Practical Implementation and Stability Analysis of ALOHA-Q for Wireless Sensor Networks

Selahattin Kosunalp, Paul Daniel Mitchell, David Grace, and Tim Clarke

This paper presents the description, practical implementation, and stability analysis of a recently proposed, energy-efficient, medium access control protocol for wireless sensor networks, ALOHA-*Q*, which employs a reinforcement-learning framework as an intelligent transmission strategy. The channel performance is evaluated through a simulation and experiments conducted using a real-world test-bed. The stability of the system against possible changes in the environment and changing channel conditions is studied with a discussion on the resilience level of the system. A Markov model is derived to represent the system behavior and estimate the time in which the system loses its operation. A novel scheme is also proposed to protect the lifetime of the system when the environment and channel conditions do not sufficiently maintain the system operation.

Keywords: *Q*-learning, medium access control, resilience level, wireless sensor networks.

I. Introduction

Wireless sensor networks (WSNs) have appeared as a rapidly growing research topic owing to their potential application areas, ranging from environmental monitoring to industry, military, and health applications [1]. A typical WSN is expected to consist of a potentially large number of inexpensive sensor nodes with the capabilities of sensing, computation, and communication, each of which is likely to be battery-powered, small in size, and able to communicate over short distances. In many cases, a distinctive feature of a WSN is that the sensor nodes are randomly deployed in remote areas, which often makes recharging or replacing the batteries difficult. A typical WSN needs to be able to self-organize and be robust to environmental changes such as node failures.

Sensor nodes in a WSN share the same communication medium, which may result in a packet transmission failure through multiple concurrent accesses. Medium access control (MAC) protocols have the responsibility of controlling and regulating users in accessing a shared transmission medium. A huge number of MAC protocols have been proposed to bring to light significant improvements in energy efficiency, channel throughput, delay performance, and fairness [2], [3]. Although most proposed protocols have provided significant performance improvements, the design of the protocols has resulted in considerable complexity and overhead. Taking real sensor platform architectures into consideration, that is, simple devices with limited power and memory, the practicality of MAC protocols must be considered based on hardware limitations and constraints. Many of the proposed schemes have only been evaluated through simulations, which may not reflect the actual performance of the protocols owing to unrealistic assumptions. Therefore, it is important to develop

Manuscript received Dec. 1, 2015; revised June 20, 2016; accepted July 4, 2016.

Selahattin Kosunalp (corresponding author, skosunalp@bayburt.edu.tr) is with the Department of Electrical and Energy, Bayburt University, Turkey.

Paul Daniel Mitchell (paul.mitchell@york.ac.uk), David Grace (david.grace@york.ac.uk), and Tim Clarke (tim.clarke@york.ac.uk) are with the Department of Electronics, University of York, United Kingdom.

http://dx.doi.org/10.4218/etrij.16.0115.1030

simpler protocols to provide flexibility for practical implementation.

ALOHA-based schemes have a key benefit of simplicity but suffer from a blind transmission strategy as the nodes are allowed to transmit packets as soon as they become ready for transmission, requiring no pre-coordination with other users accessing the transmission medium. This drawback limits the achievable maximum channel throughput because it may result in collisions. In the case of slotted-ALOHA, time is divided into discrete slots, and each user is required to transmit at the beginning of a slot. The use of an intelligent slot selection technique will potentially improve the channel performance. Reinforcement learning (RL) has been applied to slotted-ALOHA as an effective transmission strategy, enhancing the channel performance significantly in both single- and multi-hop communication scenarios [4], [5]. *Q*-Learning is an RL algorithm used for an intelligent slot selection strategy [6]. RL provides a means of learning the behavior of a system by interacting with a dynamic environment through trial-and-error. It allows the determination of an optimum transmission strategy from the consequences of a device's action on its environment. The advantage of learning for slotted-ALOHA is that users are able to find unique transmission slots in a fully distributed manner, resulting in a scheduled outcome.

This is the first time that ALOHA-*Q* has been practically evaluated to achieve perfect scheduling, thereby improving the channel performance significantly. The performance of ALOHA-*Q* is compared with a well-known MAC protocol, Z-MAC [6]. We investigated its resilience to a loss of convergence in order to consider the weakness of the scheme associated with packet losses during a steady state. A steady state occurs when all nodes have found a unique slot. The level of resilience to a loss of convergence is presented according to various packet loss probabilities. A Markov model is derived to estimate the time to loss of convergence for a single user. A novel technique is then proposed to protect the convergence lifetime in the presence of packet loss.

The rest of this paper is organized as follows. Section II presents related studies highlighting the main features of existing MAC protocols for WSNs. Section III introduces a brief description and practical performance of ALOHA-*Q* as well as the experimental setup used for this study. A stability analysis of ALOHA-*Q* with the derived Markov model is provided in Section IV. The proposed scheme is described in Section V. Finally, Section VI provides some concluding remarks regarding this research.

II. Related Works

The overwhelming majority of MAC protocols proposed for WSNs are contention-based and inherently distributed, but suffer from overhearing, collisions, idle-listening, and re-transmissions. Although schedule-based protocols can alleviate these problems by dynamically assigning transmission schedules, these schemes introduce complexity and overhead, and time synchronization is their key requirement.

Sensor MAC (S-MAC) [7], perhaps the most studied MAC scheme, is a representative contention-based protocol, and many recent protocols center on the concept of S-MAC. S-MAC incorporates a tunable periodic listening and sleep schedule in which each node turns its radio off to preserve energy. Each node determines its own schedule based on virtual clusters that are formed by neighboring nodes. S-MAC follows the traditional four-way handshaking technique (RTS/CTS/DATA/ACK) for collision and overhearing avoidance. The schedule needs to be synchronized among neighboring nodes. To update the schedule, a small SYNC packet is exchanged with neighbors during the listening period. S-MAC adopts a message passing technique to reduce the contention latency. A long message is fragmented into many small fragments that are sent in bursts. The duty-cycle period in S-MAC is of a fixed duration, and energy can therefore be unnecessarily wasted at low traffic load levels.

Timeout MAC (T-MAC) [8] extends S-MAC by introducing an adaptive duty-cycle to shorten energy consumption while maintaining reasonable throughput. The active period is dynamically ended if nothing is heard after a timeout period. As in S-MAC, the nodes wake up at the beginning of each active period, listen to the medium to sense any activity, and return to sleep mode if no activation event occurs. T-MAC allows the nodes to remain awake after completion of a packet transmission or reception to observe potential incoming traffic. T-MAC introduces a future request-to-send (FRTS) to solve the early sleeping problem, which means that if a node loses contention, the destination will switch to sleep mode. Using FRTS packets, the intended destination is informed of the future packet reception time so the destination will wake up at the appropriate time.

