n_{\mathrm{b}}^{'}$ acts as a valid indicator of the effectiveness of the reserved active time.

The reward function comprises of two independent components. The first reflects on the internal state information of each node, while the second embeds indirect information regarding the status of other nodes in the networks. The two reward components are separated by a single time slot such that for a given frame index k,

$$r^k = r_t^k + r_{t+1}^k \tag{9}$$

where the first component reflects on the internal state of a node at time t, while the second on the state of other nodes as perceived at time $t+1$. At frame k, let $s_k = n_{\mathrm{b}}$, $a_k = t_{\mathrm{r}}$, the reward function is formulated as follows:

$$r_t^k(n_{\mathrm{b}}, t_{\mathrm{r}}) = \begin{cases} \frac{(n_{\mathrm{s}}+n_{\mathrm{r}}+1)T_{\mathrm{p}}}{t_{\mathrm{r}}-t_{\mathrm{s}}} - \eta\frac{n_{\mathrm{b}}^{'}-n_{\mathrm{b}}}{\sqrt{B}} & t_{\mathrm{r}}, n_{\mathrm{b}} \neq 0, n_{\mathrm{b}}^{'} > n_{\mathrm{b}} \\ \frac{(n_{\mathrm{s}}+n_{\mathrm{r}}+1)T_{\mathrm{p}}}{t_{\mathrm{r}}-t_{\mathrm{s}}} & t_{\mathrm{r}}, n_{\mathrm{b}} \neq 0, n_{\mathrm{b}}^{'} \leq n_{\mathrm{b}} \\ -\eta\frac{n_{\mathrm{b}}^{'}-n_{\mathrm{b}}}{\sqrt{B}} & t_{\mathrm{r}}, n_{\mathrm{b}} = 0, n_{\mathrm{b}}^{'} \neq 0 \\ 1 & t_{\mathrm{r}}, n_{\mathrm{b}} = 0, n_{\mathrm{b}}^{'} = 0 \end{cases} \tag{10}$$

where B is the buffer size at the MAC layer, and η denotes a weight coefficient.

4.3 Early sleeping avoidance

The primary goal of early sleeping avoidance is adaptation to incoming traffic. An early sleeping occurs in scenarios whereby a node may go to sleep when a neighbour still has packets designated for it. An example of this phenomenon is when a node has no packet to send. The obvious action a node can then take, from an energy conservation perspective, is to put the radio into sleep mode for the duration of an entire frame. As a result, the node will miss all packets destined for it during that frame. To solve the early sleeping problem and also to let the RL agent adapt to incoming traffic conditions, we added a four-bit field in the data packet header called *FAIL_ATTEMPT*. The latter reflects on the delay experienced by the message received, due to the receiver's early sleeping. In other words, this field provides information to the receiving node regarding the number of failed transmission attempts made by the transmitter, before to the data is correctly received.

We embed information regarding failed transmission attempts by constructing a negative reward signal pertaining to previous actions. Accordingly, at the subsequent frame $(k+1)$, assuming the received packet number is $n_{\mathrm{r}}^{'}$ and packets 0 to $(n_{\mathrm{r}}^{'}-1)$ were received correctly, the delayed reward of state-action (s_k, a_k) is given by

$$r_{t+1}^k(n_{\mathrm{b}}, t_{\mathrm{r}}) = -\sum_{i=0}^{n_{\mathrm{r}}^{'}-1} f_i \tag{11}$$

whereby f_i denotes the number of failed transmissions corresponding to packet i.

From (10) and (11) it can be observed that the access control algorithm aims to both maximise the energy efficiency (as reflected by the instant reward $r_t^k(n_{\mathrm{b}}, t_{\mathrm{r}})$), as well as to minimise the number of missed packets due to early sleeping (expressed by the delayed reward $r_{t+1}^k(n_{\mathrm{b}}, t_{\mathrm{r}})$). The reward for the state-action pair (s_k, a_k) during frame k, r^k, is thus the combined rewards received at r_t^k and r_{t+1}^k.

