
Article:
Kosunalp, Selahattin, Chu, Yi, Mitchell, Paul Daniel orcid.org/0000-0003-0714-2581 et al. (2 more authors) (2016) Use of Q-Learning Approaches for Practical Medium Access Control in Wireless Sensor Networks. Engineering Applications of Artificial Intelligence. pp. 146-154. ISSN 0952-1976

https://doi.org/10.1016/j.engappai.2016.06.012

Use of Q-Learning Approaches for Practical Medium Access Control in Wireless Sensor Networks

Selahattin Kosunalp[1, 2], Yi Chu[2], Paul D. Mitchell[2], David Grace[2] and Tim Clarke[2]

[1]*Department of Electricity and Energy, University of Bayburt, Bayburt, TURKEY*

Email: skosunalp@bayburt.edu.tr

[2]*Communications and Signal Processing Research Group, Department of Electronics, University of York, Heslington, York YO10 5DD, UK*

Email: {yi.chu, paul.mitchell, david.grace, tim.clarke}@york.ac.uk

Abstract— This paper studies the potential of a novel approach to ensure more efficient and intelligent assignment of capacity through medium access control (MAC) in practical wireless sensor networks. Q-Learning is employed as an intelligent transmission strategy. We review the existing MAC protocols in the context of Q-learning. A recently-proposed, ALOHA and Q-Learning based MAC scheme, ALOHA-Q, is considered which improves the channel performance significantly with a key benefit of simplicity. Practical implementation issues of ALOHA-Q are studied. We demonstrate the performance of the ALOHA-Q through extensive simulations and evaluations in various testbeds. A new *exploration/exploitation* method is proposed to strengthen the merits of the ALOHA-Q against dynamic the channel and environment conditions.

Index Terms—Q-Learning, ALOHA, Medium Access Control, Wireless Sensor Networks.

1. Introduction

Wireless sensor networks (WSNs) have gained ever increasing interest given their significant potential in various applications, ranging from environmental monitoring to industry, military and health applications [1]. Such a network is normally expected to include a large number of small, inexpensive and battery-operated sensor nodes which are typically deployed in an ad-hoc fashion to perform a common task. A common and unique characteristic of a WSN after deployment is the inaccessibility of sensor nodes which makes recharging or replacing exhausted batteries challenging or impractical. This inherent feature brings about the necessity to design robust operation of the networks without external intervention and access.

A typical WSN employs a medium access control (MAC) protocol which regulates user access to the shared radio communications medium, affording significant impact on the overall network performance. Designing power efficient MAC protocols is therefore of paramount importance.

In the last two decades, researchers have focused on developing a wide range of MAC protocols for WSNs in an energy-efficient manner. Current MAC protocols can be broadly divided into *contention-based* and *schedule-based*. The majority of the proposed schemes are contention-based and inherently distributed, but they introduce energy waste through *overhearing, collisions, idle-listening and re-transmissions* [2]. Schedule-based protocols can alleviate these sources of energy waste by dynamically assigning transmission schedules, but these benefits are incurred at the expense of higher complexity and overheads. On the other hand, there are a number of MAC schemes which combine the features of both contention-based and schedule-based approaches, called *hybrid* protocols. The main drawback of the hybrid schemes is their complexity which may make them suitable for only a limited number of applications.

Reinforcement learning (RL) has been recently applied to design new MAC protocols in WSNs. Many developed RL-based schemes aim to adaptively adjust the duty cycle of the nodes which is best illustrated by S-MAC [7]. These protocols provide the nodes with an intelligent way of predicting other nodes' wake up times based upon the transmission history of the network. RL-based protocols significantly reduce the energy consumption due to both idle listening and overhearing in the context of duty cycling. ALOHA and Q-Learning have been integrated to establish a new MAC protocol, namely ALOHA-Q [3]. ALOHA-based techniques are important for certain categories of Wireless Personal Networks (WPNs) and WSNs such as those based on Radio Frequency Identification (RFID) systems which have limited memory and power capabilities. The ALOHA-Q scheme inherits the merits of contention-based and schedule-based approaches while offsetting their drawbacks. ALOHA-Q uses slotted-ALOHA as the baseline protocol with a key benefit of simplicity. It allows users to find unique transmission slots in a fully distributed manner, resulting in a scheduled outcome. ALOHA-Q aims to reach an optimal steady state where each user within an interference range has unique transmission slots. Therefore, ALOHA-Q works like a contention-based scheme gradually transforming into a schedule-based scheme, which can be fully achieved in the steady-state without the need for centralised control and scheduling information exchange in steady state conditions. In order to demonstrate the effectiveness of

this approach, performance evaluations in both single-hop and multi-hop communication scenarios have been carried out [3-5]. These studies show that ALOHA-Q can achieve a high channel utilisation while minimising the energy consumption. However, these evaluations are restricted to simulation-based evaluation, where the practical implementation issues can be avoided.

This paper extends the implementation of ALOHA-Q in both simulation and realistic testbeds, to linear-chain, grid and random topologies. Practical implementation issues of ALOHA-Q are studied based upon hardware limitations and constraints. In order to strengthen the merits of the ALOHA-Q against dynamic the channel and environment conditions, the epsilon-greedy strategy is integrated. We compare the performance of the ALOHA-Q with a well-known MAC protocol Z-MAC [6] and a self-learning protocol SSA [13]. Section II presents the related work highlighting the main features of existing RL-based MAC protocols for WSNs. Section III introduces a brief description of ALOHA-Q and *exploration/exploitation* methods proposed. The performance of ALOHA-Q in simulation in comparison to ZMAC and SSA is validated with practical results in Section IV. We evaluate experimentally the performance of ALOHA-Q in two real-world events: (1) packet losses in steady state and (2) participation of new nodes to the network. ALOHA-Q with *exploration/exploitation* is implemented to provide better performance in these two events. Finally, Section V concludes the paper.

2. Related Work

A wide range of MAC protocols for WSNs have been proposed with the primary objectives of improving energy-efficiency and channel throughput. The majority of the existing schemes are contention-based and employ CSMA as the baseline approach. In this section, our focus is on RL-based MAC schemes proposed in the literature.

S-MAC is a well-known RTS-CTS based MAC protocol that introduced the concept of a duty- cycle in which the nodes in the network periodically sleep, wake up, listen for potential transmissions, and return to sleep [7]. A large number of existing protocols have used the theme of S-MAC in order to further schedule the nodes' sleep and active periods as the duty-cycle period in S-MAC is of a fixed duration. The nodes form virtual clusters to determine a common schedule among the neighbouring nodes. A small SYNC packet is exchanged among neighbours at the beginning of each active period to assure that they wake up

concurrently to reduce the control overheads. After the SYNC exchange, the data packets can be transmitted using the RTS-CTS mechanism until the end of the active period. S-MAC adopts a *message passing* technique to reduce contention latency. A long message is fragmented into many small fragments which are sent in bursts. The main drawback of S-MAC is the predefined constant listen-sleep periods which can cause unnecessary energy consumption, particularly when some nodes those are located near the sink, may require a higher duty-cycle in order to serve the traffic passing through them.

