Competing Cognitive Tactical Networks

Siamak Dastangoo, Carl E. Fossa, and Youngjune L. Gwon

A cognitive-based strategy for transmission channel access addresses the need for an approach that is spectrally agile and efficient under scarce resources and that is resilient to adversarial threats. The strategy's five key components—sensing, learning, optimizing, transmitting, and jamming—are combined to model a tactical communications network, and game-theoretic algorithms and a performance metric are employed in a hypothetical blue-force versus red-force competition for spectrum resources.

Tactical communication networks generally operate in highly congested and contested environments. Traditionally, tactical communication radios and devices are statically configured to operate within a preallocated portions of the electromagnetic (EM) radio-frequency (RF) spectrum prior to deployment. This approach requires minimal coordination among the radios and reduces interference by other devices; however, it limits the radios' ability to take advantage of other unused spectrum resources and to avoid unintentional jamming by friendly forces or intentional jamming by adversaries. The capability of radios to dynamically search for, learn, and access available spectrum resources can greatly improve network performance in a congested environment.

Independently from communication radios, friendly electronic warfare (EW) devices have to continuously search for and suppress adversaries' communications and jamming elements while avoiding unintentional jamming of friendly radios. The ability of friendly jamming devices to dynamically sense, learn, and observe spectral activities can lead to more effective jamming strategies and can help minimize self-jamming of friendly radios, especially during missions in which both assets have to coexist. Thus, it is imperative that future communication radios and EW devices be able to dynamically sense, classify, and coordinate access to the EM spectrum for more improved performance under congestion and more robust operations under contention (Figure 1).

Before expanding upon strategies for spectrum sharing in the tactical domain, we would like to highlight ongoing spectrum-sharing efforts in the commercial sector.

Motivated by the growing demand in wireless devices and services and by the shortage of spectrum, the Federal Communications Commission proposed and ratified new policies for dynamic spectrum access (DSA) that allow devices dubbed as secondary users to opportunistically share certain frequency bands with the primary users authorized by licensed service providers as long as the interference caused by the former is limited to an acceptable level [1]. The departure from the static assignment of frequency to a dynamic sharing can provide many benefits, such as higher data rates for wireless services, increased use of underutilized bands, and congestion relief in overcrowded bands. The DSA policy can be implemented with a set of cognitive algorithms (observe the environment, learn from past and present conditions, develop appropriate strategies, take actions based on those strategies) embedded in the secondary users' communication devices. By sensing the environment, the radio devices can learn the behavior of various users and adaptively change the radio's channel access on the basis of traffic needs and transmission patterns. Two promising applications for DSA in the civilian domain are wireless local-area networks (WLANs) and infrastructure-based cellular systems.

The extension of DSA to tactical radios and EW devices operating in environments in which little infrastructure or adversaries' electronic attacks exist is challenging because of the lack of sensing and the lack of intercommunication among EW radios, and requires significant enhancements to protocols and algorithms. Lincoln Laboratory's research into extending DSA is cast as a blue-force versus red-force scenario in which the red force represents an adversary. The blue-force communication radios and blue-force jamming devices dynamically compete with red-force communication radios and jammers on a set of open spectrum channels as depicted in Figure 2. The blue-force network's objective is to adopt a channel access strategy to jointly achieve

- high-data-rate communication among blue-force radios,
- suppression of adversaries' attempts to communicate, and
- resilience of blue-force communicating radios when subjected to attack from red-force jamming.

System Architecture

Because legacy tactical communication radios have no sensing capabilities to make environmental observa-

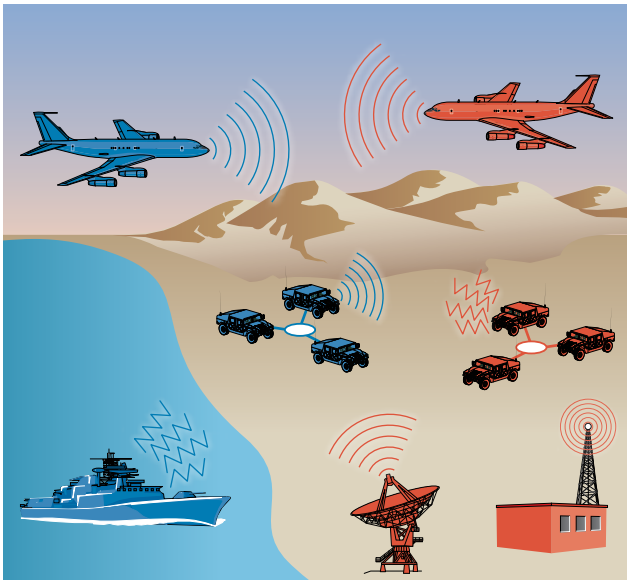

FIGURE 1. Tactical operational scenario in which both friendly (blue) and adversary (red) networks operate. Both networks compete for open spectrum resources.

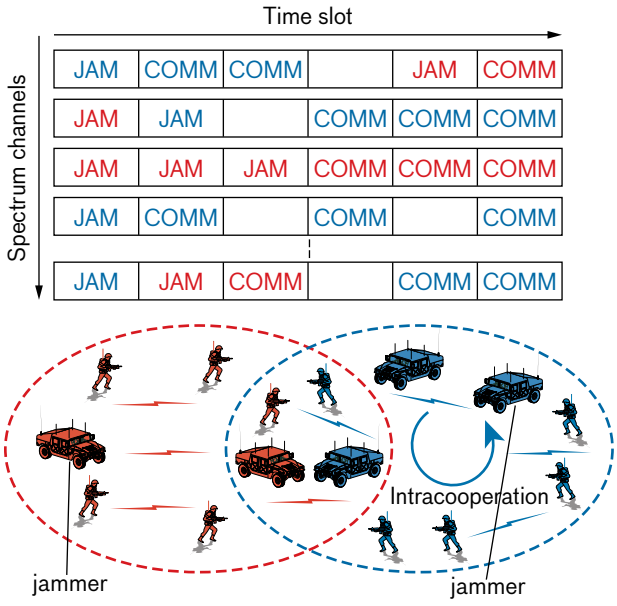

FIGURE 2. A set of blue-force communication radios and jammers compete with their red-force counterparts for a set of open spectrum channels. Spectrum resources are modeled as a set of frequency channels and time slots. Each time-frequency block represents a transmission (or jamming) opportunity for the respective networks. The blue-force nodes may cooperate in exchanging sensing information and coordinate their intent to transmit or jam.

tions, their ability to switch to different spectrum channels when the current ones are jammed or congested is limited. To address this limitation, we propose to include a sensing component in the radio architecture. A sophisticated sensing mechanism that can observe channel activities should be able to determine and classify the users' characteristics. Using the classification of the spectrum occupants constructed by the sensor function, a learning and strategy mechanism can predict the potential rewards and risks for utilizing certain channels. The rewards and risks associated with accessing channels will influence a scheduler's decision to access or not access channels per users' needs. Users may wish to transmit data or suppress (jam) the channel when adversaries transmit. The major components of a blue-force communication node are illustrated in Figure 3. (Communications entities will hereafter be referred to as comm nodes.) The red-force network may or may not be equipped with the same capabilities.

To develop an approach for the blue-force system, it is helpful to introduce the notion of a competing cognitive tactical network (CCTN) in which a network of comm nodes and jammers attempts to dominate access to an open spectrum against a hostile opponent (possibly another CCTN). We pose two compatible but distinct views for our CCTN problems, *state-agnostic* and *state-aware*, and examine both comparatively. The proposed analytical framework for *competing* networks can leverage their capability to jam their opponent by jointly coordinating with communication activities of their own.

Past research approaches have been limited to an antijamming defense strategy for minimizing adversarial attacks, as studied, for example, by Wang et al. [2]. We have devised a Bayesian setting to explore and exploit a multichannel spectrum for the CCTN nodes to achieve optimal strategies for taking appropriate actions (communicate or jam), and we have empirically validated this setting's superior performance over existing methods. The CCTN assumes little or no fixed infrastructural support. A mobile ad hoc network (MANET) would be the most convincing network model; therefore, the network-wide cooperation and strategic use of jamming against the opponents are essential components in designing a winning media-access scheme. A competing network can adopt a centralized control model in which the node actions are coordinated through a singular entity that makes coherent, network-wide decisions. On the other hand, a distributed control model allows each node to decide its own action.

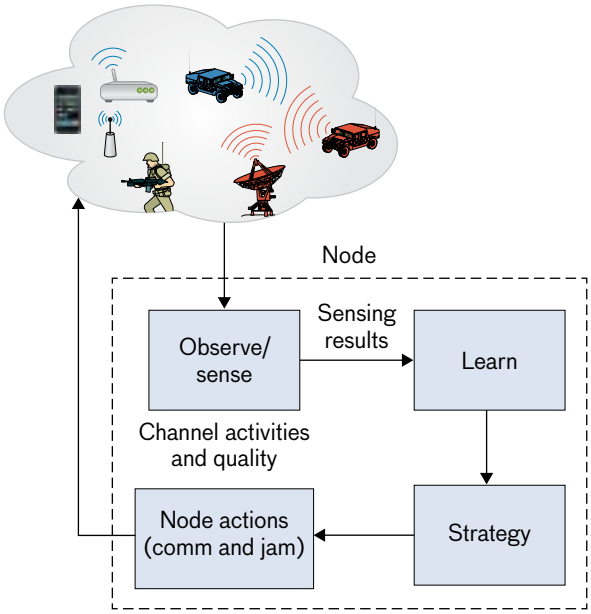

FIGURE 3. Major components of a competing cognitive tactical node. The blue-force nodes (comm or jammer) are equipped with sensing capability (e.g., energy detector, matched-filter detector, cyclostationary detector, etc.), as well as machine learning–based classification techniques to characterize the channel occupancy status. Using the channel status, a learning mechanism can infer the potential reward and risk of using a particular channel. Subsequently, a comm (or jammer) node adopts a strategy that can offer the best reward in the long run. The options for control and dissemination of information within the blue-force components will be addressed later in this article.

Communication Model

The spectrum for open access is partitioned in time and frequency. There are N nonoverlapping channels located at the center frequency f_i (MHz) with bandwidth B_i (Hz) for $i = 1, \ldots, N$. A transmission opportunity is represented by a tuple $< f_i , B_i , t , T >$, which designates a time-frequency slot at channel i and time t with time duration T (msec) as depicted in Figure 4. A simple sensing mechanism may operate like a carrier sense multiple access (CSMA) scheme in which comm nodes first sense before transmitting in a slot of opportunity. A more sophisticated sensing mechanism can lead to finer characterization of the spectral usage as described later.

In order to coordinate a coherent spectrum access and jamming strategy network-wide, we assume that the nodes (both comm and jammers) exchange necessary information via control messages. The channels used to exchange control messages are called *control channels*, and *data channels* are used to transport regular data packets. The DSA approach in Wang et al. [2] is followed; in this approach, control or data channels are dynamically determined and allocated. When a network finds all of its control channels blocked (e.g., because of jamming) at time t, the spectrum access at time $t + 1$ will be uncoordinated.

