Lightweight Jammer Localization in Wireless Networks: System Design and Implementation

Konstantinos Pelechrinis*, Iordanis Koutsopoulos†, Ioannis Broustis*, Srikanth V. Krishnamurthy*
*University of California, Riverside †University of Thessaly
{*kpele, broustis, krish*}*@cs.ucr.edu* *jordan@inf.uth.gr*

Abstract—**Jamming attacks have become prevalent during the last few years, due to the shared nature and the open access to the wireless medium. Finding the location of a jamming device is of great importance for restoring normal network operations. After detecting the malicious node we want to find its position, in order for further security actions to be taken. Our goal in this paper is the design and implementation of a simple, lightweight and generic localization algorithm. Our scheme is based on the principles of the gradient descent minimization algorithm. The key observation is that the Packet Delivery Ratio (PDR) has lower values as we move closer to the jammer. Hence, the use of a gradient-based scheme, operating on the discrete plane of the network topology, can help locate the jamming device. The contributions of our work are the following: (a) We demonstrate, through analysis and experimentation, the way that the jamming effects propagate through the network in terms of the observed PDR. (b) We design a distributed, lightweight jammer localization system which does not require any modifications to the driver/firmware of commercial NICs. (c) We implement and evaluate our localization system on our 802.11 indoor testbed. An attractive and important feature of our system is that it does not rely on special hardware[1] .**

Index Terms—**Wireless Networks, Jamming, Gradient Descent, Location Discovery, Experimentation, Analysis.**

I. Introduction

The widespread proliferation of 802.11 wireless networks makes them an attractive target for saboteurs with jamming devices [1], [2]. Numerous jamming attacks have been reported in the recent past [3], [4], [5], [6]. A jamming device continuously emits electromagnetic energy on the medium. 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. Note that jamming effects may also occur due to accidental activation of devices that do not serve a malicious cause, such as microwave ovens, cordless phones [7], etc. Following the detection of the presence of an attacker [8], an algorithm is needed for localizing the jammer, so that further countermeasures can be taken by the network (such as deactivating the jamming device, as well as isolating the attacker, capturing, punishing or even destroying it).

In this work, we design and implement a simple, low-overhead algorithm for jammer localization. The main attribute of our algorithm 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 characteristic of the wireless medium: since the power of the jamming signal degrades with distance, farther transmitters do not sense strong jamming signals. In addition, the SNR requirement at such tranceivers is often satisfied. This cannot be concealed by the attacker. The transmitter is thereby able to send more packets, while the receiver can decode more of those, resulting in an increased PDR as we move away from the jammer.

Taking this property into account we design a decentralized localization algorithm based on the gradient descent minimization method. Our algorithm progresses in a distributed manner towards the proximity of the attacker by successive forwarding of PDR measurements to neighbors. In that sense, it is reminiscent of the iterative gradient descent algorithm for identifying the minimum of a real-valued function f. This algorithm moves from one point $\boldsymbol{a}$ of the function's domain S to another $\boldsymbol{b} \in S$. The point $\boldsymbol{b}$ is towards the opposite direction of the gradient of f at $\boldsymbol{a}$; this is the direction in which f exhibits the largest decrease with regards to its value at point $\boldsymbol{a}$. If the algorithm cannot proceed further, (at least) a local optimum is declared. Note that in our case, the domain set is the discrete locations of the nodes. Hence, our scheme can be viewed as a discretized version of a gradient descent algorithm.

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

- **Analytical and experimental assessments for the spatial characteristics of jamming effects in a network:** 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 in the different parts of the network (relative to the jammer's location). We experimentally validate the analytically computed expression on our testbed. We show that tranceivers further from the jammer exhibit lower (or no) degradation in terms of PDR as compared to tranceivers that are located closer to the jammer.
- **Design of a fully distributed 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. The main advantages of our approach (as compared to previously proposed localization approaches) are: (a) it

[1] This work was done partially with support from the US Army Research Office under the Multi-University Research Initiative (MURI) grants W911NF-07-1-0318 and the NSF NeTS:WN / Cyber trust grant 0721941.
I. Koutsopoulos acknowledges the support of the European Commission through the OPNEX project (FP7-ICT-224218) and the NoE NEWCOM++ (FP7-ICT-216715).

