Protocol Sequence Design Through Deep Reinforcement Learning

Chetanveer Gobin

and

Department of Telecommunications Engineering

Protocol Sequence Design Through Deep Reinforcement Learning

August 27, 2022

Internship Report

Chetanveer Gobin

Supervised by:
Chung Shue (Calvin) Chen, Nokia Bell Labs
Cédric Adjih, Inria Saclay
Iman Hmedoush, Inria Saclay

Contents

1 About Nokia Bell Labs **3**
- 1.1 Nokia's Activities . . . 3
- 1.2 Clients Of Nokia Bell Labs . . . 3
- 1.3 Consumer Products . . . 3

2 Internship Context **4**
- 2.1 General Introduction . . . 4
- 2.2 Suggested Approach . . . 4
- 2.3 Aim of the Internship . . . 4

3 Design of Protocol Sequences **5**
- 3.1 Channel Model . . . 5
- 3.2 Notion of a Protocol Sequence . . . 5
  - 3.2.1 Shift Invariant (SI) Protocol Sequences . . . 5
  - 3.2.2 Pairwise SI Protocol Sequences . . . 5
  - 3.2.3 User Irrepressibility (UI) Protocol Sequences . . . 6
  - 3.2.4 Conflict Avoiding Codes (CAC) . . . 6

4 Planning and Approach to Address the Problem **6**

5 State of the Art and Literature Review **7**
- 5.1 AlphaSeq . . . 7
  - 5.1.1 Design of Deterministic Grant-Free Access with Deep Reinforcement Learning . . . 7
  - 5.1.2 AI Coding - Learning to Construct Error Correction Codes . . . 7
  - 5.1.3 Construction of Polar Codes with Reinforcement Learning . . . 7

6 Problem Statement **8**
- 6.1 Mathematical Description of TMUI Sequences . . . 8
  - 6.1.1 UI Condition . . . 8
  - 6.1.2 Throughput Maximization . . . 9
- 6.2 Correctness of a Sequence Set . . . 9

7 Applying DRL Approach to Find Protocol Sequences **10**
- 7.1 Proposed DRL Approach . . . 10
- 7.2 Reward Function . . . 10
  - 7.2.1 Mathematical Formulation of the Reward . . . 10

8 Numerical Results **11**
- 8.1 Result Analysis . . . 11

9 Internship Activities **11**

10 Management Aspect **12**

11 Conclusion **13**
- 11.1 Contribution of the Internship . . . 13
- 11.2 Applied Knowledge . . . 13
- 11.3 Theoretical Knowledge . . . 13
- 11.4 Plan for the Next 3 Years . . . 14

1 About Nokia Bell Labs

Nokia creates technology that helps the world act together. As a trusted partner for critical networks, Nokia is committed to innovation and technology leadership across mobile, fixed, and cloud networks. They create value with intellectual property and long-term research, led by the award-winning Nokia Bell Labs. Adhering to the highest standards of integrity and security, they help to build the capabilities needed for a more productive, sustainable, and inclusive world.

1.1 Nokia's Activities

Nokia Bell Labs provides solutions that enable service providers, enterprises and governments worldwide, to deliver voice, data, and video communication services to end-users. As a leader in fixed, mobile, and converged broadband networking, IP technologies, applications, and services, Nokia Bell Labs offers end-to-end solutions that enable compelling communications services for people at home, at work, and on the move. The company has the most experienced global services team in the industry and is one of the largest research, technology, and innovation organizations in the telecommunications industry. The Research and Innovation (R&I) activities concern three primary domains: applications and services, networking, and optics and radio.

The French (R&I) centre which is now located in Nozay, near Paris, is involved in the three above topics. The lab, which is active in "pôle de compétitivité" (Innovation Center) System@TIC Paris-Région, has notably been involved in European projects within ESPRIT, RACE, ACTS, IST frameworks, and Eureka/ ITEA-CELTIC, as well as national cooperative research networks. The (R&I) mission is to prepare Nokia's future and to foster innovation by investigating new technologies. This mission is implemented by feasibility studies and incubation of new disruptive products and solutions concepts that will ensure the development of new profitable services for Nokia Bell Labs customers. (R&I) participate in research projects in global scientific and technology-related research initiatives on a worldwide basis, in conjunction with academic, industrial, and governmental research partners. It is also active in standardization bodies and forums.

1.2 Clients Of Nokia Bell Labs

Communications service providers Partnering with communications service providers to deliver critical networks with the highest performance, reliability, and security to grow their business.

1.3 Consumer Products

Licensing a range of consumer products that put better experiences at the heart of every day. Energy and resources, empowering the customer in the era of Industry 4.0 and meeting his sustainability goals and finding successful new business models in an evolving energy and resources industry is possible. Nokia helps the customer to accelerate Industry 4.0 technologies in his operation to help him achieve these goals and much more. Nokia can act together with the customer to take even his most challenging, remote operation to the next level of intelligence, automation, sustainability, and more with proven mission-critical network and cloud solutions.

