International Journal of Pure and Applied Mathematics

Volume 56 No. 4 2009, 533-550

MITIGATING JAMMING ATTACKS IN WIRELESS SENSOR NETWORKS: AN ENERGY-EFFICIENT METHOD IN A MOBILE JAMMER ENVIRONMENT

Peter Soreanu[1], Zeev (Vladimir) Volkovich[2], Zeev Barzily[3], Mati Golani[4 §]

[1,2,3,4]Department of Sfotware Engineering
Ort Braude College
P.O. Box 78, 51, Snunit Str., Karmiel 21982, ISRAEL

[1]e-mail: speter@braude.ac.il
[2]e-mail: vlvokov@braude.ac.il
[3]e-mail: zbarzily@braude.ac.il
[4]e-mail: matig@braude.ac.il

Abstract: Wireless sensor networks (WSN) are practical implementations of distributed computing ad hoc wireless networks. WSN are typically empowered by scarce energy resources and limited computing power, they are mainly used for in situ data acquisition and monitoring of the deployment area. As such, they are susceptible to various forms of jamming, mostly but not exclusively, at the physical and data link layers. Of special interest are the moving jammers, which impose added strain on the WSN. In the present paper we developed a mitigation method that takes into consideration the jammer's behavior and accordingly adopts new routes, in order to maximize the WSN life. The model is an improvement of the well known *LEACH* energy–efficient routing protocol. In simulations, we have found that our protocol achieves a significant improvement of the WSN lifetime offering a proved improvement in the resilience of WSN against moving jamming attacks.

AMS Subject Classification: 90B10
Key Words: wireless sensor networks, sensing coverage, energy efficiency, circular sector model

Received: August 8, 2009

§Correspondence address: P.O.Box 78, Snunit 51, Karmiel, 21982, ISRAEL

1. Introduction

Assuring the security of WSN is a difficult task, mainly because of the hardware resource's limitations. The wireless medium increases the vulnerability, enabling adversaries to easily gain access and overtake communication channels used by the sensor nodes. The use of conventional security mechanisms allows defending against attacks by packet injection and by spoofing network-level control information.

WSN networks are quasi defenseless vis--vis radio interference attacks targeting their communication links. These jamming attacks exploit the shared nature of the wireless medium, preventing devices from communicating. This kind of physical layer-oriented attacks were quite extensively studied by military communications [20], [41], [24] and relatively successfully dealt with, mainly by the use of sophisticated spread spectrum techniques.

The advent of WSN poses new challenges for efficient anti-jamming methods. The sensor nodes themselves don't use spread spectrum communication methods like frequency hopping or direct sequence transmission. Instead, they use mostly specially developed low-speed protocols like IEEE 802.15.4 or ZigBee [39, 40], based on a carrier sensing approach to multiple access. This makes them prone to a simple and inexpensive jamming, where the adversary may simply disregard the medium access protocol and transmit continuously on the wireless channel [38]. In order to maximize the disruption of WSN activity, the jammer has to be mobile, using various moving algorithms [6]. Some algorithms were developed for locating the presence of the jammers, detecting the jammed sensor nodes and alerting their neighbors. Typical of them is the *JAM* protocol [30], which ultimately finds new routes in the non-jammed area.

Of special importance in finding ad hoc routes in WSN are the routing protocols that minimize the used energy, extending subsequently the life span of the WSN [21], [7]. The seminal and still typical representative of the energy-efficient class of routing protocols is *LEACH* [9], which was further improved by *PEGASIS* and other even more energy-efficient protocols [15], [8], [16].

Our protocol uses ideas and heuristics inspired from the jammer's possible movements, together with those provided by the *JAM* protocol, in order to develop energy-efficient rules. These rules are further used to modify *LEACH*, in order to significantly increase its resilience toward moving jammer attacks.

The paper is organized as follows. In Section 2 we present different jamming attack strategies that might be used against WSN. We analyze possible mitigation methods in Section 3. Section 4 is dedicated to the moving behavior of the

jammer. The original *JAM* protocol is summarized in Section 5, and *LEACH* is shortly described in Section 6. Our protocol, which is the main result of the research, is the object of Section 7. The simulation environment is described in Section 8, while Section 9 discusses the obtained results and concludes the paper.

2. Jamming Attack Strategies

Jamming is the process which allows the adversary to interfere with the ability of WSN node to communicate. It also includes interfering with the sensing abilities of the motes (nodes), but this is rarely used. It may be done by an external source or by a compromised node. The attack may be at all the layers of the protocol stack: Application, Transport, Network, Data Link (mostly Media Access Control-*MAC*) and Physical Interface (PHY) layer. The classical definition of jamming considers only the later disturbance. Some research also focuses on scenarios with attacks at more than one layer, i.e. Network and *MAC* simultaneously.

