Reinforcement Learning for Reactive Jamming Mitigation

Marc Lichtman and Jeffrey H. Reed

Wireless @ Virginia Tech, Virginia Tech, Blacksburg, VA, USA;
e-mail: {marcll, reedjh}@vt.edu

Received 25 March 2014; Accepted 25 March 2014;
Publication 2 July 2014

Abstract

In this paper, we propose a strategy to avoid or mitigate reactive forms of jamming using a reinforcement learning approach. The mitigation strategy focuses on finding an effective channel hopping and idling pattern to maximize link throughput. Thus, the strategy is well-suited for frequency-hopping spread spectrum systems, and best performs in tandem with a channel selection algorithm. By using a learning approach, there is no need to pre-program a radio with specific anti-jam strategies and the problem of having to classify jammers is avoided. Instead the specific anti-jam strategy is learned in real time and in the presence of the jammer.

Keywords: Reactive jamming, reinforcement learning, Markov decision process, repeater jamming, Q-learning.

1 Introduction

Wireless communication systems are becoming more prevalent because of their affordability and ease of deployment. Unfortunately, all wireless communications are susceptible to jamming. Jamming attacks can degrade communications and even cause total denial of service to multiple users of a system. As communications technology becomes more sophisticated, so does the sophistication of jammers. Even though jamming techniques such as

Journal of Cyber Security, Vol. 3 No. 2, 213–230.
doi: 10.13052/jcsm2245-1439.325

repeater jamming have been known for decades [2], recently there has been research into other forms of complex jamming with receiving and processing capabilities (reactive jamming) [8].

In this paper, we propose a strategy to mitigate or even avoid these reactive forms of jamming using a reinforcement learning (RL) approach. Through a learning approach, the problem of having to detect and classify which type of jammer is present in real time is avoided. In addition, there is no need to preprogram a radio with specific mitigation strategies; instead the strategy is learned in real time and in the presence of the jammer. The proposed mitigation strategy focuses on finding an effective channel hopping and idling pattern to maximize link throughput. Not only can this approach enable communications, which would otherwise fail in the presence of a sophisticated and reactive jammer, it can also act as an optimization routine that controls the link layer behavior of the radio.

The proposed strategy is well suited for a frequency-hopping spread spectrum (FHSS) system, which are widely used in modern wireless communications. The strategy could also be applied to an orthogonal frequency-division multiple access (OFDMA) system in which users hop between different subcarriers or groups of subcarriers. Countless users and systems depend on wireless communications and therefore it is important to secure them against jamming. While there exists many methods to counter barrage jamming (the most basic form of jamming), there are few methods that are designed to address the more intelligent behaviors a jammer can exhibit.

2 Related Works

Wireless security threats are typically broken up into two categories: cybersecurity and electronic warfare (i.e. jamming). Electronic warfare attacks target the PHY and/or MAC layer of a communication system, while cybersecurity attacks are designed to exploit the higher layers. In this paper we are only concerned with jamming, and in particular jamming of an intelligent nature. A series of intelligent jamming attack models are introduced in [8], including the reactive jammer model. The authors propose a basic detection algorithm using statistics related to signal strength and packet delivery ratio. For an overview on electronic warfare and jamming, we refer the reader to [1].

A RL or Markov decision process (MDP) approach has been previously used in the wireless domain for channel assignment [9], general anti-jamming in wireless sensor networks [10], and jammer avoidance in cognitive radio

networks [4, 7]. The authors of [9] apply reinforcement learning to the problem of channel assignment in heterogeneous multicast terrestrial communication systems. While this paper does not deal with jamming, it has similar concepts to the techniques proposed in this paper. The authors of [10] propose an anti-jamming scheme for wireless sensor networks. To address time-varying jamming conditions, the authors formulate the anti-jamming problem of the sensor network as a MDP. It is assumed that there are three possible anti-jamming techniques: transmission power adjustment, error-correcting code, and channel hopping. These techniques are not explored any further; the set of actions available to the radio is simply which technique is used. While this work is similar to the technique described in this paper, it greatly generalizes the anti-jamming strategies. In other words, this work does not offer a jamming strategy, it offers a method of choosing the best jamming strategy from a given set. The authors of [7] use a MDP approach to derive an optimal anti-jam strategy for secondary users in a cognitive radio network. For the jammer model, the authors use reactive jammers seeking to disrupt secondary users and avoid primary users. In terms of actions, in each timeslot the secondary user must decide whether to stay or hop frequencies. The authors propose an online learning strategy for estimating the number of jammers and the access pattern of primary users (this can be thought of as channel availability). Even though the authors use a reactive jammer model similar to the one described in this paper, they assume the jammer is always successful, and the entire analysis is within the context of dynamic spectrum access.