978-1-4244-4148-8/09/$25.00 ©2009

is simple, (b) it does not require any special hardware support, (c) it is distributed in nature, and (d) it can be integrated with higher layer functions, such as routing, to circumvent the jammer's location.

- **Implementation and evaluation of our scheme on our testbed:** We implement our scheme on our wireless testbed using the `Click modular router` [9]. We validate its performance via experimentation; our results show that a satisfactory performance is achieved by our localization strategy; we also identify and discuss scenarios where our scheme has a difficulty in localizing the jammer.

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 rest of the paper is organized as follows. Section II provides a brief description of related studies. Section III describes our analytical framework for quantifying the jamming effects on the PDR. Section IV provides a full description of our algorithm. We present our experimental set-up and evaluations in Section V. Section VI discusses issues related to our approach. Our conclusions form Section VII.

II. Related Studies

Signal processing localization techniques: Secure mobile device localization, and in particular jammer localization, has been studied in the literature during the past years. Various approaches have been proposed in order to locate the malicious device, such as the efforts in [10], [11], [12], [13], [14]. 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 make the wide deployment of such techniques rather infeasible in currently commercial wireless networks. A detailed description of various secure positioning systems, that exclusively operate at the PHY layer, can be found in [15].

Received Signal Strength (RSS) based localization techniques: In addition to the above schemes, various studies utilize RSS measurements to discover the location of wireless devices, and in particular the positions of access points (APs). Most of these techniques require measurements of the RSS at various positions (***wardriving***). Some well known approaches belonging to this category are the *(weighted) centroid* [16] and *trilateration* [17]. Both these techniques combine measurements of the RSS at various locations in order to infer the position of the AP. Subramanian *et al.* [18] 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 [19] 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 AP's position. In a slightly different context Chen *et al.* [20] combine environmental information gathered from sensor networks in order to perform localization. All the 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 [21]. However, the advantage of our approach over the RSS-based systems is that it can be executed online and in a fully distributed manner.

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.* [22] propose the use of a gradient based algorithm for the efficient forwarding of queries in sensor networks. Poor [23] 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 the cost, and only the nodes that have a smaller cost relay the packet. In a similar fashion, Ruhil *et al.* [24] forward the message to the neighbor node that is closer to the direction of the destination.

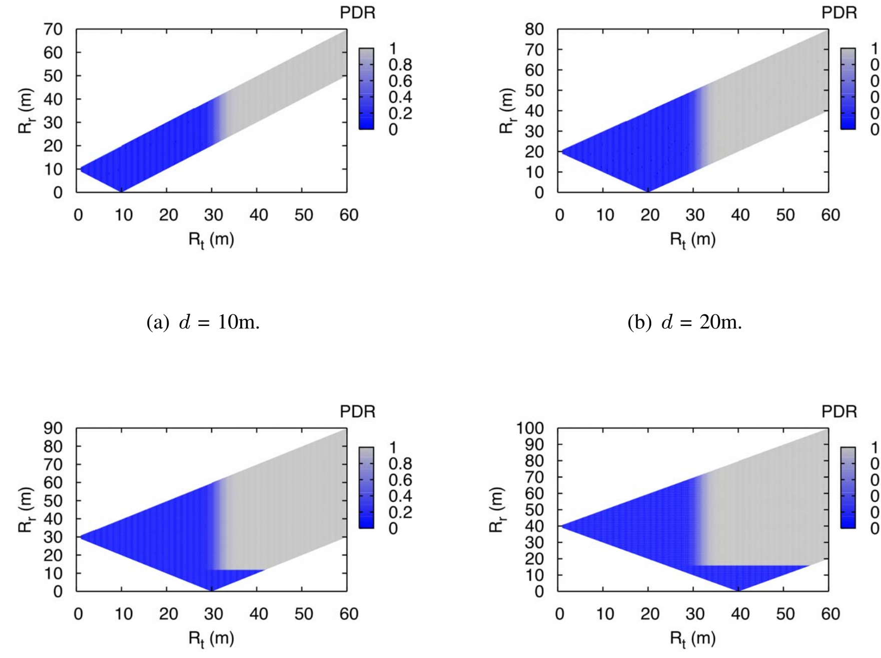
III. Jamming Effects on PDR

The presence of a jammer can affect the PDR on a link. In particular there are 3 possible ways that a (successful) packet transmission can be affected: **(i)** the transmitter (T_x) senses the medium busy due to jamming signals, **(ii)** the reception at the receiver (R_x) fails due to low SNR at its antenna because of the jamming signals and **(iii)** the reception of the MAC layer ACK packet fails due to low SNR at the T_x antenna. Since the above events are statistically independent, the PDR can be expressed in the following way:

$$PDR = P_{T_{x_{send-DATA}}} \cdot P_{R_{x_{receive-DATA}}} \cdot P_{T_{x_{receive-ACK}}}, \quad (1)$$

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 the SNR requirement at R_x is satisfied and $P_{T_{x_{receive-ACK}}}$ is the probability that the SNR requirement at T_x (for receiving the ACK) is satisfied too. Note that, we do not include the probability that the R_x is 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.

978-1-4244-4148-8/09/$25.00 ©2009



(a) d = 10m.

(b) d = 20m.

(c) d = 30m.

(d) d = 40m.

Fig. 1. Analytical results using Equation 1. Shorter links are more robust, while areas around the jamming device exhibit low PDR.

In order to calculate these probabilities we need to incorporate a signal propagation model. A widely used model [25] calculates the received power P_r at distance r relative to transmission power P to be:

$$P_r = \frac{P}{r^{\alpha}} \cdot Y, \tag{2}$$

where α is the path loss exponent and Y is a random variable that is log-normally distributed. Y captures the shadow fading effects and has a mean value of 1 and a standard deviation equal to the shadow fading variation which we can obtain from measurements. Using this propagation model the components of Equation 1 can be expressed as follows[2]:

$$P_{T_{x_{send-DATA}}} = P\{P_{JT} < CCA\} = Pr\{\frac{P_J}{r_T^a} \cdot Y < CCA\}$$
$$= P\{Y < \frac{CCA \cdot r_T^a}{P_J}\} = \frac{1}{2} + \frac{1}{2} \cdot erf\left(\frac{ln(\frac{CCA \cdot r_T^a}{P_J}) - \mu}{\sqrt{2} \cdot \sigma}\right) \tag{3}$$

$$P_{R_{x_{receive-DATA}}} = P\{SNR_{R_x} > u\} = P\{Y > \frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_R^a}}\}$$
$$= \quad \frac{1}{2} - \frac{1}{2} \cdot erf\left(\frac{ln(\frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_R^a}}) - \mu}{\sqrt{2} \cdot \sigma}\right) \tag{4}$$

$$P_{T_{x_{receive-ACK}}} = P\{SNR_{T_x} > u\} = P\{Y > \frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_T^a}}\}$$
$$= \quad \frac{1}{2} - \frac{1}{2} \cdot erf\left(\frac{ln\left(\frac{N \cdot u}{\frac{P}{d^a} - u \cdot \frac{P_J}{r_T^a}}\right) - \mu}{\sqrt{2} \cdot \sigma}\right). \tag{5}$$

In the above equations,

- 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,
- u is the SNR requirement for the particular rate used (in our case 6 Mbps), and
- $(\mu,\ \sigma)$ are the parameters of the log normal distribution (computed from the mean value and the standard deviation of the r.v. Y).

Substituting Equation (3)-(5) in (1) provides an expression for the PDR on a link as a function of r_T and r_R. Figure 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 [25]).

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

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

In Section V we present experimental results for cross validation of our analysis.

IV. Our Localization Algorithm

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

$$\vec{x}_{n+1} = \vec{x}_n - \gamma_n \cdot \nabla f(\vec{x}_n), \tag{6}$$

where $\nabla f(\vec{x_i})$ is the *gradient* of f. The gradient of f at point $\vec{x}$ is the direction of the maximum increase of the function at $\vec{x}$. The idea with this algorithm is that starting from a point, we greedily move towards the direction of the maximum decrease of the function at the neighborhood of this point (-$\nabla f(\vec{x}_n)$) using a step of γ_n at every iteration. After a series of iterations, the algorithm will converge to the minimum (at least a local one) of the function[3].