RL-MAC is a RL-based protocol that adaptively adjusts the sleeping schedule based on local and neighbouring observations [8]. The key property of this scheme is that the nodes can infer the state of other nodes using a Markov Decision Process (MDP). For local observation, each node records the number of successfully transmitted and received packets during the active period to be a part of the determination of a duty cycle. As for neighbouring 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).

A decentralised RL approach is proposed to schedule the duty-cycle of sensor nodes based only on local observations [9]. The main point of this study is to shift active periods in time based on transmission paths and ranges along a route. Similar to that of S-MAC, the wake-up periods of the nodes which need to communicate with each other are synchronised, whereas the schedules of nodes on neighbouring branches are desynchronised to avoid interference and packet losses. The active periods are further divided into time slots and the nodes are allowed to wake for a predefined number of slots in each period. The slots where a node will wake up are decided by the learning process. Each slot within a frame is given a quality value which shows the efficiency of the slots. The quality values of slots are updated by rewarding successful transmissions and penalising the negative interactions. As a result of the learning process, the quality values of one group of slots become strictly higher than the rest.

Another similar approach combining the Slotted-ALOHA and Q-Learning algorithms which achieves high channel utilisation while mitigating energy consumption is ALOHA-Q [3]. Each node repeats a frame structure which includes a certain number of time slots. The packets are transmitted in these time slots. Each node stores a Q-value (equivalent to the quality value above) for each time slot within a frame. The Q-value of a slot is updated individually when a transmission happens and this is used to explore each slot more frequently. Successful transmissions are denoted by small acknowledgement packets which

are immediately sent upon the correct reception of the packets. The nodes generate a positive outcome (reward) when the acknowledgement packet is received, otherwise a punishment is applied to update the Q-value. Consequently, the slots with higher Q-values are preferred for data communication and this behaviour repeats the same actions. This process continually returns rewards which serve to decrease the probability of unsuccessful transmission. Eventually, the learning process leads the network to a steady state condition where unique slots are assigned to the nodes. Although this approach appears to be a schedule-based network, it does not require any scheduling information exchange. However, it is critical to set a sufficient number of slots within a frame. Redundant slots in a frame will result in achieving lower channel throughput. None of the work described above has addressed optimum frame size assignment problem. Instead, Reference [4] presents a distributed frame size selection algorithm for ALOHA-Q in a single-hop scenario. Furthermore, the selection of the frame size has been discussed for a multi-hop wireless sensor networking [5].

QL-MAC, Q-Learning-based MAC, is proposed to deal with the issue of finding an efficient way of scheduling the duty-cycle of sensor nodes [10]. Each node considers its own traffic load and the network load of its neighbours. The basic underlying concept of QL-MAC resembles that of a decentralised RL approach, whereby time is divided into time slots (*frame*s) which are further divided into smaller time units (slots). Every node using the Q-Learning algorithm individually limits the total number of slots in which the node wakes up. The frame length and the number of slots constituting the frame remain constant.

All of the protocols introduced in this section have only been evaluated through simulations. The feasibility of the schemes for practical implementation on real sensor hardware is an important issue, because the performance evaluation on real testbeds can engender numerous challenges such as unrealistic assumptions and resource limitations of the sensor hardware [11]. We therefore believe that practicality of the new protocols must be considered since they can have a significant impact on the performance [12].

Z-MAC has been chosen for performance comparison because it is an effective MAC scheme which provides high channel utilisation and energy-efficiency. It is therefore worth describing the underlying basics of Z-MAC. Zebra MAC (Z-MAC) [6] 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 neighbour discovery. Neighbour discovery is performed by sending ping packets to one-hop

neighbours. Each node generates a list of two-hop neighbours. Using the two-hop neighbourhood, Z-MAC applies a distributed slot assignment algorithm to make sure that any two nodes in the two-hop neighbourhood are not given the same slot, thereby reducing the potential for collision between two-hop neighbours. 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.

Self-learning scheduling approach (SSA) is another RL-based protocol which uses a similar Q-Learning algorithm as ALOHA-Q to control the nodes' sleeping schedule [13]. All nodes in the network share the same duty cycle and periodic wake up time to ensure the reception of control packets. Each node can be in active, sleep or idle state during each duty cycle, and the probability of entering each state is determined by each node's local learning algorithm. History of energy costs and packet queue length all contribute to the updates of the Q values, thereby searching for a long-term solution to achieve both energy efficiency and throughput performance after convergence of the learning algorithm.

3. Use of Q-Learning in Medium Access Control

Reinforcement learning (RL) is a method by which a learning agent, using reward and punishment, learns from the consequences of its actions by interacting with a dynamic environment through trial-and-error [14]. It allows determination of an optimum transmission strategy from the consequences of a device's action on its environment. Q-learning is one of the most popular and widely used RL algorithms. In particular, each node acts as an agent with a set of actions, and the agents choose actions according to reward signals received in the form of numerical values. The objective of an agent is to find the best actions that maximize its long-term benefits.

3.1 ALOHA and Q-Learning: ALOHA-Q

ALOHA-Q divides time into repeating frames where a certain number of slots are included in each frame for data transmission. Each slot is initiated with a Q-value to represent the willingness of this slot for reservation, which is initialised to 0 on start-up. Upon a transmission, the Q-value of corresponding slot is updated, using the Q-learning update rule given by Eq. (1):

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

Where *i* indicates the present node, *s* is the slot identifier, *R* is the current reward and α is the learning rate. Upon a successful transmission, *R* takes a value of r = +1 which constitutes a reward. Upon a failed transmission, *R* takes a punishment value of p = -1. In both cases, the learning rate is set to 0.1 as chosen in [3] which controls the speed of the Q-value. Consequently, nodes always select the slots with maximum Q-values. Nodes are restricted to access only one slot per frame for their generated packets and they can use multiple slots in a frame for relaying the received packets. In ALOHA-Q, the nodes broadcast small *ping* packets during their transmissions in order to inform the receiving nodes about the future transmission allocations. In this way, the receiving nodes will only listen to the current preferred slots of the transmitting nodes.

An illustrative example is now presented for updating the Q-values for 10 slots per frame and a node is allowed to send a maximum of 3 packets in each frame. In fig. 1, slot 1 and slot 5 of the frame 1 are randomly chosen for the first 2 packet transmission because all Q values are equal to 0. As these packets were successfully transmitted, the Q-values for slot 1 and slot 5 are incremented according to Eq. (1). In frame 2, there are now 3 packets to be transmitted; slot 1 and slot 5 are certainly selected as they have higher Q-values and slot 10 is randomly selected among others. However, the transmissions in slot 5 and slot 10 fail and the associated Q-values fall immediately. In the next round, slot 1 will be the first preferred and two slots among the unchosen slots are selected at random. This process will continue to explore 3 slots which have Q-values approaching +1 for the next N packets.