To the best of our knowledge, there have been no RL or MDP based approaches designed to mitigate a wide range of reactive jamming behaviors. This paper will provide initial insights into the feasibility and suitability of such an approach.

3 System Model and Problem Formulation

Consider the typical wireless communications link, with the addition of a jammer that both receives the friendly signal (but not necessarily demodulates it) and transmits a jamming signal, as shown in Figure 1. For the sake of simplicity we will only consider a unidirectional link, although this analysis also applies to bidirectional links, that may be unicast or broadcast, as well as a collection of links.

While reactive jamming can take on different forms, we will broadly define the term as any jammer that is capable of sensing the link and reacting to sensed information. We will assume this sensed information is in the form of

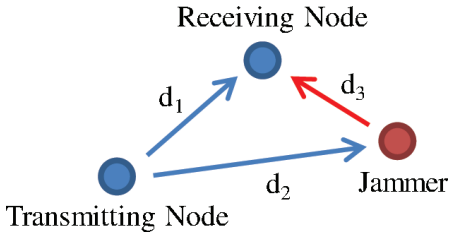

Figure 1 System model of a transmitter, receiver, and reactive jammer

the presence or absence of energy, because any additional information such as modulation scheme or actual data would be irrelevant for this mitigation strategy. A simple example of a reactive jammer is one that senses the spectrum for activity, and immediately transmits wideband noise when it senses any activity [8]. This strategy allows the jammer to remain idle while the channel is idle and thus save power and avoid being easily detected. Another form of reactive jamming, commonly known as repeater or follower jamming [6], works by immediately transmitting what it receives with noise added to it. This can be modeled as a jammer that senses a set of channels, and immediately transmits noise on any channel that appears to be active.

Reactive jamming is only feasible when the geometry of the system is such that the jammer's transmitted signal reaches the target receiver before it hops to a new channel or stops transmitting. As such, reactive jamming is only possible when the jammer is physically located near or between the target transmitter and receiver. If η represents the fraction of each hop duration that must remain not jammed for communications to succeed, then we have the following inequality limiting the distances d_2 and d_3 [6]

$$d_2 + d_3 \leq (\eta T_d - T_j)c + d_1 \tag{1}$$

where T_d is the hop duration, T_j is the jammer's processing time, c is the speed of light, and d_1, d_2, and d_3 are the distances indicated in Figure 1. In addition to this limitation, the jammer-to-signal ratio at the receiving node must be high enough to degrade quality of service. In this paper we assume the jammer is close enough to the transmitter and receiver, and that the jammer-to-signal ratio is significantly high during periods of jamming.

As part of the analysis and simulation we will investigate two specific reactive jamming models. The first, labeled in this paper as simply "reactive jamming", will be defined as a jammer that successfully jams any transmission that remains active for too long, regardless of the channel/frequency in use.

The second jammer model is based on repeater jamming, and it is described as a jammer which successfully jams any transmission that remains on the same channel/frequency for too long. While there are other ways to formulate reactive jamming models, the analysis and simulation in this paper will focus on these two. More formal definitions of these two jammer models is as follows:

- **Reactive Jammer** - Begins jamming any transmission that remains active for more than N_{REACT} time steps, and will only cease jamming once the target is idle for at least N_{IDLE} time steps.
- **Repeater Jammer** - Begins jamming any transmission that remains on the same channel for more than N_{REP} time steps.

In this analysis we will investigate a transmitter and receiver pair that can hop among a certain number of channels using a FHSS approach, or any other approach that involves radios capable of changing channels. Therefore, at any time step, the transmitter has the option to either remain on the channel, change channel, or go idle. Because the actions of the transmitter must be shared with the receiver beforehand, it is expected that decisions are made in advanced.