4.4 The Q-learning algorithm

In our learning process, at the end of each frame, the RL agent evaluates the temporal difference, updates the Q-value and selects the next action according to the ϵ-greedy method (Sutton and Barto, 1998). Using this approach, with probability $1-\epsilon$, the agent executes the action with the highest Q value and with probability ϵ the agent randomly chooses an alternative action. This is done to balance exploitation of presumed optimal state-action pairs and exploration of novel policy modifications.

Intuitively, when there are more packets queued for transmission, one would expect the reserved active time to be longer. Therefore, we can specify the action space $A(s)$ for a given state s to be a subset of A. Also, since the traffic load and the networking condition vary in our case, we adopt a constant learning rate $\alpha = 0.1$ in order to adapt to the non-stationary environment. We further note that if the traffic load is constant over a relatively long period of time, the queued packet length (i.e. the state) will concentrate in a certain range, which greatly accelerates the learning process. The learning algorithm is given in Table 1.

Table 1 Algorithm proposed for the RL agent governing the RL-MAC protocol

Initialise $A(s)$ for all $s \in S$;Initialise α, γ, ϵ;
Set $Q(s,a) = 0\ \forall s \in S, \forall a \in A(s)$
Loop for $k = 1, 2...$
  Choose $a_k = t_r$ in $s_k = n_b$ using policy derived from Q ($\epsilon-$greedy)
  Take action $a_k = t_r$, observe $r_t^k(n_b, t_r)$ and $r_{t+1}^k(n_b, t_r)$, $s_{k+1} = n_b^{'}$
  $r^k(n_b, t_r) = r_t^k(n_b, t_r) + r_{t+1}^k(n_b, t_r)$
  $Q_{t+1}(n_b, t_r) = Q_t(n_b, t_r) + \alpha[r^k(n_b, t_r) + \gamma \max_{t_r^{'}} Q_t(n_b^{'}, t_r^{'}) - Q_t(n_b, t_r)]$
End Loop

4.5 Overhearing and collision avoidance

A similar overhearing avoidance mechanism to that used by S-MAC is utilised in RL-MAC. The difference is that in S-MAC, when a node hears RTS or CTS messages indicating that a packet is destined for one of its neighbours, it transitions into sleep mode until the beginning of the next frame, while in our protocol, the node only sleeps for the time specified in the CTS packet and then wakes up for potential transmitting or receiving if the reserved active time does has not expired.

When a node sends an RTS, but does not receive a CTS in return, it may be caused by collision and/or the receiver being in sleep mode. At the beginning of the frame, the probability of collision is high due to node synchronisation, in particular when the traffic load is heavy. However, the chance of collision is slimmer towards the middle of the frame. For this reason, when no CTS is received, a node backs off randomly for a duration of time that exponentially increases. Moreover, the node turns off its radio during back off to preserve energy. The node would wake up if and only if the reserved active time has not expired. It is worth mentioning that since our protocol utilises larger frames than T-MAC, the collision due to synchronisation is significantly reduced.

5 Simulation results

In order to evaluate the performance of the proposed protocol, several simulations were carried out in comparison to the S-MAC protocol using the latest version of the ns-2 simulation platform. Communication patterns in WSN can be broken down into single-hop communications, which are present in the star topology and multi-hop communications, which exist in linear topology. To that end, both the star and linear network topologies were considered. In both scenarios, we have used constant bit rate and Poisson traffic sources with different time intervals. UDP was employed as the transport layer protocol. We have varied the packet inter-arrival time in each batch of the simulations. The main parameters used by the simulations are given in Table 2.

