Monte Carlo Tree Search and Delay-Aware Feedback Adaptation for Underwater Acoustic Link Tuning

Wu Shuangshuang
Department of Electrical and Computer Engineering, National University of Singapore

Mandar Chitre
Department of Electrical and Computer Engineering & ARL, Tropical Marine Science Institute, National University of Singapore

Prasad Anjangi
Subnero Pte. Ltd. Singapore

***Abstract*—The unique properties of Underwater Acoustic Communication (UAC) channels such as limited bandwidth, strong multipath propagation, and large delay spread over tens or even hundreds of milliseconds result in severe non-stationary fading statistics. When the channel statistics change, the performance of a modulation scheme designed for a specific channel model might deteriorate. This motivates the need for real-time link tuning. We use adaptive modulation with a high degree of freedom in modulation and coding schemes to optimize channel throughput in the time-varying UAC channel. The key idea involves dealing with the exploration versus exploitation dilemma, which is formulated as a Markov Decision Process, to maximize average data rate without any prior knowledge of the UAC channel. Reinforcement learning methods help estimate Channel State Information and schedule packet transmissions to achieve higher throughput. We present a hybrid algorithm that includes short-term planning in MDPs to select MCSs. This helps us reduce computational complexity while achieving a comparable performance to the algorithms that perform full-planning. We also study an online learning strategy to determine an appropriate number of transmission packets between every two feedback packets to reduce the time spent on obtaining feedback, as the long propagation delay characteristic makes frequent feedback impractical.**

I. Introduction

The ocean is a challenging acoustic communication medium, usually characterized by limited bandwidth, and large delay spread over tens or even hundreds of milliseconds which result in severe frequency-selective fading [1]. A modulation scheme optimally designed for specific channel conditions may perform poorly when the channel changes. Therefore, Adaptive Modulation (AM) techniques have the potential to offer higher performance by selecting Modulation and Coding Schemes (MCSs) that provide higher link throughput in time-varying channel conditions [2].

One of the key challenges in most of the AM techniques is in striking a balance between exploration versus exploitation in selecting MCSs. Such problems are usually formulated as Markov Decision Processes (MDP). In recent years, Reinforcement Learning (RL) has emerged as a technique for learning in an unknown dynamic environment with regular feedback information [3]–[7]. The AM problem formulated as an MDP can use Channel State Information (CSI) to select schemes that help in achieving higher throughput. However, the state of the underwater acoustic environment is typically not known initially, and is time-varying. This requires a large number of MCSs to be tried before starting to exploit the information gained through feedback. The authors in [8] show that traditional RL algorithms usually suffer from high computation complexity due to large or continuous state spaces [9] and frequent planning [10]. An alternative approach such as Monte Carlo Tree Search (MCTS) [11] provides a computationally attractive alternative. MCTS is a decision-making algorithm that enables search on large combinatorial spaces represented by trees using random sampling [12]. In [13], MCTS is used as the backbone of AlphaGo developed by Google DeepMind. Since then, MCTS has been increasingly applied in various planning and sequential decision making problems [10]. The state space that we observe in our problem is large and well suited for the application of MCTS.

Timely feedback from the channel is necessary for tracking CSI and is required for selecting MCSs [14]. The speed of sound in water is approximately 1500 m/s, resulting in propagation delays that are $200{,}000\times$ higher than those experienced in terrestrial radio communication networks [15]. In a traditional stop-and-wait style protocol, feedback incurs a waiting time equivalent to two-way propagation delay. It is therefore impractical to expect feedback after every transmission and still achieve a high average data rate. We propose strategies that can be utilized to determine the interval at which the feedback should be gathered to transfer data efficiently.

The rest of the paper is organized as follows. The problem formulation is elucidated in Section II. The proposed algorithm, called K-MCTS is described in Section III. In Section III, the performance of different policies are compared to help understand K-MCTS policy to select a better scheme. Different feedback strategies are presented in Section IV. In Section V, scheme selection policy determined is combined with different feedback strategies to demonstrate the advantages of the proposed technique. The abbreviations and commonly used symbols are listed in Table I and II.

