Model-based Data-driven Learning Algorithm for Tuning an Underwater Acoustic Link

Prasad Anjangi
Subnero Pte. Ltd.
Singapore
Email: prasad@subnero.com

Mandar Chitre
Acoustic Research Laboratory, Tropical Marine Science Institute
National University of Singapore
Email: mandar@nus.edu.sg

***Abstract*—Underwater acoustic channels are fast varying in both spatial and temporal domain and hence are characterized by non-stationary fading statistics. When the channel statistics change, a modulation scheme designed for a specific fading model will underperform which motivates the need for link tuning algorithms. In order to alleviate this problem, data-driven adaptive modulation techniques are studied previously. Since channel information is unknown, these algorithms solve the explore-exploit dilemma in order to take actions that result in maximizing the average data rate. Channel physics information is often ignored in the design of these algorithms. The information gained through channel physics such as delay spread, coherence time, doppler spread etc. of the channel plays an important role in narrowing down the search space of modulation scheme parameters. However, the channel physics by itself is not sufficient to find a good performing solution. Therefore, we develop a hybrid algorithm which utilizes both, the information gained from channel physics and techniques from data-driven algorithms to solve the explore-exploit dilemma. A simplified Orthogonal Frequency Division Multiplexing (OFDM) system is used to illustrate the concept and its parameters are tuned in an online fashion. In particular, an online learning algorithm is developed to track the goodness of the schemes and a multi-armed bandit like problem is solved for taking decisions sequentially in order to maximize the average data rate of an underwater acoustic (UWA) communication link.**

I. Introduction

A key technique for data transfer in a wide range of underwater applications is acoustic communications [1]. Tuning the transmission parameters according to varying channel conditions in order to optimize the communication performance is vital in underwater acoustic (UWA) channel [2]–[5]. Moreover, advancements in computer architecture have resulted in the design of underwater acoustic modems which are increasingly software-driven [6]–[10]. The software-defined modems provide the flexibility, a wide variety of algorithms and the ability to tune the modulation specific parameters. We use the term *link tuning* instead of *adaptive modulation* throughout this paper. We consider the concept of a link to have the ability to use any modulation scheme supported by an underwater acoustic modem and its corresponding parameters are to be tuned in conjunction.

Strategies to tune the parameters of the modulation scheme are developed in [3], [4], [5]. In [3], the authors propose an effective signal-to-noise ratio (ESNR) metric and show that it performs better than other previously studied metrics in estimating the channel. The authors in [4] develop a decision-tree based algorithm to chose modulation schemes for different channel conditions. The algorithm is tested on a dataset recorded in a particular location. The work presented in [5] also develops strategies for tuning parameters of the modem but the algorithms developed are data-driven. The information gained using channel physics such as the delay spread, the doppler spread, the ambient noise level over the desirable bandwidth etc., can play an important role in filtering the schemes which are viable from all the possible schemes that are realizable. In this paper, we present a hybrid algorithm which uses a model to differentiate between viable/good and bad regions in the tunable parameter space and at the same time utilizes the statistical information gained through packet transmissions to sequentially chose better decisions resulting in maximizing the average data rate.

The rest of the paper is organized as follows. A simple example to illustrate the data-driven algorithm is presented in Section II. An online learning algorithm to tune the parameters of the model and differentiate between viable and bad regions in the tunable parameter space is developed in Section III. The techniques presented in Sections II and III are combined and a hybrid algorithm is presented in Section IV. Discussions and conclusions are presented in Section V.

II. Data-driven algorithms: Illustration with simple example

In order to better understand the data-driven algorithms and their application in selecting better schemes, we start with a toy problem. Consider two schemes s_1 and s_2 for an underwater acoustic modem to use. Each time a packet is to be transmitted, one of these two schemes is selected. Each of these schemes s_1 and s_2 is associated with a known data rate γ_1 and γ_2 respectively. Upon the transmission of a packet, the underwater channel induces errors. We denote the probability of packet success for these two schemes by p_{s_1} and p_{s_2} respectively which are unknown.