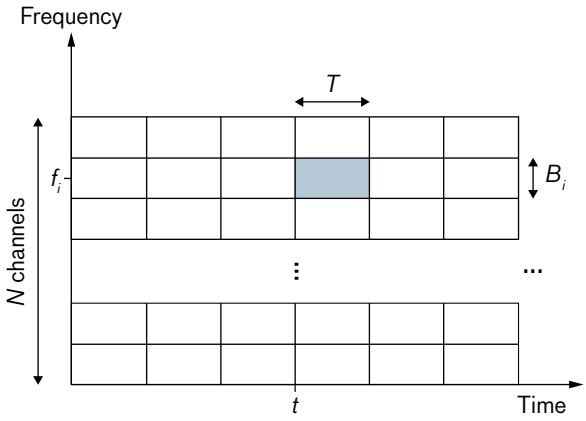

FIGURE 4. Transmission opportunity < f_i , B_i , t , T > (shaded region).

Sensing Model

One of the key prerequisites of a cognitive-based system is the ability to sense its surrounding environment and differentiate among the various users. Conventional detection methods include energy detection, matched-filter detection, and cyclostationary detection, as shown in Figure 5. Sensing and discriminating among blue-force signals are made easier because their signal coding is known. Conversely, the red-force signals are more difficult to sense because of their unknown signatures. Detection and classification of multiple users occupying the same frequency channel can be very challenging, especially because of the unknown nature of the adversary's transmission signatures. The blue-force network, however, can have multiple users occupying a single channel, provided that they are segregated in the so-called coding space. For example, in a code-division multiple-access (CDMA) setting, several users generally spread their energy over a wider bandwidth, using pseudosequences to simultaneously transmit on a channel.

One method of detecting these "coding space" users is to use a matched filter that corresponds to the known spreading codes. Techniques based on synchronous CDMA allow for the use of orthogonal codes; users can completely neglect the presence of one another. Unfortunately, these signals do not have good correlation properties with noise and require perfect synchronization with the transmitter. To reduce the complexity of this design, we only consider asynchronous CDMA. Spreading sequences with good correlation properties can maintain orthogonality among different users, allowing them to be detected asynchronously [3]. The greater the number of users, the lower the effective signal-to-noise ratio (SNR) becomes. More sophisticated schemes using multiuser detection (MUD) can improve performance [4].

The sensing mechanism employed in the blue-force side of the CCTN assumes the ability of the radios and the jamming devices to be able to distinguish among its own transmissions and those of the red force. The detail of this mechanism is beyond the scope of this article. The logical flowchart of the sensing mechanism is illustrated in Figure 5. The combined detection algorithm is described as follows. First, spectral samples are processed by the energy detector to determine if a signal is present. If the energy is below a given threshold, which is adaptively determined by a method of moments estimator (MOME), the channel is considered empty and thus available for use [5]. Conversely, if there is sufficient energy above the threshold value in the band, a matched filter then demodulates the signal using the known parameters of the blue-force comm and jammer to determine if either of these signals is present. If a blue-force comm signal is detected, the CDMA multiuser detector checks for each known blue-force comm code and distinguishes which user is transmitting. The remaining scenario—that is, power was received above the noise threshold but is not a known friendly signal—assumes the existence of a red-force signal.

One should note that Figure 5 describes only the logical flow that was discussed, not the ideal implementation. All of the described methods operate on the original time-domain samples of the received signal and, thus, can be computed in parallel. The possibility of detectors producing conflicting results is handled

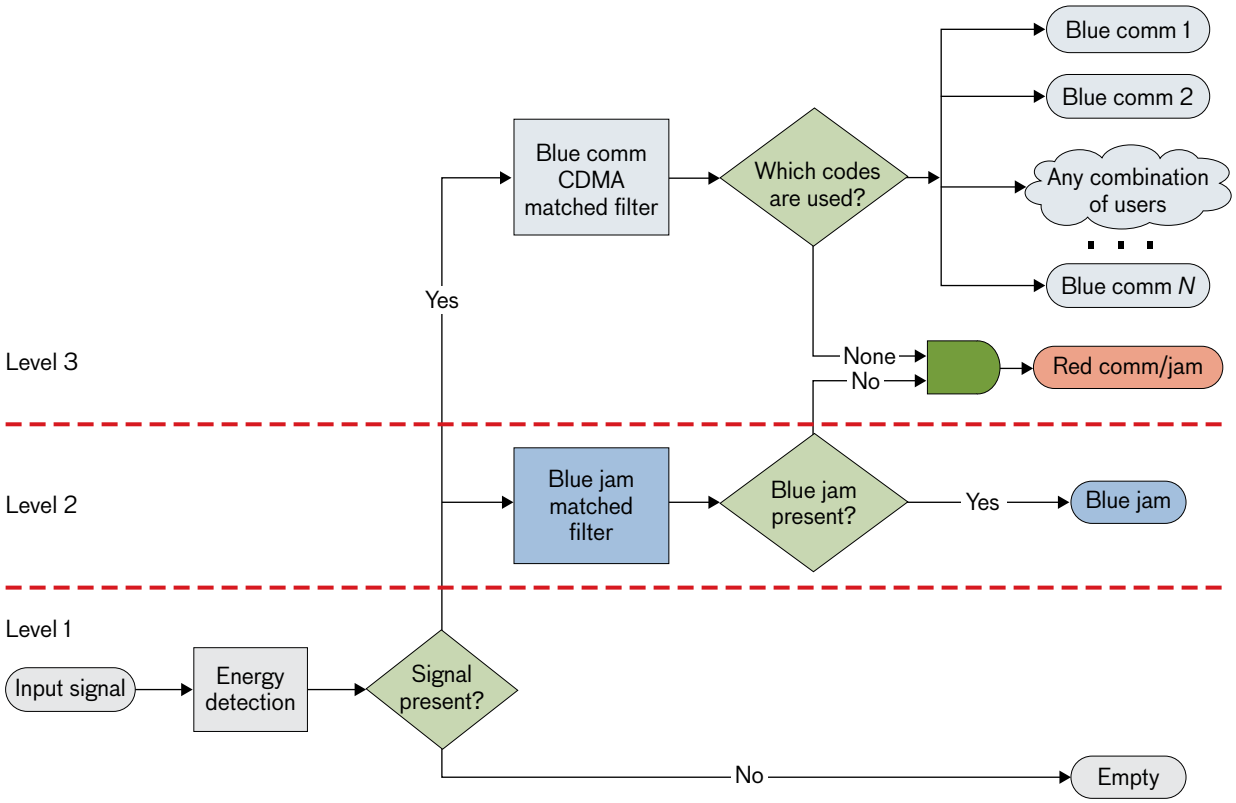

FIGURE 5. Logic flowchart of the sensing decision mechanism. The multiple levels of the sensing mechanism provide a finer profile of the channel occupants. Level 1 determines whether a particular channel is active or not. When the channel is active, the sensing mechanism proceeds to distinguish between a blue-force jam signal and a blue-force comm signal at level 2. Further classification and partitioning of the detected signals into the corresponding blue-force comm users or red-force users are achieved at level 3. Further classification of the red-force signal into the comm and jam components may require knowledge of the red-force waveforms.

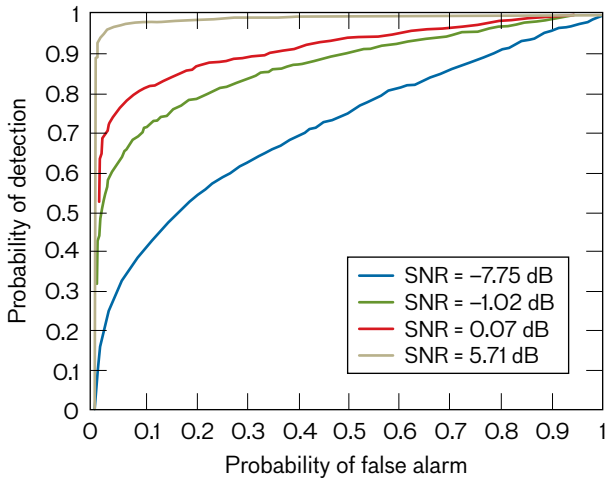

FIGURE 6. A sample receiver operating characteristic (ROC) curve for the blue-force sensing mechanism. The desired location on a ROC curve is upper-left (100% detection with no false alarms).

by the introduction of tertiary states (as opposed to binary), in which the sensor reports a signal as being strong, weak, or completely absent. One can forward this information to a cognitive algorithm to provide greater resolution of the current state and express a discrete level of uncertainty in the current decision. This logic is described in Table 1.

The performance and sensitivity of a sensing scheme can be determined through the receiver operating characteristic (ROC) curve. The ROC curve for the blue-force network is difficult to derive analytically. Figure 6 shows empirical ROC curves of simulated data at various SNRs. The calculations were performed with the implementation of the sensing decision tree and its constituent detection algorithms in MATLAB. The ROC values are instrumental in developing cognitive-based channel access strategies. The values of the ROC curves are further

Table 1. Truth Table for Resolving Conflicts Between Detection Algorithms ("1" Indicates a Positive Detection)

ENERGY DETECTOR (MTM*)	MATCHED FILTER (BLUE COMM)	MATCHED FILTER (BLUE JAM)	DECISION
0	0	0	Empty
0	0	1	Weak blue jam
0	1	0	Weak blue comm
0	1	1	Weak blue comm and jam
1	0	0	Red
1	0	1	Strong blue jam
1	1	0	Strong blue comm
1	1	1	Strong blue comm and jam

* MTM stands for multitaper method

used by the learning and strategy functions of the CCTN to take appropriate actions.

Channel Access Models

There are three types of channel access models. Conventional multiple-access radio systems are based on a *static* channel access model. For example, frequency-division multiple access (FDMA) is a static channel access scheme to assign a particular frequency band to each node. We consider a similar static channel access model in which each radio node is configured to a fixed channel that remains the same throughout. Note that static channel access is a noncognitive channel access.

Another class of multiple-access systems considers random-channel access such as ALOHA [6]. In a random-channel access model, a node decides on a different channel each time it starts to transmit; the node draws from the spectrum composed of N total channels. Random-channel access, however, is still considered a noncognitive channel access scheme because randomization is done in the absence of sensing and cognition.

Lastly, we consider a cognitive channel access model. In particular, we take state-agnostic and state-aware approaches for CCTN. Our state-agnostic approach is modeled after the "multiarmed bandit" (MAB) scheme for sequential resource-allocation problems [7], and our state-aware approach is based on Markov games [8]. The state-agnostic approach is solely driven by channel sensing and learning from sequentially inferring channel rewards and does not keep track of the system state. The state-aware approach explicitly defines a set of discrete system states and provides a plausible means to compute them via reinforcement Q-learning [9], a popular algorithm in contemporary machine learning. A Markov decision process (MDP) underlies the state-aware CCTN.

