Wormhole-Based Anti-Jamming Techniques in Sensor Networks

Mario Čagalj[†*] Srdjan Čapkun[§]

Jean-Pierre Hubaux[†]

[†]Laboratory for Computer Communications and Applications (LCA)

Faculty of Informatics and Communication (I&C)

Swiss Federal Institute of Technology Lausanne (EPFL)

CH-1015 Lausanne, Switzerland

[§]Informatics and Mathematical Modelling Department

Technical University of Denmark (DTU)

DK-2800 Lyngby, Denmark

E-mail: {mario.cagalj@epfl.ch, sca@imm.dtu.dk and jean-pierre.hubaux@epfl.ch}

November 24, 2005

*Corresponding author.

Abstract

Due to their very nature, wireless sensor networks are perhaps the most vulnerable category of wireless networks to "radio channel jamming"-based Denial-of-Service (DoS) attacks. An adversary can mask the events that the sensor network should detect by stealthily jamming an appropriate subset of the nodes; in this way, he prevents them to report what they are sensing to the network operator. Therefore, in spite of the fact that an event is sensed by one or several nodes (and the sensor network is fully connected), the network operator cannot be informed on time. We show how the sensor nodes can exploit channel diversity in order to establish wormholes 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, whereas 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 to report what they are sensing to the network operator. Timely detection of such stealth attacks is particulary important in scenarios in which sensors use reactive schemes to communicate events to the network sink [15].

Event masking attacks result in a *coverage paradox*: in spite of the fact that an event is sensed by one or several nodes (and the sensor network is fully connected), the network operator cannot be informed about the event on time (see Fig. 1). We will explain that the solution to this problem is

far from trivial: proactive schemes, in which sensors spend their time (and battery) assessing the state of their communication links are clearly suboptimal; equally, jamming detection schemes are generally over-sensitive and generate many false alarms making the system vulnerable to straightforward Denial of Service (DoS) attacks.

We show that *wormholes* [6], 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 and motivate 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 anti-jamming technique, based on uncoordinated channel hopping. In this approach, the nodes form low-bandwidth anti-jamming communication channels by randomly hopping between the given set of orthogonal channels; moreover, this solution does not require the nodes to be synchronized.

The organization of the paper is the following. In Section 2, we motivate 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 the related work in Section 6. We conclude in Section 7. Finally, in Appendix, we develop the mathematical model used in this paper.

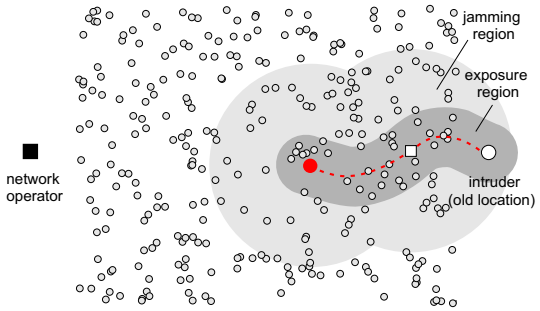


Fig. 1: The *coverage paradox* – in spite of the fact that an intruder is detected by the sensor nodes (and the network is connected), the network operator cannot be informed on time: The intruder moves in the network and gets detected by the nodes located in the *exposure region*; The intruder then stealthily jamms all communication within the *jamming region* (the white square represents a jamming device left behind by the intruder on his way). To avoid detection of jamming by the nodes that do not sense its presence, the intruder can employ a "stealth" jamming strategy.

2 Motivation and existing tradeoffs

Our work is motivated by 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 several reasons: faulty or compromised sensors, unreliable or disrupted communication links. In this work, we focus on the latter ones.

In a wireless sensor network, all mutual communication between sensors and between sensors and the network operator is wireless (and multi-hop) [3]. 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 within 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 to the operator (on time). 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", since the sensor nodes from the exposure region may eventually detect the jamming activity of the adversary. However, this is not so easy task considering the computational capabilities of sensor nodes [15]. 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 [11].

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.

2.1 Proactive vs. 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 lower energy consumption and therefore longer network lifetimes.

In the case of proactive sensor networks, several simple solutions can be proposed to ensure that the operator receives event reports or detects jamming. One solution consists in 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 re-transmission 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; if they are sent

infrequently, the batteries will last longer, but the time elapsed between an event happened 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 need 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, because 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 the 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 the *probabilistic wormhole*. An example of a probabilistic

wormhole, realized through wires, is shown on Fig. 2(a).

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

3 Wormholes via wired pairs of sensor nodes

In this solution, we propose to augment 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 on Fig. 2(a): isolated points represent regular nodes and connected pairs are denoted as connected points. A similar form of a hybrid sensor network already appears in the context of the NIMS project [7], and in the work by Sharma and Mazumdar [12].

3.1 Rationale of wired pairs

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 on Fig. 2(a). In this scenario, the attacker (A), represented by sign 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 on Fig. 2(a) and Fig. 2(b), we model the exposure region by a circle

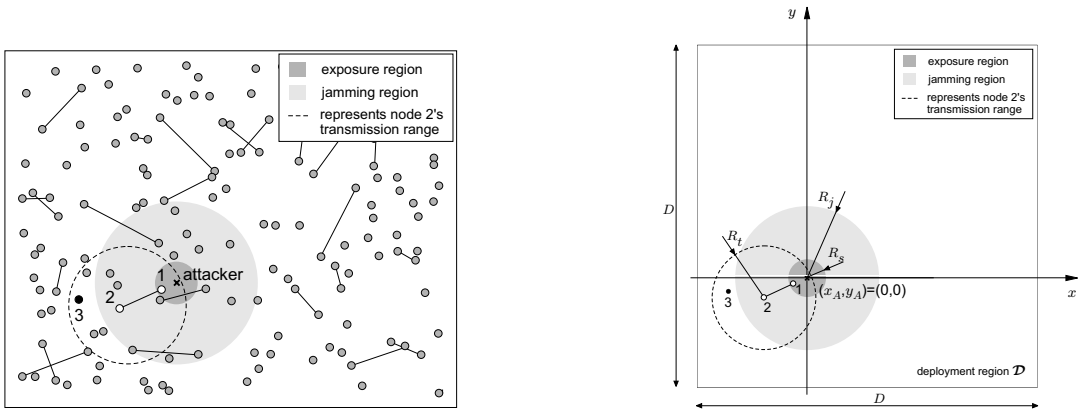


Fig. 2: Probabilistic wormholes via wired pairs of sensor nodes: (a) Hybrid sensor network with randomly deployed sensor nodes: isolated points are regular nodes, connected points represent sensor nodes connected through a wire.; An attacker who jams surrounding nodes. Connected pair $(1, 2)$ and regular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