It is assumed that channel quality indicators (e.g. whether or not the information was received) are sent back to the transmitter on a per-hop basis. These indicators could be binary (indicating an ACK or NACK), or they could take on a range of values indicating the link quality. Lastly, it is assumed that the receiver is not able to simply detect the presence of a jammer.

4 Strategy for Mitigation of Reactive Jamming

The mitigation (a.k.a. anti-jam) strategy described in this paper is based on modeling the system as a MDP, where the transmitter is the decision maker, and using RL to learn a strategy for dealing with the broad category of reactive jamming. This strategy will be in the form of a channel hopping pattern, where going idle is considered as hopping to the "idle channel" for a certain duration. However, we are not concerned with choosing the best channel to transmit on at any given time, nor identifying corrupt channels that have excessive noise. The mitigation strategy described in this paper is designed to work in tandem with this kind of algorithm, i.e. one that indicates which specific channels are suitable for use and which are not. Likewise, we are not concerned with the PHY-layer waveform characteristics that the transmitter or jammer uses (i.e.

bandwidth, modulation, type of noise, etc.). Adaptive modulation and coding can be performed alongside the proposed strategy.

4.1 Reinforcement Learning Background

RL is the subset of machine learning concerned with how an agent should take actions in an environment to maximize some notion of cumulative reward. The agent is the entity interacting with the environment and making decisions at each time interval, and in this paper we will consider the transmitter as the agent (although the actions it chooses must be forwarded to the receiver). An agent must be able to sense some aspect of the environment, and make decisions that affect the agent's state. For example, reinforcement learning can be used to teach a robot how to walk, without explicitly programming the walking action. The robot could be rewarded for achieving movement in a forward direction, and the robot's action at each time step could be defined as a set of angular servo motions. After trying a series of random motions, the robot will eventually learn that a certain pattern of motion leads to moving forward, and thus a high cumulative reward. In this paper, we apply this concept to a transmitter that learns how to hop/idle in a manner that allows successful communications under a sophisticated reactive jamming attack.

There are four main components of a RL system: a policy, reward, value function, and the model of the environment [5]. A policy (denoted as π) defines how the agent will behave at any given time, and the goal of a RL algorithm is to optimize the policy in order to maximize the cumulative reward. A policy should contain a stochastic component, so that the agent tries new actions (known as exploration). A reward, or reward function, maps the current state and action taken by the agent to a value, and is used to indicate when the agent performs desirably. In a communication system, a possible reward function may combine the throughput of a link, spectral efficiency, and power consumption. While the reward function indicates what is desirable in the immediate sense, the value function determines the long-term reward. A state may provide a low immediate reward, but if it leads to other states that provide a high reward, then it would have a high "value".

The model of the environment is used to either predict a reward that has not been experienced yet, or simply determine which actions are possible for a given state. For example, it is possible to create a RL agent that learns how to play chess, and the environment would be a model of the chess board, pieces, and set of legal moves.

In RL, the environment is typically formulated as a MDP, which is a way to model decision making in situations where outcomes are partially random and partially under the control of the decision maker. The probability of each possible next sate, s', given the current state s and action a taken, is given by [5]

$$P_{ss'}^{a} = Pr\{s_{t+1} = s' | s_t = s,\ a_t = a\} \tag{2}$$

Equation 2 provides what are known as transition probabilities, and because they are only based on the current state and action taken, it assumes a memoryless system and therefore has the Markov property. The expected reward (obtained in the next time step) for a certain state-action pair is given by Equation 3. The goal of a learning agent is to estimate these transition probabilities and rewards, while performing actions in an environment.

