Reinforcement learning-based link adaptation in long delayed underwater acoustic channel

Jingxi Wang¹, Chau Yuen², Yong Liang Guan³, and Fengxiang Ge^{1*}

¹Beijing Normal University, CIST, 100875 19 St. Xijiekou Beijing, China

²Singapore University of Technology and Design, Engineering Product Development, 487372, 8 Somapah Road, Singapore

³Nanyang Technological University, EEE, 639798, 50 Nanyang Avenue, Singapore

Abstract. In this paper, we apply reinforcement learning, a significant area of machine learning, to formulate an optimal self-learning strategy to interact in an unknown and dynamically variable underwater channel. The dynamic and volatile nature of the underwater channel environment makes it impossible to employ pre-knowledge. In order to select the optimal parameters to transfer data packets, by using reinforcement learning, this problem could be resolved, and better throughput could be achieved without any environmental pre-information. The slow sound velocity in an underwater scenario, means that the delay of transmitting packet acknowledgement back to sender from the receiver is material, deteriorating the convergence speed of the reinforcement learning algorithm. As reinforcement learning requires a timely acknowledgement feedback from the receiver, in this paper, we combine a juggling-like ARQ (Automatic Repeat Request) mechanism with reinforcement learning to minimize the long-delayed reward feedback problem. The simulation is accomplished by OPNET.

1 Introduction

In an underwater scenario, electromagnetic waves and optical waves decay rapidly and are sharply absorbed, restricting the range of communication. In contrast, acoustic waves can be transmitted relatively more successfully for a longer distance without absorption loss of similar magnitude. Therefore, acoustic wave is the only effective medium to communicate underwater. However, underwater acoustic channel is a complicated channel with time-varying and spatial-varying factors i.e., multi-path effect, background noise interference and the Doppler shift effect etc.

Other unique challenges faced by the underwater channel in comparison with air wireless channel includes a highly dynamic, long propagation delay, bandwidth limitation and high transmission error rate, highlighting the clear distinctions between the two channels.

Compared to the wireless networking standard 802.11 developed in the IEEE Standards Association, currently no digital underwater communication standards exist[1]. In standard 802.11, numerous parameters are proposed to obtain higher throughput. Accordingly, it is a challenging task to select the link parameters based on underwater channel conditions due to the large pool of configuration set, such as number of spatial streams, channel bonding, guard intervals, frame aggregation and different modulation and coding schemes [2]. Machine learning has attracted attention currently due to its flexibility in a variety of area, specifically, in extracting

information from raw data. As a significant branch of machine learning, reinforcement learning (RL) is optimal for interactions in an unknown and dynamically variable environment. RL is also good at self-learning in order to design a suitable strategy. Multiple RL algorithms are employed to address the selection problem among the large configuration set.

For underwater acoustic link adaptation, the channel environment is constantly dynamic and unpredictable, making it impossible to employ channel quality information in order to select the optimal parameters to transfer the data packets. RL can be adopted to resolve the problem and performs better throughput without any environment pre-information. Besides, in underwater, the speed of an acoustic wave is about 1500 m/s, with 0.67 s/km transmission delay. This is unacceptable in RL as timely feedback acknowledgement is essential during the learning process.

For middle range communication (1-10 km), it may yield about 1 to 7 seconds delay, which might lead to slow convergence and throughput decrease. Besides, due to limitations caused by turbulence in an underwater channel, underwater modem is currently operating in half-duplex, which means the underwater devices are unable to transmit and receive packets simultaneously [3] as per the norm in air wireless communication.

In this paper, RL is proposed to enhance the adaptability of routing algorithm, simultaneously reducing transmission path selections and energy consumption. The nodes continuously learn from the

* Corresponding author: ge@bnu.edu.cn

environment based on the feedback information; for instance, modulation and coding scheme, residual energy and packets error rate. MDP (Markov Decision Process) is adopted to address the problem of spatial-temporal uncertainty caused by low bandwidth and high latency as well as insufficient network state observations in underwater acoustic sensor network, and in the end improve network throughput. Several papers have employed RL to choose the optimal selections to obtain higher throughput but none of them has considered the long propagation delay in underwater; all the rewards in the RL algorithm are considered to be received immediately, unlikely in a real underwater situation.