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 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, to successfully leave 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 on Fig. 2(a) and Fig. 2(b), 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. Since 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 on Fig. 2(a)

and Fig. 2(b)) 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 the 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 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. 2(b)). The deployment region $\mathcal{D}$ is modelled by 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. 2(b)). 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 any loss of generality, we set $(x_A, y_A) = (0, 0)$ (Fig. 2(b)). 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

[1]This assumption is more legitimate in the context of the solution based on frequency-hoping 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.

of the locations of the connected pairs.

For the given attacker, located at point $(x_A, y_A) = (0, 0)$, we calculate $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$, *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 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 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 $P\big[\text{at least one wormhole}|(x_A, y_A)\big] \geq p_w$, where p_w is a targeted probability. Let K_0 denote the critical (minimum) number of connected pairs for which $P\big[\text{at least one wormhole}|(x_A, y_A)\big] = p_w$ holds. Then, from (1) we have the following result.

Theorem 1

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

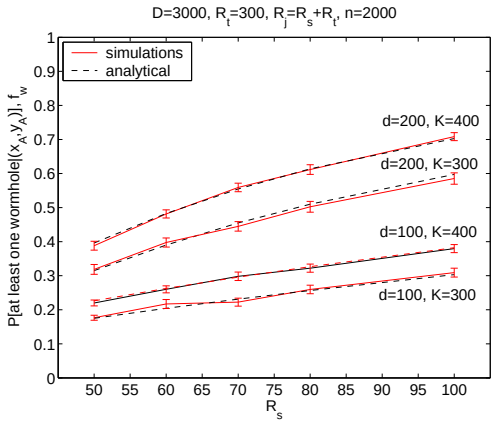
where p_s is given by the expression (16) in Appendix.

The result from Theorem 1 is common in stochastic geometry.

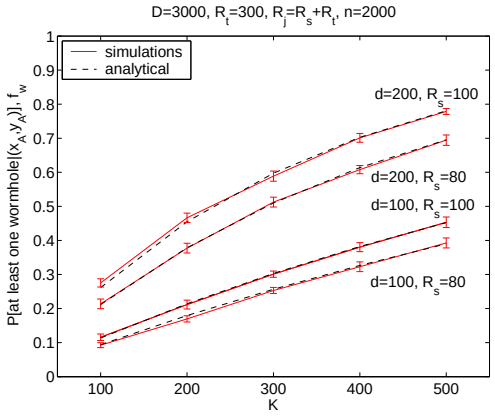
3.3 Simulations and model validation

We investigated the proposed analytical model (see Appendix) by means of simulations. We evaluated probability $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ 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 generate randomly the network topology with n regular nodes and K connected pairs (see Fig. 2(a)). Next, we throw randomly $N = 500$ 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 95% confidence interval.

The results are shown on Fig. 3 and Fig. 4, together with numerical results obtained from the analytical model developed in the previous section (and Appendix). As we can see from the figures, the analytical model predicts quite accurately $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$. Other interesting conclusions can be drawn from the figures. We can see that the increase in either R_s and K results in nearly linear increase in $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$. 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 for 20 units (from 80 to 100), for $K = 300$ and $d = 200$, results in the increase of $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ of around 0.1 (Fig. 3(a)). However, an increase of K for 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. 3(b)). 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 $P_w \equiv P\big[\text{at least one wormhole}|(x_A, y_A)\big]$

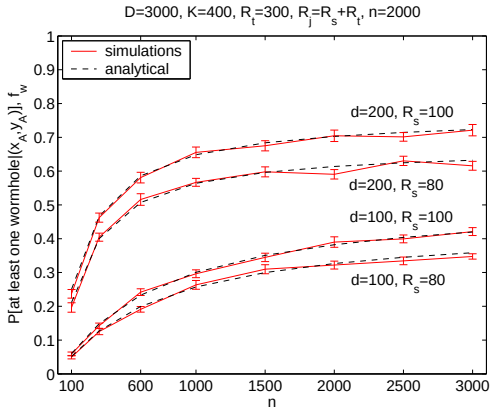


(a)

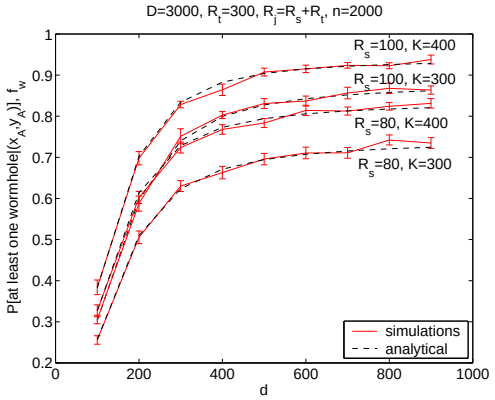


(b)

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



(a)



(b)

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

with respect to p_s and K. From expression (1) we have

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

Since p_s increases in R_s, it follows readily that it is more advantageous to increase R_s than K. From Fig. 3(a) and Fig. 3(b) 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 Fig. 4(a) and Fig. 4(b) we observe that increasing n and d is beneficial only until a certain saturation point; this can easily be deduced from our model developed in 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 $d/2$ [13]).

The results from this section show that while 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 major drawback that it requires the wires to be deployed in the field. Moreover, as we saw in Section 3.3, in order to achieve a reasonably high $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$, 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 that the pairs are formed exclusively through wireless links resistant to jamming. By using a wireless link, not only do we avoid cumbersome wires, we can also afford longer links between pairs; as we saw in Section 3.3 (Fig. 4(b)), the increase in d (maximum length of a wire) has a profound impact on $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$.

