A Q-learning Approach for the Support of Reliable Transmission in the Internet of Underwater Things

[o]S. Shivani, [o]A. Surudhi, [o]N. Prabagarane, and [*]L. Galluccio
[o]Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India
[*]Dipartimento di Ingegneria Elettrica Elettronica ed Informatica, University of Catania, Italy

***Abstract*—Underwater networks are characterized by undesirable time variability of the channel conditions due to Doppler effect, serious multipath and variable impulse response. Energy efficiency is a primary concern in these scenarios, because nodes batteries cannot be recharged or replaced. Accordingly, in order to increase network lifetime and effectiveness of underwater channel transmissions, in this paper, we present a reinforcement learning approach for communication in multihop underwater networks. The proposed methodology employs a Markov underwater channel model to characterize the link status that allows the relay device to choose the most efficient next hop node to forward data towards a remote gateway device for connection to the terrestrial internet. Simulation results are presented to show the effectiveness of the proposed methodology in terms of energy consumption and latency performance so allowing to increase network lifetime.**

I. Introduction

Oceans and seas represent about three quarters of the world surface and include a richness of ecosystems and environments that are considered almost completely unexplored. Internet of Underwater Things focuses on creating a worldwide network of smart interconnected underwater devices and objects. This will be an enabler to digitally connect our oceans and lakes to the "dry" Internet and realize a connected world of devices to answer a number of scientific questions about this undersea universe. To realize this vision, smartness of devices is needed in such a way to increase network lifetime, in spite of the high communication costs incurred in underwater.

Communications in water can employ three main paradigms, i.e., optical, acoustic and radio frequency (RF). RF communications, which are the most efficient under an energy point of view, perform poorly as we increase the distance from the sea surface [1]. Howbeit, RF waves are recommended for supporting communications between underwater devices close to the surface and terrestrial or surface stations located out of the water. Optical underwater communications are still at their infancy and require considerable hardware costs while suffering serious propagation issues when no line-of-sight communication can be provided (for example in case of seafloor obstacles). Acoustic communications are today the most widespread technique for underwater communications since they allow to trade-off relatively low hardware costs and good transmission ranges [1]. However, because of the relevant transmission energy costs as a result of the employed hardware, it is required to design appropriate energy efficient techniques for network management. Indeed, in underwater scenarios, devices' batteries cannot be recharged or replaced and, thus, maximizing network lifetime becomes a primary concern [2]. Support of energy efficiency should be performed jointly at all layers of the protocol stack and calls for considerations on both the channel conditions, and the position of nodes. Moreover, reduction in signaling data is required for the purposes of network management. As a matter of fact, all these features impact on the delivery delay performance, network lifetime and fairness in energy consumption.

In this paper, we present a machine learning framework for support of efficient underwater communications in noisy environments. The proposed approach considers an underwater IoT network, where acoustic modems equipped with sensing capabilities transmit data to a gateway device employed for communication with the terrestrial Internet. Underwater devices are vulnerable to possible bad channel conditions, noise and/or ongoing jamming actions. In order to circumvent these issues, in this work, we employ a Q-learning [3] approach to make optimal relay choices by taking into account their individual residual energy and the average energy of nodes in their neighborhood, to pursue a fair energy balancing inside the network. In this way, a trade-off between energy consumption, delivery delay and network lifetime, is achieved.

Underwater acoustic networks have been envisioned mainly for stand-alone applications and control of underwater scenarios with a substantial focus on network security. In [1], it was discussed that the traditional method of communication in underwater networks fails due to absence of failure detection, real-time monitoring and low storage capabilities. Moreover, there is a possibility of path loss and signal spreading while the signal propagates. Numerous design challenges, thus, emerge and are mainly related to low bandwidth, impaired channel as a consequence of multipath propagation and fading, very high propagation delay associated to the dynamic channel behavior, as well as high bit error rate and temporary loss of connectivity. Furthermore, the high installation costs of underwater networks compared to terrestrial networks result in a need for a trade-off between cost and power. As cited in [2], there exist many protocol solutions relying on the use of link quality information for cross layer relay determination to calculate optimal hop distance and reduce energy per bit of information sent [4]. *Software-defined networking (SDN)* [5],

This work was partially supported by MIUR under contract PONO3PE_00214_2 " Sviluppo e applicazioni di materiali e processi innovativi per la diagnostica e il restauro di beni culturali (DELIAS)".