Agent is the decision maker or the entity which selects one or the other scheme. The agent takes decisions based on the *state* it is in. The *state* of an agent is defined as $\mathrm{S} := \{m_1, p_1, m_2, p_2\}$ where, m_1 and m_2 denote the number of times scheme s_1 and s_2 are tried. p_1 and p_2 denote the number of times the packet was successfully transmitted and received, i.e., no bits were in error at the receiver. A policy Π is a function that maps from state space to action space $\Pi : \mathcal{S} \rightarrow \mathcal{X}$, where $\mathcal{S}$ and $\mathcal{X}$ denote the state space and action space. At each time step, the *reward* is a simple number, $R_t \in \mathrm{R}$. The immediate reward as a result of taking an action $\Pi(\mathrm{S})$ in state S is $R(\mathrm{S}, \Pi(\mathrm{S}))$. If an action s_i is selected, the immediate reward $R(\mathrm{S}, s_i)$ is γ_i, if the packet is successful,

978-1-5386-6442-1/18/$31.00

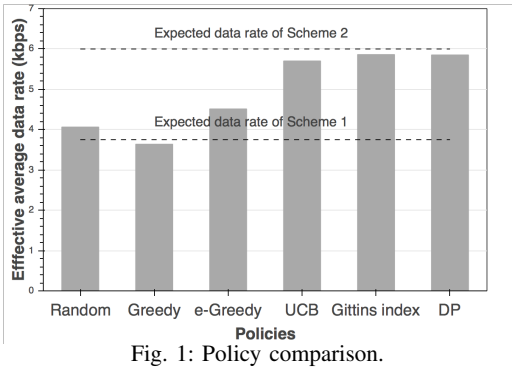

Fig. 1: Policy comparison.

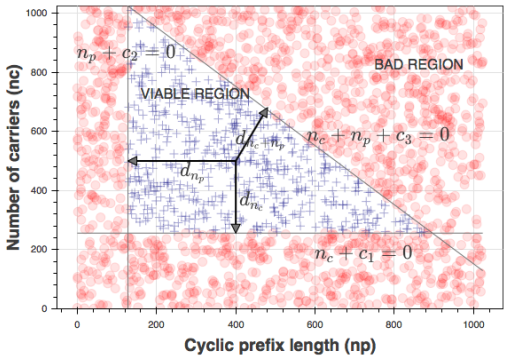

Fig. 2: The boundaries distinguishing the good and bad regions in the $\langle n_c, n_p \rangle$ plane.

and 0 otherwise. Therefore, the total expected reward as a result of selecting scheme i is:

$$\mathrm{E}[R(\mathrm{S}, s_i)] = \gamma_i p_{s_i}. \tag{1}$$

For the setup elucidated, we ask the following question: *What is the policy to determine which scheme to use for each transmission in order to maximize the average data rate in the long run?* This problem lies in the ambit of multi-armed bandit problems and is well-studied [11]. In order to understand these algorithms, we present a comparison of few well-known policies.

A. Comparison of policies

1) Random: The action taken in this policy is random, i.e., it is independent of the state the agent is in. The agent picks a random action uniformly among schemes s_1 and s_2:

$$\Pi(S) = \begin{cases} s_1, & \text{with probability 0.5} \\ s_2, & \text{with probability 0.5.} \end{cases} \tag{2}$$

2) Greedy: The action taken in this policy is based on a greedy approach. A scheme is chosen with the maximum current value of the reward:

$$\Pi(S) = s_{\arg\max_i \gamma_i \frac{p_i}{m_i}}. \tag{3}$$

3) ϵ-Greedy: This policy exploits by selecting schemes based on the maximum value of the reward and explores other schemes randomly with a small probability ϵ:

$$\Pi(S) = \begin{cases} s_{\arg\max_i \gamma_i \frac{p_i}{m_i}}, & \text{with probability } 1-\epsilon \\ s_1 \text{ or } s_2, & \text{with probability } \epsilon. \end{cases} \tag{4}$$