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 having FH capabilities [1]. It is important to stress, however, that we do not propose to equip all

the nodes in the network with FH radio (a case study of Bluetooth sensor networks can be found in [9]). The reason is that FH radio imposes a substantial overhead on sensor nodes in multihop networks [9]; the need for "synchronization" (at multiple levels) between senders and designated receivers (synchronization of hopping sequences, time synchronization) might be a major deterrent to using FH radios in multihop wireless sensor networks [9].

Instead, we propose to deploy 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 nodes agree to form a 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 be 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 FH pairs form at least one wormhole, with a high probability, in the event of a jamming attack (see Fig. 2(a)). 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 a 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, however, 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),

[2]An alternative would be to use a similar approach as in the probabilistic key pre-distribution schemes [5], where the nodes would be pre-loaded with a certain number of FH sequences chosen randomly from a common pool.

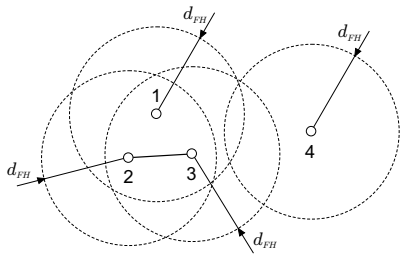


Fig. 5: Opportunistic FH pairing process: the thick line connecting FH nodes 2 and 3 means that they form a FH pair, while FH nodes 1 and 4 remain "unpaired" (d_{FH} is the radio transmission range of the FH nodes).

it is very likely that some FH nodes will not find any "free" FH neighbor.

Consider the example on 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 a 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 analysis of the FH pairs based solution far more difficult. We will now show how to effectively overcome this difficulty.

4.2 Analysis of the FH pairs based solution

Again, our goal is to estimate $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ - 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 throw randomly a certain number of FH enabled nodes in a deployment region of size $D \times D$ with $D = 3000$. Then we combine FH nodes randomly into FH pairs, with the restriction that a single FH node can be a

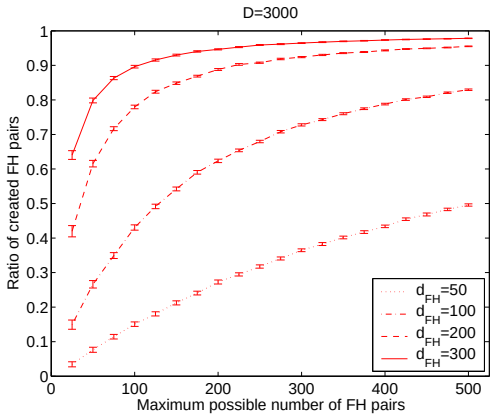


Fig. 6: Ratio of created FH pairs vs. maximum possible number of FH pairs (= the number of FH enabled nodes deployed $\times$ 2); we use 95% confidence interval.

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% confidence intervals, is presented on 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 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 a FH pair (the normalized average distance of a FH link) and the corresponding standard deviation. On 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 on this figure is that the normalized average distance of a FH link is approximately $0.66 \approx \frac{2}{3}$, irrespectively of d_{FH}. Moreover, the standard deviation is approximately 0.23.

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

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

This results suggests that, the random process of opportunistic FH pairing exhibits similar behavior as the process of picking a random point from the circle of radius d_{FH} centered at the given FH node. To confirm this hypothesis, we have 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 probability of a distance between FH peers falling into the same intervals; we use pdf given in (3) to calculate this probability.

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

This matching inspires the following approach to modelling the creation of a random FH pair in the opportunistic pairing protocol. Consider a 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 picking 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 picking model. Then, from the independence between

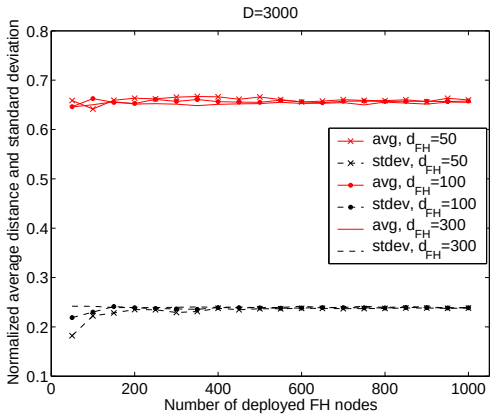
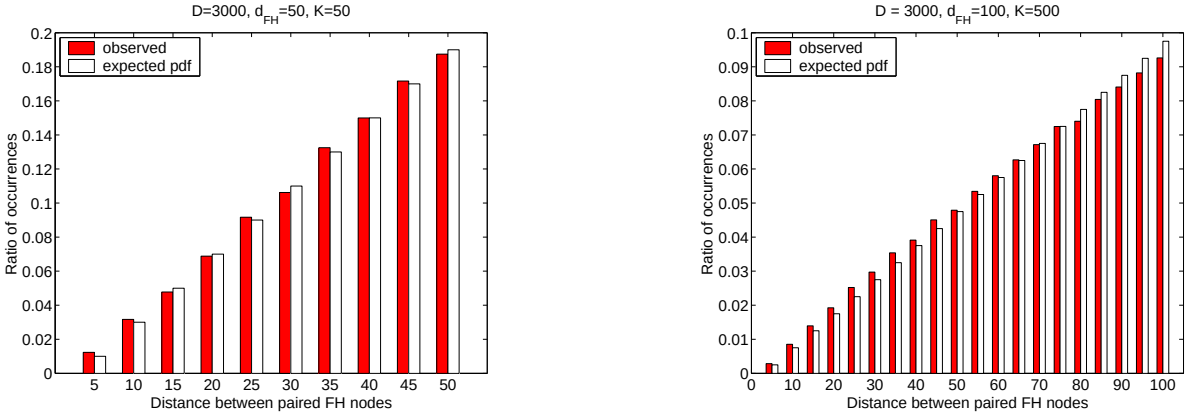


Fig. 7: Normalized average distance between FH peers vs. the number of FH enabled nodes deployed ("avg" - average, "stdev" - standard deviation).