Our localization algorithm: As seen in the previous section, the PDR value decreases as we move to within the proximity of the jammer. Hence we can modify the above gradient descent method in order to localize the jammer. Function f is the PDR, while the next candidate points $\vec{x}_{n+1}$ are the neighbors of the node under consideration. Since we are moving in a discrete space, the PDR differential is the discrete approximation to the gradient's magnitude of the continuous function of PDR. In particular, every node will try to find its neighbor node with the largest decrease in PDR. Algorithm 1 presents a pseudocode for the algorithm executed at every node.

```
Data: Neighbors' PDR
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 1: Pseudocode for the Localization Scheme for node i.

In the above notation, PDR_i is the PDR of node i. However, PDR is related with 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[4]. Specifically:

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

where NS the set of neighbors of i, PDR_{im} is the PDR on link i-m and $|NS|$ is the cardinality of set NS, i.e. the number of neighbors of node i. Using this average value makes sense since one can expect the jammer to impact the PDR on all of a victim's associated links.

V. System Evaluation

Testbed description: Our testbed is deployed in the 3^{rd} 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 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 5-dBi omnidirectional antennae for every node. We use the Madwifing driver for our Atheros based cards and a proprietary version of the *ipw2200* driver/firmware of the *Intel-2915* card, which allows for tuning the CCA. More details on our testbed deployment can be found in [26].

Jammer implementation: For the purposes of our work we implement our own constant/deceptive jamming utility [21]. 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). In addition, having the jammer transmit broadcast packets allows the deferral of back-to-back transmissions for the minimum possible time[5] (i.e. $DIFS + min_{BackOff}$).

Validation of our analytical assessments: We perform experiments on our testbed in order to validate our analytical model presented in Section III. We activate the jamming nodes (one at a time) and we measure the PDR observed on various links on our testbed. We perform our experiments late at night in order to avoid interference from other 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 I shows a subset of our experimental results in comparison with the theoretical predictions. We observe that there is a good match between the measurements and the analysis (similar matches exist for the rest of the experiments as well). However, there are some discrepancies observed that

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

[4] We can also pick to use the minimum or maximum value of PDR of these links.

[5] Transmissions of MAC layer ACK packets are by default disabled for broadcast traffic.

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

TABLE I
OUR ANALYTICAL MODEL PREDICTS WELL THE EFFECT OF A JAMMER ON THE PDR OF A LINK.

can be attributed to the fact that the path loss exponent used in the model might not match exactly with the one of the real environment. In any case (qualitative) the effects of a jammer observed on our testbed are similar with the ones expected from our analysis.

System implementation details: We have implemented a prototype version of our localization scheme, using 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 [27] in order to retrieve the (average) PDR for every node (with regards to its neighbors). Our algorithm uses these values in order to perform the localization of the jammer. The dissemination of the PDR information takes places along the lines of the ETT [28] 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 $\tau = 100ms$ and $w = 1sec$). This implementation of our algorithm allows its integration with higher layer operations (and in particular routing) with no additional overhead. In the rest of this section we present some proof-of-concept experiments on our testbed and their interpretations.

Experimental results: Our main goal is to observe how our algorithm progressively percolates through the network topology. Every node independently runs the localization algorithm and makes local decisions with regards to the next node that it 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 complete percolation can be thought of as a "route discovery" propagation towards the jammer.