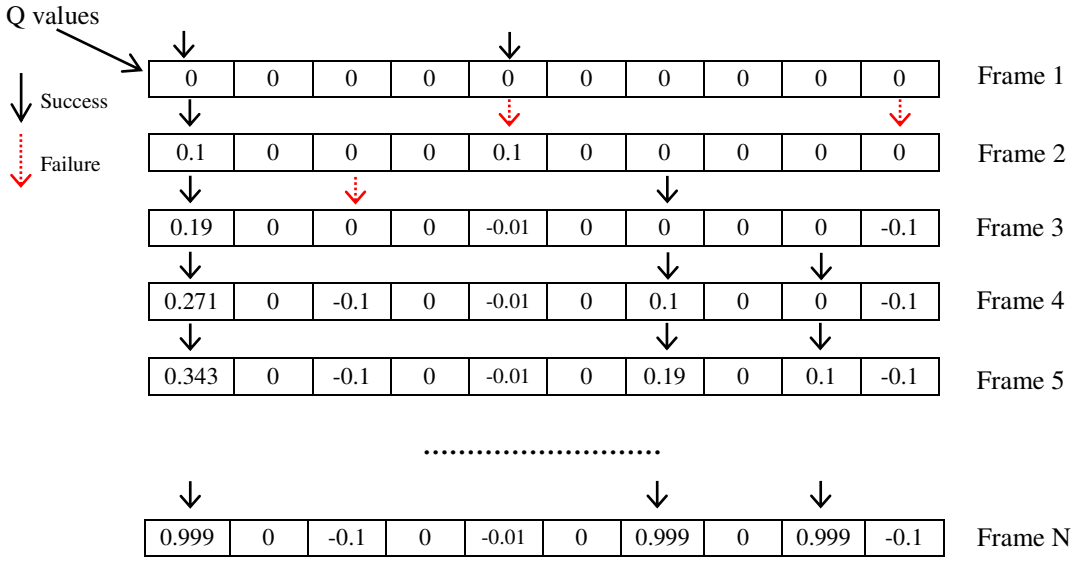

Fig. 1. Example of the slot selection technique, $\alpha = 0.1$.

3.2 Exploration and Exploitation

A reinforcement learning agent needs to explicitly explore its environment in order to find the best action. Exploitation is used to capitalise on the experience already available. The balance between the exploration and exploitation is a fundamental issue in reinforcement learning, and has long been an important challenge. Exploration is, of course, particularly crucial when the environment is non-stationary. Therefore, an agent must adapt to environmental changes.

Several methods have been proposed to ensure a good balance between the exploration and exploitation. This study will focus only on the methods used in the context of Ad-Hoc techniques. The first strategy is *greedy* selection. The agent always chooses the action with the highest action-value as in ALOHA-Q. This is, however, not an efficient exploration method because it does not control the exploration time. Hence, it might take too long to adapt to any change. A variation on the *greedy* approach is ϵ*-greedy* selection. In this method, at each time step the agent generates a random value between 0 and 1 and if this value is smaller than a fixed-value $0 < \epsilon < 1$, the agent explores, otherwise the agent exploits. Therefore, the duration of exploration is controlled by the ϵ value. In exploration, the agent randomly selects an action. A derivative of ϵ*-greedy* for mainly selecting the ϵ value is the *decreasing*-ϵ method. Basically, ϵ is set to a high value at the beginning to allow more exploration and it is reduced at each time step for more exploitation.

As stated above, ALOHA-Q naturally uses the *greedy* method in which the exploration and exploitation are always performed together, since the slot with the highest Q-value is preferred and the Q-value of this slot is immediately updated after transmission. It can be clearly seen from the Eq. (1) that the Q-value increments based on successful transmissions are extremely small after it converges close to 1 as the Q-value has a limit of 1. However, in this case, a single failure results in a significant decrement of the Q-value. More specifically, only 7 consecutive failures cause a Q-value to return to 0 at a learning rate of 0.1, as the punishment has more impact on the Q-value when the Q-value is positive. We see that a few failures would lead the associated user to seek to find a new slot despite previous thousands or even millions of successful transmissions. Therefore, the Q-learning algorithm actually considers the short-term channel history in order to obtain enough knowledge. This inherent tradeoff can have a significant influence on network performance in dynamic channel and environment conditions. In the previous studies of ALOHA-Q, this issue has not arisen due to

the simulations being carried out in a static environment and with time-invariant channel conditions.

3.3 ALOHA-Q with ϵ-greedy method: ALOHA-Q-EPS

Using the ϵ-*greedy* policy, nodes select a transmission slot with the highest Q-value with probability 1- ϵ and select a random slot with probability ϵ. The main drawback is that the selection of ϵ value is unclear. An epsilon value of 1.0 will result in exploring constantly, while a value of 0.0 will result in always exploiting. Therefore, the ϵ value is ideally set to a small value in order to allow more exploitation while providing sufficient exploration. In all experiments of this work, it is typically set to 0.1. Also, during exploration, an equal probability is given for the non-ideal slots to be selected which may have low Q-values. Due to the constant value of ϵ, the transmissions at randomly selected slots in steady-state can cause collisions which may reduce the maximum achievable throughput. ALOHA-Q protocol with ϵ-*greedy* is called the ALOHA-Q-EPS and its performance will be demonstrated in section 4.

3.4 ALOHA-Q with decreasing-ϵ greedy method: ALOHA-Q-DEPS

The efficiency of knowledge obtained is represented by Q-values which also represent the goodness of the exploration level. An intelligent control of the tradeoff between exploration and exploitation is investigated with respect to the behaviour of the Q-value. A *decreasing-*ϵ method is developed to allow nodes to explore more until they achieve a certain level of exploration. The duration of ϵ is controlled by Q-value as shown in Eq. (2). Basically after convergence, the Q-value is updated after transmission if in exploration, but it is not updated after exploitation. As the Q-value of a slot increases, the exploitation in this slot occurs more frequently and vice versa. In ALOHA-Q, the term *convergence* in a slot occurs when the Q-value of this slot approaches to 1. However, this would not allow exploration in this strategy after convergence. To solve this problem, we define $Q_{convergence}$ in which the slots will be accepted as converged when the Q-values of them exceed $Q_{convergence}$. This will decide the ratio of the exploration after convergence. The decision of $Q_{convergence}$ depends on the application scenario such as topology, density of the nodes and environment conditions.

$$\epsilon = \begin{cases} 1 - Q_{value} & \text{before convergence} \\ 1 - Q_{convergence} & \text{after convergence} \end{cases} \qquad (2)$$

We denote this modification with a new protocol name, ALOHA-Q-DEPS. The benefits of the ALOHA-Q-DEPS will be practically presented in the following section. The value of $Q_{convergence}$ is set to 0.9, so that the nodes will explore 10% of the time after convergence. In exploration, the slots are randomly chosen as in *Є-greedy.* However, this random selection in steady state is not efficient as the packets with random slots would potentially collide with others' unique slots. Therefore, the nodes reasonably select the slots with higher Q-values in exploration after convergence.

