Computer Communications 86 (2016) 75–85

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Jammer localization in wireless networks: An experimentation-driven approach[☆]

Konstantinos Pelechrinis[a,*], Iordanis Koutsopoulos[b], Ioannis Broustis[c], Srikanth V. Krishnamurthy[c]

[a] University of Pittsburgh, United States
[b] AUEB, Greece
[c] UC Riverside, United States

ARTICLE INFO

Article history:
Received 8 August 2015
Revised 7 March 2016
Accepted 17 April 2016
Available online 23 April 2016

Keywords:
Wireless networks
Jamming attacks
Gradient descent
Location discovery
Testbed experimentation

ABSTRACT

Jamming attacks have become prevalent during the last few years facilitated by the open access to the shared wireless medium as well as the increased motivation and easiness to create damage as a result of sophistication of wireless devices, both legitimate and jamming ones. Among the challenges that a wireless network faces while trying to confront the jammer, jammer localization is of utmost importance. This entails estimating the physical location of the jammer. Successful jammer localization can trigger a series of corrective measures to ensure sustainable network operation. However, locating the jammer is a difficult problem. Our primary goal in this paper is to design a simple, lightweight and generic approach for localizing a jamming device through a set of measurable parameters. The key observation guiding our design, is that the Packet Delivery Ratio (PDR) that can be readily measured locally by a device decreases as a receiver moves closer to the jammer. Further, we draw on the gradient-descent principle from optimization theory, and we adapt it to operate on the discrete plane of the network topology so that the jamming device location can be estimated. The very nature of the gradient-descent algorithm allows the distributed execution of our localization scheme. In this paper, we compute and experimentally validate the impact of jammer on the PDR of a link and we show that this impact decreases as the link moves away from the jammer. We further design a distributed, lightweight jammer localization system, which does not require any modifications to the driver/firmware of commercial NICs, while we implement a prototype system to evaluate our scheme on our 802.11 indoor testbed. Finally, we evaluate the performance of our system via extensive simulations in larger scale settings. Its performance in terms of average location estimation error in combination with its simplicity and distributed operations hold great promise.

1. Introduction

The widespread proliferation of 802.11 wireless networks makes them an attractive target for various types of attacks [1–3]. Its open access nature makes it fairly easy for saboteurs with jamming devices [4,5] to disrupt WiFi communications. A jamming device continuously emits electromagnetic energy on the medium. Numerous jamming attacks have been reported in the recent past [6–9]. The effect of this behavior on a CSMA/CA network is twofold: **(a)** at the transmitter side it renders the medium busy resulting in large back-off times and, **(b)** at the receiver side, it dramatically decreases the SNR resulting in a large number of packet collisions. Jamming effects may also occur due to accidental activation of devices that do not serve a malicious cause, such as microwave ovens, cordless phones [10], etc. Following the detection of the presence of an attacker [11], localizing the jammer allows an administrator to pursue further countermeasures (such as deactivating the jamming device, isolating the attacker and capturing, punishing or even destroying it).

In this work, we design and implement a simple, lightweight approach for jammer localization. The main attribute of our approach that makes it attractive to use and straightforward to implement, is that it relies on Packet Delivery Ratio (PDR), a metric that is readily available at each node and is an indication of transmission corruption. Our technique exploits an intrinsic

[☆] An earlier version of this work has appeared in IEEE Globecom 2009.

[*] Corresponding author. Tel.: +195182244780.

E-mail addresses: kpele@pitt.edu (K. Pelechrinis), jordan@aueb.gr (I. Koutsopoulos), broustis@cs.ucr.edu (I. Broustis), krish@cs.ucr.edu (S.V. Krishnamurthy).

http://dx.doi.org/10.1016/j.comcom.2016.04.009
0140-3664/© 2016 Elsevier B.V. All rights reserved.

characteristic of the wireless medium: since the power of the jamming signal degrades with distance, farther transmitters do not sense strong jamming signals. As a consequence, the requirements for successful packet delivery at such transceivers are satisfied. This property cannot be manipulated by an attacker. A transceiver pair located further away from a jammer is more likely to be successful in exchanging packets; the transmitter is able to send more packets, while the receiver can decode more of those, due to increased SINR, resulting in an increased PDR.

Taking this property into account we design a simple localization algorithm, that borrows its rationale from the gradient-descent method in a continuous-valued variable space. Our algorithm starts from an initial node and terminates at another node, that is closer to the jammer than any of its neighbors. In particular, it is distributed and is progressively executed by nodes moving towards the proximity of the attacker. Specifically, nodes successively forward PDR measurements to neighbors towards assessing patterns related to PDR growth or degradation. The above structure of the algorithm is reminiscent of the iterative gradient-descent algorithm for identifying the minimum of a real-valued function f. The gradient-descent algorithm iteratively searches for a global optimum by moving from one point $\vec{x}_n$ of the function's domain S to another $\vec{x}_{n+1} \in S$. The point $\vec{x}_{n+1}$ is towards the opposite direction of the gradient of f at $\vec{x}_n$; this is the direction in which f exhibits the largest decrease with regards to its value at point $\vec{x}_n$. Note that in our case, the domain set consists of the discrete locations of the nodes. Hence, our scheme can be viewed as a discretized version of a gradient-descent algorithm. If the algorithm cannot proceed further, an optimum is declared[1]. As one can deduce, our scheme is greedy in nature, since each node takes the locally optimal choice to derive the global optimum (i.e., the position of the jammer).

Our full-fledged localization approach considers different starting points for the gradient-descent-based algorithm. We examine two algorithms as candidates for our approach. The first considers the distribution of the stopping points/nodes and applies a weighted centroid algorithm to estimate the position of the jammer. The second, which we include in our approach as the best solution, considers all the nodes where the *kernel*[2] algorithm stops, and declares as the jammer's position, the one with the smallest PDR. As might be evident, the latter scheme, similar to the kernel algorithm, always exhibits a non-zero error (since the position of the jammer is always assumed to be the same as that of a network node). However, as our evaluations indicate, it significantly reduces the uncertainty with respect to the position, as compared to both the vanilla gradient-descent-based algorithm and the weighted centroid algorithm.

Our main contributions in this work can be summarized as follows:

- **Analytical and experimental assessment of the spatial effects of jamming:** As previously mentioned, the jammer may affect both the transmitter and receiver operations; this has an impact on the PDR. We provide an analytical expression for quantifying the change in PDR at different locations in the network (relative to the jammer's location). We validate the analytically computed expression via real experiments on our 802.11 wireless testbed. Specifically, we show that the tranceivers that are further from the jammer exhibit lower (or no) degradation in terms of PDR as compared to transceivers that are located closer to the jammer.
- **Design of a lightweight jamming localization algorithm:** Having shown that PDR is minimized in the vicinity of the malicious device, we design a gradient-descent based algorithm to locate the adversarial node. We further design two algorithms that are built on top of the above core algorithm to improve accuracy; one is based on weighted centroid localization and an *annealing*-like extension which provides the best performance in terms of localization and thus, it is used in our approach. The main advantages of our approach (as compared to previously proposed localization approaches) are: (a) simplicity, (b) does not require any special hardware support, and (c) can be easily integrated with higher layer functions, such as routing, to circumvent the jammer's location.
- **Implementation and evaluation of our scheme:** We implement a prototype of our approach on our wireless testbed using the `Click modular router` [12]. We validate its performance through experiments on our indoor 802.11 testbed. We also evaluate the scalability of our approach through simulations (with larger topologies).

Our work in perspective: Our goal is to exploit the inherent propagation characteristics of the wireless channel in order to expose the presence of jamming devices and localize them. The jamming attacker might be able to hide itself from all but the wireless channel's propagation characteristics. The attributes of the jamming signals (and in particular their spatial properties) can affect measurable attributes (such as the PDR) to varying degrees in different parts of the network, thereby revealing important information with regards to the location of the malicious device. The key novelty of our scheme is its distributed nature and its lightweight operations.

