Reactive Jamming Attacks in Multi-Radio Wireless Sensor Networks: An Efficient Mitigating Measure by Identifying Trigger Nodes

Incheol Shin, Yilin Shen, Ying Xuan, and My T. Thai
Dept. of Comp. and Info. Sci. and Eng.
University of Florida
Gainesville, FL 32611
{ishin, yshen, yxuan, mythai}@cise.ufl.edu

Taieb Znati
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15215
znati@cs.pitt.edu

ABSTRACT

There exist many studies against reactive jamming attacks, however, these methods, i.e. frequency hopping or channel surfing, require excessive computational capabilities on wireless devices which are serious side effects in wireless sensor networks. To avoid the problems in existing methods, we propose a novel approach against reactive jamming attacks by identifying the trigger nodes, whose transmissions activate any reactive jammers. The identification of these trigger nodes can help us (i) carefully design a better routing protocol by switching these nodes into only receivers to avoid activating jammers and (ii) locate the jammers based on the trigger nodes, thus providing an alternative mechanism against reactive jamming attacks. In this paper, we provide an efficient method to identify the trigger nodes by utilizing the group testing techniques and minimum collection of disjoint disk covers to reduce the message and computational overhead. The theoretical analysis and experimental results show that our solution performs extremely well in terms of time and message complexities, which in turn provides a good approach to defend reactive jamming attacks.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

General Terms

Algorithms, Security, Reliability, Theory

Keywords

Denial of Service, Jamming, Security, Group Testing

FOWANC'09, May 18, 2009, New Orleans, Louisiana, USA.

1. INTRODUCTION

The jamming attack is one of the most critical security issues in wireless networks, which disseminates out sufficient adversarial signals into the radio frequencies used by normal sensor nodes, without following any legitimate protocols. Since the jammer interferes with radio reception by producing noise, it could decrease the probability of successful broadcasting in the wireless communication. The jammers do not need to explore lots of internal information of the network components, so this light weight attack is easy to launch and favored by attackers. Furthermore, in reactive jamming attacks [20], the jammers keep idle until being triggered by messages disseminated within their transmission ranges, thereby further reducing the jammers' operation overhead and making it hard to detect, thus this intelligent attack can be utilized by malicious users in more real-world scenarios. In this paper, an efficient defense mechanism against this reactive jamming attack will be presented.

There are many existing studies against jamming attacks [18, 17, 12, 4, 13, 8, 11, 16, 5, 19, 10, 1, 14]. However, the high computational overhead of these methods badly reduces the effect in resource-limited network environments, such as wireless sensor networks (WSNs). For example, in the channel surfing [18, 17] and frequency hopping [12] methods, the transmission frequency or channels are changed to a range where there is no interference from the adversary. However, it is not suitable for WSNs, especially in multi-channels WSNs, since the sensors have to scan all the channels to detect the jamming attacks and hop in new frequencies all the time, even in the middle of communication. Due to the fact that most of the sensor nodes have half-duplex transceiver on it, scanning the channels during the transmission causes communication stalls for checking the availability of the current channel. Frequent communication stalls result in longer transmission duration and more energy consumption. Consequently, these methods cannot avoid high overhead and resource consumption.

More importantly, since the reactive jammers would keep track of the frequency shifting sequences or channel selecting mechanisms, excessive exposure of these methods against the reactive jammers might be vulnerable to achieve effective communication performances among the victim nodes.

Recently, Wood *et. al* have proposed a new method to locate possible jammed regions and de-route all the messages going through this area [1]. However, this approach can create unnecessarily big jammed region in the case of reactive jamming attack and result in isolated networks in the worst case. Additionally, the message overhead is relatively high during mapping processing.

To overcome the mentioned shortcomings, we introduce a novel method against reactive jamming attack in multi-channels WSNs by identifying the trigger nodes. The identification of *trigger* nodes can have several benefits against reactive jamming attacks. First of all, we can construct a routing algorithm in which the triggers are only the receivers, thus avoiding activating the jammers and minimizing the effect of jamming attacks. This can overcome the limitations of the channel surfing and frequency hopping. In the case of a trigger node needs to send a message, we may still utilize the use of channel surfing, however, only a few nodes may require this operation, thus greatly reducing the computational costs required by existing methods. In addition, the identification of triggers nodes will not create an unnecessary big jammed region as in [1].