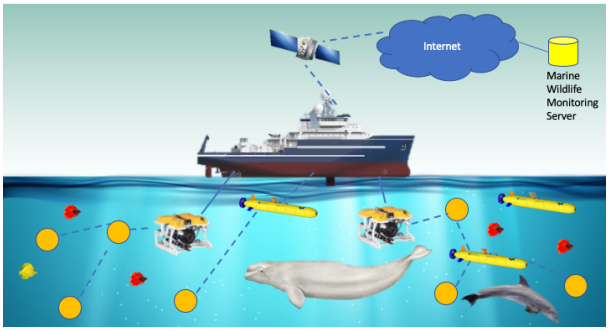

Fig. 1: System architecture.

which separates the data and control plane, has been recently introduced to support communication in underwater networks and increase security. However, this requires that the nodes exchange, at network set up, control traffic with a Controller node, which has to be underwater as well. In [6], deep neural networks are employed for symbol detection in orthogonal frequency division multiplexing (OFDM) systems by working at the physical layer. Similarly, in [7], machine learning is employed again for tuning physical layer parameters, such as modulation and coding. Nevertheless, none of the above approaches considers the use of machine learning at the network layer apart for [8], which does not take into account channel instability.

Taking inspiration from [8], in our work, we propose to use reinforcement learning (RL) to model an underwater system, where nodes are able to preserve energy efficiency and network lifetime by using network intelligence to tune transmission choices according to the channel state. In our work, Q-learning is combined with Markov channel modeling and the efficiency of this joint approach is assessed through simulation.

The rest of the paper is organized as follows. In Section II, we provide a description of the considered system. Section III details the distributed communication protocol employed by Underwater IoT nodes. In Section IV, we expound the Q-learning framework being considered while, in Section V, the Markov channel model is presented. In Section VI, we provide some numerical results to assess the effectiveness of the proposed approach. Finally, in Section VII, conclusions are drawn.

II. System architecture

In this section, we describe the system architecture considered in our work, as depicted in Fig. 1. Our network consists of different underwater devices that are able to sense, interpret and react to external conditions. These devices are heterogeneous, both in their capabilities (e.g., complex underwater vehicles or much simpler sensing devices equipped with acoustic transducers) and in the type of data they sense and elaborate (e.g., pictures of marine wildlife or data related to the salinity of water, variation in sea water acidity as a consequence of fossil fuels pollution, temperature, etc.,). Some devices could consist of vehicles remotely operated by a surface vessel by means of a cable connection (e.g., a remotely operated vehicle (ROV)), while others can be unmanned underwater vehicles (e.g., an autonomous underwater vehicle (AUV)), which can move without any human interaction, either close to the surface or in depth; other devices can be static. We also assume that the network of underwater things can be interconnected with the terrestrial IoT or with a remote marine wildlife monitoring center equipped with appropriate servers, where the collected data can be stored for future processing. A surface vessel equipped with long distance connection (e.g., satellite or cellular network) to an onshore station can allow to connect the underwater devices to the terrestrial network. Each underwater physical device can be thus characterized and described as a smart object, which, in order to efficiently collect and forward information, takes into account both real time and historical information on how the node acted in the recent past based on the sensory data and the physical conditions of the channel.

Interconnecting underwater devices is still a challenging task and standard approaches proposed for wireless terrestrial networks cannot be employed due to the high time variability of the underwater channel caused by Doppler effect and physical channel features, as well as due to noise caused by cargo or ships moving in the area. Moreover, latency in seawater is much larger than in wireless networks, because acoustic waves, as considered in this work, propagate relatively slowly (propagation speed around 1500 m/s) as compared to RF waves in the air. Accordingly, channel coherence time can be shorter than packet length, thus causing severe instability. Bit error rate (BER) is a very relevant issue, because, in case of multicarrier modulation techniques, it has been observed that the BER is still high and no lower than 10^{-5}. Therefore, use of smart effective communication protocols, able to support efficient information transmission is to be advocated in spite of the limited energy resources and the impossibility to recharge or replace nodes' batteries. In the following, we will illustrate the communication protocol run by underwater nodes, which employ a lightweight smart mechanism to choose the intermediate relayers to support simultaneously fair energy consumption and efficiency in path selection towards the gateway.