In particular, our proposed algorithms offers the benefit that they rely on the operations of existing network functionalities and measurable quantities at a device level. Hence, no additional hardware or mechanisms are needed. Moreover, to reiterate, the nature of the gradient-descent algorithm allows the distributed execution of our localization scheme. Furthermore, the achieved localization error, which is at the range of one communication hop[3] significantly reduces the area that one needs to search for locating the misbehaving device. Equally novel and crucial is the adoptability of the designed scheme. In particular, the kernel can be used as a standalone module, the output of which can be processed in many various ways (e.g., a simulated annealing-like algorithm, a simple centroid calculation algorithm etc.). This flexibility further allows for building systems that can deal with more advanced attack models (see Section 5.5).

The rest of the paper is organized as follows. Section 2 provides the required background and describes related studies. Section 3 describes our analytical framework for quantifying the jamming effects on the PDR. Section 4 provides a progressive description of our component algorithms starting from the basic version to the full-fledged scheme. We present our experimental setup and evaluations in Section 5. Our conclusions form Section 6.

2. Background and related studies

In this section we present representative studies of different types of localization algorithms. We further briefly introduce the gradient-descent optimization method and discuss approaches that have utilized it for network operations.

Signal processing-based localization techniques: Secure mobile device localization, and in particular jammer localization, has been studied in the literature. Various approaches have been

[1] As we will see later this optimum is possibly local.

[2] We will use the words core, kernel and *vanilla* interchangeably in the rest of the manuscript.

[3] We prefer referring to this relative notion of error, since it puts results in perspective. For instance, a localization error of 20 m can be considered small for a WiFi network but it is certainly not small in the context of sensor or bluetooth networks, whose communication ranges are much smaller.

proposed in order to locate the malicious device, such as the studies in [13–17]. However, all of these studies use advanced signal processing techniques and operate at the PHY layer. In addition, they require special, additional infrastructure in order to achieve their goal (e.g. ultrasound, infrared or laser infrastructures). These features obstruct the wide deployment of such techniques in current commercial wireless networks. A detailed description of various secure positioning systems, that exclusively operate at the PHY layer, can be found in [18].

Received signal strength (RSS) based localization techniques: In addition to the above schemes, various studies utilize RSS measurements in order to discover the location of wireless devices, and in particular the positions of access points (APs). Most of these techniques require measurements of RSS at various positions (***wardriving***). Some well-known approaches belonging to this category are the *(weighted) centroid* [19] and *trilateration* [20]. Both these techniques combine measurements of the RSS at various locations in order to infer the position of the AP. Subramanian et al. [21] propose a localization algorithm that utilizes steerable, directional antennas in order to get information with regards to the Angle of Arrival (AoA). This can significantly reduce the localization error. In a different approach, the authors in [22] manage to derive AoA equivalent information by simply measuring the RSS. All of these schemes, require *wardriving* and can be considered as centralized algorithms; a set of previously collected measurements, including coordinates and the corresponding RSS, are needed in order to apply the algorithms and identify the position of the AP. In a slightly different context Chen et al. [23] combine environmental information gathered from sensor networks in order to perform localization. All data are gathered at the base station and are analyzed in order to identify the locations needed; centralized localization is again performed.

Our approach is different from the previously proposed schemes. In particular it does not require additional, specialized infrastructure in order to operate (in contrast with signal processing systems). No changes at the driver/firmware of commercial NICs are required. Our localization system can be integrated with higher layers, as we discuss later in this paper. One could expect that the RSS-based algorithms could be modified in order to locate a jamming node; areas close to the jamming device might exhibit extremely high RSS values due to the jamming signals [24].

Recent work from Liu et al. [25] further provides a way to estimate the jamming signal strength, which can provide a more accurate localization. However, the advantage of our approach over the RSS-based systems is that it does not require calibration measurements and it can be executed online and in a distributed manner. Recently, and after our initial study [26] on using readily available network metrics for coarse-grained jammer localization, some studies use similar rationale. Liu et al. [27] propose Virtual Force Iterative Localization (VFIL), which is an iterative method. An initial, coarse-grained, position estimation is performed, and in each iteration of the algorithm the accuracy and the granularity of localization are improved. Cheng et al. [28] further use basic geometric concepts to improve the performance of VFIL, while the same authors [29] provide a rudimentary system based on SNR measurements and bifurcation points on skeletons of jammed areas. In [30] the authors design a jammer localization scheme based on jamming-caused neighbor changes. A least-squares (LSQ) problem derived from the changes in nodes' hearing and sending range is formulated and solved for the jammer localization. Their wireless sensor network emulations show that the scheme can achieve (mean) localizations errors between 10–20 m when there are signal irregularities. Cai et al. [31] developed a specialized system that is based on RSS indirectly by utilizing the busy-time periods of access points to first detect jammed access points and then perform a coarse-grained localization of reactive jammers in enterprise WLANs. In comparison, our scheme requires only PDR measurements, which can be obtained from existing device functionalities. For instance, the probing functionality of link quality-based routing protocols (e.g., ETT [32]) already provides this information. More importantly, we show the feasibility of our proposed scheme through experiments on our 802.11 testbed with a prototype implementation of our main algorithm.

We would like to point out here that the current study is an extension of our preliminary work [26]. Compared to our previous study, the current work includes two full-fledged localization algorithms, which build on top of the core algorithm that was the main focus of our preliminary investigation. We have further included additional measurement results to validate/evaluate the accuracy of our analytical model on the effects of jamming on the PDR. Finally, we also provide specific proposals/steps for adapting the scheme to accommodate more advanced jamming strategies such as mobile and on-off jammers.

Gradient-descent minimization: Gradient-descent is an optimization method for real valued functions. In particular, assume that function f is defined on R^n and it is convex. In order to find the minimum of f, one may start from a point $\vec{x}_0 \in R^n$ and continue finding a series of points using

$$\vec{x}_{n+1} = \vec{x}_n - \gamma_n \cdot \nabla f(\vec{x}_n), \tag{1}$$

where $\nabla f(\vec{x}_i)$ is the *gradient* of f and and γ_n is the step at iteration n. The gradient of f at point $\vec{x}$ is the direction of the maximum increase of the function at $\vec{x}$. Starting from an arbitrary point, the algorithm greedily moves towards the direction of maximum decrease of the function at the neighborhood of this point (-$\nabla f(\vec{x}_n)$). After a series of iterations, the algorithm will converge, at least to a local optimum and possibly to a global optimum[4].

Gradient-based routing: The idea of incorporating features from gradient optimization into network operations has been used in the past for routing. In particular, Faruque et al. [33] propose the use of a gradient-based algorithm for the efficient forwarding of queries in sensor networks. Poor [34] presents an *on-demand* routing protocol for ad hoc networks, which uses a gradient-descent logic in order to forward the packets based on the *cost to destination*. In particular, the source broadcasts the message along with its cost to deliver it. Consequently, only the neighboring nodes that can deliver the message at a smaller cost relay the packet. In a similar fashion, Ruhil et al. [35] forward the message to the neighbor node that is closer to the direction of the destination.

3. System model and jamming effects on PDR

System model and metrics: We consider a wireless multi-hop (ad-hoc or mesh) network. We further assume that there exists a static malicious device whose location is unknown to the network operator. This device is a MAC layer jammer that aims at packet disruption at the transmitter and/or receiver of a wireless link. For our attack model we will consider a continuous-deceptive jammer that transmits continually seemingly legitimate packets on the medium [24]. Finally, central to our work is the PDR. PDR is defined as the ratio of the number of packets that are acknowledged at the transmitter and the number of packets that enter in its MAC layer queue. In the literature, PDR is defined as the ratio of acknowledged packets and the packets that are transmitted. For a carrier sensing access protocol, Packet Sent Ratio (PSR) is also used to capture the performance of the transmitter. In particular, PSR refers to the ratio of the number of packets that are sent out from the transmitter and the total packets that enter its MAC layer

[4] Depending on the initial point, the algorithm might be trapped at a local minimum.

queue from the upper layers. However, in order to keep our analysis tractable we will use our above definition, which implicitly integrates PSR into PDR calculations.