4) Upper confidence bound: A popular strategy to solve the explore-exploit dilemma in multi-armed bandit problems is the upper confidence bound (UCB) algorithm. The problem in consideration is a Bernoulli process with series of packet successes and failures. A binomial proportion confidence interval is an interval estimate of a success probability μ when only the number of experiments m_i and the number of successes p_i are known. The Argesti-Coull interval [12] has the simplest analytical representation and is used for this policy:

$$\Pi(S) = s_{\arg\max_i \frac{p_i}{m_i} + \frac{z}{m_i} \cdot \sqrt{\frac{p_i(1-p_i)}{m_i}}} \tag{5}$$

where $z = 1.96$ for 95% confidence. This policy selects the schemes which have higher potential of being a good scheme rather than being greedy and hence explores the schemes which have been tried less often to balance the exploration vs. exploitation.

5) Gittins index: Gittins and Jones [13] and Gittins [14] characterized the optimal policy for multi-armed bandit problem. The arm with the highest Gittins index is chosen at each time. At each step, the Gittins index is computed using the current state of the schemes. The formula used to compute Gittins index is selected from [15].

6) Dynamic programming: The optimal way to solve multi-armed bandit problem is to set it up as a Markov decision process (MDP) and use Markov decision theory to solve it. However such an approach does not scale well with number of schemes because of the curse of dimensionality. The optimal solution is given by solving the Bellman equation which is

$$\Pi^*(S) = \arg\max_{\Pi} R(S, \Pi(S)) + \lambda V(S') \tag{6}$$

where S' is the state which the agent is in after taking the action $\Pi(S)$ and λ is the discounting factor to make sure the return is bounded in an infinite horizon problem. The function $V(S)$ is the value function and represents the goodness of the state. An approximate value function is computed for the purpose of this simulation by running 10-level look ahead Monte-Carlo runs.

For the simulation study we set the unknown probability of packet success as $p_{s_1} = 0.3, p_{s_2} = 0.15$. The known data rate of the schemes are set to $\gamma_1 = 20$ kbps and $\gamma_2 = 25$ kbps. Note that the expected effective data rates of s_1 and s_2 are $20 \times 0.3 = 6$ kbps and $25 \times 0.15 = 3.75$ kbps. The discounting factor $\lambda = 0.999$ is used. These values are marked in Fig. 1 by two horizontal lines. The Monte-Carlo simulation with 10000 runs is carried out for each of these policies and the average effective data rate is shown in Fig. 1. In conclusion, Gittins index and the dynamic programming (DP) policy works best. The UCB, Gittins and DP policies do better than the random, greedy and ϵ-greedy policies. Considering the combination of factors such as computational complexity, scalability and the performance, the UCB policy seems to be the best choice for a problem with larger number of schemes which is presented in the next section.

III. Model-based tuning algorithm

In this section, we present a model (also used in [16, Section 7.2.3]) that is trained using an online algorithm to detect good and bad regions in the space of all possible schemes. Given that there is no knowledge of the channel beforehand, a model-based online learning algorithm is presented to tune the parameters of the model each time a scheme is used

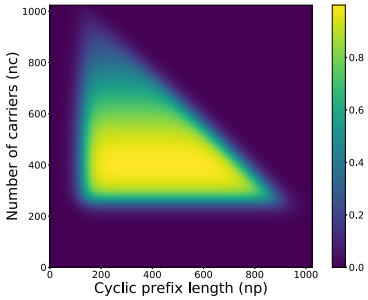

Fig. 3: Visualization of the model of probability of success over the space of possible schemes. The viable region and the behaviour inside the viable region is shown by the colormap.

