Off-sensing and Route Manipulation Attack: A Cross-Layer Attack in Cognitive Radio based Wireless Mesh Networks

Moinul Hossain and Jiang Xie
Department of Electrical and Computer Engineering
The University of North Carolina at Charlotte
Email: {mhossai4, Linda.Xie}@uncc.edu

***Abstract*—Cognitive Radio (CR) has garnered much attention in the last decade, while the security issues are not fully studied yet. Existing research on attacks and defenses in CR-based networks focuses mostly on individual network layers, whereas cross-layer attacks remain fortified against single-layer defenses. In this paper, we shed light on a new vulnerability in cross-layer routing protocols and demonstrate how a perpetrator can exploit this vulnerability to manipulate traffic flow around it. We propose this cross-layer attack in CR-based wireless mesh networks (CR-WMNs), which we call *off-sensing and route manipulation* (OSRM) attack. In this cross-layer assault, *off-sensing* attack is launched at the lower layers as the point of attack but the final intention is to manipulate traffic flow around the perpetrator. We also introduce a learning strategy for a perpetrator, so that it can gather information from the collaboration with other network entities and capitalize this information into knowledge to accelerate its malice intentions. Simulation results show that this attack is far more detrimental than what we have experienced in the past and need to be addressed before commercialization of CR-based networks.**

I. Introduction

The pervasive adoption of unlicensed bands to provide wireless broadband services has made it a bottleneck to meet the ever-increasing demand of wireless bandwidth. Therefore, the regulatory bodies and the research communities have come forward to explore innovative ways to meet this demand. According to the Federal Communications Commission (FCC), a significant portion of the radio spectrum remains highly underutilized whereas high volume of traffic appears in a small portion of the spectrum. Thereby, FCC has proposed a new radio spectrum utilization paradigm, where an unlicensed user (or SU) can opportunistically access the licensed spectrum without affecting the existing licensed users (or PUs). Cognitive Radio (CR) appears to be a cornerstone to realize this paradigm and enables opportunistic wireless access for unlicensed users.

One of the promising applications of the CR technology is wireless mesh networks (WMNs) [1], [2], because WMNs usually suffer from inter-flow interference [3] and insufficient channels to mitigate it, whereas the CR technology offers an intelligent solution to the interference problem in WMNs via accessing licensed bands in an opportunistic manner. However, since CRs adapt to the surrounding radio environment based on sensing the radio channels around them and collaborations with peer nodes, it is crucial that their belief of their own surroundings is not compromised and diverted in a wrong direction by a perpetrator.

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant No. 1343355, 1718666, and 1731675.

A CR-WMN consists of CR-enabled wireless mesh routers/access points (CR-WMRs or SUs interchangeably), mobile devices connected to the CR-WMRs, and a gateway which is connected to the Internet. Internet traffic between mobile devices and the gateway is carried by the CR-WMRs and CR-WMRs can opportunistically access the spectrum when no PUs are using it. In reality, the spectrum usage by PUs varies over time and space. Thus, spectrum availability is different to CR-WMRs depending on their locations. Hence, due to the uncertainty of any single channel being available to all SUs at all the time, it is very challenging to have a dedicated common control channel (CCC) in the network. Therefore, two CR-WMRs are blind to each other (i.e., they do not know any information about each other) until they rendezvous [4] on a common available channel. The state-of-the-art work usually proposes that two SUs hop onto different channels from one time slot to another until they meet on a common available channel and then, they can exchange control information.

However, the policy of accessing licensed channels in a non-interfering basis could make it a potential vulnerability. In CR-based networks, an attack that exploits this vulnerability is called Primary User Emulation (PUE) attack [5]. In this attack, an attacker impersonates as a PU by mimicking PU signal characteristics and transmitting on the licensed channel. SUs falsely believe it as a benign PU transmission and abstain from accessing the channel. To defend such attacks, numerous solutions have been proposed based on spectrum sensing. Most of the existing spectrum sensing approaches require that a SU should sense the spectrum periodically for returning PUs. In order to successfully detect returning PUs, the sensing period has to be designed in a way that the sensing interval coincides with the transmission of PUs. When the transmission from a PU is not detected, a SU may end up interfering the PU's transmission. Consequently, it impacts the throughput of both primary and secondary networks.

Moreover, this realistic way of PU misdetection can be leveraged by an attacker for creating a new window of vulnerability. An attack that exploits this vulnerability is called *off-sensing* attack [6]. In this attack, a perpetrator interferes a neighboring SU's transmission only when the neighboring SU is not sensing but transmitting. Using this strategy, the perpetrator can interfere to corrupt the data transmission of the victim SU and trick it into believing that the victim SU is interfering a PU's transmission. Since FCC regulations require a SU to leave the channel within 2 seconds upon a PU's arrival [7], if the perpetrator corrupts the victim SU's transmission

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

long enough, the victim SU will leave the channel and hop to the next available channel. It reduces the channel availability experienced by the victim SU.

Furthermore, in CR-based networks, the cross-layer nature of some networking protocols may create a new degree of vulnerability, because the coupling of multiple layers entails that the decisions made in one layer can be altered by changing the dynamics of other layers. In this paper, we propose such an attack under which an attacker can manipulate the routing decisions in the network layer by employing off-sensing attack as the front-end attack to change the channel availability in lower layers. As a result, the attacker can influence the traffic flow traversing around it and direct them to a target node (i.e., route manipulation). In particular, the perpetrator will create a Denial-of-Service (DoS) situation for the victim SU node and divert the traffic flow which initially should go through the victim SU. We call it as off-sensing DoS (OS-DoS) attack. With the careful selection of which neighboring SU to perform the OS-DoS attack on, the perpetrator can direct the diverted traffic flow to a designated target node. We name this *off-sensing and route manipulation* (OS-RM) attack, a cross-layer attack. To the best of our knowledge, no existing efforts have been made to exploit the vulnerability in spectrum sensing mechanisms and cross-layer routing protocols together in a CR-based network in order to manipulate traffic flow.

In this paper, we study the effect of OS-RM attack in CR enabled WMNs under different scenarios. The main contributions of this paper are summarized in the following:

1. We propose an off-sensing attack based cross-layer attack, where the OS-DoS attack is used to exploit the cross-layer dependency in the routing protocol. Here, the perpetrator can influence a significant portion of the network traffic flow around it and divert them to a designated target node. To the best of our knowledge, this work is the first to study a cross-layer route manipulation attack in CR-based networks, without advertising false routing updates.
2. We propose an intelligent attacker model where an attacker will use the Hidden Markov Model (HMM) based learning technique to learn channel parameters of PUs and then apply the learned knowledge to strengthen its malicious actions. To the best of our knowledge, even though the concept of learning has been used widely in defense techniques, it has never been considered in the domain of CR based networks where an attacker can also capitalize the knowledge around it.