Attacks at the upper layers in WSN are quite similar to those of wired/wireless networks, with infrastructure or ad hoc alike. We concentrate on attacks at the *MAC* on physical layers. We present a short overview of such possible attacks that may be used against the WSN. From the wide variety of possible strategies, we highlight jammer models proved to be effective. The following models were described in [32]:

— The constant jammer emits continually a radio signal. This can be implemented either using a waveform generator to which continuously sends a radio signal, or by using a transmitter that sends out random bits, without respecting any *MAC* rules. Because these rules allow sending packets only if the channel is idle, the legitimate nodes cannot get hold of the channel to send their traffic.

— The deceptive jammer sends a constant stream of regular packets in the channel, without leaving any gap between them. This behavior induce legitimate nodes to think that the channel is busy. The nodes remain indefinitely in the receive state, even if they have packets waiting to be transmitted.

— The random jammer alternates between jamming and "sleeping". During the jamming phase, it may behave as a constant or as a deceptive jammer. The sleeping is introduced for energy conservation reasons, in order to maximize the jamming period. This policy might be very useful for jammers without

unlimited power supply.

— The reactive jammer maximizes its lifespan. It stays quiet when the channel is idle, and starts to transmit radio signals only when it senses activity on the communication channel. Such a jammer is much harder to detect.

Further models, based on packets inter-arrival times in three representative *MAC* protocols (S-*MAC*, L*MAC*, and B-*MAC*) were proposed in [11]. They are based on statistical analysis of the channel statistics.

— Periodic listening interval jammers are based on predicting the listen/sleep periods of the sensor nodes. Periodic listening interval jammers sleep most of the time and attack when the nodes are in listening period.

— Periodic control interval jammers calculate the control time slots in the data frames, and chose these time slots for attacks.

— Periodic data packet jammers listen to the channel in the control interval, but attacks only when the data transmission begins (with the decoding of Clear to Send-CTS message).

— Periodic cluster jammers use K-means clustering method to identify data frames vs. control frames, and launch the attack only when data frames are transmitted.

Four other classes of jamming attacks are proposed by Wood et al [31]:

— The interrupt jammer stays in passive listening mode and awakes for attack only when a hardware-triggered interrupt, detecting a preamble and a start of frame delimiter – SDF, occurs.

— The activity jammer prevents the detection of the preamble and SDF fields. It is initiated when the Received Signal Strength Indicator – RSSI is greater than a preset threshold.

— The scan jammer is used when frequency hopping is the WSN communication methods. It hops faster than normal network nodes, and attacks when a transmission is detected in a channel.

— The pulse jammer remains on the same channel and sends constantly or intermittently small packets, to block the communication.

Some special cases of attacks on the *MAC*/PHY layers were also researched:

— In [5] the jamming activity modifies the value of received signal strength, thus interfering with the RSSI-based nodes localization..

— Worm attacks, which cause nodes to crash or take control of the nodes. The use of a well-known vulnerability called buffer-overflow is described in [34].

3. Jamming Detecting and Defense Mechanisms

3.1. Jamming Detection Mechanisms

To be able to apply some kind of defense strategy, the WSN has to detect the presence of the jammer. This is a very difficult task, especially when the attackers mimic legitimate scenarios of bad connectivity, like traffic congestion or device failures. The main tools for discriminating between legitimate traffic and jamming are based on measuring signal strength, carrier sensing times, and packet delivery ratio.

These measurements are inherently limited in their scope. Signal strength alone cannot detect jamming, as it is unable to determine an exact threshold separating between legitimate and jamming situation. The value of carrier sensing time is significant as an indicator of constant or deceptive jamming situations. The packet delivery ratio is a more powerful tool, but it cannot differentiate between real jamming and poor connectivity. A marginally drop in the packet delivery ratio, even when caused by jamming, may insignificantly affect the traffic and does not need any preemptive or defensive action. Consequently, a combination of detection techniques was found as the most effective [32].

3.2. Jamming Defense Strategies

After detecting a jamming attack, a mote (node) in a WSN can adopt some defensive strategies, like the ones described below:

— Spatial retreat, i.e. physically evading the affected area. However, this solution is applicable only for mobile motes – and these constitute a minority of their population [37].

— Power control, i.e. improving the signal-to-noise ratio of the transmitted signal. This solution speeds up the depletion of the battery, and also increases the likelihood of collisions and interference in the network. Unfortunately, there is no success guarantee for this method [32]

— Code throttling is a good approach, but usually the motes have no suitable processing and radio facilities to implement [32]

— Powerful error correcting code and other security mechanisms lower the information rate transmitted, correspondingly increasing the power consumption and the likelihood of successful delivery of packets. Nevertheless, they

usually require too much computing power [28], [36], [17], [19]

— Spread spectrum techniques, such as frequency hopping or access codes changing are the most effective tools, but they are only available to devices that use one of these communication channel access methods [12].