2 Internship Context

2.1 General Introduction

The Internet of Things (IoT) aims to connect millions of devices, most of which are power and memory constrained. Such constraints require efficient network access and very low complexity multiple access protocols. One can trace back to J. Massey's consideration of collision channel without feedback (CCw/oFB) problem [1] in which binary protocol sequences were made for each user. Through combinatorics and number/coding theory such as CRT (Chinese Reminder Theorem), one can design a set of conflict-avoiding sequences or codewords (CAC) for a user service requirement or corresponding QoS guarantee [2]. The central question here is: what are the binary sequences or codewords that would have better/best service performance? The metrics can be the throughput of the user, the success delivery probability (cf. highly ultra-reliable communications), the individual service delay, the group delay (cf. collaborative task completion), the latency, etc. Conventional Bernoulli random access transmission scheme is one common practice. However, would a set of carefully chosen codewords (sequences) work better than a set of random binary sequences? What is the optimal design for such a given system?

2.2 Suggested Approach

We will first consider the simple collision channel model adopted by [1] and then the realistic interference or SINR based model (cf. signal-to-interference-plus-noise ratio). One can also consider a wireless ad hoc mobile network or a hexagonal cellular mobile network spatial model [3]. We can consider a fully asynchronous or a partially coordinated system. Would the structure of the codewords matter and why (for example their pairwise cross-correlation properties)? An even more basic question is: what structure would work better for a system (how much better and in which manner/metric)?

There is some more understanding of the above question when considering a simple collision channel model in the existing literature. Knowledge and results under SINR or spatial model are much less. From a combinatorics perspective, the number of possible binary sequences is large; for sequence length equal to L, the number of possible combinations is given by 2^L. Among all the combinations, is there a set of sequences which would offer better performance? We cannot do an exhaustive search to find the best sequence set. However, there are not many mathematical or analytical tools that we can use to formulate the problem and derive suitable codewords. For example, one can revisit the literature on prime sequences, orthogonal optical code (OCC), etc [4–6]. Besides, one can also consider successive interference cancellation (SIC) system for achieving higher channel capacity. It is also suitable to consider multiple-packet reception (MPR) capability [7]. AI/ML/RL is a promising alternative and can be used to discover sequences of desired properties. There are promising new results and interesting applications, see for example [8, 9] and references therein. One can also consider the sequence discovery problem as an episodic symbol-filling game. Each episode ends with a completely filled sequence set, upon which a reward is given based on the desirability of the sequence set with Markov Decision Process (MDP). Compared with traditional sequence construction by mathematical tools [10], machine learning is particularly suitable for problems with complex objectives intractable by mathematical analysis [11]. In telecommunications, AI/ML has shown many successful applications to physical layer (PHY). However, its use and effectiveness for MAC still needs exploration and further investigation.

2.3 Aim of the Internship

Sequences play an important role in many applications and systems. During the internship, we will focus on AI/ML for MAC protocol design. We intend to implement and compare various machine learning methods, to optimize their schemes and analyze the performance. Among many possible applications of protocol sequences, we will consider modern random access system for wireless IoT. Techniques can be reused for other similar problems.

3 Design of Protocol Sequences

In this section, we are going to introduce the channel model as well as the most common protocol sequences.

3.1 Channel Model

We consider the collision channel without feedback (CCw/oFB) model, as in [1]. It can have many applications, notably in the field of wireless IoT. In applications such as monitoring and surveillance, wireless sensor networks have computing power limitations and energy consumption constraints. It will be favorable to have a simple multiple access protocol that does not require frequent monitoring of the channel for feedback information and does not require complicated processing mechanism such as back-off algorithm and random number generation, such as what is being used in Wi-Fi nowadays. Besides, when considering an ad-hoc system with dynamic network topology which may be due to user mobility or time-varying propagation delays, sharing a radio channel among many devices with the requirement of well-coordinated transmissions and time offset synchronization could be very complicated. This is particularly hard for thin devices. The above model considers the situation where M users share a common communication channel but because of random starting time of the users and also the lack of a feedback link for exchanging information, they cannot coordinate their transmissions for achieving a collision-free scheme such as TDMA. The above scenario leads to our following exploration.

3.2 Notion of a Protocol Sequence

Protocol sequences are used for defining the user access in the collision channel without feedback. Each user is assigned a deterministic zero-one binary sequence, called a protocol sequence. The zeros and ones in a protocol sequence are read out periodically, and a packet is sent if and only if it is one. A collision occurs if two or more users transmit at the same time. Each user is required to make its packet transmissions at times determined by a protocol sequence that is independent of the data to be sent.

Depending on the quality of service requirement (QoS), there are many design perspectives for protocol sequences. According to the state of the art, the following types of sequences are the mainstream ones. A brief explanation is provided for each one of them.

3.2.1 Shift Invariant (SI) Protocol Sequences

In the context of shift invariant (SI) protocol sequences [12], all users (let's say M users) transmit according to their respective attributed sequence. SI means that there is a guarantee that each user will have at least one success (i.e., collision-free transmission) and the goodput (i.e., the number of successes) is constant no matter what the time offsets are among the users, over the sequence period. A time offset applied to a user can be considered as a time shift in the binary sequence. In the example shown below, we see that there are 3 sequences, each of length $L = 8$. If each user transmits according to their assigned protocol sequence, we can see that for each user there is exactly one success out of L, no matter what their relative time offsets are.

$$\begin{aligned} S_0 &= \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \\ S_1 &= \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \\ S_2 &= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

