Wormhole-Based Antijamming Techniques in Sensor Networks

Mario Čagalj, Srdjan Čapkun, and Jean-Pierre Hubaux

Abstract—Due to their very nature, wireless sensor networks are probably the category of wireless networks most vulnerable to "radio channel jamming"-based Denial-of-Service (DoS) attacks. An adversary can easily mask the events that the sensor network should detect by stealthily jamming an appropriate subset of the nodes; in this way, he prevents them from reporting what they are sensing to the network operator. Therefore, even if an event is sensed by one or several nodes (and the sensor network is otherwise fully connected), the network operator cannot be informed on time. We show how the sensor nodes can exploit channel diversity in order to create wormholes that lead out of the jammed region, through which an alarm can be transmitted to the network operator. We propose three solutions: The first is based on wired pairs of sensors, the second relies on frequency hopping, and the third is based on a novel concept called uncoordinated channel hopping. We develop appropriate mathematical models to study the proposed solutions.

Index terms—Wireless sensor networks, security, jamming DoS attacks, wormholes, probabilistic analysis, simulations.

1 Introduction

In this paper, we investigate an attack where the attacker masks the event (*event masking*) that the sensor network should detect by stealthily jamming an appropriate subset of the nodes. In this way, the attacker prevents the nodes from reporting what they are sensing to the network operator. Timely detection of such stealth attacks is particularly important in scenarios in which sensors use reactive schemes to communicate events to the network sink [14].

Event masking attacks result in a *coverage paradox*: Even if an event is sensed by one or several nodes (and the sensor network is otherwise fully connected), the network operator cannot be informed on time about the event (see Fig. 1). We will explain how the solution to this problem is far from trivial. Proactive schemes, in which sensors spend their time (and batteries) assessing the state of their communication links, are clearly suboptimal. Equally, jamming detection schemes are generally oversensitive and generate many false alarms, making the system vulnerable to straightforward Denial of Service (DoS) attacks.

We show that *wormholes* [5], which were so far considered to be a threat, can be used as a reactive defense mechanism. In our solution, thanks to channel diversity, the nodes under the jamming attack are able to create a communication route that escapes jamming; thus, appropriate information can be conveyed out of the jammed region. The creation of a wormhole can be triggered by the absence of an acknowledgment after several transmissions. We explain the principle of *probabilistic wormholes* by analyzing three approaches based on this principle. In the first, a network with regular wireless sensor nodes is augmented with a certain number of wired pairs of sensor nodes, therefore resulting in a *hybrid sensor network*. In the second, the deployed nodes (or a subset of them) organize themselves as frequency hopping pairs (e.g., using Bluetooth). For both approaches, we compute the probability that at least one wormhole can be formed. Finally, in the third approach, we propose a novel antijamming technique based on uncoordinated channel hopping. In this approach, the nodes form low-bandwidth antijamming communication channels by randomly hopping between the given set of orthogonal channels. This solution does not require the nodes to be synchronized.

The organization of the paper is as follows: In Section 2, we explain the need for the approach based on wormholes. In Section 3, we focus on the solution based on wired pairs of sensor nodes. In Section 4, we analyze the solution based on frequency hopping. In Section 5, we analyze the solution based on uncoordinated channel hopping. We give related work in Section 6. We conclude in Section 7. Finally, in the Appendix, we develop the mathematical model used in this paper.

2 Motivation and Existing Trade-Offs

We consider the following scenario: A network of wireless sensors is deployed to detect an event (e.g., the presence of a thief in a museum). Upon detection of the event, a (motion) sensor reports it to the network operator, who then reacts accordingly. Any failure by the sensor to report the event would result in the event being undetected by the operator and would prevent any action to be taken (in our example, the presence of a thief would be undetected). This failure can occur for two main reasons: 1) faulty or

- *M. Čagalj is with the Department of Electrical Engineering, Mechanical Engineering, and Naval Architecture (FESB), University of Split, R. Boskovica bb, 21000 Split, Croatia. E-mail: mario.cagalj@fesb.hr.*
- *S. Čapkun is with the Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland. E-mail: srdjan.capkun@inf.ethz.ch.*
- *J.-P. Hubaux is with EPFL-I&C-LCA, BC 207 (BC Building), Station 14, CH-1015 Lausanne, Switzerland. E-mail: jean-pierre.hubaux@epfl.ch.*

Manuscript received 21 Nov. 2005; revised 20 Apr. 2006; accepted 10 May 2006; published online 15 Nov. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0346-1105.

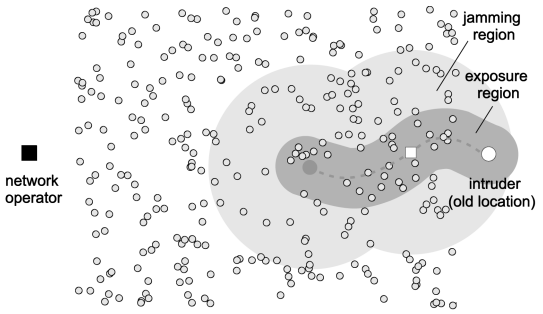


Fig. 1. The *coverage paradox*—even if an intruder is detected by the sensor nodes (and the network is connected), the network operator cannot be informed on time. The intruder moves in the network and gets detected by the nodes located in the *exposure region*; the intruder then stealthily jams all communication within the *jamming region* (the white square represents a jamming device left behind by the intruder on his way).

compromised sensors and 2) unreliable or disrupted communication links. In this work, we focus on the latter.

In a wireless sensor network, all mutual communication between sensors and between the network operator and sensors is wireless (and multihop) [2]. This makes it possible for the attacker to jam the communication between sensors and the operator. We show an example of this scenario in Fig. 1. This figure shows an intruder (adversary) whose presence is sensed by sensors located within the exposure region (the region from which the adversary's presence can be sensed). It also shows that all communication from the sensors (located in the exposure region) to the rest of the network (to their neighboring sensors) is jammed by the adversary (and an additional jamming device—the white square on the figure), resulting in the presence of the adversary not being reported on time to the operator. This example shows that an adversary can, by jamming communication between the sensors, effectively *delay* the report about his presence and, in some cases, prevent being detected at all. Here, we speak about the "delay" as the sensor nodes from the exposure region may eventually detect the jamming activity of the adversary. However, this is not an easy task considering the limited computational capabilities of sensor nodes [14]. At the time a report arrives at the network operator, it may already be too late to take any meaningful action. Note also that the attacker can use a smart jamming strategy to avoid being detected by the nodes that do not sense its presence (the nodes outside the exposure region—Fig. 1). Usually, packets in sensor networks have no protection apart from a simple CRC; therefore, only a short jamming pulse is sufficient to destroy a whole packet [10].

Furthermore, even if jamming is detected, the network operator still cannot precisely locate the adversary; only the boundary of the jamming region can be determined (Fig. 1). Therefore, there is a clear need for defense mechanisms that can ensure *timely data delivery* in spite of jamming attacks. In this work, we assume the existence of an effective attack detection mechanism (see [14]).

2.1 Proactive versus Reactive Sensor Networks

Generally, we distinguish two basic types of sensor networks: proactive and reactive. Proactive networks involve a periodic flow of data between sensor nodes and the sinks. On the contrary, in reactive networks, packets are sent only when some event of interest occurs and is sensed. Reactive networks are characterized by low energy consumption and, therefore, long network lifetimes.

In the case of proactive sensor networks, several simple solutions are possible for ensuring that the operator receives event reports or detects jamming. One solution consists of having sensors periodically report their status to the network operator (e.g., upon query from the operator): If a sensor does not report its status within an expected period, the operator can request a retransmission or conclude that the communication from that sensor is prevented by an adversary. If these status reports are sent very frequently, sensor batteries will be exhausted in a short time, whereas if they are sent infrequently, the batteries will last longer, but the time elapsed between an event happening and its reporting can be long and might render the alarm useless. Another similar solution is that sensors hold the list of their neighbors and periodically poll them to check if the communication links between them are still valid. This solution has similar drawbacks as the first proposal, as it either has high energy cost (if the polls are frequent) or opens a time window within which an event is undetected (if the polls are not frequent).

These and similar proactive solutions require the sensors to periodically communicate even if no event has occurred. Furthermore, these solutions do not ensure that the network operator is informed about the event immediately after it happens. We therefore argue that, instead of being proactive, in many applications, event reporting needs to be reactive, saving energy (as the sensors communicate only when an event is detected) and enabling the network operator to be informed about an event within a reasonably short time period.

Reactive event reporting is, however, vulnerable to jamming. If the communication from a sensor to the operator is jammed, the operator will not raise any alarm as it does not expect any reports to come at any given time. It is therefore important to ensure that, if a sensor detects an event, it can communicate this event to the network operator despite adversary's jamming.

2.2 Our Solution: Probabilistic Wormholes

In our solution, a portion of pairs of sensor nodes create (probabilistically) communication links that are resistant to jamming. By not requiring all the sensor nodes in the network to have this capability, we actually trade off the network robustness with the network complexity (and cost). For the given randomly located adversary (attacker), there is a positive probability that a sensor node residing in the exposure region of the attacker forms a (multihop) path from the exposure region to the region not affected by jamming in such a way that this path is not affected by ongoing jamming. We call such a path a *probabilistic wormhole*. An example of probabilistic wormhole, realized through wires, is shown in Fig. 2a.

In the following three sections, we present and analyze three mechanisms to achieve timely event reporting, namely: 1) *wired pairs of sensor nodes*, 2) *coordinated frequency-hopping pairs*, and 3) *uncoordinated channel-hopping pairs of nodes*.

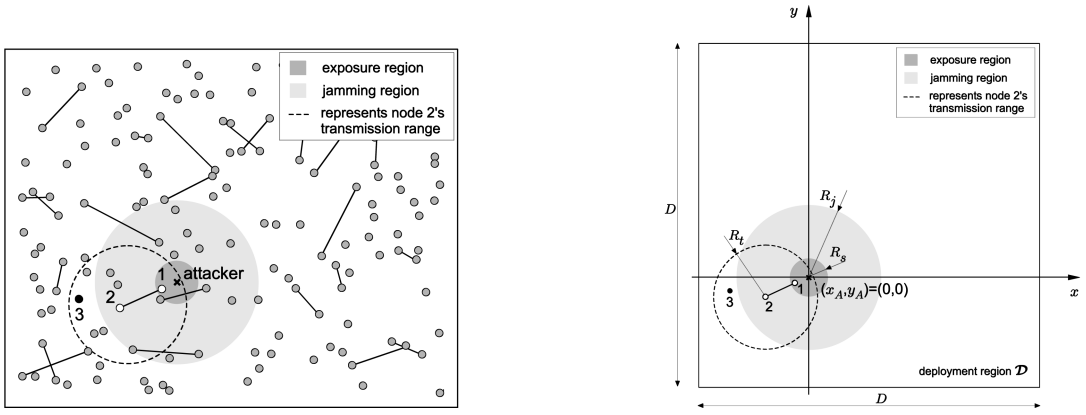


Fig. 2. Probabilistic wormholes via wired pairs of sensor nodes: (a) Hybrid sensor network with randomly deployed sensor nodes: Isolated points are regular nodes, connected points represent sensor nodes connected through a wire. Connected pair (1, 2) and regular node 3 create a *wormhole* that leads out of the exposure region to the region that is not jammed. (b) Geometry used in the analysis of the solution based on probabilistic wormholes.