Table 2 Parameters used by the RL-MAC simulations

Parameter	*Value*
Frame length	3.576 sec
Slot time	$T_s = 123$ ms
Packet transmission time	$T_p = 114$ ms
Queued packet length	[0, 1, 2,...16]
Reserved active time	[0, 0.123, 0.246,...3.567] sec
Weight coefficient	$\eta = 1.5$
Buffer size	$B = 16$
Discount factor	$\gamma = 0.5$
Learning rate	$\alpha = 0.1$
Transmission power	0.5 W
Receiving power	0.3 W
Idle power	0.05 W
Radio transmission rate	20 kbps

5.1 Star topology

The first set of simulations pertained to the star network topology depicted in Figure 2. In this scenario above, nodes 1 – 4 attempt to send 50-byte packets. The packet inter-arrival time varies from 10 to 1 sec, thereby reflecting on different traffic loads. As a result, the generating throughput varies between 20 and 200 byte/sec. We measure the average active time per frame (in percentage) at the receiving node (node 0). The receiver's active time is adaptively determined by the incoming traffic load, as illustrated in Figure 3. It is clearly noted that the higher the traffic load the higher percentage of time the receiver stays active, as would be expected.

Figure 2 Star sensor network topology

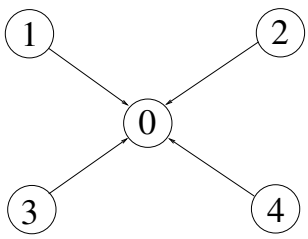
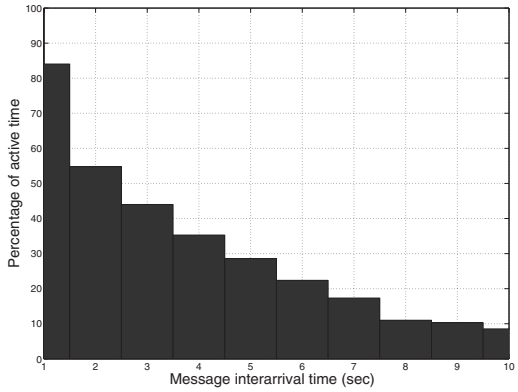


Figure 3 Percentage of receiver's active time in star topology



A comparison is carried out between S-MAC and RL-MAC considering the primary performance metrics in the context of MAC protocol for WSN: energy efficiency, data throughput and latency. For energy efficiency, we calculate the energy cost per-byte of the goodput (data throughput excluding any overhead). Figures 4 and 5 clearly demonstrate that RL-MAC can achieve a much higher throughput when traffic load is heavy. This is due to the fact that RL-MAC adaptively applies a higher duty cycle, by means of increasing the reserved active time, in response to increased traffic load, as depicted in Figure 3.

Figure 4 Data throughput versus message inter-arrival time for CBR and poisson traffic in star topology

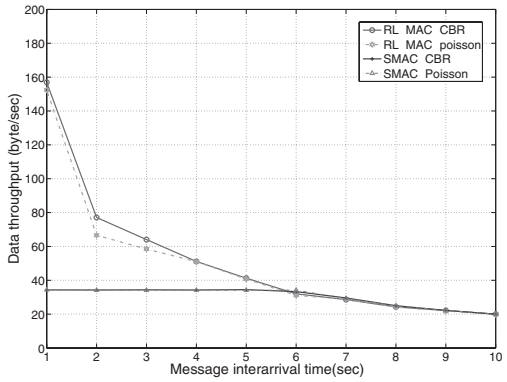
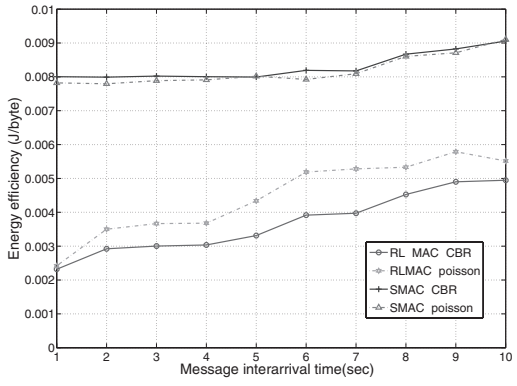
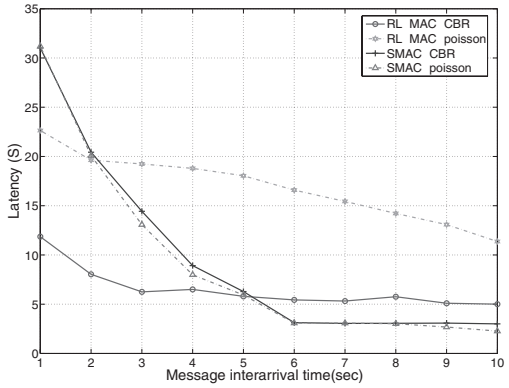