3.2.2 Pairwise SI Protocol Sequences

Pairwise shift invariant protocol sequence [13] is another family of protocol sequences. It is a relaxed version of SI protocol sequences. In this setting, one may consider that there are M users in the system wanting to share a common channel. However, only two users are active and transmitting at the same time. Each user is then guaranteed to have at least one success and the goodput is constant for any relative time offset. The following is an example of a set of pairwise SI protocol sequences. One can check that for any two sequences among the three, the goodput of the user is constant no matter what the relative offset is.

$$S_0 = (1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ \ 0\ 0)$$
$$S_1 = (1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \ 0\ 0)$$
$$S_2 = (1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ \ 0\ 0)$$

3.2.3 User Irrepressibility (UI) Protocol Sequences

UI protocol sequences aim for non-totally blockage systems [14]. The design objective, called user-irrepressibility, is to guarantee that even in the worst case, each user can send at least one packet successfully in each period [15]. In other words, no matter what the relative time offsets are, there is at least one successful packet for each user in each period [16].

Note that if frame synchronization, which is a strong assumption, is allowed, the above medium access control can simply apply a conventional time division multiple access (TDMA) scheme such that ideally transmission collision would be avoided. However, if only slot synchronization is allowed, the relative time offsets among the users are uncontrollable and transmission collision can occur.

3.2.4 Conflict Avoiding Codes (CAC)

The notion of user-irrepressibility can be addressed from the perspective of conflict-avoiding codes (CAC) [17]. In the study of CAC, there are M potential users, among which at most K of them can be active at the same time. Given the sequence period L, the objective in the construction of CAC is to maximize the number of potential users, which is denoted by K, with the service guarantee that there is at least one collision-free transmission for each active user in a period of time.

4 Planning and Approach to Address the Problem

Proper planning was set up as shown in the time diagram below, see Fig. 1. First, we conduct a study of the literature on protocol sequences and their existing AI/ML/RL based construction methods. The second step is to analyze the mathematical characteristics of UI sequences and also understand the constraints of various AI/ML/RL models, thus think about which learning algorithms from the state of the art can be relevant for addressing our problem and how to apply. The third step to select suitable AI/ML/RL schemes and implement to investigate and evaluate (we would use Stable Baselines3 [18]). Finally, we provide suggestion or recommendation of suitable method for designing protocol sequences through AI/ML/RL techniques.

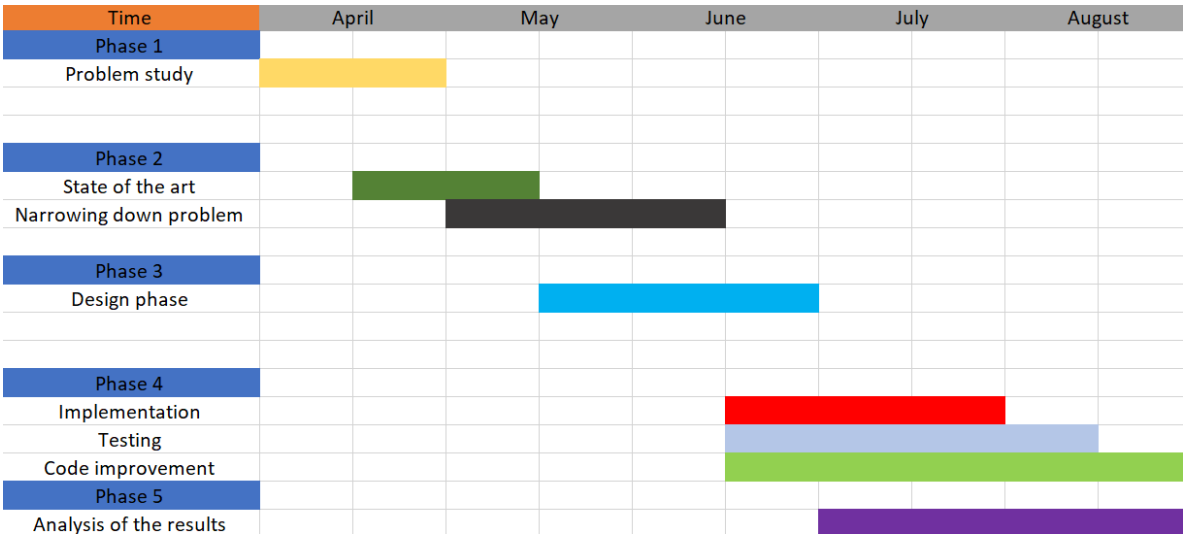

Figure 1: Time plan of the internship

5 State of the Art and Literature Review

From the literature, we find that Deep Reinforcement Learning (DRL) is a natural technique to use when we need to design sequences. In the following, we present a summary of the main papers that we studied and how they applied DRL for sequence design and how they would outperform the other methods.

5.1 AlphaSeq

The most interesting paper that relates to our problem is AlphaSeq [11]. The authors proposed a new algorithm for sequence discovery. The algorithm is inspired by AlphaGo of DeepMind. This is used to discover desired sequences algorithmically using Deep Reinforcement Learning (DRL) techniques. AlphaSeq treats the sequence discovery problem as an episodic symbol-filling game, in which a player fills symbols in the vacant positions of a sequence set sequentially during an episode of the game. Each episode ends with a completely filled sequence set, upon which a reward is given based on the desirability of the sequence set. AlphaSeq models the game as a Markov Decision Process (MDP) and adapts the DRL framework of AlphaGo to solve the MDP. Compared with traditional sequence construction by mathematical tools, AlphaSeq is particularly suitable for problems with complex objectives intractable by mathematical analysis. In this paper, it was demonstrated that AlphaSeq could successful solve the following two engineering problems, notably:

- Rediscovering a set of ideal complementary codes that can zero-force all potential interferences in multi-carrier code-division multiple access (CDMA) systems.
- Discovering new sequences that triple the signal-to-interference ratio benchmarked against the Legendre sequence.