The presence of a jammer has a significant effect on the performance of a link. In particular, there are three possible ways that a (successful) packet transmission can be affected: **(i)** the transmitter (denoted as T_x) senses the medium busy due to jamming signals, **(ii)** the reception at the receiver (denoted as R_x) fails due to low SINR at its antenna because of the jamming signals and **(iii)** the reception of the MAC layer ACK packet fails due to low SINR at the T_x antenna. Since the above are statistically independent, the PDR can be expressed as

$$PDR = P_{T_{x_{send-DATA}}} \cdot P_{R_{x_{receive-DATA}}} \cdot P_{T_{x_{receive-ACK}}}, \tag{2}$$

where $P_{T_{x_{send-DATA}}}$ is the probability that T_x will sense the medium idle and transmit its packets, $P_{R_{x_{receive-DATA}}}$ is the probability that the SINR requirement at R_x is satisfied and $P_{T_{x_{receive-ACK}}}$ is the probability that the SINR requirement at T_x (for receiving the ACK) is satisfied as well. Note that we do not include the probability that the R_x is 传感介质空闲 sensing the medium idle for the transmission of the MAC layer ACK; once R_x correctly receives the DATA packet it does not perform carrier sensing in order to send out the ACK [36].

In order to calculate these probabilities we need to assume a signal propagation model. We adopt the model from [37] and we calculate the received power P_r at distance r when the transmission power is P as

$$P_r = \frac{P}{r^{\alpha}} \cdot Y, \tag{3}$$

where α is the path loss exponent, and Y is a random variable that is log-normally distributed, it captures the shadow fading effects, and it has a mean value of 1 and a standard deviation equal to the shadow fading variation which we can obtain from measurements. In our analysis, we will use the following notation:

- P_{JT} is the signal strength of the jamming signal at T_x,
- P_J is the transmission power of the jammer,
- r_T is the distance between the jammer and T_x,
- r_R is the distance between the jammer and the R_x,
- P is the transmission power on the link,
- d is the distance between T_x and R_x,
- CCA is the Clear Channel Assessment threshold,
- u is the SINR requirement for the rate used,
- N is the thermal noise floor, and
- (μ, σ) are the parameters of the log normal distribution (computed from the mean value and the standard deviation of the r.v. Y).

Using the introduced propagation model the terms in (2) can be expressed as follows[5]:

$$\begin{aligned} P_{T_{x_{send-DATA}}} &= P\left\{P_{JT} < CCA\right\} = Pr\left\{\frac{P_J}{r_T^a} \cdot Y < CCA\right\} \\ &= P\{Y < \frac{CCA \cdot r_T^a}{P_J}\} = \frac{1}{2} + \frac{1}{2} \cdot erf\left(\frac{\ln(\frac{CCA \cdot r_T^a}{P_J}) - \mu}{\sqrt{2} \cdot \sigma}\right) \end{aligned} \tag{4}$$

$$\begin{aligned} P_{R_{x_{receive-DATA}}} &= P\left\{SINR_{R_x} > u\} = P\{Y > \frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_R^a}}\right\} \\ &= \frac{1}{2} - \frac{1}{2} \cdot erf\left(\frac{\ln(\frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_R^a}}) - \mu}{\sqrt{2} \cdot \sigma}\right) \end{aligned} \tag{5}$$

$$\begin{aligned} P_{T_{x_{receive-ACK}}} &= P\{SINR_{T_x} > u\} = P\{Y > \frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_T^a}}\} \\ &= \frac{1}{2} - \frac{1}{2} \cdot erf\left(\frac{\ln\left(\frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_T^a}}\right) - \mu}{\sqrt{2} \cdot \sigma}\right). \end{aligned} \tag{6}$$

Substituting Eq. (4)–(6) in (2) we obtain an expression for the PDR on a link as a function of r_T and r_R. Fig. 1 presents the PDR for various distances from the jammer and various link lengths. In generating these plots we have used the following values: (i) $P = P_J = 18$ dBm, (ii) $CCA = -80$ dBm, (iii) shadow fading signal variation is 10 dBm (value measured on our testbed) and (iv) path loss exponent is equal to 5 (this is a typical value for the path loss exponent in indoor environments [37]).

There are two main observations that we can derive from these analytical results. First, **areas in the vicinity of the jamming device (approximately 25–30 m - one hop away), exhibit very low PDR.** This forms the basis for our localization algorithm described in the following section. Second, **shorter links, i.e., links where the transmitter and the receiver are in close distance, are more robust to jamming, since they can satisfy the SNR requirements with higher probability.**

In Section 5 we present experimental results that validate our analysis. Our analytical and experimental results, demonstrate the decrease in the PDR due to the presence of a jammer and its minimization in the proximity of the latter, thus, justifying its usage in our localization scheme.

4. The proposed jammer localization algorithm

In this section we develop our localization scheme. We start by formally introducing our core algorithm, namely, the gradient-descent-based localization. We then present two full-fledged (also referred to as *wrapper* in what follows) algorithms. The first one is based on computing a weighted centroid, while our algorithm resembles the annealing optimization procedure.

4.1. Gradient-based kernel

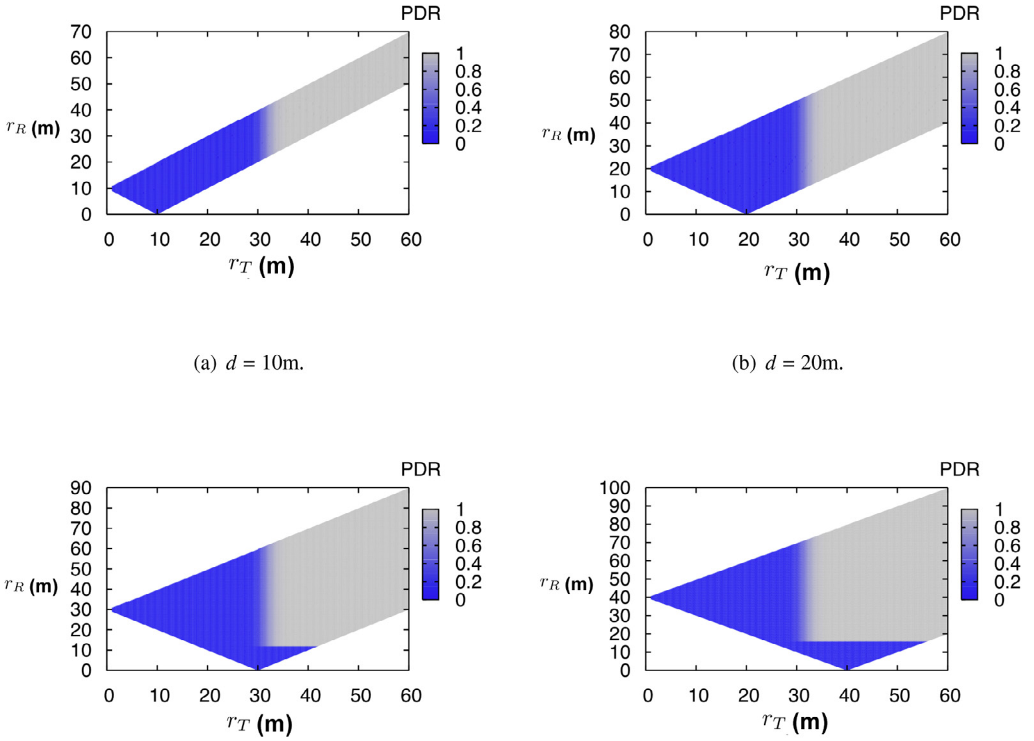
As it was mentioned in the previous section, the PDR value decreases as we move closer to the jammer. Hence, we can modify the gradient-descent method in order to localize the jammer. Function f can represent now the PDR, while the next candidate points $\vec{x}_{n+1}$ may stand for the neighbors of the node under consideration. Since we adapt the continuous-valued gradient-descent method to work in a discrete solution space, the differential PDR of two neighboring locations is the discrete analog to the gradient's magnitude of a continuous-valued function. In particular, each node tries to find its neighbor node with the largest decrease in PDR. Algorithm 1 presents a pseudocode for this basic scheme, while Algorithm 2 presents a pseudocode for the operations executed at each node and used by Algorithm 1.