Note that this paper focuses on the details and impact of the OS-RM attack. Countermeasure of this attack is out of this paper's scope. The rest of this paper is organized as follows. In Section II, conventional cross-layer attacks and their defenses are reviewed briefly. Then in Section III, the system model that is considered in this paper is explained. The details of our proposed OS-RM attack are discussed in Section IV. Simulation results are shown and discussed in Section V, followed by the conclusions in Section VI.

II. Related Work

The presence of a PUE attacker can harshly affect the operations of a CR-based network. Numerous defense strategies have been proposed [8]–[13] and all of them consider that the SU would sense the PU's transmission. However, a perpetrator can ingeniously avoid the sensing interval of the victim SU and attack by interfering the victim SU's transmission in off-sensing intervals [6]. We consider this scenario in our proposed work.

In recent years, some cross-layer attacks have been proposed in the CR based networks. Cross-layer attacks have proven to be more detrimental than single-layer based attacks, due to their immunity to the single-layer based defense strategies. In [14], the coordination of two cross-layer attacks at the PHY layer and MAC layer is studied. The use of PUE attack as an auxiliary attack in order to degrade the throughput performance of TCP has been studied in [15]. In [16], the authors propose a MAC-TCP cross-layer attack where an attacker periodically preempts itself to use the shared channel and impacts the TCP performance by creating large variations in round-trip-time (RTT). Though the study of cross-layer attacks in terms of PHY-MAC-Transport layer has gained significant attention, very few efforts have been focused on security vulnerabilities in the network layer. A network layer attack in CR-based networks named routing-toward-primary-user (RPU) is proposed in [17], where a malicious node intentionally directs a large amount of traffic toward the PUs, aiming to cause interference to them. However, this is not a cross-layer attack and the perpetrator is an active participant in the attack, hence, less difficult to identify. In Hammer and Anvil attack [18], a jamming aided cross-layer attack is proposed in the multihop infrastructureless network. Nevertheless, a CR-based network is inherently immune to jamming attacks due to their ability to change operating channels dynamically.

Neither of the attacks mentioned above have considered an intelligent attacker who can gather information and learn about the whole network by leveraging the control information flowing in the collaborative CR-based network. With this knowledge, an attacker can conduct more sophisticated attacks with less risk of being flagged.

III. System Model

In this section, we provide an outline of the assumptions made for the basic functionalities of the PHY, MAC, and network layers in our considered CR-WMNs.

A. Primary User and Secondary User Model

We consider totally M homogeneous channels each with a fixed bandwidth for the PUs and SUs in the network, and N CR-WMRs trying to opportunistically access the channels. Each PU randomly selects a channel to access. An SU is allowed to access a channel when it senses no PU is using it. During the transmission, if an SU senses the channel busy, it stops transmitting on that channel and performs a spectrum handoff. Each SU is equipped with only one radio for spectrum sensing, control information exchange, and data transmission. Each PU alternates between the ON and OFF state according

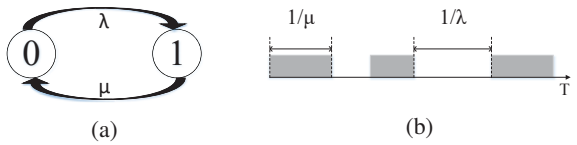


Fig. 1: PU activity model. (a) Transition rate of the Markov chain; (b) PU activity in the time line.

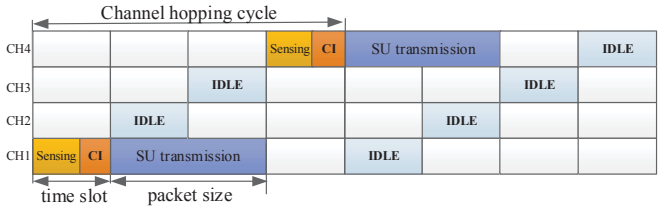


Fig. 2: Network coordination scheme

to a continuous-time Markov process. In Fig. 1, let λ denote the transition rate from the OFF to ON state, and let μ denote the transition rate from the ON to OFF state. Thereby, the mean sojourn time in the ON and OFF state is $1/\mu$ and $1/\lambda$, respectively, and both follow the exponential distribution.

B. Network Coordination Scheme

Rendezvous is a pre-requisite step before two SUs can communicate and exchange control information with each other in the absence of a dedicated CCC. A successful rendezvous happens when both transmitting and receiving SUs are on the same channel and have completed a successful handshake between them, e.g., a Request-to-Send/Clear-to-Send (RTS/CTS) exchange.

We consider the common frequency-hopping as the network coordination scheme [19], [20] which means that the channel hopping pattern is the same for all SUs. Fig. 2 illustrates the operation of the common frequency-hopping-based network coordination. We consider a time-slotted system. Each time slot consists of a sensing interval (sensing) and a contention interval (CI) with the transmission of an RTS/CTS pair. When there is no packet in the buffer of an SU, it keeps hopping through the channels from one time slot to another based on the predetermined common channel-hopping pattern.

We adopt the MAC model from [21], [22] for network coordination. Whenever a SU has a packet to send, it first senses the channel. If the channel is idle, the SU chooses a random number between 0 and $CW - 1$ (in terms of mini-slots) as its backoff time to avoid contention on the channel. If it hears no RTS before the backoff time runs out, it sends an RTS on the channel. Otherwise, it saves the remaining time in the backoff timer and will try to resend the RTS in the next time slot. After sending an RTS, the source SU waits for the CTS from the intended SU receiver. If the RTS sender fails to receive a CTS, it means the RTS/CTS exchange has failed in this slot and the source SU will continue the same process in the next time slot. After a successful RTS/CTS exchange, both SUs stop channel-hopping and start the data transmission on the same channel. After a successful transmission, both SUs start channel-hopping again by following the common hopping sequence. Meanwhile, all other SUs keep hopping through the

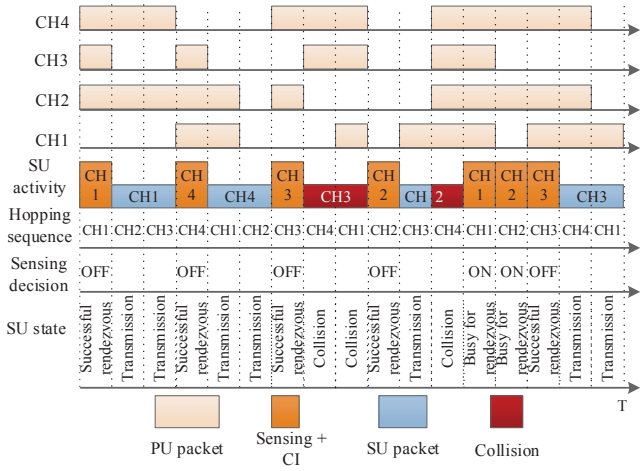


Fig. 3: An illustration of the network coordination with an ON/OFF PU model.