5.1.1 Design of Deterministic Grant-Free Access with Deep Reinforcement Learning

Another paper that is inspired by AlphaSeq [11] is [19], which aims to design deterministic grant-free access protocol through deep reinforcement learning method. In the paper, the authors designed interference canceling (IC) codes, with the consideration of successive interference cancellation technique at the physical layer. The result is for ultra reliable low-latency communications (URLLC). Note that it is difficult to obtain IC codes by the existing mathematical tools or traditional search algorithms. To fill this gap, a DRL based algorithm is put forward to search IC codes, with carefully designed metrics and reward functions as per the underlying mathematical constraints. Result indicates that the algorithm can efficiently discover IC codes. Performance evaluation by simulations shows that the discovered IC codes yield significantly lower failure probability than existing random access protocol given the same latency requirements, and thus they are more suitable for URLLC.

5.1.2 AI Coding - Learning to Construct Error Correction Codes

In [20], a Reinforcement Learning (RL) approach is investigated to design Error Correction Codes (ECC). A constructor-evaluator framework is proposed in which the code constructor can be realized by various AI algorithms and the code evaluator provides code performance metric measurements. In the paper, the authors used the Advantage Actor Critic (A2C) for the constructor. The code constructor keeps improving the code construction to maximize the code performance, which is evaluated by the code evaluator until the performance metric converges. The results show that comparable code performance can be achieved for the existing codes. It is noteworthy that the method described can provide superior performance to classic constructions in certain cases, for example in decoding polar codes.

5.1.3 Construction of Polar Codes with Reinforcement Learning

Another interesting approach is [21], which formulates the sequence discovery problem as a maze-traversing game. The attempt is to use reinforcement learning techniques to construct polar codes for Successive-Cancellation List (SCL) decoder.

The tabular RL algorithm SARSA(λ) was adopted to solve the game. Simulation results showed that the game-based constructions matched the standard polar code constructions under the SC decoder for short codes. For short codes under the SCL decoder and longer codes under both the SC and SCL decoders, the game-based method was able to find polar code constructions that outperform existing standard constructions quite significantly. Moreover, the game-based method is very efficient during training in terms of the number of required training samples.

6 Problem Statement

It is known that protocol sequences can be designed according to their required QoS. Traditional sequence design is to use tools from mathematical analysis, combinatorics, number theory, and coding theory. They are often designed by algebraists and information theorists using mathematical tools such as finite field theory, algebraic number theory, character theory, etc. However, the design criteria for sequences may be complex and cannot be put into a clean mathematical expression for a solution by available mathematical tools. To avoid the design complexity, another interesting direction of how to design protocol sequences is to use ML techniques. Reinforcement learning (RL) is an important branch of machine learning known for its ability to derive solutions for Markov decision processes (MDPs) through a learning process.

In the literature, a well-known family of sequences is the UI sequences. This family of sequences offers the property of being non-totally blockage. However, the user throughput of UI sequences is often low as they are not designed for achieving high throughput. Therefore, in this internship we decide to provide a new family of sequences, namely "Throughput Maximizing User Irrepressible Sequences" (TMUI). The new sequences, not only offers the property of user irrepressibility, but aims to have the goodput of each user higher than the original UI sequences. Note that, by definition, the throughput (goodput) of the sequence is a measure of successful transmissions. The objective of TMUI is not only to satisfy UI property but also to send as much data as possible, especially in a mobile environment. An example of applications where users might require this QoS is industrial IoT (IIoT)), where moving robots need reliable communication channel with strong QoS guarantee in order to coordinate with each other to complete a given task.

In this work, we are interested in the use of Reinforcement Learning techniques for protocol sequence design, as done for instance in [11, 20, 21]. We intend to use similar techniques to discover a family of TMUI sequences.

6.1 Mathematical Description of TMUI Sequences

In this section, we provide a general definition of TMUI protocol sequences and the related optimization problem formalization in order to find optimal TMUI protocol sequences.

Let $\mathcal{S}$ denote a set of binary $\{0, 1\}$ scheduling sequences for deterministic access, which consists of M sequences, each is of length N, i.e., $\mathcal{S} \triangleq \{s_0, s_1, \ldots, s_{M-1}\}$. To transmit a packet, the user will use a sequence $s_i \triangleq (s_i[0], s_i[1], \ldots, s_i[N-1])$, for $i = 0, 1, 2, \ldots, M-1$. If a user uses the sequence s_i to transmit its packet in the frame, it transmits at the slot t if and only if $s_i[t] = 1$.

We define the duty factor of a sequence as "the fraction of the time in a period N for which each user is transmitting." For example, duty factor equal to $\frac{1}{2}$ means that the user is transmitting 50% of the time, i.e., $\frac{N}{2}$, where N is the number of slots in a sequence.

In order to construct TMUI sequences, we want to achieve the following two conditions:

- The sequence set needs to be UI.
- Throughput maximization of each user is obtained.

