Jamming-Resistant Collaborative Broadcast Using Uncoordinated Frequency Hopping

Liang Xiao, *Associate Member, IEEE*, Huaiyu Dai, *Senior Member, IEEE*, and Peng Ning, *Member, IEEE*

Abstract—We propose a jamming-resistant collaborative broadcast scheme for wireless networks, which utilizes the Uncoordinated Frequency Hopping (UFH) technique to counteract jamming without preshared keys, and exploits node cooperation to achieve higher communication efficiency and stronger jamming resistance. In this scheme, nodes that already obtain the broadcast message serve as relays to help forward it to other nodes. Relying on the sheer number of relay nodes, our scheme provides a new angle for jamming countermeasure, which not only significantly enhances the performance of jamming-resistant broadcast, but can readily be combined with other existing or emerging antijamming approaches in various applications. We present the collaborative broadcast protocol, and analyze its successful packet reception rate and the corresponding cooperation gain for both synchronous and asynchronous relays for a snapshot scenario. We also investigate the full broadcast process based on a Markov chain model and derive a closed-form expression of the average broadcast delay. Simulation results in both single-hop and multihop networks indicate that our scheme is a promising antijamming technique in wireless networks.

Index Terms—Antijamming communication, collaborative broadcast, frequency hopping, wireless networks.

I. INTRODUCTION

BECAUSE of the broadcast nature of radio propagation, wireless networks are highly vulnerable to jamming attacks, where jammers aim at interrupting the ongoing legitimate information exchange by injecting replayed or faked signals into wireless media [1], [2]. Jamming-resistant broadcast is not only important for many safety-critical applications such as emergency alert broadcast and navigation signal dissemination, but

also critical for the distribution of important network information such as the public key and control information in wireless systems [3]–[6].

Jamming attacks are easily launched for wireless communications, and cannot be fully addressed through conventional cryptography. Spread spectrum techniques, including Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping (FH), have been commonly adopted to counteract jamming [7]. One key vulnerability for these conventional antijamming techniques is the requirement of preshared secret keys (such as spreading codes in DSSS or frequency hopping pattern in FH) at the senders and legitimate receivers [3], [4], [8], [9]. This requirement suffers from scalability concerns due to the need to distribute a preshared secret, and may not even be feasible in the face of network dynamics and compromised receivers [10]–[13].

There have been several recent efforts to address this problem, providing antijamming spread spectrum communication without preshared keys [3]–[6], [9]–[11], [13]–[16]. In particular, Uncoordinated Frequency Hopping (UFH) [3] and its variations [10], [14], [15] achieve frequency hopping communication without using any predefined hopping patterns. In these schemes, a broadcast message is divided into multiple short packets, and each packet is transmitted over a selected channel only known to the sender. Such rapid channel switching over a large frequency range effectively thwarts the jamming attempts.

In this paper, we propose a Collaborative UFH-based Broadcast (CUB) scheme to achieve higher communication efficiency and stronger jamming resistance than existing jamming-resistant broadcast schemes. The main idea is to allow the set of nodes that already receive the message to help broadcast, as all the nodes are expecting the same broadcast message. This process may start slowly, but as more and more nodes join the relaying, the broadcast process accelerates much like an avalanche.

This scheme exploits the node cooperation to enhance both the efficiency and the security. Unless all the channels are simultaneously blocked (assumed impossible for a fairly large spreading ratio), it is always possible for some nodes to obtain the message through unjammed channels. These nodes then relay it across more channels to increase the success rate of reception. With time on its side, our scheme is fundamentally more powerful than most recent attempts for antijamming broadcast [3]–[6], [9]–[11], [13]–[16].

To the best of our knowledge, this paper is the first collaborative UFH-based broadcast solution that does not require shared keys for jamming resistance. This solution can exploit the

spectral diversity and spatial diversity to improve the communication efficiency and jamming resistance. We give a detailed performance analysis for a snapshot scenario. The impact of synchronization error on the performance is discussed and the cooperation gain is provided under various broadcast scenarios. In addition, we model the whole broadcast process as a finite Markov chain and present a closed-form expression of the average broadcast delay. Simulation results regarding energy consumption and broadcast delay are also provided for both single-hop and multihop networks. Note that our approach is not restricted to UFH, and can be readily combined with other jamming countermeasures in various applications; we provide an illustrative example, collaborative uncoordinated DSSS (UDSSS) at the end of this paper.

The remainder of this paper is organized as follows. We review related work in Section II and introduce the network model and jamming model in Section III. We then investigate some key issues in the collaborative UFH-based broadcast in Section IV, and present the broadcast protocol in Section V. Next, we analyze its performance in a snapshot scenario in Section VI, and investigate the whole broadcast process in Section VII. Simulation results are provided in Section VIII. Finally, we discuss the collaborative UDSSS in Section IX and conclude in Section X.

II. RELATED WORK

Recently, there has been a series of promising research efforts on antijamming communication without preshared keys, including UFH [3], [10], [14], [15], UDSSS [9], [13], [16], and BBC [4].

In [3], the UFH approach was proposed to counteract jamming based on rapid channel switching over a large frequency range. On the down side, each packet has to be sent multiple times, due to the low communication efficiency resulting from the uncoordinated channel selection between the sender and legitimate receivers. To this end, a BMA scheme was proposed to improve the communication efficiency, incorporating error control coding and one-way authenticator [10]. In BMA, the erasure coding combined with a one-way authenticator based on bilinear maps can also efficiently address false data injection attacks by jammers. Some additional efficient packet verification methods were proposed in [14].

In spite of all these efforts, the UFH-based techniques still suffer from low communication efficiency. The relative throughput of the original UFH compared with coordinated FH is only on the order of 10^{-3} for a spreading ratio of 200 [3], and the approaches in [10] only reduce the communication latency up to one-half. To address this problem, in [15], the USD-FH scheme was proposed to further improve the efficiency and robustness, where the hopping pattern is conveyed through UFH to allow message transmission through coordinated FH.

The UDSSS technique provides an antijamming broadcast mechanism for DSSS-based systems, where each sender and receiver independently and randomly selects a spreading code sequence out of a large code set to transmit or receive [9]. Though UDSSS removes the need to have a preshared key between the sender and receivers, it is still vulnerable to intelligent reactive jammers. The UDSSS scheme was later extended to delayed key UDSSS by generating the spreading code sequence using a key

disclosed at the end of the message transmission, and was also integrated with UFH into a hybrid UFH-UDSSS scheme [11]. With a similar strategy to disclose index codes at the end of message transmission to indicate selected code sequence, RD-DSSS further uses permutation and multiple code sequences to defend against possible DoS attacks [16]. Delayed key UDSSS and RD-DSSS use delayed disclosure of certain data to determine the code sequence used for spreading. In DSD-DSSS [13], a content-based subset selection technique was proposed to protect the delayed disclosure of seed (key) against powerful intelligent real-time reactive jammers. Another scheme TREKS was proposed in [14] to use gradual disclosure of bits in a key to enable DSSS communication without preshared keys. The above DSSS-based antijamming communication techniques are complementary to the research presented in this paper.

Another line of works [8], [12], [17] is worth mentioning, which addresses the same topic, jamming-resilient communications in single-hop multichannel wireless networks without preshared secrets, from a computation-theoretic perspective. Relaying is considered in these works to facilitate message distribution. In [12], the authenticated pairwise message exchange problem is investigated, while in [8], [17] the gossip problem (where all nodes like to get initial values of all others) is treated. A common assumption of the above works is that the number of nodes is (much) larger than the number of channels. The computation-theoretic protocols of these works are mainly evaluated by their running time (the number of synchronous runs) in the order sense. Other related work includes antijamming study in wireless sensor networks and 802.11-based networks [18]–[22] and approaches to identify insider jammers [23]–[26].

III. PROBLEM FORMULATION

A. Network Model

We assume that a source node intends to transmit a message to N nodes in a wireless network by best efforts. Our discussion will mainly be focused on the single-hop setting, which is more amenable to analysis and allows us to better reveal the potential of our new approach. Nonetheless, this collaborative broadcast scheme can be readily extended to the multihop setting with suitable modifications, and we provide some relevant results in Section VIII. The broadcast message is divided into M short packets of the same length, each of which can be sent during one slot (frequency hop) duration t_p. These packets are broadcast periodically by the source node and relay nodes. For simplicity, each node (including the source) is assumed to send or receive a packet over one channel at a time. Our results can be easily extended to the multiradio case.

Our Collaborative UFH-based Broadcast (CUB) approach is an extension of the previous pair-wise UFH schemes [3]. To distinguish CUB from the straightforward extension of UFH in the broadcast scenario without node cooperation, we refer to the latter as Noncooperative UFH-based Broadcast (NUB). Specifically, in NUB, each node selects one out of C channels in each time slot to receive a packet, and repeats this process until obtaining the whole broadcast message. NUB does not have relay nodes in the broadcast process. It is intuitive that the noncollaborative broadcast to multiple receivers takes longer than the

pairwise transmission to a single receiver; this is also verified by our simulation results in Section VIII.

Wireless networks are prone to attacks in various aspects. Message authenticity and confidentiality can generally be achieved on the application layer [11]; the latter is also generally not a concern for broadcast applications. Efficient packet verification and reassembly techniques for UFH have already been discussed in literature (e.g., [10], [14]). Our study thus mainly considers the message availability issue, and focuses on the jamming attacks, which have been shown more detrimental than other types of attacks (such as insertion and overwriting) [3], [10]. The reader is referred to relevant literature in Section II for existing countermeasures of other attacks. The jammers make their best efforts to disrupt the legitimate communications: they desire to block as many receivers as possible, delay the broadcast process as long as possible, and incur as much energy consumption of the legitimate nodes as they can.