We illustrate the functionality of our algorithm with the following sample experiment. We activate one of our jamming devices on the testbed and run our localization algorithm on the rest nodes of our testbed in order to find the routes towards the jammer. Figure 2 shows the various paths towards the jammer that were reported from our algorithm for various starting points (nodes). In this experiment, the jammer is node 50 (see figure 2). The arrows represent the route towards the jammer that our algorithm finds, for various starting points. We extract the following interesting observations:

- Different starting points might end up into different end points.
- All successful localization iterations[6] end at nodes within one hop distance from the jammer (i.e. 20-35 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 can be trapped at a local minimum (e.g. path 11 → 36 → 27). This is an inherited feature of our scheme, from gradient descent minimization technique. In Section VI we discuss possible ways of overcoming this problem.

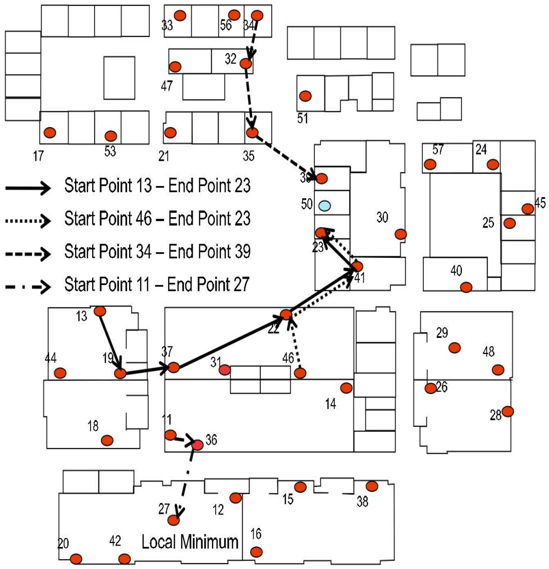

33
56
34
32
47
51
17
53
21
35
39
50
30
57
24
45
25
23
41
40
Start Point 13 – End Point 23
Start Point 46 – End Point 23
Start Point 34 – End Point 39
Start Point 11 – End Point 27
13
22
29
48
44
19
37
31
46
26
14
28
18
11
36
27
12
15
38
20
42
Local Minimum
16

Fig. 2. Paths to the jammer (node-50), for various starting points.

A more detailed observation at our experimental results reveals that when we start our search from nodes 13 and 46 we end up at node 23; the latter 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, we have to 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.

One side-effect from incorporating the gradient descent minimization method is that our scheme can be trapped to local minima. The performance of our proposed method is heavily dependent on the choice of the initial point/node. For the example in figure 2, our measurements reveal that if the localization procedure starts at node 11, it will result in a faulty 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. As examples: **(a)** 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 figure 2). **(b)** 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

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

978-1-4244-4148-8/09/$25.00 ©2009

general, a reason for getting to local minima is 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 that a node further from the jammer. We elaborate on this side-effect in the following section.

VI. Discussion and Future Directions

Sensitivity to local minima: A significant performance improvement can be attained with the elimination of the local minima sensitivity. Thus, intelligent ways that can help avoid local-minima regions are needed. One possible way could be to gather all the information with regards to PDR (collected by each legitimate node) and try to fuse these data. A majority rule could subsequently be used in order to decide upon the location of jammer(s). However, since dense-deployment regions might contain most of the nodes, the majority of the votes might still point to a local minimum. Therefore, what is required is a way to increase the confidence of the users' decisions with regards to the location of the jammer. In particular, nodes need to be able to effectively distinguish between jamming interference and heavy, legitimate interference. Xu *et al.* [21] propose the use of *consistency checks* in order to detect the presence of a jammer. In a nutshell, if the PDR experienced is small, while the received signal strength (RSS) is high, the presence of a jammer is inferred. However, a more thorough investigation would have to be performed, since high levels of (legitimate) interference can also lead to the same observations; low PDR values in conjunction with high RSS.

Future directions: Our work reveals the potential of gradient descent based localization algorithms. Using simple metrics, such as PDR, we can localize the jammer with no additional overhead. Currently, we have implemented a prototype version in order to demonstrate the potential of this approach. In the future, we plan to examine various modifications in order to decrease the sensitivity of our algorithm to local minima, thereby improving its performance. Identifying a good starting point for our algorithm is also critical for the efficacy of localization. Furthermore, we will examine the use of different definitions for PDR_i. Using the ratio of PDR_i under jamming to the one under benign conditions or the minimum/maximum value of PDR for the links of node i, is a possible approach. This might change the number of hops required in order to reach the jammer and/or the sensitivity to local minima; we plan to further examine the applicability of such approaches. Finally, we expect that under the presence of multiple jammers our algorithm will independently reach the proximity of each; we seek to investigate the performance of our scheme under these scenarios.