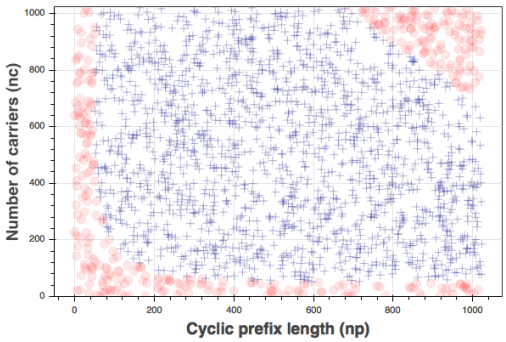

Fig. 4: Visualization of the good and bad regions when the boundaries are initialized.

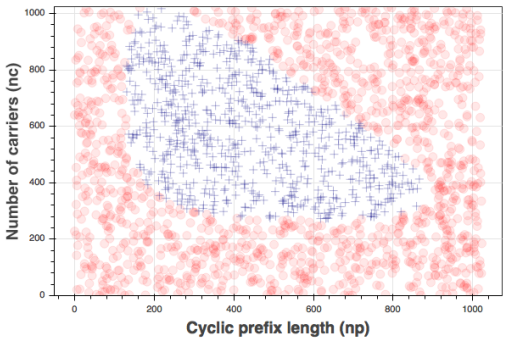

Fig. 5: Visualization of the good and bad regions when the boundaries have converged after learning from the real experiences.

for the packet transmission. To illustrate this idea, let us consider a simplified scheme in an OFDM system. The two key parameters in the OFDM technique are the number of sub-carriers n_c and the cyclic prefix length n_p. We define a scheme as a point on the $\langle n_c, n_p \rangle$ plane, i.e., for each packet transmission, the values of n_c, n_p must be selected such that the effective average data rate is maximized. Let B be the bandwidth occupied by the OFDM signal. In order to obtain a good performance, we know that the cyclic prefix duration T_p should be longer than the delay spread τ_{ds} of the channel. It is also necessary that the channel does not change significantly during a symbol duration. Therefore, the symbol duration T_s must be less than the channel coherence time τ_c, i.e.,

$$T_p > \tau_{ds} \implies n_p > B\tau_{ds}, \tag{7}$$

$$T_s < \tau_c \implies n_c + n_p < B\tau_c. \tag{8}$$

In addition to the above requirements, the bandwidth of each sub-carrier must be less than the coherence bandwidth of the channel for flat fading on each sub-carrier, i.e.,

$$n_c > \frac{B\tau_{ds}}{0.423}. \tag{9}$$

The relationship between T_s, T_p, B, n_c and n_p can be found in [16, Section 7.2.3]. The linear inequalities (7), (8) and (9) represent the boundaries in the $\langle n_c, n_p \rangle$ plane as shown in Fig. 2. Note that these requirements are necessary for good performance of OFDM system. In Fig. 2, the viable region is represented by blue or crossed points whereas the bad region is represented by red or circled points. Any scheme outside the viable region will perform poorly with a very high probability. Whereas a scheme within the viable region performs better. In order to mathematically represent the probability of success as visualized in the $\langle n_c, n_p \rangle$ plane, we utilize the behaviour of a sigmoid function. Consider a point (n_c, n_p) as shown in Fig. 2. The distances to the three linear boundaries are represented by d_{n_c}, d_{n_p} and $d_{n_c+n_p}$. Depending on which side of the boundary the point (n_c, n_p) lies, the sigmoid function returns either 0 or 1. A parametrised model for probability of success can be defined as following:

$$p^{\text{success}}_{c_1,c_2,c_3}(n_c, n_p) = s(d_{n_c})s(d_{n_p})s(-d_{n_c+n_p}) \tag{10}$$