Figure 5 Energy efficiency versus message inter-arrival time for CBR and Poisson traffic in star topology



In term of power efficiency, we can see from Figure 5 that RL-MAC offers, on average, over 50% energy savings when compared to S-MAC. Moreover, in both algorithms, power efficiency increases (or, alternatively, the per-byte cost of energy decreases) as the traffic load increases, which is reasonable given that more energy is used in transmission and reception rather than idle listening, when traffic load is heavier. This effect is more obvious in RL-MAC. As illustrated in Figure 6, RL-MAC efficiently trades off latency for energy efficiency when traffic load is light – a highly acceptable attribute in a MAC protocol for WSN. The reason for this efficient trade off is that RL-MAC has a larger frame length than that used by S-MAC. However, the latency of RL-MAC is even lower when traffic load is heavy, because in RL-MAC, the packets will not backlog due to inadequate duty cycle as they do in S-MAC.

Figure 6 Latency versus message inter-arrival time for CBR and Poisson traffic in star topology

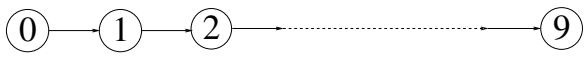


We then apply Poisson traffic patterns to test the performance of S-MAC and RL-MAC under non-uniformly distributed message lengths for the same topology. In this scenario, each node generates Poisson traffic based on statistically distributed inter-arrival times. The average inter-arrival time varies from 1 to 10 sec. Figures 4 and 5 illustrate how RL-MAC outperforms S-MAC in both the throughput and energy efficiency metrics, however the gap between S-MAC and RL-MAC is somewhat compromised simply because the irregularity in traffic load interrupts and decelerates the learning process.

5.2 Linear topology

We next examine a linear network topology, as shown in Figure 7. The network consists of 10 nodes, whereby each (with the exception of the edge nodes) can communicate with its two immediate neighbours. Traffic originates at node 0 and is destined to node 9, and visa versa. The sender generates a 200-byte long packet at a rate that ranges from 0.1 to 1 packet/sec, such that the generating throughput is between 20 and 200 byte/sec.

Figure 7 A 10-node linear network topology



Figures 8 and 9 clearly demonstrate that RL-MAC achieves higher throughput and higher energy efficiency than that attributed to S-MAC, for the linear topology as well. However, the effectiveness of RL-MAC is compromised since the traffic fluctuates substantially at the intermediate hops, and so the RL agent is distracted from the learning process. When traffic load is very high (200 byte/sec), the throughput can only achieve 35% (as compared to 80% in the star topology) of the generated traffic. However, in practical scenarios, as in clustered WSN, the traffic within a cluster (raw data exchanged between neighbours for in-cluster processing) is much higher than the traffic destined for more distant nodes (processed and often compressed data directed to a sink node). Thus, in the context of the data throughput metric, the performance of RL-MAC is good in both star and linear topologies. Moreover, RL-MAC saves 30.93% energy compared to S-MAC in linear topology while in star topology the corresponding number is 55.57%. A clear advantage in the mean latency is observed as well.