— Wormhole-based anti-jamming techniques, using the concept called uncoordinated channel hopping. This is a new spread-spectrum techniques that does not rely on secret keys [4], [26].

— Techniques which identify trigger nodes – the nodes whose transmissions activate the reactive jammers [23]

— Robust *MAC* protocols, designed to protect the nodes from adaptive jamming, by optimizing the use of the wireless channel during attacks [3].

— Mitigation of jamming when the attack is directed to the direct link connecting the sink (base station) to the WSN [10].

— Ex-filtration of data received from the jammed area [2].

— Jamming area avoided by changing the route for transmitting data to sink [18]. Although this procedure will not better the coverage of jammed area, it will nevertheless increase the life of the WSN. Our proposed method belongs to this category.

4. Mobile Jammer Behavior

The study of mobility in WSN was used to develop new jamming techniques and corresponding anti-jamming countermeasures. The mobility in WSN was researched in various contexts:

— Mobile sensing nodes, able to change their position, in order to improve or repair the coverage of the sensed field [35].

— Mobile nodes which are used as data collectors/aggregators [6]. A special case is the mobile sink (gateway) [13]. They may extend the life of the WSN, and improve their chance to remain undetected.

— Mobile jammers, used to disturb the functions of a WSN [12], [14], [27].

Approaches to implement mobility in WSN, in order to maintain connectivity and to maximize the network lifetime, are summarized in [6]. Although the paper is dedicated to mobile sinks and mobile data collection, the main ideas can be used to also characterize the behavior of mobile jammers in a WSN environment. From the presented methods, we concentrate on two paradigms, as summarized below:

Random Mobility. The Data Collectors move randomly and collect data from sensors in their direct communication range [22]. Their performance is evaluated using a Markov model based on a two-dimensional random walk.

Predictable Mobility. Knowing the trajectory, the Data Collector's behavior can be modeled and predicted [25]. Consequently, more efficient defensive strategies can be adopted. A data collecting model, based on queuing theory, is developed and the energy-efficiency is analyzed.

The same modus operandi can be used to characterize the behavior of a mobile jammer, based on the assumption that, in order to achieve a perturbation at the communication channel level, both the jammer and the data collector device have to be in the wireless range of the motes. The difference is only in the action taken – jamming vs. data collection. The energy-efficient solutions proposed, modeled and evaluated in this research are based on the above mentioned similarity.

We further refer to the random mobility model as the Rendezvous Moving Jamming (RMJ) technique Its behavior could be described as follows: if a vehicle wants to move from a certain location to another without being noticed by the WSN, it calls the RMJ to come and meet it at a rendezvous point and hide its movements by escorting it to its destination [6].

The most common real life situation is the one with predictable mobility, and preprogrammed moving track. It is permanently on move and does not obey any sophisticated moving algorithm. We refer to it as the Simple Moving Jamming (SMJ) approach.

5. *JAM* Protocol: Jammed-Area Mapping Service for Sensor Networks

This protocol was the first in a series of denial-of-service attacks prevention strategies in WSN environments [30]. Further variants and enhancements are presented in [29]. When referring to denial-of-service attacks, attacks on every layer are included. The *JAM* protocol works with almost all kind if jamming, independent of the layer were the attack takes place.

When a node detects that it is jammed, it sends a JAMMED message to its neighbors, using some power management/carrier sense strategies to temporary override the jamming. Nodes which received jamming notifications group themselves, coalescing further to yield a map of the jammed region (also known as a jamming hole). Some such techniques are discussed in [33].

If the jamming attack stops, the nodes recover and inform their neighbors about the status change. In such a way, a dynamic map of the jamming hole is maintained. The changes of the map reflect the results of the moving jammer.

The main disadvantage of the protocol is the possibility of network partitioning. Also, the required computing overhead is relatively high during the mapping process. This shortcoming is compensated by the more powerful nodes used today.

6. *LEACH* Protocol: Low-Energy Adaptive Cluster Hierarchy

WSN can contain a very large number of independent and cheap nodes. Being battery-operated and lacking field-replace ability, their energy-efficiency is of primordial importance. The main energy components are related to communication (ε_c), processing (ε_p) and sensing (ε_s) functions. The total lifetime of the WSN is directly related to ε, representing the total energy consumption in equation (6.1):

$$\varepsilon = \varepsilon_c + \varepsilon_p + \varepsilon_s. \tag{6.1}$$

An energy-efficient algorithm is designed to minimize the communication costs, by choosing optimum routing paths for data transmission. Such an algorithm is *LEACH*, a self-organizing, adaptive routing protocol that uses randomization to distribute the energy load evenly among the sensors clusters, with the cluster-heads bearing the main energy load, being the higher-power radio station connecting to other cluster-heads and delivering the aggregated data to an external link. *LEACH* randomizes the successive selection of cluster-heads, so that the communication load is divided evenly between the sensors. It also implements local data fusion, further reducing the energy dissipation and enhancing the lifetime of the WSN system.