3 Wormholes via Wired Pairs of Sensor Nodes

In this solution, we propose augmenting a wireless sensor network with a certain number of pairs of sensor nodes that are each connected through a wire. Connected sensor nodes are also equipped with wireless transceivers, just like regular sensor nodes. As a result, we obtain a hybrid sensor network as shown in Fig. 2a: Isolated points represent regular nodes and connected pairs are denoted as connected points. A similar form of hybrid sensor network already appears in the context of the NIMS project [6] and in the work by Sharma and Mazumdar [11].

3.1 Rationale

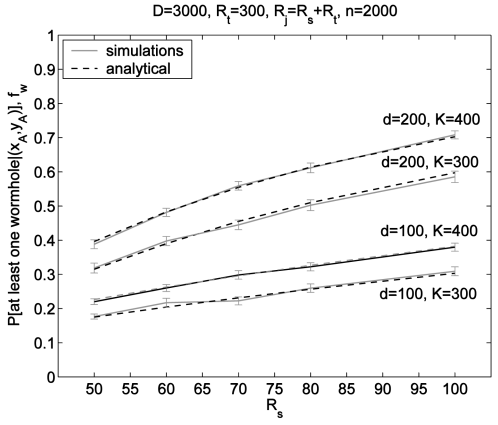
We now explain the operating principles underlying the approach based on wired pairs of sensor nodes. We denote with d the length of the wire connecting a pair of nodes; we assume all pairs to be connected with wires of the same length. Assuming random deployment of connected pairs (e.g., by throwing them from an aircraft), the distance between the nodes of a given connected pair, once the pair lands in the field, is a random variable taking values from interval $[0, d]$. We further denote with R_t the transmission range of the wireless transceivers mounted on the sensor nodes. Let us now consider the scenario shown in Fig. 2a. In this scenario, the attacker (A), represented by sign $\mathbf{x}$, stealthily jams the region (called *jamming region*) within jamming range R_j. We call the *exposure region* the region that surrounds the attacker and from which the attacker's presence can be detected. As can be seen in Figs. 2a and 2b, we model the exposure region by a circle centered at the location of the attacker. We denote with R_s the radius of the exposure region. The exposure region is related to the sensing capabilities of the employed sensors, which is the reason for using subscript s in R_s. Note, however, that the notion of the exposure region is much broader. For example, when the attacker jams an area, the nodes whose transmissions are affected by this attack can deduce that an attack is taking place by observing multiple failures to receive the ACK from their intended destinations. In this case, all such nodes make the exposure region.

In order to prevent any report (e.g., a report about the attacker's presence), generated by the regular nodes located within the exposure region, from successfully leaving the exposure region, the attacker simply jams the area within jamming range $R_j \geq R_t + R_s$. In this situation, the connected pairs serve as a rescue. In our example in Figs. 2a and 2b, the connected pair (1, 2) creates a link resistant to jamming from the exposure region. When node 1 senses the presence of the attacker, it makes use of the wired channel to communicate a short report to its peer node 2. As the wired channel between nodes 1 and 2 is not affected by the jamming activity of the attacker, the report sent by node 1 is successfully received by node 2. In turn, node 2 simply transmits (broadcasts) this report using the wireless transceiver with transmission range R_t. A node (e.g., node 3 in Fig. 2a and Fig. 2b) that is located within transmission range R_t from node 2 and outside of the jamming region will potentially receive the report and pass it further, possibly over multiple hops, to the sink. Therefore, the 2-hop path between nodes 1 and 3 can be thought of as a *wormhole* that is resistant to ongoing jamming activity by the attacker.

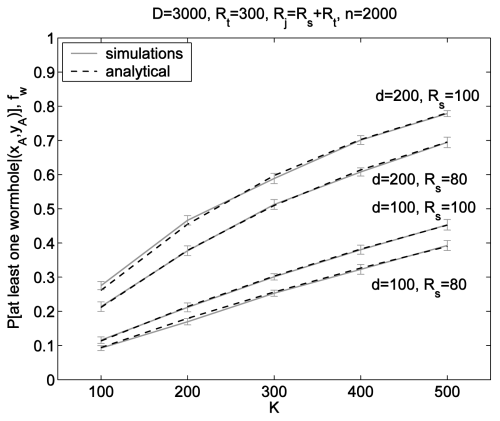
Naturally, the attacker can simply increase the jamming region in such a way that the attacker also jams node 3. However, in the same way, the network operator can further increase the transmission range (R_t) of the wireless transceivers, the length of the wire (d), as well as the exposure region (by deploying more advanced sensors with more advanced sensing capabilities). In addition, if a jamming signal is stronger, the probability that it gets detected and reported increases. In the following section, we develop a model that allows us to better understand the potential benefits of changing the system parameters: R_t, R_s, d, and R_j, as well as the node density.

3.2 Performance Analysis

We assume the regular sensor nodes to be deployed randomly with uniform distribution in the deployment region $\mathcal{D}$ (Fig. 2b). The deployment region $\mathcal{D}$ is modeled by



(a)



(b)

Fig. 3. $P[\text{at least one wormhole } |(x_A, y_A)]$ and relative frequency $f_W(500)$ versus (a) the size of the exposure region R_s and (b) the number of connected pairs K. We use a 95 percent confidence interval.

a $D \times D$ square, $D < \infty$. We denote with n the number of regular nodes deployed in $\mathcal{D}$. We further approximate exposure and jamming regions with circles of radius R_s and R_j, respectively (the Boolean model). Finally, we assume that the jamming range satisfies $R_j \geq R_s + R_t$. The center point $(x_A, y_A) \in \mathcal{D}$ of the exposure (jamming) region represents the location of the attacker (Fig. 2b). In our model, we assume both exposure and jamming regions to be contained completely within the deployment region. This is to avoid cumbersome technicalities with boundary regions. Without loss of generality, we set $(x_A, y_A) = (0, 0)$ (Fig. 2b). We also assume that the attacker is ignorant of the locations of connected pairs.[1] In other words, the attacker's location is assumed to be independent of the locations of the connected pairs.

For the given attacker, located at point $(x_A, y_A) = (0, 0)$, we calculate $P[\text{at least one wormhole } |(x_A, y_A)]$, *the probability that at least one wormhole exists from the corresponding exposure region into the region not affected by the attacker's jamming activity*.

Let $P[S]$ be the probability that an arbitrary pair forms a wormhole from the exposure region around (x_A, y_A) to the area not affected by jamming. Let p_s denote the value of $P[S]$. By assumption: 1) the location of any connected pair (i, j) is independent of the attacker's position (x_A, y_A), and 2) the positions of the connected pairs are sampled from the same distributions and independently. Therefore, p_s is equal for all the deployed connected pairs. Let us denote with K the number of connected pairs deployed randomly and independently. Then, we have:

$$P\big[\text{at least one wormhole}|(x_A, y_A)\big] = 1 - (1 - p_s)^K \approx 1 - e^{-Kp_s}, \tag{1}$$

where the approximation is valid for small p_s and large K. In our analysis (see the Appendix), we obtain a complex expression for probability $p_s = P[S]$ that we solve numerically. We validate our model in the following section by simulations.

Assume now that we want to achieve

$$P[\text{at least one wormhole } |(x_A, y_A)] \geq p_w,$$

where p_w is a targeted probability. Let K_0 denote the critical (minimum) number of connected pairs for which $P[\text{at least one wormhole } |(x_A, y_A)] = p_w$ holds. Then, from (1), we have the following result:

Theorem 1.

$$K_0 = \frac{\ln(1 - p_w)}{\ln(1 - p_s)} \approx -\frac{\ln(1 - p_w)}{p_s}, \tag{2}$$

where p_s is given by (18) in the Appendix.

The result from Theorem 1 is common in stochastic geometry.

3.3 Simulations and Model Validation

We investigate the proposed analytical model (see the Appendix) by means of simulations. We evaluate probability $P[\text{at least one wormhole}|(x_A, y_A)]$ as a function of parameters K, R_s, n, and d. In our simulations, we set $R_j = R_s + R_t$. For each parameter, we perform 20 experiments as follows: For each different value of a given parameter (i.e., R_s, K, n, d), we first randomly generate the network topology with n regular nodes and K connected pairs (see Fig. 2a). Next, we randomly throw $N = 500$ jamming regions (circles of radius R_j) in the deployment area of size $D \times D$. Then, we count the number $n_W \leq N$ of jamming regions for which there is at least one wormhole. From this, we calculate the relative frequency $f_W(N) = n_W/N$. Finally, we average the results obtained from 20 experiments and present them with a 95 percent confidence interval.

The results are shown in Fig. 3 and Fig. 4, together with numerical results obtained from the analytical model developed in the previous section (and the Appendix). As we can see from the figures, the analytical model predicts quite accurately $P[\text{at least one wormhole } |(x_A, y_A)]$. Other

1. This assumption is more legitimate in the context of the solution based on frequency-hopping pairs (studied in Section 4). Note, however, that information about the locations of connected pairs becomes less relevant as the density of the connected pairs increases.