B. Jamming Model

As in [3] and [10], we consider omniscient jammers with bounded computation and transmission capability, each transmitting on a channel for at least $t_{\bar{p}}$ time to effectively jam a packet. The jamming attacks are usually categorized into nonresponsive and responsive ones, based on whether a jammer attempts to detect the ongoing transmission before jamming.

We assume that a nonresponsive jammer can block C_J channels simultaneously, and needs t_J time to switch jamming channels. Since it takes only $t_{\bar{p}}$ time to jam a channel, it is in the interest of a jammer to block as many channels as possible in each hop duration. The best she can achieve is jamming $n_J C_J$ channels in one slot, where $n_J = (t_p)/(t_{\bar{p}} + t_J)$, through the so-called sweep strategy [3].

We also assume a responsive jammer can sense C_s channels simultaneously, taking t_s time to switch sensing channels. So in one hop duration, she can sense up to $n_s C_s$ channels, where $n_s = (t_p - t_{\bar{p}} - t_J)/(t_s)$, accounting for the number of sensing cycles in a time slot.

In our study, we consider the most powerful jamming [3], [10], responsive-sweep, where a jammer conducts both nonresponsive and responsive jamming independently and simultaneously. The corresponding jamming probability for a single source (without relay) is given by

$$p_J = \frac{n_s C_s + n_J C_J}{C} \tag{1}$$

where C is the total number of orthogonal channels in the UFH system (assumed a sufficiently large number in our study). The analysis based on the responsive-sweep jamming provides a lower bound for the broadcast performance. For ease of reference, the commonly used notations are summarized in Table I.

IV. COLLABORATIVE UFH-BASED BROADCAST

We propose a Collaborative UFH-based Broadcast (CUB) scheme for antijamming broadcast without preshared keys in wireless networks. The source node performs the UFH approach: it sequentially and repeatedly sends the M packets, each of which is transmitted over a randomly selected channel.

The main idea of CUB is to allow nodes that have successfully received the whole message to help relay it over multiple channels for a duration determined by both the acknowledgement (ACK) signals and a time-out mechanism.

In CUB, each node first enters the receiving mode, adopting one of the receiving channel selection strategies discussed in Section IV-B. When successfully obtaining all the packets, a node then relays the packets to the remaining nodes, through one of the channel selection strategies discussed in the following.

A. Relay Channel Selection

We propose three relay channel selection strategies: Random Relay Channel selection (RRC), Sweep Relay Channel selection (SwRC), and Static Relay Channel selection (StRC). In RRC, each relay node randomly and independently selects one out of the C channels for the transmission of each packet, similar to the source node. This strategy is amenable to distributed implementation and has good scalability, while sometimes some relay nodes (as well as the source node) may happen to select the same channel, leading to collision and failure of transmission. Even with perfect synchronization and collaboration among the source and the relays such that the same packet is broadcast by all relays and the source at the same time, such overlap still leads to waste of energy and reduced opportunity.

To evaluate the theoretical maximum of RRC, we also consider its idealized version SwRC. The relays with SwRC take nonoverlapping channels for each packet transmission: the first relay node randomly selects one from C channels, the second relay node randomly chooses one out of the remaining $C - 1$ channels, and so on. This approach is proposed mainly as an alternative for RRC to facilitate the discussion on the trade-off between performance and complexity (the protocol overhead for coordination). The SwRC strategy can avoid the possible collision incurred in RRC, but requires information exchange among local nodes to determine the nonoverlapping channel ID in the broadcast.

In contrast to RRC and SwRC where relay channels change randomly from one transmission to the other, the relays in the StRC strategy take *fixed* nonoverlapping channels through the message broadcast process. We assume that the nodes have preassigned unique IDs, which, together with a suitable algorithm (see, e.g., [17]), guarantees that the probability of channel collision is negligible.[1] Each node is assumed to know the IDs of the nodes within its communication distance, and hence the IDs of all potential relay channels in its area, which constitute its initial relay channel list. Since a fixed set of relay channels are employed during the whole message broadcast process, it is reasonable to assume that the relay channels are (after some time) known to both the yet-to-inform receivers and jammers.

At a first glance, the StRC strategy seems to be a dumb approach: the jammers will go ahead to block these relay channels (even without sensing), so the hope of the receivers still lies in the UFH-based source node transmission. Actually this approach captures the essence of collaborative broadcast. As long as all the channels are not blocked simultaneously, the

---

[1]For simplicity, we assume that $N(<C)$ nodes in the network are indexed with $1, 2, \ldots, N$, and assign the ith channel to node i.

M	Number of packets in the message
N	Number of nodes to receive the message
C	Number of channels in the system
n	Number of relay nodes at a given time
$p_a(n)$	Successful packet reception rate given n relay nodes
$P_{m,n}$	One step transition probability from having m to n relay nodes
J	Number of jammers
p_J	Probability for a channel to be jammed
C_J	Number of channels concurrently blocked by a jammer
C_s	Number of channels concurrently sensed by a jammer
n_J	Number of jamming cycles in a slot
n_s	Number of sensing cycles in a slot

number of relays will increase with time. When the turning point is reached so that the jammers can no longer block all relay channels, the communication efficiency will be boosted dramatically.

An alternative view is that, as the UFH-based source node already provides uncertainty in its channel selection to counteract jamming, the StRC strategy introduces certainty in the relay selection to improve the communication efficiency. Furthermore, the StRC strategy is also easy to implement: it saves the efforts of channel switching, and requires little communication overhead for coordination. While for the StRC strategy, the source node could also avoid known relay channels to improve efficiency, we assume that the source node employs the same UFH strategy for fair comparison of different relay strategies.

B. Receiving Channel Selection

We mainly adopt a Random Receiving Channel selection (RRxC) scheme similar to that in UFH [3]. Each receiver with the RRxC strategy hops randomly and independently over the C channels. For the StRC relay strategy, we also devise an Adaptive Receiving Channel selection (ARxC) strategy. As mentioned, each node is assumed to know the initial relay channel list. We also assume that, after listening to a potential relay channel for a sufficient time, a node can determine whether the channel is clear, active (relaying packets), or jammed. Each receiver with the ARxC strategy first continuously sweeps its relay channel list, one at a time, in an order only known to itself. When it encounters an active channel, it receives a packet there. Once finding out that all the relay channels are jammed, it switches to the RRxC mode. While in the RRxC mode, when coming across a clear or active relay channel, it restricts itself to the relay channels again. This process repeats until a receiver successfully obtains all packets.

In essence, a receiver taking the ARxC strategy first attempts to take advantage of the available relay channels. However, these relay channels are also known to the jammers, and it is definitely in the jammers' interest to first block them. In the face of strong jamming such that all known relay channels are blocked, a receiver then switches back to RRxC. However, instead of continuously jamming the static relay channels, jammers may just spend enough energy to spoof the receivers away, and hence a smart receiver is allowed to check for such scenarios and come back to relay channels.

Smart jammers (insiders) may also open fake relay channels to lure the receivers and take advantage of them. Packet verification techniques [10], [14] and message authentication techniques on the application layer [11] can be employed to identify such fake channels, and remove them from the active relay list. The relay channel list is continuously updated through the broadcast process, and the dual-mode operation (ARxC and RRxC) achieves a good balance between efficiency and security.

C. Control of Transmission Duration

In its simplistic form, relays in our scheme could keep on broadcasting for a sufficiently long time as the source does; in practice, this will lead to significant energy waste. In CUB, each transmitter, either the source node or a relay node, stops sending packets once receiving ACKs from all of its neighboring nodes or reaching the maximum transmission duration Δ, whichever comes first. The limit on the transmission duration is set to deal with the possible loss of ACK signals due to channel imperfection, packet collision, or security attacks.

Each ACK signal, including the message ID, receiver ID, and time stamp, is sent on a fixed and known channel by a node right after successfully obtaining all M packets.[2] We assume an authentication mechanism on the ACK signals to prevent the spoofing from jammers. Local interference (including intentional jamming) on this common channel can be detected, and further actions can be taken accordingly. If the ACK mechanism fails, we resort to the time-out mechanism to control the transmission duration, in which Δ is a key parameter. We propose to set this parameter based on an estimate of average broadcast delay with the RRxC strategy.

Assume that one transmitter periodically broadcasts M packets to l nodes within its communication range. It is clear that the probability for all these l independent receivers to obtain all the M packets during the first m slots is $P[m] = (1 - (1 - p_a)^m)^{Ml}$, where $p_a = (1 - p_J)/C$ is the successful packet reception rate of a receiver. Following the analysis in [3], the average broadcast delay in terms of time slots can be approximately estimated as

$$T_{\text{avg}}^{\text{hop}} = M \sum_{m=0}^{\infty} \left[1 - (1 - (1 - p_a)^m)^{Ml} \right].$$

In a single-hop network, $l = N$. In a multihop setting with uniform node placement, the average number of the nodes within the reach of one hop is given by $l = D^2 N / R^2$, where D is the signal coverage radius and R is the radius of the network. Finally, the transmission duration Δ can be set as

$$\Delta = \alpha T_{\text{avg}}^{\text{hop}} = \alpha M \sum_{m=0}^{\infty} \left[1 - (1 - (1 - p_a)^m)^{Ml} \right] \quad (2)$$

where the constant α can be fine tuned in practice. It should be noted that Δ depends on the jamming probability, which could potentially be exploited by intelligent jammers. Hence Δ should (ideally) be updated during the broadcast process, and this issue deserves further study. It is found through simulation that the system performance is not sensitive to the choice of Δ, and in

---

[2]The reception of ACK signals at the relays can be implemented through frequency-division duplex (i.e., transmitting and receiving at different frequency bands) or time-division duplex techniques.

practice the parameter α may be adjusted according to the need (e.g., set to a larger value if jamming is particularly a concern).