4. Performance Evaluation

In order to evaluate the performance of the ALOHA-Q in comparison to SSA and Z-MAC, several simulations have been carried out under three main topologies; *linear-chain, grid and random*. The simulation results are validated experimentally using MicaZ [15] and IRIS [16] nodes. These nodes run on TinyOS [17], an efficient component-based and event-driven operating system, providing software support for the application design requirements of MicaZ and IRIS nodes. All the practical experiments were conducted in an unobstructed area with line of sight. The nodes were manually placed outside in a flat area transmitting at the same power level. Significant interference from the environment (e.g. WiFi or Bluetooth) has not been experienced as ALOHA-Q achieved its theoretical maximum performance. To enable multi-hop networking, each node reduces the transmission power level to its minimum value. In all simulations, the default values of Z-MAC (8 contention slots for slot owners and an extra 32 contention slots for non-owners) are used. In the SSA simulations, all parameters for updating Q values are set to the same as in [13]. The action selection is assigned to greedy policy (exploit action with the largest Q value) for fair comparison because ALOHA-Q is a greedy algorithm. Table I. summarises the simulation and experimental parameters. It is worth noting that the length of ACK packets and the slots are larger in practice because of the preamble, synchronisation, header and CRC bits sent from the radio chip. ALOHA-Q and ALOHA-Q-DEPS will have the same throughput performance in steady state, whereas ALOHA-Q-EPS might have slightly lower throughput due to the random selection of transmission slots through continued exploration. Therefore, we run only ALOHA-Q in simulations. ALOHA-Q-EPS and ALOHA-Q-DEPS are implemented in practical experiments.

TABLE I
EXPERIMENT PARAMETERS

Parameters	Values
Channel bit rate	250 Kbits/s
Data packet length(ALOHA-Q)	1024 bits
Data packet length(Z-MAC)	840 bits
ACK packet length (simulation)	20 bits
ACK packet length (experiment)	144 bits
Slot length (simulation)	1050 bits
Slot length (experiment)	1200 bits
Experiment Period	500,000 slots
Learning rate (α)	0.1
RTS (SSA)	36 bits
CTS (SSA)	52 bits
Duty cycle (SSA)	10%

4.1 Linear-chain Topology

A linear network topology is created with 8 nodes lined up hop by hop, where the sink node is placed at the end of the chain as depicted in fig.2. The packets are generated at the source nodes and forwarded to the sink by the intermediate nodes. The number of source nodes can be varied up to the number of nodes in the network. For our implementation, networks with 1-source, 2-sources, 3-sources and all-sources topologies are evaluated respectively. According to the practical observations, packet transmission can be successful within one-hop neighbourhood (transmission range of 1-hop) but can be interfered within a 2-hop neighbourhood (interference range of 2-hop). It is very difficult to get a sharp boundary of interference range in real-world environments, so this model is commonly employed as a baseline with which to understand and compare protocol performance. Therefore, four neighbour nodes have to select different transmission slots to avoid collisions. The frame size should be appropriately set in order to allow every node in the network to find a unique slot which can be theoretically calculated with respect to the network topology. For a 1-source topology (node 1), the optimum frame size is 4 slots/frame, whereas it is 7 slots/frame for 2-source topology (node1 and node5) as the intermediate nodes along the source 2 route receive a packet and transmit 2 packets in a frame. The 3-source topology require 10 slots/frame as source 2 receives a packet and sends 2 packets in frame and source 3 receives 2 packets and transfers 3 packets in a frame. When all nodes act as source and intermediate, 22 slots per frame is estimated to be sufficient.

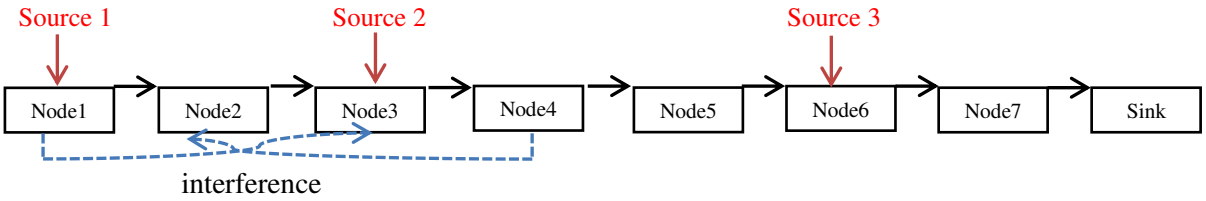

Fig. 2. 7-hop network topology and source nodes.

Figure 3 demonstrates the channel performance of ALOHA-Q, SSA and Z-MAC. The channel throughput exhibits an increasing trend and achieves its maximum value for all scenarios. The throughput stabilises at its maximum when the source nodes generate more traffic, because the traffic passed through the chain is limited by the frame structure. For example, the maximum throughput in the all-source scenario is 0.318 Erlangs (7 packets / 22 slots) as the sink node can receive seven packets at most in a frame. When the traffic load is low, Z-MAC achieves similar throughput performance. However, under a high level of contention, Z-MAC achieves a lower maximum throughput than ALOHA-Q due to greater overhead and potential for contention (nodes can potentially contend for their non-owned slots). SSA achieves a higher maximum throughput than ZMAC due to its lower overheads. Under medium to high traffic levels, contention and loss of control packets (RTS/CTS) have affected the throughput performance.

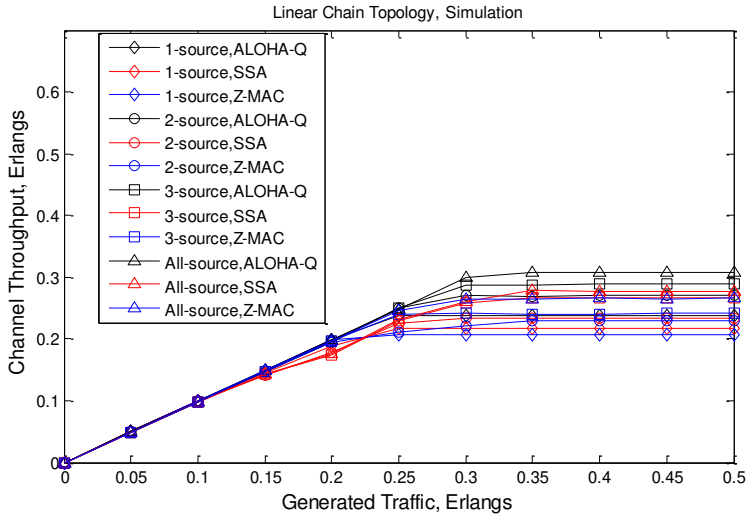

Fig. 3. Throughput comparisons.

Fig. 4 shows that the practical results closely agree with the simulations. In all practical experiments, saturated traffic conditions are considered where each node always has data to transmit. The behaviour of the channel throughput dependent on the frame size is observed. In all scenarios, the maximum throughput is achieved when the frame size is set to its

optimum value as estimated above. As the frame size increases beyond its optimum value, the channel throughput reduces because some slots are unused after each node finds a unique slot in the frame.