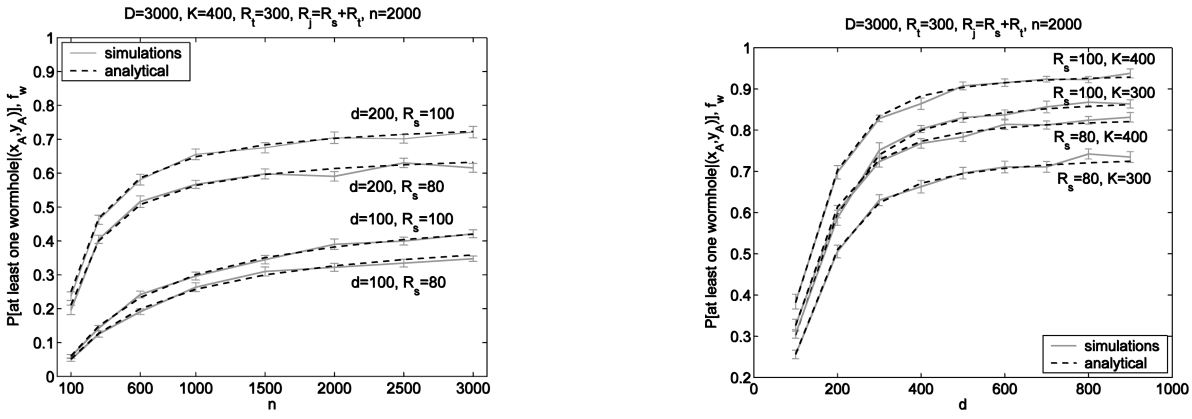


Fig. 4. $P[\text{at least one wormhole}|(x_A, y_A)]$ and relative frequency $f_W(500)$ versus (a) the number of regular nodes n and (b) the maximum wire length d. We use a 95 percent confidence interval.

interesting conclusions can be drawn from the figures. We can see that the increase in either R_s or K results in a nearly linear increase of $P[\text{at least one wormhole } |(x_A, y_A)]$. We can further see that the best "investment" for the network operator is to increase the size of the exposure region (e.g., by using more advanced sensing mechanisms). For example, an increase of R_s of 20 units (from 80 to 100) for $K = 300$ and $d = 200$ results in an increase of $P[\text{at least one wormhole } |(x_A, y_A)]$ of around 0.1 (Fig. 3a). However, an increase of K of 100 units (300 to 400) for $d = 200$ and $R_s = 100$ results in nearly the same increase of $P[\text{at least one wormhole}|(x_A, y_A)]$, i.e., around 0.12 (Fig. 3b). Therefore, we can trade off the number of wired pairs required with the size of the exposure region (for example, by using more advanced sensing technology). The advantage of increasing R_s versus K can easily be seen by taking the first derivative of $P_w \equiv P[\text{at least one wormhole } |(x_A, y_A)]$ with respect to p_s and K. From (1), we have

$$\frac{\partial P_w}{\partial p_s} \approx K e^{-Kp_s} \quad \text{and} \quad \frac{\partial P_w}{\partial K} \approx p_s e^{-Kp_s}.$$

Since p_s increases in R_s, it follows readily that it is more advantageous to increase R_s than K. From Figs. 3a and 3b, we can further see that the cable length plays a major role; we note, however, that this is partially because we take $R_j = R_t + R_s$.

From Figs. 4a and 4b, we observe that increasing n and d is beneficial only until a certain saturation point; this can be easily deduced from our model developed in the Appendix. Note that the average distances between connected peers are significantly shorter than the maximum length d. The average distance between two connected nodes is around $0.45 \times d$ (which is consistent with the expected distance between two randomly selected points from a disk of radius $d/2$ [12]).

The results from this section show that, although feasible, the solution based on pairs of nodes connected through wires is expensive in terms of the number of wires needed and their length. In the following section, we propose and analyze an alternative and "light" approach to creating wormholes.

4 Wormholes via Frequency Hopping Pairs

The solution based on pairs of nodes connected through wires has the obvious major drawback that it requires wires to be deployed in the field. Moreover, as we saw in Section 3.3, in order to achieve a reasonably high $P[\text{at least one wormhole } |(x_A, y_A)]$, the number of connected pairs (and, therefore, wires) to be deployed can be very high. In this section, we propose a solution similar to the previous one, with the only difference being that the pairs are formed exclusively through wireless links resistant to jamming. By using a wireless link, not only do we avoid cumbersome wires, but we can also afford longer links between pairs. As we saw in Section 3.3 (Fig. 4b), the increase in d (maximum length of a wire) has a profound impact on $P[\text{at least one wormhole } |(x_A, y_A)]$.

4.1 Rationale of Frequency Hopping (FH) Pairs

In the solution based on coordinated frequency hopping pairs, we distinguish two types of sensor nodes. The first type are *regular nodes* equipped with an ordinary single-channel radio. The second type are sensor nodes equipped with two radios: the regular radio and a radio with frequency-hopping (FH) capability (e.g., Bluetooth). We note that there already exist several sensor platforms with FH capabilities [1]. It is important to stress, however, that we do not propose to equip all the nodes in the network with FH radios (a case study of Bluetooth sensor networks can be found in [8]). The reason is that FH radios impose a substantial overhead on sensor nodes in multihop networks [8]. The need for "synchronization" (at multiple levels) between senders and designated receivers (synchronization of hopping sequences, time synchronization) may be a major reason against the usage of FH radios in multihop wireless sensor networks [8].

Instead, we propose deploying a certain number of FH-enabled nodes along with the regular nodes. We assume that the attacker cannot jam the employed FH radio. Once deployed (in the bootstrapping phase; no attack takes place yet), each FH-enabled node begins to look for another FH node among its FH neighbors. Once two FH neighboring

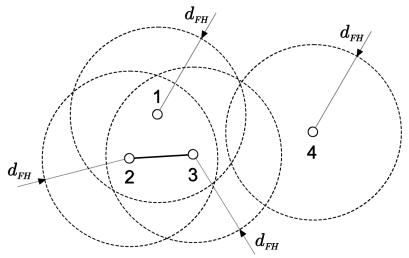


Fig. 5. Opportunistic FH pairing process: The thick line connecting FH nodes 2 and 3 means that they form an FH pair, while FH nodes 1 and 4 remain "unpaired" (d_{FH} is the radio transmission range of the FH nodes).

nodes agree to form an FH pair, they generate a random frequency-hopping sequence (which is, ideally, unique in the 2-hop neighborhood of a given pair). In this work, we restrict each FH node to being a member of, at most, one FH pair. We denote with d_{FH} the transmission range of the FH radio (i.e., FH nodes), where d_{FH} may be different from the transmission range R_t of regular nodes (radio).

The solution based on FH pairs is similar to the previous one based on wired wormholes. Here again, our goal is to ensure that, with a high probability, FH pairs form at least one wormhole in the event of a jamming attack (see Fig. 2a). The important difference with respect to the solution based on wires is that the formation of FH pairs takes place once the nodes are deployed in the field—the *opportunistic pairing process*. FH hopping-enabled nodes will use some form of a *pairing protocol* to discover their FH-enabled neighbors and to eventually form a pair with one of them. A simple opportunistic pairing protocol would be to let every node advertise its availability until it makes an FH pair with a randomly selected "available" node or it fails to find some "free" (available) neighbor. The details of such a pairing protocol are out of the scope of this work. We expect it to be probabilistic in nature[2] (for example, due to the probabilistic channel access mechanisms). For this reason (and because of the random deployment of FH enabled nodes), it is very likely that some FH nodes will not find any "free" FH neighbor.

Consider the example in Fig. 5, where FH nodes 1, 2, and 3 are all neighbors to each other (i.e., they are located within d_{FH} of each other) and FH node 4 has no neighbors. The link between nodes 2 and 3 means that they form an FH pair. Since we allow each node to be a member of at most one FH pair, node 1 has no "free" FH neighbors to form a pair with. Likewise, node 4 has no FH neighbors at all and, so, remains "unpaired" too. From this simple example, we can see that the event that some FH node i forms a pair with its FH neighboring node j is *not* independent of the status of the other FH nodes from the i and j's neighborhood. This fact makes the analytical study of the FH pairs-based solution far more difficult. We will now show how to effectively overcome this difficulty.

2. An alternative would be to use a similar approach as in the probabilistic key predistribution schemes [4], where the nodes would be preloaded with a certain number of FH sequences chosen randomly from a common pool.

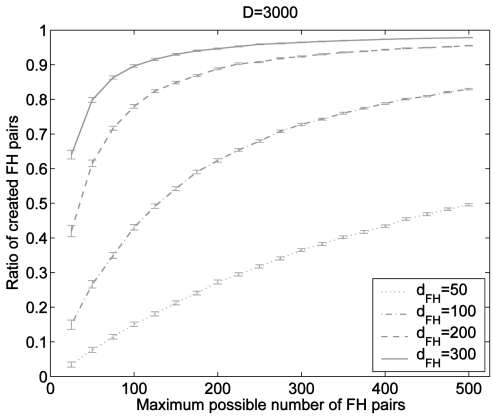


Fig. 6. Ratio of created FH pairs versus maximum possible number of FH pairs ($= 1/2 \times$ the number of FH enabled nodes deployed); we use a 95 percent confidence interval.

4.2 Analysis of the FH Pairs-Based Solution

Again, our goal is to estimate

$$P[\text{at least one wormhole } |(x_A, y_A)],$$

the probability that at least one FH pair forms a wormhole from the exposure region to the region not affected by jamming. As we discussed in the previous section, due to the probabilistic nature of the pairing process, not all deployed FH nodes are guaranteed to be a member of some FH pair. To better understand the extent of this potential difficulty, we have conducted the following simulations: We randomly throw a certain number of FH-enabled nodes in a deployment region of size $D \times D$ with $D = 3{,}000$. Then, we combine FH nodes randomly into FH pairs, with the restriction that a single FH node can be a member of at most one FH pair and two FH nodes can make a pair only if they are within distance $d_{FH} = \{50, 100, 200, 300\}$ of each other. For each different transmission range and the number of FH nodes, we generate 100 network instances. For each instance, we count the number of FH pairs created. The average number of FH pairs, with 95 percent confidence intervals, is presented in Fig. 6.

From this figure, we can see that, except for modest transmission ranges (e.g., $d_{FH} = 50$), the number of created FH pairs is sufficiently high. As expected, the larger the density of the FH nodes is, the larger the number of created FH pairs is. Therefore, with an appropriately selected radio transmission range of FH nodes, we can ensure that almost all of the FH nodes will be effectively used.

From the same set of simulations, we have extracted two additional values, namely, the average distance between two FH nodes that make an FH pair (the normalized average distance of an FH link) and the corresponding standard deviation. In Fig. 7, we show the normalized average distance between two FH peers and the corresponding standard deviation as functions of the number of the deployed FH nodes. We normalize the distance with respect to the corresponding radio transmission range d_{FH}. A striking result in this figure is that the normalized average distance of an FH link is approximately $0.66 \approx \frac{2}{3}$, irrespective of d_{FH}. Moreover, the standard deviation is approximately 0.23.

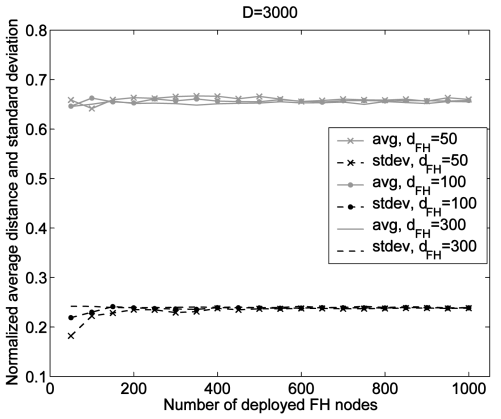


Fig. 7. Normalized average distance between FH peers versus the number of FH enabled nodes deployed ("avg"—average, "stdev"—standard deviation).