In the following section, we describe these two conditions.

6.1.1 UI Condition

To understand the UI condition, first we need to understand the notion of a characteristic set. The characteristic set is an alternative representation of a sequence. It simply denotes the positions of 1's in the sequence. Note that a protocol sequence can be presented by a sequence of numbers, where

each number is used to describe the positions of the slots where the user transmits. As an example, consider the following sequence, $s \doteq$ [0,1,1,1,0,0,0,1]. Its characteristic set is given by $I_s \doteq \{1, 2, 3, 7\}$. The characteristic set is a very convenient way of representing a sequence and good for mathematical analysis [16]. This is why during this internship, We will adopt the same representation in the following.

We denote a shift by $\tau \in \{0, 1, ..., M-1\}$. It is an addition modulo $I_{s'}$=$I_s \bigoplus \tau$. Note that s' is the shifted sequence by τ.

Consider one family of sequences consisting of two users. ω_1 represents the set of all the shifts for the sequence S_1.

$$\omega_1 = \left\{ s_1^{(1)}, s_1^{(2)}, ..., s_1^{(N)} \right\}$$

In the meantime, ω_2 is the set of all the shifts for the sequence s_2.

$$\omega_2 = \left\{ s_2^{(1)}, s_2^{(2)}, ..., s_2^{(N)} \right\}$$

For UI property to be true, the following condition must be satisfied for all shifts:

$$\omega_1 \cap \omega_2 = \emptyset.$$

6.1.2 Throughput Maximization

We define the throughput of one sequence as the number of successful transmissions that could be obtained using this sequence. We denote the throughput of a sequence s_i as $T(s_i)$. The goal is to maximize the throughput of each sequence such that it reaches the target that we set.

The two properties, user impressibility and throughput maximization, are required for our method to obtain, by TMUI sequences. We would use a Deep Reinforcement Learning (DRL) approach.

6.2 Correctness of a Sequence Set

This notion is used to derive a metric of how good a sequence set is. It is a fundamental part of the derivation of a reward function for the agent in the RL setup. To quantify the correctness of a sequence set, a metric is necessary. It is denoted by the function $C(S)$, where S is the sequence set.

The diagram, in Fig. 2, illustrates the metric $C(S)$ of the correctness of a sequence set graphically.

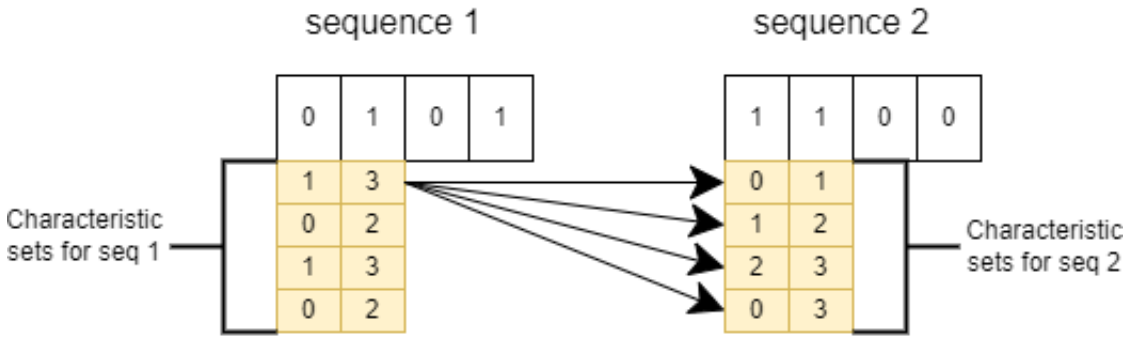

Figure 2: Calculation of sequence set correctness

Fig. 2 shows two sequences. Each sequence has associated with it the characteristic sets for all the possible shifts. In this example, we want a target throughput of 1 out of 4 per user. In other words, we want each user to succeed one out of four transmissions.

Consider the first arrow, i.e., the pairs $\{1, 3\}$ from the first sequence and $\{0, 1\}$ from the second sequence. $\{1\}$ is present only in sequence 1, hence $\{1\}$ is a success for sequence 1. Similarly, $\{0\}$ is present only in sequence 2. This means that $\{0\}$ is a success for sequence 2. Therefore, the pair $\{1, 3\}$ and $\{0, 1\}$ meet the objective of the target throughput per user. The correctness of the sequence set is defined as the number of characteristic set pairs that satisfy the target throughput per user. In the example shown above, the correctness of this sequence set $C(S) = 16$. That is, 16 characteristic set pairs meet the predefined target throughput(of 1) per user. It can be seen that the maximum correctness $C_{\max}(S) = L^2$, where L is the length of the sequence.

7 Applying DRL Approach to Find Protocol Sequences

We use a Reinforcement Learning (RL) framework. This paradigm is an important branch of machine learning known for its ability to derive solutions for Markov decision processes (MDPs) through a learning process. In the framework, an agent interacts with an environment in a sequence of discrete-time steps. At time step t, the agent observes that the environment is in state s_t. Based on the observation of s_t, the agent then takes an action a_t, which results in the agent receiving a reward R_{t+1} and the environment moving to state s_{t+1}, respectively.