(a) (b)

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 experiments=3500; (b) $d_{FH} = 100,\ K = 500$, number of experiments=10000.

different created FH pairs, $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ can be calculated as follows:

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

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 rather re-use the model developed for wired sensor pairs (Section 3.2 and Appendix) by exploiting the similarity between the solution based on

wired pairs and the solution based on FH pairs.

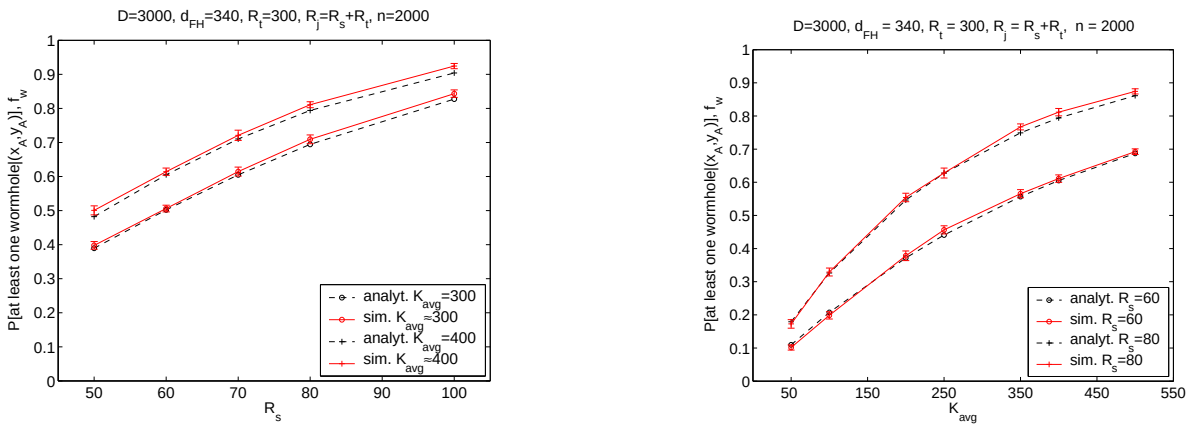
Note first that there is a subtle difference in the way we model the deployment of pairs connected through wires and the way we model the creation of FH pairs. In the first case, we use so called "disk line picking" model, i.e., two points are selected randomly and independently from the disk of radius $\frac{d}{2}$ (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 $\frac{128}{45\pi}\frac{d}{2}$ [13]. In the second case, one point (FH node i) is given and its FH peer is modelled 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 modelling 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 $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ for the solution based on FH pairs, we first scale d using expression (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 expression (1). In the following section, we evaluate the proposed model.

4.3 Simulations and model validation

We investigated the proposed analytical model by means of simulations. We evaluated probability $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ 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



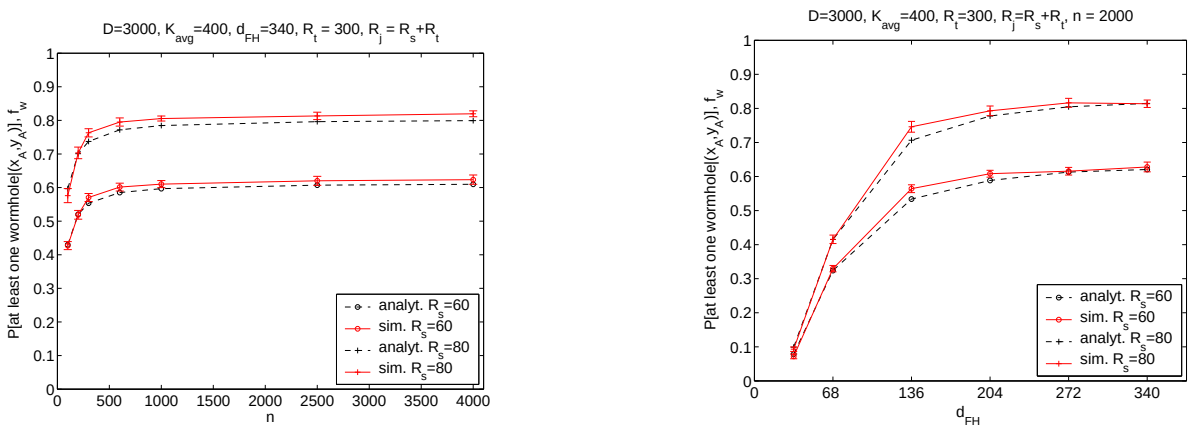
(a) (b)

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

value of a given parameter, we first generate randomly the network topology with n regular nodes and K_{FH} FH nodes. To simulate the FH pairing protocol, we iterate randomly 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 throw randomly $N = 500$ 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 95% confidence interval. To obtain the numerical results, for each value of d_{FH}, we first scale d using expression (5) and then we plug resulting d in expression (1) to obtain $P\big[$at least one wormhole$|(x_A, y_A)\big]$. The values of K are obtained as the average number of created FH pairs for different number of FH nodes K_{FH} (see Fig. 6).

The results are shown on Fig. 9-10, 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



(a) (b)

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

from the figures, the analytical model predicts quite accurately $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$. The results obtained have identical properties as in the solution based on pairs connected through wires. The important difference between wired pairs and FH pairs is that the later achieve the same $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$ with transmission ranges d_{FH} smaller than the maximum wire length d; i.e., $d_{FH}/d \approx 0.6791$ (expression (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 FH nodes that make a pair. In this section, we 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 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).

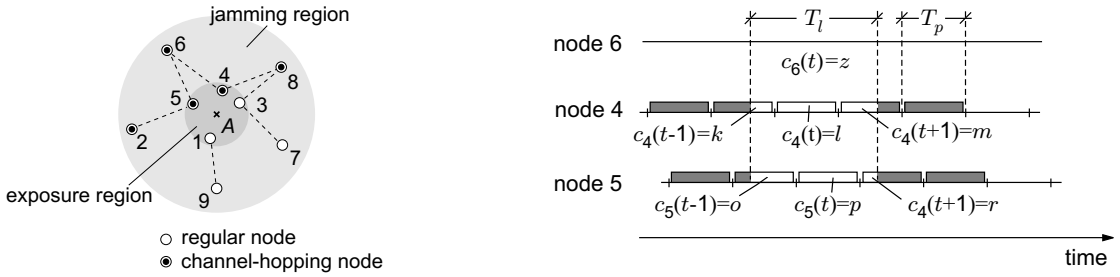


Fig. 11: (a) A network example with channel-hopping nodes; (b) Example of scheduling for nodes 4, 5 and 6, with $T_l = 2T_p$ (T_l is the listening period, T_p is a packet "length", $c_4(t) = l$ denotes that node 4 transmits a packet on channel l at time t, and $c_6(t) = z$ denotes that node 6 listens on channel z at time t).