RL-MAC [9] is a reinforcement learning-based protocol that adaptively adjusts the sleeping schedule based on local and neighboring observations. For a local observation, each node records the number of successfully transmitted and received packets to be a part of the determination of the duty cycle. As for neighboring observations, the number of failed attempts is added to the header to inform the receiver, which saves energy by minimizing the number of missed packets (early sleeping). The key property of this scheme is that the nodes can infer the state of other nodes using a Markov decision process.

Low-energy adaptive clustering hierarchy (LEACH) [10] is a self-organizing, adaptive clustering-based MAC and routing protocol. The concept of LEACH is to divide nodes into local clusters in which one node is nominated as the cluster-head in each cluster. The cluster-head is responsible for coordinating the cluster and forwarding the data from nodes in its cluster to the sink. To balance the energy dissipation among the nodes, the role of the cluster-head node is randomly rotated among the nodes within a cluster based on the amount of energy left in the current cluster-head. LEACH assumes that each node in a cluster can directly communicate with the cluster-head. The nodes in a cluster transmit their data using a TDMA schedule created by the cluster-head. Hence, the nodes can switch the radio off when they are not scheduled to transmit or receive, thus saving energy consumption in these nodes. In each cluster, a different CDMA code is used for transmission to avoid interference with a nearby cluster (inter-cluster).

The traffic-adaptive medium access protocol (TRAMA) [11], a TDMA-based algorithm, is a good example of a schedule-based protocol whereby the nodes arrange common schedules by exchanging their neighborhood information with their neighbors. The time is divided into single slots for both data and signaling transmissions. Each node selects a slot based on a distributed election algorithm according to their current traffic information. TRAMA results in significant processing burden and high overhead associated with scheduling.

Zebra MAC (Z-MAC) has been chosen for a performance comparison because it is an effective MAC scheme that provides high channel utilization and energy-efficiency. It is therefore worth describing the underlying basics of Z-MAC. Z-MAC [12] is a hybrid protocol that combines the advantages of TDMA and CSMA. It uses a CSMA scheme at low traffic loads and a TDMA scheme at higher traffic loads. It has a preliminary set-up phase if there is a neighbor discovery. A neighbor discovery is conducted by sending ping packets to one-hop neighbors. Each node generates a list of two-hop neighbors. Using the two-hop neighborhood, Z-MAC applies a distributed slot assignment algorithm to make sure that any two nodes in the two-hop neighborhood are not given the same slot, thereby reducing the potential for collisions between two-hop neighbors. In Z-MAC, each user has its own slot, but if the user does not have any data to transmit, other users may borrow the slot.

Unfortunately, most of the protocols introduced in this section have only been evaluated using simulation tools. Z-MAC has been extensively implemented practically in both single- and multi-hop topologies. Its effectiveness in terms of channel utilization has been demonstrated. S-MAC has also been implemented in practice, the aim of which was to measure the energy consumption. None of the other protocols have been practically evaluated. Computationally complex and overwhelming algorithms may render the MAC protocols infeasible. We conclude that low complexity and overhead are important for many practical deployments owing to the limitations and constraints of low-cost, simple sensor devices. ALOHA-Q therefore represents a good example of simplicity while providing perfect scheduling in an intelligent way with minimal additional overhead. It only has an initial poorer performance phase based on a Q learning algorithm in which each node learns to explore a unique transmission slot that will be dedicated to the node at the end of the phase. Eventually, if there are a sufficient number of slots, all nodes will find a unique transmission slot and keep transmitting there. Many contention-based schemes rely on CSMA features to perform channel sensing and hand-shake procedures, which introduce additional overhead. The only overhead of the ALOHA-Q scheme is an acknowledgement (ACK) packet, which is commonly implemented in MAC protocols to confirm the successful reception of a packet. Compared with schedule-based schemes, the RL process provides a similar collision-free channel access with no pre-coordination with other users accessing the channel requiring no scheduling information exchange. It only requires a small amount of time to converge to an optimal steady state of one slot assigned per node.

III. Q-Learning Based ALOHA: ALOHA-Q

In this section, the ALOHA-Q protocol is briefly described along with its fundamental design properties and underlying features. The throughput performance of ALOHA-Q is then presented practically to validate its performance with respect to simulations under ideal conditions (that is, no packet loss). The performance evaluations have been carried out under two main topologies, a single-hop and a linear-chain, which are described in detail later in this section.

1. Protocol Description

A repeating frame structure is introduced within slotted-ALOHA. Each frame consists of a number of slots, N, which should be appropriately set in order to allow each node to have a unique slot. In a single-hop scenario, N is optimally set to the number of nodes in the system. However, in a multi-hop scenario, N is determined by a local transmission and interference range of the nodes, network topology, the density of the nodes, and number of source nodes along the route. A node is allowed to transmit at most one packet in a frame. The generated packets are queued first-in-first-out with the packet at the head being transmitted. For each node, every slot in the repeating frame is given a Q value, which is initialized to zero upon startup. The Q values are subsequently updated according to (1).

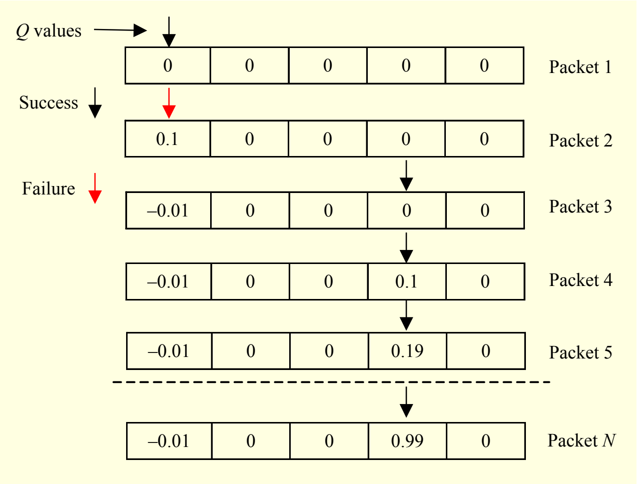

Fig. 1. Example of the slot selection technique, $\alpha = 0.1$.

$$Q_{t+1}(i,s) = Q_t(i,s) + \alpha(R - Q_t(i,s)), \qquad (1)$$

where i indicates the present node, s is the preferred slot, R is the current reward, and α is the learning rate. A transmitter will always choose the slot within a frame with the highest Q-value. If more than one slot has the same Q value, the transmitter randomly chooses one of them. If the packet transmission is successful, R takes a value of $R_\text{r} = +1$, which constitutes a reward. If the packet transmission fails, then R takes a punishment value of $R_\text{p} = –1$. Consequently, a sequence of successful transmissions using the same slot will cause the associated Q-value to increase, finally converging on a value very close to +1. There is no consensus on the choice of values for R_r and R_p, but it has been shown [4], [5] that +1 and –1 respectively produce a convergence to +1 for a successful slot choice, and zero for all other non-chosen slots in that frame. We define this condition as being a steady state slot because, for a particular slot; it will always choose a high Q slot. The learning rate, α, is an important parameter that controls the speed of convergence. It determines to what extent the recently acquired information will be considered.