The agent aims to maximize an objective, which is the reward function. It learns to do this by playing many episodes of the game, improving itself along the way. This boils down to an optimization problem, that we solve using a policy gradient method. In our case, we use the algorithm Proximal Policy Optimization (PPO) [22].

7.1 Proposed DRL Approach

The agent is a Neural Network (NN) and it is going to be interacting with an environment as described below.

The proposed framework, treats the sequence discovering process unclear as an episodic bit-flipping game, the bit flipper being the environment. At each step of the game, the agent flips two opposite bits in a sequence.

- The state is the collection of sequences with their respective characteristic sets.
- The action is the position of the bit flips.
- The reward is described below.

  In the following example, see Fig. 3, the length of the sequence $L = 8$. In the first step, the neural network flips the bits 1 and 2 for user 1, i,e., the action is (1,2) for user 1. In the second step, it flips the bits 4 and 5 for user 2. The episode either ends after a predefined number of steps or when a correct sequence set is found. To avoid the problem of sparse rewards [23], a reward is returned at each step of the game. These step rewards are stored in a buffer at every step including the end of the episode, it is the maximum of the stored reward that is returned by the environment. The reward is derived based on a metric, which is the correctness of a sequence. The more the number of episodes, the better the agent (neural network) gets at the bit-flipping game. In the next section, we detail the reward function and its calculation.

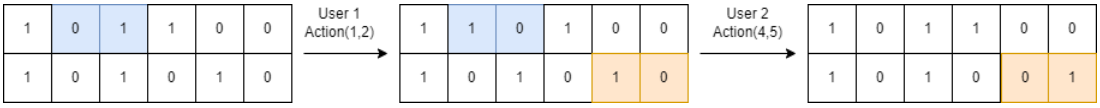

Figure 3: Actions taken on sequences

7.2 Reward Function

To come up with a good reward function, the notion of correctness of the sequence is needed.

The used technique is known as dubbed "segmented induction" [11], and it involves changing the reward function progressively to guide the neural network toward a good sequence set during the learning phase. This means that a low target is initially set and gradually this target is increased.

7.2.1 Mathematical Formulation of the Reward

The reward defined at each step is an indication of how close the sequence set is to the one that meets the objective of being completely UI and maximizing the throughput of each user. The negative sign is added in front of the reward function so that it maximizes the reward in a minimum number of steps.

$$R = -(1 - \frac{C(s)}{C_{\max}(s)})$$

$C_{\max}(s)$ denotes the maximum correctness. Hence, the maximum reward $R_{\max} = 0$

8 Numerical Results

Our framework has been tested for two users as a proof of concept. The duty factor of each user was fixed at $\frac{1}{2}$. This means out of a sequence of length L, each user is transmitting $\frac{L}{2}$ times. If L is odd, the floor of $\frac{L}{2}$ is taken. The length of the protocol sequence was varied from $8 \leq L \leq 20$. However, the search space increases exponentially with L. Note that the search space can be improved and this is a future work of the topic.

8.1 Result Analysis

In Fig. 4, we show the reward convergence with respect to the number of learning episodes for different seeds. It can be observed that the reward converges to 0, which is the maximum reward. This proves that the proposed framework is working very well towards finding our targeted sequences.

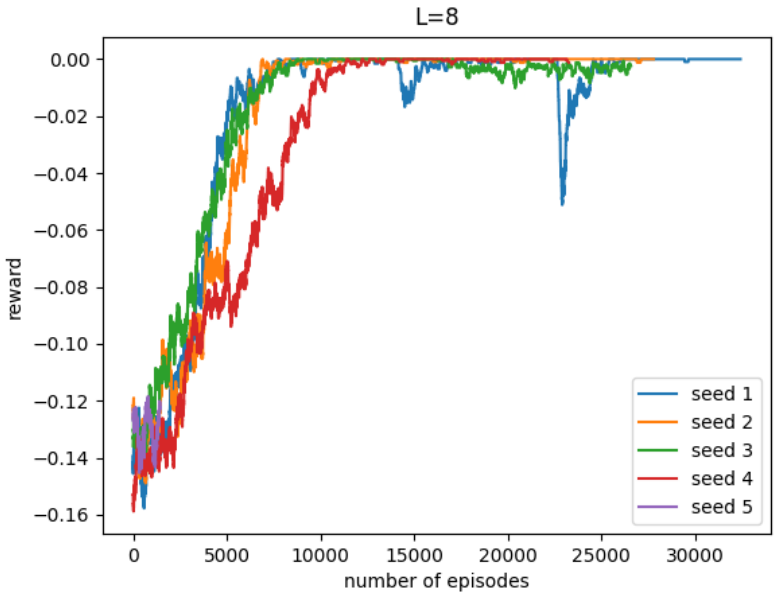

Figure 4: Reward convergence for $L = 8$

For $L = 8$, we do see that the curve converged nicely at around 10000 episodes. The sequences at that point have a guaranteed throughput of 2 success transmissions out of 8 transmissions.

For $L = 12$, as shown in Fig. 5, we do see that the curve converged nicely at around 10000 episodes. The sequences at that point have a guaranteed throughput of 3 out of 12.

For $L = 16$, as shown in Fig. 6 we do see that the curve converged nicely at around 150000 episodes. The sequences at that point have a guaranteed throughput of 4 out of 16. Some seeds do not make it to the optimal reward of 0. This happens when the training has not been running for long enough at the time.