where the values of d_{n_c}, d_{n_p} and $d_{n_c+n_p}$ can be computed based on the parameters c_1, c_2 and c_3 which define the position of the boundaries in the $\langle n_c, n_p \rangle$ plane (see Fig. 2 for linear equations representing boundaries). $s(d) = \frac{1}{1+e^{-\beta d}}$ is the sigmoid function and β represents the slope of the sigmoid function. In order to simulate the real world experience (i.e., packet transmission in underwater channel and observing success or failure), we consider a function which varies with n_c, outputs a maximum value at $n_c = h$, reduces with increasing n_c due to doppler sensitivity and reduces with decreasing n_c due to frequency-selective fading:

$$g(n_c) = e^{\frac{-(n_c-h)^2}{\sigma^2}} \tag{11}$$

where σ is a parameter of the function. The simulated probability of success $p^{\text{success}}_{\text{true}}(n_c, n_p)$ therefore is given by:

$$p^{\text{success}}_{\text{true}}(n_c, n_p) = \begin{cases} g(n_c), & \text{if (7), (8) and (9) are satisfied} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The product of (10) and (12) represents the model for a given set of parameters c_1, c_2 and c_3. Fig. 3 visualizes this model for the chosen parameters c_1, c_2 and c_3 on the $\langle n_c, n_p \rangle$ plane as a colormap. It clearly distinguishes the viable region from the bad region and represents the behavior of the probability of packet success within the viable region. Note that the simulated behavior $g(n_c)$ does not cause a change in the design of our algorithm. A more complex model of the channel can be used with the proposed algorithm 1 by replacing $g(n_c)$.

A. Learning algorithm for tuning parameters

The parameters c_1, c_2 and c_3 can be initialized to the values computed using (7), (8) and (9) if we know the measured values of delay spread τ_{ds} and channel coherence time τ_c. In cases, where these values cannot be measured, they can be initialized to their default values which may depend on the

deployment environment, known conditions etc. After each iteration in the algorithm, the agent keeps improving the estimates of these parameters as it gains more experience of using schemes from the $\langle n_{\rm c}, n_{\rm p}\rangle$ plane in the real world. Note that in Section II, we considered just two schemes and it was easy to track goodness of each scheme by tracking its state information. In this case, the model (10) is used to compute the goodness of schemes. Note that better the estimates of parameters c_1, c_2 and c_3, the better is the goodness of schemes computed using the model (10).

We implement a stochastic gradient descent (SGD) algorithm for tuning the parameters c_1, c_2, c_3. A cross-entropy cost function is formulated as:

$$J(c_1,c_2,c_3) = -p^{\text{success}}_{\text{true}}(n_{\rm c},n_{\rm p})\log(p^{\text{success}}_{c_1,c_2,c_3}(n_{\rm c},n_{\rm p})g(n_{\rm c})) \\ -(1-p^{\text{success}}_{\text{true}}(n_{\rm c},n_{\rm p}))(1-\log(p^{\text{success}}_{c_1,c_2,c_3}(n_{\rm c},n_{\rm p})g(n_{\rm c}))). \tag{13}$$

The objective of this algorithm is to tune the parameters in an online manner with each packet transmission and eventually converge to values which closely represent the true channel. In order to verify the performance of the algorithm, we set the boundaries in the $\langle n_{\rm c}, n_{\rm p}\rangle$ plane representing the true channel as following: $n_{\rm c} > 256$, $n_{\rm p} > 128$ and $n_{\rm c} + n_{\rm p} < 1152$. The true values of the probability of success is computed using these boundaries. The gradient of the cost function (13) with respect to the parameters are computed using `Theano` module in `Python`. The values of $c_1 = -50$, $c_2 = -50$ and $c_3 = -2000$ is set as initial values. We can see that the viable region represented by these initial boundaries is much larger (see Fig. 4) and hence will result in errors each time a scheme is selected outside the region represented by the true values. We run the algorithm for 1000 iterations and the result can be observed in Fig. 5. The viable region is converged and now represents the true region more accurately. The values of parameter at the convergence are $c_1 = 261.7$, $c_2 = 132.4$ and $c_3 = 1806$.