TABLE I
LIST OF ABBREVIATIONS

UAC	Underwater Acoustic Communication
AM	Adaptive Modulation
MCSs	Modulation and Coding Schemes
RL	Reinforcement Learning
MCTS	Monte Carlo Tree Search
CSI	Channel State Information
MDP	Markov Decision Process

TABLE II
LIST OF COMMONLY USED SYMBOLS

l	Distance between transmitter and receiver
n	Number of schemes (MCSs)
τ	Transmission duration
τ_{pd}	Propagation delay
τ_{fd}	Feedback delay
τ'	Feedback packet duration
T	Total transmission time of the entire communication process
N	Number of bits to be transmitted
$x^i \in \mathcal{X}$	The i_{th} scheme in action space
d^i	Data rate of the i_{th} scheme
p_j^i	Estimated probability of packet success of the i_{th} scheme at state S_j
J	Number of transmission packets to transfer N bits of information
$S_j \in \mathcal{S}$	The j_{th} state in state space
D_j	Expected reward at state S_j
$D_j^{'}$	Amount of transmitted information
K	Step size in one K-level look-ahead planing process
k	Index used for packet step in a planning iteration
h	Number of transmitted packets between every 2 feedback packets

II. PROBLEM FORMULATION

We begin by focusing on a problem where a transmitter and a receiver are placed at a distance l in a static underwater environment. A total of n schemes (MCSs) in action space $\mathcal{X}$ = $\{x^i, i \in 1 \cdots n\}$ are available to transmit N bits of information between the transmitter and receiver. For the j^{th} transmission packet, scheme x^i associated with data rate d^i is selected to transmit packet within a fixed time duration τ and thus each packet might carry different number of bits. We consider finite-horizon MDPs (file transfer application) with state space and action space denoted by $\mathcal{S}$ and $\mathcal{X}$ respectively. We use $k = 0, 1, \cdots, K$ to denote packet-step inside a look-ahead planning iteration. Total N bits will be transmitted in J packets where J is unknown until N bits are all transmitted and $j = 0, 1, \cdots, J$ denotes the index of a state. The probability of packet success γ^i of each scheme x^i is unknown initially and can only be estimated based on the feedback information that is collected. Receiving feedback information from the receiver after every transmission turns out to be expensive due to two-way propagation delay and therefore we consider feedback packets to be received only when h packets have been transmitted.

State $S_j \in \mathcal{S}$ is arrived at when the j^{th} packet is transmitted. Each state $S_j = \{D_j', G_j\}$ is defined by two parameters D_j' and G_j. Here, D_j' is the total number of bits transmitted till state S_j, and $G_j = \{m_j^1, \theta_j^1, \cdots, m_j^n, \theta_j^n\}$ denotes a summary of the knowledge of the channel. m_j^i is the number of times scheme x^i has been tried and θ_j^i is the number of times scheme x^i has been successful, i.e., no bits were in error after forward error correction at the receiver. Now, at state S_j, the probability of success p_j^i of each scheme x^i is estimated as:

$$p_j^i = \frac{\theta_j^i}{m_j^i}. \tag{1}$$

When scheme x^i is selected at state S_j, the immediate reward r_j is equal to $d^i\tau$ bits if transmission is successful and 0 bits otherwise. Therefore, the expected reward D_j when scheme x^i is selected is given by:

$$D_j = p_j^i d^i \tau. \tag{2}$$

An *agent* makes decision on which scheme to select from action space $\mathcal{X}$ available at the current state S_j. The policy

Π is a function that maps from state space to action space $\Pi : \mathcal{S} \rightarrow \mathcal{X}$. Guided by different policy functions, a scheme $\Pi(S_j) = x^i$ is selected by the agent for the next h packets. After the h packets are transmitted, the receiver responds with an outcome v and the agent transitions to a new state S_{j+h}. The outcome v records the number of successful transmissions during those h transmitted packets. Therefore, the state transition function is represented as $\Gamma(S_j, x^i, h, v) : S_j \rightarrow S_{j+h}$. The updated parameters of the state $S_{j+h} = \{D'_{j+h}, G_{j+h}\}$ are now represented as:

$$G_{j+h} = \{m_j^1, \theta_j^1, \cdots, m_j^i + h, \theta_j^i + v, \cdots, m_j^n, \theta_j^n\}, \quad (3)$$

$$D'_{j+h} = D'_j + v d^i \tau. \quad (4)$$

Now, the expected reward D_{j+h} in the new state S_{j+h} is:

$$D_{j+h} = p_{j+h}^i d^i \tau. \quad (5)$$

We wish to maximize the average data rate over the entire communication sequence through continuous improvement. Exploitation of the gained knowledge through feedback from the receiver usually means selecting valuable schemes to get a maximal immediate reward while exploration is defined as trying other schemes in the action space which may bring a greater benefit at the cost of time. Therefore, the policy to select scheme x^i must balance between exploration and exploitation. As shown in (3), only when the transmitter obtains the outcome v, the agent can update the next state S_{j+h} and our estimate of p_j^i gets closer to γ^i.