V. BROADCAST PROTOCOL

Having discussed some key issues in CUB, we now present two major antijamming collaborative broadcast protocols for wireless networks. The first one is based on RRC relay and RRxC receiving strategies, and the second is based on the StRC relay and ARxC (dual-mode) receiving strategies. Other cases can be readily derived from these two. Both protocols are distributed: each node can execute Algorithm 1 or 2 with the knowledge of unique IDs of nodes within its communication range, to receive the message and then serve as a relay.

A. RRC-Based Broadcast

In the RRC-based protocol, each node other than the source node first enters the receiving mode, in which a node independently and randomly selects one out of C channels and listens, and switches to another randomly selected channel after one or several time slots to counteract jamming. This process repeats until the node successfully receives all M packets. Next, the node informs its neighbors about this information with an ACK signal, which contains the message ID, node ID, and time stamp, and is sent on a fixed and known channel.

The transmission mode of different nodes starts at different times (e.g., the source node enters this mode from the very beginning while a node at the edge of the network may never enter the transmission mode). In the transmission mode, each node randomly selects a channel out of the C channels and sends a packet. In order to deal with the possible loss of ACK signals due to channel imperfection or jamming, each transmission stops after Δ slots, given by (2), even without receiving enough ACK signals. The node repeats this process to send all M packets in sequence, until it receives all the ACK signals from its neighbors, or Δ time slots elapse, whichever comes first.

B. StRC-Based Broadcast

In the StRC-based protocol, after having successfully received all M packets, each node relays the message on a fixed channel, assumed to be distinctly related to its unique node ID. Each node is assumed to know the relay channels that its neighbors may use. In order to counteract a smart jamming, each node has two receiving modes, based on whether any relay channel is not blocked: If that is true, the node is focused on the relay channels by randomly selecting one of the potential relay channels in the neighborhood; otherwise, the node randomly selects one out of all the C channels.

Correspondingly, the StRC protocol uses a status flag that is set to be true at the beginning, and updates it according to the checking results of recently received packets. When working on the relay channels, the receiver changes the flag to be false, if failing to receive all the recent R_p packets, which means that all these relay channels are very likely to be jammed. The parameter R_p can be set as the actual number of neighboring nodes, or the average number of neighboring nodes if the former is unknown. When coming across a clear relay channel, the node sets the flag to be true and focuses on the relay channels again. After receiving all the packets, the node sends an ACK signal to its neighbors and enters to the transmission mode. Then the node sends the packets on a *fixed* channel corresponding to its preassigned unique node ID. The transmission duration is also controlled by a timer of length Δ.

Algorithm 2: StRC Version of the Anti-Jamming Collaborative Broadcast Protocol

$FlgClearRelayChannel$=True;
while *The node has not received all M packets yet* **do**
  if $FlgClearRelayChannel$=*True* **then**
    $ChID$=an integer randomly selected from the relay channel set in its neighborhood;
  else
    $ChID$ = an integer randomly selected from $[1,C]$;
  end
  Listen to the $ChID$-th channel;
  if $FlgClearRelayChannel$=*True* **then**
    if *All recent R_p packets are jammed* **then**
      $FlgClearRelayChannel$=False;
    end
  else
    if *The $ChID$-th channel is a unblocked relay channel* **then**
      $FlgClearRelayChannel$=True;
    end
  end
end
Send ACK (Message ID, Node ID, Time Stamp);
$\Delta \leftarrow$ Eq. (2) ;
$ChID$ = an integer derived from its Node ID ;
for $i \leftarrow 1$ **to** Δ **do**
  if *has not received ACKs from all its neighbors yet* **then**
    Send a packet on the $ChID$-th channel;
  else
    Stop the transmission immediately
  end
end

Algorithm 1: RRC Version of the Anti-Jamming Collaborative Broadcast Protocol

while *The node has not received all M packets yet* **do**
  $ChID$ = an integer randomly selected from $[1,C]$;
  Listen to the $ChID$-th channel;
end
Send ACK (Message ID, Node ID, Time Stamp);
$\Delta \leftarrow$ Eq. (2) ;
for $i \leftarrow 1$ **to** Δ **do**
  if *has not received ACKs from all its neighbors yet*
  then
    $ChID$ = an integer randomly selected in $[1,C]$;
    Send a packet sequentially on the $ChID$-th channel;
  else
    Stop the transmission immediately
  end
end

VI. Snapshot Scenario Analysis

In this and the next section, we will provide some performance analysis for our collaborative broadcast protocols. Then in Section VIII, we further evaluate our protocols through simulations. In this section, we consider a single-hop network and evaluate our proposed CUB scheme in a simplified snapshot scenario against responsive-sweep jamming. More specifically, we compute the successful reception rate of each receiver $p_a(n)$, defined as the probability that a receiver successfully receives a packet at a given time slot, in the presence of a source node, a jammer with jamming probability p_J, and n relay nodes for various strategies in CUB. We first analyze $p_a(n)$ under the assumption of perfect relay synchronization in Section VI-A, and provide the corresponding cooperation gain in VI-B. Then we investigate the impact of synchronization error among transmitters in VI-C, which turns out to be marginal for a fairly large number of available channels C. Finally in VI-D, numerical results are provided to better illustrate the above results. Receivers are assumed to hop at a much slower speed so that the synchronization error between the transmitters and receivers is not a concern.

A. Ideally Synchronous Relay

We first evaluate the successful reception rate for various strategies in CUB, assuming perfect timing and content synchronization, where all $n+1$ transmitters are synchronized so that the same packet is sent in the same hop. In this idealized case, a receiver can obtain the packet even when multiple nodes simultaneously transmit over that channel, i.e., multiple transmissions on the same channel do not incur conflicts. Our results about the successful packet reception rates for different relay and receiving channel selection strategies are given below.

Lemma 6.1: With RRC and RRxC strategies, the successful packet reception rate is given by

$$p_a^{\text{RRC}}(n) = \left(1 - \left(1 - \frac{1}{C}\right)^{n+1}\right)(1 - p_J). \qquad (3)$$

Proof: The probability for a source or relay node to transmit over a specific channel is $1/C$. With RRC and RRxC strategies, these $n+1$ nodes randomly and independently choose their transmission channels. Hence the probability that none of them picks the same channel with the receiver is $(1 - (1)/(C))^{n+1}$. Assuming perfect relay timing and content synchronization, the receiver can obtain the packet, if working on a channel that is clear from jamming with probability of $1 - p_J$, and is selected by at least one of these transmitters (with probability of $(1 - (1 - (1)/(C))^{n+1})$. Hence the successful packet reception rate is $(1 - (1 - (1)/(C))^{n+1})(1 - p_J)$. ∎

Lemma 6.2: With SwRC and RRxC strategies, the successful packet reception rate is given by

$$p_a^{\text{SwRC}}(n) = \left(1 - \left(1 - \frac{1}{C}\right)\left(1 - \frac{n}{C}\right)\right)(1 - p_J). \qquad (4)$$

Proof: The relay nodes with SwRC strategy work over a set of n nonoverlapping channels randomly selected out of C channels. Hence the probability that the receiver misses all these

$n+1$ transmitters is $(1 - (1)/(C))(1 - (n)/(C))$. The following part of the proof is similar to that for Lemma 6.1. ∎

Lemma 6.3: To maximize the average number of blocked packets, the best strategy for a jammer knowing that relay nodes perform the StRC strategy, is to first block as many relay channels as possible, and then continue to attack the nonrelay channels, if at all possible.

Proof: By using the StRC strategy, one copy of the packet is sent by a relay node on each of the n relay channels, while only the source node can access the remaining $C - n$ channels. Without loss of generality, suppose that the first n channels are relay channels, and let x denote the number of *relay* channels that the jammer blocks in one time slot. We have $0 \leq x \leq \min\{n, C_{SJ}\}$, where C_{SJ} is the total number of channels that a jammer can block within one time slot. The jammer may also block in each time slot $C_{SJ} - x$ nonrelay channels, each of which carries a packet with probability $1/C$. Thus the average number of blocked packets is

$$\mathrm{E}[N_{\text{jammed}}(x)] = x + (C_{SJ} - x)\frac{1}{C}. \qquad (5)$$

As $C > 1$, $\mathrm{E}[N_{\text{jammed}}(x)]$ monotonically increases with x, and hence is maximized when $x = \min\{n, C_{SJ}\}$, i.e., the jammer first blocks as many relay channels as possible, and continues to attack the nonrelay channels if at all possible. ∎

Lemma 6.4: With StRC and RRxC strategies, the successful packet reception rate is given by

$$p_a^{\text{StRC,RRxC}}(n) = \begin{cases} \frac{1}{C}\left(n - n_J C_J + 1 - \frac{n}{C}\right), & n > n_J C_J \\ \frac{1 - p_J}{C}, & \text{otherwise.} \end{cases} \qquad (6)$$

Proof: According to Lemma 6.3, when knowing that the relay nodes perform the StRC strategy, the jammer first blocks as many relay channels as possible. The sensing capability of the jammer does not help to block these static known relay channels, and the jamming probability against the relay channels is solely determined by the maximal transmission power and the blocking capability of the jammer. Therefore, the turning point happens at $n = n_J C_J$.