III. Communication protocol in a nutshell

In this section, we detail the communication protocol, which is employed by underwater sensor devices to send information to the gateway surface node. Nodes periodically send in broadcast, a packet carrying their ID, as well as their residual energy value. This information can be also piggybacked into data packets that they periodically send in order to reduce the signaling overhead. A source node, upon generating a packet, which has to be delivered to a surface gateway node, includes, together with the payload, its ID and residual energy level. Then the source issues this packet in broadcast to its one hop neighbors. Before sending this packet, based on the history of successes and failures and the information on its one hop neighbors' status, the sender identifies the best next relay and puts this information in the packet it issues. The one hop neighbors, and among them, the node, which has

been selected as best next relay by the previous relayer (or by the source itself), upon hearing this packet, extracts the sender information, in particular its residual energy, and its Q value, which is associated to the learning procedure detailed in the following sections. Consequently, the one hop neighbor of the current relay node being selected as next relay updates the corresponding entry in the local neighbor list that each node stores and will be able, on its turn, to calculate the Q-value as well for its one hop neighbors for selection at the next step. Nodes not selected as next hop relayers will discard the packet. In order to implement a confirmed relay service while also not increasing the redundancy too much, a node can get an implicit confirmation whether or not a packet is successfully delivered to other nodes by analyzing the traffic issued by the selected next hop relay node. Therefore, after a node sends a packet, the packet heard by the previous forwarder will be taken as an implicit acknowledgment; in case of successful transmission, the packet will be deleted from the memory. If instead the relayed packet will not be heard, retransmissions will be triggered till a maximum number of trials, after which, the packet is assumed to be lost. All nodes store the sequence of previous actions so that each current relay, upon choosing the next relay, can decide on the basis of previous forwarding experiences. The choice is done based on the exploitation of a Q-Learning methodology aimed at weighting properly the residual energy of a possible relay node and guaranteeing a fair distribution of energy into the network in such a way to choose the optimal relay and maximize lifetime.

IV. Mathematical model

This section is devoted to the description of the mathematical model employed to characterize the system behavior.

Reinforcement learning is a branch of machine learning, where agents take actions and, by trial and error interactions with the dynamically changing environment, aim to maximize a given reward. The description of the environment relies on a Markov Decision Process (MDP), which consists in a set of states $\mathbf{S}$, a set of actions $\mathbf{A}$, a reward function $\mathbf{R}$ and a state transition matrix $\mathbf{P}$ whose elements represent the probability of making a transition from a state s_i to a state s_j using action $a \in \mathbf{A}$. Similarly, elements in $\mathbf{R}$ represent the corresponding reward for making a transition from a state s_i to a state s_j using action a. Note that, actions that are taken do not affect only the sender's reward, but also impact on all the following evolutions of the system.

Q-learning [3] is a model-free RL methodology, which relies on the value of state-action pairs. It is a method, where agents can learn to act optimally in Markovian environments by estimating the consequences of their actions. A policy π is a way of associating each state, $s \in S$ and possible action, $a \in A$, to the probability of executing this action when in state s. The value of taking action a in state s under a policy π is defined as $Q(s, a)$ and represents the expected return for taking action a and following the policy π. If we take time evolution into account, the optimal policy at time t is denoted as $V^*(s_t)$ and is represented by the maximum over all possible actions $a \in \mathbf{A}$ of the value $Q(s_t, a)$, i.e.,

$$V^*(s_t) = \max_a\{Q(s_t, a)\}, \tag{1}$$

where,

$$Q(s_t, a) = r_t + \gamma \sum_{s_{t+1} \in S} p^a_{s_t s_{t+1}} \max_a\{Q(s_{t+1}, a)\}. \tag{2}$$

In (2), the term r_t represents the reward after executing any action a from state s at time t and is thus given by

$$r_t = \sum p^a_{s_t, s_{t+1}} R^a_{s_t, s_{t+1}}, \tag{3}$$

where, $p^a_{s_t, s_{t+1}} \in \mathbf{P}$ and $R^a_{s_t, s_{t+1}} \in \mathbf{R}$.