In [4][5], they provide rudimentary methods of the combination of RL and underwater link adaptation. Early state of RL is employed to tune the link parameters. But only a few modulations and coding schemes are taken into consideration. In [6][7], RL is proposed as an on-line learning algorithm to optimize the link adaptation with minimal assumption of the operating environment. It is inadequate under current tuning configurations for IEEE 802.11ac standard, which concludes guard interval, frame aggregation and multiple bandwidth [4][8]. And [9] is fundamental in the previous standard IEEE 802.11n with the same idea and application of reinforcement learning. In [10], due to the large number of configurations, i.e. multi-antenna and multiuser, pre-coding, spatial mode selection and also limited feedback about the channel, machine learning algorithm is used to select the modulation and coding scheme based on the IEEE 802.11ac standard.

This research aims to improve throughput of the underwater communication link by using RL in a long delay scenario to select the optimal schemes and parameters for the next transmitting packets adaptively. Current Channel State Information (CSI) cannot be directly adopted as simultaneous channel indicator due to the highly dynamic feature of underwater channel.

There are four main parts in RL, namely, agent, states, actions, rewards (more detailed explanations will be presented later). The goal of RL is to let the agent learn and determine the future action given state to gain maximum rewards. Rewards in underwater link adaptation can be indicated through the packet error rate or frame error rate from the receiver [4][13], which takes long delay to send the acknowledgement feedback to the transmitter (the agent). The delayed reward may cause the slow convergence of the RL algorithm and lower throughput. Therefore, in this paper, we propose an algorithm to address the implementation of the RL in tuning underwater communication link, taking the long delayed rewards into consideration.

2 System Model

2.1 Underwater modem parameters

Currently, the most commonly used modulations are BPSK (2-differential phase shift keying), QPSK (4-differential phase shift keying), and 16QAM (Quadrature Amplitude Modulation)[5,7,9]. The adoption of different

modulation depends on the underwater acoustic channel quality, such as signal-to-noise ratio (SNR) and multipath spread.

Forward error correction (FEC) encoder is employed to combat with changeable and unpredictable underwater channel and also to deal with highly-cost re-transmission. FEC encodes the data with extra redundancy, which enables the receiver to detect the errors and correct the errors without retransmission. This merit is critical in long delayed underwater communication. Two different coding rates are considered in our underwater modem parameters, namely, $\frac{1}{2}$ and $\frac{3}{4}$. We consider two patterns of packets transmitting, based on the varying rate of the channel. That is, if the channel stays relatively stable or slowly varying, the transmitter prefers to adopt longer series of packets, i.e. 8 packets. However, when the channel is volatile, the transmitter will tend to send packet in series of 4 to compensate for the volatility between the transmitting time.

Bandwidth is severely limited in underwater acoustic channel due to characteristics of sound attenuation via scattering loss and sea water absorption. For middle range underwater communication, i.e. from 1km to 10km, two bandwidths, 400Hz and 800Hz, are available to be selected by the transmitter based on the channel information.

Guard interval is used to guarantee different transmission will not interfere with each other and then avoid the propagation delay, multipath and reflection problems. Guard interval is set to be either 0.05s or 0.1s. Eventually, the transmitter has to choose the transmission power level based on the link attenuation. The power level contains 500w and 1000w. Setting distinct power level is to reduce the energy consumption when lower power level can arrive at the receiver successfully.

Table 1. MCS Index.

MCS index	The number of packets	Modulation	Coding rate	Bandwidth	Guard interval	Power
0	4	BPSK	1/2	400Hz	0.05	1000
1	4	QPSK	1/2	400Hz	0.05	1000
2	4	QPSK	3/4	400Hz	0.05	1000
3	4	16QAM	1/2	400Hz	0.05	1000
4	4	16QAM	3/4	400Hz	0.05	1000
5	8	BPSK	1/2	800Hz	0.05	1000
6	8	QPSK	1/2	800Hz	0.05	1000
7	8	QPSK	3/4	800Hz	0.05	1000
8	8	16QAM	1/2	800Hz	0.05	1000
9	8	16QAM	3/4	800Hz	0.05	1000

In conclusion, there are 3 different modulations, 2 coding rates, 2 different bandwidths, 2 packets amount, 2 spans of guard interval, and 2 levels of transmission power to be considered. Therefore, exempting several configurations, in total, there are 80 modem parameters can be selected. Table 1 has shown 10 out of 80 MCS index.