We now present an illustrative example of the Q-learning algorithm for five slots per frame. In Fig. 1, the first packet is transmitted on slot 1 of the frame. This is randomly chosen because all Q values are equal to zero. Because that packet was successfully transmitted, the Q value for slot 1 is incremented according to (1). However, in this case, the next packet transmission in slot 1 fails. The Q value falls immediately to –0.01 and another slot is selected randomly. In this scenario, slot 4 continues to be successful for the next N packets. We see the Q value approaching a value of +1. We will later show that it takes many more successful transmissions for the Q value to approach +1 than it takes to reduce back toward zero owing to successive packet transmission failures.

Previous studies have shown that ALOHA-Q reaches a steady-state operation following a quick period of convergence during a simulation, where a packet loss is due solely to collisions. The behavior of the algorithm after convergence in a practical environment characterized by a more significant packet loss is unclear, and is therefore considered in Section IV. This leads to the observation that the scheme has low resilience against packet loss and that convergence is quickly lost. This motivated the development of a modified punishment strategy, which is introduced and evaluated in Section V.

2. Experimental Setup

We use MicaZ nodes [13], which are IEEE 802.15.4-compliant devices featuring an ATmega128L low-power microcontroller and a CC2420 [14] radio transceiver operating at 2.4 GHz. They have 4 Kbytes of data memory and 128 kbytes of programmable flash, and provide a data rate of 250 kbits/s. Our nodes run on TinyOS [15], which is an efficient component-based and event-driven operating system, and provides software support for the application design requirements of MicaZ nodes.

TinyOS provides only one packet format, which consists of a fixed-sized header, the payload, and a cyclic redundancy check (CRC). The length of the header depends on the specific radio platforms. The CC2420 header is 11 bytes long. A 2-byte CRC follows the last field in the packet format, which is automatically generated by the hardware. The length of the payload can be varied up to 114 bytes because the maximum complete packet length provided by IEEE 802.15.4 is 127 bytes. These three fields form a MAC protocol data unit (MPDU). The MPDU is automatically prefixed with a preamble and start of frame delimiter based on the radio and frame length by the microcontroller when transmitting a packet. Figure 2 shows a complete packet structure that complies with IEEE 802.15.4.

In our implementation, the SHR, PHR, MHR, PAYLOAD, and CRC fields comprise 5 bytes, 1 byte, 11 bytes, 114 bytes, and 2 bytes, respectively. The control packets are normally expected to be very small without a PAYLOAD. We therefore created another packet type, the acknowledgement packet, which has a 5-byte SHR, a 1-byte PHR, an 11-byte MHR, and

Bytes:	4	1	1	11	*n*	2
	Preamble	Start of frame delimiter	Frame length	Header	Payload	CRC
	Synchronisation header (SHR)		PHY header (PHR)	MAC header (MHR)	MAC payload (PAYLOAD)	MAC footer (MFR)

Fig. 2. IEEE 802.15.4-compliant packet format.

http://dx.doi.org/10.4218/etrij.16.0115.1030

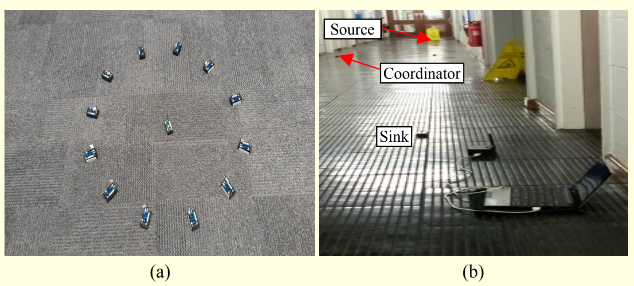

Fig. 3. Application environments: (a) single- and (b) multi-hop scenarios.

a 2-byte CRC.

For the single-hop scenario, the performance is evaluated for an indoor topology, as depicted in Fig. 3(a), in an unobstructed area with line-of-sight communication, which comprises 12 users. All nodes are in the range of each other, are equidistant from the receiver, and transmit at the same power level. Each node generates packets and sends them directly to the receiver. All nodes have the same mean packet inter-arrival time, which is exponentially distributed and synchronized by the receiver. After the deployment of the nodes, the transmitters wait in the receive mode for a specific packet called a *Hello* packet to be sent from the receiver. Once the receiver is powered up, and after a certain time, it transmits a *Hello* packet to all nodes. This synchronizes the transmitters to enable their packet generation process to run concurrently.

For a multi-hop scenario, the performance is evaluated under a linear network topology comprising five nodes, as presented in Fig. 3(b). Here, the packets are generated by node 1 (source) to be transferred through the line hop-by-hop to node 5 (sink), meaning that each packet travels through nodes 2, 3, and 4, and arrives at node 5. Each node transmits at the minimum transmission power level that allows them to receive the packets from only one hop neighbors. The interference range is also one-hop. To synchronize the nodes at the onset of the repeating frames simultaneously, the *Hello* packet strategy described above is used through a *network coordinator* broadcasting at the maximum transmission power level, which therefore covers all nodes. To avoid a short-term clock drift, a modest guard band is included in the slot timings (as indicated from the parameters listed in Table 1). This eliminates any potential packet losses that may occur owing to this potential problem. For a longer operation, the central receiver or network coordinator can be set to periodically transmit *Hello* packets for the purpose of resynchronization. Alternatively, some form of established distributed synchronization algorithm can be implemented.

Table 1. Experiment parameters.