Calculation of the elements of the reward function $\mathbf{R}$ will be detailed in the following section while derivation of the transition matrix $\mathbf{P}$ will be described in Section V.

By iterations, $Q(s_t, a)$ can be updated as:

$$Q(s_t, a) \leftarrow (1-\alpha)Q(s_t, a) + \alpha[r_t + \gamma \cdot \max_a\{Q(s_{t+1}, a)\}], \tag{4}$$

where, α is the learning rate, which models the rate, at which we update the Q-values and $\gamma \in [0, 1]$ (also employed in (2)) is the discount factor of the rewards in the future in consideration that recent actions affect the current value more than future ones. More specifically, γ specifies the importance given to the future rewards and its estimates. When γ is set to 0, the system only considers the current reward and acts in such a way to maximize the reward in a short term perspective. When γ is set to 1, the system will try to achieve a long term relevant reward. A balance between these two opposite trends leads to a choice of γ in the range $\big[0.5, 0.99\big]$.

A. Calculation of the reward function

In underwater scenarios, in spite of the unreliability and time-variability of the link conditions, it is important to deliver data to the destination, in addition to maximizing network lifetime. This can be obtained by considering a reward function, which appropriately weights both the energy consumption at each individual node and the average consumption across all one hop neighbor nodes in such a way that a fair energy consumption distribution is supported in the network, hence, preserving network connectivity. More specifically, all nodes have an initial energy, which is assigned and spent to forward packets. Accordingly, residual energy keeps decreasing at each node n every time it is used as a relay. In order to account for this energy consumption, two terms, $c(n)$ and $d(n)$, can be defined, which have to be included in the reward function. In particular,

- $c(n)$ is the cost function related to the residual energy at a node n, i.e., $c(n) = 1 - \frac{E_{res}(n)}{E_{init}(n)}$
- $d(n)$ is the reward for energy distribution of a whole group of one hop sensor nodes, i.e., $d(n) = \frac{E_{res}(n) - E_{avg}(n)}{E_{init}(n)}$,

where, $E_{avg}(n)$ is the average energy of the group.

A reward function $R^{a_m}_{n,m}$ can thus be defined, which appropriately weights the costs in the one hop transmission from a

node n to a neighbor m[1]. In particular, if the transmission is successful, the reward function will be defined as:

$$R_{n,m}^{a_m} = -g - \alpha_1[c(n) + c(m)] + \alpha_2[d(n) + d(m)], \quad (5)$$

while in case of failure in transmission from node n to m, the reward function will be:

$$R_{n,n}^{a_m} = -g - \beta_1 c(n) + \beta_2 d(n). \quad (6)$$

In both equations, g is a constant cost incurred when a node attempts to forward a packet independently of the outcome of the packet transmission. Terms α_1 and α_2 should be chosen in such a way to appropriately weight the cost function, thus trading off reduction in the number of hops to the destination while also selecting nodes with higher residual energy. Similar considerations apply to β_1 and β_2. Equations (5) and (6) can thus be substituted in (3) to calculate r_t[2]. However, note that, $R_{s_t,s_{t+1}}^{a}$ is always negative and, thus, all the $Q(s,a)$ values for the non destination nodes are negative. The Q value of the destination node, will be set to zero, because it has to be the largest among all Q values. Based on the above described model, each packet forwarding attempt consumes energy, channel bandwidth, and contributes to the number of hops to the destination (i.e., delay). Based on the weights assigned to the cost terms, it is possible to balance between opposite targets: on the one hand trying to minimize the delay and, thus, the number of hops; on the other hand trying to balance the network energy consumption across the network, possibly increasing the hop counter, but with the advantage of improving network lifetime, because nodes remain alive for a longer time and the network gets more chances to remain connected.