2.2 Underwater acoustic channel

Underwater acoustic communication is severely limited from numerous aspects, such as low transmission velocity, transmission attenuation increasing by frequency, and complicated ambient noise. The speed of acoustic wave is about 1500 m/s, but the exact sound velocity at given point is affected by water temperature, salinity and depth etc. Doppler effect is also one of main feature of underwater acoustic communication due to low velocity of sound in the water. However, in our model, the nodes are relative fixed, or at most suffer unintentional motion at 0.5 m/s, which can be ignored in our simulation [14]. Another character which is critical for underwater acoustic channel is propagation loss, during which the acoustic energy is absorbed by seawater medium. The propagation loss is a function that depends on distance l and signal frequency f ,

$$A(l, f) = l^k [\alpha(f)]^l \quad (1)$$

where k is the extension factor and $\alpha(f)$ denotes the seawater absorption coefficient. For given frequency f and given transmission power P , the receiving power can be expressed as transmission loss (TL),

$$\begin{aligned} TL &= 10 \log(A(l, f)) \\ &= k \cdot 10 \log(1000 \times l) + l10 \cdot \log \alpha(f) \end{aligned} \quad (2)$$

In practice, k usually takes 1.5, and the absorption coefficient can be represented by *Thorp* function in dB,

$$10 \log \alpha(f) = 0.02 + 0.11 \frac{f^2}{1+f^2} + 0.011 f^2 \quad (3)$$

The underwater ambient noise is complex, being a cumulative function of tidal, wave, shipping, thermal noise etc. Explicitly, the seawater ambient noise can be described by *Wenz* model, which contains turbulence noise, shipping noise, wind noise and thermal noise,

$$\begin{cases} 10 \log N_t(f) = 17 - 30 \log f \\ 10 \log N_s(f) = 40 + 20(s - 0.5) + 26 \log f - 60 \log(f + 0.03) \\ 10 \log N_w(f) = 50 + 7.5 w^{1/2} + 20 \log f - 40 \log(f + 0.4) \\ 10 \log N_{th}(f) = -15 + 20 \log f \end{cases} \quad (4)$$

where N_t , N_s , N_w , and N_{th} respectively represents turbulence noise, shipping noise, wind noise and thermal noise, and s is the shipping density. w denotes the wind velocity of water surface. Numerous aspects have been considered in *Wenz* model, perfectly reflecting the feature of underwater channel and bringing severe dynamic ambient noise. In conclusion, the overall noise power spectral density can be expressed by (5)

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \quad (5)$$

The channel simulation is realized in pipeline stage in OPNET, and several wireless channel models have been set up to simulate the dynamic environment. Here are more details about channel simulation setup. *Wal_txdel* and *Wal_propdel* models compute transmission delay, which additionally take guard interval into consideration and also count modulation time (which is important in underwater scenario and the design for J-ARQ). *Thorp* function and *Wenz* model are implemented in *Wal_power* model and *Wal_noise* model respectively to determine the transmission loss and SNR,

which cause dynamic and uncertainties in underwater wireless channel.

2.3 RL combined with Juggling-like ARQ

For middle range underwater communication, i.e. 1 km to 10 km, the propagation delay can be up to 6.7s. This resulted in the delay of the acknowledgement from the receiver to be up to 13.4s, double of the propagation delay. This greatly decreases the throughput and also the learning efficiency of RL, as it is shown in Fig.1.

Juggling-like ARQ (J-ARQ) as it illustrated in in Fig.2, is effective to deal with long delayed communication [6], which utilizes the idle time between waiting for the acknowledgement signals from the receiver, to keep sending packets with a periodic interval.

As most underwater devices operate in half-duplex, meaning both transmitter and receiver cannot send and receive data simultaneously; in this case the periodic interval plays two roles. Firstly, to allow the receiver to send the feedback acknowledgement; secondly to enable the transmitter to receive the previous acknowledgement. The key factor of J-ARQ is to design the length of the interval to make sure that the acknowledgement can be successfully received at the transmitter side, and will not collide with the sending packets.

Fig. 1. Stop and wait for half-duplex operation. The grey strip presents the transmitting data packets and the bold black line denotes the ACK feedback.

Fig. 2. J-ARQ for half-duplex operation. The shadow area denotes the maximum possible time zone that the ACK feedback would arrive at the transmitter side.