The performance comparison of RL-MAC and S-MAC is almost identical except that the learning speed is somewhat lower, due to the irregularity introduced by the Poisson traffic (Figures 8–10).

Figure 8 Data throughput versus message inter-arrival time for CBR and Poisson traffic in linear topology

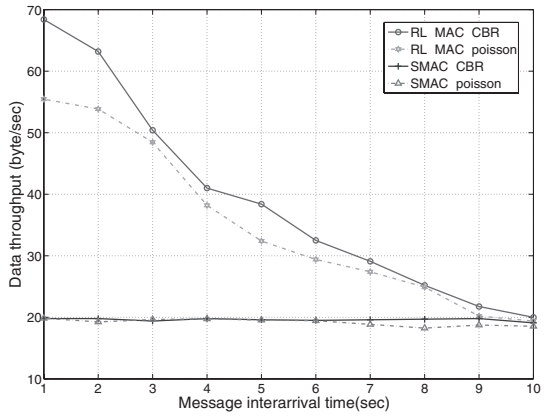


Figure 9 Energy efficiency versus message inter-arrival time for CBR and Poisson traffic in linear topology

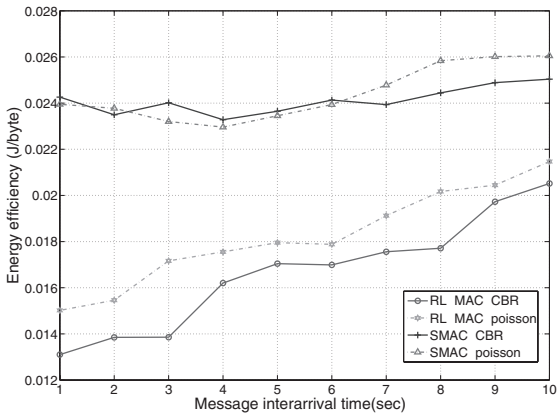
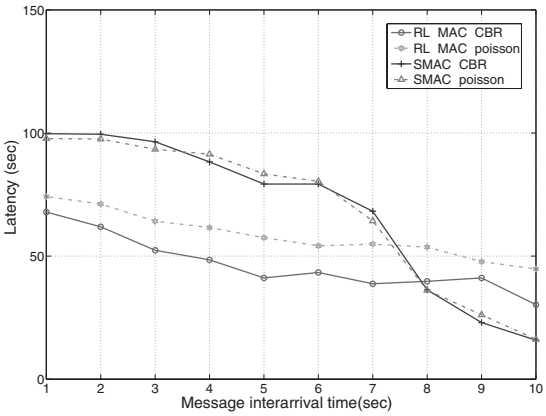


Figure 10 Latency versus message inter-arrival time for CBR and poisson traffic in linear topology



5.3 Mesh topology

Next, we use a dense sensor network with 100 nodes, where nodes are uniformly distributed over a 100 × 100 grid region. The sink is placed in the leftmost-bottom of the space, as we randomly activated a subset of the sensor nodes to generate traffic. The latter is statistically generated such that inter-arrival times are exponentially distributed with the expected value ranging from 10 to 1 sec. The transmission range for each sensor node is 20 units, and the communication pattern is mixed, involving both single- and multi-hop communications. We compare the overall data throughput and energy efficiency for S-MAC, T-MAC and RL-MAC in this scenario.

Figures 11 and 12 show that RL-MAC achieves much higher throughput when the traffic load is heavy. When comparing the energy efficiency results for RL-MAC with T-MAC it is observed that the former conserves much more energy. This is due to the fact that T-MAC wastes significant energy on synchronisation and unexpected collisions.