9 Internship Activities

I have been with the ML & System Research Department of Nokia Bell Labs. The hosting department of my internship consists of research scientists, research engineers, Ph.D. students, and research interns. It is my great pleasure to learn from the team and there are very good research discussions and interactions among the colleagues.

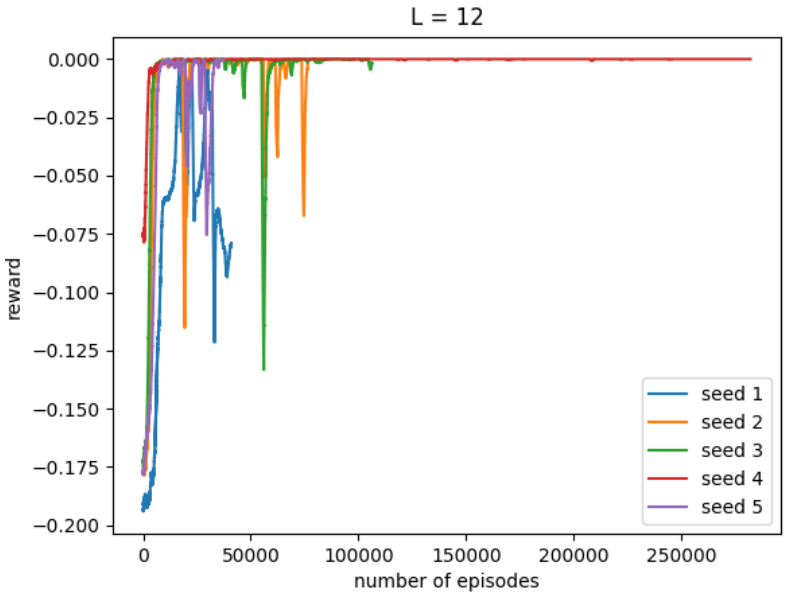

Figure 5: Convergence curve $L = 12$

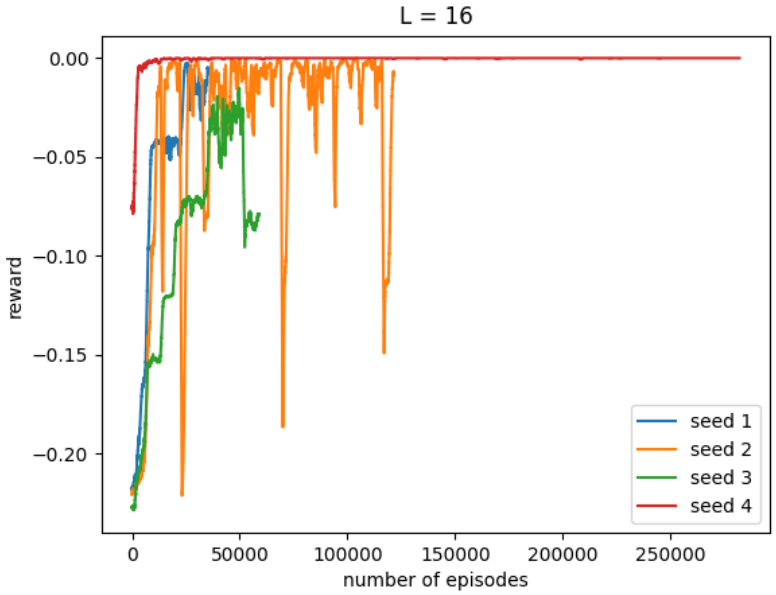

Figure 6: Reward convergence $L = 16$

10 Management Aspect

Since this is a research internship, a lot of my work was done autonomously. In the meantime, there were a lot of python codes and various versions to work with and manage. For this purpose, a version control system, Git was used to keep track of every change and to revert in case of a mistake or bug.

To share the code with my supervisors, Chung Shue (Calvin) Chen, Cédric Adjih, and Iman Hmedoush, we used the hosting service GitHub. This tool allowed us to track the code changes made by each editor.

For communication and daily exchange, we used emails and teleconferencing and also held weekly face-to-face work meetings.

11 Conclusion

During my internship at Bell Labs, I had the opportunity to learn various research projects. My research project is focus on protocol sequences. In addition, I practiced what I have learned at INSA Lyon, in the telecommunications department. I believe this is the most essential part of a good internship, as this is also what French engineering schools are concentrating on, the combination of theoretical knowledge with applied knowledge.

However, there were still several difficulties that confronted me during the internship. Notably, the lack of a strong background in Reinforcement Learning, especially with complex algorithms such as PPO [22]. Furthermore, this internship was quite different compared to general engineering training. For instance, being a researcher, one would probably not end up with the expected result but this result is still worthy, whereas being an engineer, the outcome of a project is the only criteria for evaluating one's work. In other words, researchers and engineers may not have the same goal. Consequently, when confronted with issues, they may have different points of view to tackle a problem and perform their work differently.

Thanks to these difficulties experienced during the internship, I have enhanced my knowledge and become more autonomous when facing a challenging topic. Most importantly, I acquired a new manner to look at questions and I believe that this skill would be helpful to my future studies or career, since I understand various thinking and would be able to work effectively under various environment. Besides, another valuable character I learned is perseverance, which is especially important in scientific research.

11.1 Contribution of the Internship

Several contributions have been conducted during the internship.