In the case $n > n_J C_J$, the jammer can effectively block up to $n_J C_J$ channels there. All the nonrelay channels and $n - n_J C_J$ relay channels are clear from jamming. The receiver can obtain the packet, if listening to either an unblocked relay channel, or a nonrelay channel selected by the source node. Since n out of C channels provide relay packets and the jammer blocks $n_J C_J$ of these relay channels, the probability that the receiver obtains a copy of the packet from an unblocked relay channel is $(n)/(C)(1 - (n_J C_J)/(n))$. In addition, the receiver can obtain a copy of the packet sent by the source node over the nonrelay channels with the probability $(1 - (n)/(C))(1)/(C)$. Thus,

$$\begin{aligned} p_a^{\text{StRC,RRxC}}(n) &= \frac{n}{C}\left(1 - \frac{n_J C_J}{n}\right) + \left(1 - \frac{n}{C}\right)\frac{1}{C} \\ &= \frac{1}{C}\left(n - n_J C_J + 1 - \frac{n}{C}\right). \end{aligned}$$

In the case $n \leq n_J C_J$, the jammers can block all the relay channels and $p_J C - n$ nonrelay channels (the channel

sensing function does help when attacking nonrelay channels). Each receiver randomly selects one out of C channels, and can get the packet if that channel is a nonrelay channel, selected by the source node, and free from jamming. Hence, $p_a = (1 - (n)/(C))(1)/(C)(1 - p_J')$, where $p_J' = (n_s C_s + n_J C_J - n)/(C - n)$ is the probability that a nonrelay channel is jammed, obtained by replacing $n_J C_J$ with $n_J C_J - n$ and C with $C - n$ in (1). Thus for $n_J C_J \geq n$, we have

$$
\begin{aligned}
p_a^{\text{StRC,RRxC}}(n) &= \left(1 - \frac{n}{C}\right) \frac{1}{C} \left(1 - \frac{n_s C_s + n_J C_J - n}{C - n}\right) \\
&= \left(1 - \frac{n}{C}\right) \frac{1}{C} \left(\frac{C - n_s C_s - n_J C_J}{C - n}\right) \\
&= \frac{C - n}{C} \cdot \frac{1}{C} \cdot \frac{C - C p_J}{C - n} = \frac{1 - p_J}{C}.
\end{aligned}
$$

$\blacksquare$

Lemma 6.5: With StRC and ARxC strategies, the successful packet reception rate is given by

$$
p_a^{\text{StRC,ARxC}}(n) = \begin{cases} 1 - \frac{n_J C_J}{n}, & n > n_J C_J \\ \frac{1 - p_J}{C}, & \text{otherwise.} \end{cases} \tag{7}
$$

Proof: In the case $n \leq n_J C_J$, ARxC and RRxC strategies behave the same, implying that $p_a(n)$ is also equal to $(1 - p_J)/(C)$. For the case $n > n_J C_J$, a receiver adopting the ARxC strategy will focus on the known relay channels that always carry copies of the packet. According to Lemma 6.3, the probability for a jammer to block a given relay channel is $(n_J C_J)/(n)$. Since the operations of the jammer and receiver are independent, the successful packet reception rate is equivalent to the probability that a randomly selected relay channel is unblocked: $1 - (n_J C_J)/(n)$. $\blacksquare$

Remark 1: Comparing (6) and (7), we can see that the advantage of ARxC lies in the first branch, when the number of relays overpowers the (hard) jamming capability, i.e., $n > n_J C_J$. This is indeed the scenario when the gain through cooperation becomes significant: as seen from (7), as the number of relay nodes increases from $n_J C_J$ to $n_J C_J + 1$, the successful packet reception rate rises from $(1 - p_J)/(C)$ to $(1)/(n_J C_J + 1)$.

B. Cooperation Gain

After the derivation of the successful packet reception rate $p_a(n)$ with n relay nodes for various strategies, we evaluate the corresponding cooperation gain for perfect relay synchronization, defined as $G(n) \triangleq p_a(n)/p_a(0)$, where the benchmark performance of NUB, $p_a(0)$, can be obtained by taking $n = 0$.

For RRC and SwRC, by (3) and (4), the cooperation gains for sufficiently large C under perfect synchronization can be approximated by

$$
G^{\text{RRC}}(n) \approx G^{\text{SwRC}}(n) \approx n + 1. \tag{8}
$$

For the case of StRC, $p_a(0) = (1 - p_J)/C$, and by (6) and (7), the cooperation gain for RRxC and ARxC are given, respectively, by

$$
G_{\text{RRxC}}^{\text{StRC}}(n) = \begin{cases} \frac{1}{1 - p_J} \left(n - n_J C_J + 1 - \frac{n}{C}\right), & n > n_J C_J \\ 1, & \text{otherwise} \end{cases}
$$

and

$$
G_{\text{ARxC}}^{\text{StRC}}(n) = \begin{cases} \frac{C}{1 - p_J} \left(1 - \frac{n_J C_J}{n}\right), & n > n_J C_J \\ 1, & \text{otherwise.} \end{cases} \tag{9}
$$

Remark 2: In contrast to RRC and SwRC where the cooperation gain grows roughly linearly with the number of relays, the cooperation gain for the StRC strategy is dichotomous: below the threshold $n_J C_J$, there is no cooperation gain; once the number of relay nodes passes the threshold, the cooperation gain rises dramatically, especially for the ARxC receivers. For instance, given $C = 256$, $p_J = 0.2$, and $n_J C_J = n_s C_s$, the cooperation gain with $n = 40$ relay nodes is 115.2 for StRC with the adaptive receiver, approximately 2.9 times greater than RRC or SwRC; meanwhile, it is as small as 19 for StRC with the RRxC receiver.

C. Impact of Synchronization Error

We now relax the assumption on perfect relay synchronization, and consider a more realistic scenario where two or more transmissions on the same channel always lead to a failure in reception (due to difference in arrival time or transmitted packets). We revisit the results of $p_a(n)$ for various strategies, taking into account relay synchronization error. First, for the RRC strategy, a receiver can successfully obtain a packet in the asynchronous scenario, when listening to a channel selected by either the source node or exactly one relay, and thus

$$
p_a^{\text{RRCAsyn}}(n) = (n + 1) \frac{1}{C} \left(1 - \frac{1}{C}\right)^n (1 - p_J). \tag{10}
$$

Proof: Each source or relay node under the RRC strategy transmits over a given channel with a probability $1/C$. Hence the probability for exactly one out of these $n + 1$ transmitters to work on a given channel (i.e., one node selects that channel and all the other n nodes choose other channels) can be written as $(n + 1)/(C)(1 - (1)/(C))^n$. Without perfect synchronization, the receiver can obtain the packet, if working on a channel that is clear from jamming with probability of $1 - p_J$, and is selected by exactly one of the $n + 1$ transmitters. Thus the successful packet reception rate is given by $(n + 1)/(C)(1 - (1)/(C))^n (1 - p_J)$. $\blacksquare$

As shown in (10), the successful packet reception rate in the RRC strategy is proportional to $(n + 1)$, because more packet copies are provided to the receiver by more relay nodes. On the other hand, the factor $(1 - (1)/(C))^n$ decreases with n, which accounts for more channel collision happening with more relay nodes.

Next, for the case of SwRC, as the relay channels are nonoverlapping, the successful packet reception rate becomes

$$
p_a^{\text{SwRCAsyn}}(n) = \left(\frac{1}{C} \left(1 - \frac{n}{C}\right) + \left(1 - \frac{1}{C}\right) \frac{n}{C}\right) (1 - p_J). \tag{11}
$$

Finally, in the StRC strategy, n relay nodes always send the signal on fixed nonoverlapping relay channels, and never interfere with the source node over the nonrelay channels; we have $p_a^{\text{StRCAsyn}}(n) = (1 - p_J)/(C)$ for $n_J C_J \geq n$. On the other hand, when $n_J C_J < n$, the successful packet may be influenced

when the source node interferes with one of the spare relay channels. For the receiver with the RRxC strategy, we have

$$p_{\mathrm{RRxC},a}^{\mathrm{StRCAsyn}}(n) = \frac{n}{C}\left(1 - \frac{n_J C_J}{n}\right)\left(1 - \frac{1}{C}\right)$$
$$+ \left(1 - \frac{n}{C}\right)\frac{1}{C}. \quad (12)$$

And for the receiver with the ARxC strategy, we have

$$p_{\mathrm{ARxC},a}^{\mathrm{StRCAsyn}}(n) = \left(1 - \frac{n_J C_J}{n}\right)\left(1 - \frac{1}{C}\right). \quad (13)$$

Remark 3: Comparison with the results in Section VI-A shows that, when C is large, there is little degradation in the broadcast performance due to small synchronization error, which will be confirmed in Fig. 1(c). Another observation is that the successful packet reception rate for the proposed CUB scheme mostly increases with the number of relay nodes, as verified below. By (10), we have $(\partial)/(\partial n)p_a^{\mathrm{RRCAsyn}} = (1)/(C)(1-p_J)(1-(1)/(C))^n[1+(n+1)\ln(1-(1)/(C))] \geq 0$, unless $n \geq n_0 = (-1/\ln(1-(1)/(C)))-1)$, which is very large for $C \gg 1$. For example, we have $n_0 = 127$ for $C = 128$, indicating that the successful packet reception rate for RRC increases with n, unless $n \geq 128$. For SwRC, by (11), we have $(\partial)/(\partial n)p_a^{\mathrm{SwRCAsyn}} = (1-(2)/(C))(1-p_J)/(C) > 0$. Finally, for StRC, by (13), we have that $(\partial)/(\partial n)p_{\mathrm{RRxC},a}^{\mathrm{StRCAsyn}} \geq 0$ always holds. Numerical results in Fig. 1(c) show that the increasing rate is comparable to that in the synchronous case.