Nodes can elect themselves to cluster-heads. The decision to become a cluster-head depends on the amount of energy left at the node and how many times (if any) it acted as a cluster-head. In this way, nodes with more energy remaining will perform the energy-intensive functions of the network. Each node makes its decision about whether to be a cluster-head independently of the other nodes in the network and thus no extra negotiation is required to determine the cluster-heads.

To choose the next cluster-head, *LEACH* calculates the energy threshold $T(n)$ of each node, as defined by the deterministic cluster-head selection for-

mula, in equation 6.2.

$$T(n)_{new} = \frac{P}{1 - p\left(r \bmod \frac{1}{P}\right)} \times \left[\frac{E_{n\_current}}{E_{n-\max}} + \left(r_s div \frac{1}{P}\right)\left(1 - \frac{E_{n\_current}}{E_{n\_\max}}\right)\right]. \quad (6.2)$$

$E_{n\_current}$ is the current energy, E_{n-max} the initial energy of the node, and rs is the number of consecutive rounds in which a node has not been cluster-head. When rs reaches the value 1/P the threshold $T(n)$ new is reset to the value it had before the inclusion of the remaining energy into the threshold-equation. A higher threshold increases correspondingly the chance of the node to become cluster-head. Additionally, rs is reset to 0 when a node becomes cluster-head. Thus, we ensure that data is transmitted to the sink (gateway) as long as nodes are alive.

A stochastic approach to threshold calculation and cluster-head election can increase the lifetime of a *LEACH* wireless sensor network by 20% - 30 %

7. Proposed Method

Our jamming predictive protocol processes the output obtained by consecutive runs of the *JAM* protocol, and presents to *LEACH* a list of nodes that are likely to be jammed in the near future. When choosing a new cluster-head for data routing purposes, *LEACH* excludes these nodes. The assumption is that nodes located within the jamming hole can actually override the jamming signal. The protocol is able (with some limits) to predict, in real time, the possible direction of the movement. This is an important property of anti-jamming techniques for moving jammer attacks [1].

7.1. Prediction of a Moving Jammer

The *JAM* mapping algorithm enables us to locate jammers in the WSN field but does not supports tracking and predicting future locations of moving jammers. We believe that by adding prediction capabilities to an energy-aware routing algorithm, like the stochastic or deterministic node election versions of *LEACH*, will further improve the performances of a jammed WSN. We use the following two approaches for jammer's movement prediction:

— Vector prediction: works well with RMJ, which do not follow any pre-

defined path. It is also useful for jammers that follow a predefined path like the SMJ. This is due to the fact that the paths are unknown to the prediction algorithm in an early stage. It is possible to predict by a path, only after the algorithm mapped it.

— Path prediction: suited best for SMJ types. The algorithm uses a component called Path builder, which follow the movement of the jammers and builds paths by merging processed observations of the *JAM* algorithm. After a learning period the algorithm uses the stored paths to predict the possible next location of the moving jammers.

The input data of the prediction process is obtained using the *JAM* algorithm and is given in the form of discrete points. Every point in that input represents the speculated location of a jammer. Since we are dealing with moving jammers it is necessary to change the equality operator of the points – e.g., when comparing two points to see if they are actually the same point. Instead of using: $(a, b) = (c, d)$ iff $a = c$ and $b = d$, we divided the field, where the nodes are laid, into grid blocks. Since it is possible to attribute a point in a plain to a grid cube, we do so, e.g. $(a, b) = (c, d)$ if they both belong to the same grid cube. We use this mechanism in the path prediction algorithm, when we actually represent a path as a sequence of cubes and provide to the *LEACH* clustering algorithm the cubes we believe are about to be jammed.

7.2. Processing JAM Output

The application of the suggested protocol starts with the processing of the raw output data of the *JAM* algorithm. The protocol uses the jammed groups that where created by the BUILD messages of *JAM* [40]. The steps of this stage of the protocol are described below:

Step 1. The data is divided according to the associated consecutive time slices. Each time slice contains the various jammed groups center locations.

Step 2. Each time slice's groups are divided into clusters, by grouping together jammed close nodes. Each cluster center is an average of all the nodes that are members of that cluster.