In the above notation, PDR_i is the PDR of node i. However, PDR is measured on a link, rather than a node. Hence, in order to calculate PDR_i we can use the average value of the PDR of the links between node i and its neighbors. Specifically

$$PDR_i = \frac{\sum_{m=1}^{|\mathcal{NS}_i|} PDR_{im}}{|\mathcal{NS}_i|}, \tag{7}$$

where $\mathcal{NS}_i$ the set of neighbors of i, PDR_{im} is the PDR on link i-m and $|\mathcal{NS}_i|$ is the cardinality of set $\mathcal{NS}_i$, i.e., the number of neighbors of node i. Using this average value makes sense since

[5] Note that we consider a single rate network, and in particular a network operating at the basic rate (6 Mbps).



(a) $d = 10$m.

(b) $d = 20$m.

(c) $d = 30$m.

(d) $d = 40$m.

Fig. 1. Analytically computed PDR. Shorter links are more robust, while areas around the jamming device exhibit low PDR.

```
Data: Starting node $x_s$
Result: Node closest to the jammer, $x_e$
begin
    $x_* = x_s$
    while $x_* \neq$ grad_node($x_*$) do
        $x_* =$ grad_node($x_*$)
    end
    $x_e = x_*$
    return
end
```

Algorithm 1: Pseudocode for the gradient-based kernel algorithm starting at point x_s– `grad_kernel(`x_s`)`.

```
Data: Neighbors' i PDR, $PDR_i$
Result: Next node $n$ closer to the jammer
begin
    Pick $k : (PDR_i - PDR_k) > (PDR_i - PDR_j)\ \forall j \neq k$
    $\Delta = (PDR_i - PDR_k)$
    if $\Delta > 0$ then
        $n = k$
        else
            $n = i$
        end
    end
    return
end
```

Algorithm 2: Pseudocode for the localization scheme running on node i – `grad_node(i)`.

one can expect the jammer to impact the PDR on all of a victim's associated links to approximately the same extent. However, in our experimental evaluations different variants of the definition above are tested such as, $PDR_i = \min_{1 \le m \le |\mathcal{NS}_i|} PDR_{im}$ and $PDR_i = \max_{1 \le m \le |\mathcal{NS}_i|} PDR_{im}$.

4.2. Weighted centroid-based wrapper

Gradient-descent minimization is sensitive to the starting point of the algorithm. A bad starting point may lead to a local optimum, and in our case a location far from the jammer. One way to eliminate or at least reduce this sensitivity is not to rely on a single starting point, but apply our kernel algorithm (Algorithm 1) multiple times and in parallel with randomly picked distinct nodes that initialize the process.

Each run of the algorithm will stop at a particular node. Considering the weight of each node j to be the frequency with which the `grad_node` terminates at node j, we can then compute their weighted centroid. This is the estimated position of the jamming device. Of course each node that is a terminal node at least once, needs to report its weight to a central entity which will then assume the role of computing the centroid. Note here that, central controllers exist in many wireless network architectures such as, mesh and WiFi enterprise networks. Algorithm 3 presents the steps followed by our weighted centroid full-fledged localization scheme. The centroid computation mentioned in the last step (line 11) follows the traditional definition from analytical geometry. In particular, the weighted centroid of n points is given by

$$\vec{x} = \frac{\sum_{i=1}^{n} w(i) \cdot \vec{x_i}}{n} \quad (8)$$

```
Data: Set of starting nodes $\mathcal{K} \subseteq \mathcal{N}$, coordinates $\vec{c_i}$, $\forall i \in \mathcal{K}$
Result: Estimation of jammer's position $\vec{x}$
begin
    $\mathcal{S} = \emptyset$ // The set of stopping nodes
    for $\forall i \in \mathcal{N}$ do
        $w(i) = 0$ // The weight of each node
    end
    for $\forall i \in \mathcal{K}$ do
        $y = \texttt{grad\_kernel}(i)$
        if $y \notin \mathcal{S}$ then
            $\mathcal{S} = \mathcal{S} \bigcup \{y\}$
        end
        $w(y) = w(y) + 1$
    end
    for $\forall i \in \mathcal{S}$ do
        $w(i) = w(i)/|K|$
    end
    $\vec{x} = Centroid(w_i \cdot \vec{c_i})$, $over\, all\, i \in \mathcal{S}$
    return
end
```

Algorithm 3: Pseudocode of our weighted-centroid full-fledged localization scheme.

where $w(i)$ is the weight assigned to point $\vec{x_i}$. The cardinality $|\mathcal{K}|$ of the set of initializing nodes $\mathcal{K}$, is a control knob of the algorithm, whose effect will be evaluated and quantified in our evaluations.

In essence, the algorithm above requires multiple runs of the `grad_kernel(`x_s`)`, for different starting points x_s. With this process, we get the relative frequency (which serves as an estimate of the probability) that the algorithm will terminate at a given node. This relative frequency is then used to weigh the contribution of each stopping point, in the centroid calculation.

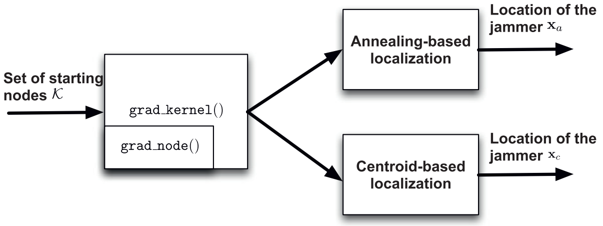
4.3. Annealing-like wrapper

Given that many localization algorithms in the literature make use of the notion of centroid, the weighted centroid-based wrapper can serve as a baseline algorithm. However, the fact that we utilize multiple points where the core algorithm has stopped at, may in some cases increase the location estimation error introduced by the core algorithm. Since, there can be only one global minimum[6], taking into consideration multiple stopping points in essence means that we use multiple local minima. Hence, the error introduced may still be large.

Therefore, we propose an alternative wrapper algorithm, which resembles an annealing process. In particular, we run our gradient-based kernel algorithm again starting from multiple randomly selected nodes. Choosing the stopping node with the minimum PDR as the location of the jammer can reduce the error significantly. In other words, we identify local minima, and we pick the location of the node with the smallest PDR from among them as an *approximation* of the jammer's position. Algorithm 4 formalizes these steps. Again $|K|$ is a parameter of the algorithm, whose effect we examine in the evaluations. Even though this algorithm will always exhibit a non-zero location estimation error since the location of a jammer will always be identified as a position of an existing network node, its overall performance is significantly better compared to the kernel approach and the weighted-centroid wrapper (Section 5).

```
Data: Set of starting nodes $\mathcal{K} \subseteq \mathcal{N}$, coordinates $\vec{c_i}$, $\forall i \in \mathcal{K}$
Result: Estimation of jammer's position $\vec{x}$
begin
    $\mathcal{S} = \emptyset$ // The set of stopping nodes
    for $\forall i \in |K|$ do
        $y = \texttt{grad\_kernel}(i)$
        if $y \notin \mathcal{S}$ then
            $\mathcal{S} = \mathcal{S} \bigcup \{y\}$
        end
    end
    $\vec{x} = c_r \,|\, PDR(c_r) \leq PDR(c_j),\ \forall j \in \mathcal{S},\ j \neq r$
    return
end
```

Algorithm 4: Pseudocode of our annealing-based full-fledged localization scheme.