D. Numerical Evaluation

To better illustrate the properties of CUB as given in (3)–(13), some numerical evaluations are provided in Fig. 1 for $C = 256$, assuming a responsive-sweep jammer with equal sensing capability and blocking ability, i.e., $n_s C_s = n_J C_J = C p_J/2$. As shown in Fig. 1(a), the cooperation gain with RRC or SwRC is approximately proportional to n, and SwRC only slightly outperforms RRC. Intuitively, the probability of channel conflict is negligible for a large C. Thus the coordination overhead of SwRC is not justified, and RRC is generally preferred over SwRC in practice. The advantage of StRC is obvious under weak jamming (small p_J) or large-scale cooperation (large n), and the superiority of the adaptive receiving strategy ARxC over RRxC is also clearly demonstrated. Each of the StRC performance curves has a turning point, e.g., $n = 51$ for $p_J = 0.4$ and $n = 2$ for $p_J = 0.01$ in Fig. 1(a), actually corresponding to $n = n_J C_J = C p_J/2$ as discussed in Remark 1.

Fig. 1(b) is complementary to Fig. 1(a), demonstrating performance with respect to the jamming probability. It is seen that the performance of both RRC and SwRC degrades gracefully with the jamming strength, and RRC is more robust than StRC against strong jamming. Similar to Fig. 1(a), there is also a turning point for the StRC performance at $p_J = 2n/C$, e.g., $p_J = 0.016$ for $n = 2$. We also observe the boost of performance by orders of magnitude through our collaborative broadcast.
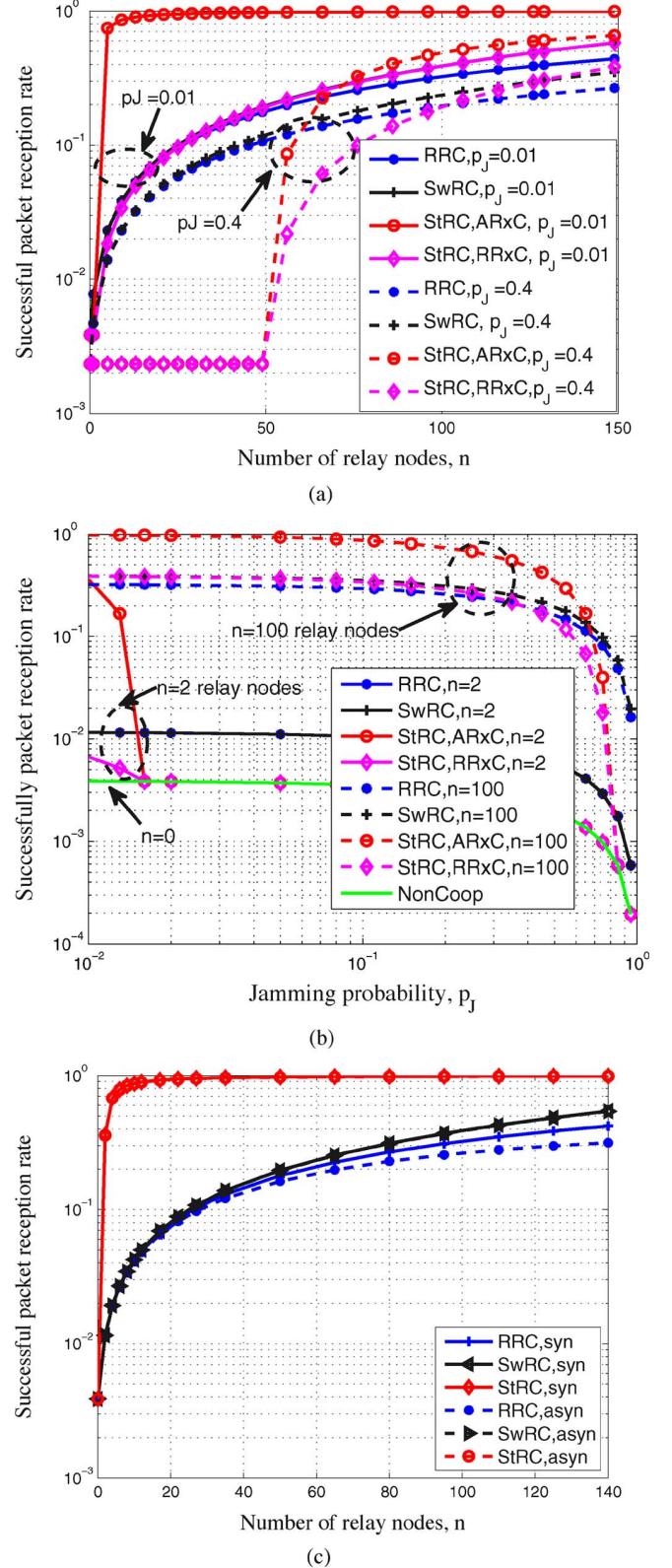

Fig. 1. Probability for a node with either the default RRxC strategy or the ARxC strategy (only for StRC) to successfully receive a packet during a time slot, in the collaborative broadcast using the RRC, SwRC, or StRC relay strategies over $C = 256$ channels in the UFH system, against one responsive-sweep jammer with jamming probability p_J. (a) Ideally synchronous relay with $p_J = 0.01$ or $p_J = 0.4$. (b) Ideally synchronous relay with $n = 2$ or 100 relay nodes. (c) $p_J = 0.01$, ARxC for StRC.

Finally, the impact of synchronization error on the performance is illustrated in Fig. 1(c). Conforming to our discussion in
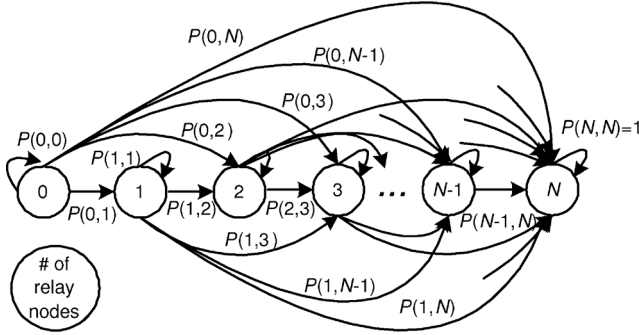
Fig. 2. Transition diagram of the finite Markov chain for the broadcast process to N nodes, where each state represents the number of relay nodes in the network and all nodes receive the message at the absorbing state N.

Remark 3, all three relay strategies are robust against synchronization error, either in time or in packet content. This robustness against synchronization error is highly desirable for practical implementation. In the remaining part of the paper, we thus assume synchronous relay for simplicity.

VII. FULL BROADCAST PROCESS ANALYSIS

In this section, we give some quantitative analysis on the whole broadcast process in synchronous single-hop networks through Markov chain modeling. In particular, the full process of the broadcast to N nodes can be modeled as a homogenous finite Markov chain with $N+1$ states. As illustrated in Fig. 2, each state $X \in \{0, \ldots, N\}$ represents the number of relay nodes at a given time. At the beginning of the broadcast process, there is no relay node. Hence the initial state probability vector for the $N + 1$ states in the Markov chain is $\mathbf{p}_s(0) = [1, 0, \ldots, 0]^T$. The number of relay nodes increases as more nodes successfully obtain the message. The broadcast completes when all N nodes obtain the message, indicating that the state N is the absorbing state in the Markov chain. For simplicity, we assume $M = 1$ in the following discussion. The broadcast delay is defined as the duration from the beginning of broadcast until the time when all nodes in the network successfully receive the whole message.

In this process, the one-step transition probability $P_{m,n}$ represents the probability that the number of relay nodes changes from m to n after one time slot. As the number of relay nodes increases with time, we have $P_{n,m} = 0$ for $n > m$.[3] All receivers are assumed to have the same reception performance. Consequently, the broadcast to $N - n$ receivers with exactly n relay nodes during one time slot can be modeled as a binomial distribution with $N - n$ trials, whose success probability equals to $p_a(n)$ given by (3)–(7) for various scenarios. Thus the transition probability can be rewritten as

$$
P_{m,n} = \begin{cases} \mathcal{C}_{N-m}^{n-m} p_a^{n-m}(m)[1 - p_a(m)]^{N-n}, & m \leq n \leq N \\ 0, & \text{otherwise} \end{cases}
$$

(14)

where $\mathcal{C}_{N-m}^{n-m}$ is the binomial coefficient.

Lemma 7.1: The average broadcast time T to N nodes is given by

$$
T = \sum_{l=1}^{N} (\mathbf{M})_{1,l}
$$

(15)

[3]For simplicity, we ignore the time-out mechanism in this section.

where $(\mathbf{M})_{1,l}$ is the element of matrix $\mathbf{M}$ in (16) on the first row and lth column.