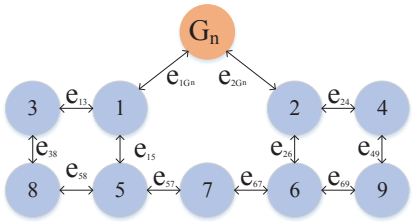


Fig. 4: An illustration of a network graph.

channels. To better illustrate the activity of a SU under the coordination scheme, we provide an example in Fig. 3. In this example, we consider that the SU always has a packet in its buffer, wins contentions and a SU packet length is two time slot long.

C. Routing Scheme

Many routing protocols have been proposed for CR-based networks [23]–[26]. In all these papers, spectrum availability has been given the highest weight for routing decisions. Therefore, it is clear that CR-based routing protocols consider spectrum availability as a significant cost metric.

Our focus in this paper is not to propose a new routing protocol. Instead, we adopt a link-state based routing protocol with channel availability as the only cost metric for routing decisions. Our goal is to show the impact of our proposed attack on routing performance. In our CR-WMN, CR-WMRs calculate their link cost periodically with a period of 'Δ' and broadcast it. We also define an activity threshold τ (in Δ interval) above which a PU will be considered busy and hence the channel is not available. Along with cost, nodes also share their available channel list (ACL). For the calculation of the shortest path from a CR-WMR to the gateway, we consider the CR-WMN as an undirected graph $G = \{V, E\}$, called a connectivity graph. Each node $i \in V = \{1, \cdots, N\}$ represents a CR-WMR, which is characterized by a circular transmission range and an interfering range. Each edge E represents the connectivity between neighboring CR-WMRs and the edge cost is characterized by the spectrum availability. Fig. 4 illustrates a network graph with 9 nodes and a gateway (G_n). Link cost between node i and j is defined as e_{ij}.

IV. Proposed Off-Sensing and Route Manipulation (OS-RM) Attack Model

In reality, it is very unlikely for one to take control of a significant portion of the CR-WMRs in a CR-WMN without being flagged. However, under our proposed attack model, without even compromising a significant amount of routers, the perpetrator can still have the control over a significant portion of traffic flow around him. This can be done by exploiting and taking advantage of the many cross-layer routing protocols in CR enabled networks, where affecting lower layers can result in influencing decisions in the network layer.

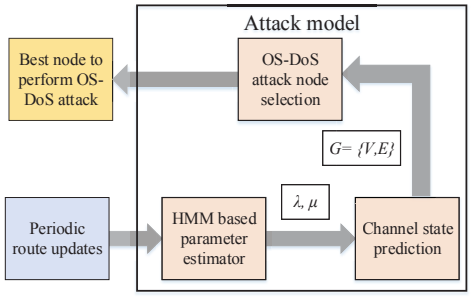


Fig. 5: Proposed attack model.

The configuration of the proposed HMM-based system for the OS-RM attack is shown in Fig. 5. Time is slotted into a duration of routing updates Δ. Therefore, we consider a discrete-time model, where the time variable takes values in $\{0, 1, ..., T\}$. The attacker has a separate HMM block for each channel. The input to the system at time t consists of the routing updates received from the neighboring nodes.

The attacker model consists of three components: OS-DoS attack node selection, channel state prediction, and HMM-based channel parameter estimator. The OS-DoS attack node selector chooses the best node as the victim node based on the updated network graph $G = \{V, E\}$. The output of the system consists of the best neighbor to perform the OS-DoS attack, in order to divert traffic flow through the target node. The attacker updates the network graph G depending on the adjacency list (i.e., neighboring list of the SUs) and prediction of the future state of the channels, and the HMM-based channel parameter estimator facilitates to estimate the channel activity based on the routing updates.

We consider the frequency of routing updates comparable to the frequency of channel status change. Also, due to computational and physical efforts by the attacker, we consider a constant delay between when the routing update arrives and the attacker conducts an OS-RM attack without learning. We will see that this delay degrades the attack performance and hence it indicates the importance of predicting network conditions beforehand to counteract the effect of the delay.

A. OS-DoS Node Selection

The victim of the OS-DoS attack will be disconnected from the network (or has a very high cost to use it) and traffic flows that have been going through it, will switch to the next best available route. The performance of the OS-RM attack depends on the right neighbor node to perform the OS-DoS attack on. Depending on the predicted network graph, the attacker finds the neighboring node whose traffic flow is most likely to traverse through the target node, if attacked. Here, the attacker's goal is to choose a neighbor in such a way that the rebound effect will divert most traffic flows to the target node.

The attacker will use a shortest-path algorithm (i.e., Dijkstra's algorithm) to figure out the best route for each node in the network to reach the gateway. At every step, the attacker first calculates the number of routers choosing the target router as a forwarder, under no attack (i.e., successor routers, π_{max}). Then, it finds the best neighbor router to perform the OS-DoS attack which will maximize its objective. It does it by measuring what would happen if it attacks a neighbor. If there is no neighbor that offers $\pi > \pi_{max}$, it will not conduct the OS-DoS attack and wait for the next update to come. Here, π is the number of successor nodes, under attack. Algorithm 1 and 2 show the pseudocode for calculating the successor CR-WMRs of the target CR-WMR and OS-DoS node selection, respectively. Next, we will discuss how an attacker can update the network graph G. Here, the target node and the gateway node are denoted as T_n and G_n, respectively.