The cost involved in gathering feedback comprises of the propagation delay τ_{pd} and the feedback duration τ_{fd} as illustrated in Fig. 1. Rather than following either an exploration or an exploitation strategy, the objective is to investigate policies which target maximizing the long-term average data rate W while transmitting N bits. Now, using a policy function would result in a sequence of actions such as $\mathbf{\Pi} = \{\Pi(S_0), \Pi(S_1), \dots\}$. Similarly an outcome sequence $\mathbf{V}$ is also generated. The outcome sequence $\mathbf{V}$ consists of 1 or 0 indicating either a packet success or a failure. The corresponding data rate sequence is denoted by $\mathbf{d}$. In transmitting N bits of information, it takes a total of J data packets and H feedback packets (both of which are unknown). Therefore, the objective function is formulated as minimizing the total time $T = J\tau + H(\tau_{fd} + \tau_{pd})$ in transmitting N bits of information:

$$\begin{aligned} &\min J\tau + H(\tau_{fd} + \tau_{pd}), \\ &s.t. \sum_{q=0}^{J} \mathbf{V}[q]\mathbf{d}[q]\tau = N. \end{aligned} \quad (6)$$

III. Comparison of scheme selection policies

We describe and compare a few well-known policies here:

1) Random: An action x^i is randomly selected from action space $\mathcal{X}$.

$$\Pi(S_j) = x^i, \text{with probability } \tfrac{1}{n}. \quad (7)$$

For this policy, if sufficient number of packets are transmitted, the average data rate W tends to be the mean of n scheme data rates:

$$W = \frac{\sum_{i=1}^{n} d^i}{n}. \quad (8)$$

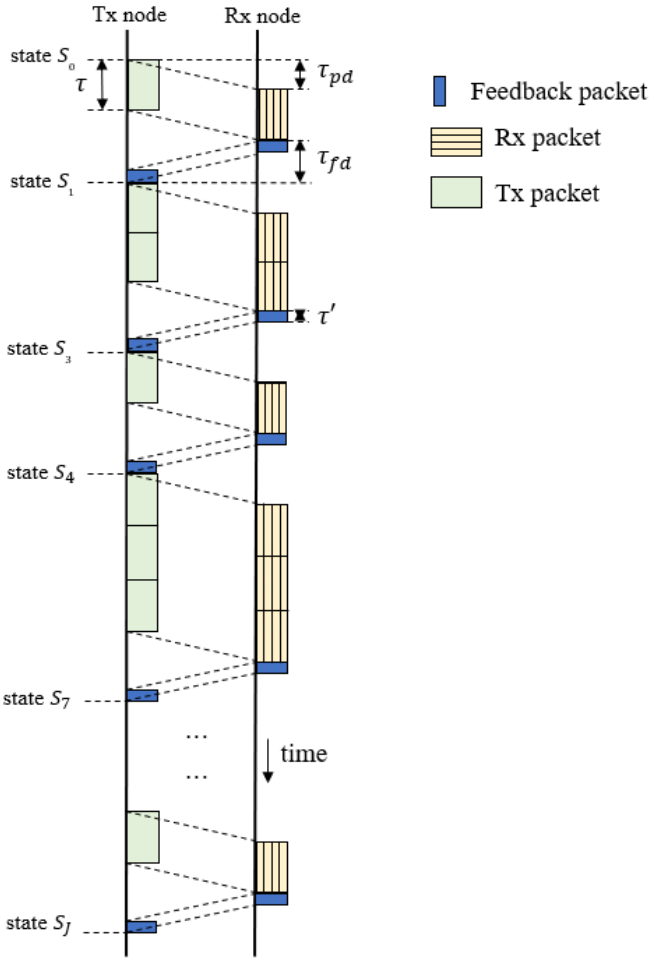

Fig. 1. An illustration of the delays involved in a typical packet exchange between transmitter and receiver nodes.

2) Greedy: A greedy approach always selects a scheme with maximal expected reward D_j, therefore

$$\Pi(S_j) = \underset{x^i \in \mathcal{X}}{\operatorname{argmax}} D_j. \quad (9)$$

With this approach, when a good scheme fails, agent is easy to be deceived and it is possible that sub-optimal schemes are recommended.