Figure 11 Data throughput versus message inter-arrival time in mesh topology

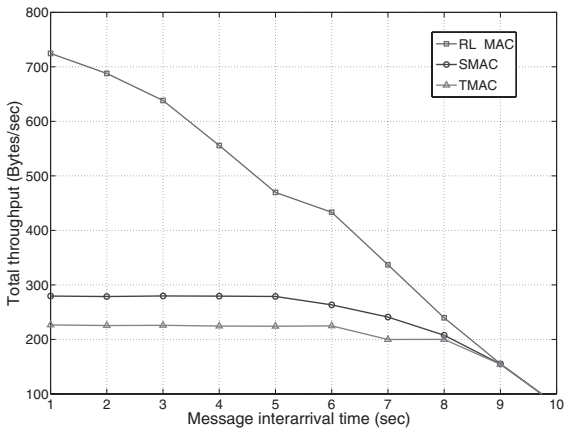
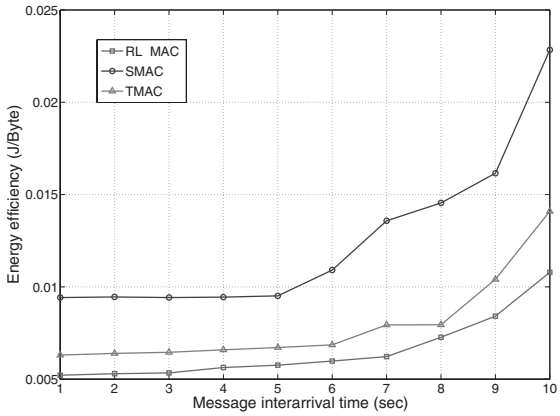


Figure 12 Energy efficiency versus message inter-arrival time in mesh topology



6 Conclusions

This paper formulates the MAC problem, in the context of optimising throughput and minimising energy utilisation, in packetised wireless sensor communications. We utilise an RL algorithm to solve the posed optimisation problem. The RL framework assumes an underlaying MDP, which allows nodes to infer the state of other nodes in the network in order to dynamically optimise their MAC policy. Compared to S-MAC and other schemes, our protocol offers very high throughput and high energy efficiency characteristics, even in the presence of high traffic load.

We note that the RL algorithm described does not require knowledge of preemptive state transition probabilities and uses only the feedback information provided by the protocol to issue its decisions. Simulation results in our study also indicate that the proposed scheme provides a simple, systematic, self-organised and distributed algorithm to achieve highly effective channel resource management in WSN. Furthermore, the proposed framework can be served as basis for cross-layer optimisation and the study of collaborative information processing in ad hoc clusters of sensor nodes.

Acknowledgements

This work has been partially supported by the Department of Energy (DOE) under research grant DE-FG02-04ER25607, and by the Woodrow W. Everett, Jr. SCEEE Development Fund in cooperation with the Southeastern Association of Electrical Engineering Department Heads.

References

Bertsekas, D.P. and Tsitsiklis, J.N. (1996) *Neuro-Dynamic Programming*, Athena Scientific.

Dam, T.V. and Langendoen, K. (2003) 'An adaptive energy-efficient mac protocol for wireless sensor networks', *SenSys '03: Proceedings of the first International Conference on Embedded Networked Sensor Systems*, New York, NY: ACM Press, pp.171–180.

Havinga, P.J.M. and Smit, G.J.M. (2000) 'Energy-efficient tdma medium access control protocol scheduling', *Proceedings of the Asian International Mobile Computing Conference (AMOC 2000)*, November.

Pandana, C. and Liu, K.J.R. (2005) 'Near-optimal reinforcement learning framework for energy-aware sensor communications', *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 4, pp.788–797.

Sutton, R.S. and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*, Cambridge MA: MIT Press.

Ye, W., Heidemann, J. and Estrin, D. (2004) 'Medium access control with coordinated adaptive sleeping for wireless sensor networks', *IEEE/ACM Transactions on Networks*, Vol. 12, No. 3, pp.493–506.

Zheng, T., Radhakrishnan, S. and Sarangan, V. (2005) 'Pmac: an adaptive energy-efficient mac protocol for wireless sensor networks', *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, pp.65–72.