Further more, after the identification of *trigger* nodes, victim nodes would be scheduled to transmit messages in order to minimize the damage from the attackers by keeping silent during the transmission of the *trigger* nodes, which prevents to waste communication messages against the jammers. Finally, we may use the information of trigger nodes to later locate the jammers , which will not be addressed here.

Finally, we may use the information of trigger nodes to later locate the jammers.

In this paper, we will focus on the identification of trigger nodes, which can provide a general framework to build an efficient countermeasure for reactive jamming attacks in multi-channels WSNs. So the key problem is how to efficiently identify these trigger nodes. By utilizing traditional group testing (GT) theory [7, 6] coupling with disk cover based grouping and clique based clustering, our proposed solution can identify all the trigger nodes with low overhead in terms of running time, computation and message complexity.

The key idea of GT is to pool the items into several groups, test each group simultaneously, and identify the defective item by analyzing the test outcomes, instead of testing each item individually. In the context of identifying trigger nodes in our solution, all the *victim* nodes which are within the transmission range of the jammers, are first identified using some tree-based broadcasting and pooled into multiple groups. By letting all the victim nodes broadcast to trigger possible nearby jammers, one victim node in each group is selected to hear the noises and generate a testing result. A group with no noise heard is called negative group, which means all the victim nodes within this group cannot trigger any jammers. On the contrary, a group hearing noises is called positive group, which means at least one of the victim nodes is a trigger node, thus requires further tests on this group until the trigger nodes are identified. Therefore, our detection based on GT is of multiple-fold: 1) how to efficiently find out all victim nodes; 2) how to properly pool the victim nodes into multiple groups so that different groups cannot interfere each others test outcome. (i.e., a victim node trigger the activation of a jammer, whose noise reaches another group's hearing nodes, in which case, the latter group will have positive outcome even if it has no trigger node); 3) how to collect and decode the outcomes.

Our main contributions of this paper are as follows.

- This is the first work to introduce the concept of trigger nodes in reactive jamming attacks.
- By utilizing GT theory, disk cover based grouping and clique based clustering , the proposed protocol can accurately identify the trigger nodes among the victim nodes with low message and computational complexity. This is critical and suitable for WSNs since they have only limited resources and energy conservation.
- Detailed theoretical analysis and simulation results show the novel performance of this protocol in terms of time and message complexity.

The remainder of this paper is structured as follows: Section 2 provides the problem definition, network model as well as some assumptions and notations. Preliminaries for maximum clique problem and GT theory are introduced in Section 3. We then provide our solution *Identifying Trigger Nodes* (ITN) and its performance analysis in Section 4, meanwhile present a routing protocol *TNLT* based on the identified trigger nodes in Section 5. The simulation setups and results are included in Section 6. Related works are briefly presented in Section 7 and finally Section 8 concludes our paper.

2. NETWORK MODEL AND PROBLEM DEFINITION

The WSN in our problem consists of N sensor nodes, each having the same transmission range r and one base station BS with the transmission range $\rho = \beta r$ where $\beta > 1$. Up to $J \ll N$ static jammer nodes, whose transmission ranges are uniformly $R = \alpha r$ where $\alpha > 1$, are deployed within the network. However, their positions cannot be known beforehand, except that all jammers are assumed to be sparsely deployed so that they can jam as large area as possible. Each sensor node or jammer node is equipped with k channels and m radios ($m < k$).

We model the considered network as a connected graph $G(V, E)$ where V is a set of N nodes and $E = \{(u, v) | \delta(u, v) \leq r,\ u, v \in V\}$ representing communication links between nodes. Any sensor nodes whose broadcasting can trigger some jammers are called *trigger nodes*, while any sensor nodes whose communications are interfered by jammers are called *victim nodes*. Therefore, any node v is a victim node if $\delta(J, v) \leq R$ for some jammer node J, whilst w is a trigger node if $\delta(J, w) \leq r$. **Note that trigger nodes are also victim nodes** as the noise range (transmission range of jammers) R is larger than sensor transmission range.