Step 3. The grouped data is stored, for the next steps processing. Each time slice grouped data represents the predictive locations of active jammers at that time (multiple jammers allowed).

For a WSN containing n nodes, the time complexity of processing the *JAM* outputs is obviously $O(n)$.

7.3. Making Predictions

The following steps deal with the prediction part of the protocol. The goal is to predict, as accurately as possible, the path and the directions of the moving jammer. As described earlier, there are different procedures for paths creations for SMJ and RMJ situations.

Step 4. For each point from time slice $t - \Delta t$, match an appropriate point in time slice t. T matched points are associated with the same jammer.

Step 5. Compute a normalized vector for each set of matching points; the origin of the vector is the point belonging to time slice $t - \Delta t$.

Step 6. Create a prediction by multiplying the vector's length by a predefined value m, $m > 1$ (assuming a movement in a straight line, without sudden change of direction).

The processing of pairs of nodes for these steps leads a time complexity of $O(n2)$.

The next steps describe the path building (Steps 7 and 8) and the path predictor (Steps 8 and 9) part of the protocol.

Step 7. Represents all points that belong to the time interval it processes as cubes, and merges the points that are in the same time slot and the same cube as one cube entry. The obtained path will be expanded in Step 8.

Step 8. Match cubes to paths:

— if the current cube is the same or is a neighbor of a last cube in an active path then, add the cube to that path (provide it is not the same cube).

— if the currently processed cube could be added to the end of more than one path, close those paths and add the current cube to a new path.

— if there is no matching active path, creates a new one, and assigns the current cube to that new path.

Step 9. Examine backward the data and, using iteratively Steps 7 and 8, creates short sequences of three consecutive cubes.

Step 10. For all paths longer or equal to three cubes, search for them in all the paths. If a three or more cubes sequence is found, move on that path the corresponding number of steps.

The algorithm detects/predicts the paths of the moving jammer, enabling the building of a new communication route. This route will be used as many time slots as possible (depending, however, of the mobile jammer's actual movements). Obviously, the compromised/jammed nodes will not be part of the new