Set of starting nodes $\mathcal{K}$
grad_kernel()
grad_node()
Annealing-based localization
Location of the jammer x_a
Centroid-based localization
Location of the jammer x_c

Fig. 2. The inter-workings of our algorithms.

4.4. Protocol implementation

The weighted centroid-based wrapper in pseudocode 3 and the annealing-like wrapper in pseudocode 4 describe the general steps of two complete localization schemes. However, Algorithm 4 is our final proposed scheme. Even though the core algorithm presented in Section 4.1 is fully distributed, in order to localize the jammer using the full-fledged schemes, cooperation between the nodes is required. Each node, needs to initiate once (either independently[7] or upon being prompted by a central authority – e.g., the central controller of a mesh network or of a WiFi enterprise network) in parallel, the gradient-descent-based method above. Each node records the number of times it has been the terminal node of this search (initiated from other peers in the network). After reporting this information along with its PDR to their peers via control packets, the jammer's position can be estimated by each scheme presented.

These protocols are distributed in nature but might require light coordination using a central entity for processing the data as is the case for any localization algorithm in a multi-hop network. For instance, in [30] a *designated* node is required to collect evidence from the network and perform the localization. Note here that, even in the case where each node can independently estimate the jammer's location, a central authority is likely to be responsible for further actions against the malicious entity such as physical capture and disablement of the jammer.

Fig. 2 depicts a flow chart that shows how our algorithms are combined to the full-fledged localization scheme. The core of the operations is `grad_kernel()`, which further relies on the `grad_node()` that runs individually on each node. The output from

[6] Of course, in theory there are functions that exhibit multiple minimums, but in our case the global minimum is found only around the area of the jammer. This is especially true if we consider that all the nodes in the network are exposed to the same wireless environment and no node suffers from severe fading as compared to his peers.

[7] E.g., with a pre-defined ruleset or with local neighborhood voting-based schemes.

these algorithms (for different starting nodes $\mathcal{K}$) is passed to our wrapper algorithms, which provide us with the final location of the jammer.

4.4.1. Computational complexity and system overhead

An important aspect of every protocol/system is its computational complexity as well as the overhead imposed by its use. The `grad_kernel()` algorithm is linear to the size of the network, i.e., $O(n)$, where n is the number of nodes in the network. The overall complexity of the full-fledged localization algorithms is then $O(kn)$, where k is the number of starting points for the localization. In practice (see Section 5.3) the number of iterations needed for the `grad_kernel()` to converge is a fraction of n. With respect to the overhead added from our localization scheme this is zero! In particular, all the functionality of the localization is piggybacked on the routing functionality of ETT as described in more details in Section 5.3. Hence, there is no additional control message overhead imposed by the system.

5. Performance evaluation

In this section we present the experimentation and simulation-based evaluations of our scheme. We first verify our analytical results through measurements on a real testbed. We continue by presenting the evaluations of our full-fledged schemes through simulations on a large scale topology. Finally, we describe a prototype implementation of the core algorithm based on gradient-descent minimization. This serves as a proof-of-concept for the practicality of our design. We further showcase its applicability through a small-scale experiment.

Testbed description: Our testbed is deployed in the 3rd floor of Engineering Building 2, at the University of California, Riverside. The testbed consists of 42 nodes; 22 of them are Soekris net5501 nodes, which mount a Debian Linux distribution with kernel v2.6 over NFS and are equipped with a miniPCI *EMP-8602 6G* 802.11a/g WiFi card with the Atheros chipset. The other 20 nodes are Soekris net4826; they mount the same Debian Linux distribution, and are additionally equipped with an *Intel-2915* mini-PCI card. We use a 5 dBi omnidirectional antenna for every node and the transmission rate is 6Mbps unless otherwise stated. We use the Madwifi-ng driver for our Atheros based cards and a proprietary version of the *ipw2200* driver/firmware of the *Intel-2915* cards, which allows for tuning the CCA. More details on our testbed deployment can be found in [38].

Jammer implementation: For the purposes of our work we implement our own constant-deceptive jamming utility [24]. In particular, a constant-deceptive jammer transmits continually seemingly legitimate packets on the medium. The implementation is based on a specific configuration (CCA = 0 dBm) and a user space utility that sends broadcast packets as fast as possible. By setting the CCA threshold to such a high value, we force the device to ignore all legitimate 802.11 signals even after carrier sensing. Packets arrive at the jammer's circuitry with powers less than 0 dBm (even if the distances between the jammer and the legitimate transceivers are very small [39]). In addition, having the jammer transmit broadcast packets allows the deferral of back-to-back transmissions for the minimum possible time. Given that transmissions of MAC layer ACK packets are by default disabled for broadcast traffic, the jammer needs to wait for the *DIFS* time, plus the minimum possible backoff period $min_{BackOff}$ (i.e., $DIFS + min_{BackOff}$) [36].

5.1. Validation of our theoretical assessments

Table 1
Our analytical model predicts well the effect of a jammer on the PDR of a link.

d(m)	r_T (m)	r_R (m)	PDR measured	PDR analytical
10	32	36	0.68	0.64
10.5	18.7	18.9	0.02	0
8.1	28	25.3	0.1	0.013
7.3	30	25	0.12	0.19

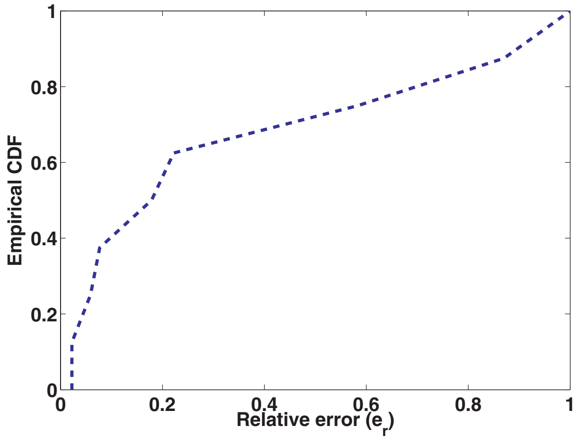


Fig. 3. Empirical CDF for the relative error e_r.

We start by validating our model presented in Section 3. We activate the jamming nodes one at a time and we measure the PDR on different links on our testbed. We perform our experiments late at night in order to avoid interference from other collocated wireless LANs that are active during the day, and we also operate each link in isolation (no other link active at the same time).

Table 1 shows the detailed results for a subset of our experiments. In particular, we compare the PDRs observed in practice with those that are anticipated from theory. We observe that there is a good match between the measurements and the analysis.

Fig. 3 depicts the empirical CDF for the relative error e_r:

$$e_r = \frac{|PDR_{measured} - PDR_{analytical}|}{PDR_{measured}} \tag{9}$$

The majority of the relative errors are small, which translates to a fairly accurate analytical model. However, there are some discrepancies that can be attributed to the fact that the path loss exponent used in the model might not match exactly the one that characterizes the real environment. Nevertheless, the jamming effects observed on our testbed are similar to what is estimated from our analysis. More than 60% of the estimations using our analytical model exhibit error within 20% of the actual PDR value measured on the testbed. Moreover, note that our scheme does not rely on the analytical model parameters. We use online measurements to capture the actual PDR.

These results can be further seen as a validation of our theoretical intuition that the PDR values decrease in areas closer to the jammer. Hence, they form a strong motivation for our framework design for our localization algorithms.