Since each sensor node has the same transmission range r and only the *neighbor* nodes within r can receive its message, the graph $G(V, E)$ is a *Unit Disk Graph (UDG)*.

The objective of the problem is to find out all the trigger nodes within minimum time and message complexity. After the identification, a new routing path would be constructed to avoid activating any reactive jammers.

Some notations used throughout this paper are depicted in Table 1.

Table 1: Notations

Symbol	Meaning
r	The transmission range of each sensor
R	The noise range of the jammers
ρ	The transmission range of the base station
V	The set of nodes in WSN
N	The number of nodes in WSN
W	The set of victim nodes in WSN
W_i	The set of left victim nodes in WSN after cover i
n	The number of victim nodes in WSN
$\mathcal{N}_i$	The number of victim nodes covered in cover i
n_i	The number of victim nodes before cover i
n_{ij}	The number of victim nodes in group j in cover i
U	The set of trigger nodes in WSN
d	The number of trigger nodes
d_{ij}	The number of trigger nodes in group j in cover i
k	The number of channels in WSN
m	The number of radios in WSN
$\Delta(G)$	The maximum node degree of graph G
$\kappa(D)$	The number of nodes disk D covers
$\delta(u,v)$	The distance between two nodes u and v
$H(\delta)$	Unit Disk Graph H with disk radius δ
t_i	The total number of testing rounds in cover i
$\mathcal{C}$	The total number of testing covers
$\mathcal{T}$	The total testing time

3. PRELIMINARIES

In this section, we introduce some preliminaries on MAXIMUM CLIQUE PROBLEM and NON-ADAPTIVE GROUP TESTING, based on which we discuss how to apply them to our problem.

3.1 Maximum Clique Problem

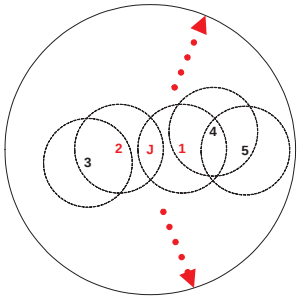
The Maximum Clique Problem is defined as follows. Given an arbitrary undirected graph $G(V, E)$, a subgraph $G'(V', E')$ $(V' \in V)$ is a clique if all its vertices $v' \in V'$ are pairwise adjacent. The maximum clique is a clique with $\max |V'|$. The maximum clique problem is also one of the first problems shown to be **NP**-complete [2].

So far, the best polynomial-time approximation algorithm for the maximum clique problem was developed by Boppana and Halldorsson [2], and achieves an approximation ratio of $n^{(1-o(1))}$. In [2], Hastad shows that this is actually the best we can achieve and it cannot be approximated within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$. There are some other results in the literature concerning the approximation of the maximum clique problem on arbitrary or special graphs. [2, 3, 9]

In this paper, the maximum clique problem is applied to find out the upper bound of the number of trigger nodes based on the number of reactive jammers. Since an active jammer can only be triggered by the nodes within the certain distance, we can construct a unit disk graph of all nodes with the radius twice the distance to find the upper bound of the number of trigger nodes.

3.2 Non-Adaptive Group Testing

Non-adaptive Group Testing(GT) [7, 6] methods are to minimize the testing period by sophisticatedly grouping and testing the items in pools simultaneously, instead of individually testing them. The way of grouping is based on an 0-1 matrix $M_{t\times n}$ where the matrix rows represent the testing group and the each column refers an item. $M[i, j] = 1$ implies that the j^{th} item participates in the i^{th} testing group, and the number of testing is the number of rows. The result of each group is represented as an outcome vector with the matrix row size t — 0 with a negative group result and 1 with a positive result. To achieve the minimum testing length for non-adaptive GT, M is required to be d-disjunct [7], where the union of any d columns does not contain any other column.