In the next section, we will detail how the underwater channel status can be modeled in such a way to determine the values of the state transition matrix $\mathbf{P}$ needed to completely describe the Q-learning model.

V. Markov channel model

In this section, we present the Markov model used to characterize the behavior of the underwater channel. In underwater scenarios, an acoustic signal propagation from a sender to a receiver is impacted by seabed and sea surface effects resulting in a multitude of paths. As a consequence, waves traveling across different paths can lead to in phase or out of phase contributions. The propagation loss at different frequencies is also impacted by vertical temperature and pressure variations, as well as by salinity of water and pH. Sound waves will be impacted as well by ambient noise sources such as noise related to environmental features in proximity of the surface (e.g., wind, rain, etc.,) or exogenous sources, such as ship activity of cargoes and thermal noise or turbulence.

[1]In this work we identify the state $s_t = n$ with the condition when a packet at time t is held by node n and we identify action a_m as the action to forward a packet to node m.

[2]Upon substituting the above equations in (3) note that $R_{s_t,s_{t+1}}^{a} = R_{n,m}^{a_m}$ or $R_{s_t,s_{t+1}}^{a} = R_{n,n}^{a_m}$ depending on if the transmission was successful or not, and provided that at time t the status was $s_t = n$ and at time $t+1$ the state is $s_{t+1} = m$ or n and the action is a_m.

In [9], traces simulated by considering the Mediterranean sea parameters were considered to propose a Discrete Time Markov Chain (DTMC) of the underwater channel. A DTMC is a discrete-time stochastic process, where no memory is assumed, thus the current state of the channel only depends on the previous one and not on the history of previous process states. As a consequence, we can identify a transition probability matrix with generic element, $p_{i,j}$ representing the probability that the process is in the state j at time t provided that at time $t-1$ it was in the state i. In the transition probability matrix, the sum of the elements of each row is always equal to 1, i.e.,

$$\sum_{j=0}^{n} p_{i,j} = 1 \quad i = 0, 1, 2, ..., n. \quad (7)$$

These transition probabilities give information about the efficiency of the nodal links. Note, furthermore, that, the considered stochastic process describing the underwater channel state can be modeled through a DTMC provided that its stationary nature is proved. Accordingly, in [9], the authors used a Reverse Arrangements Test [10] and calculated the distribution of the sojourn time; in doing this, they assessed that it is exponentially distributed with the aid of the Kolmogorov-Smirnov test [11] and, subsequently derived the DTMC models of order K with the aim of capturing different degradation levels in the underwater channel. Similarly to what has been proposed in [9], in this work too, we consider a DTMC channel model of order K, where multiple channel states are possible. Upon considering a set of real traces with a Doppler frequency of 3 Hz, a BER of approximately 6% and assuming an OFDM multiplexing, the K-order model (K=3, i.e. 8 states) can be expressed as:

$$\mathbf{P} = \begin{pmatrix} 0.54 & 0.15 & 0.04 & 0.04 & 0.12 & 0 & 0.07 & 0.04 \\ 0.31 & 0 & 0 & 0.4 & 0 & 0 & 0.19 & 0.1 \\ 0.97 & 0.02 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0.9 & 0.09 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0.6 & 0.01 & 0 & 0.27 & 0 & 0 & 0.09 & 0.03 \\ 0 & 0 & 0 & 0.2 & 0 & 0 & 0.4 & 0.4 \\ 0.95 & 0.03 & 0 & 0.0 & 0.01 & 0 & 0 & 0.01 \\ 0.7 & 0 & 0 & 0.19 & 0 & 0 & 0.09 & 0.02 \end{pmatrix} \quad (8)$$

The choice to consider $K = 3$ comes from a trade-off between model complexity and accuracy. For worth of completeness, we also considered the possibility to model the underwater channel as $K = 1, 2$ state Gilbert-Elliott channel model [12], [13]. In this case, only a *good* and a *bad* channel states are considered and the transition probability matrix can be simplified as:

$$\mathbf{P} = \begin{pmatrix} 0.8718 & 0.1282 \\ 0.4659 & 0.5341 \end{pmatrix} \quad (9)$$