5.2. Localization error performance

We evaluate the weighted centroid and annealing-based localizations using large scale simulations in MATLAB. We consider a network of 100 nodes, randomly placed in a rectangular area of 500 x 500 m^2. The jammer is also randomly positioned within the area. We run 100 different topologies and obtain the distribution of the localization error. We utilize the lognormal shadow fading propagation model with the following parameters: (a) CCA = −80 dBm, (b) the shadow fading variation is 5 dBm (as it has

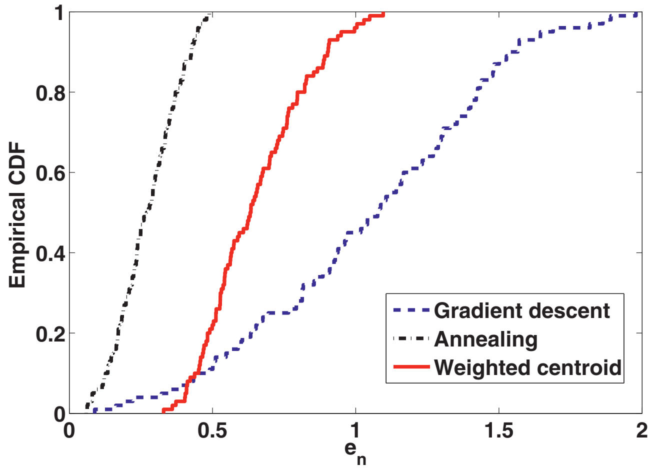


Fig. 4. Empirical CDF for the normalized error e_n.

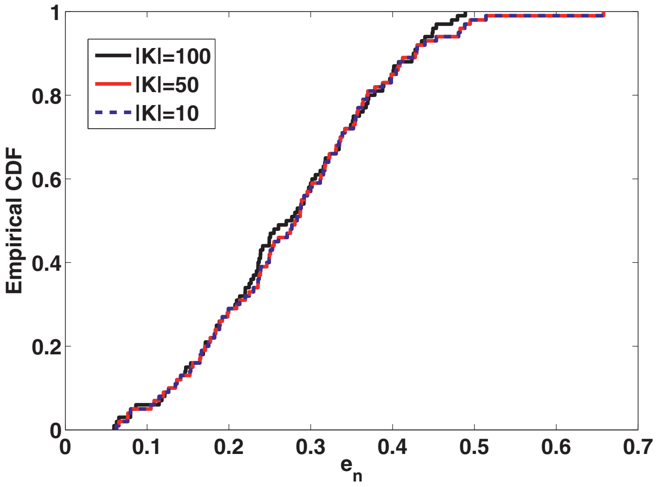


Fig. 5. Empirical CDF for the normalized error e_n of the annealing-like algorithm for different cardinality of set $\mathcal{K}$.

been measured on our testbed) and (c) the path loss exponent is 5 [37,40]. Our evaluation metric is the normalized error with respect to the communication range, namely, "one-hop" length of a node:

$$e_n = \frac{absolute\ localization\ error}{communication\ range} \tag{10}$$

Given that our algorithms operate on the discrete solution plane, the absolute error will not reveal a lot of information. The closest we can get is "one-hop" away from the jammer, that terminates at the node that is nearest to the malicious device. Hence, our objective is to keep the above error smaller than 1, which means that the distance between the estimated and the actual jammer's position is within the network's communication range[8]. Fig. 4 presents the results for both the weighted centroid and the annealing-like algorithms, when $\mathcal{K} = \mathcal{N}$. We also present the performance of the gradient-based kernel algorithm. As we observe, annealing outperforms all the other schemes, with the error being always less than half the communication range. The weighted centroid algorithm exhibits much higher error, since we use all terminating points of the gradient-descent core. Hence, we are using many local minima in our estimation, which leads to a higher localization error. Finally, our core algorithm (with a randomly chosen starting point) performs the worst, as one might have expected.

In the above simulations we have set $\mathcal{K}$, the set of starting points, to be equal to the set of the nodes in the network ($\mathcal{N}$). In other words, each node in the network initiates the kernel algorithm in parallel. However, we have examined the performance of the annealing-based algorithm with a smaller cardinality of set $\mathcal{K}$. Fig. 5 shows that even though there is a slight degradation when the number of starting nodes is smaller, this degradation is not significant. In all cases, the observed error is always smaller than approximately 68% of the coverage range. Hence, we could reduce the number of parallel runs of the algorithm significantly (by an order of 10), without compromising accuracy.

Finally, we examine how the annealing-based algorithm performs for different definitions of the PDR of a node. Recall that all of our results up to now are obtained by assuming that the PDR of node i is defined as per Eq. (2). In what follows, we examine two alternative definitions, that is, the PDR of node i is considered to be the max or min of its PDRs across its neighbor nodes. Fig. 6 depicts our results. When we use the minimum PDR of a node in our scheme, the performance is degraded. This can be attributed to the fact that we are taking a conservative approach; we project the worst performance of the links of the node under consideration (say i) to all of its links. Hence, this may cause the iteration to be trapped around local minima with higher probability, since it might be the case that this worse performance is due to the wireless channel uncertainty. A neighboring node (say j) that is more affected from the jammer than i, might exhibit a lower average PDR but higher than that of the worse - fading induced - PDR of the latter. This will further lead to a local minimum. On the contrary, when we use the maximum value of a node's PDRs, we adhere to a more optimistic approach since we assume that the effect of the jammer on node i is equal to that on its best link. This provides a performance almost identical to the average PDR, since all links of node i are expected to be affected at the same degree from a node, except the ones that experience additional severe fading due to the channel conditions.

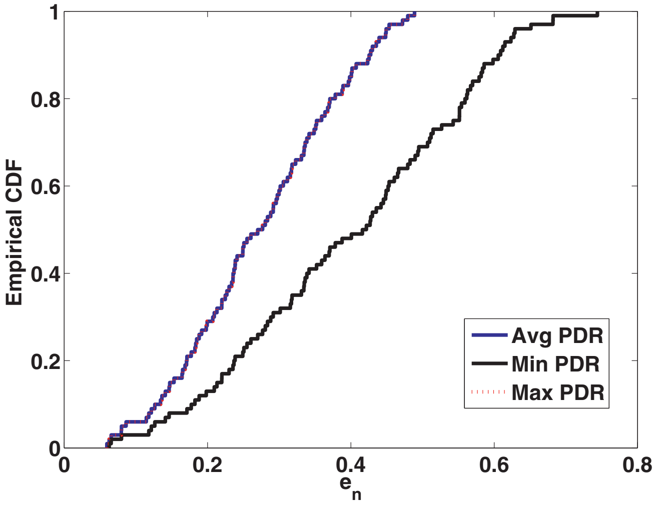


Fig. 6. Empirical CDF for the normalized error e_n of the annealing-like algorithm for different definitions of the PDR of a node.

5.3. Prototype system implementation of our core algorithm

We have implemented a prototype version of our core localization scheme in order to show the practicality of the proposed

[8] A typical communication range for an indoors 802.11b/g system is 25–50 m. With the above parameters our channel model gives a coverage range at the high end of the above interval (i.e., ≈ 50 m)

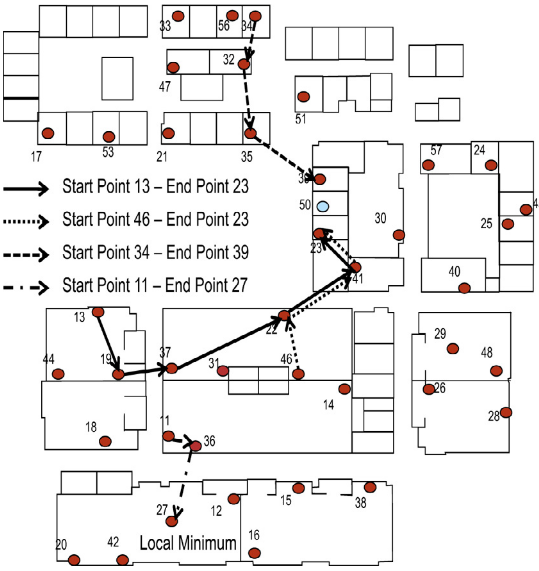

33
56
34
32
47
51
17
53
21
35
57
24
Start Point 13 – End Point 23
Start Point 46 – End Point 23
Start Point 34 – End Point 39
Start Point 11 – End Point 27
39
50
30
45
25
23
41
40
13
29
48
44
19
37
22
31
46
26
14
28
18
11
36
27
12
15
38
20
42
Local Minimum
16