Algorithm 1 Calculating the number of nodes that has the target node in their forwarding set to the gateway

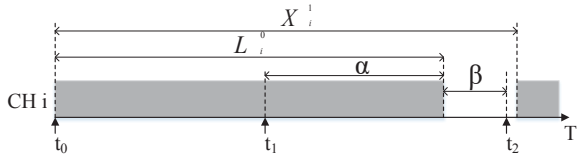
Input: G, T_n, G_n
Result: T_n's successor node quantity ϕ_{T_n}
function ComputeSuccessors(G, T_n, G_n)
$\phi_{T_n} = 0$;
for $i = 1 : N$ **do**
  Use Dijkstra's algorithm to calculate the shortest path to the gateway, $P_i = \{i, \cdots$ forwarding nodes $\cdots, G_n\}$
  if $T_n \in P_i$ **then**
    $\phi_{T_n} = \phi_{T_n} + 1$;
  end
end
return ϕ_{T_n};

Algorithm 2 Selecting the best node to perform OS-DoS attack

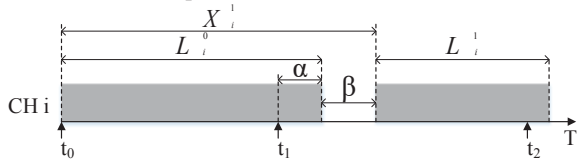
Input: G, T_n, G_n
Result: *OS-DoS node*
π_{max}= ComputeSuccessors(G, T_n, G_n);
OS-DoS node = *empty*;
for i = 1 : *all the neighbors* **do**
  Detach the neighbor i from G ▷ i= neighbor index
  Update network graph, $G' = \{V', E'\}$ ▷ $i \notin V', (\cdot, i) \notin E'$
  π= ComputeSuccessors(G', T_n, G_n);
  if $\pi > \pi_{max}$ **then**
    $\pi_{max} = \pi$;
    OS-DoS node $= i$;
  end
end
if *OS-DoS node* $\neq$ *empty* **then**
  Perform OS-RM attack
else
  Wait for the next update
end

B. Channel State Prediction

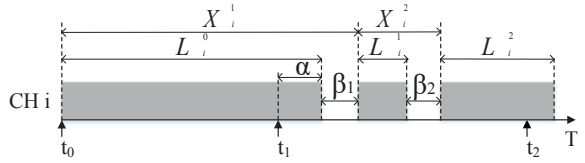
Channel state predictor assists in updating the network graph G in each period, based on routing updates. In this



(a) No PU packet arrives between t_1 and t_2



(b) Only one PU packet arrives between t_1 and t_2



(c) Two PU packets arrive between t_1 and t_2

Fig. 6: The PU activity on channel i; $N_i(t_1) = 1$.

section, we propose the prediction model to forecast future channel activity to update the network graph.

By utilizing the periodic routing update, an attacker can make predictions of the channel availability before the next route update arrives. Based on the prediction results, an attacker decides whether to change the link costs or not. We propose two criteria for determining whether the channel should be considered busy or idle: 1) the predicted probability that the channel is busy or idle and 2) the expected length of the activity or inactivity.

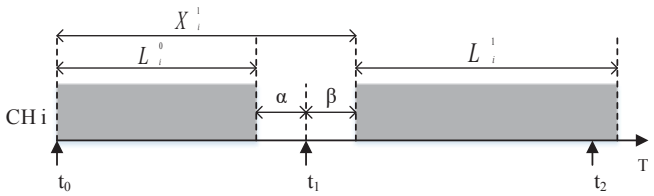
In Fig. 6, t_0 represents the last moment PU becomes active, t_1 represents the last moment route update arrives, and t_2 represents the expected moment of the next route update. Fig. 6 shows the PU traffic activity on channel i, where X_i^k represents the inter-arrival time of the kth packet. We denote $Y(t_2)$ as the number of PU packets that arrive between t_1 and t_2 and $N_i(t_2)$ as the status of the channel at time t_2, which is a binary variable between 0 and 1 representing the idle and busy state, respectively.

In the following, we calculate the probability that the channel state is active upon the next route update. All the figures are normalized to routing update length Δ. As shown in Fig. 6(a), where $N_i(t_1) = 1$, the probability that the next channel state will be active and no PU packet arrives between t_1 and t_2 is

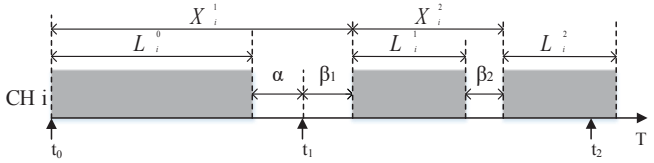
$$\begin{aligned} &Pr\{N_i(t_2) = 1, Y(t_2) = 0\} \\ &= Pr\{X_i^1 > t_2 - t_0\} Pr\{\alpha > \tau\} \\ &= Pr\{X_i^1 > t_2 - t_0\} Pr\{L_i^0 - (t_1 - t_0) > \tau\}, \end{aligned} \tag{1}$$

where $L_i(k)$ denotes the length of the kth new PU packet in channel i and τ represents the activity threshold of PU. $X_i(k)$ and $L_i(k)$ depend on the channel parameters λ_i and μ_i.

As shown in Fig. 6(b), the probability that the channel state will be active and only one PU packet arrives between t_1 and



(a) Only one PU packet arrives between t_1 and t_2



(b) Two PU packets arrive between t_1 and t_2

Fig. 7: The PU activity on channel i; $N_i(t_1) = 0$.

t_2 is

$$\begin{aligned} &Pr\{N_i(t_2) = 1, Y(t_2) = 1\} \\ &= Pr\{\beta < 1 - \tau\} Pr\{\alpha + L_i^1 > \tau\} \\ &= Pr\{X_i^1 - L_i^0 < 1 - \tau\} Pr\{\alpha + L_i^1 > \tau\}. \end{aligned} \tag{2}$$

Similarly, in Fig. 6(c), the probability that channel i is active and two packets come between t_1 and t_2 is,

$$\begin{aligned} &Pr\{N_i(t_2) = 1, Y(t_2) = 2\} \\ &= Pr\{\beta_1 + \beta_2 < 1 - \tau\} Pr\{\alpha + (L_i^1 + L_i^2) > \tau\} \\ &= Pr\{X_i^1 + X_i^2 - (L_i^0 + L_i^1) < 1 - \tau\} \\ &\quad Pr\{\alpha + L_i^1 + L_i^2 > \tau\}. \end{aligned} \tag{3}$$