This result reminds us of the process of choosing a random point (x, y) from the unit circle centered at point (x_0, y_0). Then, we can calculate the expected distance $E[L]$ between points (x, y) and (x_0, y_0) to be $E[L] = \frac{2}{3}$ and the standard deviation $STD(L) = \sqrt{1/18} \approx 0.2357$. Indeed,

$$f_L(x) = \frac{2x\pi}{r^2\pi} = \frac{2x\pi}{1^2\pi} = 2x, \quad E\big[L\big] = \int_0^1 x f_L(x) = \int_0^1 2x^2 = \frac{2}{3},$$
$$STD(L) = \sqrt{\int_0^1 x^2 f_L(x) - \big(E\big[L\big]\big)^2} = \sqrt{\frac{1}{18}}. \tag{3}$$

This result suggests that the random process of opportunistic FH pairing exhibits behavior similar to the process of choosing a random point from the circle of radius d_{FH} centered at the given FH node. To confirm this hypothesis, we performed another set of experiments. For the given transmission range d_{FH}, we partition length d_{FH} into a certain number of mutually exclusive intervals, each of the same size δ. Then, we generate a large number of networks (for the fixed parameters d_{FH}, K, and D) and determine the relative frequency with which distances between created FH pairs fall into each interval. Finally, we compare the relative frequency with the corresponding probability obtained from the probability density function given in (3).

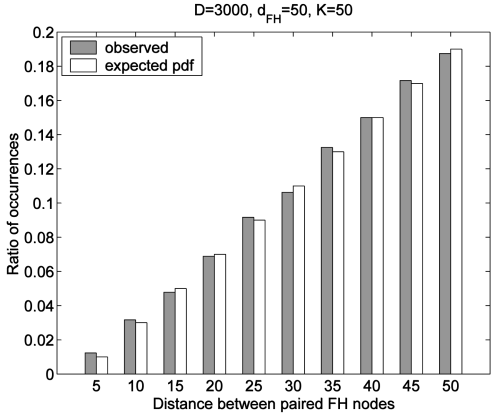
As can be seen from Fig. 8a and Fig. 8b, the relative frequency matches the probability calculated from the postulated probability density function (3) very well. This is the case even for low values of d_{FH} and K.

This matching inspires the following approach to modeling the creation of a random FH pair in the opportunistic pairing protocol: Consider an FH node i that is a member of some FH pair. Then, we model the creation of this FH pair, from the FH node i's point of view, as choosing a random point from the circle with radius d_{FH}, centered at node i. Moreover, since FH nodes are deployed randomly and independently of each other, the creation of one FH pair is independent of the creation of another FH pair in the random point choosing model. Then, from the independence between different created FH pairs, $P[\text{at least one wormhole } |(x_A, y_A)]$ can be calculated as follows:

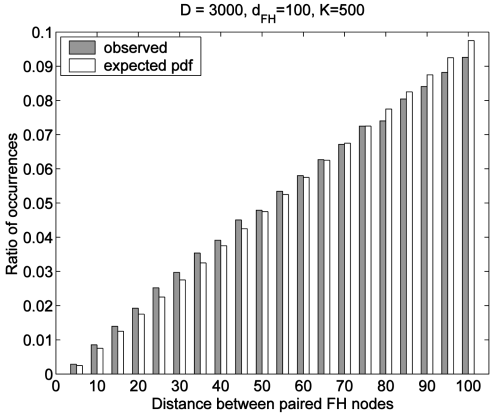
$$P\big[\text{at least one wormhole}|(x_A, y_A)\big] = 1 - \big(1 - p_s^{FH}\big)^{K_{FH}} \approx 1 - e^{-K_{FH}p_s^{FH}}, \tag{4}$$

where p_s^{FH} is the probability that a single FH pair forms a wormhole and K_{FH} is the number of created FH pairs.

In order to calculate p_s^{FH}, we can proceed as in the case of the probability p_s for wired pairs. However, instead of calculating p_s^{FH} from scratch, we prefer to reuse the model developed for wired sensor pairs (Section 3.2 and the Appendix) by exploiting the similarity between the solution based on wired pairs and the solution based on FH pairs. In this direction, we will first establish the relationship between the maximum wire length d and the transmission range of FH node, d_{FH}. As we will see, the important difference between wired pairs and FH pairs is that the latter achieve the same $P[\text{at least one wormhole } |(x_A, y_A)]$ with transmission ranges d_{FH} smaller than the maximum wire length d, i.e., $d_{FH}/d \approx 0.6791$.



(a)



(a)

Fig. 8. Matching between postulated pdf and the relative frequency with which outcomes fall in different intervals of size $\delta = 5$: (a) $d_{FH} = 50$, $K = 50$, number of $\text{experiments} = 3{,}500$; (b) $d_{FH} = 100$, $K = 500$, number of $\text{experiments} = 10{,}000$.

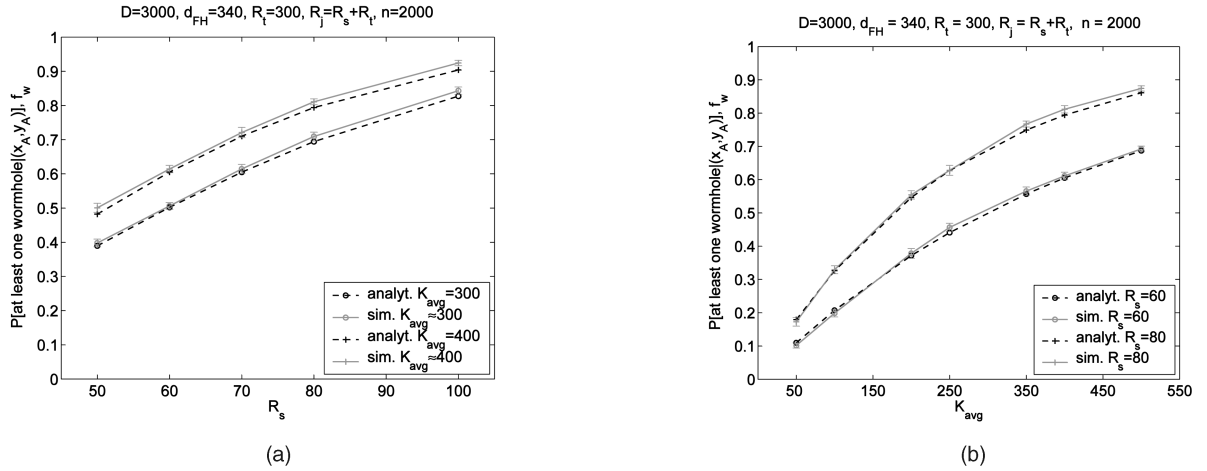


Fig. 9. $P[\text{at least one wormhole}|(x_A, y_A)]$ and relative frequency $f_W(500)$ versus (a) the size of the exposure region R_s and (b) the average number of connected pairs K_{avg}. We use a 95 percent confidence interval.

Note first that there is a subtle difference in the way we model the deployment of pairs connected through wires and the way we model the creation of FH pairs. In the first case, we use the so-called "disk line picking" model, i.e., two points are selected randomly and independently from the disk of radius $\frac{d}{2}$ (d is the maximum cable length). A well-known result from stochastic geometry says that the expected distance between two randomly selected points from the disk of radius $\frac{d}{2}$ is $\frac{128}{45\pi}\frac{d}{2}$ [12]. In the second case, one point (FH node i) is given and its FH peer is modeled as a random point selected from the circle of radius d_{FH}, centered at the location of FH node i. We have established above that the expected distance between two such selected points is $\frac{2}{3}d_{FH}$. Now, the key step in our modeling is that, for the given d_{FH}, we scale d (used in the expressions of Section 3.2) in such a way that the expected distances between the random points in the "disk line picking" model and the random points in the model describing the creation of FH pairs are equal, that is, $\frac{128}{45\pi}\frac{d}{2} = \frac{2}{3}d_{FH}$. From this, it follows:

$$d \approx \frac{d_{FH}}{0.6791}. \tag{5}$$

Now, in order to calculate

$$P[\text{at least one wormhole } |(x_A, y_A)]$$

for the solution based on FH pairs, we first scale d using (5) and use d to calculate $p_s = P[S]$ (see Section 4.3). Then, for the given number of deployed FH nodes, we estimate the average number of created FH pairs (see Fig. 6) and use this value as K in (1). In the following section, we evaluate the proposed model.

4.3 Simulations and Model Validation

We investigate the proposed analytical model by means of simulations. We evaluate probability

$$P[\text{at least one wormhole } |(x_A, y_A)]$$

as a function of parameters K_{FH}, R_s, d_{FH}, and n. As before, we set $R_j = R_s + R_t$. For each parameter, we perform 20 experiments as follows: For each different value of a given parameter, we first randomly generate the network topology with n regular nodes and K_{FH} FH nodes. To simulate the FH pairing protocol, we randomly iterate through the FH nodes (K_{FH}) and, for each unmatched FH node i, we try to find another unmatched FH node from i's neighborhood. In case node i has more than one free FH neighbor, i is matched with a randomly selected one; note that some FH nodes may happen to remain unmatched at the end of the pairing protocol.

Next, we randomly throw $N = 500$ jamming regions (circles of radius R_j) in the deployment area of size $D \times D$. Then, we count the number $n_W \leq N$ of jamming regions for which there is at least one wormhole. From this, we calculate the relative frequency $f_W(N) = n_W/N$ for each different value of the given parameter. Finally, we average the results obtained from 20 experiments and present them with a 95 percent confidence interval. To obtain the numerical results, for each value of d_{FH}, we first scale d using (5) and then we plug the resulting d into (1) to obtain $P[\text{at least one wormhole } |(x_A, y_A)]$. The values of K are obtained as the average number of created FH pairs for different numbers of FH nodes K_{FH} (see Fig. 6).

The results are shown in Figs. 9 and 10, together with numerical results obtained from the analytical model. In the figures, K_{avg} represents the average number of created FH pairs. As we can see from the figures, the analytical model quite accurately predicts $P[\text{at least one wormhole } |(x_A, y_A)]$. The results obtained have identical properties as in the solution based on pairs connected through wires. The important difference is that the FH approach achieves the same $P[\text{at least one wormhole } |(x_A, y_A)]$ with transmission ranges d_{FH} smaller than the maximum wire length d, i.e., $d_{FH}/d \approx 0.6791$ (5).