VI. Performance analysis

In this section, we present some preliminary results to assess the effectiveness of the proposed protocol solution as compared to standard routing approach, which do not take

into account any learning based on previous transmission history and consider only shortest path routing to deliver data to the gateway device. In particular, we assumed a network topology, where 10 underwater nodes are deployed at different depths. Nodes are assumed to be in contact with each other if their distance is lower than 300 m, provided that the channel state is good and transmission can be successfully supported. In our scenario, we assume that node 10 is located on the sea surface and is the gateway device towards which all transmitted packets have to be delivered. Packets can be generated randomly by any network node. An initial energy of 1 KJ is available at each network node and, for each packet transmission being executed, 1 J of energy is spent. Parameters for RL and reward function have been chosen as $\alpha_1 = \beta_1 = 0.5$, $\alpha = 0.5$ $g = 1$ and $\gamma = 0.5$. Parameters $\alpha_2 = \beta_2$ vary in such a way to have $\alpha_1/\alpha_2 = \beta_1/\beta_2$.

As discussed above, two Markov models for the underwater channel characterization are employed, namely the one in (9) and the one in (8) to consider a 2 states and an 8 states model, respectively. For worth of comparison, we also consider a shortest path (SP) approach, where a node, upon needing to send a packet to the gateway, only looks for the shortest path to this node. Furthermore, we consider that, when 2 nodes i and j are in each other's coverage range (i.e., their distance is lower than 300 m), the link conditions are variable and, thus, when considering the 2 states model, if the channel state is good, transmission will be successful, otherwise it will not. In case of an 8 states model, depending on the link status, G_i, where $i \in \{1, 2, \dots 8\}$, transmission will be successful with probability Π_i given by the solution of the Markov conditions [14], i.e.,

$$\begin{cases} \mathbf{\Pi} \cdot \mathbf{P} = \mathbf{\Pi} \\ \sum_i \Pi_i = 1 \end{cases} \tag{10}$$

In Fig. 2, we illustrate the average latency in the network (intended as the average number of delay units from a sender node to the gateway) as a function of the ratio α_1/α_2. Note that, when the ratio α_1/α_2 is lower than 1, which implies that α_2 takes higher values, the system gives higher priority to delay and thus the latency decreases. As soon as we increase the ratio instead, the weight given to delivery delay decreases as α_1 becomes several times larger than α_2, i.e., the value of α_2 decreases. As a result, the latency increases. However, note that, for shortest path case, the average delivery delay is not impacted by the ratio α_1/α_2 and the delivery delay is significantly higher than that obtained by applying the Q-learning approach. This is because the shortest path does not consider the link status but only the distance. Furthermore, from Fig. 2, we may observe that, the Q-learning approach yields reduced average latency than SP. This is because, upon choosing the path from the sender node towards the gateway, in case of shortest path the link status is not accounted and, thus, multiple retransmissions can be needed.

Fig. 3 shows the normalized network residual energy obtained using the proposed approach, again as a function of the ratio α_1/α_2. Note that, the average residual energy remains

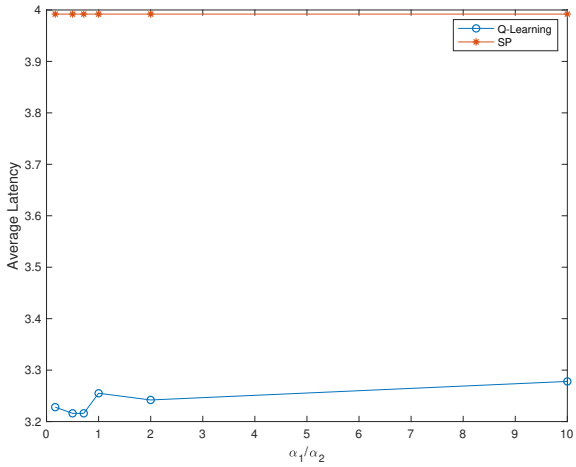

Fig. 2: Average Latency.

almost always constant in the range $[85\%, 86\%]$. Similarly, in the shortest path case, the residual energy remains constant as well and comparable values are achieved.