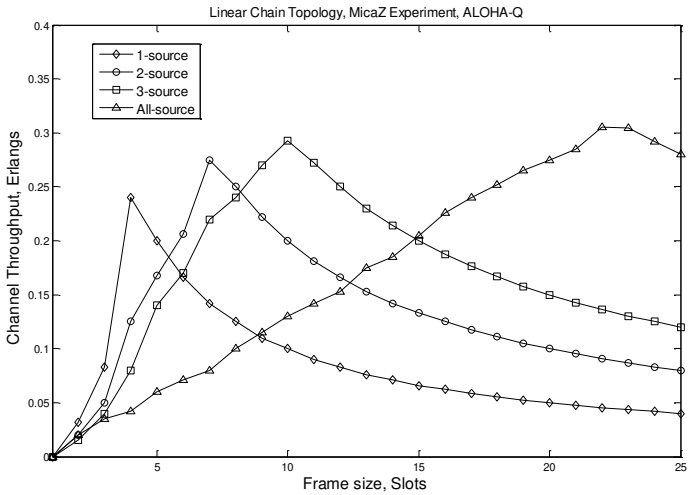

Fig. 4. Practical experiments of the channel throughput.

4.2 Grid Topology

A 12-node grid topology is considered as presented in fig. 5. Each node acts as a source, aiming to deliver all the generated and the received packets to the sink. Each node routes the packets using the shortest path. In all implementations, fixed routing paths as shown in fig. 5 are used and as in the chain topology, all nodes always have packet to send in practical implementation.

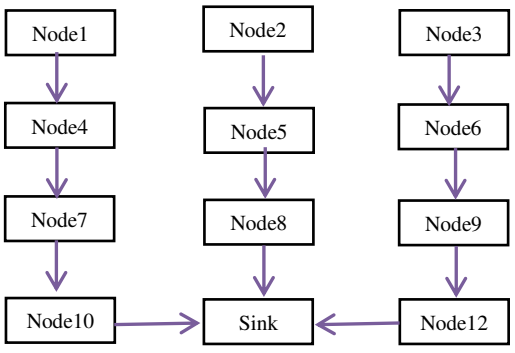

Fig. 5. Grid topology and the routing paths.

ALOHA-Q, SSA and Z-MAC have been simulated with various frame sizes to compare their performance in terms of channel throughput. Fig. 6 shows that ALOHA-Q achieves much higher throughput when the traffic load is heavy. This is because the large contention windows used for channel sensing limit the performance of Z-MAC. The performance of

ALOHA-Q and Z-MAC is almost identical at lower traffic loads. In all cases, the channel throughput of both schemes grows linearly with increasing traffic and reaches its maximum. The throughput of SSA is higher than ZMAC but lower than ALOHA-Q due to similar reasons as in chain scenarios. The performance with different wake-up intervals is similar because SSA does not control the traffic load by wake-up interval. The only difference is that a longer wake-up interval causes greater overheads (given the same duty cycle).

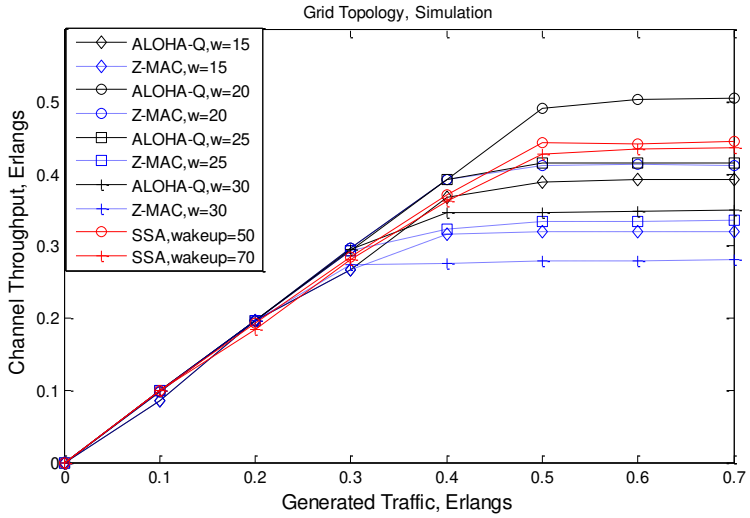

Fig. 6. Throughput comparisons.

In simulation, ALOHA-Q with the frame size of 20 slots/frame is the optimum and has a maximum throughput of approximately 0.5 Erlangs. However, practical experiments as presented in fig. 7 indicate that the maximum throughput can be achieved at a frame size of 23. The reason is that we set a 2-hop interference range in simulation, but the interference range varies in practice. This is due to the irregularity of the interference range, so that practical observations experience lower throughput, but the system nonetheless can achieve the steady state if frame size is set 23 slots/frame.

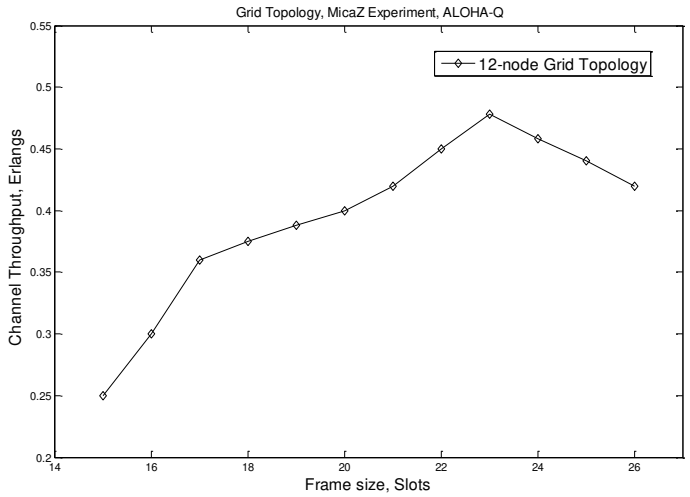

Fig. 7. Practical experiments of the channel throughput.

4.3 Random Topology

In most realistic WSN applications, nodes may have limited information about the number of neighbours and their locations due to the random nature of deployment. A fully-distributed network of 21 nodes as a more realistic deployment is constructed as presented in fig. 9. Shortest path is used for routing, and fixed routing paths are set in the nodes. The sink is located in the leftmost-bottom of the network. All nodes generate traffic and also operate as relay nodes.