VII. Conclusions

We design a low-overhead, generic and distributed jammer localization algorithm. Our main observation that guides the construction of our system is related to the spatial effects of the jammer. In particular, links that are further from the adversary experience higher PDRs as compared to nodes that reside closer to the jamming device. We incorporate gradient descent methods in order for each network user to decide upon the immediate closer neighbor to the jammer node, from among its neighbors. The algorithm is greedy in nature; every node makes the locally optimal choice with regard to the direction towards the jammer. Our experiments indicate that our algorithm can indeed perform efficient and effective jammer localization.

Acknowledgment: We thank Dr. K. Papagiannaki from Intel Research, for providing the source code of the prototype driver.

References

[1] SESP jammers. http://www.sesp.com/.

[2] ISM Wide-band Jammers. http://69.6.206.229/e-commerce-solutions-catalog1.0.4.html.

[3] Jamming attack at hacker conference. http://findarticles.com/p/articles/mi_m0EIN/is_2005_August_2/ai_n14841565.

[4] Techworld news. http://www.techworld.com/mobility/news/index.cfm?newsid =10941.

[5] RF Jamming Attack. http://manageengine.adventnet.com/ products/wifi-manager/rfjamming-attack.html.

[6] ISA: Users fear wireless networks for control. http://lists.jammed.com/ISN/2007/05/0122.html.

[7] Dueling with Microwave Ovens. http://www.wi-fiplanet.com/tutorials/article.php/3116531.

[8] M.Li, I.Koutsopoulos, and R.Poovendran. Optimal Jamming Attacks and Network Defense Policies in Wireless Sensor Networks. In *IEEE INFOCOM*, 2007.

[9] Click Modular Router. http://read.cs.ucla.edu/click/.

[10] Konstantin Gromov, Dennis Akos, Sam Pullen, Per Enge, and Bradford Parkinson. GIDL: Generalized Interference Detection and Localization System. In *ION GPS, Salt Lake City, UT*, 2000.

[11] Liu and Xiangqian. Signal Detection and Jammer Localization in Multipath Channels for Frequency Hopping Communications. In *DTIC*, July 2005.

[12] S.D. Coutts. 3-D jammer localization using out-of-plane multipath. In *RADARCON, Dallas, Texas, USA*, 1998.

[13] E.F. Velez and G.M. Amin. Improved jammer localization using multiple focussing. In *Advanced signal-processing algorithms, architectures, and implementations*, 1990.

[14] A.M. Dean. Detection of active emitters using triangulation and trilateration techniques: Theory and practice. In *AGARD, Radiolocation Techniques*.

[15] I. Broustis, M. Faloutsos, and S.V. Krisnamurthy. Overcoming the Challenges of Security in a Mobile Environment. In *IPCCC, Phoenix, AZ*, 2006.

[16] Y. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale Wi-Fi localization. In *ACM MobiSys, Seattle, WA*, 2005.

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

[18] A. Subramanian, P. Deshpande, J. Gaojgao, and S. Das. Drive-by localization of roadside WiFi networks. In *IEEE INFOCOM*, 2008.

[19] D. Han, D.G. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan. Access Point Localization using Local Signal Strength Gradient. In *PAM*, 2009.

[20] Y. Chen, S. Chen, and W. Trappe. Exploiting Environmental Properties for Wireless Localization and Location Aware Applications. In *PerCom*, 2008.

[21] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *ACM MOBIHOC*, 2005.

[22] J. Faruque and A. Helmy. Gradient-Based Routing in Sensor Networks. In *ACM MobiCom (poster session)*, 2003.

[23] R. Poor. Gradient Routing in Ad Hoc Networks. In *www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf.*

[24] A.P. Ruhil, D.K. Lobiyal, and I. Stojmenovic. Positioned Based Gradient Routing in Mobile Ad Hoc Networks. In *ICDCIT*, 2005.

[25] K. Pelechrinis, G. Yan, S. Eidenbenz, and S.V. Krisnamurthy. Detecting Selfish Exploitation of Carrier Sensing in 802.11 Networks. In *IEEE INFOCOM*, 2009.

[26] The UCR testbed. http://networks.cs.ucr.edu/testbed/.

[27] MIT Roofnet. http://pdos.csail.mit.edu/roofnet.

[28] R. Draves, J. Padhye, and B.Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *ACM MOBICOM*, 2004.

978-1-4244-4148-8/09/$25.00 ©2009