Assume that U is the maximum number of PU packets that could come between t_1 and t_2. Hence, the probability of having the channel active and arriving h ($h \in [1, U]$) PU packets is

$$\begin{aligned} &Pr\{N_i(t_2) = 1, Y(t_2) = h\} \\ &= Pr\{\sum_{k=1}^{h} \beta_k < 1 - \tau\} Pr\{\alpha + \sum_{k=1}^{h} L_i^k > \tau\} \\ &= Pr\{\sum_{k=1}^{h} X_i^k - \sum_{k=0}^{h-1} L_i^k < 1 - \tau\} Pr\{\alpha + \sum_{k=1}^{h} L_i^k > \tau\}. \end{aligned} \tag{4}$$

Therefore, the probability that channel i is active at time t_2 can be obtained by,

$$\begin{aligned} &Pr\{N_i(t_2) = 1 | N_i(t_1) = 1\} \\ &= Pr\{X_i^1 > t_2 - t_0\} Pr\{L_i^0 - (t_1 - t_0) > \tau\} \\ &+ \sum_{h=1}^{U} \left[Pr\{\sum_{k=1}^{h} X_i^k - \sum_{k=0}^{h-1} L_i^k < 1 - \tau\} Pr\{\alpha + \sum_{k=1}^{h} L_i^k > \tau\} \right]. \end{aligned} \tag{5}$$

Likewise, in Fig. 7(a), where $N_i(t_1) = 0$, the probability that next channel status will be active and one PU packet arrives between t_1 and t_2 is

$$\begin{aligned} &Pr\{N_i(t_2) = 1, Y(t_2) = 1\} \\ &= Pr\{\beta < 1 - \tau\} Pr\{L_i^1 > \tau\} \\ &= Pr\{X_i^1 - L_i^0 - \alpha < 1 - \tau\} Pr\{L_i^1 > \tau\}. \end{aligned} \tag{6}$$

Similarly, in Fig. 7(b), the probability that channel i is active and two packets come between t_1 and t_2 is,

$$\begin{aligned} &Pr\{N_i(t_2)=1, Y(t_2)=2\} \\ &= Pr\{\beta_1+\beta_2<1-\tau\}Pr\{(L_i^1+L_i^2)>\tau\} \\ &= Pr\{X_i^1+X_i^2-(L_i^0+L_i^1)-\alpha<1-\tau\} \\ &\quad Pr\{L_i^1+L_i^2>\tau\}. \end{aligned} \tag{7}$$

Therefore the probability that channel i is active at time t_2 can be obtained by,

$$\begin{aligned} &Pr\{N_i(t_2)=1|N_i(t_1)=0\} \\ &= Pr\{X_i^1-L_i^0-\alpha<1-\tau\}Pr\{L_i^1>\tau\} \\ &+\sum_{h=1}^{U}\left[Pr\{\sum_{k=1}^{h}X_i^k-\sum_{k=0}^{h-1}L_i^k-\alpha<1-\tau\}Pr\{\sum_{k=1}^{h}L_i^k>\tau\}\right]. \end{aligned} \tag{8}$$

Thus, if the channel statistics (e.g., λ and μ) are known, the predicted probabilities can be calculated. Therefore, based on the prediction, the policy that we consider the channel as active, when

$$Pr\{N_i(t_2)=1\}>\Gamma, \tag{9}$$

where Γ is the threshold above which the channel is considered active by the predictor model. After making channel decisions, the attacker will calculate the corresponding link costs.

However, learning the channel statistics requires significant efforts and hence, we design and propose a HMM based technique to estimate the channel parameters λ and μ.

C. HMM based Parameter Estimator

A slotted discrete-time model is used for the channel activity. The decision on whether a channel is busy or not is made based on the channel activity during the last period. If the channel activity exceeds the given threshold τ, then it is assumed to be in the ON state or otherwise OFF.

We first present the structure of the HMM and then we give a brief introduction of the forward-backward procedure in Baum-Welch (BW) algorithm [27]. Finally, by analyzing the estimated parameters, we calculate the channel parameters.

1) Hidden Markov Model: A Hidden Markov process is a Markov process consisting of two states, where X is the hidden process that is never observable and Z is the observation process that can be seen by the observers (i.e., the OS-RM attacker). X_t and Z_t denote the hidden state and observation state at time t, respectively. The hidden process follows a Markov process with a finite number of states and the observable process is another probabilistic function which generates *symbols* based on the hidden states. The set of symbols comes from a defined *alphabet* A. In our case, $A=\{0,1\}$ (i.e., 0 = OFF and 1 = ON).

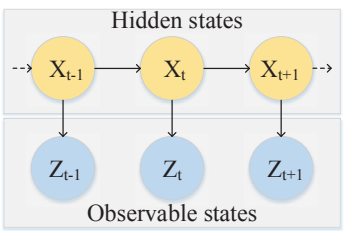


Fig. 8: The Hidden Markov model.

The general concept of an HMM is illustrated in Fig. 8. A system of discrete time is changing randomly from one state to another, within a finite state space S. In our case, the finite space $S=\{0,1\}$. The evolution of the hidden sequence $X_1, X_2, ..., X_T$ is hidden, which represents PU states. However, it can be expressed by a sequence of observed symbols from the alphabet A (i.e., $Z_t \in A$), which represents routing updates. In order to model the HMM, it is necessary to define the parameters first:

- Number of hidden states, $s=2$
- Number of symbols, $a=2$
- Initial state distribution, $\pi=\{\pi_i\}$, where $i=0,\cdots,s-1$
- One-step state transition probabilities, $P=p_{ij}$, where $i,j=0,\cdots,s-1$
- Symbol emission probability, $B=b_j(k)$, where $j=0,\cdots,s-1$ and $k=0,\cdots,a-1$

Therefore, the one-step state transition probability is

$$\begin{aligned} &Pr(X_t=j|X_{t-1}=i, X_{t-2}=i_{t-2},\cdots,X_2=i_2, X_1=i_1) \\ &\quad = Pr(X_t=j|X_{t-1}=i) \\ &\quad = p_{ij}, \end{aligned} \tag{10}$$

where, $i_1, i_2, .., i_{t-2}, i, j \in \{0,1\}$ and $t \geq 2$. And the emission probability is

$$b_j(k)=Pr(Z_t=k|X_t=j). \tag{11}$$

The BW algorithm is an iterative approach to estimate the HMM parameters $\eta=[\pi,P,B]$ such that the $Pr(Z|\eta)$ is maximized. To estimate the parameters, we define the following parameters:

- Forward probability, $\alpha_t(i)=Pr(Z_1,Z_2,\cdots,Z_t,X_t=S_i|\eta)$, for $S_i\in\{0,1\}$
- Backward probability, $\beta_t(i)=Pr(Z_{t+1},Z_{t+2},\cdots,Z_{T-1},Z_T,X_t=S_i|\eta)$, for $S_i\in\{0,1\}$
- Estimate of state transitions, $\gamma_t(i,j)=Pr(X_t=S_i,X_{t+1}=S_j|Z,\eta)$, for $S_i,S_j\in\{0,1\}$. It represents the probability of being in state S_i at instant t and in state S_j at instant $t+1$, given the observation sequence Z and the model parameters $\eta=[\pi,P,B]$
- Estimate of the state at each observation, $\delta_t(i)=Pr(X_t=S_i|Z,\eta)$, for $S_i\in\{0,1\}$. It represents the probability of being in state S_i at instant t, given the observation sequence Z and the model parameters $\eta=[\pi,P,B]$

The estimation variables for the HMM parameters are expressed in terms of $\gamma_t(i,j)$ and $\delta_t(i)$:

$$p_{ij}=\frac{\sum_{t=1}^{t=T-1}\gamma_t(i,j)}{\sum_{t=1}^{t=T-1}\delta_t(i)}. \tag{12}$$

$$b_j(k)=\frac{\sum_{t=1,Z_t=k}^{t=T}\delta_t(j)}{\sum_{t=1}^{t=T}\delta_t(j)}. \tag{13}$$

$$\pi_i=\delta_1(i). \tag{14}$$

In (12) the numerator represents the expected number of transitions from state S_i to state S_j over the interval $T-$

1, while the denominator represents the expected number of times a transition happens from state S_i. The numerator in (13) represents the expected number of transitions from state S_j at which symbol k is observed. In (12)-(14), $\gamma_t(i,j)$ and $\delta_t(i)$ are calculated as follows:

$$\gamma_t(i,j) = \frac{\alpha_t(i)p_{ij}b_j(Z_{t+1})\beta_{t+1}(j)}{Pr(Z|\eta)}. \quad (15)$$

$$\delta_t(i) = \sum_{all\ S_j \in \{0,1\}} \gamma_t(i,j). \quad (16)$$

The forward and backward probabilities in the above equations are calculated recursively as follows:
Initialization:

$$\alpha_1(i) = \pi_i b_i(1), \quad 0 \leq i \leq s-1. \quad (17)$$

$$\beta_t(i) = 1, \quad 0 \leq i \leq s-1. \quad (18)$$

Recursion:

$$\alpha_{t+1}(j) = \left[\sum_{i=0}^{s-1} \alpha_t(i)p_{ij}\right] b_j(Z_{t+1}). \quad (19)$$

$$\beta_t(i) = \sum_{j=0}^{s-1} p_{ij}b_j(Z_{t+1})\beta_{t+1}(j). \quad (20)$$

The recursion process terminates when $Pr(Z|\eta)$ maximizes, which is the probability of observing the sequence Z given the parameter $\eta = [\pi, P, B]$.

$$Pr(Z|\eta) = \sum_{i=0}^{s-1}\prod_{t=1}^{T} \alpha_t(i). \quad (21)$$

2) Analysis of PU Activity: In this section, we need to extract the PU activity from the estimated HMM parameters $\eta = [\pi, P, B]$. To do this, we first introduce a new set of PU parameters, $\theta = [\lambda, \mu]$, where λ means the traffic arrival rate and μ means the traffic departure rate. From our network model, the length of the ON and OFF state are exponentially distributed. In [28], a useful method to compute the state transition rate matrix from the state transition probability matrix is provided. We denote the transition rate matrix as Q and

$$Q = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}. \quad (22)$$

As described in η, P is the one-step state transition probability matrix. We know that $P = \exp(Q\Delta)$ and $Q = \log(P)/\Delta$, where Δ is the route update period. However, the computational procedure is cumbersome and $\log(\cdot)$ has a limitation when P has a non-positive eigenvalue. Therefore, we adopt the mapping approach introduced in [28], which provides an easier computational approach and provides enough degree of accuracy. If the two-dimensional transition rate matrix is the form shown in (22), then the transition probability matrix is:

$$P = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} = \begin{pmatrix} \exp^{-\lambda\Delta} & 1-\exp^{-\lambda\Delta} \\ 1-\exp^{-\mu\Delta} & \exp^{-\mu\Delta} \end{pmatrix}. \quad (23)$$

In (23), the relation between P and Q unfolds the relationship between η and θ.

V. Performance Evaluation

We evaluate the impact of the OS-RM attack by conducting simulations in Matlab. We consider a grid size distribution of 25 CR enabled nodes, with 24 being CR-WMRs and a gateway (Fig. 9). The attacker and the target node are colored with red and green color, respectively. The gateway has three neighboring CR-WMRs via which other routers can communicate with the gateway. In reality, traffic is not uniformly distributed among these three CR-WMRs due to their different spectrum availability. We consider a uniform distribution of PUs in the network. Parameters of our simulations are listed in Table I.

TABLE I: Simulation Parameters

Simulation area	1000x1000
Simulation time	50 seconds
Training time	25 seconds
SU sensing range	200
The number of PUs	10
The number of SUs	25
Bandwidth	2 Mbps
The size of (RTS+CTS)	160 + 112 bits (802.11b/g)
Sensing duration	1 ms (802.22)
SU traffic	$\rho = \lambda_s/\mu_s$= 0.05 $\sim$ 0.25
SU packet size	750 bytes
Number of channels	10

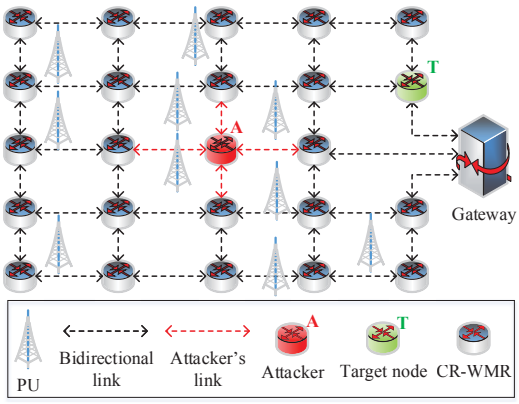
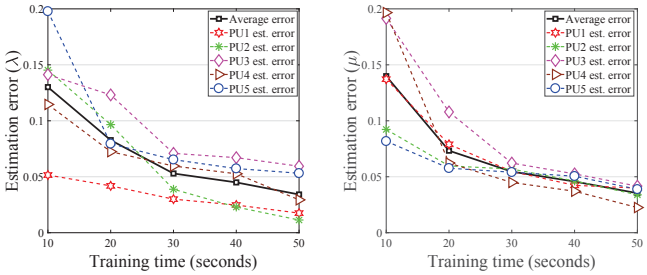


Fig. 9: Simulation scenario.