5.1 Rationale of the approach

In this approach, we can image a part of the deployed nodes or all of them to 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 on 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 a report should be acknowledged by 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 has to 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. Another subtlety of the channel hopping approach is that listening CH nodes enter the channel hopping mode only occasionally (at some predefined rate); 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 between 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 a 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 [11], 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 time (that will depend on the jamming radio), while giving an opportunity to transmissions on the other channels to successfully finish. We perform performance analysis of this approach in Section 5.4.

The implementation of channel-hopping strategies is easily achieved with sensor nodes that use highly programmable software radios (e.g., MICA motes [2]).

5.2 System model and assumptions

We consider a scenario in which a single attacker is restricted to jam only one channel at a time. This basic model is sufficient for the understanding of the case when the attacker is capable of jamming on several channels at a time; we leave this task for future work.

We next introduce some notation. 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; on Fig. 11(a), $I = \{4, 5\}$. 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; on Fig. 11(a), $O = \{2, 6, 8\}$. Also, let I_i be the set of channel-hopping neighbors from I of node $i \in O$; on Fig. 11(a), $I_2 = \{5\}$, $I_6 = \{4, 5\}$ and $I_8 = \{4\}$.

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 assume that the nodes from the set I always transmit, while 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. 11(b)). In our analysis we set $T_l = 2T_p$.

We characterize the strength of the attacker by time periods T_s and T_j, where T_s is the time it takes to switch between two channels (and possibly to scan a given channel to detect some activity), and T_j is the minimum jamming period that the attacker has to jam a given transmission in order to destroy the corresponding packet. We further 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 are the packets to jamming. Note that our game makes sense only if the jamming ratio is sufficiently high. In [11], Noubir and Lin present a set of different coding strategies (based on *low-density parity-check* (LDPC) codes) that can achieve $\rho_j = 10 - 15\%$.

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. The attacker can potentially learn (by scanning the available channels) that there is some activity on the channels occupied by transmitters. In this way, he can avoid loosing time on jamming currently unused channels.

Consider the scenario shown on Fig. 12, where nodes i, j and k are transmitting packets on 5 orthogonal radio channels ($\{1, 2, 3, 4, 5\}$) to two listening nodes A and B. Since the attacker has no knowledge about nodes A and B (i.e., the channels they use, the level of de-synchronization, their location, etc.), a reasonable attacking strategy is to jam sequentially only active channels, in such a way that channels that have not been visited for the longest time are given advantage compared to other channels. In the example on Fig. 12, the attacker jams the channels in the following order: $(3, 2, 5, 2, 1, 5, 1, 3, 4, 3)$. Here we assumed that the attacker knows somehow which channels are to be active; in practice, this involves scanning the channels (which could potentially incur some additional time cost to the attacker).

During a period of duration T_p, the attacker can visit and jam successfully at most $\frac{T_p}{T_j+T_s}$ channels. Clearly, the following has to be satisfied for the channel-hopping approach to make sense:

$$m > \frac{T_p}{T_j + T_s} \approx \frac{1}{\rho_j}, \quad \text{for } \frac{T_s}{T_p} \ll 1 \ .$$

Otherwise, the attacker can always visit and successfully jam all the active channels within the packet period T_p.

Let $\overline{n}$ denote the expected number of the hopping channels that get occupied by transmissions of nodes from the set I (the set of all the transmitters residing in the exposure region). We observe that $\overline{n}$ corresponds to the expected number of occupied bins out of total of m, given that we throw uniformly $|I|$ balls. Then, $\overline{n}$ satisfies the following (for m large "enough"): $\overline{n} \approx m\left(1 - e^{-|I|/m}\right)$. We note that it is prudent to ensure $\overline{n} > \rho_j^{-1}$, since, otherwise, the attacker can typically visit and jam successfully all the occupied channels.

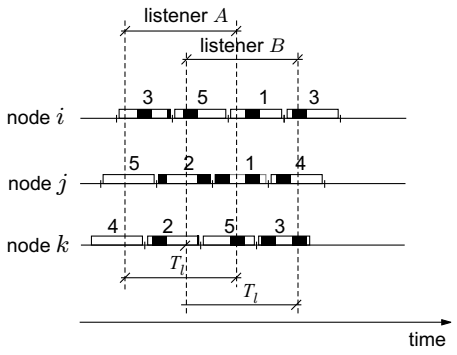


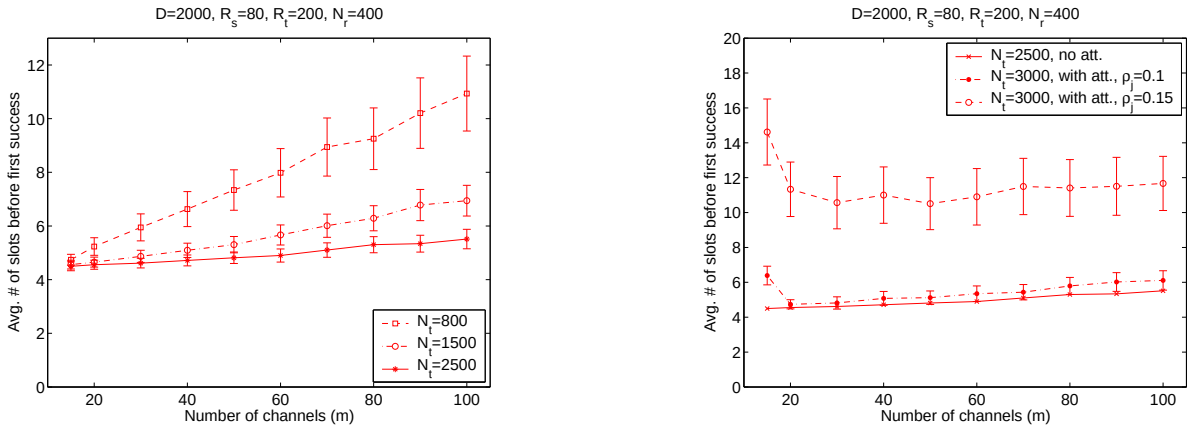
Fig. 12: Example of optimal jamming strategy (the black part of a packet represents the part being jammed).