3) ϵ-Greedy: ϵ-Greedy policy is more proficient in dealing with the exploration-exploitation dilemma via:

- exploring schemes randomly to avoid missing better choice with probability ϵ;
- adopting greedy policy to help the agent select a scheme with the maximal immediate estimated reward with probability $1 - \epsilon$.

$$\Pi(S_j) = \begin{cases} \underset{x^i \in \mathcal{X}}{\operatorname{argmax}} D_j, & \text{with probability 1-}\epsilon \\ \text{Random}, & \text{with probability } \epsilon \end{cases}. \quad (10)$$

4) Upper Confidence Bound: Upper Confidence Bound (UCB) is the most widely used solution for exploration-exploitation dilemma in MDPs. The series of transmission successes and failures is formulated as a Bernoulli process. UCB is a family of algorithms and the Wilson score interval developed by Edwin Bidwell Wilson [16] has the asymmetric analytical representation which avoids the *overshoot* and *zero-width interval* problems. Therefore, Wilson score interval can be safely employed with small samples and skewed observation in our initial transmission phase [17]:

$$\Pi(S_j) = \underset{x^i \in \mathcal{X}}{\operatorname{argmax}} \left(\frac{\theta_j^i + \frac{1}{2}z^2}{m_j^i + z^2} + \frac{z}{m_j^i + z^2} \sqrt{\frac{\theta_j^i(1-\theta_j^i)}{m_j^i} + \frac{z^2}{4}} \right) d^i \tau, \tag{11}$$

where $z = 1.96$ for 95% confidence. The second term inside the bracket is for confidence or used as a measure of the knowledge of every schemes, i.e. for each scheme, the less we understand, the greater the second term. Therefore, this policy selects schemes that have been tried less and continually tends to select schemes with higher estimated reward. Therefore, UCB policy balances the exploration and exploitation and eventually leads to the optimal scheme.

5) K-MCTS: Monte Carlo Tree Search (MCTS) is a powerful approach to designing game-playing bots or solving sequential decision problems. Based on the rollout-based Monte-Carlo planning algorithms [12], we propose a new K-MCTS algorithm which builds its K-level look-ahead tree by repeatedly sampling a sequence with length K of state-action-cost triplets from the current state by Bellman equation (13). In such trees, every node denotes one state and at one state, a pair of edges represent successful and failed outcomes with an action selected. Our generic scheme K-MCTS is shown in Fig. 2. An action at each state is selected using (12) to minimize the cost function:

$$\Pi(S_j) = \underset{x^i \in \mathcal{X}}{\operatorname{argmin}} C(S_j), \tag{12}$$

where $C(S_j)$ is the cost involved in a state to adjust the estimated remaining transmission time. During the K-level look-ahead tree, the cost of S_j to S_{j+K-1} is calculated using (13), i.e.,

$$\begin{aligned} C(S_j) &= \min_{x^i \in \hat{\mathcal{X}}} \tau + C(S_{j+1}) \\ &= \min_{x^i \in \hat{\mathcal{X}}} \tau + p_j^i C(\Gamma(S_j, x^i, h=1, v=1)) \\ &\quad + (1-p_j^i) C(\Gamma(S_j, x^i, h=1, v=0)), \end{aligned} \tag{13}$$

but in S_{j+K}, unless the terminal state has arrived the cost of which is 0, the cost is approximated by the average remaining transmitted time with the selected schemes, i.e.,

$$C(S_{j+K}) = \frac{N - D'_{j+K}}{p_j^i d^i}. \tag{14}$$

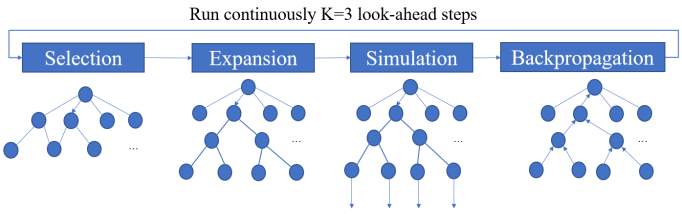

Fig. 2. Monte Carlo Tree Search phases.

The success probability is estimated by UCB method to avoid *zero* estimation if the selected $\theta_j^i = 0$ using (1) when evaluating the cost using (13) and (14):

$$p_j^i = \frac{\theta_j^i + \frac{1}{2}z^2}{m_j^i + z^2} + \frac{z}{m_j^i + z^2} \sqrt{\frac{\theta_j^i(1-\theta_j^i)}{m_j^i} + \frac{z^2}{4}}. \tag{15}$$

For each K-level look-ahead tree, there are four phases shown in Fig. 2:

- *Selection* - A scheme is selected according to (12) at each depth. This phase terminates when a terminal state of the problem has been reached.
- *Expansion* - Before reaching a terminal state, *expansion* determines all possible child nodes (states) in K levels. When *expansion* comes to the terminal state (all N has been transmitted), it skips to *backpropagation* phase.
- *Simulation* - Compared with traditional MCTS, our K-MCTS follows an approximate way to implement iterative deepening within K look-ahead levels using (13) and (14). However, with a larger action space n, the computational complexity increases since all possible actions are considered at each state. Consequently, to narrow down the searching space and decide which ones to throw away, we used the 70^{th} percentile in our simulation, which means that we only sample the 30% with higher results in (15) than 70% of the others for $k = 0, \cdots, K$. Then, a smaller set of actions $\hat{\mathcal{X}}$ is used for expansion.
- *Backpropagation* - Propagate the costs back to all states along the path.