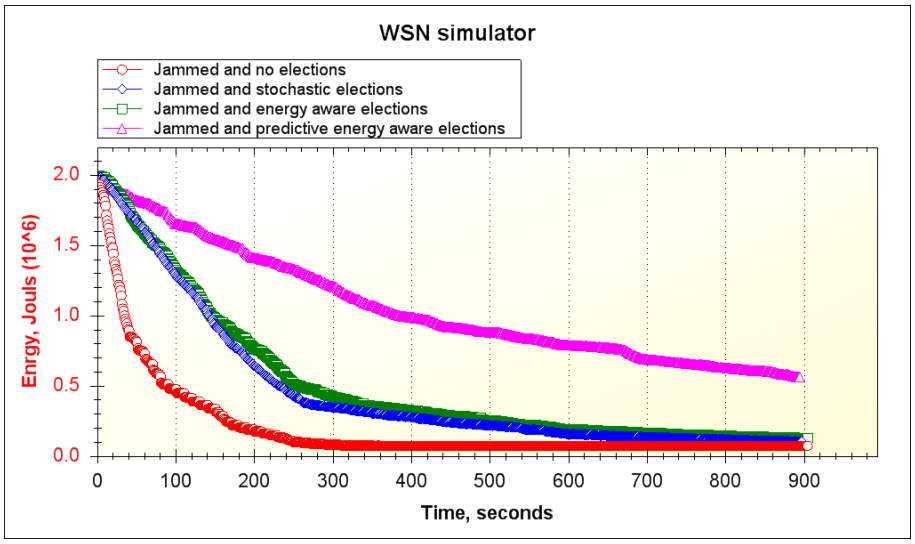

Figure 1: Total energy usage

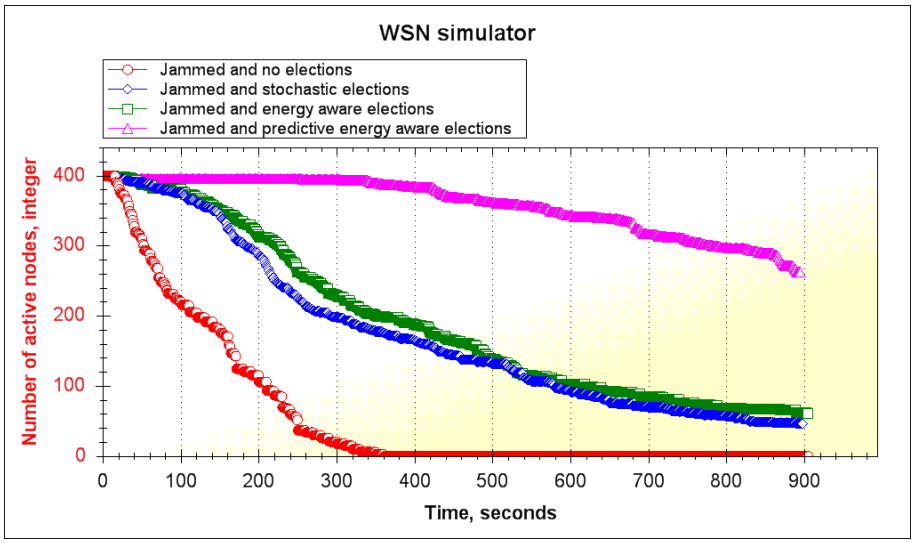

Figure 2: Number of active nodes

route. The nodes believed to be shortly jammed are also excluded from the cluster head selection process of *LEACH*.

The joint time complexity of both the path predictor and path builder steps

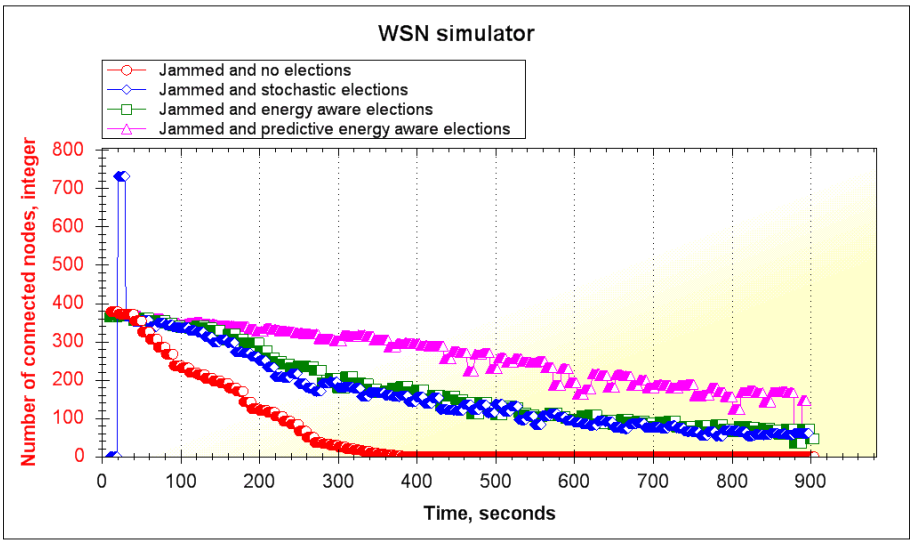

Figure 3: Number of active nodes

is $O(n3)$.

8. Simulation Environment and Obtained Results

The algorithm was implemented and tested by a dedicated WSN simulator, written in *C#*. In order to run the simulations, *JAM*, *LEACH*, and the moving jammers' behavior were also implemented. The field settings contained: size, nodes distance, number and type of moving jammers (SMJ, RMJ). The simulator also allowed for the setting of 11 prediction parameters (range, timing, active paths, etc), 8 *JAM* parameters (coalesce bounds, speed, delay, etc.) and 17 miscellaneous parameters to configure the running of the simulation (time and energy-related, jammers' properties, etc.). More controls were dedicated to the GUI and results' view.

Figures 1, 2, and 3 describe the results of the simulation, for the WSN and one mobile jammer. The 400 nodes were randomly placed on an 800 x 600 grid and various parameters were tested during successive runs.

As can be seen, our predictive jamming hole detection algorithm improved significantly the performance of the *LEACH* protocol: lowered the energy dissipation, increased the number of active and connected nodes. We obtained a

more than threefold increase of the WSN lifetime span. We think, that it may be deduced that improvements will also be found if our protocol will be compared with other *LEACH*-based routing protocols, or even non-energy-efficiency related ones.

9. Conclusions and Future Work

The paper covers technical issues related to the way WSN can identify and map active jammers. Both *JAM* and *LEACH* are proven solutions, but their integration with our algorithm achieved better results. We believe that the new combined algorithm produces a solid and robust solution that could help WSN to better cope with moving jammers and minimize total energy consumption, thus extending the longevity of the network.

Future related research will explore the fine-tuning of the algorithm for more moving jammers' behavior and the impact of the algorithm on other routing protocols. Also, the scenarios of multiple mobile jammers will be investigated.

Another predictive algorithm, concerning mobile sinks, was described in [13]. We intend to apply the ideas exposed there to a jamming environment, simulate it, and compare the results.

Acknowledgments

Finally, we would like to thank students D. Reznick and A. Tomaschoff, who wrote the WSN simulator for our research.

References

[1] N. Ahmed, S.S. Kanhere, S. Jha, The holes problem in wireless sensor networks: A survey, *ACM SIGMOBILE Mobile Computing and Communications Review*, **9**, No. 2 (April 2005), 4-18.