5.4 Performance analysis

We carried out an evaluation of this approach using simulations written in Matlab. For the 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, the attacker follows the strategy described in the previous section; i.e., every T_{jam} period, the attacker picks 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 $D = 2000$. 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 pick randomly 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 a 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 de-synchronization 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.



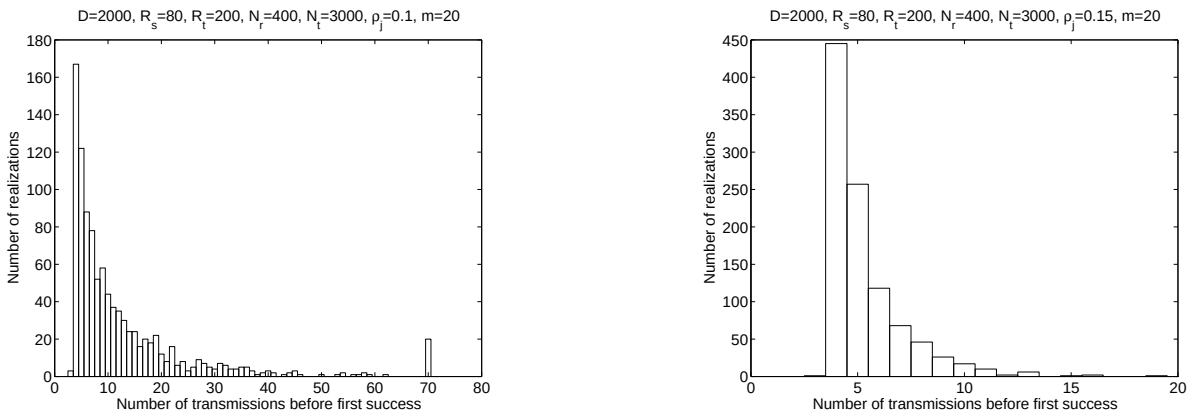
(a) (b)

Fig. 13: 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 95% confidence intervals.

We repeat our experiments for different number m of hopping channels. For each fixed channel number, we average the results across 20×50 experiments described above.

The results are presented on Fig. 13(a) and Fig. 13(b), with 95% confidence interval. On Fig. 13(a), 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 decreases 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). Finally, the value of $\overline{N}_{succ}$ is reasonably small, so that we can speak of *timely data delivery* in the approach based on uncoordinated channel-hopping approach. For example, with the communication speed of 19.2 Kbps, the packet size of 20 bytes (including the preamble) and with negligible inter-packet delay, $\overline{N}_{succ} = 10$ corresponds to approximately 85 ms.

Next we observe $\overline{N}_{succ}$ in scenarios with an active attacker. The results for $\rho_j = \{0.1, 0.15\}$ are shown on Fig. 13(b). Note that $\rho_j = 0.1$ and $\rho_j = 0.15$ imply that the attacker can jam successfully at most $1/0.1 = 10$ and $1/0.15 \approx 7$ packets during time period T_p. In this figure, the curve obtained



(a) (b)

Fig. 14: Distribution of the "number of transmissions before the first success" for $m = 20$ and: (a) $\rho_j = 0.1$, (b) $\rho_j = 0.15$. The number of samples is 1000.

for $N_t = 2500$ 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 reasonably small value. For example, for $N_t = 3000$ and $\rho_j = 0.1$, $\overline{N}_{succ}\big|_{m=15} = 15$ and $\overline{N}_{succ}\big|_{m\geq 20} \approx 11$. 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 $\overline{N}_{succ}$.

Another important observation is that the uncoordinated channel-hopping approach is feasible for modest values of ρ_j and m, which directly impacts implementation costs of this approach. This is better seen on Fig. 14(a) and Fig. 14(b), where we plot histogram (distribution) of the number of transmissions before the first success (N_{succ}) for $m = 20$. On Fig. 14(a), 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. We can further observe that variance of the N_{succ} is much higher in the case $\rho_j = 0.1$ compared to $\rho_j = 0.15$, which is consistent with the plots on Fig. 13(b). Finally, we can see that the frequency of N_{succ} quickly decreases as N_{succ} increases, therefore confirming our conclusion that uncoordinated channel-hopping is well suited for real-time intruder detection tasks.

6 Related work

The issues of jamming detection and prevention in wireless sensor networks have received a significant attention recently. In [4], Wood and Stankovic briefly study potential techniques to avoid jammed regions. A more elaborate study was presented by Wood, Stankovic and Son in [14]. In this work, they propose a proactive protocol that first detects and then maps 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, we, however, propose reactive solutions that do not require periodic exchange of information. Xu et. al. [16] 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 that the solution based on spatial retreats is appropriate for sensor networks. In [16], Xu et. al. study the feasibility of reliably detecting jamming attacks. They showed that reliable detection can be a quite challenging task in wireless sensor networks. Moreover, all the proposed detection mechanisms are by their nature proactive. In [11], Noubir and Lin show how to use low density parity check (LDPC) codes to cope with jamming. In [8], Karlof and Wagner introduce a new attack against wireless sensor networks called sinkholes. In [10], McCune et al. propose a scheme for the detection of denial-of-message attacks on sensor network broadcasts.