Table 2 summarizes the interactive criteria that have been evaluated for this article. As the development of cognitive strategies is the main technical objective, we run our Q-learning and MAB-based strategies with the blue-force network against noncognitive static and random strategies with the red-force network. Next, we run one cognitive strategy against the other by running Q-learning with the blue-force network and MAB with the red-force network.

Jamming Model

Xu et al. [10] describe a sound taxonomy of red-force jamming. A *static* jammer continuously dissipates power into a channel selected for transmitting arbitrary waveforms. A *deceptive* jammer can instead send junk bits encapsulated in a legitimate packet format to conceal its intent to disrupt comm nodes. A *random* jammer alternates between jamming and remaining quiet for random time intervals. The randomization can also take place over channel selection, and the jammer can randomly choose a channel to jam. A *reactive*

jammer listens to a channel, stays quiet when the channel is idle, and starts transmitting upon sensing an activity.

Cognition and intelligence allow a more effective jamming strategy. We use the term strategic jamming to extend the statistical jamming described in Pajic and Mangharam [11]. A strategic jammer operates on knowledge obtained from past jamming action and outcomes, as well as any observed (non-random) media access patterns. Strategic jamming can operate for long periods without being detected, causing significantly more damage than existing jamming methods.

Table 2. Channel Access Models and Evaluation Plan

RED FORCE \ BLUE FORCE	Q-LEARNING	MAB
Static	Yes	Yes
Random	Yes	Yes
MAB	Yes	

Network Control Model

The network control model dictates how CCTN node actions at each time slot are determined. Our research considers two different control models: centralized and distributed. Under the centralized control model, a sole decision maker (agent) computes actions for all comm nodes and jammers that want to act on channels for a given time slot. The central decision maker employs a strategy (e.g., Q-learning, MAB) to compute the actions. Under the distributed control model, each CCTN node determines its own action. The lack of centralized decision making can cause conflicting node actions within the same network. Despite this disadvantage, distributed control is popular in tactical MANETs because it is more robust. The networking performance of distributed control can be improved by imposing an extra best-effort protocol to resolve conflicting actions of the nodes.

Figure 7a illustrates the CCTN with a central decision maker computing the network-wide strategy and disseminating *all* node actions. It is assumed for the centralized control that the decision maker should be able to collect sensing results from each node and the exact outcome of every action in order to make sound decisions over time.

Figure 7b illustrates distributed decision making. Here, each node makes its own decision on the basis of the best information that is collected on its node. In contrast, the centralized control model requires tight intranetwork

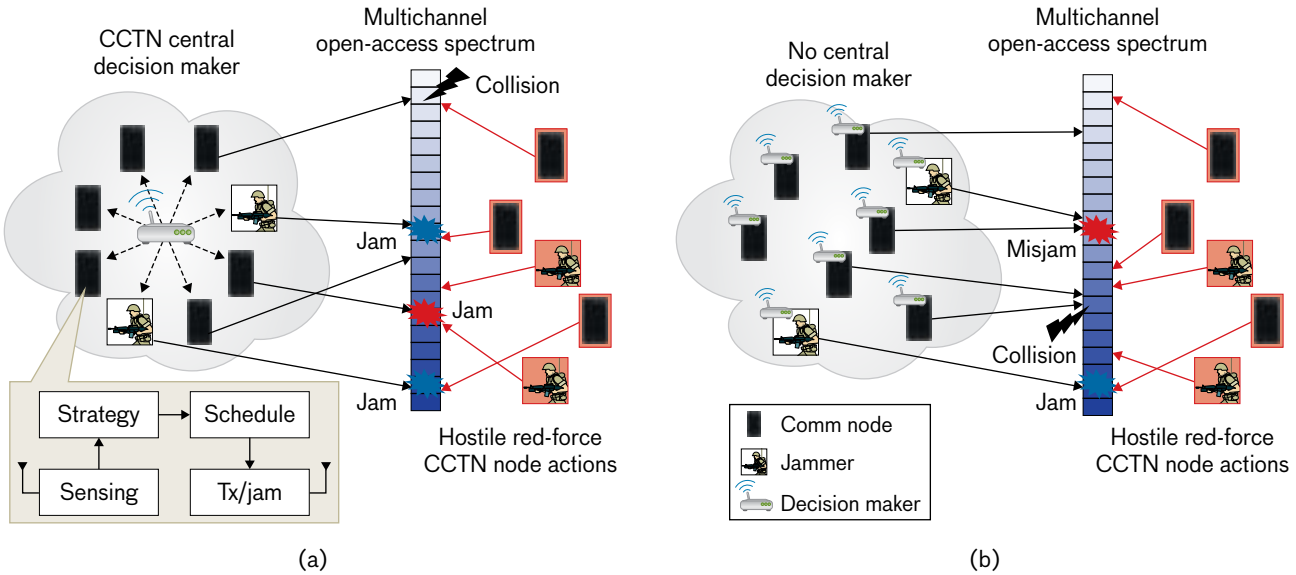

FIGURE 7. CCTN control model. In (a), the blue-force network on the left coordinates its node actions through a central decision maker. In (b), each node in the blue-force network on the left computes its own action in a distributed manner. In both figures, the red-force network on the right is assumed to be an arbitrary control mechanism.

communication to collect information and disseminate the strategy. After taking its action, a node under distributed control observes the outcome, computes the success (reward), and maintains statistics that can be shared with others in the network.

Mathematical Formulation of CCTN

The mathematical formulation of the blue-force and red-force CCTNs can be developed in two compatible, yet different, frameworks: a state-aware model that explicitly defines and tracks the CCTN states and a state-agnostic model that operates CCTN without any state knowledge. The state-aware CCTN assumes an underlying MDP, whereas the state-agnostic counterpart is purely driven by channel sensing and sequential reward sampling.

State-Aware Q-Learning Strategy

In the Q-learning approach (a model-free reinforcement learning technique), the dynamics of the CCTN are cast in a stochastic game framework [8], which extends the MDP [12] by incorporating an agent (as the game's policy maker) who interacts with an environment possibly containing other agents. Under the centralized control model, CCTN considers one agent per network that computes strategies for all nodes in the network, whereas there are multiple agents (i.e., each node) under the distributed control model. Let tuple $G_{\text{CCTN}} = \langle S, A_B, A_R, R, T \rangle$ describe the blue-force versus red-force channel access game in the CCTNs and their interaction, where S denotes the set of states—how many and which comm and jammed channels are active—and $A_B = \{A_{B,\text{comm}}, A_{B,\text{jam}}\}$ and $A_R = \{A_{R,\text{comm}}, A_{R,\text{jam}}\}$ are the action sets for blue-force and red-force networks. The reward function $R : S \times \prod A_{\{B,R\},\{\text{comm, jam}\}} \rightarrow \mathbb{R}$ maps CCTN node actions to a reward value at a given state. The state transition $T : S \times \prod A_{\{B,R\},\{\text{comm, jam}\}} \rightarrow PD(S)$ is the probability distribution over S. Figure 8 shows a sample MDP state diagram, actions, and transition probabilities.

Consider that the spectrum under competition is partitioned in N channels, each of which can be described by a Markov chain. The action sets break down to include both the comm and jammer actions. Each CCTN has C comm nodes and J jammers. Ideally, we would want the condition $2 \times (C + J) << N$, where N designates channelization of the spectrum. This condition would allow the cognitive strategies to diversify their actions for higher potential

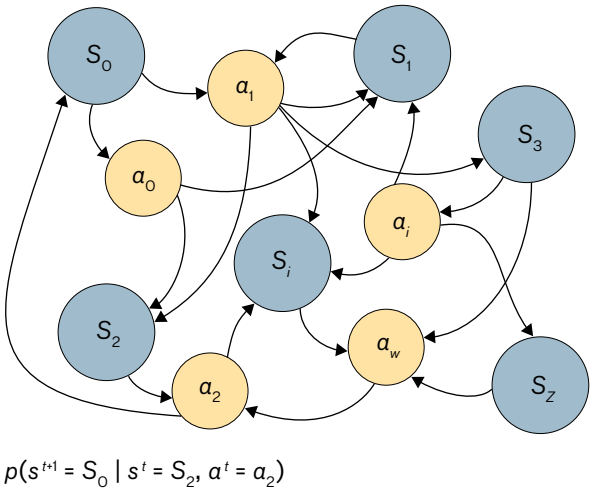

FIGURE 8. An example of a Markov decision process (MDP). The large blue circles represent the MDP states, and the smaller yellow circles represent the actions.

rewards. S, A, and R evolve over time, thus are viewed as functions of time. We use lowercase versions of these letters with superscripted t for their realization in time, e.g., $s^t \in S$ means the CCTN state at time t. The blue-force and red-force network actions at t are $a_B^t = \{a_{B,\text{comm}}^t, a_{B,\text{jam}}^t\}$ and $a_R^t = \{a_{R,\text{comm}}^t, a_{R,\text{jam}}^t\}$ containing both comm and jamming actions, the size-C vectors $a_{B,\text{comm}}^t$ and $a_{R,\text{comm}}^t$, and the size-J $a_{B,\text{jam}}^t$ and $a_{R,\text{jam}}^t$. An ith element in $a_{B,\text{comm}}^t$ designates the channel number that the ith blue-force comm node tries to transmit at t. Similarly, a jth element in $a_{B,\text{jam}}^t$ is the channel that the jth blue-force jammer tries to jam at t. The objective of CCTN is to win in the competition of dominating the spectrum access, which can be achieved by transporting blue-force data bits or jamming red-force data bits. The strategy of the game is defined by $\pi : S \rightarrow PD(S)$, which denotes the probability distribution over the action set. The blue-force network's objective is equivalent to finding the optimum strategy π^* by identifying the maximum value of the total reward

$$\pi^* = \arg\max_{\pi} \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R(s^t, a_B^t, a_R^t)\Big],$$

where $\mathbb{E}$ is the weighted reward over time, and γ is the discount factor for future rewards. The range of values for the discount factor, $0 < \gamma < 1$, allows the decision maker to exploit the resources now or explore more over time for better payoffs.