The period P as it is illustrated in the Fig.2 consists of two parts, one is the transmitting time of the sending packet t_T , another one is the ACK possible arriving time t_A .

$$t_A = t_o + t_R + \delta \quad (6)$$

where t_o is the packet overhead time, t_R denotes the ACK transmitting time, and δ represents the guard time causing by uncertainty of the transmission. Therefore, the transmission period should satisfy (8)

$$P = n \cdot t_T + t_A \quad (7)$$

$$kP = 2D \quad (8)$$

where k refers to the number of transmitting period waiting for one ACK coming back. Obviously, $k \geq 1$. D represents the propagation delay from transmitter to receiver. For each sending episode, n packets will be transmitted. In order to make J-ARQ possible,

$$kP = k(n \cdot t_T + t_o + t_R + \delta) \leq 2D \quad (9)$$

hence,

$$k \leq \frac{2D}{(n \cdot t_T + t_o + t_R + \delta)} \quad (10)$$

therefore, for any given transmission distance and related transmitting characters, we can figure out how many transmitting periods we can exploit while waiting for long-delayed ACK.

Despite the half-duplex limitation, packets are being constantly sent while the corresponding ACK is received during the interval. In this way, J-ARQ improves throughput.

J-ARQ is inefficient when two nodes are positioned too close to each other because the delay time is insufficient in between packets. While for long-delayed communication, J-ARQ is effective to improve the throughput by utilizing the idle time while waiting for the ACK.

Synergistically, J-ARQ also makes contributions to improve the convergence speed of the RL algorithm. RL is employed to select the MCS index intelligently to be adaptive to the fast varying underwater acoustic channel. In proposed RL algorithm, the reward in link adaptation is the feedback ACK received, which means the reward will definitely be long delayed. The long-delayed reward deteriorates the learning speed of RL agent. J-ARQ will keep sending the packets with a guard time, and simultaneously receive the feedback reward during the guard time. Therefore, when RL is combined with J-ARQ, the reward feedback which will receive by transmitter agent will increase for any given time period. The agent will process more data and learning more, and then get to track the varying channel better. In the end, total throughput will increase.

3 Reinforcement Learning

3.1 State

Due to the volatile nature of an underwater acoustic channel, it is not feasible to obtain the channel information to represent the state of the channel. RL algorithm stands out in this case on the basis that it is independent of any model. We can define our state as following. The transmitter continuously sends a certain amount of packets, but each packet adopts different MCS index. If one or some of them is suitable to the current channel, it will be successfully demodulated at the receiver side. So we define the proportion of the success packets as the state of RL. The learning agent of RL is to achieve 100% successful transmitted proportion. In this way, we avoid having to utilize the complex underwater acoustic channel characters as the feedback to the transmitter, which is dynamic and thus unreliable.

3.2 Action

The actions available in link adaptation is numerous MCS indexes shown in Table 1. ϵ – greedy is used to balance the exploration and exploitation dilemma. Even

though the final goal of RL is to accumulate as much reward as possible, if the agent always adopts the most rewarded actions cumulatively, and not explore the new one, better route of actions not yet discovered may be omitted. ϵ – greedy set a small probability to discover new actions, like $\epsilon = 0.1$. The agent will explore new action space with 0.1 probability.

3.3 Reward

The reward is based on the successful proportion that is received on the receiver side. As it is materially possible that there are different MCS indexes realizing the same successful state, extra definition of reward should be added. Distinct MCS index has different performance, 16QAM leads to higher throughput and lower transmission power level, reducing energy consumption. By rewarding more highly the MCS index with better performance and higher energy efficiency, the agent shall learn and devise a better strategy. The reward information is attached in the ACK back to transmitter, which contains both the reward information and the corresponding MCS index.

3.4 Agent

The agent will process the actions it takes and feedback on the corresponding states and actions, with the final goal to accumulate the highest possible reward. This results in a continuous improvement in performance when tracking the dynamism of the channel in link adaptation. The key part of the agent is the value function, i.e. Q-function. By updating Q-function, the agent will form an Q-value matrix and the agent takes actions based on the value of matrix.

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma Q(s' + a') - Q(s, a)) \quad (11)$$

where s is the current state, and a represents the action the agent takes. s' and a' refers to the next state and the next action respectively. r is the reward. α is the learning rate and γ denotes the attenuation factor.

4 Performance Analysis

The performance of our algorithm is compared with the stop and wait ARQ (S&W-ARQ), which does not utilize the idle time and only sends the next packets episode until the previous ACK is sent back. In this way we can clearly compare the performance of J-ARQ combine with our RL algorithm.

The node model, as it illustrated in Fig.3, contains the process model, energy model, and a pair of transmitter and receiver. The process model in Fig.4 is the key part of the simulation, which simulates the packets transmitting and receiving and implements RL and ARQ. The energy model in Fig.5 measures the energy consumption according to three processing states, i.e. transmitting and receiving and idle states.