$$R_{ss'}^{a} = E\{r_{t+1} | s_{t+1} = s',\ s_t = s,\ a_t = a\} \tag{3}$$

In order for an agent to take into account the long-term reward associated with each action in each state, it must be able to predict the expected long-term reward. For a certain policy π, we calculate the expected return from starting in state s and taking action a as [5]

$$Q^{\pi}(s,a) = E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s,\ a_t = a \right\} \tag{4}$$

where γ is known as the discount rate, and represents how strongly future rewards will be taken into account. Equation 4 is known as the action-value function, and in a method known as Q-Learning, the action-value function is estimated based on observations. While performing actions in an environment, the learner updates its estimate of $Q(s_t, a_t)$ as follows [5]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{5}$$

where r_{t+1} is the reward received from taking action a, and α is the learning rate, which determines how quickly old information is replaced with new information. Because Q-Learning is an iterative algorithm, it must be programmed with initial conditions ($Q(s_0, a_0)$). Optimistically high values are typically used for initialization, to promote exploration. However, even once some initial exploration is performed, there needs to be a mechanism that prevents the agent from simply sticking to the best policy at any given

actions apply to all jammer models. time. An approach known as Epsilon-greedy forces the agent to take the "best action" with probability 1–ϵ, and take a random action (using a uniform probability) with probability ϵ. Epsilon is usually set at a fairy high value (e.g. 0.95) so that a majority of the time the agent is using what it thinks is the best action. For an in-depth tutorial on MDPs and RL, we refer the reader to [5].

4.2 Markov Decision Process Formulation

We will now formulate a MDP used to model the transmitter's available states and actions. The states exist on a two dimensional grid, in which one axis corresponds to the time that the transmitter has been on a given channel (including the "idle channel"), and the other axis corresponds to the time the transmitter has been continuously transmitting. Time is discretized into time steps, and we will assume the step size is equal to the smallest period of time in which the transmitter must remain on the same channel. Figure 2 shows the state space and actions available in this MDP.

The transmitter will always start in the top-left state, which corresponds to being idle for one time step. It then must choose whether to remain idle, or "change channel" (which can be interpreted as "start transmitting" when coming from an idle state). If it decides to change channel, then in the next time step it must decide whether to remain on that channel or change to a new channel, which we will assume is chosen randomly from a list of candidate channels. It should be noted that the result of each action is deterministic,

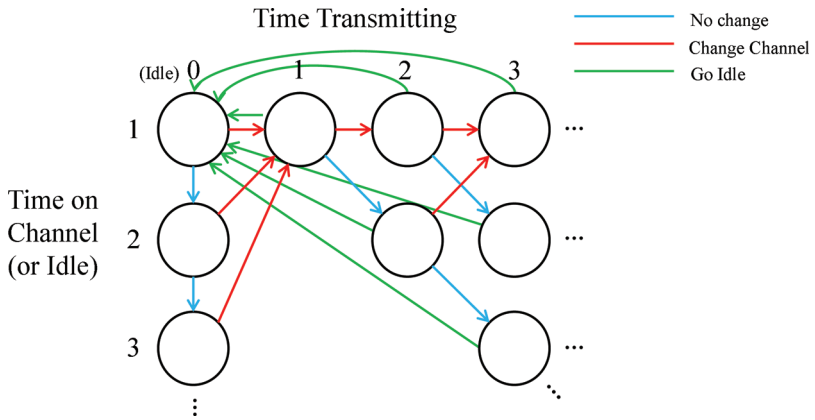

Figure 2 Markov Decision Process associated with hopping channels and going idle. These states and actions apply to all jammer models

however the rewards may contain a stochastic component. Due to what the states represent, the MDP is theoretically infinitely long in directions indicated by ellipsis in Figure 2. However, in practical systems the width and height of the MDP must be limited, as we discuss later.

The reward associated with each state transition is based on the actual data throughput that occurs during the time step. As such, the rewards are based on the jammer model, and may be stochastic in nature. Figure 3 shows the rewards associated with a transmitter and receiver operating in the presence of a reactive jammer with $N_{REACT} = 3$ and $N_{IDLE} = 1$ (model and parameters defined in the previous section). This example shows that when the radio is transmitting for more than three continuous time steps, the link becomes jammed (red states) and the reward becomes zero until the jammer goes idle and then starts transmitting again (the radio is not rewarded while idle). Although the rewards are shown on top of each state, they are actually associated with the previous state and action taken, and won't always be equal for a given resulting state. The numbers 1, 1.3, and 1.47 are examples to demonstrate the fact that remaining on the same channel is more favorable than switching channels, due to the time it takes to switch frequencies. In a