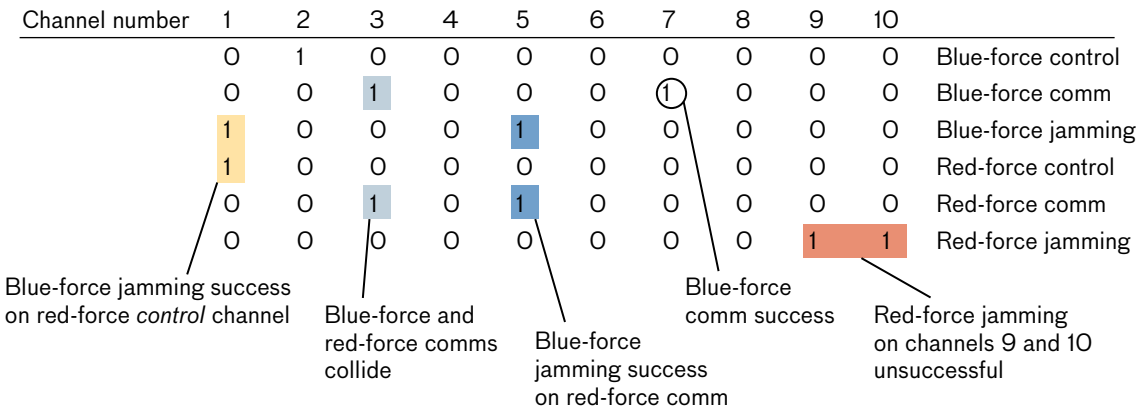

FIGURE 9. CCTN action-state computation example. The yellow band represents a case where the blue-force jammer successfully jams the red-force control channel. The light blue band represents a case where both blue-force and red-force comms collide. The dark blue band represents the case where the blue-force jammer successfully jams the red-force comm transmission on channel 5. Finally, the red band represents a case where red-force jammers unsuccessfully jam channels 9 and 10.

If there are L discrete states for each channel, we must track L^N states for CCTN. Unfortunately, this tracking results in $O(L^N)$, an exponentially complex class with respect to the number of channels. We instead choose a terser state representation $s = \langle I_C, I_D, J_C, J_D \rangle$, where I_C denotes the number of blue-force control channels collided, I_D the number of data channels collided, J_C the number of control channels jammed, and J_D the number of data channels jammed. Given the current state and the action sets of blue-force and red-force nodes, the next state of CCTN is computable. The actions of the opponent are inferred from channel measurements and sensing. For a complete mathematical description of the MDP parameters and transition probabilities and related details, see the article by Gwon et al. [13].

For illustrative purposes, we present an example in which each blue-force and red-force network has $C = 2$ comm nodes and $J = 2$ jammers, and there are $N = 10$ channels in the spectrum. Suppose the channels are numbered 1 through 10. The blue-force node actions at t are $a_B^t = \{a_{B,\text{comm}}^t, a_{B,\text{jam}}^t\}$, where $a_{B,\text{comm}}^t$ and $a_{B,\text{jam}}^t$ are vectors of sizes C and J; similarly for the red-force node actions, $a_R^t = \{a_{R,\text{comm}}^t, a_{R,\text{jam}}^t$. Let $a_{B,\text{comm}}^t = [7,3]$; this means that blue-force comm node 1 transmits in channel 7, and blue-force comm node 2 in channel 3. Let $a_{B,\text{jam}}^t = [1,5]$; that is, blue-force jammer 1 jams channel 1, and blue-force jammer 2 jams channel 5. For the red-force network, let $a_{R,\text{comm}}^t = [3,5]$ and $a_{R,\text{jam}}^t = [10,9]$. Also, the blue-force network uses channel 2 for control, and the red-force control channel is channel 1. These node actions and control channel usages form the bitmap shown in Figure 9; a 1 indicates transmit, jam, or markup as control channel. Both blue-force jammers are successful here, jamming the red-force control and comm data transmissions in channels 1 and 5, respectively. Blue- and red-force comm data transmissions collide in channel 3, and the blue force has a successful data transmission in channel 7. Thus, the red force has no success in either of its comm data channels. Red-force jammers end up unsuccessful, jamming empty channels 9 and 10. This example results in state $s^t = \langle I_C = 0, I_D = 1, J_C = 1, J_D = 1 \rangle$.

JAMMING AND ANTIJAMMING STRATEGIES

The coexistence of the two opposing kinds of signals (i.e., comm and jammer) in blue- and red-force networks decomposes CCTN into two subgames, namely antijamming and jamming games. Figure 10 illustrates the antijamming-jamming relationship among the nodes. In the antijamming game, the blue-force comm nodes strive to maximize their throughput primarily by avoiding hostile jamming from the red-force jammers. Additionally, imperfect coordination within the blue-force network that causes a blue-force jammer to jam its own comm node (i.e., misjamming) should be avoided. Collision avoidance among comm nodes is another objective of the antijamming game. In the jamming game, the blue-force jammers

try to minimize the red-force data throughput by choosing the best channels to jam. A blue-force jammer can target a data channel frequently accessed by the red-force comm nodes or alternatively aims for a red-force control channel, thus resulting in a small immediate reward but a potentially larger value in the future by blocking red-force data traffic. Misjamming avoidance is also an objective for the jamming game. For the blue-force network, the primary means to avoid misjamming is to coordinate the actions of its own jammers. This case is different from that of the antijamming game in which the avoidance is done by coordinating the actions of the blue-force comm nodes.

The quality of the actions chosen by the decision maker is described by the Q function, which is a realization of the Bellman equations [14].[1] A minimax-Q assumes a zero-sum game that implies $Q_B(s^t, a_B^t, a_R^t) = -Q_R(s^t, a_B^t, a_R^t) = Q(s^t, a_B^t, a_R^t)$. This zero-sum action holds tightly for the CCTN jamming subgame in which the jammer's gain is precisely the comm throughput loss of the opponent. In order to solve the antijamming and jamming subgames jointly, we propose a slight modification to the original minimax-Q algorithm in Littman [15]. First, we divide the strategy of the blue force's network π_B into its antijamming and jamming substrategies, π_{B1} and π_{B2}. Then, we add an extra minimax operator to the function $V(s^t)$, describing the value of the particular system state.

$$Q(s^t, a_B^t, a_R^t) = r(s^t, a_B^t, a_R^t) + \gamma \sum_{s^{t+1}} p(s^{t+1} \mid s^t, a_B^t, a_R^t)\, V(s^{t+1})$$

$$V(s^t) = \max_{\pi_{B1}(A_B, \text{comm})} \min_{a_{R,\text{jam}}^t} \max_{\pi_{B2}(A_B, \text{jam})} \min_{a_{R,\text{comm}}^t} \sum_{a_B^t} Q(s^t, a_B^t, a_R^t)\, \pi_B(a_B^t)$$

Two extensions of the Q-learning other than minimax-Q strategy are Nash-Q and friend-or-foe Q (FFQ), which can solve a general-sum game in addition to zero-sum games with an improved convergence in the latter case. Nash-Q makes an important distinction to minimax-Q by requiring one extra term $\hat{\pi}_R(a_R^t)$, which is an estimate of the strategy of the opponent's agent. For CCTN, the blue-force agent needs to learn $\hat{\pi}_{R1}$ and $\hat{\pi}_{R2}$, the antijamming and jamming substrategies of the red-force network. For a general-sum game, the blue-force agent should compute Q_B and Q_R separately while observing its reward $r_B^t = r_B(s^t, a_B^t, a_R^t)$ and estimating the red force's reward r_R^t (see the article by Gwon et al. [13] for complete details).

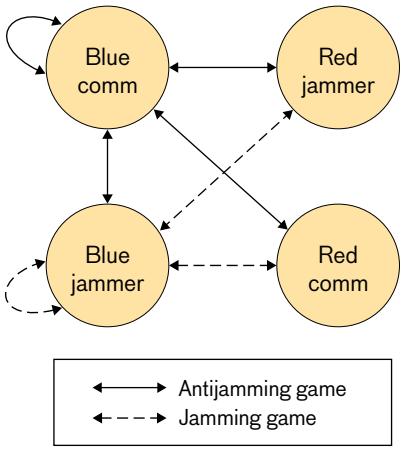

FIGURE 10. Jamming and antijamming relationship. The circles represent all possible node types in the blue force–red force CCTN game. The solid lines represent the antijamming game relationship between nodes in opposing forces and nodes within the same force. For example, the solid line between blue comm and red jammer describes the blue-force comm action to avoid the red jammer. The diagonal solid line describes the blue-force comm action to avoid colliding with the red comm. Similarly, the vertical solid line describes the blue-force comm action to avoid being misjammed by the blue jammer. The solid curved line describes the blue comm actions to avoid colliding with another blue comm (e.g., intranetwork coordination and cooperation). The dashed lines represent the jamming game relationships. The dashed horizontal line describes blue jammer action to disrupt red comm activities. The dashed diagonal line describes blue jammer action to avoid its jamming energy resources on the channel that has already been jammed by the red jammer. Finally, the dashed curved line describes the blue jammers' actions to avoid their jamming energy resources on the same channel that has already been jammed by a blue jammer.

1. The Q function can be described as the quantitative reward as a result of the actions taken. The Bellman equations optimize the *Q* function, which is associated with the values of the MDP states, the *V* function. Evaluating the *Q* function requires the transition probabilities of the MDP, which are difficult to compute for large systems. A more practical approach would be to evaluate the *Q* function iteratively starting with some initial conditions. The values of the *Q* function are later used to derive values for *V*. This interrelation implies optimal actions chosen by the players, using the Bellman optimization in $Q(s, a) = R(s, a) + \sum_{s'} p(s' \mid s, a) V(s')$ and $V(s) = \max_{a'} Q(s, a')$.

State-Agnostic Multiarmed Bandit Strategy

In the state-agnostic framework, actions are based on current sensing and past action results. Thompson [16] introduced the multiarmed bandit (MAB) approach to address the problem of action–result research issues. This section presents the MAB formulation for CCTN with the goal of accumulating optimal rewards from unknown parameters of the channel-node interactions that need to be learned sequentially. Each CCTN network has C comm nodes and J jammers. The blue-force and red-force node actions at time t are $a_B^t = \{a_{B,\text{comm}}^t, a_{B,\text{jam}}^t\}$ and $a_R^t = \{a_{R,\text{comm}}^t, a_{R,\text{jam}}^t\}$, containing both comm and jamming actions, the size-C vectors $a_{B,\text{comm}}^t$ and $a_{R,\text{comm}}^t$, and the size-J vectors $a_{B,\text{jam}}^t$ and $a_{R,\text{jam}}^t$. An ith element in $a_{B,\text{comm}}^t$ designates the channel number that the ith blue-force comm node tries to access at time t. Similarly, a jth element in $a_{B,\text{jam}}^t$ is the channel number that the jth blue-force jammer tries to jam at time t. Let Ω^t be a size-N vector that describes the outcome of the blue- and red-force node actions used to determine the rewards:

$$a_B^t \times a_R^t \rightarrow \Omega^t$$

It is more convenient to compute a reward from each channel (than from each node), and we use $r_{B,k}^t$ to designate the instantaneous reward for the blue force resulting from channel k at time t. The total reward at time t is the sum over all N channels: $R_B^t = \sum_{k=1}^{N} r_{B,k}^t$. The blue-force network strategy, σ_B^t, is a function over time. It takes necessary information, such as sensing results and past action-outcome/reward statistics, as input and determines the blue-force node actions. Under the centralized decision making,

$$\{x_B^j\}_{j=1}^{t}, \{a_B^j, \Omega^j\}_{j=1}^{t-1} \xrightarrow{\sigma_B^t} a_B^t \ ,$$

where x_B^t is the blue-force sensing results up to time t. The first term in the second braces is all past actions up to $t-1$ for all nodes; the second term is all past outcomes up to $t-1$, which implicitly contain information on red-force actions. The centralized decision maker applies the strategy σ_B^t over all nodes and produces the action a_B^t.