The final goal of our work is to improve the throughput by increasing the RL learning rate. However, if the packets are continuously transmitting to the

receiver during the waiting time (J-ARQ), this will arithmetically result in an extra throughput, which cannot be attributed to the improvement of RL learning rate. As it is shown in Fig.6, the J-ARQ sends two more packets during the waiting time, theoretically, the J-ARQ would have a throughput three times higher than S&W-ARQ, while the blue line is more than three times of the red line. Therefore, we can conclude that J-ARQ provides a higher throughput, more than three times of the S&W-ARQ contributed by the improvement of RL learning rate.

Fig. 3. Node model of end-to-end nodes.

Fig. 4. Process model of process state transition.

Fig. 5. Process model of energy state transition.

The MCS index selection in Fig.7 has shown denser blue dots than red ones, because the ACK feedback more frequently for J-ARQ than the S&W-ARQ which means a quicker learning rate and better performance in tracking the channel dynamics.

Fig. 6. Throughput of WAL-case1-DES-1 and WAL-case1-DES-2, which represents J-ARQ and S&W-ARQ respectively.

Fig. 7. The MCS index selection during the simulation time. Blue dots denote the J-ARQ and the red dots are the Stop & Wait ARQ.

Fig. 8. Energy efficiency.

The energy consumption of J-ARQ is definitely higher than the S&W-ARQ but as for the energy efficiency, which is normalized by the total number of packets, a higher yield in energy consumption efficiency is achieved, as it is shown in Fig.8. This may be attributed to the MCS index design and better selection of the RL agents.

5 Conclusions

Reinforcement learning has been introduced to optimize the link adaptation in highly dynamic underwater scenario. When we combine J-ARQ to deal with the long delay feedback, it increases the learning efficiency of RL agent and materially improved the throughput. The simulation results show a significant better performance in throughput. Also, higher energy efficiency attributed to the new design of MCS index, which adds different levels of transmitting energy.

This work was supported in part by the National Natural Science Foundation of China under Grants 61571048 and 61431004, and in part by the Key Laboratory of Acoustic Environment, Institute of Acoustics, Chinese Academy of Sciences and Acoustic Science and Technology Laboratory, Harbin Engineering University.

References

1. J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, K. McCoy, *JANUS underwater communications standard* ('14 UComms, 1–4 2014)
2. R. Karmakar, S. Chattopadhyay, S. Chakraborty, *Dynamic link adaptation in IEEE 802.11ac : A distributed learning based approach* ('16 IEEE 41st Conf. LCN, 87–94 2016)
3. J. Zhang, X. Ma, G. Qiao, et al, *A full-duplex based protocol for underwater acoustic communication networks* ('13 OCEANS San Diego, 2013)
4. S. Shankar, M. Chitre, *Tuning an underwater communication link* ('13 MTS/ IEEE OCEANS Bergen, 1–9 2013)
5. S. Shankar, M. Chitre, M. Jayasuria, *Data driven algorithms to tune physical layer parameters of an underwater communication link* (OCEANS'10 IEEE Sydney, 1–5 2010)
6. J. P. Leite, P. H. P. De Carvalho, R. D. Vieira, *A flexible framework based on reinforcement learning for adaptive modulation and coding in OFDM wireless systems* ('12 IEEE WCNC, 809–814 2012)
7. S. Yun, C. Caramanis, *Reinforcement learning for link adaptation in MIMO-OFDM wireless systems*, ('10 IEEE GLOBECOM, 1–5 2010)
8. A. Rico-Alvarino, R. W. Heath, *Learning based link adaptation in multiuser MIMO-OFDM*, (EUSIPCO '13, 1–5 2013)
9. K. Zhou, *Robust cross-layer design with reinforcement learning for IEEE 802.11 and link adaptation* ('11 IEEE ICC, 1–5 2011)
10. Wang, C., Wang, Z., Sun, W., et al, *IEEE Access*, **6**, 2541-58 (2018)
11. V. Mnih, K. Kavukcuoglu, D. Silver, et al., *Nature*, **518**, 529 -533 (2015)
12. M. Chitre, W. Soh, *IEEE J. Ocean. Eng.*, **40**, 93–103 (2015)
13. D. Kim, B. C. Jung, H. Lee, et al, *IEEE Trans. Wireless Commun.*, **7**, 5195-5201 (2008)
14. M. Stojanovic, *Underwater acoustic communications: Design considerations on the physical layer*, ('08 5th Annu. Conf. IEEE/IFIP WONS, 1-10 2008)