5 Wormholes via Uncoordinated Channel-Hopping

The solution based on the coordinated FH pairs, though simple, still requires a certain level of synchronization between the FH nodes that make a pair. In this section, we

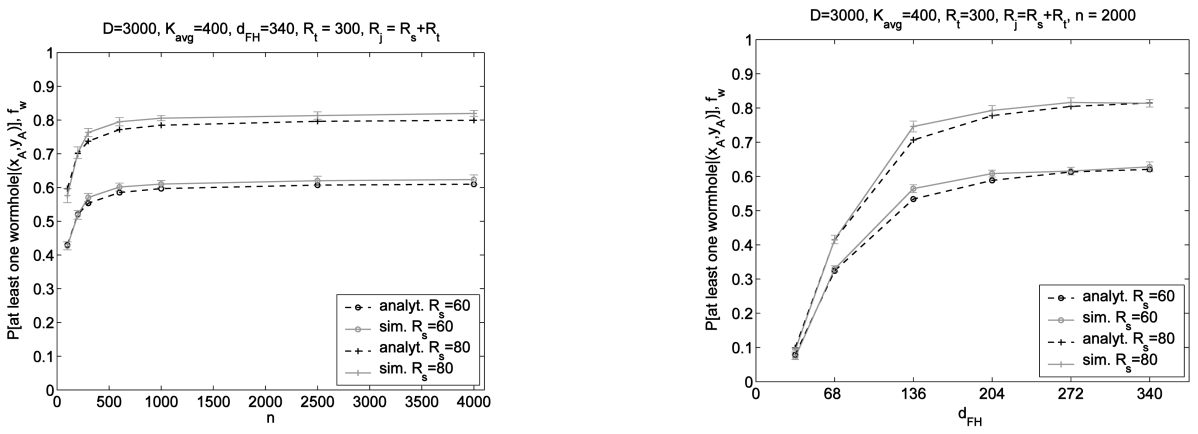


Fig. 10. $P[\text{at least one wormhole}|(x_A, y_A)]$ and relative frequency $f_W(500)$ versus (a) the number of regular nodes n and (b) the transmission range of FH enabled nodes d. We use a 95 percent confidence interval.

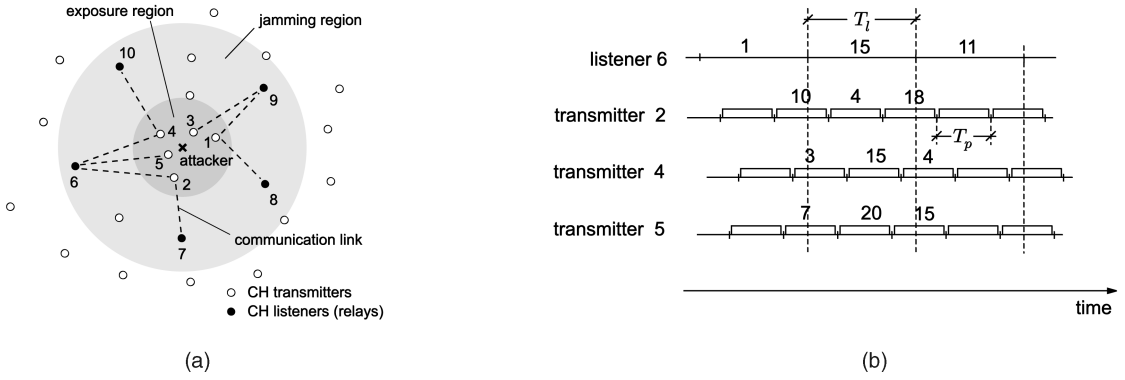


Fig. 11. (a) A network example with channel-hopping listeners. (b) Example of scheduling for nodes 2, 4, 5, and 6, with $T_l = 2T_p$ (the numbers above the packets represent channel indexes).

explore the feasibility of a completely uncoordinated *channel-hopping* approach. In this solution, we seek to create *probabilistic wormholes* by using sensor nodes that are capable of hopping between radio channels that ideally span a large frequency band. The major difference between channel-hopping (CH) and frequency-hopping is that, with the former one, an entire packet is transmitted on a single channel. In other words, with channel-hopping, sensor nodes hop between different channels (frequencies) in a much slower way (per packet basis), as compared to classical frequency-hopping (e.g., Bluetooth).

5.1 Rationale of the Approach

In this approach, we can imagine that a part of the deployed nodes—or all of them—have channel-hopping capabilities. Regular communication still takes place over a single channel common to all the nodes. We do not assume channel hopping nodes to be either coordinated or synchronized (see an example of scheduling in Fig. 11). However, we assume that all the channel-hopping nodes share the common pool of orthogonal channels.

When a channel-hopping sensor node senses the presence of an attacker, it first tries to transmit the report about this event to its neighbors. Each such report should be acknowledged by the intended receivers. In case no (or very few) acknowledgment is received, the node can conclude that an attacker is obstructing his communication. The node then switches to the channel-hopping mode and repeatedly transmits the same report over different orthogonal channels. In order for this report to potentially be received, the transmitting node must have at least one neighbor (with channel-hopping capabilities) that listens on one of those channels. Note that we do not assume the two nodes to be synchronized or coordinated. Therefore, the two nodes will happen to occupy the same channel only with some probability. Note also that the attacker can potentially jam this channel. We can likewise envision a scenario in which a set of specialized *relaying-only* nodes are deployed. Relaying-only nodes would spend most of the time in the listening mode, hopping randomly among the available orthogonal channels.

When such a node happens to receive the report from the exposure region, it can forward the report further either over the regular channel or by entering in the channel hopping mode.

For this approach to work, we have to ensure that it is not sufficient for the attacker to destroy a whole packet by simply flipping one or a few bits of the packet. Otherwise, a fast-hopping attacker could easily destroy all the packets transmitted by quickly hopping between the operational channels and jamming every channel for a very short period of time. By encoding packets using appropriate

error-correcting codes (e.g., *low-density parity-check* (LDPC) codes), we can achieve a certain level of resistance against jamming [10], which we capture by the notion of a *jamming ratio* (defined in the following section). In this way, we can "keep" the attacker "busy" on one channel for some minimum amount of time (which will depend on the jamming radio), while giving an opportunity to transmissions on the other channels to successfully finish.

5.2 System Model and Assumptions

Let us first introduce some notations. Let I denote the set of nodes from the exposure region, which have the channel-hopping capability and which have at least one channel-hopping neighbor outside of the exposure region: In Fig. 11a, $I = \{1, 2, 3, 4, 5\}$. Let O be the set of channel-hopping nodes that reside outside of the exposure region and that have at least one channel-hopping neighbor in the exposure region: In Fig. 11a, $O = \{6, 7, 8, 9, 10\}$. Also, let I_i be the set of channel-hopping neighbors from I of node $i \in O$: In Fig. 11a, $I_6 = \{2, 4, 5\}$, $I_7 = \{2\}$, $I_8 = \{1\}$, $I_9 = \{1, 3\}$, and $I_{10} = \{4\}$.

We assume that there are $(m + 1)$ orthogonal channels available to the sensor nodes. One channel is reserved for the normal mode of operation, i.e., when there is no attack. We further assume that the nodes from the set I always transmit, whereas the nodes from the set O are always in the listening mode. Both the transmitting nodes and the listening nodes randomly hop between different channels, i.e., the probability of selecting any given channel for the next hop is $1/m$. We assume that an attacker knows this strategy, including the channels allocated for hopping.

Further, we denote with T_p and T_l the duration of a packet transmitted by node $i \in I$ and the period during which node $j \in O$ is listening, respectively. By setting $T_l \geq 2T_p$, we can ensure that, even if $j \in O$ and $i \in I_j$ are not synchronized, at least one packet of i will fall within period T_l of listener j (see Fig. 11b). In our analysis, we set $T_l = 2T_p$.

We characterize the strength of the attacker by the following two metrics: 1) *channel sensing time* T_s (i.e., the time it takes to scan a given channel to detect some activity) and 2) the number of channels, m_j, that the attacker can jam simultaneously. We denote with T_j the minimum jamming period that the attacker has to jam a given transmission in order to destroy the corresponding packet. Finally, we define the *jamming ratio* (ρ_j) as follows:

$$\rho_j \stackrel{def}{=} \frac{T_j}{T_p} \leq 1. \tag{6}$$

The higher ρ_j is, the more resistant the packets are to jamming. Note that our game makes sense only if the jamming ratio is sufficiently high. In [10], Noubir and Lin present a set of different coding strategies (based on *low-density parity-check* (LDPC) codes) that can achieve $\rho_j = 0.1 - 0.15$.

5.3 Attacking Strategies

We assume that the attacker does not have information about potential collisions between multiple simultaneous transmissions by nodes from set I; the less information about set O the attacker has, the more realistic this assumption is. We next derive a reasonable jamming strategy for the attacker in our model.

Clearly, if the attacker visits (scans) a given "busy" channel (occupied by transmission), it is optimal for him to jam it. Otherwise, the attacker would not check this channel in the first place. The attacker has two alternatives: 1) scan a channel and then jam it if necessary and 2) jam every channel visited (without scanning it). When scanning channels, the attacker spends either T_s or $T_s + T_j$ per channel, depending on whether the visited channel is busy or not. This strategy is advantageous for the attacker if $T_s < T_j$ and if the attacker has fast enough hardware to sense the channel. Otherwise, jamming every channel visited for the duration T_j may be a better choice.

Let us now consider a fixed packet (carrying a report about the attacker's presence) that can potentially be received by some listening node $i \in O$. To destroy this packet, the attacker needs to jam the channel on which the packet is being transmitted before a fraction $(1 - \rho_j)$ of the packet has been transmitted because packets are "protected" with an LDPC code. Assuming that the attacker adopts the strategy by which he simply jams every channel visited, he has at most

$$k = \left\lfloor \frac{(1 - \rho_j)T_p}{T_j} \right\rfloor m_j = \left\lfloor \frac{1}{\rho_j} - 1 \right\rfloor m_j \tag{7}$$

chances to jam the "correct" channel (the one carrying the fixed packet). Because transmitters choose their channels uniformly at random (i.e., with probability $1/m$, m being the number of orthogonal channels) and, from the attacker's point of view, any packet transmitted can potentially be received by some listening node (i.e., the attacker has no information about set O, the set of listening nodes), the best that he can do is to randomly choose k different channels (see (7) above) and jam those channels for a duration of T_j. The probability p_{jam} that the attacker successfully jams the fixed packet can thus be bounded as follows:

$$p_{jam} \leq \frac{k}{m} = \left\lfloor \frac{1}{\rho_j} - 1 \right\rfloor \frac{m_j}{m}. \tag{8}$$

If the attacker chooses to scan channels before potentially jamming the occupied ones, then p_{jam} can be approximated as $\min\left\{ \left\lfloor \frac{(1-\rho_j)T_p}{\overline{T}} \right\rfloor \frac{m_j}{m}, 1 \right\}$, where $\overline{T}$ is the expected time that the attacker spends per channel visited; note that $T_s \leq \overline{T} \leq T_j + T_s$. Therefore, the attacker's advantage to successfully jam a fixed packet increases (at most) linearly with m_j (the number of channels that he can jam simultaneously). As a countermeasure, the network operator can potentially increase the jamming ratio ρ_j, the number of hopping channels m, and the number of transmitting nodes $(|I|)$. Note, however, that the values of m and $|I|$ should be carefully controlled in order to avoid degradation in reporting performances due to the fact that listening and transmitting nodes are not coordinated and are likewise due to the increased number of simultaneous transmissions.