Proof: The state N of the Markov chain in Fig. 2 is absorbing, and all the other states are transient. Hence the fundamental matrix of the Markov chain [27] is given by

$$
\mathbf{M} \equiv (\mathbf{I} - \mathbf{W})^{-1}
$$
$$
= \begin{bmatrix} 1 - P_{0,0} & -P_{0,1} & \cdots & -P_{0,N-1} \\ 0 & 1 - P_{1,1} & \cdots & -P_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 - P_{N-1,N-1} \end{bmatrix}^{-1}
$$

(16)

where $\mathbf{I}$ is the identity matrix, and

$$
\mathbf{W} = \begin{bmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,N-1} \\ 0 & P_{1,1} & \cdots & P_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & P_{N-1,N-1} \end{bmatrix}
$$

is the transition probability matrix for the transient states. The expected time to absorption from the nonrecurrent state 0 can be interpreted as the average broadcast time. Following the analysis in [27, Ch. 15.5.2], we have $T = \sum_{l=1}^{N} \mathbf{M}_{1,l}$, where $\mathbf{M}_{1,l}$ is the element of matrix $\mathbf{M}$ on the first row and lth column. ∎

Corollary 7.2: For $N = 2$ nodes, the average broadcast delay for the RRC strategy is about $2/(3 - 1.5/C)$ of that in the noncollaborative scheme.

Proof: For the Markov chain with three states, by (14) we have $P_{0,0} = (1 - p_a(0))^2$, $P_{0,1} = 2p_a(0)(1 - p_a(0))$, and $P_{1,1} = 1 - p_a(1)$. Then by (15), the average broadcast delay can be written as

$$
T_2 = \frac{1}{1 - P_{0,0}} + \frac{P_{0,1}}{(1 - P_{0,0})(1 - P_{1,1})}
$$
$$
= \frac{2p_a(0)(1 - p_a(0)) + p_a(1)}{p_a(0)p_a(1)(2 - p_a(0))}.
$$

(17)

For the RRC strategy with perfect synchronization, by (3), we have $p_a(0) = (1 - P_J)/C$ and $p_a(1) = (1 - P_J)(2 - (1)/(C))/C$. Thus (17) can be further simplified into

$$
T_{2\text{RRC}} = \frac{4C - 3 + 2p_J}{(1 - p_J)\left(2 - \frac{1}{C}\right)\left(2 - \frac{1 - p_J}{C}\right)}.
$$

(18)

For the NUB scheme without relay, the above formulas can still be applied, with the understanding that each state of the Markov chain denotes the number of nodes that have successfully received the whole message in the network. In this case, we have $p_a(0) = p_a(1) = (1 - P_J)/C$, and thus (17) can be simplified into

$$
T_{2\text{NUB}} = \frac{3C - 2(1 - p_J)}{\left(2 - \frac{1 - p_J}{C}\right)(1 - p_J)}.
$$

(19)

By (18) and (19), we have $T_{2\text{RRC}} \approx 2T_{2\text{NUB}}/(3 - 1.5/C)$, as $C \gg 1 \geq p_J \geq 0$. ∎

Remark 4: For a *three-node* network scenario, where a source node attempts to broadcast the message to $N = 2$ nodes, the above result shows that the collaborative broadcast scheme can reduce the broadcast delay by about 33%. For the broadcast to

$N = 3$ nodes, CUB can save nearly 49% of the transmission time. These results match the statement in Section VI that the performance gain increases with the number of nodes in the network. Hence the proposed scheme is expected to significantly improve the communication efficiency in a large network.

Remark 5: Similarly to Corollary 7.2, the average broadcast delay for the SwRC strategy can be obtained by replacing $p_a(n)$ in (17) with (4) for given n. Since SwRC is equivalent to RRC with zero or one relay node, they have the same $p_a(0)$ and $p_a(1)$, and hence the same broadcast performance with $N = 2$, i.e., $T_{2\mathrm{SwRC}} = T_{2\mathrm{RRC}}$. As to the StRC strategy, the average delay can be obtained by using (6) or (7) to replace p_a in (17). After simplification, we have the average broadcast delay of StRC to 2 nodes as

$$T_{2\mathrm{StRC,RRxC}} = \frac{2\frac{1-p_J}{C}\left(1 - \frac{1-p_J}{C}\right) + \frac{1}{C}\left(2 - \frac{1}{C} - \frac{Cp_J}{2}\right)}{\frac{1-p_J}{C}\frac{1}{C}\left(2 - \frac{1}{C} - \frac{Cp_J}{2}\right)\left(2 - \frac{1-p_J}{C}\right)} \tag{20}$$

or

$$T_{2\mathrm{StRC,ARxC}} = \frac{\frac{2}{C}(1-p_J)\left(1 - \frac{1-p_J}{C}\right) + 1 - \frac{Cp_J}{2}}{\frac{1-p_J}{C}\left(1 - \frac{Cp_J}{2}\right)\left(2 - \frac{1-p_J}{C}\right)} \tag{21}$$

under weak jamming.

By (18)–(21), the average broadcast delay to two nodes is 192, 128, and 65 slots, respectively, for NUB, CUB, with the RRC/SwRC strategy, and CUB with the StRC strategy, with $C = 128$ and $p_J = 0$. These results have been verified by simulation. However, (15) is not easy to evaluate in general, especially for a large number of nodes N. We will further resort to simulation for performance evaluation in the following section.

Lemma 7.3: Given that the transmission time is sufficiently long and the jamming probability $p_J < 1$, all the nodes in the network can receive the broadcast message with probability one.

Proof: The probability for all N nodes to receive the message after a sufficiently long time is actually the absorbing probability from state 0 in the Markov chain. If the jamming probability $p_J < 1$, the transition probability from state m to state n, $P_{m,n} > 0$ for all $N \geq n > m \geq 0$. In this case, the Markov chain has finite transient states and exactly one absorbing state. Following [27], the absorbing probability $P_{\mathrm{successBC}} = \sum_{l=1}^{N} \mathbf{M}_{1,l} P_{l,N} = 1$ in this finite-state absorbing Markov chain. Therefore, if the jamming probability $p_J < 1$, the system will stay in the absorbing state with probability one, and all the nodes in the network can receive the broadcast message with probability one. ∎

VIII. SIMULATION RESULTS

In this section, we evaluate the performance of our collaborative broadcast scheme (CUB) (see Section V) and compare it with that of NUB through simulations. Assuming synchronous relays, we consider the broadcast for both the single-hop and multihop networks, and investigate the corresponding broadcast delay (from the beginning of broadcast until the time when all nodes in the network successfully receive the whole message) and the energy consumption (i.e., the energy consumption for all the nodes in the network to send and to receive packets during the whole broadcast process).

Unless specified otherwise, in the simulations, the source node broadcasts $M = 7$ packets to N nodes over $C = 128$ channels, against J responsive-sweep jammers with $t_s = 10\,\mu s$, $t_J = 15\,\mu s$, $C_s = 1$, $C_J = 1$, $t_p = 40\,\mu s$, and $t_{\tilde{p}} = 5\,\mu s$ (thus $n_s = n_J = 2$).[4] The energy consumption for a node to send and to receive a packet is set as $E_t = 1$ and $E_r = 0.1$, respectively.

A. Single-Hop Networks

The advantage of an RRC-based CUB protocol over NUB is clearly demonstrated in Fig. 3. For example, compared to NUB, CUB takes only 14% of the time to complete the broadcast, and saves 36% of the energy, with $J = 20$ jammers and $N = 100$ nodes. While it is expected that CUB will significantly facilitate the broadcast process, it is less intuitive that the total energy cost is also reduced. This may be explained as follows. First, the CUB strategy consumes less energy than NUB to receive packets due to significantly reduced broadcast delay. Meanwhile, even though the number of transmit nodes increases in CUB, the total number of transmitted packets in CUB actually decreases, again due to much shorter communication time.

Moreover, Fig. 3(a) shows an interesting phenomenon that, as the network size N increases, NUB requires *longer* broadcast latency while CUB actually incurs *shorter* delay. Hence, the saving in broadcast time by replacing NUB with CUB is more prominent in a larger network. For example, with $J = 20$ jammers, CUB can save 59% of the broadcast latency needed for NUB with $N = 10$ nodes, which increases to 86% as N rises to 100.

Let us further examine why CUB requires shorter latency as N rises. Two main reasons account for this seemingly counterintuitive phenomenon. First, in a larger network, the probability that a packet is received by at least one receiver is higher. Take the case with one transmitter (the source node) and N independent receivers as an example: the probability for at least one receiver to obtain the packet is $(1 - (1 - (1)/(C))^N)(1 - p_J)$, which goes from $(1 - p_J)/(C)$ to $(1 - p_J)$ as N goes to infinity. Second, as mentioned before, nodes that already receive the message turn into relays to continue their contribution in another domain. Therefore, both the increased multiuser diversity and spectral diversity account for the shorter broadcast in a larger network.

It is known from [10] that approximately 2000 slots are needed for BMA to send a message of seven packets to a single node in the pairwise communication even without jamming. Clearly, the broadcast delay of the BMA-based noncooperative broadcast is even longer than that, let alone jamming attacks. In contrast, our simulation shows that it takes about 1000 slots for CUB to broadcast such a message to 100 nodes against five independent sweep-responsive jammers, each with jamming probability of 0.03.