7 Conclusion

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

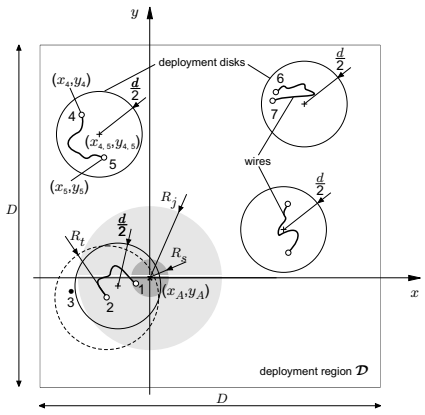


Fig. 15: Approximation model for random deployment of connected pairs (the thick curves connecting the nodes represent wires between the nodes).

It is clear that the space of investigation of this area is huge: other approaches 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 performances of hybrid solutions, by combining the three approaches proposed in this chapter. Finally, it would be interesting to implement the presented schemes.

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

To model the random deployment of connected pairs we proceed as follows. Let us consider connected pair $(4, 5)$ on Fig. 15. We first pick 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. 15). Finally, we pick 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. 15). 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 2-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 $\{(x, y) : (x-x_{i,j})^2+(y-y_{i,j})^2 \leq (d/2)^2$, 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 $P\big[\text{at least one wormhole}|(x_A, y_A)\big]$, we first define the following event:

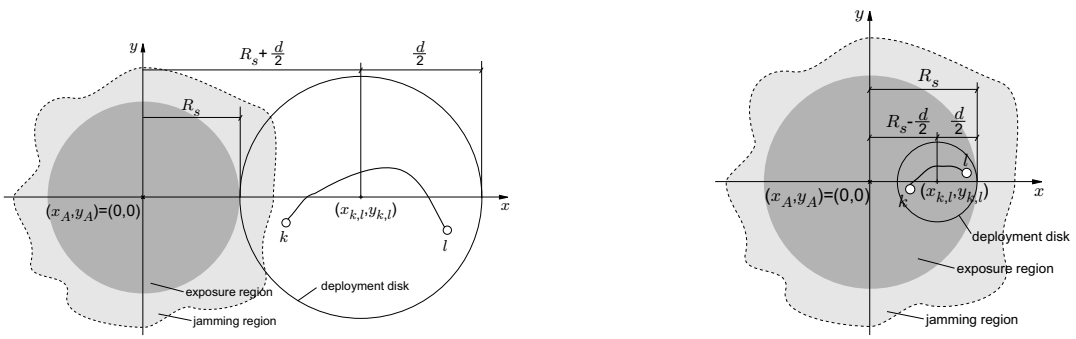
$$S \stackrel{def}{=} \big\{\text{the connected pair } (k, l) \text{ forms a wormhole from the exposure region around } (x_A, y_A) \text{ to the area not affected by jamming}\big\} .$$

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

$$P[S] = \iint_{(x,y)\in\mathcal{D}} P[S|\mathbf{P}_{k,l} = (x, y)] f_{\mathbf{P}_{k,l}}(x, y) dx dy \ . \tag{7}$$



(a) (b)

Fig. 16: 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 $R_s > d/2$.

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 $\text{dist}\big\{(x,y),(0,0)\big\} > R_s + d/2$, where $\text{dist}\big\{(x,y),(0,0)\big\}$ is the Euclidian distance between points (x,y) and $(0,0)$ (see Fig. 16(a)). Likewise, for $d/2 < R_s$, if $\text{dist}\big((x,y),(0,0)\big) < R_s - d/2$, then $P[S|\mathbf{P}_{k,l} = (x,y)] = 0$ as well (see Fig. 16(b)); in this case, since $R_j \geq R_t + R_s$, 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 $r = \text{dist}\big\{(x,y),(0,0)\big\}$, expression (7) can be rewritten as follows

$$P[S] = \frac{1}{D^2} \iint_{\substack{r\in[\underline{r},R_s+\frac{d}{2}]\\ \theta\in[0,2\pi]}} P[S|\mathbf{P}_{k,l} = (r\cos\theta, r\sin\theta)] r dr d\theta \,, \tag{8}$$

where $\underline{r} = R_s - \frac{d}{2}$ if $\frac{d}{2} \leq R_s$ and $\underline{r} = 0$ if $\frac{d}{2} \geq R_s$. For notational simplicity, in the sequel, 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 expression (8). For this we need some additional notation. We first define the following event:

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

For example, for pair $(k,l) = (1,2)$ on Fig. 15, event W_1 has occurred. Furthermore, we define

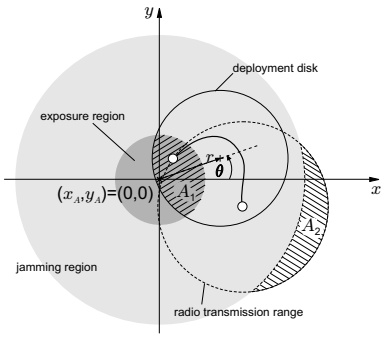


Fig. 17: Definition of regions $A_1(r,\theta)$ and A_2.

the following event:

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

For example, for pair $(k,l) = (1,2)$ on Fig. 15, 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:

$$P[S|\mathbf{P}_{k,l} = (r,\theta)] = P[W_1, W_2|\mathbf{P}_{k,l} = (r,\theta)] = P[W_1|\mathbf{P}_{k,l} = (r,\theta)]P[W_2|W_1, \mathbf{P}_{k,l} = (r,\theta)] \ . \tag{9}$$

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:

$$P[W_1|\mathbf{P}_{k,l} = (r,\theta)] = 2 \times \frac{|A_1(r,\theta)|}{(d/2)^2\pi} \times \frac{(d/2)^2\pi - |A_1(r,\theta)|}{(d/2)^2\pi} \ , \tag{10}$$

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. 17), and $|A_1(r,\theta)|$ denotes the area (not the set size) of this intersection region.