[2] G. Alnifie, R. Simon, A multi-channel defense against jamming attacks in wireless sensor networks, In: *Proc. of the 3-rd ACM Workshop on QoS and Security in Wireless and Mobile Networks (Q2SWinet 2007)*, Chania, Greece (October 2007), 95-104.

[3] B. Awebuch, A. Richa, C. Scheideler, A jamming-resistant *MAC* protocol for single-hop wireless networks, In: *Proc. of the 27-th ACM Symp. on Principles of Distributed Computing*, Toronto, Canada (August 2008), 45-54.

[4] M. Cagalj, S. Capkun, J.-P. Hubaux, Wormhole-based antijamming techniques in sensor networks, *IEEE Transactions on Mobile Computing*, **6**, No. 1 (January 2007), 100-114.

[5] Y. Chen, K. Kleisouris, X. Li, W. Trappe, R.P. Martin, A security and robustness performance analysis of localization algorithms to signal strength attacks, *ACM Transactions on Sensor Networks*, **5**, No. 1, article 2 (February 2009), 37pp.

[6] E. Ekici, Y. Gu, D. Bozdag, Mobility-based communication in wireless sensor networks, *IEEE Communications Magazine*, **44**, No. 6, 56-62.

[7] L. Gan, J. Liu, X. Jin, Agent-Based, Energy efficient routing in sensor networks, In: *Proceedings of the 3-rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, New York (July 2004), 472-479.

[8] M.J. Handy, M. Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head detection, In: *4-th IEEE Workshop on Mobile and Wireless Communications Network*, Stockholm (September 2002), 368-372.

[9] W.B. Heinzelman, *Application-Specific Protocol Architectures for Wireless Networks*, Ph.D. Thesis, Massachusetts Institute of Technology (June 2000).

[10] S. Khattab, Mosse evasion for mitigating base-station jamming in sensor networks, In: *Proceedings of the 20-th International Parallel and Distributed Processing Symposium (IPDPS)*, Rhodes Island, Greece (April 2006).

[11] Y.W. Law, L. van Hoesel, J. Doumen, P. Hartel, P. Havinga, M. Palaniswami, Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols, *ACM Transactions on Sensor Networks*, **5**, No. 1, Article 6 (February 2009), 38 pp.

[12] L. Lazos, S. Liu, M. Krunz, Mitigating control-channel jamming attacks in multi-channel ad hoc networks, In: *Proc of the 2-nd ACM Conference*

on Wireless Network Security (WiSe 2009), Zurich, Switzerland (March 2009), 169-180.

[13] E. Lee et al, A predictable mobility-based communication paradigm for wireless sensor networks, *Proc. of Asia-Pacific Conf. on Communications (APCC 2007)*, Bangkok (October 2007), 373-346.

[14] B. Li, L. Batten, Using mobile agents to recover from node and database compromise in path-based DoS attacks in wireless sensor networks, *Journal of Network and Computer Architecture*, **32**, No. 2 (March 2009), 377-387.

[15] S. Lindsey, C.S. Raghavendra, *PEGASIS*: Power efficient gathering in sensor information systems, In: *2002 IEEE Aerospace Conference Proceedings*, Big Sky, MT, Volume 3 (March 2002), 1125-1130.

[16] V. Loscri, G. Morabito, S. Marano, A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-*LEACH*), In: *IEEE Vehicular Technology Conference (VTC-2005-Fall)*, Dallas, TX, Volume 3 (September 2005), 1809-1813.

[17] I. Martinovic, P. Pichota, J.B. Schmitt, Jamming for good: A fresh approach to authentic communication in WSNs, In: *Proc of the 2-nd ACM Conference on Wireless Network Security (WiSec 2009)*, Zurich, Switzerland (March 2009), 161-168.

[18] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, G. Pantziou, JAID: An Algorithm for Data Fusion and Jamming Avoidance on Distributed Sensor Networks, *Pervasive and Mobile Computing*, Elsevier, **5**, No. 2 (April 2009), 135-147.

[19] R. Pietro Di, L.V. Mancini, A. Mei, Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks, *Wireless Networks*, Kluwer Academic Publishers, **12**, No. 6 (November 2006), 709-721.

[20] R.A. Poisel, *Modern Communications Jamming Principles and Techniques*, Artech House (2003).

[21] C. Schurgers, M.B. Srivastava, Energy efficient routing in wireless sensor networks, In: *Proceedings of Military Communications Conference (MILCOM01)*, Vienna, VA (August 2001), 357-361.

[22] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data mules: Modeling a three-tier architecture for sparse sensor networks, *Ad Hoc Networks*, Elsevier, **1**, No-s: 2-3 (September 2003), 215-233.