Fig. 7. Pictorial view of the test-bed and jammer position.

schemes. We use the Click Modular Router framework and the Roofnet implementation from MIT. In particular, we have modified the code at sr2ettmetric.cpp of the Roofnet software framework [41] in order to retrieve the average PDR for each node (with respect to its neighbors). Our algorithm uses these values in order to perform the localization of the jammer. The dissemination of the PDR information in the network takes places along the lines of what is done with the ETT [32] functionality. In particular, a probe is transmitted every τ seconds and the PDR is calculated over a sliding window of w seconds (currently we have $\tau = 100$ ms and $w = 1$ sec). This implementation allows its integration with higher layer operations (and in particular routing) with no additional overhead. Note here that by building on top of ETT, we essentially do not require a control channel for PDR information exchange. Absence of probes/responses from a completely jammed node automatically translates to a PDR = 0, as it should. Less affected nodes will still be able to exchange the expected number of probes. In the rest of this subsection we present some proof-of-concept experiments on our testbed and their interpretations.

Experimental results: Our main goal is to observe how this prototype system progressively percolates through the network topology. Every node independently runs the localization algorithm and makes local decisions with regards to the next node that is closer to the jammer, based on the PDR values of its neighbors. This procedure continues until a node cannot identify one of its neighbors as being closer to the jammer than itself. The percolation can be thought of as a "route discovery" propagation towards the jammer.

We illustrate the functionality of the core algorithm with the following sample experiment. We activate one of our jamming devices on the testbed and run our localization algorithm on the rest of the nodes in our testbed in order to find the routes towards the jammer. Fig. 7 shows the various paths towards the jammer that were reported by our algorithm for various starting points (nodes). In this experiment, the jammer is node 50 (see Fig. 7). The arrows represent the path towards the jammer that our algorithm finds, for various starting points. We extract the following observations:

- All successful localization iterations[9] end at nodes within one hop distance from the jammer (i.e. 25–30 m).
- The paths to the jammer may be different. However, once two paths meet at an intermediate node, they *converge* and follow the same path until the termination of our algorithm.
- Depending on the starting point, our system could terminate at other end points and can be trapped at a local minimum (e.g., path 11 → 36 → 27). This is a feature that our scheme inherits from gradient-descent minimization technique, and as we saw above our full-fledged solutions can eliminate.

A more detailed examination of our experimental results reveals that when we start our search from nodes 13 and 46 we end up at node 23; this node is one hop away from the jammer. However, the paths followed are different; 13 → 19 → 37 → 22 → 41 → 23 and 46 → 22 → 41 → 23. Nevertheless, note that once the two paths meet at node 22, they follow the same sub-route to the jammer's location. In addition, starting from node 34, we manage to successfully localize the jammer once again, following a totally different path this time, that is, 34 → 32 → 35 → 39.

To reiterate, one collateral effect from incorporating the gradient-descent minimization method is that our scheme can be trapped in local minima. The performance of our proposed method is dependent on the choice of the initial point/node. For the example in Fig. 7, our measurements reveal that if the localization procedure starts at node 11, it will result in an inefficient localization. Specifically, our algorithm follows the path: 11 → 36 → 27, and falsely concludes that the jammer is in the vicinity of node 27. This can happen for various reasons. For instance: **(a)** The random nature of wireless signal fading can cause the PDR in some areas of the network exhibit low values even without the presence of a jammer. **(b)** The links of node 27 might be inherently of bad quality (low PDR) as compared to the other links in the neighborhood of node 27 (indeed, this was the reason for being trapped to local minima in the experiment of Fig. 7). **(c)** Large-scale temporal variations in the medium can affect the performance of our localization scheme (e.g. instantaneous PDR drop due to movement of obstacles). In general, local minima can be attributed to the randomness of wireless channel fading. Due to channel fading randomness, it is possible that a node closer to the jammer, has a higher PDR than a node further from the jammer. In order to reduce this sensitivity, we have proposed our two wrapper algorithms in Section 4.2 and 4.3, whose evaluations on a larger scale topology revealed that can significantly reduce the localization error.

Finally, it is worth examining the computational complexity of the grad_kernel() implementation. In Section 4.4.1 we mentioned that the algorithm is linear to the number of nodes in the network. Our experimental results show that in reality the algorithm converges much faster. In particular, the three paths that converged to the right solution required on average just 3.6 iterations or just approximately 8% of the network size. Given that probes are transmitted every $\tau = 100$ ms, this means that it took on average approximately 360 ms to localize the jammer.

5.4. Summarizing our algorithms

The main problem with the kernel algorithm is its sensitivity to local minima. As our results show, for different starting points, the algorithm can terminate at a node, which is much further than the actual jammer. Further, since the position of the jammer is always declared to be a network node location, the error is always non-zero.

[9] With the term successful we refer to runs of our algorithm that indeed terminate *close* to the malicious device.

In order to overcome this pitfall, we first provided an alternative using a weighted centroid-based algorithm. In this approach, we initialize the kernel algorithm from different points in the network and based on the termination points we calculate a weighted centroid of the latter as the jammer's position. Since this position can be any point in the network, the localization error can potentially be as low as zero. However, as our evaluations indicate, even though the error is significantly lower as compared to the kernel algorithm it is still high enough. In addition, the weighted centroid wrapper requires to be run multiple times in parallel, as compared to the single run of the kernel algorithm.

In order to further improve the localization, we designed an annealing-like algorithm, which also makes use of our kernel. Even though it also needs multiple runs with different starting points and it always exhibits non-zero error (a node's position is always declared as the estimated jammer's location), the overall performance of this algorithm is significantly better and hence, this is our proposed scheme in this paper.

We would like to state that the runs that both wrapper algorithms require can take place in parallel. The overhead increase is minimal since each node forwards a single packet to the next node in the *chain*. Furthermore, running the algorithm in parallel from different starting points may also help in localizing multiple jammers. In the latter case, different starting points will be able to converge to different points of minimum PDR, which all will be candidate jammer locations.

5.5. Extensions of our scheme

Our system is based on gradient descend optimization method and hence, as we have discussed is sensitive to local minima. The algorithm guarantees that the optimum found in terms of localization error is a local one. While the simulated annealing-like wrapper is able to alleviate some of the problems, unfortunately we cannot guarantee a global optima. However, since the mechanism is of minimal computational complexity and almost zero overhead, it could be applied to an even greater number spatially diverse initial points - possible all the nodes in the network depending on its size - to check the convergence. Another narrow aspect of the current system is the attack model considered. In particular, the attack model we have considered in this paper assumes a static jammer. Nevertheless, there are more advanced jamming models that are possible. In what follows we briefly describe how our scheme can be extended to deal with different attack models.

Adaptive jammers: An adaptive jammer switches its transmission on and off in an effort to avoid being detected. The problem of optimally (i.e. as fast as possible) detecting such a jammer in a sensor network has been previously studied [42]. There, the observations were the number of collisions rather than the PDR. If the jammer is static, the localization problem is similar to that considered in the paper. The difference is that now there will be fewer PDR samples to consider in the decision, but the algorithm works in that case as well.

Mobile jammers: The problem changes from estimating the location of a static jammer to tracking the mobile trajectory process. The algorithm proposed in this paper will not converge if applied as is, since the location of the jammer changes with time. The problem would need to be casted in a different manner, i.e. the objective needs to be modified to minimizing the error of tracking the mobility process. Note however that, at each fixed point in the trajectory process, the optimization problem that needs to be solved is the one discussed in this paper, and the algorithm proposed can be applied to solve it.

Coordinated jammers: To tackle the problem, we need to have some assumptions about how the jammers coordinate. For instance, each one of them will be characterized by an on-off jamming process. If we know the number of jammers, then we can run one replica of the algorithm for each jammer to locate its location. The rest of the problem remains the same.