A. HMM Estimation

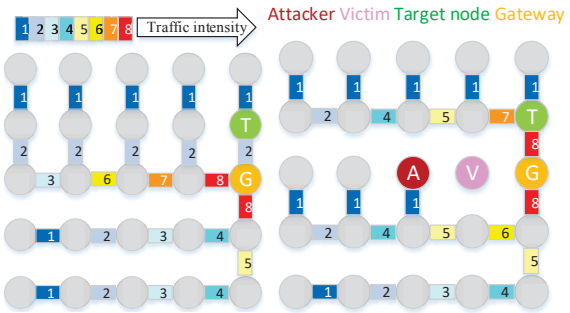
The performance of the OS-RM attack relies significantly on how accurately HMM-based estimators can estimate the parameters of PUs in the network. Furthermore, the length of a training sample is instrumental to the learning performance. In Fig. 10, we can observe the trend of estimation error over the time for packet arrival rate (λ) and service rate (μ). Estimation errors reduce to below 4% when the estimator is trained to 50 seconds.

In our simulations, we train the HMM estimator with 25 seconds of data and observe the impact of the attack for the next 25 seconds without changing the PU activity rate. Nevertheless, in reality, the PU activity rate is not going to be constant all the time and the HMM estimator should reestimate to track changes. The optimal training time length based on the traffic change rate is out of this paper's scope. In the future, we plan to propose a strategy for the attacker in a time-varying PU network.



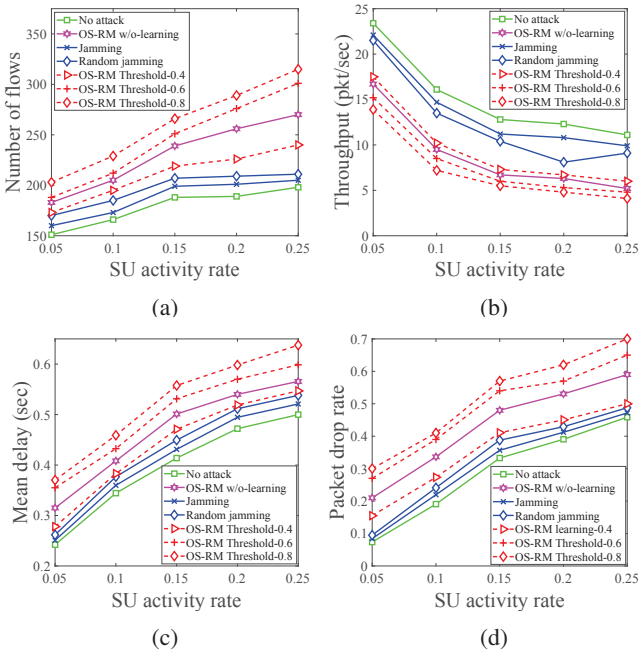
(a) Estimation error of λ. (b) Estimation error of μ.

Fig. 10: HMM estimation performance.



(a) (b)

Fig. 11: Traffic heat map. (a) no attack; (b) OS-RM attack.



(a) (b) (c) (d)

Fig. 12: Impact of lower-layer attacks on route manipulation. (a) Number of traffic flows; (b) Throughput; (c) Mean dealy (d) Packet drop rate.

B. Impact on Traffic Flow

In Fig. 11 (color and number coded), we observe changes in traffic flow due to the rebalancing effect caused by the OS-DoS attack on the victim node. Without attack, two neighboring CR-WMRs carry most of the traffic (Fig. 11(a)) except the target node. However, with the OS-RM attack, we can see that a portion of previous routes are disrupted (Fig. 11(b)). As a result, traffic flows change directions and a few nodes who were carrying less traffic are exposed to higher traffic load now. Most significant change in traffic is observed in the target node. This strategy works as the driving force to maneuver traffic to any node an attacker wants. Though we discussed only about diverting traffic towards a particular node, the same kind of strategy can be employed to divert traffic from one.

C. Impact on Network Performance

We compare the impact of lower-layer attacks, e.g., conventional jamming, random jamming, OS-RM attack without learning, and OS-RM attack with learning, used as an auxiliary attack in an effort to manipulate routes. In Fig. 12(a)-(d), we compare the impact of these front-end attacks with an increasing SU activity. From Fig. 12(a), we can observe the increased number of traffic flows going through the target node. Though the jamming attack can also influence traffic flows, it is less significant as compared to the OS-RM attack. In the jamming attack, all the nodes within the radio range of the jammer get affected, hence, the traffic flows disperse in the whole network. Moreover, it is inefficient to use the jamming strategy due to the high energy required by the jammer. Furthermore, as the attacker is an authorized network entity and has the similar power requirement as other entities, it is unrealistic to perform jamming. However, unlike the jamming attack, an OS-DoS attack can be performed on an individual node of choice. Thus, we can observe more than 50% increase in traffic flows to the target node.

In Fig. 12(b)-(d), we can observe the change in key performance metrics of the flows going through the target node (i.e., throughput, delay, and packet drop). If the perpetrator's objective is to increase congestion at the target node, then from Fig. 12(b)-(c), it is quite evidential that this attack reduces throughput and increases delay experienced by the flows going through the target node. The effect of delay stems from the queuing delay in intermediate nodes. In addition, a virtual blackhole creates in the network as more packets are being dropped. The increase in packet drop stems from the packet drop in intermediate nodes due to the timeout and blocking of new sessions. From Fig. 12, we can observe the performance improvement by implementing learning strategy of the attacker when $\Gamma \geq 0.6$.