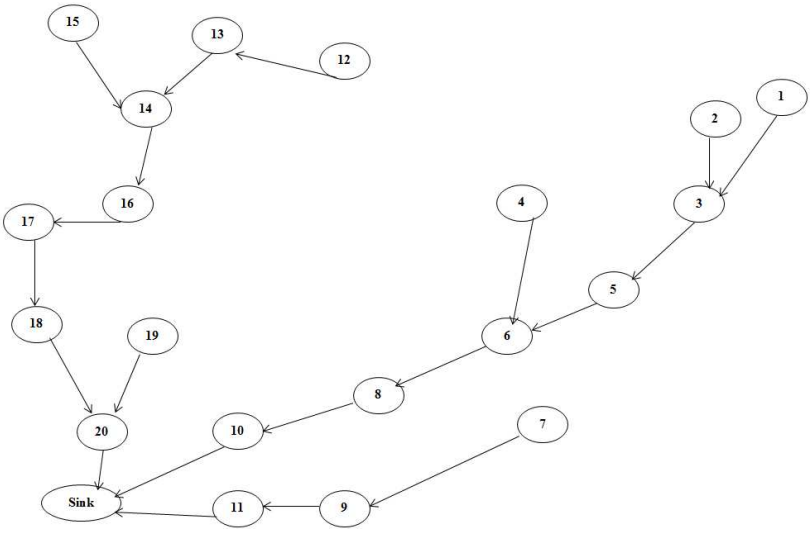

Fig. 8. Random topology

We evaluate and compare the performance of ALOHA-Q with the SSA and Z-MAC protocol with different frame sizes and increasing traffic levels in terms of the channel throughput,

delay and energy-efficiency. Based on the simulation results presented in fig. 9, the throughput of the three schemes grows linearly with increasing traffic load and converges to different limits. In all cases, ALOHA-Q has the same or superior channel throughput to SSA and Z-MAC at the same offered traffic level. The throughput performance of the ALOHA-Q and Z-MAC depends on the frame size (w). ALOHA-Q with 40slots/frame achieves the steady state with a maximum throughput of 0.48 Erlangs which is close to the theoretical limit of 0.5 Erlangs (20 packets / 40 slots) because the sink node can receive 20 packets at maximum in a frame if each node generates a packet. The performance of ALOHA-Q with 50 slots/frame exhibits a similar increasing trend but at a lower maximum because of the overestimate in the frame size. Z-MAC provides similar throughput performance at low traffic loads, but lower throughput due to the high overheads at high traffic loads. Due to the similar reasons discussed above, SSA has a higher throughput performance than Z-MAC but lower than ALOHA-Q.

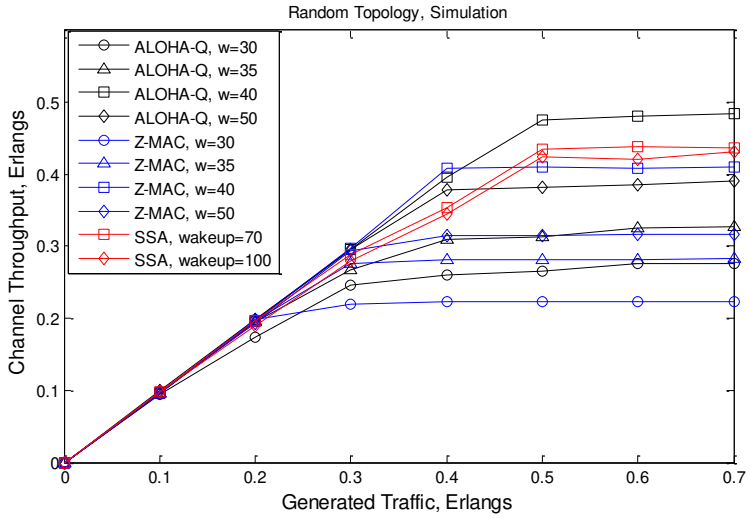

Fig. 9. Channel throughput with different frame sizes.

Fig. 10 shows the energy efficiency results per bit throughput which is calculated as (total energy cost / total amount of bits received for ALOHA-Q, SSA and ZMAC. In this figure, ALOHA-Q provides better energy consumption performance than SSA and Z-MAC. This is because Z-MAC uses clear channel assessment (CCA) in every slot when a packet is ready for transmission and some control messages such as syncronisation are periodically transmitted. ALOHA-Q with 40 and 50 slots/frame has a similar energy cost per bit because nodes avoid all collisions and achieve perfect scheduling. SSA provides a similar energy cost to ALOHA-Q at high traffic loads, but has higher energy costs than ALOHA-Q at low traffic

loads because of the fixed and reserved wake-up periods for exchanging control packets (RTS/CTS).

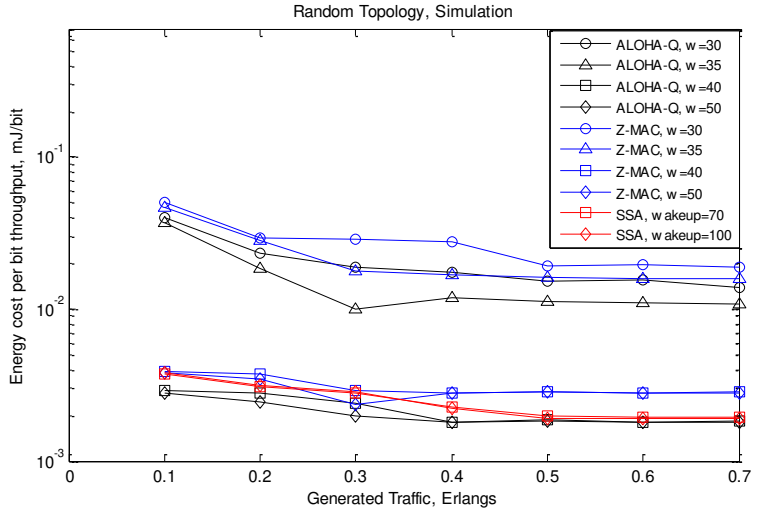

Fig. 10. Energy cost per bit throughput

Fig. 11 shows the end to end delay performance of the 3 schemes. Z-MAC has slightly better delay performance at lower traffic loads because the generated packets are immediately transmitted when the channel is clear. However, when the traffic load increases, only the owners and their one-hop neighbours can compete to transmit in the current slot. Therefore, Z-MAC has similar delay characteristics as the ALOHA-Q scheme at high traffic loads. Both schemes with less than 40 slots/frame have higher delay because of the collisions and retransmissions. On the other hand, SSA has higher delay than others due to the fixed wake-up intervals (a packet can be sent by the next hop at least one wake-up interval later).

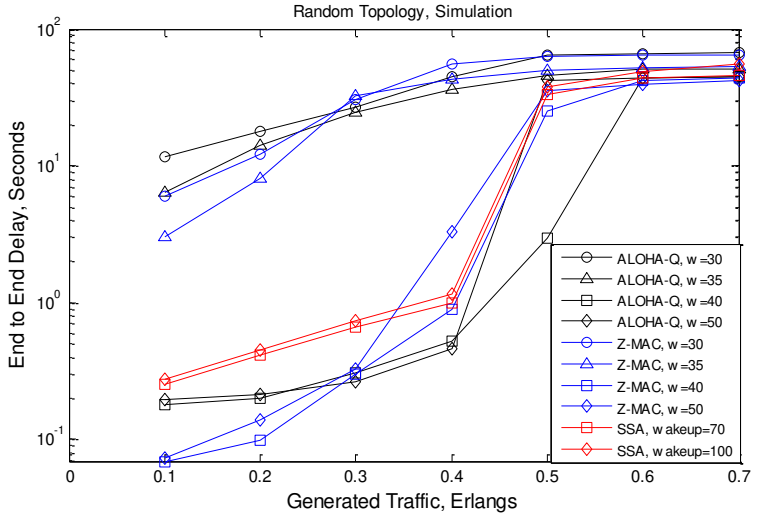

Fig. 11. End-to-end latency.