[23] I. Shin, Y. Shen, Y. Xuan, M.T. Thai, T. Znati, Reactive jamming attacks in multi-radio wireless sensor networks: An efficient mitigating measure by identifying trigger nodes, In: *Proc. of 2-nd ACM Int'l Workshop on Foundations of Wireless Ad Hoc and Sensor Networking (FPWANC 2009)*, New Orleans, LA (May 2009), 87-96,

[24] B. Sklar, *Digital Communications: Fundamentals and Applications*, Second Edition, Prentice Hall (2001).

[25] A.A. Somasundra, E. Kansal, D. Estrin, M.B. Srivastava, Controllably mobile infrastructure for low energy embedded networks, *IEEE Transactions in Mobile Computing*, **5**, No. 8 (August 2006), 958-973.

[26] M. Strasser, C. Popper, S. Capkun, Efficient uncoordinated FHSS anti-jamming communication, In: *Proceedings of the 10-th ACM Int'l Symposium on Mobile Ad Hoc Networks and Computers (MobiHoc '09)*, New Orleans, LA (November 2005), 207-217.

[27] H.-M. Sung, S.-P. Hsu, C.-M. Che, Mobile jamming attack and its countermeasure in wireless sensor networks, In: *Proc. of the 21-st Int'l Conf. on Advanced Inf. Networking (AINAW 2007)*, Washington, DC, Volume 1 (May 2007), 457-462.

[28] S. Tripathy, S. Nandi, Defense against outside attacks in wireless sensor networks, *Computer Communications*, Butterworth-Heinemann, **31**, No. 4 (March 2008), 818-826.

[29] A.D. Wood, J.A. Stankovic, Security of distributed, ubiquitous, and embedded computing platforms, In: *Wiley Handbook of Science and Technology for Homeland Security* (Ed. J.G. Voeller), John Wiley and Sons, Hoboken, NJ (2009).

[30] A.D. Wood, J.A. Stankovic, S.H. Son, *JAM*: A jamming-area mapping service for sensor networks, In: *Proceedings of the 24-th IEEE Real-Time Systems Symposium (RTSS-2003)*, Cancun, Mexico, IEEE December (2003), 286-297.

[31] A.D. Wood, J.A. Stankovic, G. Zhou, DEEJAM: Defeating energy-efficient jamming, In: *IEEE 802.15.4-based Wireless Networks, in Proc. of 4-th An-*

nual IEEE ComSoc Conf. on Sensor, Mesh, and AdHoc Communications and Networks (SECON 2007), San Diego, CA (June 2007), 60-69.

[32] W. Xu, W. Trappe, S.H. Son, The feasibility of launching and detecting jamming attacks in wireless networks, In: *Proceedings of the 6th ACM Int'l Symposium on Mobile Ad Hoc Networks and Computers (MobiHoc '05)*, Baltimore, MD (November 2005), 46-57.

[33] W. Xu, W. Trappe, Y. Zhang, Anti-jamming timing channels for wireless networks, In: *Proc. of the 1-st ACM Conference on Wireless Network Security (WiSec 2008)*, Alexandria, VA (March 2008), 203-213.

[34] Y. Yang, S. Zhu, G. Cao, Improving sensor network immunity under worm attacks: A software diversity approach, In: *Proc. of the 9-th Int'l ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc 2008)*, Hong Kong (May 2008), 149-158.

[35] S. Yoon, O. Soysal, M. Demirbas, C. Qiao, Coordinated locomotion of mobile sensor networks, In: *Proc. of 5-th Annual IEEE ComSoc Conf. on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2008)*, San Diego, CA (June 2008), 126-134.

[36] W. Yu, W. Trappe, Y. Zhang, Defending wireless sensor networks from radio interference through channel adaption, *ACM Transactions on Sensor Networks*, **4**, No. 4, Article 18 (August 2008), 33 pp.

[37] X. Yu, T. Wood, W. Trappe, Y. Zhang, Channel surfing and spatial retreats: Defenses against wireless denial of service, In: *Proc of the 3-rd ACM Workshop on Wireless Security (WiSec 2004)*, Philadelphia, PA (October 2004), 80-89.

[38] AusCERT Report AA-2004.02, Denial of Service Vulnerability in IEEE 802.11 Wireless Devices (May 2004), http://www.auscert.org.

[39] Institute of Electrical and Electronics Engineers, *Inc., IEEE Std.*, 802.15.4-2007, Wireless Medium Access Control (*MAC*) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), New York, IEEE Press (2007).

[40] *ZigBee Alliance, ZigBee Specifications*, version 2007/PRO.

[41] Pentagon Report A319383, *Digital Communications Jamming*, Storming Media (2000).