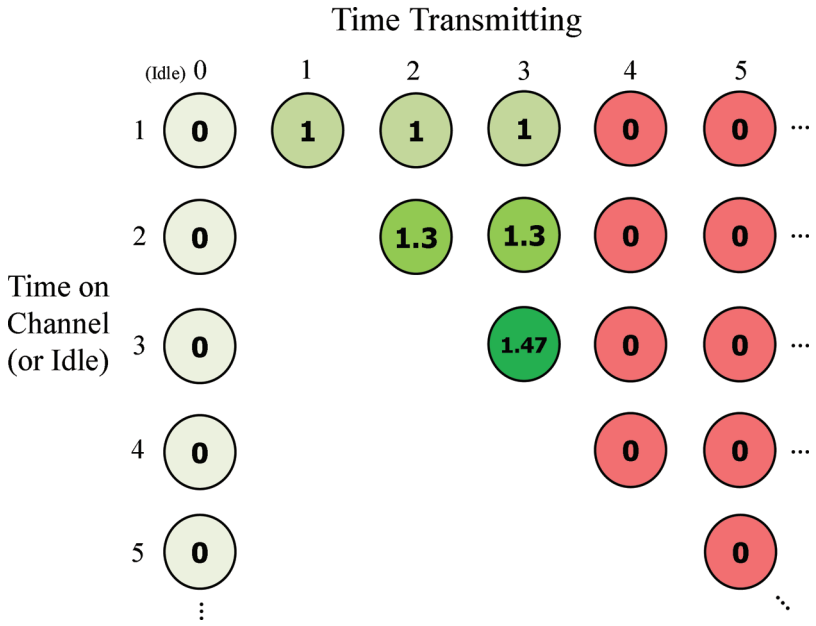

Figure 3 Rewards associated with reactive jammer model $N_{REACT} = 3$

Table 1 Summary of how to cast this mitigation approach into a RL framework

Environment States exist on a two dimensional grid, in which one axis corresponds to the time the transmitter has been on a given channel (including the "idle channel"), and the other axis corresponds to the time the transmitter has been continuously transmitting.
Agent's Actions are to either 1) remain idle or 2) "change channel" which can be interpreted as "start transmitting" when coming from an idle state.
State Transition Rules are deterministic (although a stochastic component due to external factors could be added) and based on the action taken.
Reward Function is a value proportional to the data throughput that occurred during the time step (not known until feedback is received).
Agent's Observations include the state it is currently in, and the reward achieved from each state-action pair.
Exploration vs. Exploitation is achieved using the Epsilon-greedy approach, in which the agent chooses a uniformly random action a small fraction of the time.
Task type is continuing by nature, but could be treated as episodic where each episode is an attempt to transmit for N time steps.

real implementation the reward would be based on the achieved throughput or quality of the link; not a model. A summary of how to cast this mitigation approach into a RL framework is given in Table 1.

Now that the states, actions, and rewards are established, we can investigate the learning process of the transmitter in the presence of various types of reactive jammers. In RL, the agent (in this case, the transmitter) learns by trying actions and building statistics associated with each state-action pair. At the beginning of the learning process, the agent has no information about the environment, and must try random actions in any state. After a period of learning the agent eventually chooses what it thinks is the best action for each state in terms of the predicted long-term reward. The Epsilon-greedy approach forces the agent to never consider itself "finished" with learning.

Under a reactive jammer with a certain N_{REACT} and when $N_{IDLE} = 1$, the optimal policy is to remain on the same channel for N_{REACT} time steps, and then go idle for one time step. Three optimal policies are shown in Figure 4, correspond to $N_{REACT} = 1$, 2, and 3. Each optimal policy resembles a loop that starts at idle for one time step and proceeds to transmit on the same channel for N_{REACT} time steps. In a real-world scenario, it takes the transmitter many