Parameters	Values
Channel bit rate	250 kbits/s
Data packet length (ALOHA-*Q*)	1,064 bits
Data packet length (Z-MAC)	840 bits
ACK packet length (simulation)	20 bits
ACK packet length (experiment)	152 bits
Slot length (simulation)	1,100 bits
Slot length (experiment)	1,250 bits
Experiment period	100,000 slots
Learning rate (α)	0.1

Energy efficiency is achieved by only waking up the nodes in their dedicated slot, thereby removing the cost of idle listening. In a single-hop scenario, it is straightforward to only wake up the nodes in specific slots. However, the nodes have to be awake for reception and transmission in a multi-hop scenario. To achieve this, the transmitter informs the corresponding receiver regarding its future transmission pattern, particularly the number of future packets to be transmitted in the same current slot. This is called informed receiving (IR), the details of which can be found in our previous paper [4].

Channel throughput is the percentage of channel capacity used. One Erlang represents the continuous use of a channel. The theoretical maximum throughput of a single-hop scenario for the experiment parameters given in Table 1 is close to 0.85 Erlangs (1,064/1,250 bits). In a five-hop scenario, with the transmission and interference ranges of a single hop, two adjacent nodes in each direction along the chain have to select different transmission slots to avoid collisions, and thus one in every three nodes can utilize the same transmission slot, and the optimum frame size becomes three slots per frame. Therefore, the theoretical maximum throughput at the sink is 0.33 Erlangs. A small guard band is left between the slots in order to mitigate the propagation delays and any timing offsets. During all simulations, the default values of Z-MAC (eight contention slots for slot owners, and an extra 32 contention slots for non-owners) are used.

3. Steady-State Results

To evaluate the performance of ALOHA-*Q*, we implemented it in both OPNET and MicaZ/TinyOS using the parameters given above. In addition, ideal performances of ALOHA-*Q* [4], [5] and Z-MAC were simulated based on a very small acknowledgement packet length. We also implemented a conventional slotted-ALOHA on the testbed. Figure 4 presents the results of all scenarios for varying traffic levels.

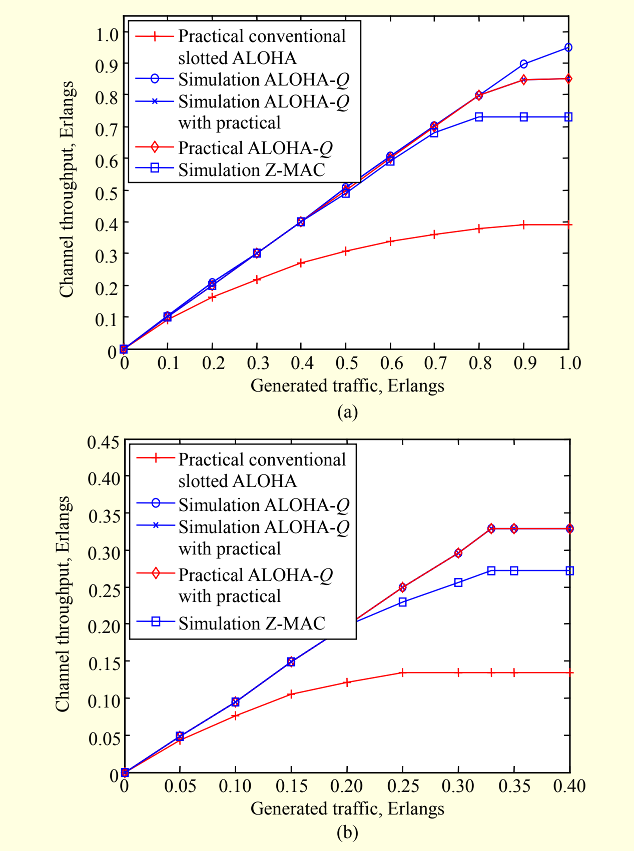

Fig. 4. Channel throughput: (a) single- and (b) multi-hop scenarios.

The generated traffic is defined in Erlang units, which represent the proportion of the channel occupied by all users. This is used to calculate the average packet inter-arrival time equally for each user, as obtained below.

$$I = \frac{L.N}{G.D}. \tag{2}$$

Here, I denotes the average packet inter-arrival time, L is the packet length in bits, N is the number of nodes in the network, G is the desired generated traffic load in Erlangs, and D is the data rate of the channel in bits/s.

In the single-hop scenario, the simulation results show that the ideal throughput of ALOHA-*Q* increases linearly and reaches a maximum of 0.95 Erlangs, which corresponds to every node finding a unique slot. The practical and simulation results of the throughput, using the same parameters as the practical system, exhibit a similar increasing trend but to a lower maximum throughput of approximately 0.85 Erlangs because there is an ACK packet overhead of 0.15 Erlangs. Z-MAC achieves a lower maximum throughput than ALOHA-*Q* owing to a greater overhead and potential for contention (nodes can potentially contend for their non-owned slots). On the other hand, the practical maximum throughput of the conventional slotted-ALOHA (nearly 0.39 Erlangs) with 12 users is slightly higher than its theoretical achievable throughput (0.368 Erlangs based on the assumption of an infinite number of nodes) [16].

In the multi-hop scenario, the throughput using ALOHA-*Q* grows linearly and reaches its maximum limit. All transmitted packets are transferred to the sink node under steady-state conditions. Slotted-ALOHA can only provide a throughput of 0.13 Erlangs owing to its inefficient transmission strategy. The throughput of ALOHA-*Q* stabilizes at 0.33 Erlangs with increasing traffic levels because it depends on the frame size. Figure 4 shows that ALOHA-*Q* achieves a much higher throughput when the traffic load is heavy because the large contention windows used for channel sensing limit the performance of Z-MAC.

IV. Stability Properties of ALOHA-*Q*

This section studies the stability of the ALOHA-*Q* protocol against possible changes in the environment and changing channel conditions, particularly packet losses during a steady state. We first begin with the main reasons for a packet loss that can occur at any time in a wireless network. The *Q*-value of a slot represents the efficiency of knowledge obtained on the slot and the willingness of this slot to be chosen. Therefore, the increments and decrements of the *Q*-value based on the outputs of the transmissions are very important to deeply understand the behavior of the network. The decline/accrual of the *Q*-value is presented to demonstrate how the *Q*-value of a slot converges to a value close to +1, and how quickly it reduces to lose its convergence. To estimate the convergence loss time in the presence of a packet loss after convergence, a Markov model is derived where each state holds a *Q*-value based on successful/failure transmissions. The convergence loss time of a single slot is then presented. To understand the behavior of the whole network against packet loss, the network performance is also observed with respect to various packet failure ratios.