From Fig. 17 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:

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

Therefore,

$$P[W_2|W_1, \mathbf{P}_{k,l} = (r, \theta)] = P[\tilde{W}_2|\mathbf{P}_{k,l} = (r, \theta)] \ . \tag{11}$$

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. 17). Then, by the law of total probability we have:

$$P[\tilde{W}_2|\mathbf{P}_{k,l} = (r, \theta)] = \iint_{(x,y)\in\bar{A}_1(r,\theta)} P[\tilde{W}_2|\mathbf{P}_l = (x, y)] \times f_{\mathbf{P}_l}(x, y)dxdy \ , \tag{12}$$

where $\bar{A}_1(r, \theta) = Disk_{k,l}(r, \theta) - A_1(r, \theta)$, $\mathbf{P}_l$ is the 2-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,

$$f_{\mathbf{P}_l}(x, y) = \frac{1}{\left|\bar{A}_1(r, \theta)\right|} = \frac{1}{(d/2)^2\pi - |A_1(r)|} \stackrel{def}{=} f_{\mathbf{P}_l}(r) \ . \tag{13}$$

Recall, $|A_1(r, \theta)| = |A_1(r)|$ (see Fig. 17).

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

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

$$P[\tilde{W}_2|\mathbf{P}_l=(x,y)]=1-\left(1-\frac{|A_2(x,y)|}{D^2}\right)^n\approx 1-e^{-n|A_2(x,y)|/D^2}\ , \tag{14}$$

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. 17), $|A_2(x,y)|$ is the area of this region, and n is the number of regular nodes deployed. Note that the approximation in expression (14) is valid for large n and $|A_2(x,y)| << D^2$.

Now, by combining expressions (9)-(14), we can calculate $P[S|\mathbf{P}_{k,l}=(r,\theta)]$ as follows

$$\begin{aligned}
P[S|\mathbf{P}_{k,l}=(r,\theta)] &\overset{(1)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]P[W_2|W_1,\mathbf{P}_{k,l}=(r,\theta)]\\
&\overset{(2)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]P[\tilde{W}_2|\mathbf{P}_{k,l}=(r,\theta)]\\
&\overset{(3)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]\iint_{(x,y)\in\bar{A}_1(r,\theta)}P[\tilde{W}_2|\mathbf{P}_l=(x,y)]f_{\mathbf{P}_l}(x,y)dxdy\\
&\overset{(4)}{=} P[W_1|\mathbf{P}_{k,l}=(r,\theta)]f_{\mathbf{P}_l}(r)\iint_{(x,y)\in\bar{A}_1(r,\theta)}P[\tilde{W}_2|\mathbf{P}_l=(x,y)]dxdy\\
&\overset{(5)}{=} 2\times\frac{|A_1(r)|}{(d/2)^2\pi}\times\frac{(d/2)^2\pi-|A_1(r)|}{(d/2)^2\pi}\times\frac{1}{(d/2)^2\pi-|A_1(r)|}\\
&\times\iint_{(x,y)\in\bar{A}_1(r,\theta)}P[\tilde{W}_2|\mathbf{P}_l=(x,y)]dxdy\\
&\overset{(6)}{\approx}\frac{32|A_1(r)|}{(d^2\pi)^2}\iint_{(x,y)\in\bar{A}_1(r,\theta)}\left(1-e^{-\frac{n|A_2(x,y)|}{D^2}}\right)dxdy\ ,
\end{aligned} \tag{15}$$

where (1) follows from the expression (9), (2) follows from the expression (11), (3) follows from (12), (4) follows from the fact that for fixed r the probability density function $f_{\mathbf{P}_l}(r)$ is a constant (see the expression (13)), (5) follows from the expressions (10) and (13) and the fact that the area $|A_1(r)|$ is independent of θ, and finally (6) follows from the approximation in the expression (14). Finally, by plugging the expression (15) in the expression (8) we obtain

$$P[S]\approx\frac{64}{D^2d^4\pi}\int_{r\in[\underline{r},R_s+\frac{d}{2}]}\left\{\iint_{(x,y)\in\bar{A}_1(r)}\left(1-e^{-\frac{n|A_2(x,y)|}{D^2}}\right)dxdy\right\}|A_1(r)|rdr\ , \tag{16}$$

where we used the fact that $|A_2(x,y)|$ (and therefore $\left\{1-\exp(-n|A_2(x,y)|/D^2)\right\}$) is independent

of θ (see Fig. 17).

Due to the complex expressions for areas $|A_1(r)|$ and $|A_2(x, y)|$, integrating analytically the resulting expression for $P[S]$ is very hard. For this reason, in Section 3.3 we solve the expression (16) numerically and validate it by simulations.

References

[1] *BTnodes*. http://www.btnode.ethz.ch/.

[2] Mica sensor platform. http://www.xbow.com.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communication Magazine*, 40(8), 2002.

[4] A. D.Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, 2002.

[5] L. Eschenauer and V. Gligor. A Key Management Scheme for Distributed Sensor Networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2002.

[6] Y. Hu, A. Perrig, and D. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In *Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2003.

[7] W. Kaiser, G. Pottie, M. Srivastava, G.S. Sukhatme, J. Villasenor, and D. Estrin. Networked Infomechanical Systems (NIMS) for Ambient Intelligence. In *Ambient Intelligence, Springer-Verlag*, 2004.

[8] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.

[9] M. Leopold, M.B. Dydensborg, and P. Bonnet. Bluetooth and Sensor Networks: A Reality Check. In *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys)*.

[10] J. McCune, E. Shi, A. Perrig, and M.K. Reiter. Detection of denial-of-message attacks on sensor network broadcasts. In *Proceedings of IEEE Symposium on Security and Privacy*, May 2005.

[11] G. Noubir and G. Lin. Low-power DoS attacks in data wireless LANs and countermeasures. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):29–30, 2003.

[12] G. Sharma and R.R. Mazumdar. Hybrid Sensor Networks: A Small World. In *Proceedings of MobiHoc'05*, Urbana-Champaign, Illinois, USA.

[13] Herbert Solomon. *Geometric Probability*. SIAM, 1978.

[14] A.D. Wood, J.A., Stankovic, and S.H. Son. JAM: A Jammed-Area Mapping Service for Sensor Networks. In *Real-Time Systems Symposium (RTSS)*, Cancun, Mexico, 2003.

[15] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of MobiHoc'05*, Urbana-Champaign, Illinois, USA.

[16] W. Xu, T. Wood, W.Trappe, and Y. Zhang. Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2004.