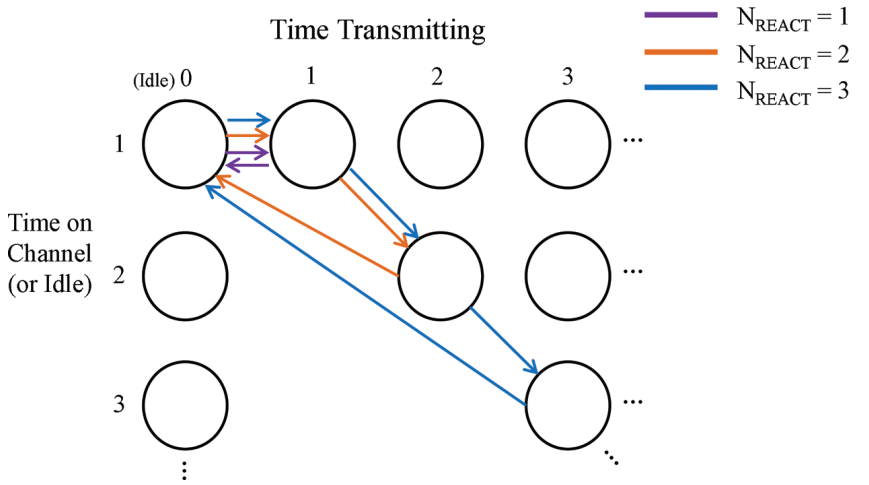

Figure 4 Optimal policies in the presence of three different reactive jammers

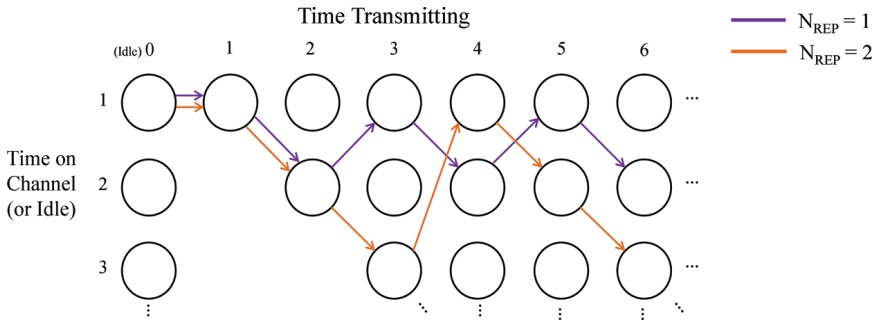

Figure 5 Optimal policies in the presence of two different repeater jammers

time steps to establish that this is the best policy to take, because it must explore each state-action multiple times to build reliable statistics.

The optimal policy for a repeater jammer is shown in Figure 5, using $N_{REP} = 1$ and 2. This zigzag pattern indicates constant-rate frequency hopping, which is well-established as the typical method for avoiding repeater jamming [6]. Unfortunately the optimal policy will always be infinitely long in the horizontal direction. To take this into account, the learning process can involve reseting the current state to the top-left state after a certain period of time continuously transmitting. This will have minimal influence on learning the optimal policy as long as the state space spans enough time steps to take

into account the slowest (i.e. the highest value of N_{REP}) perceivable repeater jammer.

Using the approach described in this paper, there is no need to perform "jammer classification". As such, the mitigation strategy will remain effective over a wide range of jamming behaviors, and may even be able to deal with new jamming behaviors that were not considered during development.

4.3 Knowledge Decay

The last component of the proposed mitigation strategy is taking into account a changing environment. A given jammer may only be present for a short period of time, and link performance would degrade if the transmitter were sticking to its acquired knowledge. As such, the learning engine must incorporate some form of knowledge decay. Due to the nature of Q-Learning, the learning rate α can be used as a form of knowledge decay, by setting it low enough so that the learner can react to a changing environment. A proper value for α would be based on how quick the transmitter is able to learn optimal policies for a variety of jammers. A detailed investigation on approaches of knowledge decay/forgetting is beyond the scope of this paper, but for more information we refer the reader to [5].