Under the distributed decision making, each node in the network computes its own action. For node i in the blue-force network (whether it is a comm node or jammer),

$$x_{B,i}^t, \{x_B^j, a_B^j, \Omega^j\}_{j=1}^{t-1} \xrightarrow{\sigma_{B,i}^t} a_{B,i}^t$$

where $x_{B,i}^t$ is the sensing information only available to node i at time t, and $\sigma_{B,i}^t$ is the strategy of node i's own. At time t, node i does not yet have all sensing results except its own $x_{B,i}^t$. For the distributed case, node strategies can differ, and there is no guarantee that conflicting actions of the nodes in the same network, such as collision and misjamming, are resolved.

The MAB strategy is best explained with a gambler facing N slot machines (arms). The gambler's objective is to find a strategy that maximizes $R^t = \sum_{j=1}^{t} r^j$, the cumulative reward over a finite time horizon. Lai and Robbins [17] introduced the concept of regret for strategy measuring the distance from optimality

$$\Gamma^t = t\mu^* - \mathbb{E}\left[R_\sigma^t\right]$$

where μ^* is the hypothetical maximum reward (i.e., the "gold standard") if a gambler's action resulted in the best possible outcome each time, and $\mathbb{E}\left[R_\sigma^t\right]$ is the expectation of the actual reward achieved with σ. The expression Γ^t is mathematically convenient, and maximizing the *expectation* of R^t turns out to be equivalent to minimizing Γ^t. Lai and Robbins [17] further derived the mathematical qualification for an optimal strategy:

$$\lim_{t\rightarrow\infty} \sup \mathbb{E}\left[T_k^t\right] \le \frac{\log t}{D_{KL}(p_k \,||\, p^*)} \ ,$$

where T_k^t is the total number of playing arms k, sup(.) is the least-upper bound, and $D_{KL}(.||.)$ is the Kullback-Leibler divergence [18] measuring the dissimilarity between the probability distribution p_k and p^*, the kth arm's reward and the maximum reward achieved by choosing only the best possible arm each time. The above equation provides the least-upper bound for the number of times should an optimal arm—which could be different each time—be played asymptotically.

The mapping of the MAB model to CCTN channel access is as follows. An arm corresponds to a channel in the spectrum under competition. The comm nodes and jammers are the players that the networks allocate to play (i.e., transmit or jam) the channels. Since each network has multiple nodes, our problem is classified

as multiplayer MAB, which is different from the classic single-player MAB formulated by Lai and Robbins [17]. In addition, two system variations depend on whether a centralized control entity or each player makes the play decisions. The CCTN with a central decision maker (e.g., base station or master node) computes the network-wide strategy and disseminates all node actions. For the centralized multiplayer MAB, it is critical that the decision maker be able to collect sensing results from each player and the exact outcome of every play in order to make sound decisions over time. For the distributed decision-making model, each node makes its own play decision on the basis of the best-effort information collected. Contrast this scenario to the centralized multiplayer MAB that requires tight intranetwork communication to collect information and disseminate the strategy. After each play, the node observes the outcome, computes its reward, and maintains its play statistics, all of which can be shared with others in the network

The MAB formulation for CCTN models the problem of sequentially sampling the total network reward from the N channel population rewards $r_1^t, r_2^t, \ldots, r_N^t$ over time. The rewards are manifested by the mixed player actions from the same and opposing networks that dynamically affect the outcome each time. Differentiated from the classic MAB problems, the player action in CCTN comprises an action (transmit) and its anti-action (jam). The anti-action does not draw the reward directly from a channel but can deprive that generated by a comm node. Formally, we search for an optimal strategy, σ_{opt}^t, that minimizes the regret Γ^t:

$$\sigma_{\text{opt}}^t = \underset{\sigma}{\operatorname{argmin}}\, \Gamma^t = \min_{\sigma}\left\{\mathbb{E}\left[\sum_{i=1}^{M}\sum_{j=1}^{t} r_{(i)}^j\right] - \mathbb{E}\left[R_\sigma^t\right]\right\}$$

For the regret expression, we use $r_{(i)}^t$, an ordered sequence of the N instantaneous channel rewards at time t such that $r_{(1)}^t \geq r_{(2)}^t \geq \ldots \geq r_{(N)}^t$. There are $M = C + J$ total number of nodes in the blue network; the summing of the $M < N$ highest rewarding channels reflects the optimal allocation of players.

Reward Model

A reward metric is used to evaluate the performance of the CCTN. When a CCTN comm node achieves a successful transmission of a packet containing B bits of data, it receives a reward of B. The definition of a successful transmission follows the rule of thumb from classical wireless networking that there should be only one comm node transmission for the transmit opportunity. If there were two or more simultaneous comm transmissions (from either the same or a different network), a collision occurs, and no comm node gets a reward. With packet capture, however, the possibility of a successful reception in the presence of multiple transmissions can increase substantially. This packet capture can further enhance the reward performance.

Jammers by themselves do not create any reward. They receive a reward by suppressing an opposing comm node's otherwise successful transmission. For example, a blue-force jammer earns a reward B by jamming the slot in which a sole red-force comm node tries to transmit B bits. If there were no jamming, the red-force comm node would have earned B. Also, a blue-force jammer can jam a blue-force comm mistakenly (e.g., caused by faulty intranetwork coordination), an occurrence we call misjamming (incurring no reward). Table 3 summarizes the outcome at a slot of transmission opportunity.

To better understand simulated results presented in the following section, we present an illustrative example for an optimal blue-force strategy against the static red-force network with C = 4 and J = 2 in Figure 11. In this example, red-force comm nodes are fixed at channels 1, 2, 3, 4, and its 2 jammers at channels 5 and 6, leaving the rest of channels 7, 8, 9, 10 free of red-force actions. Each comm node has a transmit probability of 0.5 whereas jammers jam with a probability of 1. Through learning by sensing all channels over time, an optimal blue-force strategy should place its two jammers somewhere between channel 1 and 4. Because the comm transmit probability at any given slot is 0.5 for all red-force and blue-force comm nodes, the maximum average reward earned by the blue-force jammers should be $\mathbb{E}\left[R_{B,\text{jam}}\right] \approx 0.5 \times 2 = 1$. The blue-force comm nodes at channels 7, 8, 9, and 10 should earn $\mathbb{E}\left[R_{B,\text{comm}}\right] \approx 0.5 \times 4 = 2$ (because of the comm transmit probability of 0.5, the blue-force comm reward at each time slot comes from two channels on the average). In summary, the total reward for blue-force network in this example is approximately 3, which is normalized to 3/N = 3/10 = 0.3 (per channel) and, similarly for red-force network, the average total reward is 1/10 = 0.1.

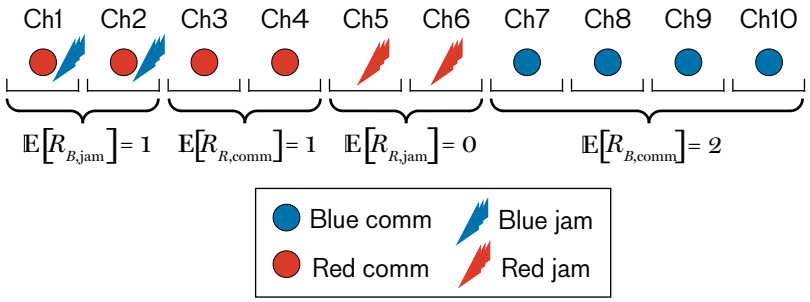

FIGURE 11. When red-force actions are stationary, the blue force can optimize its strategy with prior knowledge through learning by sensing all channels over time.

Performance Evaluation

For the numerical assessment of the proposed CCTN, we consider both the transient and steady-state results for state-agnostic and state-aware cognitive algorithms. In each experiment, we apply a cognitive algorithm to the blue-force network and a noncognitive algorithm in the form of static or random allocations to the red-force network. Both centralized and distributed spectrum access decisions are considered. Then, we compare the performance of each network in terms of total average reward accrued over the simulation time.

Simulation Parameter Configuration

Table 4 describes the CCTN simulation parameters and their corresponding values. In the case of static channel access, comm nodes and jammer nodes are assigned to their respective channels in a fixed manner for the duration of the simulation. A comm node may wish to transmit on its channel with probability P_{TX} or not transmit with probability $(1 - P_{TX})$. Similarly, a jammer may wish to jam its respective channel with probability P_J or refrain from jamming with probability $(1 - P_J)$. In the case of random channel access, comm nodes and jammer nodes select a channel uniformly with probability $1/N$ and then proceed transmitting or jamming as in the static case.

Transient Analysis

Transient analysis allows us to observe the convergence behavior of the cognitive algorithms. In the first experiment, several simulation runs are conducted. In each run, the blue-force network employs one of the two cognitive algorithms (e.g., Q-Learning or MAB) while the red-force network adopts static or random channel access. The transmit probabilities and jamming probabilities are assumed to take values of one for all nodes $P_{TX} = 1$ and $P_J = 1$. In Figure 12, we plot the cumulative

Table 3. Reward Model for Single-Channel Scenario

BLUE-FORCE ACTION		RED-FORCE ACTION		OUTCOME	REWARD	
Comm	Jam	Comm	Jam		Blue	Red
Transmit	–	–	–	Blue-force transmit success	1	0
–	Jam	Transmit	–	Blue-force jamming success	1	0
Transmit	–	Transmit	–	Blue-force/Red-force comm collide	0	0
–	Jam	–	Jam	Blue-force/Red-force jam collide	0	0
Transmit	Jam	–	–	Blue-force misjams Blue-force comm	0	0
Transmit	–	–	Jam	Red-force jam success	0	1
				...		

average rewards for the blue-force network operating Q-learning–based methods (minimax-Q, Nash-Q, FFQ) and MAB against the red-force network's static and random strategies over time. Under the chosen simulation parameters, the Q-learning and MAB algorithms converge to a steady-state distribution of the blue-force actions within 1000 iterations. Under such convergence, the blue-force average cumulative reward metric seems to approach an asymptotically optimal value. We observe that the minimax criterion results in a more aggressive strategy than Nash-Q: (1) minimax-Q converges to a steady-state cumulative average reward value faster; and (2) it outperforms Nash-Q by achieving slightly higher rewards over time. Static strategy has almost no chance against the learning algorithms as its steady-state average cumulative reward approaches zero. On the contrary, learning seems harder against the random strategy, particularly because of this strategy's effectiveness in jamming. Also, it is important to note that the state-aware algorithms are asymptotically faster than those of the state-agnostic MAB scheme.