In fig. 12, the practical results validate the simulation results in which the system achieves the steady state with 40 slots/frame. Also, ALOHA-Q-EPS and ALOHA-Q-DEPS are implemented with 40 slots/frame. The running throughput in association with the frame size is presented as a function of experiment time. The system with 50 slots/frame can converge earlier because the frame is oversized and some slots are unused. Each point in the running throughput curves is obtained from the start to the end of a window which has a length of 50 frames. ALOHA-Q-DEPS converges earliest because it provides more exploitation with the increasing number of successful transmissions, resulting in it being more protective of a chosen slot. As expected, ALOHA-Q-EPS experience a lower maximum throughput due to random access in exploration. It takes a longer time to reach its maximum throughput value because of random transmissions in exploration.

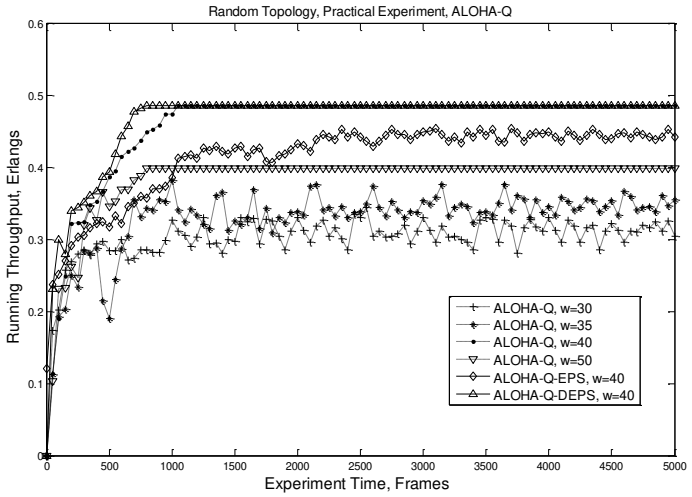

Fig. 12. Running throughput with different frame sizes.

4.4 Practical Issue; Acknowledgement packet loss

Our previous study [11] shows that IRIS sensor platform has a hardware issue which impedes the learning algorithm, resulting in some nodes failing to find unique slots. A certain amount of the acknowledgment (ACK) packets are not sent from the receiving nodes depending on the traffic load. This is because the sensor platform does not switch from reception mode to transmission mode very quickly upon a successful reception. Therefore, the transmitting node will assume that these packets that are not acknowledged are lost, even though they were received with success. We employed a guard band between the transmission and reception modes which enables the nodes to solve the problem but it incurs a significant overhead cost. In order to mitigate the effect of ACK loss, a new punishment scheme was proposed [11] that reduced the magnitude of the punishment on the Q-value update during the period in which

nodes search for an empty slot. The fixed value of the punishment is replaced with the probability of success in the preferred slot. The key benefit of this modification is to maintain the preferred slots based on the channel history. The system with this strategy was able to converge despite ACK loss.

ALOHA-Q was first implemented without guard bands in the presence of ACK loss using IRIS nodes. The practical loss of ACK packets impedes effective operation of the learning algorithm, resulting in the nodes not using unique transmission slots and causing collisions. The residual contention reduces the throughput significantly on the channel. ALOHA-Q and ALOHA-Q-EPS exhibits similar throughput performance at lower traffic loads, but ALOHA-Q-EPS has slightly lower throughput at higher traffic loads. Given that consecutive successful transmissions against infrequent failures will cause more frequent exploitation, the Q-value update will occur less frequently. Therefore, ALOHA-Q-DEPS serves to increase the channel throughput as the nodes have more willingness to protect the sequence of successful transmissions using the same slot. This is, however, not sufficient to enable all nodes to converge because of more substantial ACK packet loss when the channel load is high. ALOHA-Q-DEPS with the punishment modification described above provides the system with an extra level of protection in the transmission slots. Here, the system has a robust level of protection against the ACK loss. As a result of the modified learning process, all the nodes find unique slots and keep transmitting in these slots. Fig. 13 presents the results of all these scenarios.

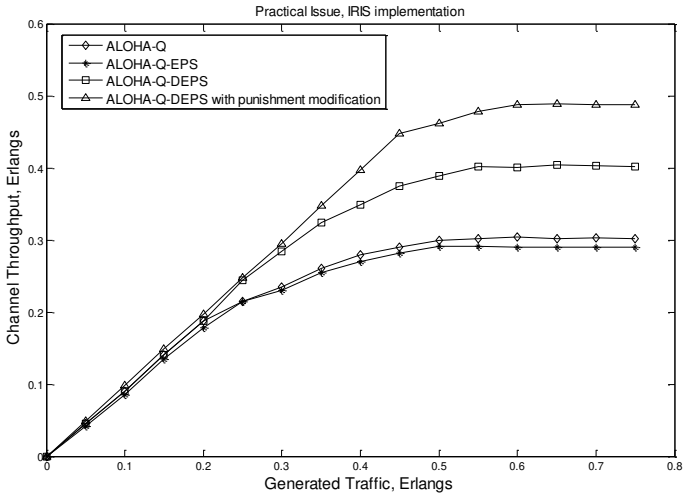

Fig. 13. Channel throughput

4.5 Extending the network with the participation of new nodes

A typical WSN needs a high level of self-organization and needs to be robust to environmental changes such as failure of nodes or addition of new nodes. Fundamentally, sensor nodes need to operate uninterruptedly for long periods from a limited-battery. Taking the constraints of sensor platform architecture into consideration, a well-designed MAC protocol should gracefully accommodate such network changes. From this perspective, the behaviour of ALOHA-Q and ALOHA-Q-DEPS is tested by adding new nodes as an event and depicted in fig. 14.

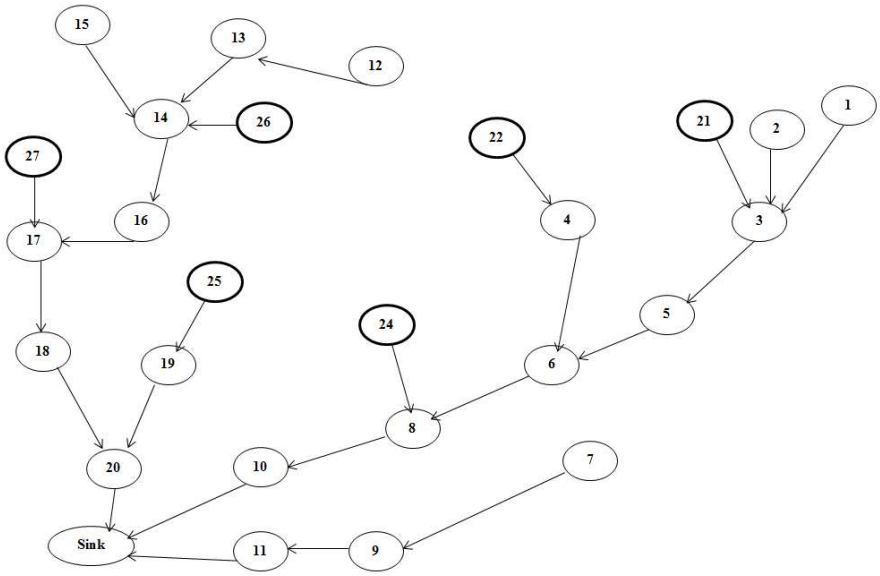

Fig. 14. Random topology with new nodes.