4.4 Comparison with Traditional Parameter Optimization

Finding an effective channel hopping and idling pattern in the presence of a reactive jammer could also be performed by optimizing the hopping rate and transmission duty cycle. This can also be thought of as adjusting T_{ON} and T_{OFF}; the transmission and idle time of a transmitter, assuming it hops frequencies after each transmission. This type of approach is often used in cognitive radio [3]. If $T_{OFF} = 0$, then T_{ON} becomes the hopping rate. Any number of optimization approaches could be used to tune these two parameters. However, even though this simpler approach can take into account the two specific jammer models described in this paper, it does not have the flexibility inherent to the RL approach. For example, consider the scenario involving a reactive jammer with $N_{REACT} = 4$, $N_{IDLE} = 1$ and a repeater jammer with $N_{REP} = 1$, both targeting the friendly node simultaneously. The optimal transmission strategy would be to hop channels every time step, but also go idle for one time step after the fourth consecutive transmission (a strategy which is likely not possible with traditional parameter optimization). In addition, if the actual jammer behavior experienced by the

transmitter does not match any models developed during creation of the mitigation strategy, then added flexibility may mean the difference between communicating and being fully jammed.

5 Simulation Results

In this section, we present some simulation results to show proof of concept of our proposed technique. To simulate this RL based mitigation strategy, a link layer simulation framework was created, which included the jammer models described in this paper. Q-learning was chosen as the RL technique [5]. In terms of Q-learning parameters, a learning rate, α, of 0.95 (the transmitter will quickly use learned knowledge) and discount factor, γ, of 0.8 was used for the simulations. This relatively low discount factor was used because of the cyclic nature of the optimal policies. Figure 6 shows the reward over time for various jamming models, depicting the learning process until saturating to an effective policy with constant reward. Because the reward from each time step is proportional to link throughput, the results can be interpreted as throughput over time. The barrage jammer was modeled by causing jamming with 20% probability at each time step, regardless of how long the transmitter has been transmitting or on a given channel. This can be thought of as a nonreactive jammer that is always transmitting, but at a jammer-to-signal ratio that is not quite high enough to cause complete denial of service.

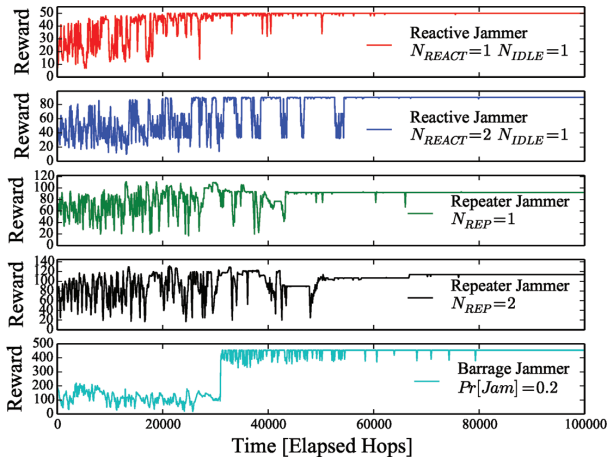

Figure 6 Simulation results showing the learning process over time in the presence of different jammers

The maximum achievable reward under each jamming behavior varies, which is expected (e.g. N_{REACT} = 2 will allow using a higher duty cycle than N_{REACT} = 1). Although not depicted in Figure 6, it should be noted that the transmitter learned the optimal policy (discussed in the previous section) only during the reactive jamming and barrage jamming scenarios. In both repeater jamming scenarios the learned policy did not traverse the entire zigzag pattern on the MDP, which is the optimal policy for the repeater jamming model as discussed earlier. Rather, the transmitter would go idle on occasion, which would essentially reset the zigzag pattern. Hence, the reward achieved under repeater jamming was not the maximum possible reward. Under barrage jamming the optimal policy for the transmitter would be to remain transmitting on the same channel indefinitely, which occurred after around 50,000 time steps, except for the occasional channel hop (as indicated by the small dips in the plot). This demonstrates how the proposed strategy can work under non-reactive jamming, despite not being designed to do so, and even provide better throughput than a constant-rate FHSS approach by avoiding the overhead associated with changing channels.

It should be noted that the time taken to learn an effective strategy for a given jammer is a function of the learning rate parameter and learning technique (Q-learning in this case). Results in Figure 6 show a learning time between 30,000 and 50,000 time steps, which is one or two seconds in a system where the minimum hop duration is on the order of tens of microseconds. While this may seem long compared to a radio that is preprogrammed with specific anti-jam strategies, it is unlikely that the presence of different jammers will change within seconds. In addition, the preprogrammed radio must spend time classifying the type of jammer present in order to know which mitigation scheme to use; a process which is not needed for the proposed strategy. We remind the reader that although wireless channel conditions are known for changing within milliseconds, the proposed strategy is meant to counter the adversary; not traditional channel imperfections such as fading or doppler shift.