1. Issue of Packet Loss

Wireless sensor networks can have a reputation for an unpredictable quality of wireless communication because they are fairly densely deployed in harsh, inaccessible environments. A number of factors govern the performance of wireless communication. These focus around the environment, the network topology, and the devices. We note that three important reasons for packet loss are multi-path interference, hardware architecture, and scalability of the network size.

Depending on the environmental characteristics, multi-path interference can occur, which results in duplicate packets being

http://dx.doi.org/10.4218/etrij.16.0115.1030

received over small time differences that may result in their destruction. Sensor devices constrained in their bandwidth and energy cannot tolerate multi-path effects having insufficient frequency diversity [17].

Owing to the very modest hardware architecture of sensor nodes, a loss of packets will occur in practice. Our previous study [18] demonstrated that a typical popular sensor node, the IRIS node, cannot operate effectively under high traffic loads because it is unable to switch quickly from reception to transmission mode to send back acknowledgement packets. Consequently, depending on the traffic load level, a certain proportion of the acknowledgement packets may not be sent. Hence, even though a packet is received successfully, the transmitting node will assume it to be lost as no acknowledgement packet is received. To overcome this problem, we proposed employing a guard band between the transmission and reception modes, although this wastes channel resources. We therefore conclude that there might be a possibility of losing some packets in WSNs because of the sensor hardware, and that such loss may not be predictable.

An important and desirable attribute of MAC protocols is scalability with the network size, and some new nodes may need to be deployed later. A good MAC scheme must comfortably meet such a change. However, during the addition of new nodes to the network, some packets might be lost owing to the arrangement of new transmission schedules. Depending on the application, this process may need to protect the current schedules of existing users.

2. Level of Resilience to Loss of Convergence

Although ALOHA-*Q* provides perfect scheduling, allowing no packet loss from collisions after convergence, as validated through simulations and in a real-world test-bed, a packet loss can still occur in practice for the reasons described above. We now systematically analyze the level of resilience to a loss of convergence in the presence of packet loss. The learning rate (α) is an important parameter because it has a significant effect on the *Q*-value updates. Therefore, various learning rates are simulated to demonstrate the behavior of the *Q*-value of a single-node unique slot, as shown in Fig 5. Establishing the best case in which all the packets are successful in a particular slot from the initialization of the system is very important for a deeper understanding of the behavior of the *Q*-value updates. Figure 5(a) presents the *Q*-value of a slot with consecutive successful transmissions.

We can see that the learning rate determines, as expected according to (1), the accrual of the *Q*-value. Smaller values result in a longer time to converge to a *Q*-value of 1. The numbers of consecutive successful transmissions required to achieve convergence for a single user with respect to the learning rates of 0.1, 0.05, 0.03, and 0.01 are 50, 100, 150, and 400, respectively. However, as a negative reward has more impact on the *Q*-value when the *Q*-value is positive, the number of successive failures required to result in a *Q*-value reduction to zero is therefore significantly fewer. We assume that the rest of the *Q*-values are set to zero after convergence, and thus once the *Q*-value of a unique slot falls back to zero, it will lead the associated user to seek to find a new slot. We see from Fig. 5(b) that only seven consecutive failures cause the *Q* value to return to zero (loss of convergence) at a learning rate of 0.1. Considering that a packet loss will occur in the real world, the risk of this rapid decline in *Q* value is significant, leading to a loss of convergence and subsequent quality of service. The system will not have a good level of robustness and will not be protected from infrequent collisions or small changes in the environment and channel conditions.

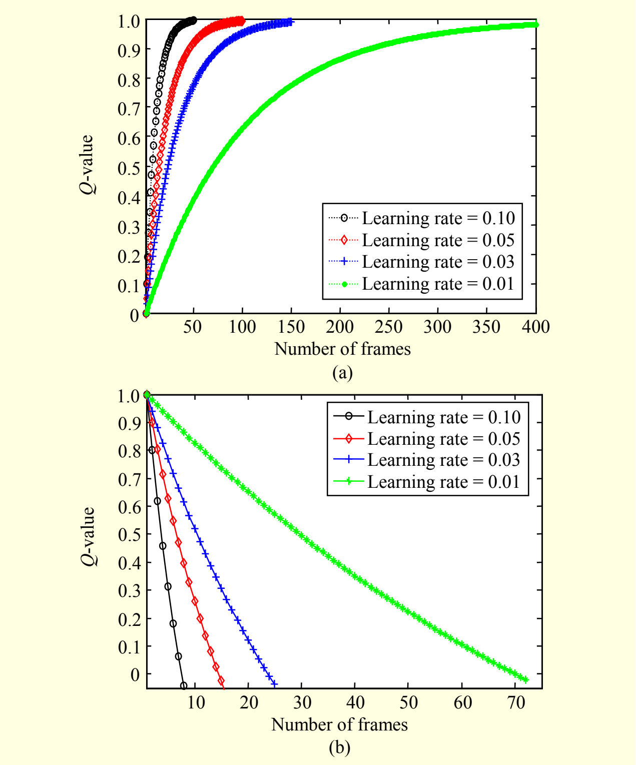

Fig. 5. Behaviour of the *Q*-value update: (a) best case where all packets are successfully transmitted and (b) worst case in which all packets are lost.

3. Markov Model

We now derive a Markov model to represent the behavior of the system after convergence. Each node has a unique slot, the *Q*-value of which is very close to 1, where the rest of the slots have *Q*-values of zero. Each state represents a particular *Q*-

value. The *Q*-value increments based on successful transmissions are much smaller than *Q*-value decrements based on failed transmissions. Upward transitions between neighboring states correspond to a single success. A single failure results in a downward transition across multiple states. The total number of states therefore depends only on the learning rate. After a transmission, the *Q*-value is updated and a state transition occurs. If a transmission succeeds, the process moves forward. If not, the process moves backward and chooses the state that has the closest *Q*-value. An example of the Markov model is shown in Fig. 6 for a learning rate of 0.1. There are 50 states required for convergence, as previously noted, for a learning rate of 0.1.

Let p denote the probability of a successful packet transmission based upon the factors previously outlined. This will be the probability of moving forward, $p_{k,k+1}$, where $k = 0, 1, 2, \ldots ,49$. It will also be the probability of staying in the last state, $p_{50,50}$. The probability of moving backwards, $p_{k,l}$, will be $(1 - p)$, where l is the corresponding state after an unsuccessful transmission (for example, $k = 15$, $l = 9$). These state transition probabilities can be formulated as follows:

$$P_{k,k+1} = p, \tag{3}$$

$$P_{k,l} = 1 - p. \tag{4}$$

4. Loss of Convergence Time Estimation

In a practical deployment, a packet loss can occur at different rates. In our previous study [18], around half of the acknowledgement packets were not sent from the receiver because of the hardware issues at a channel load of 1 Erlang. Under a real situation, the ratio of packet loss may vary. We will not necessarily observe a sustained sequence of consecutive failures. Therefore, the relationship between packet loss ratio and the time to lose convergence is important to establish.