The results of both schemes highlights the importance of setting the frame size to an appropriately high level. The channel performance drops considerably as new nodes introduce more packet transmissions which requires the window size to be reset. In the programming of the sensor nodes, pre-defined timers are set to update the frame size at specific moments as shown in fig. 15. The channel throughput, as expected, increases with a higher frame size. Both schemes with 50 slots/frame operate in an adequate fashion. ALOHA-Q-DEPS provides better channel throughput than ALOHA-Q until the network converges. ALOHA-Q-DEPS in large networks can protect the channel performance significantly while the nodes adapt to network changes. However, ALOHA-Q is prone to network failures due to the continuous exploration.

Discussion: The main aim of a typical WSN deployment is to cover large areas with hundreads or even thousands of sensor nodes. The participation time of the new nodes into

the network will be unknown as it depends on the application requirements. Therefore, pre-defined timers in updating the frame size would not be a good idea in practice. *Code dissemination* [18] has recently been shown as a means of propagating new code in order to add new functionalities to the sensor networks, and can be applied to update the frame size as required here.

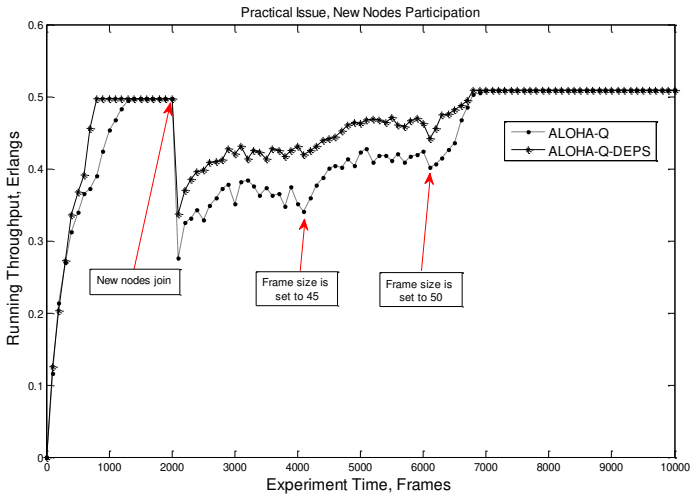

Fig. 15. Running throughput with new nodes.

5. Conclusion

This paper considers the MAC problem in an intelligent way, where a Q-learning strategy is integrated into MAC protocols design in order to explore efficient transmission schedules. The schedules are formed based on the Q-learning algorithm. A review of Q-learning based MAC protocols has highlighted the issues associated with achieving high channel performance and energy-efficient operation. We study the ALOHA-Q protocol, which combines ALOHA and Q-Learning, comparing its performance with SSA and Z-MAC through extensive simulations. Experimental evaluations have been carried out for the validation of simulation results. We showed that ALOHA-Q outperforms SSA and Z-MAC protocol in multi-hop networking. We have identified a problem with the ALOHA-Q protocol (the tradeoff between exploration and exploitation) and proposed an original solution, ALOHA-Q-DEPS. The channel performances of both ALOHA-Q and ALOHA-Q-DEPS have been evaluated for 2 practical real-world events: (1) packet losses caused by the sensor node hardware and (2) addition of new nodes to the network later on. ALOHA-Q-DEPS is more robust than ALOHA-Q in protecting the channel performance in dynamic environments.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *Communications Magazine*, 40(8), pp. 102-114, 2002.

[2] I. Demirkol, C. Ersoy and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: a Survey," *Communications Magazine*, 2006, 44(4), pp. 115–121, 2006.

[3] Y. Chu, P.D. Mitchell and D. Grace, "ALOHA and Q-learning based medium access control for wireless sensor networks," *International Symposium on Wireless Communication Systems*, pp. 511-515, 2012.

[4] Y. Yan, P.D. Mitchell, T. Clarke and D. Grace, "Distributed frame size selection for Q Learning based Slotted ALOHA protocol," *International Symposium on Wireless Communication Systems*,, pp. 733-737, 2013.

[5] Y. Yan, P.D. Mitchell, T. Clarke and D. Grace, "Adaptation of the ALOHA-Q protocol to Multi-hop Wireless Sensor Networks," 20th *European Wireless Conference*, pp. 921-926, 2014.

[6] I. Rhee, A. Warrier, M. Aia, J. Min and M. L. Sichitiu, "Z-MAC: A hybrid mac for wireless sensor networks," *IEEE/ACM Transactions on Networking*, 16(3), pp. 511-524, 2008.

[7] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," *IEEE INFOCOM*, pp. 1567-1576, 2002.

[8] Z. Liu and I. Elhanany, "RL-MAC: A QoS-aware reinforcement learning based MAC protocol for wireless sensor networks," *IEEE ICNSC*, pp. 768–773, 2006.

[9] M. Mihaylov and Y. L. Borgne Elhanany, "Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks," *Int. J. Communication Networks and Distributed Systems*, 9(3-4), pp. 207-224, 2012.

[10] G. Stefano, A. Liotta, and G. Fortino, "QL-MAC: A Q-Learning based MAC for wireless sensor networks," *Algorithms and Architectures for Parallel Processing*. Springer International Publishing, 2013. 267-275.

[11] S. Kosunalp, P.D. Mitchell, D. Grace and T. Clarke, "Practical implementation issues of reinforcement learning based ALOHA for wireless sensor networks," *International Symposium on Wireless Communication Systems*, pp. 360-364, 2013.

[12] M. A. Yigitel, O. D. Incel and C. Ersoy, "QoS-aware mac protocols for wireless sensor networks: A survey," *Computer Networks*, 55(8), pp. 1982-2004, 2011.

[13] J. Niu and Z. Deng, "Distributed self-learning scheduling approach for wireless sensor network," *Ad Hoc Networks*, 11(4), pp. 1276-1286, 2013.

[14] Sutton, R.S. and Barto, A.G., *Reinforcement learning: An introduction*, Cambridge, MA: MIT Press, 1998.

[15] Datasheet for MicaZ wireless measurement system available at [online] http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf, accessed July 2014.

[16] Datasheet for IRIS wireless measurement system available at [online] http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=264, accessed July 2014.

[17] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An operating system for wireless sensor networks," Ambient Intelligence, Springer-Verlag, pp. 115-148, 2005.

[18] H. Sangwon, and P. Ning. "Secure Code Dissemination in Wireless Sensor Networks," *Encyclopedia of Cryptography and Security*, Springer US, pp. 1099-1102, 2011.