In Fig. 4, we report the standard deviation in the residual energy at nodes. Observe that, for low values of the ratio α_1/α_2, we do not care much about energy consumption while we are more concerned about delay. Accordingly, most nodes are active although some of them are more stressed and congested due to their positions, while others are not. Correspondingly, different residual energy is obtained at different nodes and thus, the standard deviation of the residual energy increases. On the other side, upon increasing the ratio α_1/α_2, more importance is given to energy consumption and, thus, less nodes remain alive, while others go to sleep mode; so the residual energy at these remaining alive nodes is the same, thus resulting in lower standard deviation. Note also, how the use of a Q-learning approach as compared to a standard SP methodology allows us to achieve much better performance in terms of equalization of energy consumption. The above three figures refer to the case of 2 state Markov model.

In order to study the effect of using a more complex channel state model that accounts for multiple states (i.e., 8 states), we demonstrate in Fig. 5 and Fig. 6, the residual energy at each network node as a function of the number of packets sent across the network for two Markov models, i.e., 2 states and 8 states. From the results of Fig. 5 and Fig. 6, we may note that, both models describe the channel in almost an identical way. Based on this observation, we advocate that, the 2 states model with $K = 1$ is preferred as it will result in reduced complexity.

VII. Conclusions

In this paper, the problem of efficient transmission in underwater networks is addressed. In particular, in order to realize the vision of an internet of underwater things, it is important to increase network lifetime and transmission efficiency in spite of the undesirable time variability of the underwater channel related to Doppler effect, serious multipath and resulting variable impulse response. By remembering that

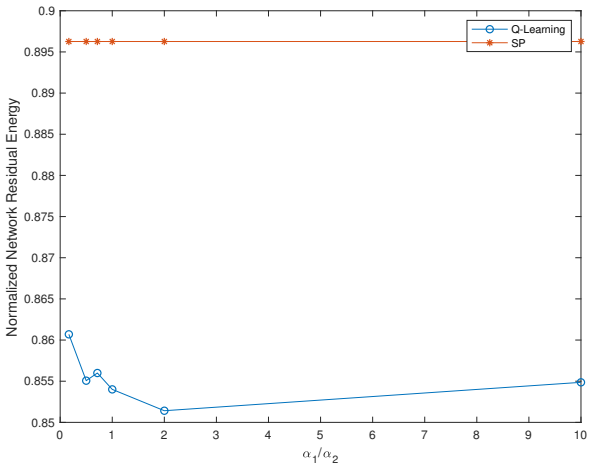

Fig. 3: Normalized network residual energy.

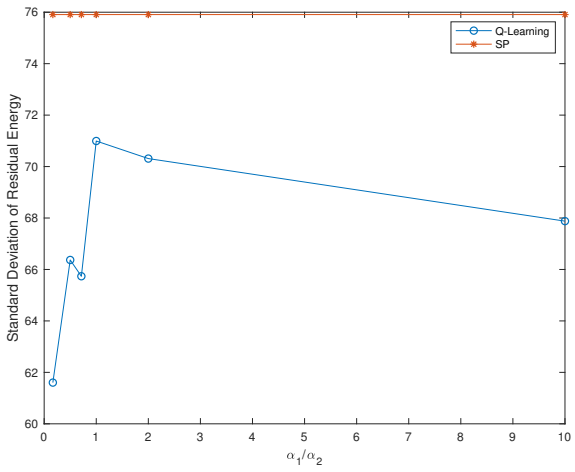

Fig. 4: Standard deviation of the residual energy.