The approach presented in [19] for ALOHA-*Q* provides the convergence time of a whole network through an analytical model. In this model, a state transition probability matrix, $\boldsymbol{P}$, which is a sparse matrix, is considered. Using the notation $\boldsymbol{P}^2$ to denote the multiplication of $\boldsymbol{P}$ by itself, the elements of $\boldsymbol{P}^2$ are

$$p_{i,j}^{(2)} = \sum\nolimits_{m=0}^{N} p_{i,m}, p_{m,j}. \tag{5}$$

Here, $p_{i,j}{}^{(2)}$ represents the transition probability from state i to state j through one transition state (two transitions). Similarly, the elements in $\boldsymbol{P}^3$ are

$$p_{i,j}^{(3)} = \sum\nolimits_{m=0}^{N} p_{i,m}^{(3)} p_{m,j}, \tag{6}$$

which is the probability of moving from state i to state j through all possible states after two transition states. Here, $\boldsymbol{P}^n$ is referred to as the matrix of state transition probabilities after $(n - 1)$ transition states, and thus $p_{i,j}{}^{(n)}$ is the probability of moving from state i to state j after n transitions. To calculate the time of the convergence loss, we need the expected number of transitions (slots), from the last state to be achieved to all states except state zero, which is obtained as follows.

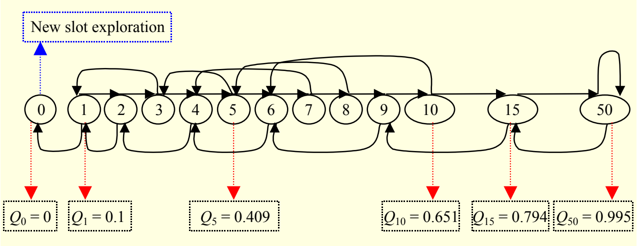

Fig. 6. Markov model with a learning rate of 0.1.

$$\mathrm{E}\{\text{convergence loss time}\} = \sum_{n=1}^{\infty} \sum_{j=N}^{1} p_{N,j}. \tag{7}$$

This is the expected convergence loss time starting with state N. A detailed derivation and proof of the model can be found in [19].

Using our Markov-model based simulation, for a given learning rate, the total number of states required for convergence is initially calculated. The *Q*-value of each state is then calculated, and the state transitions, up or down, are determined. Using a uniformly distributed random number generation, different packet loss ratios are artificially created. Here, a number is randomly generated within the range of 1 to 100. This is then compared with a predefined threshold determined to create a particular packet loss rate. If the number is greater than the threshold, the process moves forward; otherwise, it moves backward. To create a 40% loss rate, for instance, the threshold is set to 40. The simulation is initialized during the final state as the system is assumed to have converged, and the following state transitions are undertaken. The process is stopped when the process has reached a state of zero. The required number of iterations, which is equivalent to the number of frames, is then recorded. The simulation is run 100 times and the average value is taken.

Using the test-bed, we observed the time (number of frames) to lose the convergence for a particular node for a given packet loss rate. The receiver sends a certain amount of acknowledgement packets using the random number generation strategy described above. In this case, some of the packet receptions will not be acknowledged, despite these packets being successfully received. At the start of the trial, each node learns a unique slot. When a node then tries to change this slot, it sends a message containing the number of frames taken from the beginning to a base station, which is connected to a computer to monitor the data packets. We run the implementation 100 more times and again take the average. Figure 7(a) presents the time (number of frames) before the convergence loss with a different probability of failure. The running throughput at the receiver is calculated after every 10 frames at three different packet loss rates, and is presented in Figs. 7(b) and 7(c) for the two scenarios.

We can see that the practical results of the convergence loss time match the results of the Markov model and the simulations. The convergence can be lost within 100 frames to below a probability failure level of 0.3, whereas within 600 frames at a probability of failure of 0.2, which is below a level of 0.1, the convergence is never lost. Therefore, to provide an efficient operation of the protocol, the probability of failure must be less than 0.1, which is referred to as the convergence loss point (CLP) in the rest of this paper. The running throughput is obtained from initialization through each time step (10 frames for a single hop and 100 frames for multiple hops), where each curve represents an average of 100 runs. The real-time running throughput decreases faster with a reduction in the packet success rate.

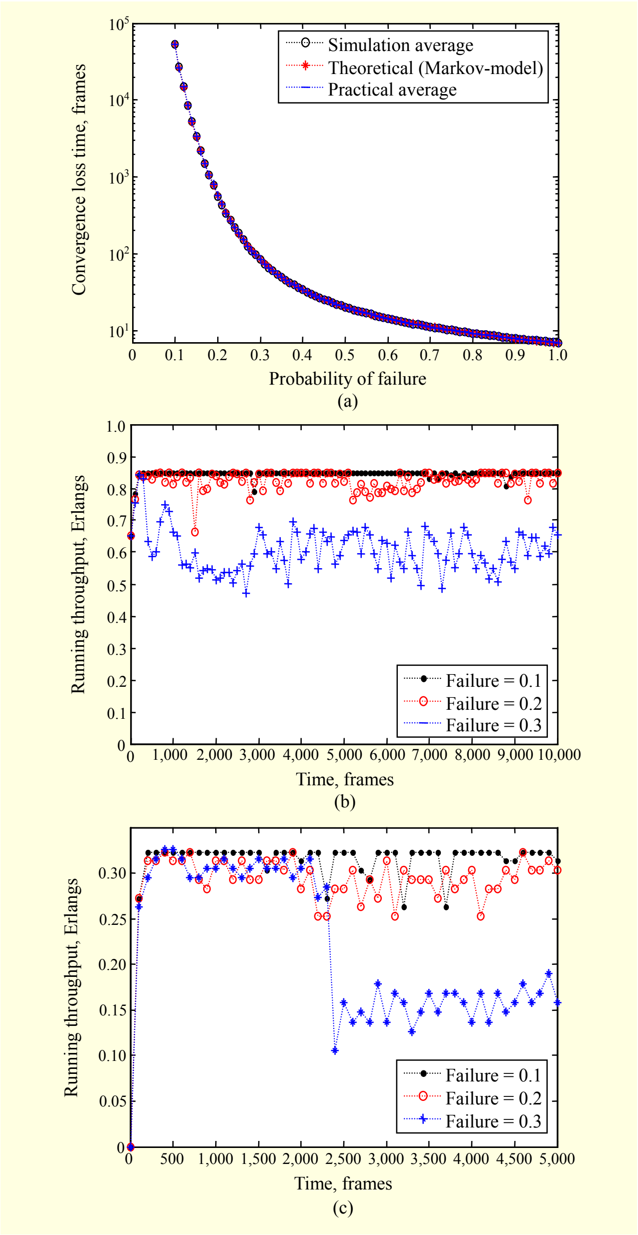

Fig. 7. Convergence loss time and overall system behaviour against packet loss: (a) average time of convergent loss, (b) running throughput for a single hop, and (c) running throughput for multiple hops.