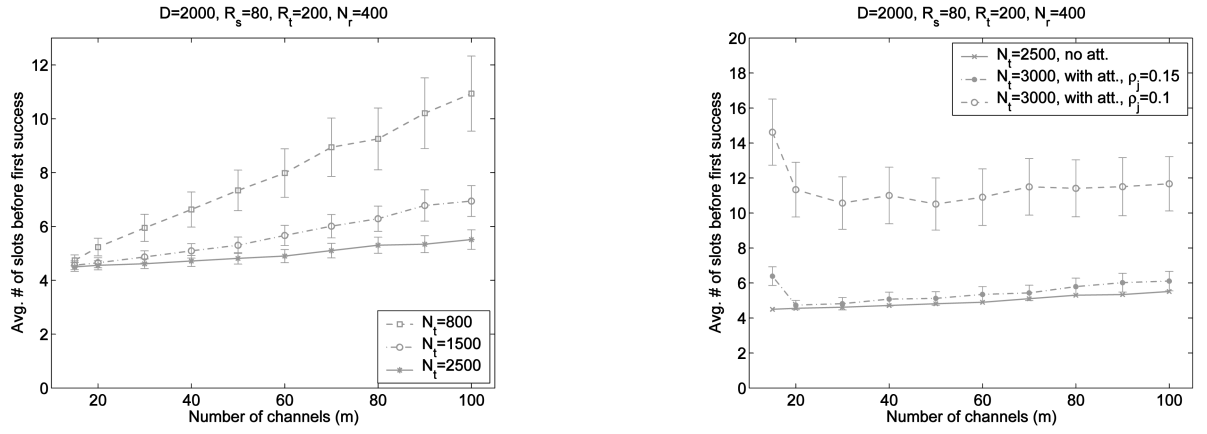


Fig. 12. Average number $\overline{N}_{succ}$ of time slots before the first packet is successfully received when (a) the attacker is not active (does not jam) and (b) the attacker is active. We use a 95 percent confidence intervals.

5.4 Performance Analysis

We carried out an evaluation of this approach using simulations written in Matlab. For a given attacker, we are interested in calculating the average number $\overline{N}_{succ}$ of time slots until the first report (from the exposure region around the attacker) is received by any listening node located outside the exposure region. Here, each time slot is T_p long (i.e., equal to the time it takes to a sensor node to transmit a packet).

In our simulations, we consider an *optimal* attacker who knows in advance which channels are to be active, thus avoiding the cost of visiting nonactive channels (equivalently, the sensing time $T_s = 0$). However, in these simulations, we consider the case with $m_j = 1$ (i.e., the attacker jams at most one channel at a time). We have implemented the following attacking strategy: Every T_j period, the attacker chooses one channel that has not been visited for the longest time among currently active channels.

We perform the following experiment for 20 randomly generated networks of size $D \times D$, with $D = 2{,}000$: For every network, we first deploy uniformly at random N_r listening (relaying) nodes and N_t channel-hopping transmitting nodes. Then, for every network, we randomly pick the location of the attacker. The attacker's location, together with the radius of the exposure region R_s and the radius of the transmission range R_t, define sets I and O. For each such scenario and fixed number m of hopping channels, we generate 50 random (hopping) schedules for both the transmitting nodes (from set I) and the listening nodes (from set O). We emulate desynchronization between the nodes by randomly shifting the generated schedules in time. For every set of random schedules, we record the time slot at which the first packet from the exposure region is successfully received by any node from O. We repeat our experiments for a different number m of hopping channels. For each fixed channel number, we average the results across the 20×50 above experiments.

The results are presented in Figs. 12a and 12b with a 95 percent confidence interval. In Fig. 12a, we plot the results for the case when the attacker is not active. From this figure, we can observe that the average number $\overline{N}_{succ}$ of time slots before the first success increases in the number of orthogonal channels m. It is important to observe that, for $m = 1$, we do not necessarily have collisions at the listening nodes all the time. The reason is that, depending on the node density, for some listening node $i \in O$, we will have $|I_i| = 1$ with a high probability. Another important observation is that $\overline{N}_{succ}$ decreases in the density of transmitting nodes from set I (i.e., in N_t, for fixed D).

Next, we observe $\overline{N}_{succ}$ in scenarios with an active attacker. The results for $\rho_j = \{0.1, 0.15\}$ are shown in Fig. 12b. Note that $\rho_j = 0.1$ and $\rho_j = 0.15$ imply that the attacker can successfully jam at most $1/0.1 = 10$ and $1/0.15 \approx 7$ packets during time period T_p. In this figure, the curve obtained for $N_t = 2{,}500$ and no attacker case serves as a reference point. As expected, for the values of m very close to (or lower than) ρ_j^{-1}, $\overline{N}_{succ}$ grows sharply, essentially meaning that the network will fail to deliver alarms. However, as m grows above ρ_j^{-1}, the value of $\overline{N}_{succ}$ stabilizes at a reasonably small value. For example, for $N_t = 3{,}000$ and $\rho_j = 0.1$, $\overline{N}_{succ}|_{m=15} = 15$ and $\overline{N}_{succ}|_{m \geq 20} \approx 11$. From this figure, we further observe that, as we increase the resistance of packets ρ_j to jamming, we can achieve a significant reduction in $\overline{N}_{succ}$.

In Figs. 13a and 13b, we plot histogram (distribution) of the number (N_{succ}) of transmissions before the first success for $m = 20$. From the two figures, we can see that the frequency of N_{succ} resembles geometric distribution (a somewhat expected result). On Fig. 13a, we can observe a jump at $N_{succ} = 70$. This is because we round all the realizations with $N_{succ} > 70$ down to value of 70. Finally, we can observe that a variance of the N_{succ} is much higher in the case $\rho_j = 0.1$ compared to $\rho_j = 0.15$. This can be explained by considering N_{succ} as a geometric random variable with variance $VAR(N_{succ})$, where $VAR(N_{succ}) = \frac{1-p_s}{p_s^2}$ and p_s is the probability that at least one report leaves the jamming region in a single time slot of duration T_p. As p_s increases in ρ_j, variance $VAR(N_{succ})$ simply decreases.

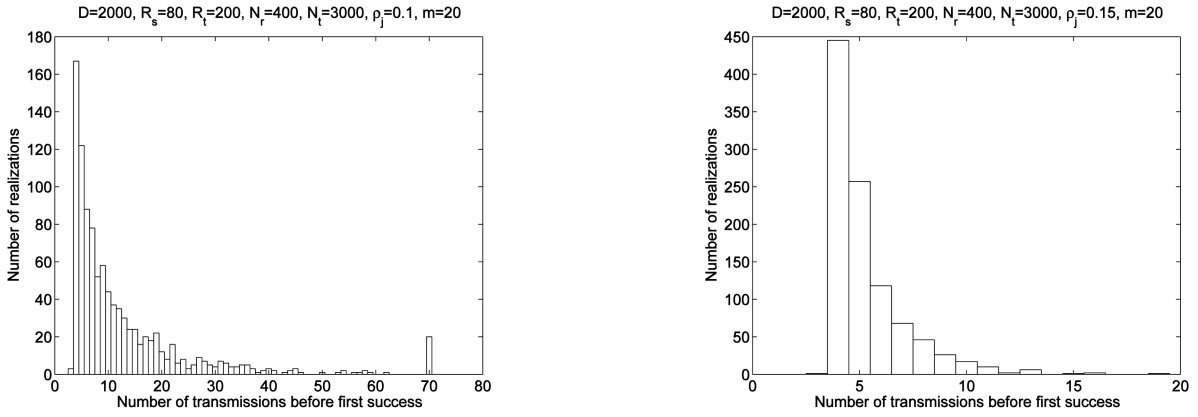


Fig. 13. Distribution of the "number of transmissions before the first success" for $m = 20$ and (a) $\rho_j = 0.1$ and (b) $\rho_j = 0.15$. The number of samples is 1,000.

6 Related Work

Recently, the issues of jamming detection and prevention in wireless sensor networks have received significant attention. In [3], Wood and Stankovic briefly study potential techniques to avoid jammed regions. A more elaborate study was presented by Wood et al. in [13]. In this work, they propose a proactive protocol that first detects and then maps the jammed area. In their approach, each node is assumed to have a detection-module that periodically returns a JAMMED or UNJAMMED message. The message output by the detection module is then broadcast locally. In our approach, however, we propose reactive solutions that do not require periodic exchange of information. Xu et al. [15] propose two countermeasures for coping with jamming: coordinated channel-hopping and spatial retreats, both of which require the nodes to be well synchronized and coordinated. It is not clear whether the solution based on spatial retreats is appropriate for sensor networks. In [15], Xu et al. study the feasibility of reliably detecting jamming attacks. They show that reliable detection can be a challenging task in wireless sensor networks. Moreover, all the proposed detection mechanisms are by their nature proactive. In [10], Noubir and Lin show how to use low density parity check (LDPC) codes to cope with jamming. In [7], Karlof and Wagner introduce a new attack against wireless sensor networks called sinkholes. In [9], McCune et al. propose a scheme for the detection of denial-of-message attacks on sensor network broadcasts.

7 Conclusion

In this paper, we have described in detail how an attacker can mask some events by stealthily jamming an appropriate subset of the nodes. We have shown how these attacks can be thwarted by means of probabilistic wormholes based on wires, frequency hopping, and uncoordinated channel hopping. We have developed appropriate mathematical models for the solutions based on wired and frequency-hopping pairs and we have quantified the probability of success in all three solutions.

It is clear that the space of investigation in this area is huge. Other solutions can be envisioned and, for the three that we have presented, the influence of other parameters can be studied. Yet, we believe that this work provides useful insights on how to quantify the effectiveness of wormhole-based defense mechanisms. In terms of future work, it would be interesting to evaluate the performance of hybrid solutions by combining the three approaches proposed in this paper. The effect of interference between nodes that belong to two or more jamming regions (and its dependency on the node density) is also a subject for future work.

Appendix

Analytical Model for the Solution Based on Wired Pairs

For the given attacker, located at point $(x_A, y_A) = (0, 0)$, we want to calculate the probability that at least one wormhole exists from the corresponding exposure region into the region not affected by the attacker's jamming activity, i.e., $P[\text{at least one wormhole } |(x_A, y_A)]$.