---

[4]For single-hop networks, it is equivalent to a powerful jammer with combined capability of $C_s = J$ and $C_J = J$. Our simulation may also incorporate a scenario where normal nodes are compromised.
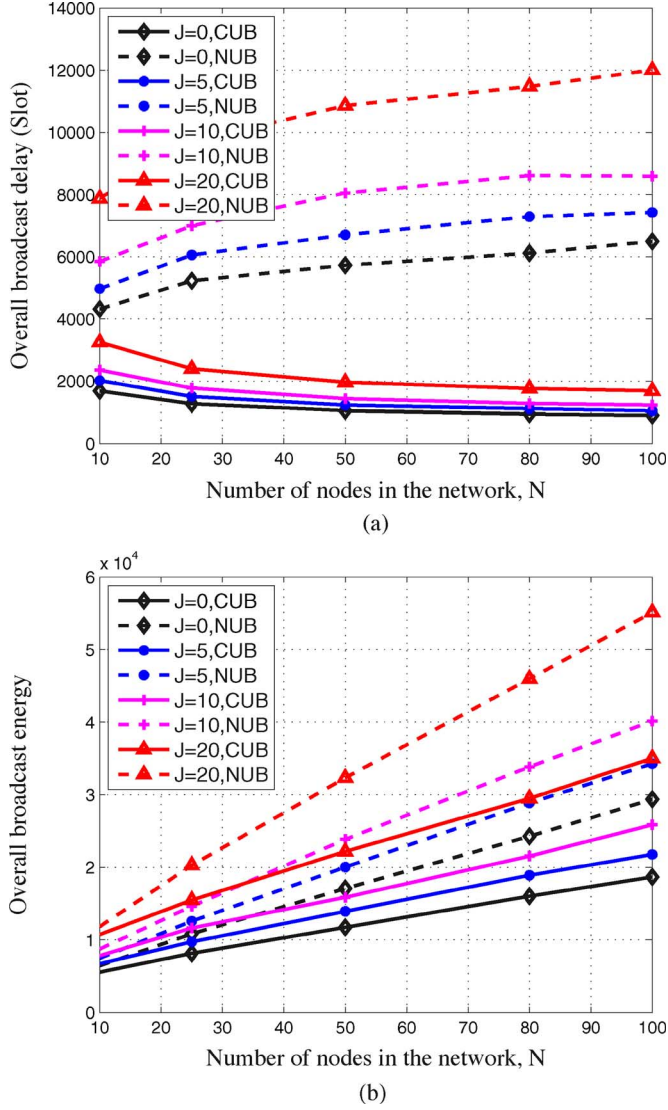
Fig. 3. Performance of the collaborative broadcast (CUB) and noncooperative broadcast (NUB), as a source node sending $M = 7$ packets to N nodes over $C = 128$ channels, with $C_J = 1$, $C_s = 1$, $t_p = 40\ \mu s$, $t_{\bar{p}} = 5\ \mu s$, $t_J = 15\ \mu s$, and $t_s = 10\ \mu s$ (thus $n_s = n_J = 2$), assuming the energy consumption for a node to send (or receive) a packet is $E_t = 1$ (or $E_r = 0.1$), and the RRC strategy is applied to counteract J responsive-sweep jammers that are equivalent to a powerful jammer with combined capability of $C_s = J$ and $C_J = J$. (a) Broadcast delay. (b) Energy consumption.

Fig. 4. Broadcast performance with $\nu = N/R^2 = 50$, for the broadcast of $M = 7$ packets to N nodes with changing network radius R and signal coverage radius $D = 1$, against J responsive-sweep jammers. (a) Average broadcast delay. (b) Average energy consumption.

B. Multihop Networks

In this subsection, we evaluate the performance of the proposed CUB protocols in a multihop network. Without loss of generality, the source node is assumed to be located at the center of a disk area of radius R, broadcasting a message to N identical randomly located nodes, each of which employs a common communication range D. In our simulation, we consider a network density $\nu = N/R^2 = 50$ and J jammers with normalized jamming power $\rho = (D_c/D)^\gamma = 2$, where D_c is the jamming radius and the path-loss exponent $\gamma = 3.8$. The energy consumption for a node to send a packet is modeled as $E_t = D^\gamma$, while E_r is still set as 0.1. We evaluate the average performance over 100 realizations for each network setting.
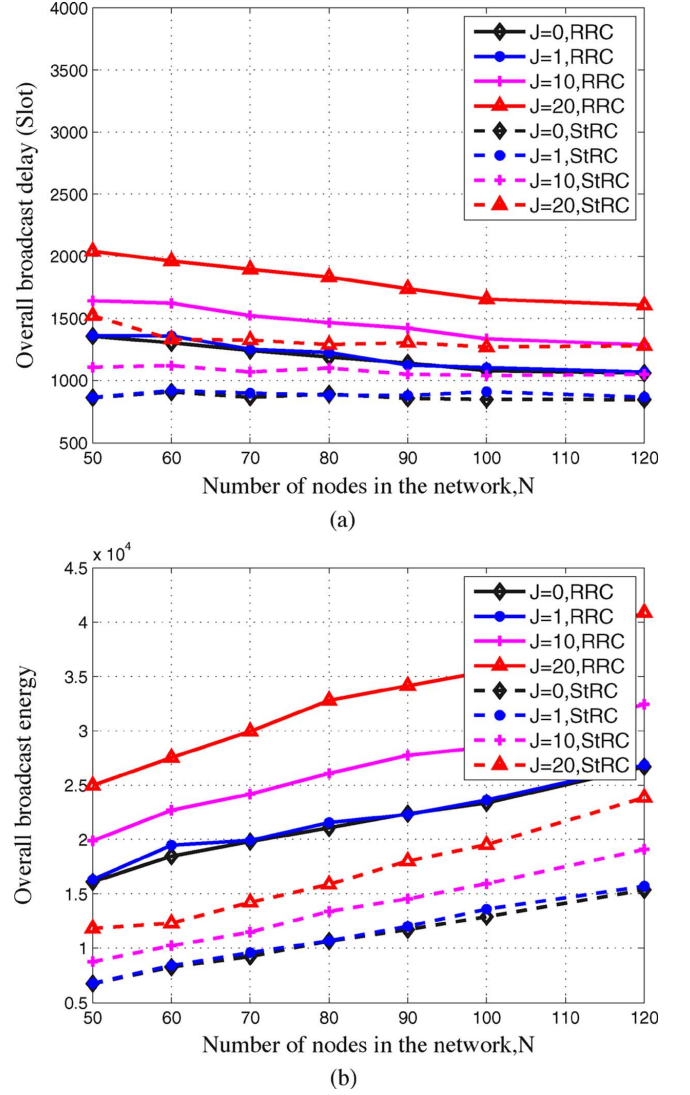
Fig. 4 presents the RRC performance in solid curves and the StRC performance in dashed curves, against $J = 0$ to 20 responsive-sweep jammers described in different colors. In comparison with Fig. 3, it is shown that our collaborative broadcast schemes continue to achieve significant performance gain in multihop networks. These results also confirm that the StRC strategy outperforms the RRC in most cases, by providing some degree of certainty in the relay channel selection.

In addition, Fig. 4(a) shows a similar phenomenon as in the single-hop setting that the broadcast delay *decreases* with the network size N, for a given node density. For example, the average broadcast delay of the RRC strategy *reduces* from 2000 slots to 1600 slots as the total number of nodes N *increases* from 50 to 120, for $J = 20$. Similar observation also holds for the StRC strategy, especially with a large number of jammers. This phenomenon can be similarly explained through increased multiuser, spectral, and spatial diversities.
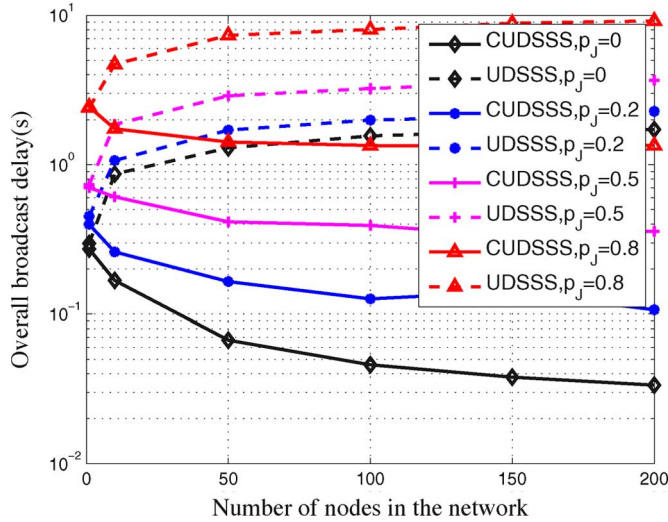
Fig. 5. Average overall delay to broadcast a 1000-bit message to all the nodes in a collaborative UDSSS system, against one responsive-sweep jammer with jamming probability p_J, with a public set of 50 orthogonal spreading codes, 23-dB spreading gain, 0.5 channel encoding rate, and 4-Mb/s data transmission rate. Each receiver has a computing capability of 400 million instructions per second (MIPS).

Finally, as expected, Fig. 4(b) shows that the overall energy consumption rises with the size of the network, as the number of receivers and the relay nodes increases.

IX. COLLABORATIVE UDSSS

Our collaborative scheme is complementary to existing and emerging jamming countermeasures. We have shown its effectiveness with the UFH approach above. As a concept proof, we further discuss its application to UDSSS techniques in this section [9], [11]. Without loss of generality, consider an ideally synchronous single-hop wireless network, where the source randomly selects one spreading sequence from a public set for each message transmission. In this collaborative UDSSS, each node that has successfully decoded the message transmits it to the other nodes with a spreading sequence randomly selected from the public sequence set. Most techniques in UDSSS can be directly applied here, such as message verification, bit interleaving, and packet encoding approaches [9], [11].