Following these 4 phases, x^i is selected and employed for transmission of h packets. With feedback, the agent transfers to a new state and backpropagates costs to the initial state. Since K-MCTS algorithm adjusts between exploration and exploitation by comparing expected rewards of each scheme, its strategy favors exploration initially and gradually switches to a pure exploitation mode. Algorithm 1 presents a step by step procedure to determine an optimal scheme using K-MCTS policy.

IV. Feedback Strategies Comparison

The frequency and reliability of feedback information are crucial to perform adaptive modulation efficiently in UAC systems. Due to the large propagation and feedback delay, it is unrealistic to obtain feedback information for every packet that is transmitted. Therefore, an adaptive delay-aware feedback

Algorithm 1 K-MCTS algorithm

Input: n, N, K, $k = 0$, $j = 0$.
Output: Action $\Pi(S_j)$, for $j = 0, 1, \cdots, J$.

function search(S_j, k)
  if S_j is the terminal state **then**
    cost(S_j) = 0.
  end if
  if $k = K$ **then**
    cost(S_j) according to (14).
  else
    cost(S_j) = τ + search(S_{j+1}, k+1).
  end if
  return cost(S_j).

while N bits not finished, **do**
  $C(S_j) = search(S_j, K)$.
  Action $\Pi(S_j) = \underset{x^i \in \hat{\mathcal{X}}}{\operatorname{argmin}} C(S_j)$.
  Calculate h.
  Transmit h packets by $\Pi(S_j)$,
  Feedback is obtained and new state S_{j+h} is arrived;
end while

strategy to determine the appropriate number of transmission packets h to be transmitted until the next feedback is necessary. Intuitively, when the channel information is insufficient, providing feedback actively from the receiver is necessary. As the agent gathers more channel information, the feedback interval can be reduced. Therefore different feedback strategies can be employed and are studied as follows.

1) Fixed feedback strategy: A naïve strategy is to have a fixed number of transmission packets between every 2 feedback packets. The transmitter will receive the feedback packet after h packets have been transmitted.

2) Packet-varying feedback strategy: With increasing number of packets transmitted, the agent gathers channel information. Active feedback helps the agent learn the UAC environment quickly in the initial phase and the agent gradually reduces its dependence on the feedback to make decisions in the later phases. Therefore h is approximately given by:

$$h = \lceil \beta j \rceil, \tag{16}$$

where β is to determine the change rate of h versus transmitted packets.

3) Target-oriented feedback strategy: A change in channel conditions can render previously learned knowledge invalid, and consequently the packet-varying feedback strategy might become a poor choice to adopt. We need a more adaptive feedback strategy to adjust h according to the varying channel conditions. If we have an estimate for an achievable data rate w_a in the channel (we call this the *target data rate*), we can calculate the ratio r_w between the immediate data rate w_c of the transmitted h packets and w_a, and use it to adapt the value of h. Although we typically do not know w_a, we can estimate it from our knowledge of the channel:

$$w_a = \max p_j^i d^i, \tag{17}$$

and

$$r_w = \frac{w_c}{w_a}. \tag{18}$$

The value of h can then be adapted using a sigmoidal transformation:

$$h = \left\lceil \frac{h_m}{1 + e^{-f(r_w)}} \right\rceil, \tag{19}$$

in which $f(\cdot)$ is chosen to ensure h stays bounded in the range $[1, h_m]$. The value of h_m is updated according to h' (the value of h from the previous state), and $\triangle r_w$ (the difference between r_w calculated at the previous state $S_{j-h'}$ and the current state S_j):

$$h_m = h'(1 + \alpha(\triangle r_w)), \tag{20}$$

where $\alpha(\triangle r_w)$ is:

$$\alpha(\triangle r_w) = \begin{cases} (\lg \frac{N}{n})^{\triangle r_w} & \triangle r_w > 0 \\ \frac{\triangle r_w}{\lg \frac{N}{n}} & \triangle r_w \leq 0. \end{cases} \tag{21}$$

Equations (19)-(21) ensure that h follows the change of r_w closely: h will be larger (or close to the maximal value h_m) when r_w increases slightly (or dramatically). If r_w becomes smaller, indicating that our selected scheme is not the most appropriate and we need to reconsider our policy, then a smaller h (or even $h = 1$) is selected to help track the channel rapidly.