To model the random deployment of connected pairs, we proceed as follows: Let us consider connected pair $(4, 5)$ in Fig. 14. We first choose a point $(x_{4,5}, y_{4,5})$ uniformly at random from $\mathcal{D}$. Next, we draw (or, rather, imagine) a *deployment disk* of radius $d/2$ around the point $(x_{4,5}, y_{4,5})$ (Fig. 14). Finally, we choose two points (x_4, y_4) and (x_5, y_5), uniformly at random and independently, from the area enclosed by the deployment disk centered at $(x_{4,5}, y_{4,5})$; (x_4, y_4) and (x_5, y_5) then correspond to the positions of connected nodes 4 and 5, respectively (Fig. 14). Note that the deployment disk (with diameter d) ensures that the link (wire) between nodes 4 and 5 does not exceed the maximum length of d. This procedure is then repeated (independently) for each of the K connected pairs to be deployed.

More formally, with each connected pair (i, j) to be deployed in the deployment region $\mathcal{D}$, we can associate three two-dimensional random variables: $\mathbf{P}_{i,j} = (X_{i,j}, Y_{i,j})$, $\mathbf{P}_i = (X_i, Y_i)$, and $\mathbf{P}_j = (X_j, Y_j)$, where $X_{i,j} \in [0, D]$ and $Y_{i,j} \in [0, D]$ are uniform (continuous) random variables, and (X_i, Y_i) and (X_j, Y_j) are (jointly continuous) uniform random variables taking values from the set

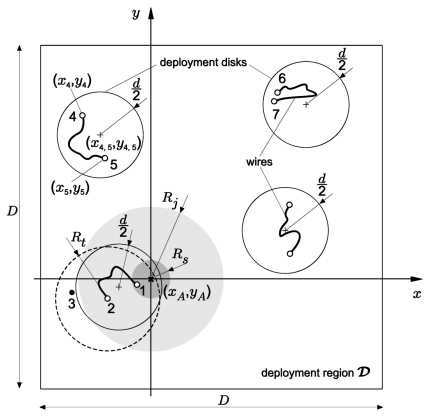


Fig. 14. Approximation model for random deployment of connected pairs (the thick curves connecting the nodes represent wires between the nodes).

$$\{(x,y) : (x-x_{i,j})^2 + (y-y_{i,j})^2 \le (d/2)^2, \text{ for fixed } (x_{i,j},y_{i,j}) \in \mathcal{D}\}.$$

Thus, for the given connected pair (i,j), $\mathbf{P}_{i,j}$ describes the location of the center point of the corresponding deployment disk, while $\mathbf{P}_i$ and $\mathbf{P}_j$ describe the locations of nodes i and j, respectively.

Let us consider a single connected pair (k,l). To calculate $P[\text{at least one wormhole } |(x_A,y_A)]$, we first define the following event:

$$S \stackrel{def}{=} \{\text{the connected pair } (k,l) \text{ forms a wormhole from the exposure region around } (x_A,y_A) \text{ to the area not affected by jamming}\}.$$

It is important to stress here that we require a wormhole to always involve at least one regular node, even in cases when the connected pair itself is sufficient to form a wormhole from the jamming region (for example, this may happen when $d > R_s + R_j$).

Let $P[S]$ be the probability of event S and let p_s denote the value of $P[S]$. Expression (1) in Section 3.2 gives a relationship between $P[S]$ and $P[\text{at least one wormhole } |(x_A,y_A)]$. For this reason, we next calculate $p_s = P[S]$.

From the definition of the random variable $\mathbf{P}_{k,l} = (X_{k,l}, Y_{k,l})$, we know that its probability density function satisfies $f_{\mathbf{P}_{k,l}}(x,y) = f_{X_{k,l},Y_{k,l}}(x,y) = 1/D^2$. Then, by the law of total probability, we can write for $P[S]$:

$$P[S] = \iint_{(x,y)\in\mathcal{D}} P[S|\mathbf{P}_{k,l} = (x,y)] f_{\mathbf{P}_{k,l}}(x,y)\,dx\,dy. \quad (9)$$

Observe now that, for many points $(x,y) \in \mathcal{D}$, we will have $P[S|\mathbf{P}_{k,l} = (x,y)] = 0$. For example, $P[S|\mathbf{P}_{k,l} = (x,y)] = 0$ for all points (x,y) that happen to be located far enough from $(x_A,y_A) = (0,0)$, that is, points for which $\text{dist}\{(x,y),(0,0)\} > R_s + d/2$, where $\text{dist}\{(x,y),(0,0)\}$ is the euclidian distance between points (x,y) and $(0,0)$ (see Fig. 15a). Likewise, for $d/2 < R_s$, if

$$\text{dist}((x,y),(0,0)) < R_s - d/2,$$

then $P[S|\mathbf{P}_{k,l} = (x,y)] = 0$ as well (see Fig. 15b). In this case, since $R_j \ge R_t + R_s$, neither node k nor node l can reach any regular node that is located outside of the jamming region. Therefore, using the polar coordinates $(x,y) = (r\cos\theta, r\sin\theta)$, where $r = \text{dist}\{(x,y),(0,0)\}$, (9) can be rewritten as follows:

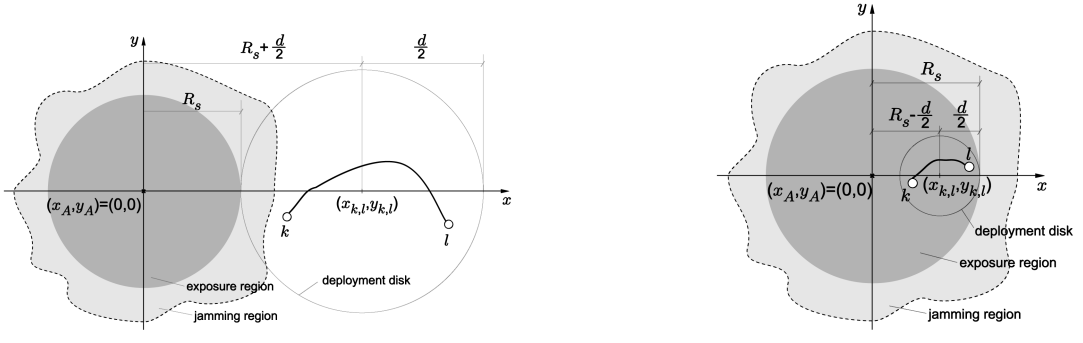
$$P[S] = \frac{1}{D^2} \iint_{\substack{r\in[\underline{r},R_s+\frac{d}{2}]\\ \theta\in[0,2\pi]}} P[S|\mathbf{P}_{k,l} = (r\cos\theta, r\sin\theta)]\,r\,dr\,d\theta, \quad (10)$$

where $\underline{r} = R_s - \frac{d}{2}$ if $\frac{d}{2} \le R_s$ and $\underline{r} = 0$ if $\frac{d}{2} \ge R_s$. For notational simplicity, we will use $P[S|\mathbf{P}_{k,l} = (r,\theta)]$ as the shorthand for $P[S|\mathbf{P}_{k,l} = (r\cos\theta, r\sin\theta)]$.

We next calculate $P[S|\mathbf{P}_{k,l} = (r,\theta)]$ to be able to calculate $P[S]$ from (10). For this, we need some additional notation. We first define the following event:

$$W_1 \equiv \{\text{one node of the connected pair } (k,l) \text{ is located within the exposure region and the other is outside of the exposure region}\}.$$

For example, for connected pair $(k,l) = (1,2)$ in Fig. 14, event W_1 has occurred. Furthermore, we define the following event:



(a) (b)

Fig. 15. Examples where connected pair (k,l) cannot create a wormhole (note that only a part of the jamming region is shown). (a) An example where connected pair (k,l) cannot create a wormhole with $R_s < d/2$. (b) An example where connected pair (k,l) cannot create a wormhole with $R_s > d/2$.

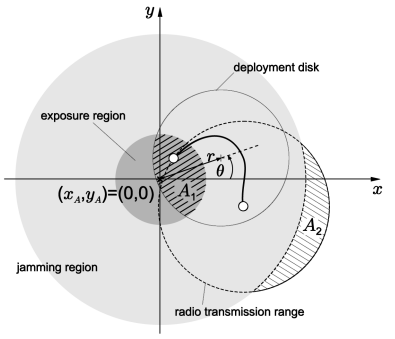


Fig. 16. Definition of regions $A_1(r,\theta)$ and A_2.

$W_2 \equiv \{\text{for the connected pair } (k,l),\ \text{there exists at least one regular node that is located outside of the jamming region but within the transmission range } R_t \text{ of either } k \text{ or } l\}$.

For example, for connected pair $(k,l)=(1,2)$ in Fig. 14, event W_2 has occurred, since node 2 has regular node 3 that is located within node 2's radio transmission range and outside of the jamming range. It is easily seen that, given $R_j \geq R_t + R_s$, event S happens if and only if both event W_1 and event W_2 happen, i.e., $S \equiv W_1 \wedge W_2$. From this, we have the following:

$$\begin{aligned} P[S|\mathbf{P}_{k,l}=(r,\theta)] &= P[W_1,W_2|\mathbf{P}_{k,l}=(r,\theta)] \\ &= P[W_1|\mathbf{P}_{k,l}=(r,\theta)]P[W_2|W_1,\mathbf{P}_{k,l}=(r,\theta)]. \end{aligned} \tag{11}$$

Since the positions of peer nodes k and l are chosen randomly and independently in the corresponding deployment disk (of radius $d/2$) centered at $(x,y)=(r\cos\theta, r\sin\theta)$, we have:

$$P[W_1|\mathbf{P}_{k,l}=(r,\theta)] = 2\times\frac{|A_1(r,\theta)|}{(d/2)^2\pi}\times\frac{(d/2)^2\pi-|A_1(r,\theta)|}{(d/2)^2\pi}, \tag{12}$$

where $A_1(r,\theta)$ is the set of points $(x,y)\in\mathcal{D}$ that are located in the *intersection region* obtained as the intersection between the deployment disk (of the pair (k,l)) centered at $(x,y)=(r\cos\theta, r\sin\theta)$ and the exposure region (see Fig. 16), and $|A_1(r,\theta)|$ denotes the area (not the set size) of this intersection region.

From Fig. 16, we can observe that $|A_1(r,\theta)|=|A_1(r)|$, i.e., the area $|A_1(r,\theta)|$ does not depend on θ; note that this is the consequence of setting $(x_A,y_A)=(0,0)$ and our assumption that jamming and exposure regions are contained completely within the deployment area.[3] The value of $|A_1(r)|$ can be computed by the well-known formula for the area of circle-to-circle intersection.