Table 4. CCTN Simulation Parameters and Values

PARAMETER TYPE	PARAMETER VALUE
Number of blue-force comm nodes, C_{BF}	2
Number of blue-force jammer nodes, J_{BF}	2
Number of red-force comm nodes, C_{RF}	2
Number of red-force jammer nodes, J_{RF}	2
Number of channels, N	10
Probability of comm transmission, P_{TX}	$0 \le P_{TX} \le 1$
Probability of selecting a channel (random access only)	$1/N$
Simulation time	2000 slots

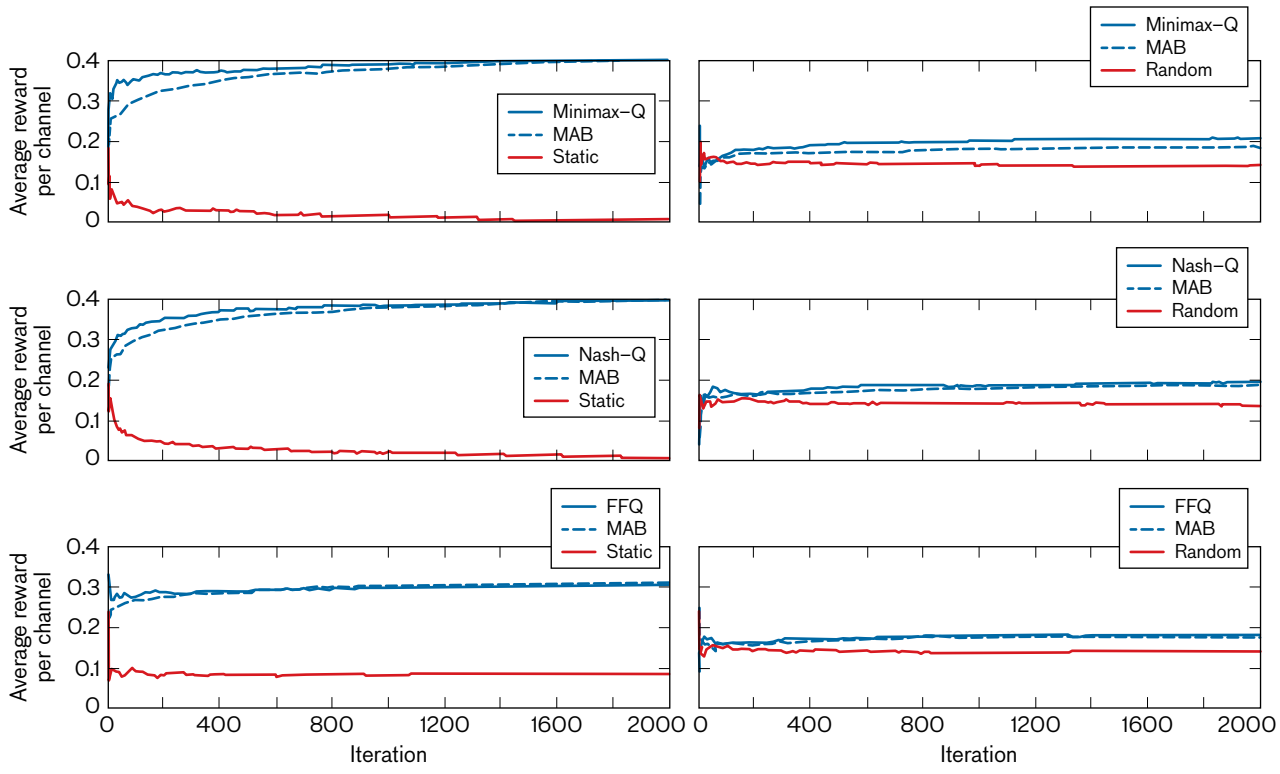

FIGURE 12. Transient behavior of blue-force Q-learning and multiarmed bandit (MAB) strategies versus red-force static (left) and random (right) strategies.

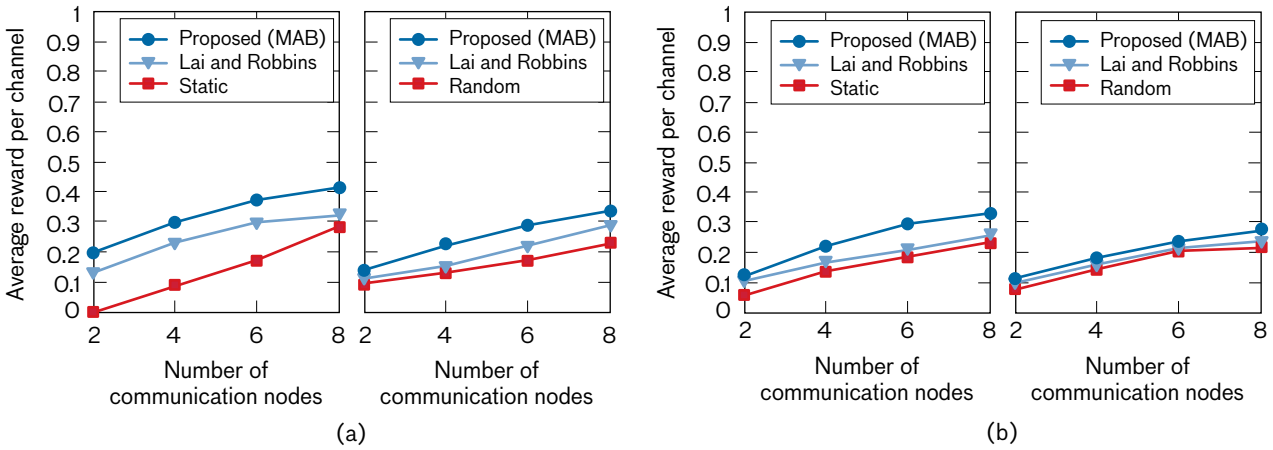

FIGURE 13. Steady-state behavior of blue-force MAB strategies versus red-force static and random strategies in (a) centralized control and (b) distributed control for the blue network. Each point in these plots is obtained after running 1000 time slots. The unit of the *y*-axis is the average reward per channel for each network. For example, the blue network average reward is 30% while the red network is around 10% at $C = 4$ comm nodes in the left panel of subfigure (a). This result can be intuitively observed by the red-force static strategy, which occupies two channels 50% of the time, resulting in a 10% reward of the 10-channel system. On the other hand, the blue network's reward consists of jamming the 2 channels occupied by the red comm nodes at 50% of the time, and its 4 comm nodes occupying unjammed channels at 50% of the time, incurring a 30% reward of the 10-channel system.

Steady-State Analysis

In the steady-state regime, the blue-force cumulative reward is compared with the red-force cumulative reward as a function of the number of comm nodes. In this experiment, the number of jammers per network is kept constant, but the number of comm nodes per network is varied. Also, the transmit probability per node is kept at $P_{TX} = 1/2$. Figure 13a illustrates an increase in the average cumulative reward obtained per network as a function of comm nodes. As expected, our proposed MAB cognitive algorithm outperforms Lai and Robbins' MAB algorithms, as indicated in panel (a) for the centralized control case in the blue network. The random strategy used by the red-force network performs better than the static strategy, but underperforms the blue-force network's MAB strategy as depicted in panel (a). Figure 13b depicts the same strategy matchups between the blue and red networks, however, under the distributed control in the blue network. Similarly, we can observe that the blue-force performance is superior to the red-force performance. However, this performance is slightly degraded due to the imperfect coordination of the distributed control model.

The steady-state results for the blue-force cumulative reward when Q-learning strategies are employed outperforms the red-force network's static strategy as indicated in the left panels of Figure 14a and b. As in the previous case, the transmit probability per node is kept at the value of $P_{TX} = 1/2$. The cognitive algorithms' learning ability diminishes when the red force's strategy is random, as illustrated in the right panels in Figure 15a and b. The cognitive algorithms always perform better than the noncognitive algorithms when the number of channels is much larger than the total number of comm and jammer nodes in the network, $N >> C + J$. The intuition behind this case is that cognitive strategies are more effective when the decision makers have sufficiently larger action space.

In the final experiment, the blue-force network is set up with minimax-Q learning and the red-force network with the state-agnostic MAB strategy. Using $N = 10$, $C = 4$, and $J = 2$, we varied the comm transmit and jamming probabilities equally for the two CCTNs and tested the two cognitive algorithms under both the centralized and distributed control scenarios (Figure 15). The surface plots allow us to observe each network's performance as a function of both comm transmit probability and jammer jamming probability, P_{TX} and P_J.

The performance of the blue-force strategy is on par with that of the red-force strategies since both networks can learn about each other's strategy. This observation motivates the need to further research more effective and

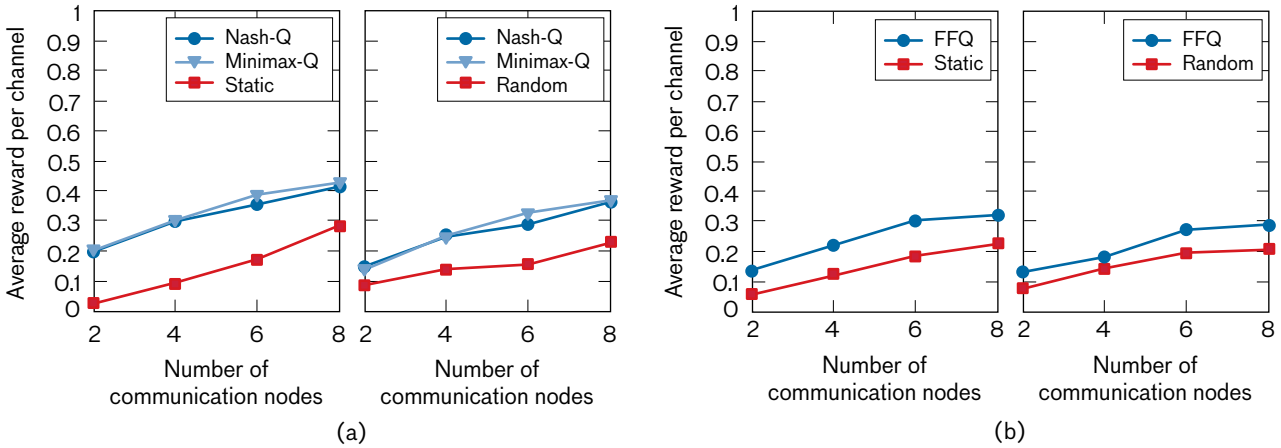

FIGURE 14. Steady-state behavior of the blue-force Q-learning strategy versus red-force static and random strategies in (a) centralized control and (b) distributed control for the blue network. Each point in these plots is obtained after running 1000 time slots. The Nash-Q is more sensitive to learning the opponent's exact action. This sensitivity may have caused small perturbations in Nash-Q performance compared to minimax-Q as indicated in subfigure (a). Friend-or-foe Q-learning (FFQ) is more suitable for a distributed control setting and outperforms noncognitive red-force strategies consistently, as shown in subfigure (b).