V. Simulation results

A. Discussion on various look-ahead levels K

In order to select an appropriate look-ahead level K in K-MCTS, we try $K = 0, 1, 2, 3$. The simulation is setup with the following parameters:

1) The transmitter and receiver are placed very close, i.e., $l = 0$.
2) Feedback delay duration $\tau_{fd} = 0$ and the propagation delay $\tau_{pd} = 0$.
3) The fixed feedback strategy is employed in this section with $h = 1$.
4) When n is set to $2, 5, 10$, the examples of simulation parameters are generated and are tabulated in Table III. The data rate $d^i \in [300, 1500]$ bps is randomly generated and the probability of packet success is generated by a Beta distribution $\gamma^i \sim Be(2, 4)$ but is unknown to the agent. $\hat{w_u}$ is the maximal effective data rate, given by $\hat{w_u} = \max \gamma^i d^i$ and $\hat{w_l}$ is the minimal effective data rate, given by $\hat{w_l} = \min \gamma^i d^i$.
5) We run 1000 simulations for every n and each is associated with different d^i and γ^i. Similarly, $n = 100$ has also been simulated to test 70^{th} percentile method.
6) Although the transmission duration τ in practical modems might vary, we assume $\tau = 1s$ and hence the length of packets $d^i\tau$ are possible to be different depending on d^i with selected x^i.

7) The stopping criteria for all policies is when $N = 50000$ bits are successfully transmitted.
8) ϵ is set to be 10% in the ϵ-Greedy policy.

TABLE III
SIMULATION PARAMETERS

Simulations	Scheme x^i	γ^i	d^i/(bps)	$\hat{w}_u$/(bps)	$\hat{w}_l$/(bps)
Simulation 1	x^1	0.12	1204	526.08	144.48
	x^2	0.59	896		
Simulation 2	x^1	0.59	826	768.3	167.2
	x^2	0.28	861		
	x^3	0.61	1090		
	x^4	0.13	1270		
	x^5	0.53	1452		
Simulation 3	x^1	0.87	1371	1189	149
	x^2	0.53	697		
	x^3	0.30	491		
	x^4	0.35	1020		
	x^5	0.30	1391		
	x^6	0.11	1340		
	x^7	0.49	664		
	x^8	0.32	1461		
	x^9	0.46	1141		
	x^{10}	0.41	586		

Simulation results with $n = 2, 5, 10$ are shown in Fig. 3, Fig. 4, and Fig. 5. The result of *Random* policy is about 50% between the maximal and minimal effective data rate as expected. UCB and K-MCTS wisely exploit by taking advantage of prior knowledge and explore to try new schemes and hence their advantages are obvious. With the increase of look-ahead levels K, K-MCTS is more prominent. Especially, when look-ahead level $K = 0$ which means no exploration, the action is selected by UCB policy. A similar advantage of K-MCTS is also observed in Fig. 6 when n is set to 100.

B. Different feedback strategies comparison

K-MCTS policy outperforms the other strategies and therefore we select this policy for studying the different feedback strategies. For this simulation, the distance between transmitter and receiver is $l = 1$ km. As the sound speed in underwater environment is around 1500 m/s, the propagation delay $\tau_{pd} = 0.67$ s. The duration of one feedback packet $\tau' = 1$ s and the feedback delay $\tau_{fd} = \tau_{pd} + \tau'$.

For packet-varying feedback strategy, β is set to 0.1. For target-oriented feedback strategy, $f(r_w) = 12(r_w - 0.5)$ in (19) is helpful to realize $h \in [0, h_m]$ when $r_w \in [0, 1]$. Schemes are generated in Table IV and results are shown in Fig. 7.

We can see the target-oriented feedback strategy outperforms other strategies when the feedback delay duration $\tau_{fd} > 0$.

VI. CONCLUSION

We developed a hybrid algorithm to simultaneously optimize the link adaptation and minimize the feedback delay

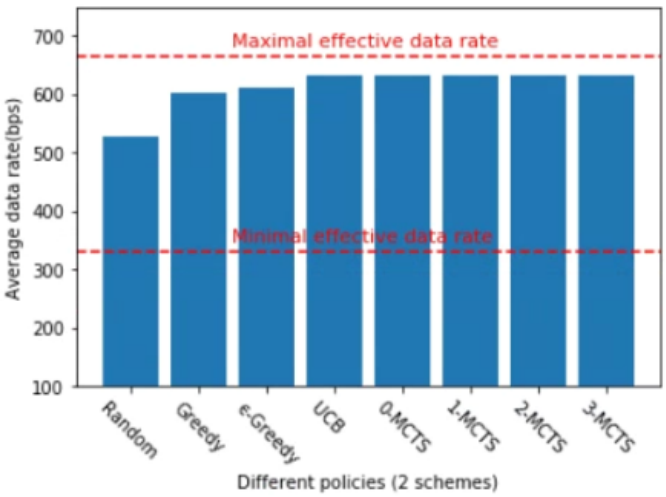

Fig. 3. Policy comparison when n = 2.