- Design and implementation of protocol sequences using DRL
- Proof of reproducibility through multiple simulations

11.2 Applied Knowledge

The list below describes some applied knowledge that I gained and practiced from the internship. In particular, I will highlight some main items.

- Python programming: My implementation involves a lot of python programming. Due to the internship, I am now very skillful in this domain and I can handle it independently.
- Linux: Another tool that I have been using a lot during the internship is Linux. Specifically, I used it to access a remote server by SSH to run simulations and wrote a few bash scripts to automate my experiments.

I started my internship by investigating the state-of-the-art of protocol sequences. By doing so, I understood the mathematical structure of these sequences and got an idea of how to manipulate them.

I would like to also point out that what is most important is not only the knowledge itself but the way how to learn and master it. I believe that with this ability I can always keep pace to catch the latest technology and science no matter how fast it develops.

11.3 Theoretical Knowledge

In terms of theoretical knowledge, I acquired it, especially through paper reading and discussions with my supervisors and colleagues.

- Reinforcement learning
- Implementation aspects using Stable Baselines3
- Strong ability to be autonomous

11.4 Plan for the Next 3 Years

Before starting the internship, I was hesitant to pursue a possible Ph.D. program. However, after the research work at Bell Labs, I am more convinced that pursuing a Ph.D. program would be a great choice. Moreover, by doing firstly my study at a French engineering school and then taking a research-oriented internship, I become more well prepared and all around. I believe this could help me a lot when deciding on my future career.

References

[1] J. Massey and P. Mathys, "The collision channel without feedback," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 192–204, 1985.

[2] C. S. Chen, W. S. Wong, and Y.-Q. Song, "The design and analysis of protocol sequences for robust wireless accessing," in *IEEE Global Telecommunications Conference*, pp. 3666–3671, 2007.

[3] C. S. Chen, V. M. Nguyen, and L. Thomas, "On small cell network deployment: A comparative study of random and grid topologies," in *IEEE Vehicular Technology Conference*, 2012.

[4] C. S. Chen, *Resource Management in Wireless Multimedia Systems.* PhD thesis, The Chinese University of Hong Kong, 2005.

[5] W. S. Wong, "Transmission sequence design and allocation for wide-area ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 869–878, 2014.

[6] L. Salaün, C. S. Chen, Y. Chen, and W. S. Wong, "Constant delivery delay protocol sequences for the collision channel without feedback," in *19th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 429–434, 2016.

[7] C. Dumas, L. Salaün, I. Hmedoush, C. Adjih, and C. S. Chen, "Design of coded slotted ALOHA with interference cancellation errors," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 12742–12757, 2021.

[8] E. Nisioti and N. Thomos, "Decentralized reinforcement learning based MAC optimization," in *IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–5, 2018.

[9] I. Ayoub, I. Hmedoush, C. Adjih, K. Khawam, and S. Lahoud, "Deep-IRSA: A deep reinforcement learning approach to irregular repetition slotted ALOHA," in *10th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks*, pp. 1–6, 2021.

[10] K. W. Shum, C. S. Chen, C. W. Sung, and W. S. Wong, "Shift-invariant protocol sequences for the collision channel without feedback," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3312–3322, 2009.

[11] Y. Shao, S. C. Liew, and T. Wang, "AlphaSeq: Sequence discovery with deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3319–3333, 2020.

[12] C. S. Chen, W. S. Wong, and Y.-Q. Song, "Constructions of robust protocol sequences for wireless sensor and ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 3053–3063, 2008.

[13] Y. Zhang, K. W. Shum, and W. S. Wong, "On pairwise shift-invariant protocol sequences," *IEEE Communications Letters*, vol. 13, no. 6, pp. 453–455, 2009.

[14] C. S. Chen and W. S. Wong, "A robust access protocol for wireless sensor networks," in *IEEE Military Communications Conference (MILCOM)*, 2006.

[15] C. S. Chen, K. W. Shum, C. W. Sung, W. S. Wong, and G. E. Oien, "User unsuppressible protocol sequences for collision channel without feedback," in *International Symposium on Information Theory and Its Applications*, pp. 1–6, 2008.

[16] K. W. Shum, W. S. Wong, C. W. Sung, and C. S. Chen, "Design and construction of protocol sequences: Shift invariance and user irrepressibility," in *IEEE International Symposium on Information Theory*, pp. 1368–1372, 2009.

[17] K. W. Shum, W. S. Wong, and C. S. Chen, "A general upper bound on the size of constant-weight conflict-avoiding codes," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3265–3276, 2010.

[18] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, "Stable Baselines3," 2019.

[19] H. Yu, Y. Kang, Z. Shi, Y. Shao, Y. Lin, and Y. Zhang, "Design of deterministic grant-free access with deep reinforcement learning," in *IEEE 20th International Conference on Communication Technology (ICCT)*, pp. 944–948, 2020.

[20] L. Huang, H. Zhang, R. Li, Y. Ge, and J. Wang, "AI coding: Learning to construct error correction codes," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 26–39, 2020.

[21] Y. Liao, S. A. Hashemi, J. M. Cioffi, and A. Goldsmith, "Construction of polar codes with reinforcement learning," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 185–198, 2022.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR arXiv*, 2017.

[23] J. Hare, "Dealing with sparse rewards in reinforcement learning," *CoRR arXiv*, 2019.