V. Proposed Scheme: Punishment Modification Strategy

In this section, a new punishment scheme is proposed to deal with the issue of packet loss that may occur in less ideal environments. The performance enhancements through the proposed punishment modification are presented. Our scheme is shown to ensure a good level of protection against packet loss.

It was found that the convergence will not be lost if the packet loss rate does not exceed 10%. The main objective is to maximize the CLP to protect the convergence from an unknown instant or long-term change. According to (1), the punishment value, assuming a fixed learning rate, plays an important role in updating the Q-value. In particular, the use of a fixed punishment value (–1) reduces the Q-value more quickly when the Q-value is positive. It is therefore clear that the use of a reformulated numerical value of the punishment can serve to protect the convergence loss. We intend to dynamically change the magnitude of the punishment when a packet loss occurs after convergence., which will result in the Q-value reducing more slowly.

As we pointed out previously, the number of consecutive failures required to lose the convergence is 7, whereas the number of consecutive successes required to achieve to the convergence is 50. We propose equating this imbalance by updating the punishment value when a packet transmission fails. In this case, 50 consecutive failures will cause a loss of convergence. After a packet failure, the punishment value is re-calculated to update the Q-value, and thus the process will take the previous state in the Markov model. Let us consider the two neighboring states to demonstrate the modification of the punishment value, as depicted in Fig. 8.

Here, Q_{N-1} represents the Q-value of state $N-1$, and the Q_N is the Q-value of state N. If the packet transmission fails when the process is in state N, the new Q-value will be Q_{N-1}. If the packet transmission succeeds when the process is in state $N-1$, the new Q-value will be Q_N.

$$Q_{N-1} = Q_N + \alpha(R_p - Q_N), \tag{8}$$

$$Q_N = Q_{N-1} + \alpha(R_r - Q_{N-1}). \tag{9}$$

We then substitute (8) with (9) to obtain the new punishment value, which will take the process to the previous state:

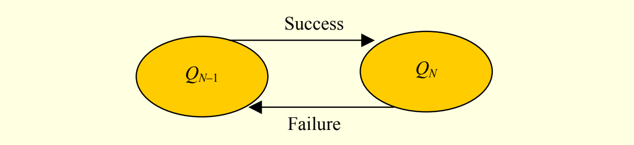

Fig. 8. Two neighbouring states in the Markov model.

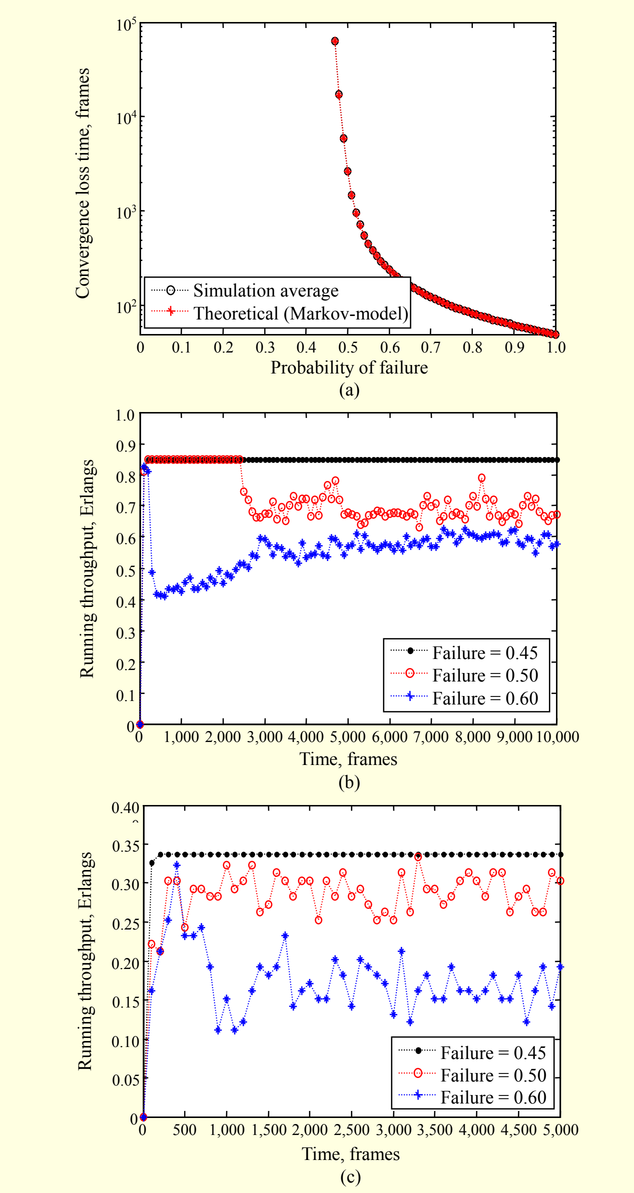

Fig. 9. Convergence loss time and overall system behaviour against packet loss: (a) average time of convergence loss, (b) running throughput for a single hop, and (c) running throughput for multiple hops.

$$R_{\mathrm{p}} = \frac{QN(2-\alpha)-r}{1-\alpha}, \tag{10}$$

which is the new punishment equation based on the current Q-value. Therefore, after an unsuccessful transmission, the punishment value is calculated and the *Q*-value is updated.

Similar to the results shown in Fig. 7, we present the results of the new punishment scheme obtained from the Markov model and simulations. The Markov model results match the average results of the simulations. It can be clearly seen that our scheme achieves better results, improving the time of convergence loss. The CLP is now 0.47, which indicates that the network will operate adequately as long as around half of the packets are successfully received. Here, we experimentally validate the analytical results, implementing the proposed scheme in an indoor test-bed. Again, the running throughput is evaluated for three loss rates with the convergence already having been achieved prior to the start of the test. The practical results prove that the system does not lose convergence beyond a loss rate of 0.47. However, as the loss rate increases, the system loses convergence quickly. All results are presented in Fig. 9.