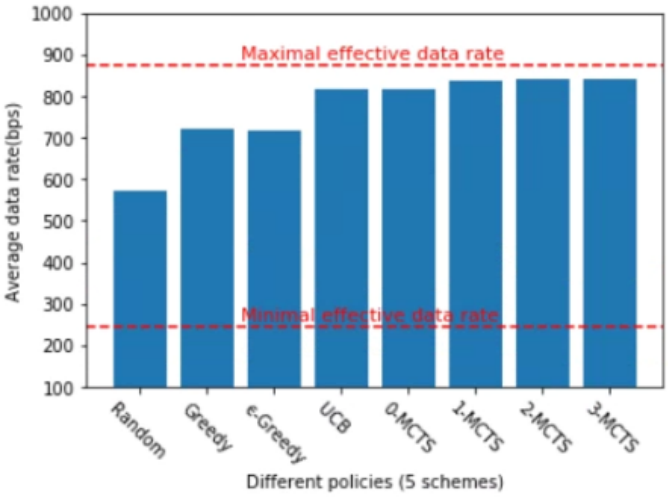

Fig. 4. Policy comparison when n = 5.

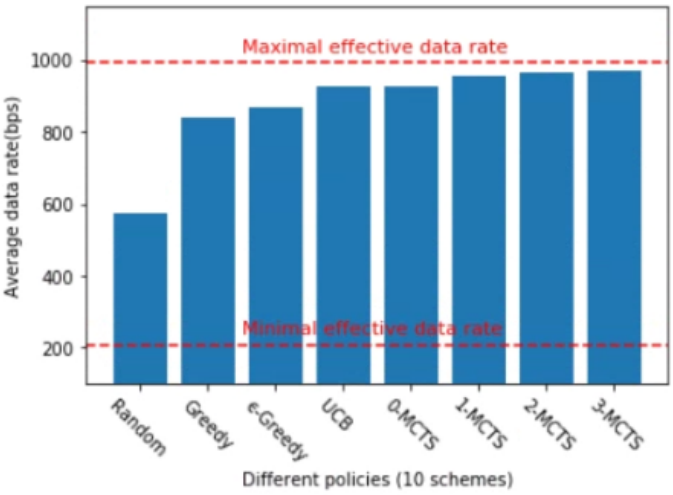

Fig. 5. Policy comparison when n = 10.

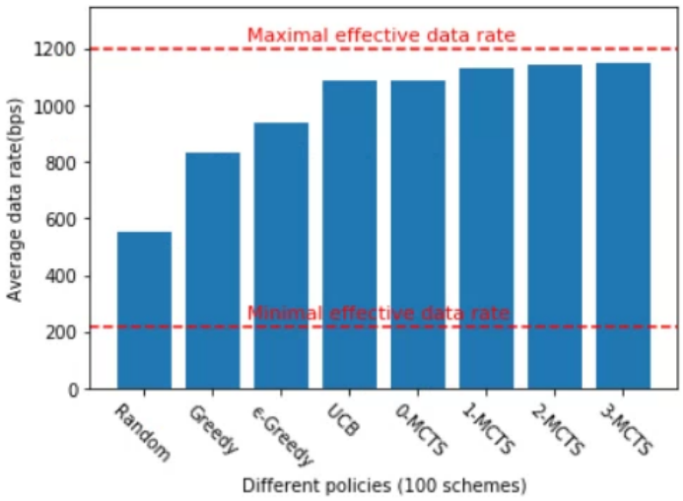

Fig. 6. Policy comparison when n = 100.

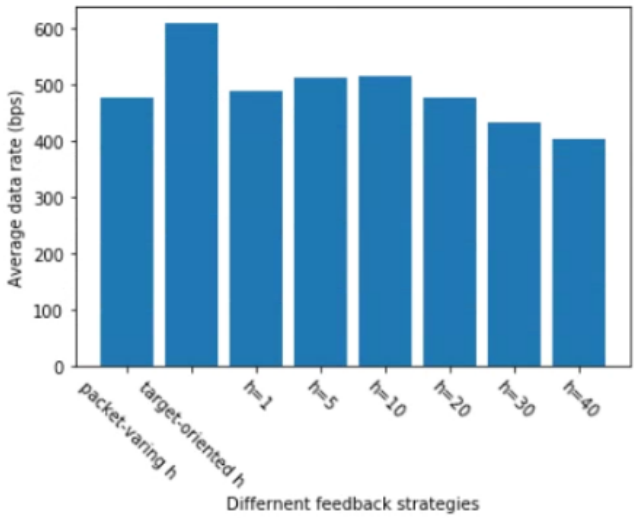

Fig. 7. Comparison of different feedback strategies.