Next, we evaluate the conditional probability

$$P[W_2|W_1,\mathbf{P}_{k,l}=(r,\theta)].$$

Since event W_1 has happened, it means that one node from the observed pair (k,l) resides in the exposure region (say node k) and the other one (node l) is located outside of the exposure region. But, this implies that node k has no neighbors among regular nodes that are located outside of the jamming region. Then, the event W_2 conditioned on W_1 (which we denote with $\tilde{W}_2$) actually reads:

$\tilde{W}_2 \equiv \{\text{node } l \text{ has at least one neighboring regular node that is located outside of the jamming region}\}$.

Therefore,

$$P[W_2|W_1,\mathbf{P}_{k,l}=(r,\theta)] = P[\tilde{W}_2|\mathbf{P}_{k,l}=(r,\theta)]. \tag{13}$$

Let us denote with $Disk_{k,l}(r,\theta)$ the set of all the points from the pair (k,l)'s deployment disk, centered at $(x,y)=(r\cos\theta, r\sin\theta)$ (see Fig. 16). Then, by the law of total probability, we have:

$$\begin{aligned} P[\tilde{W}_2|\mathbf{P}_{k,l}=(r,\theta)] = &\iint_{(x,y)\in\bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l=(x,y)] \\ &\times f_{\mathbf{P}_l}(x,y)dxdy, \end{aligned} \tag{14}$$

where $\bar{A}_1(r,\theta)=Disk_{k,l}(r,\theta)-A_1(r,\theta)$, $\mathbf{P}_l$ is the two-dimensional random variable describing the location of node l, and $f_{\mathbf{P}_l}(x,y)$ is the probability density function of the location of node l, that is,

$$f_{\mathbf{P}_l}(x,y)=\frac{1}{|\bar{A}_1(r,\theta)|}=\frac{1}{(d/2)^2\pi-|A_1(r)|}\stackrel{def}{=} f_{\mathbf{P}_l}(r). \tag{15}$$

Recall that $|A_1(r,\theta)|=|A_1(r)|$ (see Fig. 16).

Since the regular nodes are deployed uniformly at random in $\mathcal{D}$, we have, for $(x,y)\in\bar{A}_1(r,\theta)$,

$$P[\tilde{W}_2|\mathbf{P}_l=(x,y)] = 1-\left(1-\frac{|A_2(x,y)|}{D^2}\right)^n \approx 1-e^{-n|A_2(x,y)|/D^2}, \tag{16}$$

where $A_2(x,y)$ is the set of points from the node l's transmission region, which does not fall in the jamming region (see Fig. 16), $|A_2(x,y)|$ is the area of this region, and n is the number of regular nodes deployed. Note that the approximation in (16) is valid for large n and $|A_2(x,y)|\ll D^2$.

Now, by combining (11)–(16), we can calculate $P[S|\mathbf{P}_{k,l}=(r,\theta)]$ as follows:

$$\begin{aligned} &P[S|\mathbf{P}_{k,l}=(r,\theta)] \\ &\stackrel{(1)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]P[W_2|W_1,\mathbf{P}_{k,l}=(r,\theta)] \\ &\stackrel{(2)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]P[\tilde{W}_2|\mathbf{P}_{k,l}=(r,\theta)] \\ &\stackrel{(3)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]\iint_{(x,y)\in\bar{A}_1(r,\theta)} \\ &\qquad P[\tilde{W}_2|\mathbf{P}_l=(x,y)]f_{\mathbf{P}_l}(x,y)dxdy \\ &\stackrel{(4)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]f_{\mathbf{P}_l}(r)\iint_{(x,y)\in\bar{A}_1(r,\theta)} \\ &\qquad P[\tilde{W}_2|\mathbf{P}_l=(x,y)]dxdy \\ &\stackrel{(5)}{=} 2\times\frac{|A_1(r)|}{(d/2)^2\pi}\times\frac{(d/2)^2\pi-|A_1(r)|}{(d/2)^2\pi}\times\frac{1}{(d/2)^2\pi-|A_1(r)|} \\ &\qquad\times\iint_{(x,y)\in\bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l=(x,y)]dxdy \\ &\stackrel{(6)}{\approx}\frac{32|A_1(r)|}{(d^2\pi)^2}\iint_{(x,y)\in\bar{A}_1(r,\theta)}\left(1-e^{-\frac{n|A_2(x,y)|}{D^2}}\right)dxdy, \end{aligned} \tag{17}$$

3. By relaxing this assumption, intersection areas A_1 take more complex forms, which significantly increases the complexity of their evaluation.

where (1) follows from the (11), (2) follows from (13), (3) follows from (14), (4) follows from the fact that, for fixed r, the probability density function $f_{\mathbf{P}_i}(r)$ is a constant (see the (15)), (5) follows from (12) and (15) and the fact that the area $|A_1(r)|$ is independent of θ, and, finally, (6) follows from the approximation in (16).

Finally, by plugging (17) into (10), we obtain

$$P[S] \approx \frac{64}{D^2 d^4 \pi} \int_{r \in [\underline{r}, R_s + \frac{d}{2}]} \left\{ \iint_{(x,y) \in \bar{A}_1(r)} \left(1 - e^{-\frac{n|A_2(x,y)|}{D^2}}\right) dxdy \right\} |A_1(r)| r dr, \tag{18}$$

where we used the fact that $|A_2(x,y)|$ (and, therefore, $\{1 - \exp(-n|A_2(x,y)|/D^2)\}$) is independent of θ (see Fig. 16).

Due to the complex expressions for areas $|A_1(r)|$ and $|A_2(x,y)|$, analytically integrating the resulting expression for $P[S]$ is very hard. For this reason, in Section 3.3, we solve (18) numerically and validate it by simulations.

ACKNOWLEDGMENTS

The work presented in this paper was partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

REFERENCES

[1] *BTnodes,* http://www.btnode.ethz.ch, 2006.
[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Magazine,* vol. 40, no. 8, 2002.
[3] A.D. Wood and J.A. Stankovic, "Denial of Service in Sensor Networks," *Computer,* vol. 35, no. 10, pp. 54-62, Oct. 2002.
[4] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks," *Proc. ACM Conf. Computer and Comm. Security,* 2002.
[5] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," *Proc. INFOCOM,* 2003.
[6] W. Kaiser, G. Pottie, M. Srivastava, G.S. Sukhatme, J. Villasenor, and D. Estrin, "Networked Infomechanical Systems (NIMS) for Ambient Intelligence," *Ambient Intelligence,* 2004.
[7] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Elsevier's AdHoc Networks J.,* special issue on sensor network applications and protocols, vol. 1, nos. 2-3, pp. 293-315, Sept. 2003.
[8] M. Leopold, M.B. Dydensborg, and P. Bonnet, "Bluetooth and Sensor Networks: A Reality Check," *Proc. First ACM Conf. Networked Sensor Systems (SenSys),* 2003.
[9] J. McCune, E. Shi, A. Perrig, and M.K. Reiter, "Detection of Denial-of-Message Attacks on Sensor Network Broadcasts," *Proc. IEEE Symp. Security and Privacy,* May 2005.
[10] G. Noubir and G. Lin, "Low_Power DoS Attacks in Data Wireless LANs and Countermeasures," *SIGMOBILE Mobile Computing Comm. Rev.,* vol. 7, no. 3, pp. 29-30, 2003.
[11] G. Sharma and R.R. Mazumdar, "Hybrid Sensor Networks: A Small World," *Proc. MobiHoc '05,* 2005.
[12] H. Solomon, *Geometric Probability.* SIAM, 1978.
[13] A.D. Wood, J.A. Stankovic, and S.H. Son, "JAM: A Jammed_Area Mapping Service for Sensor Networks," *Proc. Real-Time Systems Symp. (RTSS),* 2003.
[14] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," *Proc. MobiHoc '05,* pp. 46-57, 2005.
[15] W. Xu, T. Wood, and Y.W. Trappe, "Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service," *Proc. ACM Workshop Wireless Security (WiSe),* 2004.

Mario Čagalj received the DiplIng degree in computer science and electrical engineering from the University of Split, Croatia, in 1998, and the PhD degree in communication systems from the Ecole Polytechnique Fédérale de Lausanne (EPFL) in February 2006. In 2000 and 2001, he completed the Predoctoral School in Communication Systems, EPFL. From 2001 to 2006, he was a research assistant in the Laboratory for Computer Communications and Applications (LCA) at EPFL. In September 2006, he joined the faculty of electrical engineering, mechanical engineering, and naval architecture (FESB) at the University of Split, Croatia, as an assistant professor. His research interests include the design and analysis of security protocols for wireless networks, applications of game theory to wireless (and wired) networks, and the design of energy-efficient communication protocols for wireless networks. More details can be found at http://www.fesb.hr/~mcagalj.

Srdjan Čapkun received the DiplIng degree in electrical engineering/computer science from the University of Split, Croatia, in 1998. From October 1999 until July 2000, he completed the Doctoral School in Communication Systems at EPFL (Swiss Federal Institute of Technology—Lausanne). In 2004, he received the PhD degree in communication systems from EPFL. From October 2000 until August 2005, he was a postdoctoral researcher in the Networked and Embedded Systems Laboratory (NESL), University of California, Los Angeles. From August 2005 until September 2006, he was an assistant professor in the Informatics and Mathematical Modeling Department (IMM), Technical University of Denmark (DTU). Since September 2006, he has been an assistant professor in the Department of Computer Science, ETH Zurich. His research interests include the design and analysis of security and network protocols for wired and wireless networks.

Jean-Pierre Hubaux joined the faculty of EPFL in 1990; he was promoted to full professor in 1996. His research activity is focused on wireless networks, with a special interest in security and cooperation issues. He has been strongly involved in the National Competence Center in Research, named "Mobile Information and Communication Systems" (NCCR/MICS) since its genesis in 1999; this center is often nicknamed "the Terminodes project." In this framework, he has notably defined, in close collaboration with his students, novel schemes for the security and cooperation in multihop networks; in particular, he has devised new techniques for key management, secure positioning, and incentives for cooperation in such networks. He has also made several contributions in the areas of power management in sensor networks and of group communication in ad hoc networks. He has recently written, with Levente Buttyan, a graduate textbook entitled *Security and Cooperation in Wireless Networks*. He is an associate editor of the *IEEE Transactions on Mobile Computing* and *Foundations and Trends in Networking*. He served as the general chair for the Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02) held on the EPFL campus. He has been serving on the program committees of numerous conferences and workshops, including SIGCOMM, INFOCOM, MobiCom, MobiHoc, SenSys, WiSe, and VANET. He has held visiting positions at the IBM T.J. Watson Research Center and at the University of California at Berkeley. After completing his studies in electrical engineering at Politecnico di Milano, he worked 10 years in France with Alcatel, where he was involved in R&D activities, primarily in the area of switching systems architecture and software. For more information, please check http://people.epfl.ch/jean-pierre.hubaux.