6 Conclusions

In this paper, we have developed a RL based strategy that a communication system can use to deal with reactive jammers of varying behavior by learning an effective channel hopping and idling pattern. Simulation results provide a proof of concept and show that a high-reward strategy can be established

within a reasonable period of time (the exact time being dependent on the duration of a time step).

This approach can deal with a wide range of jamming behaviors, not known a priori. Without needing to be preprogrammed with anti-jam strategies for a list of jammers, our approach is able to better adapt to the environment. The proposed technique is best used in tandem with an algorithm that finds a favorable subset of channels to use, as well as modern optimization techniques such as adaptive modulation and forward error correction. In future work we will investigate expanding the MDP state space to take into account additional factors, as well as explore more stochastic jammer models. In addition, it is likely that the RL procedure can be tuned to provide even greater performance.

Acknowledgement

This material is based on research sponsored by the Air Force Research Laboratory under agreement number FA9453-13-1-0237. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

Refernces

[1] David L Adamy. *EW 101*. Artech House, 2001.

[2] Joseph Aubrey Boyd, Donald B Harris, Donald D King, and HW Welch Jr. Electronic countermeasures. *Electronic Countermeasures*, 1, 1978.

[3] S.M. Dudley, W.C. Headley, M. Lichtman, E.Y. Imana, Xiaofu Ma, M. Abdelbar, A. Padaki, A. Ullah, M.M. Sohul, Taeyoung Yang, and J.H. Reed. Practical issues for spectrum management with cognitive radios. *Proceedings of the IEEE*, 102(3): 242–264, March 2014.

[4] Shabnam Sodagari and T Charles Clancy. An anti-jamming strategy for channel access in cognitive radio networks. In *Decision and Game Theory for Security*, pages 34–43. Springer, 2011.

[5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

[6] Don J Torrieri. Fundamental limitations on repeater jamming of frequency-hopping communications. *Selected Areas in Communications, IEEE Journal on*, 7(4): 569–575,

[7] Yongle Wu, Beibei Wang, and KJ Ray Liu. Optimal defense against jamming attacks in cognitive radio networks using the markov decision process approach. In *IEEE Global Telecommunications Conference 2010*, pages 1–5. IEEE, 2010.

[8] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57. ACM, 2005.

[9] Mengfei Yang and David Grace. Cognitive radio with reinforcement learning applied to heterogeneous multicast terrestrial communication systems. In *Cognitive Radio Oriented Wireless Networks and Communications, 2009,* pages 1-–6. IEEE, 2009.

[10] Yanmin Zhu, Xiangpeng Li, and Bo Li. Optimal adaptive antijamming in wireless sensor networks. *International Journal of Distributed Sensor Networks,* 2012, 2012.

Biographies

Marc Lichtman is a Ph.D. student at Virginia Tech under the advisement of Dr. Jeffrey H. Reed. His research is focused on designing anti-jam approaches against sophisticated jammers, using machine learning techniques. He is also interested in analyzing the vulnerability of LTE to jamming. Mr. Lichtman received his B.S. and M.S. in Electrical Engineering at Virginia Tech in 2011 and 2012 respectively.

Jeffrey H. Reed currently serves as Director of Wireless @ Virginia Tech. He is the Founding Faculty member of the Ted and Karyn Hume Center for National Security and Technology and served as its interim Director when founded in 2010. His book, Software Radio: A Modern Approach to Radio Design was published by Prentice Hall. He is co-founder of Cognitive Radio Technologies (CRT), a company commercializing of the cognitive radio technologies; Allied Communications, a company developing technologies for embedded systems. In 2005, Dr. Reed became Fellow to the IEEE for contributions to software radio and communications signal processing and for leadership in engineering education. He is also a Distinguished Lecture for the

IEEE Vehicular Technology Society. In 2013 he was awarded the International Achievement Award by the Wireless Innovations Forum. In 2012 he served on the Presidents Council of Advisors of Science and Technology Working Group that examine ways to transition federal spectrum to allow commercial use and improve economic activity.