IV. Hybrid algorithm for tuning OFDM link

Now that we understand how the parameters of the model presented in (10) are tuned and how the statistical information gained with each real-world experience causes improvement in the agent's knowledge of the channel, we propose in this section a hybrid algorithm which utilizes both as part of the link tuning.

Note that the reward associated with each scheme $\langle n_{\rm c}, n_{\rm p}\rangle$ is the effective data rate computed as:

$$\gamma_{n_{\rm c},n_{\rm p}} = \frac{n_{\rm c}B}{n_{\rm c}+n_{\rm p}}. \tag{14}$$

Algorithm 1 utilizes the model represented in (10) to keep tuning the boundaries forming the viable region in the time-varying channel conditions and at the same time, the statistical information gained through real experience is utilized in taking better decisions.

A. Simulation results

The following setup is considered for the simulation. A multipath underwater channel is considered with delay spread $\tau_{\rm ds} = 10$ ms, a channel coherence time $\tau_{\rm c} = 70$ ms, bandwidth $B = 25$ KHz. Based on the above values the boundaries on the $\langle n_{\rm c}, n_{\rm p}\rangle$ plane are computed as $n_{\rm c} > 591.01$, $n_{\rm p} > 250$ and $n_{\rm c} + n_{\rm p} < 1750$. These boundaries serve as the simulator for the real world experience. The learning rate $\alpha = 50$ and sigmoid slope, $\beta = 0.1$ are used. The model parameters $h = \sigma = 700$ is set. A total of 480 schemes are generated, with $n_{\rm c}$ ranging from 32 to 4096 and $n_{\rm p}$ ranging from 240 to 300. Since the probability of success peaks at the value of 700 and stays the same for all values of $n_{\rm p}$, it is expected that the maximum reward/data rate should converge close to $\frac{700\times 25000}{700+300} = 17.5$ kbps. The random policy, ϵ-Greedy policy and UCB policy with and without the model are examined and the results are compared in Fig. 6.

Algorithm 1 Hybrid link tuning algorithm

procedure LINKTUNER(learning rate α, sigmoid slope β, policy Π, Number of schemes n)
  initialize parameters c_1, c_2, c_3
  Set initial state $S^0 := \{m_0^0, p_0^0, m_1^0, p_1^0, \cdots, m_n^0, p_n^0\}$
  for every time step $k = 0, 1, 2, \cdots$ **do**
    Take action based on policy, $s_i \leftarrow \Pi(S^k)$ where i is the index of the tuple corresponding to $(n_{\rm c}, n_{\rm p})$
    Update the state, $m_i \leftarrow m_i + 1$, $p_i \leftarrow (1-\frac{1}{k})p_i + \frac{1}{k}p^{\text{success}}_{c_1,c_2,c_3}(n_{\rm c},n_{\rm p})$
    Update the reward, $r \leftarrow (1-\frac{1}{k})r + \frac{1}{k}\gamma_{n_{\rm c},n_{\rm p}}p_i$
    Compute the cost using (13) and update the parameters c_1, c_2, c_3 based on the computed gradients,
$$\begin{cases} c_1 \leftarrow c_1 - \alpha\frac{\partial J(c_1,c_2,c_3)}{\partial c_1} \\ c_2 \leftarrow c_2 - \alpha\frac{\partial J(c_1,c_2,c_3)}{\partial c_2} \\ c_3 \leftarrow c_3 - \alpha\frac{\partial J(c_1,c_2,c_3)}{\partial c_3} \end{cases}$$
    Update the parametric model, $p^{\text{success}}_{c_1,c_2,c_3}(n_{\rm c},n_{\rm p})$

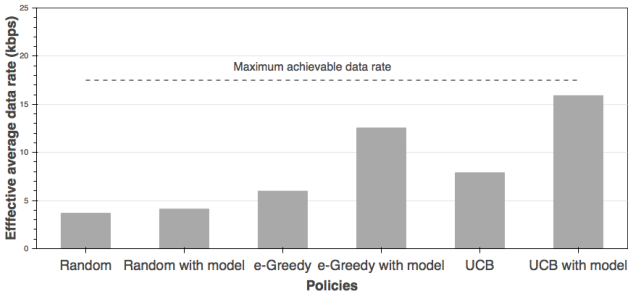

Fig. 6: Policy comparison with and without model.