TABLE IV
FEEDBACK STRATEGY SIMULATION PARAMETERS

scheme x^i	γ^i	d^i/(bps)	$\hat{w}_u$/(bps)	$\hat{w}_l$/(bps)
x^1	0.26	1158	629.2	32.9
x^2	0.38	984		
x^3	0.06	483		
x^4	0.09	995		
x^5	0.12	441		
x^6	0.44	1253		
x^7	0.28	602		
x^8	0.54	1149		
x^9	0.65	351		
x^{10}	0.68	348		

cost. Compared with various well-known data-driven algorithms for link tuning, simulation results show that K-MCTS outperforms in all circumstances and is also performant when action space is larger. Used along with with K-MCTS, our delay-aware feedback strategy makes intelligent decisions by minimizing the average transmission delay while taking into account the issue of exploration and exploitation dilemma. The next steps are to implement these algorithms on practical devices (e.g., underwater acoustic modems) where the parameters may be tuned, and test the algorithm in the ocean.

REFERENCES

[1] M. Stojanovic, "Underwater acoustic communications: Design considerations on the physical layer," in *2008 Fifth Annual Conference on Wireless on Demand Network Systems and Services*, 2008, pp. 1–10.

[2] L. Huang, Q. Zhang, L. Zhang, J. Shi, and L. Zhangb, "Efficiency enhancement for underwater adaptive modulation and coding systems: via sparse principal component analysis," *IEEE Communications Letters*, pp. 1–1, 2020.

[3] Q. Fu and A. Song, "Adaptive modulation for underwater acoustic communications based on reinforcement learning," in *OCEANS 2018 MTS/IEEE Charleston*, 2018, pp. 1–8.

[4] C. Wang, Z. Wang, W. Sun, and D. R. Fuhrmann, "Reinforcement learning-based adaptive transmission in time-varying underwater acoustic channels," *IEEE Access*, vol. 6, pp. 2541–2558, 2018.

[5] W. Su, J. Lin, K. Chen, L. Xiao, and C. En, "Reinforcement learning-based adaptive modulation and coding for efficient underwater communications," *IEEE Access*, vol. 7, pp. 67 539–67 550, 2019.

[6] D. Lee, Y. G. Sun, S. H. Kim, I. Sim, Y. M. Hwang, Y. Shin, D. I. Kim, and J. Y. Kim, "Dqn-based adaptive modulation scheme over wireless communication channels," *IEEE Communications Letters*, pp. 1–1, 2020.

[7] J. P. Leite, P. H. P. de Carvalho, and R. D. Vieira, "A flexible framework based on reinforcement learning for adaptive modulation and coding in ofdm wireless systems," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 809–814.

[8] Y. Efroni, N. Merlis, M. Ghavamzadeh, and S. Mannor, "Tight regret bounds for model-based reinforcement learning with greedy policies," 2019.

[9] X. Xu, C. Liu, S. X. Yang, and D. Hu, "Hierarchical approximate policy iteration with binary-tree state space decomposition," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1863–1877, 2011.

[10] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, vol. 11, no. 51, pp. 1563–1600, 2010. [Online]. Available: http://jmlr.org/papers/v11/jaksch10a.html

[11] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[12] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *Machine Learning: ECML 2006*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 282–293.

[13] D. Silver, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, 2016.

[14] M. R. Khan, B. Das, and B. B. Pati, "Channel estimation strategies for underwater acoustic (UWA) communication: An overview," *Journal of the Franklin Institute*, vol. 357, no. 11, pp. 7229–7265, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003220302325

[15] J. P. Alan C. Farrell, "Performance of IEEE 802.11 mac in underwater wireless channels," *Procedia Computer Science*, vol. 10, no. 12, pp. 62–69, 2012. [Online]. Available: https://doi.org/10.1016/j.procs.2012.06.012

[16] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[17] S. Wallis, "Binomial confidence intervals and contingency tests: Mathematical fundamentals and the evaluation of alternative methods," *Journal of Quantitative Linguistics*, vol. 20, no. 3, pp. 178–208, 2013. [Online]. Available: https://doi.org/10.1080/09296174.2013.799918