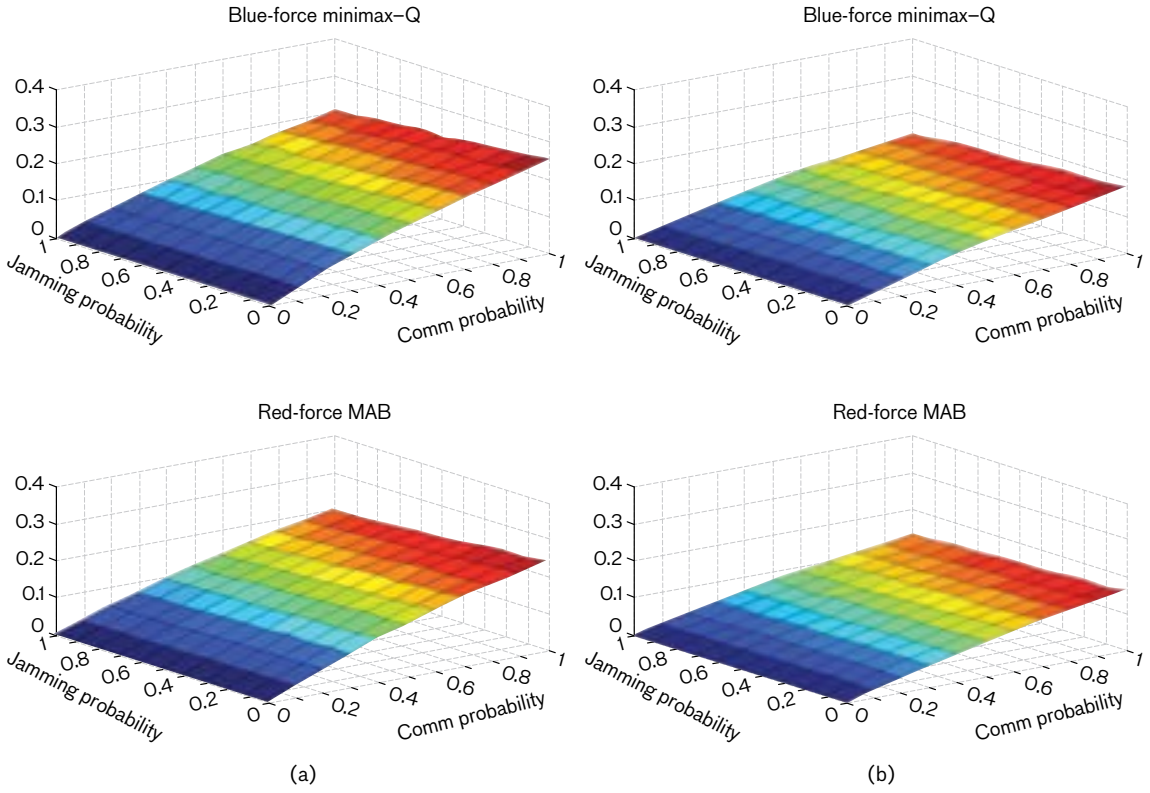

FIGURE 15. Steady-state behavior of the blue-force Q-learning strategy versus the red-force multi-armed bandit (MAB) strategy for (a) centralized control and (b) distributed control scenarios. The upper panel in subfigure (a) illustrates the average reward performance for the minimax-Q strategy on the blue force, whereas the lower panel depicts the red force's MAB strategy. The performances of both networks seem comparable because their respective cognitive strategies in the steady-state achieve similar learning. However, as depicted in Figure 12, the transient behaviors of these two strategies differ in convergence property. The faster convergence gain of the Q-learning strategies is accompanied with a higher computational cost of keeping track of CCTN system states. The performances of both blue and red networks equally degrade by the imperfect coordination resulting from distributed control, as illustrated in subfigure (b).

agile cognitive strategies that can be implemented in real or near-real time, and can scale to large networks.

Performance Prediction Test Bed

For test and evaluation purposes, we have developed models of the CCTN functions and their corresponding algorithms in the discrete-event operations network simulation environment. This modeling environment offers an integrated view of the CCTN algorithms and allows us to observe the dynamics of various algorithms and their interactions under mobility and realistic channel conditions in real time. Furthermore, architectural design studies and algorithmic trade-offs can be easily accomplished. Finally, various user applications, such as real-time video and voice data, can be incorporated and tested in this environment, allowing researchers and engineers to quantitatively and qualitatively evaluate their respective algorithms. Figure 16 is a snapshot of a scenario in which both a blue-force network and a red-force network are operating. The embedded instrumentations in the models provide real-time performance statistics for sensing. Channel access rewards (e.g., data throughput for each network) are shown on the right side of the network visualizer of the modeling environment.

In summary, we have provided two cognitive approaches to strategize the joint comm and jamming actions for a CCTN, namely the state-aware reinforcement Q-learning and the state-agnostic MAB. Preliminary evaluations of the proposed cognitive algorithms indicate that they outperform the rudimentary schemes such as static and random channel access. The cognitive strategies perform better against the static channel access compared to the random channel access because the nodes randomly explore channels for higher rewards in the latter case. More analysis is needed to derive upper bounds for the blue force cognitive strategies' reward performance against random scheme as a function of the

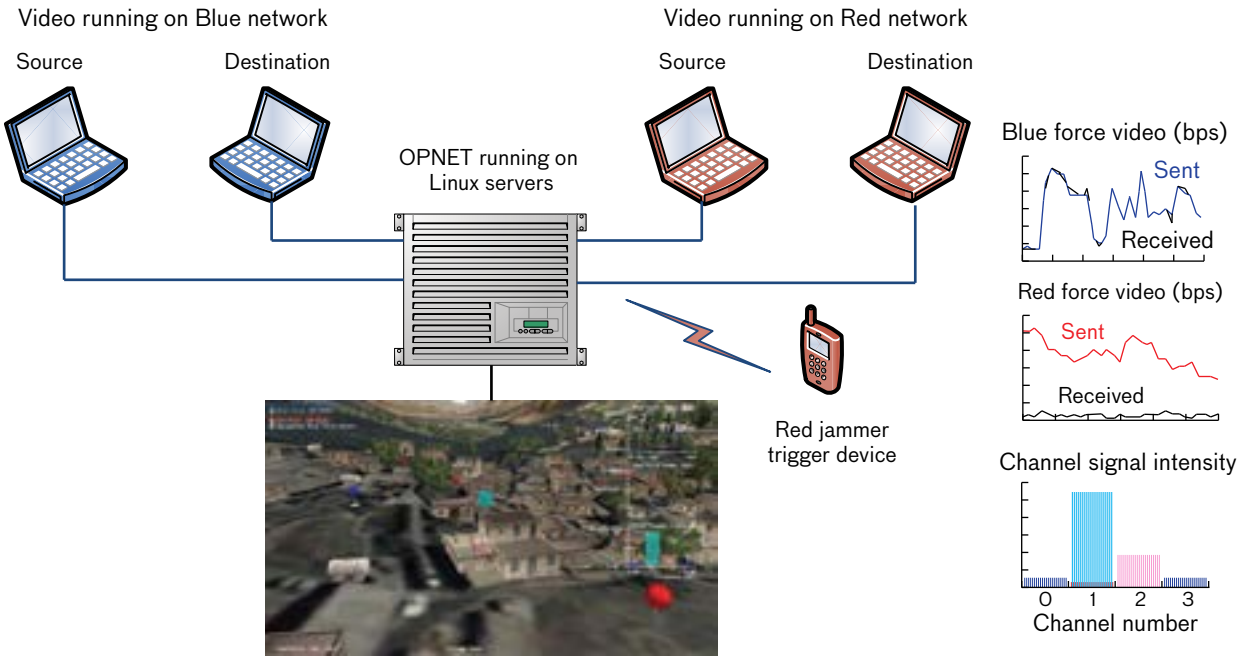

FIGURE 16. A network view in the operations network environment shows both blue- and red-force networks in CCTN.The graphs on the screen image are reproduced on the right. The bar chart on the lower right represents the channel signal intensity from one blue node's perspective. For example, two channels (channel 0 and 1) are occupied by the blue comm nodes at this instance of simulation. Channel 1 is used by the red comm but it is currently being jammed by one of the blue jammers (see tall light blue bar). One of the red jammers is jamming channel 2 which was previously occupied by the blue comm. However, the blue comm had sensed it and cognitively switched to channel 3. The upper chart represents the throughput performance for the blue network. Both sent and received traffic curves coincide as expected. The middle chart illustrates the throughput performance of the red network. The red curve represents traffic sent and the gray curve represents the traffic received, which in this case is close to zero because the channel currently being used by the red comm is being jammed by the blue jammer.

number of channels. This should allow us to apply theoretical performance bounds to choose practical system design parameters. Furthermore, we have demonstrated that the reward performance of CCTN is superior when node actions are decided in a centralized control manner compared to a distributed control scheme.

Future Work

Competing Cognitive Tactical Networks operate in hostile environments and strive for dominant access to an open spectrum. Our notion of CCTN emphasizes the optimal data throughput for comm nodes in a friendly coalition and maximal suppression of hostile comm and jamming entities. A discrete-event simulation environment in which the CCTN functions and algorithms are modeled allows us to better understand and optimize the relevant design parameters and conduct different experiments.

Our immediate future work includes (1) algorithmic improvements to scale the number of nodes in a network efficiently, adding more friendly and enemy networks to the current two-network model, and (2) rigorous analysis on the accidental use of incorrect information (resulting from sensing errors) in learning, failover, and system component design, such as cognitive sensing and jamming detection at the physical and media access control (MAC) layers for current and future tactical communications protocols. We also envision enhancing our computational framework through more robust linear programming methodologies and parallelization.