V. Conclusions

We presented a technique which utilizes the channel physics such as the dependence on delay spread, channel coherence time and bandwidth to keep track of the parameters of the model representing the probability of packet success. The data-driven algorithm is used in conjunction with the parameter tuning to take better sequential decisions resulting in maximizing the average data rate. The UCB algorithm along with model parameter learning was shown to be promising and simple to use. The algorithm presented is extensible and more complex physical models can be included. The physical model to use depends on the modulation schemes supported on an underwater acoustic modem. Using a different physical model neither effects the data-driven part nor the learning algorithm to tune the model parameters and therefore the proposed algorithm can be studied with various channel models applied to the viable region.

REFERENCES

[1] M. Stojanovic, "Underwater acoustic communications," in *Electro/95 International. Professional Program Proceedings.* IEEE, 1995, pp. 435–440.

[2] V. D. Valerio, C. Petrioli, L. Pescosolido, and M. Van Der Shaar, "A reinforcement learning-based data-link protocol for underwater acoustic communications," in *Proceedings of the 10th International Conference on Underwater Networks & Systems*. ACM, 2015, p. 2.

[3] L. Wan, H. Zhou, X. Xu, Y. Huang, S. Zhou, Z. Shi, and J.-H. Cui, "Adaptive modulation and coding for underwater acoustic OFDM," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 327–336, 2015.

[4] K. Pelekanakis, L. Cazzanti, G. Zappa, and J. Alves, "Decision tree-based adaptive modulation for underwater acoustic communications," in *Underwater Communications and Networking Conference (UComms), 2016 IEEE Third*. IEEE, 2016, pp. 1–5.

[5] S. Shankar and M. Chitre, "Tuning an underwater communication link," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–9.

[6] E. Demirors, J. Shi, R. Guida, and T. Melodia, "SEANet G2: toward a high-data-rate software-defined underwater acoustic networking platform," in *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems*. ACM, 2016, p. 12.

[7] H. Luo, K. Wu, R. Ruby, F. Hong, Z. Guo, and L. M. Ni, "Simulation and experimentation platforms for underwater acoustic sensor networks: Advancements and challenges," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 28, 2017.

[8] H. S. Dol, P. Casari, T. Van Der Zwan, and R. Otnes, "Software-defined underwater acoustic modems: Historical review and the NILUS approach," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 3, pp. 722–737, 2017.

[9] M. Chitre, I. Topor, and T.-B. Koay, "The UNET-2 modem—an extensible tool for underwater networking research," in *OCEANS, 2012-Yeosu*. IEEE, 2012, pp. 1–7.

[10] M. Chitre, R. Bhatnagar, and W.-S. Soh, "Unetstack: An agent-based software stack and simulator for underwater networks," in *Oceans-St. John's, 2014*. IEEE, 2014, pp. 1–10.

[11] D. A. Berry and B. Fristedt, "Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability)," *London: Chapman and Hall*, vol. 5, pp. 71–87, 1985.

[12] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion," *Statistical science*, pp. 101–117, 2001.

[13] J. C. Gittins and D. M. Jones, "A dynamic allocation index for the discounted multiarmed bandit problem," *Biometrika*, vol. 66, no. 3, pp. 561–565, 1979.

[14] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[15] J. Chakravorty and A. Mahajan, "Multi-armed bandits, gittins index, and its calculation," *Methods and Applications of Statistics in Clinical Trials: Planning, Analysis, and Inferential Methods, Volume 2*, pp. 416–435, 2014.

[16] M. Chitre, "Underwater acoustic communications in warm shallow water channels," *Ph.D. Thesis, Singapore: National University of Singapore*, 2006.