VI. Conclusion

This paper thoroughly analyzed the stability properties of a recently proposed, energy-efficient MAC protocol for single- and multi-hop communications, called ALOHA-*Q*, which combines a slotted-ALOHA, with its benefits of simplicity and low computation, and *Q*-Learning, thereby providing an intelligent slot selection strategy. We began with the practical implementation issues of ALOHA-*Q*, which provides perfect scheduling under a rapidly achieved steady state. We then showed that ALOHA-*Q* is prone to a loss of convergence in the presence of packet losses that are due to changes in the environment and the radio conditions. A Markov model representing the behavior of a user has been provided and used to estimate the time taken to lose the convergence. It was shown through the Markov model and a test-bed that the convergence can be quickly lost because of a high punishment level. A novel punishment technique has been proposed to deal with a low packet failure in order to protect the operation of the network. The proposed scheme serves to protect the lifetime of the convergence by dynamically adjusting the punishment level.

References

[1] I.F. Akyildiz et al., “A Survey on Sensor Networks,” *Commun. Mag.*, vol. 40, no. 8, 2002, pp. 102–114.

[2] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC Protocols for Wireless Sensor Networks: A Survey,” *Commun. Mag.*, vol. 44, no. 4, 2006, pp. 115–121.

[3] M.A. Yigitel, O.D. Incel, and C. Ersoy, “QoS-Aware Mac Protocols for Wireless Sensor Networks: A Survey,” *Comput. Netw.*, vol. 55, no. 8, 2011, pp. 1982–2004.

[4] Y. Chu, P.D. Mitchell, and D. Grace, “ALOHA and Q-Learning Based Medium Access Control for Wireless Sensor Networks,”

Int. Symp. Wireless Commun. Syst., Paris, France, Aug. 28–31, 2012, pp. 511–515.

[5] Y. Yan et al., "Distributed Frame Size Selection for Q Learning Based Slotted ALOHA Protocol," *Proc. Int. Symp. Wireless Commun. Syst.*, Ilmenau, Germany, Aug. 2013, pp. 733–737.

[6] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA, USA: MIT Press, 1998.

[7] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Ann. Joint Conf. IEEE Comput. Commun. Soc.*, New York, USA, 2002, pp. 1567–1576.

[8] T.V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. Int. Conf. Embedded Netw. Sensor Syst.*, Los Angeles, CA, USA, Nov. 5–7, 2003, pp. 171–180.

[9] Z. Liu and I. Elhanany, "RL-MAC: A QoS-Aware Reinforcement Learning Based MAC Protocol for Wireless Sensor Networks," *Proc. IEEE Int. Conf. Netw. Sens. Contr.*, Lauderdale, FL, USA, Apr. 23–25, 2006, pp. 768–773.

[10] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. Ann. Hawaii Int. Conf. Syst. Sci.*, Hawaii, HI, USA, Jan. 7, 2000, pp. 1–10.

[11] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Wireless Netw.*, vol. 12, no. 1, Feb. 2006, pp. 63–78.

[12] I. Rhee et al., "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, June 2008, pp. 511–524.

[13] *Datasheet for MicaZ wireless measurement system*, Accessed Nov. 2015. http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[14] *Datasheet for CC2420 IEEE 802.15.4-compliant RF Transceiver*, Accessed Nov. 2015. http://www.ti.com/lit/ds/symlink/cc2420.pdf

[15] P. Levis et al., "TinyOS: An Operating System for Wireless Sensor Networks," in *Ambient Intelligence*, Berlin, Germany: Springer, 2005, pp. 115–148.

[16] N. Abramson, "The Throughput of Packet Broadcasting Channels," *IEEE Trans. Commun.*, vol. 25, no. 1, Jan. 1977, pp. 117–128.

[17] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," *Proc. Int. Conf. Embedded Netw. Sensor Syst.*, Los Angeles, CA, USA, Nov. 5–7, 2003, pp. 1–13.

[18] S. Kosunalp et al., "Practical Implementation Issues of Reinforcement Learning Based ALOHA for Wireless Sensor Networks," *Proc. Int. Symp. Wireless Commun. Syst.*, Ilmenau, Germany, Aug. 27–30, 2013, pp. 360–364.

[19] Y. Chu et al., "Application of Reinforcement Learning to Medium Access Control for Wireless Sensor Networks," *Eng. Appl. Artif. Intell.*, vol. 46, Nov. 2015, pp. 23–32.

Selahattin Kosunalp received his BS degree in electronics and telecommunications engineering from Kocaeli University, Kocaeli, Turkey in 2009, and his MS degree in communications engineering and his PhD in electronics engineering from the University of York, York, U.K. in 2011 and 2015, respectively. He is now with the Department of Electricity and Energy, Bayburt University, Bayburt, Turkey. He is the author of several refereed journal and conference papers and has experience asa reviewer for a number of conferences and journals. His research interests lie in wireless sensor networks, medium access control (MAC) protocol designs, energy harvesting technologies, and real-time embedded systems.

Paul Mitchell received his MS degree and his PhD from the University of York in 1999 and 2003, respectively. He has been a member of the Department of Electronics at York since 2002, and is currently a senior lecturer. His research interests include medium access control and routing, wireless sensor networks, underwater communications, cognitive radio, traffic modeling, queuing theory, and satellite and mobile communication systems. Dr. Mitchell is an author of over 90 refereed journal and conference papers and has served on numerous international conference program committees.

David Grace received his PhD from University of York in 1999. Since 1994 He has been in the Department of Electronics at York, where He is now Professor (Research) and Head of Communications and Signal Processing Group. Current research interests include aerial platform based communications, cognitive dynamic spectrum access and interference management. He is currently a NonExecutive Director of a technology start-up company, and a former chair of IEEE Technical Committee on Cognitive Networks.

Tim Clarke received his BA degree in biology from the University of York in 1975. He joined the Royal Air Force as an Air Traffic Control Officer before becoming an education officer. He underwent advanced training at the Royal Military College of Science, Shrivenham, U.K., where he received his MS degree in guided weapons systems engineering. He is senior lecturer in the Control Engineering Department and is head of the Control Systems Laboratory, Intelligent Systems Group, Department of Electronics at the University of York. His research interests are in the areas of biologically inspired engineering and control systems.