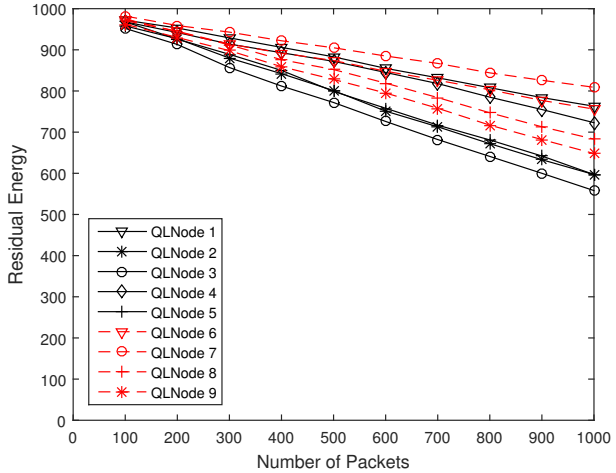

Fig. 5: Residual energy for the Markov model with $\alpha_1/\alpha_2 = 3$ and $K = 1$.

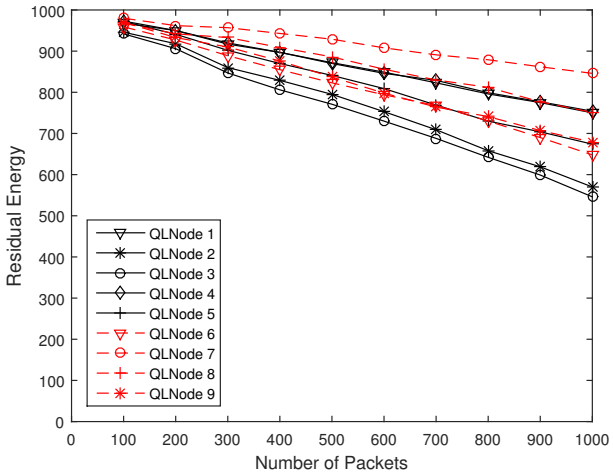

Fig. 6: Residual energy for the Markov model with $\alpha_1/\alpha_2 = 3$ and $K = 3$.

energy efficiency is a primary concern since nodes batteries cannot be recharged or replaced, we presented a reinforcement learning approach for communication in multihop underwater networks. The proposed methodology employs a Markov underwater channel model to characterize the link status and allow the relay device to choose the most efficient next hop node to forward data towards a remote gateway device for connection to the terrestrial internet. Performance results proved the effectiveness of the proposed methodology in terms of fair energy consumption distribution inside the network and improvement of network lifetime.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[2] P. Casari and M. Zorzi, "Protocol design issues in underwater acoustic networks," *Computer Commun.*, vol. 34, no. 17, pp. 2013–2025, 2011.

[3] G. Cybenko, R. Gray, and K. Moizumi, "Q-learning: A tutorial and extensions," *Mathematics of Neural Networks*, 1997.

[4] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "Channel-aware routing for underwater wireless networks," in *proc. IEEE Oceans*, 2012.

[5] I. F. Akyildiz, P. Wang, and S.-C. Lin, "Softwater: Software-defined networking for next-generation underwater communication systems," *Ad Hoc Networks*, vol. 46, pp. 1–11, 2016.

[6] Y. Z. et al., "Deep learning based underwater acoustic OFDM communications," *Appl. Acoust.*, vol. 154, 2019.

[7] M. S. M. Alamgir, M. N. Sultana, and K. Chang, "Link adaptation on an underwater communications network using machine learning algorithms: Boosted regression tree approach," *IEEE Access*, vol. 8, 2020.

[8] T. Hu and Y. Fei, "Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Trans. on Mobile Comput.*, vol. 9, no. 6, pp. 796–809, 2010.

[9] F. Pignieri, F. De Rango, F. Veltri, and S. Marano, "Markovian approach to model underwater acoustic channel: Techniques comparison," in *proc. MILCOM 2008-2008 IEEE Military Commun. Conf.*, 2008, pp. 1–7.

[10] J. Bendat and A. Piersol, *Random Data: Analysis and Measurement Procedures*. Wiley, NY, 1986.

[11] C. Montgomery, *Applied Statistics and Probability for Engineers*. Wiley, NY, 2003.

[12] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Tech. Journal*, vol. 39, 1960.

[13] E. O. Elliott, "Estimates of error rates for codes on burst-error channels," *Bell Systems Tech. Journal*, vol. 42, 1963.

[14] W. Sullivan, *Markov Models*. PublishDrive, 2019.