6. Conclusions

We design a low-overhead, distributed jammer localization algorithm. Our main observation that guides the construction of our system is related to the spatial effects of jamming. In particular, links that are further from the jammer experience higher PDRs as compared to nodes that reside closer to the jamming device. We adopt the rational of gradient-descent methods in order to resemble the searching process for the node that is closer to the jammer. The algorithm is greedy in nature; each node makes the locally optimal choice in terms of the neighbor with the least PDR and proceeds in that fashion towards the direction of the jammer location. This decision is based on parameters that are readily observable in experimental platforms like the PDR. In order to overcome the inherent problem of gradient descent methods with local optima, we propose two algorithms that attempt to find a good starting point for the algorithm. Our evaluations indicate that the annealing-like algorithm performs the best and can indeed efficiently estimate the jammer's location with a low position error.

Acknowledgment

We thank Dr. K. Papagiannaki from Telefonica Research (formerly with Intel Research), for providing the source code of the prototype driver. I. Koutsopoulos acknowledges the support of ERC08- RECITAL project, co-financed by Greece and the European Social Fund through the Operational Program Education and Lifelong Learning- NSRF (National Strategic Reference Framework) 2007–2013.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset, Commun. Surv. Tut. IEEE 18 (1) (2016) 184–208, doi:10.1109/COMST.2015.2402161.
[2] H. Berghel, J. Uecker, Wi-Fi attack vectors, Commun. ACM 48 (8) (2005) 21–28, doi:10.1145/1076211.1076229.
[3] K. Bicakci, B. Tavli, Denial-of-service attacks and countermeasures in IEEE 802.11 wireless networks, Comput. Stand. Interfaces 31 (5) (2009) 931–941, doi:10.1016/j.csi.2008.09.038.
[4] SESP jammers, (http://www.sesp.com/).
[5] ISM wide-band jammers, (http://69.6.206.229/e-commerce-solutions-catalog1.0.4.html).
[6] Jamming attack at hacker conference. http://findarticles.com/p/articles/mi_m0EIN/is_2005_August_2/ai_n14841565,
[7] Techworld news, (http://www.techworld.com/mobility/news/index.cfm?newsid=10941).
[8] RF jamming attack, (http://manageengine.adventnet.com/products/wifi-manager/rfjamming-attack.html).
[9] ISA: users fear wireless networks for control, (http://lists.jammed.com/ISN/2007/05/0122.html).
[10] Dueling with microwave ovens. http://www.wi-fiplanet.com/tutorials/article.php/3116531.
[11] M. Li, I. Koutsopoulos, R. Poovendran, Optimal jamming attacks and network defense policies in wireless sensor networks, IEEE INFOCOM, 2007.
[12] Click modular router, (http://read.cs.ucla.edu/click/).
[13] K. Gromov, D. Akos, S. Pullen, P. Enge, B. Parkinson, GIDL: generalized interference detection and localization system, ION GPS, Salt Lake City, UT, 2000.
[14] X. Liu, Signal detection and jammer localization in multipath channels for frequency hopping communications, DTIC, July 2005.
[15] S.D. Coutts, 3-D jammer localization using out-of-plane multipath, RADARCON, Dallas, Texas, USA, 1998.
[16] E.F. Velez, G.M. Amin, Improved jammer localization using multiple focussing, Advanced signal-processing algorithms, architectures, and implementations, 1990.
[17] A.M. Dean, Detection of active emitters using triangulation and trilateration techniques: theory and practice, in: AGARD, Radiolocation Techniques.
[18] I. Broustis, M. Faloutsos, S.V. Krisnamurthy, Overcoming the challenges of security in a mobile environment, IPCCC, Phoenix, AZ, 2006.
[19] Y. Cheng, Y. Chawathe, A. LaMarca, J. Krumm, Accuracy characterization for metropolitan-scale Wi-Fi localization, ACM MobiSys, Seattle, WA, 2005.

[20] A. Savvides, C.C. Han, M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, ACM MobiCom, Rome, IT, 2001.

[21] A. Subramanian, P. Deshpande, J. Gaojgao, S. Das, Drive-by localization of roadside Wi-Fi networks, IEEE INFOCOM, 2008.

[22] D. Han, D.G. Andersen, M. Kaminsky, K. Papagiannaki, S. Seshan, Access point localization using local signal strength gradient, PAM, 2009.

[23] Y. Chen, S. Chen, W. Trappe, Exploiting environmental properties for wireless localization and location aware applications, PerCom, 2008.

[24] W. Xu, W. Trappe, Y. Zhang, T. Wood, The feasibility of launching and detecting jamming attacks in wireless networks, ACM MOBIHOC, 2005.

[25] Z. Liu, H. Liu, W. Xu, Y. Chen, An error-minimizing framework for localizing jammers in wireless networks, Parallel Distr. Syst. IEEE Trans. 25 (2) (2014) 508–517.

[26] K. Pelechrinis, I. Koutsopoulos, I. Broustis, S.V. Krishnamurthy, Lightweight jammer localization in wireless networks: system design and implementation, IEEE GLOBECOM, 2009.

[27] H. Liu, W. Xu, Y. Chen, Z. Liu, Localizing jammers in wireless networks, in: Pervasive Computing and Communications, 2009 (PerCom 2009) IEEE International Conference on, 2009, pp. 1–6, doi:10.1109/PERCOM.2009.4912878.

[28] T. Cheng, P. Li, S. Zhu, An algorithm for jammer localization in wireless sensor networks, IEEE AINA, 2012.

[29] T. Cheng, P. Li, S. Zhu, Multi-jammer localization in wireless sensor networks, IEEE CIS, 2011.

[30] Z. Liu, H. Liu, W. Xu, Y. Chen, Exploiting jamming-caused neighbor changes for jammer localization, Parallel Distr. Syst. IEEE Trans. 23 (3) (2012) 547–555, doi:10.1109/TPDS.2011.154.

[31] Y. Cai, K. Pelechrinis, X. Wang, P. Krishnamurthy, Y. Mo, Joint reactive jammer detection and localization in an enterprise Wi-Fi network, Els. Comput. Netw. 57 (18) (2013) 3799–3811.

[32] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, ACM MOBICOM, 2004.

[33] J. Faruque, A. Helmy, Gradient-based routing in sensor networks, ACM MobiCom (poster session), 2003.

[34] R. Poor, Gradient routing in ad hoc networks, in: www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf.

[35] A.P. Ruhil, D.K. Lobiyal, I. Stojmenovic, Positioned based gradient routing in mobile ad hoc networks, ICDCIT, 2005.

[36] B. O'hara, A. Petrick, IEEE 802.11 Handbook, a Designer's Companion, second ed., IEEE Press, 1999. ISBN 0-73-814449-5, (http://www.amazon.com/IEEE-802-11-Handbook-Designers-Companion/dp/0738144495).

[37] K. Pelechrinis, G. Yan, S. Eidenbenz, S.V. Krisnamurthy, Detecting selfish exploitation of carrier sensing in 802.11 networks, IEEE INFOCOM, 2009.

[38] The UCR testbed., (http://networks.cs.ucr.edu/testbed/).

[39] K. Pelechrinis, I. Broustis, S.V. Krishnamurthy, C. Gkantsidis, Ares: an anti-jamming reinforcement system for 802.11 networks, ACM CoNEXT, 2009.

[40] S. Zvanovec, P. Pechac, M. Klepal, Wireless LAN networks design: site syrvey or propagation models? Radioengineering, Vol. 12, Dec. 2003.

[41] MIT Roofnet, (http://pdos.csail.mit.edu/roofnet).

[42] M. Li, I. Koutsopoulos, R. Poovendran, Optimal jamming attack strategies and network defense policies in wireless sensor networks, IEEE Trans. Mobile Comput. 9 (8) (2010) 1119–1133.