Acknowledgment

The authors would like to acknowledge Jim Vian, assistant group leader in the Wideband Tactical Networking Group, for his continued support and influential ideas throughout our work. ■

References

1. Federal Communications Commission Press Release, "FCC Adopts Rules for Unlicensed Use of Television White Spaces," November 2008.
2. B. Wang, Y. Wu, K. Liu, and T. Clancy, "An Anti-jamming Stochastic Game for Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, 2011, pp. 877–889.
3. M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications; Vols. 1–3*. New York: Computer Science Press, 1985.
4. S. Verdu, *Multiuser Detection*. New York: Cambridge University Press, 1998.
5. D. Browne, "Detection of Unknown Signals in Unknown, Non-Stationary Noise," *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009.
6. N. Abramson, "THE ALOHA SYSTEM: Another Alternative for Computer Communications," *Proceedings of the ACM Fall Joint Computer Conference*, 1970, pp. 281–285.
7. W. R. Thompson, "On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, vol. 25, no. 3–4, 1933, pp. 285–294.
8. L. S. Shapley, "Stochastic Games," *Proceedings of the National Academy of Sciences*, 1953, pp. 1095–1100.
9. C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, 1992, pp. 279–292.
10. W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," *Proceedings of the 6th ACM International Symposium on Mobile ad hoc Networking and Computing*, 2005, pp. 46–57.
11. M. Pajic and R. Mangharam, "Anti-jamming for Embedded Wireless Networks," *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, 2009, pp. 301–312.
12. M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, N.J.: Wiley, 1994.
13. Y. Gwon, S. Dastangoo, C. Fossa, and H. T. Kung, "Competing Mobile Network Game: Embracing Antijamming and Jamming Strategies with Reinforcement Learning," *IEEE Conference on Communications and Network Security*, October 2013.
14. R. Bellman, *Dynamic Programming*. Princeton, N.J.: Princeton University Press, 1957.
15. M. L. Littman, "Markov Games as a Framework for Multi-agent Reinforcement Learning," *Proceedings of the 11th International Conference on Machine Learning*, 1994, pp. 157–162.
16. W. R. Thompson, "On the Likelihood That One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, vol. 25, no. 3–4, 1933, pp. 285–294.
17. T. L. Lai and H. Robbins, "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Mathematics*, vol. 6, no. 1, 1985, pp. 4–22.
18. T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, N.J.: Wiley-Interscience, 1991.
19. L. de Haan and A. Ferreira, *Extreme Value Theory: An Introduction*. New York: Springer, 2006.
20. R. A. Fisher and L. H. C. Tippett, "Limiting Forms of the Frequency Distribution of the Largest and Smallest Member of a Sample," *Proceedings of the Cambridge Philosophical Society*, 1928, pp. 180–190.
21. B. V. Gnedenko, "Sur la distribution limite du terme maximum d'une serie aleatoire," *Annals of Mathematics*, 1943, pp. 423–453.
22. Y. Gwon, S. Dastangoo, and H. T. Kung, "Optimizing Media Access Strategy for Competing Cognitive Radio Networks," *Proceedings of IEEE GLOBECOM*, 2013.

APPENDIX

Algorithmic Approaches to Find Optimal CCTN Strategies

This section describes algorithmic approaches to determine CCTN actions.

State-Agnostic Algorithm

We propose a new MAB algorithm based on extreme-value theory [19], conjugate priors, and Thompson sampling.

1. Distribution of maximum reward sequence: Let $Y^t = \max\{r_1^t, r_2^t, \ldots, r_N^t\}$, where $r_{(i)}^t$ represents the reward from channel i at time t. Since the sequence $Y_1^t, Y_2^t, \ldots, Y_N^t$ consists only of the maximum channel reward each time, it must have achieved the distribution p^* in the divergence test,

$$\lim_{t\to\infty} \sup \mathbb{E}\left[T_k^t\right] \le \log t / D_{KL}(p_k \,||\, p^*) .$$

Furthermore, the sequence should result in an upper bound of the optimal mean reward μ^*. Therefore, all that is needed is a strategy σ to empirically follow the distribution of Y^t. But how is it distributed? Fisher and Tippet [20] and Gnedenko [21] proved the existence of limiting distributions for block maxima (or minima) of random variables. Their findings became the foundation of extreme-value theory used widely in financial economics. Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables and $M_n = \max\{X_1, X_2, \ldots, X_n\}$. If real number pairs (a_n, b_n) exist such that $a_n, b_n > 0$ and

$$\lim_{n\to\infty} P\left(\frac{M_n - b_n}{a_n} \le x\right) = F(x) ,$$

where $F(.)$ is a nondegenerate distribution function, then the limiting distribution $F(.)$ belongs to only the Frechet, Gumbel, or Weibull family of probability distribution functions.

2. Conjugate priors: In Bayesian inference, the posterior is updated by the observed likelihood given the prior distribution:

$$\underbrace{p(\theta|r)}_{\text{posterior}} \propto \underbrace{p(r|\theta)}_{\text{likelihood}} \times \underbrace{p(\theta)}_{\text{prior}} ,$$

where θ is a set of parameters that have to be estimated. For example, θ could be related to the mean reward for a channel. When the probabilistic model for the likelihood is known, we can set the prior and posterior distributions conveniently of the same family of functions. This is known as conjugate prior. Since the reward distribution in this context is extreme valued, the likelihood choices for our search are left to Frechet, Gumbel, or Weibull distributions.

3. The algorithm: In summary, our algorithm below performs Thompson sampling that follows an extreme-valued likelihood and updates the posterior distribution based on its conjugate prior. However, we need to decide on which extreme-value distribution is suitable for CCTN. Since both Frechet and Gumbel distributions model unbounded random variables, we adopt a Weibull likelihood with the inverse gamma conjugate prior, reasoning that the maximum reward value for CCTN networks should be finite. A Weibull distribution has finite endpoints. Its conjugate prior, the inverse gamma distribution, has two hyperparameters $a, b > 0$. Our algorithm draws the scale parameter θ from the inverse gamma prior

$$p(\theta|a, b) = \frac{b^{a-1}e^{-b/\theta}}{\Gamma(a-1)\theta^a} \text{ for } \theta > 0 ,$$

where a and b are the sample mean and variance of the reward of a channel. (Note that here Γ is the gamma function.) The Weibull random variable generated by θ drawn from the prior state estimates the expected reward for the channel. After observing the actual reward, the posterior update follows. For more details of the CCTN state-agnostic algorithm, see Gwon, Dastangoo, and Kung's article [22].

Algorithm 1 CCTN State-Agnostic Algorithm (MAB)

Require: $a_i, b_i = 0\ \forall i$
1: **while** $t < 1$ ▷ initalized offline
2: Access each channel until $a_i, b_i \neq 0\ \forall i$, where a_i and b_i are sample reward mean and variance
3: **end**
4: **while** $t \ge 1$ ▷ online
5: Draw $\theta_i \sim$ inv-gamma(a_i, b_i)
6: Estimate $\hat{r}_i =$ weibull$(\theta_i, \beta_i)\ \forall i$ for given $0.5 \le \beta_i \le 1$
7: Access channel $i^* = \arg\max_i \hat{r}_i$
8: Observe actual $r_{i^*}^t$ to update $\{R_{i^*}^t, T_{i^*}^t\}$
9: Update $a_{i^*} = a_{i^*} + T_{i^*}^t$, $b_{i^*} = b_{i^*} + \sum_t (r_{i^*}^t)^{\beta_{i^*}}$
10: **end**

State-Aware Algorithm

In reinforcement learning, there are two approaches to finding optimal strategies for CCTN. The first is the model-based approach in which a strategy maker, termed agent, explicitly learns the Markovian model (e.g., transition probabilities) that underlies the system as described in the previous section. The second approach is model free, and the agent tries to directly formulate a strategy by learning from evaluative measures of each action at a given state. Algorithm 2 is based on Q-learning, a kind of temporal-difference learning method.

Algorithm 2 CCTN Stateful Algorithm

Require: $Q(s, a_B, a_R) = 1$, $V(s) = 1$, $\pi(s, a_B) = \frac{1}{|A|}$ $\forall$ state $s \in S$, BF action $a_B \in A$, RF action $a_R \in A$; learning rate $\alpha < 1$ with decay $\lambda \leq 1$ (α, λ nonnegative)

1: **while** $t \geq 1$
2: Draw $a_B^t \sim \pi(s^t)$ and execute
3: Observe r_B^t
4: Estimate a_R^t given observed reward
5: Compute s^{t+1}
6: $Q(s^t, a_B^t, a_R^t) = (1 - \alpha)Q(s^t, a_B^t, a_R^t) + \alpha(r_B^t + \gamma V(s^{t+1}))$
7: linprog: $\pi(s^t,.) = \arg\max_\pi \min_{a_R} \sum_{a_B} \pi(s^t, a_B) Q(s^t, a_B, a_R)$
8: Update $V(s^t) = \min_{a_R} \sum_{a_B} \pi(s^t, a_B) Q(s^t, a_B, a_R)$
9: Update $\alpha = \lambda \times \alpha$
10: **end**

We have implemented minimax-Q, Nash-Q, and FFQ learning algorithms in MATLAB, using the linprog function from MathWorks' Optimization Toolbox. We need to maintain the Q table, which is a three-dimensional array that can be looked up by using state, blue-force, and red-force action vectors. At the end of each time slot, we compute the next state from the sensing result of each channel. Recall that state computation is done by counting I_C, I_D, J_C, and J_D parameters described earlier. The action vector space is discrete, and we have pregenerated and indexed all possible action vectors for the blue and red forces. A strategy π is a two-dimensional array indexed by state and action vector (either blue or red force). The V table for the value function is indexed only by state. For a more detailed description and in-depth analysis of the state-aware algorithm, see the authors' paper [13].

About the Authors

Siamak Dastangoo is a member of the technical staff in the Wideband Tactical Networking Group, where he has been conducting research and development in the area of wireless ad hoc networks. Prior to joining the Laboratory in 2006, he spent several years in the commercial and defense industries working on a range of problems in communications and networking systems. He received a bachelor's degree and a master's degree in electrical engineering from the University of Massachusetts and a doctoral degree in electrical engineering from the George Washington University. His current research interests are in the areas of cognitive networks and performance predictions of networks.

Carl E. Fossa is currently the assistant leader of the Wideband Tactical Networking Group. Since joining Lincoln Laboratory in 2008, he has focused on the performance of mobile ad hoc networks in tactical military environments. Applications of this work include the development and deployment of a real-time emulation of the Army Warrior Information Network–Tactical (WIN-T) to Aberdeen Proving Grounds, Md. He holds a doctoral degree from Virginia Polytechnic Institute and State University, a master's degree from the Air Force Institute of Technology, and a bachelor's degree from the United States Military Academy, all with a major of electrical engineering. Prior to joining Lincoln Laboratory, he served as an Army Signal Officer for 21 years, retiring at the rank of Lieutenant Colonel. He has served in a range of tactical military positions, which included deployment to Operation Desert Shield/Storm. He also served in a number of technical engineering positions at major command headquarters and as an assistant professor of electrical engineering at the United States Military Academy.

Youngjune L. Gwon is a doctoral candidate in computer science at Harvard University, where he is advised by Professor H. T. Kung. His research interests include systems, networking, wireless communications, and machine learning. During research internships in Lincoln Laboratory's Wideband Tactical Networking Group during the summers of 2012 and 2013, he focused on a machine learning approach for cognitive tactical networks. He has a master's degree from Stanford University and a bachelor's degree from Northwestern, all in electrical engineering. He worked in Silicon Valley for 10 years before coming to Harvard.