D. Influence on Traffic vs. Distance

We also observe that the attacker is more influential when it is situated higher up in the routing tree (gateway is the root of the tree). In another word, the attacker is more influential when more number of traffic flows go around it. In Fig. 13, we can observe that the number of traffic flows actually increases when the distance between the attacker and the target node changes from 1-hop to 2-hop, which is counterintuitive to what we just mentioned. However, when the attacker is a direct neighbor to the target node, it cannot perform the OS-DoS attack on the target node. Therefore, the attacker has one less neighbor to maneuver the neighbor's traffic flows and hence the decrease in the number of flows. Therefore, we can deduce that the attacker is more potent when it is 2-hop away from

the target node.

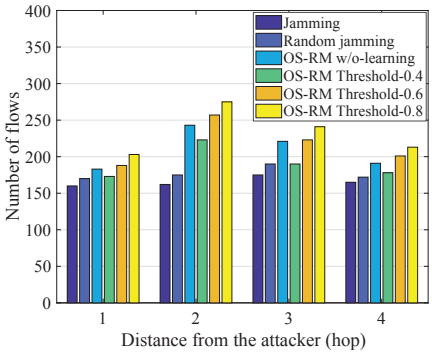


Fig. 13: Impact on traffic flows vs. distance between the attacker and target node.

Depending on the end objective of the attacker, the impact of the OS-RM attack can affect other network layers also. In our proposed attack, the target node could be actually a pre-compromised node to perform wormhole attacks, black-hole attacks or perhaps a benign node to create network congestion. From the above observations, one could imagine the atrocities an attacker can perpetuate if it achieves a significant amount of control over the traffic flow.

VI. Conclusion

In this paper, we proposed a cross-layer route manipulation attack in CR-WMNs, namely OS-RM attack. In this attack, we discussed how the off-sensing attack can be weaponized as an aid to influence routing decisions in the network layer. We considered the perpetrator as an intelligent entity and it estimates necessary network information through learning.

We illustrated a general model of the attack and analyzed through extensive simulations how to coordinate the OS-RM attack in order to achieve the best-attacking result. Our analysis and observations not only shed light on a new kind of threats to the CR-based network, but also provide some insightful findings on how to design cross-layer protocols.

References

[1] N. Bouabdallah, B. Ishibashi, and R. Boutaba, "Performance of cognitive radio-based wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 122–135, 2011.

[2] W. Zhao and J. Xie, "IMeX: intergateway cross-layer handoffs in internet-based infrastructure wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1585–1600, 2012.

[3] A. P. Subramanian, M. M. Buddhikot, and S. Miller, "Interference aware routing in multi-radio wireless mesh networks," in *Proc. IEEE Workshop on Wireless Mesh Networks*, 2006, pp. 55–63.

[4] X. Liu and J. Xie, "A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2014, pp. 2085–2093.

[5] R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *Proc. IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, 2006, pp. 110–119.

[6] M. Hossain and J. Xie, "Impact of off-sensing attacks in cognitive radio networks," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2017.

[7] T. Bansal, B. Chen, and P. Sinha, "Fastprobe: Malicious user detection in cognitive radio networks through active transmissions," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2014, pp. 2517–2525.

[8] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, 2008.

[9] S. Anand, Z. Jin, and K. Subbalakshmi, "An analytical model for primary user emulation attacks in cognitive radio networks," in *Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2008, pp. 1–6.

[10] Z. Yuan, D. Niyato, H. Li, J. B. Song, and Z. Han, "Defeating primary user emulation attacks using belief propagation in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 10, pp. 1850–1860, 2012.

[11] C. Chen, H. Cheng, and Y.-D. Yao, "Cooperative spectrum sensing in cognitive radio networks in the presence of the primary user emulation attack," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2135–2141, 2011.

[12] J. McNair, T. Tugcu, W. Wang, and J. L. Xie, "A survey of cross-layer performance enhancements for Mobile IP networks," *Transactions on Computer Networks*, vol. 49, no. 2, pp. 119–146, 2005.

[13] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[14] W. Wang, Y. Sun, H. Li, and Z. Han, "Cross-layer attack and defense in cognitive radio networks," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2010, pp. 1–6.

[15] J. Hernandez-Serrano, O. León, and M. Soriano, "Modeling the lion attack in cognitive radio networks," *EURASIP Journal on Wireless Communications and Networking*, 2011,1.

[16] D. Nagireddygari, and P. T. Johnson , "MAC-TCP cross-layer attack and its defense in cognitive radio networks." in *Proc. 10th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 2014, pp. 71-78.

[17] Z. Yuan, Z. Han, Y. Sun, H. Li, and J. Song, "Routing-toward-primary-user attack and belief propagation-based defense in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1750–1760, 2013.

[18] L. Zhang and T. Melodia, "Hammer and anvil: The threat of a cross-layer jamming-aided data control attack in multihop wireless networks," in *Proc. IEEE Communications and Network Security (CNS)*, 2015, pp. 361–369.

[19] Y. Song and J. Xie, "Performance analysis of spectrum handoff for cognitive radio ad hoc networks without common control channel under homogeneous primary traffic," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2011, pp. 3011–3019.

[20] Y. Song and J. Xie, "ProSpect: A proactive spectrum handoff framework for cognitive radio ad hoc networks without common control channel," *IEEE Transactions on Mobile Computing*, vol. 11, no. 7, pp. 1127–1139, 2012.

[21] X. Liu and J. Xie, "A self-adaptive optimal fragmentation protocol for multi-channel cognitive radio ad hoc networks," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[22] X. Liu and J. Xie, "A slot-asynchronous MAC protocol design for blind rendezvous in cognitive radio networks," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 4641–4646.

[23] X. Huang, D. Lu, P. Li, and Y. Fang, "Coolest path: Spectrum mobility aware routing metrics in cognitive ad hoc networks," in *Proc. Distributed Computing Systems (ICDCS)*, 2011, pp. 182–191.

[24] I. Pefkianakis, S. Wong, and S. Lu, "SAMER: Spectrum aware mesh routing in cognitive radio networks," in *Proc. New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2008, pp. 1–5.

[25] K. R. Chowdhury and I. F. Akyildiz, "CRP: A routing protocol for cognitive radio ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 794–804, 2011.

[26] M. Youssef, M. Ibrahim, M.A. Latif, L. Chen, and A.V. Vasilakos, "Routing metrics of cognitive radio networks: A survey.," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 92–109, 2014.

[27] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[28] K. W. Choi and E. Hossain, "Opportunistic access to spectrum holes between packet bursts: A learning-based approach," *IEEE Communications Letters*, vol. 10, no. 8, pp. 2497–2509, 2011.