Experimental results have shown that the despreading of a message in a trial-and-error manner is one of the most time-consuming operations in UDSSS. Once recording the broadcast message, each receiver randomly selects a spreading sequence from the public sequence set to despread the message, and can succeed if choosing the same synchronized sequence as one of the transmitters. It is intuitive that the successful despreading probability increases with the number of transmitters. Therefore, collaborative UDSSS exploits node cooperation to provide code diversity to facilitate despreading and hence reduce the broadcast delay. The cooperation gain for UDSSS systems is clearly demonstrated in Fig. 5, which increases with the number of nodes in the network, similar to that in UFH. This is due to the increased multiuser and code diversity, already observed in Section VIII.

It is worth mentioning that FH and DSSS are two different spread-spectrum techniques; the former realizes its immunity to

interference and security attacks through escape and avoidance, while the latter relies on the large spreading gain to mitigate. This difference reflects in several aspects for UFH and UDSSS, and for their collaborative versions. First, in UFH a receiver cannot guarantee to obtain the packet in each hop, but once tuning to the right channel, the decoding effort is minimum. In contrast, UDSSS has a more predictable transmission delay but requires significantly more decoding efforts. Therefore, collaborative schemes help UFH more on the transmission side, and UDSSS more on the reception side. It should be noted that collaborative UDSSS also introduces a higher interference level to receivers. Due to their inherent differences, the performance of UFH and collaborative UFH is mainly limited by the hardware capability (such as sensing and switching) and available power of legitimate nodes and jammers, while that of UDSSS and collaborative UDSSS, among others, is mainly restricted by the computing power.

X. CONCLUSION

We have proposed a collaborative broadcast scheme that utilizes a cooperative communication technique and exploits frequency (channel), spatial, and multiuser diversities to resist jamming and enhance communication efficiency. Collaborative broadcast protocols based on RRC and StRC relay strategies have been presented, explicitly considering ACK and time-out mechanisms for transmission control. We have analyzed their cooperation gain in terms of the successful packet reception rate in single-hop wireless networks. It has been found that RRC can provide a cooperation gain proportional to the number of relay nodes, and is amenable to simple distributed implementation. On the other hand, the StRC strategy substantially further improves the cooperation gain under weak jamming relative to the collaboration scale. In addition, our broadcast scheme has been shown to be robust against relay synchronization error. We have further provided a closed-form expression of the average broadcast delay for the full broadcast process based on Markov chain modeling.

We have verified the merits of collaborative broadcast in terms of communication efficiency and jamming resistance for both single-hop and multihop networks. Simulation results indicate significant performance gain of the proposed scheme over the noncollaborative UFH-based broadcast scheme in wireless networks.

REFERENCES

[1] R. A. Poisel, *Modern Communications Jamming Principles and Techniques*. Norwood, MA: Artech House, 2006.

[2] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005, pp. 46–57.

[3] M. Strasser, C. Popper, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *Proc. IEEE Symp. Security and Privacy*, 2008, pp. 64–78.

[4] L. Baird, W. Bahn, M. Collins, M. Carlisle, and S. Butler, "Keyless jam resistance," in *Proc. IEEE Inf. Assurance and Security Workshop*, 2007, pp. 143–150.

[5] T. Jin, G. Noubir, and B. Thapa, "Zero pre-shared secret key establishment in the presence of jammers," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, New Orleans, LA, 2009.

[6] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *Proc. ACM WiSec*, Zurich, Switzerland, 2009.

[7] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[8] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport, "Gossiping in a multi-channel radio network," in *Proc. Int. Symp. Distributed Computing (DISC'07)*, Lemesos,, Cyprus, 2007.

[9] C. Popper, M. Strasser, and S. Capkun, "Jamming-resistant broadcast communication without shared keys," in *Proc. USENIX Security Symp.*, Montreal, Canada, 2009.

[10] M. Strasser, C. Popper, and S. Capkun, "Efficient uncoordinated FHSS anti-jamming communication," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, New Orleans, LA, 2009.

[11] C. Popper, M. Strasser, and S. Capkun, "Anti-jamming broadcast communication using uncoordinated spread spectrum techniques," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 5, pp. 703–715, Jun. 2010.

[12] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport, "Secure communication over radio channels," in *Proc. ACM PODC*, Toronto, Canada, 2008.

[13] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang, "Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure," in *Proc. IEEE Ann. Computer Security Applications Conf. (ACSAC)*, Austin, TX, 2010.

[14] D. Slater, P. Tague, R. Poovendran, and B. Matt, "A coding-theoretic approach for efficient message verification over insecure channels," in *Proc. ACM Conf. Wireless Network Security (WiSec'09)*, Zurich, Switzerland, 2009.

[15] A. Liu, P. Ning, H. Dai, and Y. Liu, "USD-FH: Jamming-resistant wireless communication using frequency hopping with uncoordinated seed disclosure," in *Proc. 7th IEEE Int. Conf. Mobile Ad-Hoc and Sensor Systems*, 2010, pp. 41–50.

[16] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized differential DSSS: Jamming-resistant wireless broadcast communication," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, 2010, pp. 1–9.

[17] S. Gilbert, R. Guerraoui, D. Kowalski, and C. Newport, "Interference-resilient information exchange," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, 2009, pp. 2249,–2257.

[18] M. Li, and I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, 2007, pp. 1307–1315.

[19] W. Xu, W. Trappe, and Y. Zhang, "Channel surfing: Defending wireless sensor networks from interference," in *Proc. Int. Conf. Information Processing in Sensor Networks (IPSN)*, 2007, pp. 499–508.

[20] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, 2007, pp. 2526,–2530.

[21] T. Brown, J. James, and A. Sethi, "Jamming and sensing of encrypted wireless ad hoc networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Florence, Italy, 2006.

[22] A. D. Wood, J. A. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks," in *Proc. Ann. IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2007, pp. 60–69.

[23] A. Chan, X. Liu, G. Noubir, and B. Thapa, "Broadcast control channel jamming: Resilience and identification of traitors," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, 2007, pp. 2496–2500.

[24] I. Shin, Y. Shen, Y. Xuan, M. Thai, and T. Znati, "Reactive jamming attacks in multi-radio wireless sensor networks: An efficient mitigating measure by identifying trigger nodes," in *Proc. FOWANC*, Florence, Italy, 2009.

[25] J. Chiang and Y. Hu, "Dynamic jamming mitigation for wireless broadcast networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, 2008, pp. 1211–1219.

[26] H. Nguyen, T. Pongthawornkamol, and K. Nahrstedt, "A novel approach to identify insider-based jamming attacks in multi-channel wireless networks," in *Proc. IEEE MILCOM*, Orlando, FL, 2009.

[27] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 2002, ch. 16.
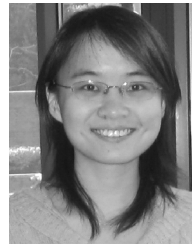
Liang Xiao (A'10) received the B.S. degree in communication engineering from Nanjing University of Posts & Telecommunications, China, in 2000, the M.S. degree in electrical engineering from Tsinghua University, China, in 2003, and the Ph.D. degree in electrical engineering from Rutgers University, NJ, in 2009.

She is currently an Associate Professor in the Department of Communication Engineering, Xiamen University, Fujian, China. From 2003 and 2004, she was with North Carolina State University, NC. Her research interests include network security, localization, cognitive radio, radio resource managements, and wireless communications.
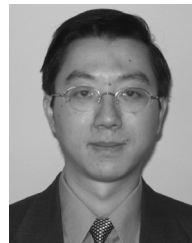
Huaiyu Dai (M'03–SM'09) received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 2002.

He was with Bell Labs, Lucent Technologies, Holmdel, NJ, during summer 2000, and with AT&T Labs-Research, Middletown, NJ, during summer 2001. Currently he is an Associate Professor of Electrical and Computer Engineering at NC State University, Raleigh. His research interests are in the general areas of communication systems and networks, advanced signal processing for digital communications, and communication theory and information theory. His current research focuses on networked information processing and crosslayer design in wireless networks, cognitive radio networks, wireless security, and associated information-theoretic and computation-theoretic analysis.

Dr. Dai has served as editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He coedited two special issues for EURASIP journals on distributed signal processing techniques for wireless sensor networks, and on multiuser information theory and related applications, respectively.

Peng Ning (M'99) is a Professor of Computer Science at NC State University, located in Raleigh, NC, where he also serves as the Technical Director for Secure Open Systems Initiative (SOSI). He served/or is serving on the editorial boards of *ACM Transactions on Sensor Networks*, *Journal of Computer Security*, *Ad-Hoc Networks*, *Ad-Hoc & Sensor Networks: an International Journal*, *International Journal of Security and Networks*, and *IET Proceedings Information Security*. He served as the Program Chair or Cochair of ICDCS-SPCC'10, ESORICS'09, ACM SASN'05, and ICICS'06, the General Chair of ACM CCS'07 and CCS'08, and Program Vice Chair for ICDCS'09 & '10-Security and Privacy Track. He is a Steering Committee member of ACM CCS and a founding Steering Committee member of ACM WiSec. He has served on the organizing committees or program committees for over 60 technical conferences or workshops related to computer and network security. His research has been supported by the NSF, the Army Research Office (ARO), the Advanced Research and Development Activity (ARDA), IBM Open Collaboration Research (OCR) program, SRI International, and the NCSU/Duke Center for Advanced Computing and Communication (CACC).

Prof. Ning is a recipient of the National Science Foundation (NSF) CAREER award. He is a senior member of the ACM, the ACM SIGSAC, and a member of the IEEE Computer Society.