Nodes					Testing
1	2	3	4	5	Outcome
1	0	1	1	0	1
0	1	0	0	1	1
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	1	1	1

Figure 1: Group Testing

Based on the properties of d-disjunctness, the decoding algorithm becomes very simple. We just need to remove all the items appeared in any negative pools and the remaining item are positive [7]. In this way, only $O(1)$ testing rounds and $O(tn)$ decoding time are needed.

3.3 Identification Procedure

We resorts to clique-based clustering and non-adaptive GT to identify all trigger nodes out of all the victim nodes, based on a new routing protocol we propose to avoid activating the jammers in Section 4.

The basic idea of this GT-based identification is as follows: Assume we have 1 jammer, n victim nodes and d trigger nodes in the adversarial signal transmission range of the jammer. Our objective is to identify all the d trigger nodes among n victim nodes in the minimum time latency.

To identify all these trigger nodes, we divide the victim nodes into groups and test based on a d-disjunct matrix in each group. The nodes in different groups broadcast in an orthogonal way (i.e. on different channels) so that the testing result will not interfere with each other. Consider the nodes in one group broadcast on the same channel, the jammer is supposed to be activated and broadcast noise on this channel if some nodes in this group are trigger nodes. Then the collectors will transmit the results to the base station, which will union all results in the same channel to get the result for each group. During the communication with base station, the collectors are required to perform channel surfing method for successful delivery of the outcomes. By decoding these results, the trigger nodes will be identified.

Consider the matrix $M_{t\times n}$ in Fig. 1, the victim nodes can be mapped into the columns and channels into rows in M, where $M[i, j] = 1$ *iff* the victim node j broadcast on channel i. As for the test outcome column V, $V[i] = 1$ *iff* the collector of row i sense some noise generated from jammers and $V[i] = 0$ otherwise. It means that no victim nodes are jammed on channel i, thus there is no trigger node in the

test. Suppose we have m radios ($m < k$), if the number of tests t is larger than the number of radios m, the testing will finish in $\lceil t/m \rceil$ rounds.

To utilize GT, we need to solve the two most challenging problems: (1) How to group the nodes to avoid interference between the results among groups so as to test these groups simultaneously. (Any two nodes u and v are called interference free nodes if any jammers triggered by either node cannot jam the other node). (2) How to accurately estimate the value of d which is the upper bound of the number of trigger nodes. Since d determines the number of tests, the tighter d is, the better time and message complexities we can obtain. We will present our proposed algorithms to solve these problems in next section.

4. THE PROPOSED ALGORITHMS

In this section, we devise an ***Identifying Trigger Nodes (ITN)*** algorithm to identify all trigger nodes in WSNs so that reactive jamming can be avoided when these trigger nodes do not transmit messages.

The basic ideas of ITN is as follows: We first detect all victim nodes from all nodes in WSNs by using the *DVN-BFST* algorithm — ***Detection of Victim Nodes based on adapted Breadth-First Search tree***. Then we test these victims to identify the trigger nodes by two steps: 1) We use the *GVN-MCDDC* algorithm — ***Group Victim Nodes Based on Minimum Collection of Disjoint Disk Cover*** to group as many as victim nodes without interference with each other in each cover. Each cover includes a set of disjoint disk where the center of this disk will act as a collector. Each of the disjoint disks can be tested simultaneously. 2) For a set of victims in each disjoint disk, we use the *DTN-NCGT* algorithm — ***Detection of Trigger Nodes based on Non-adaptive Combinatorial GT*** to detect all trigger nodes within these victim nodes. We continue covering and testing victim nodes until all victim nodes are tested.

4.1 The *DVN-BFS* Algorithm (Algorithm 1)

The *DVN-BFS* algorithm based on *adapted Breadth-First Search tree* is devised to efficiently identify all $|W|$ victim nodes within n nodes in a given network with limited ACKs. The basic idea is as follows: We define a broadcast message n-dimension mg where each node $v \in V$ has a entry in this message, with $|V| = n$. This mg is broadcasted as a message from the base station to all n nodes along a *BFS* tree. Once a node receives this message, it will set its corresponding entry in mg to 1 if the node senses any one of the channels is jammed and hop to another normal channel to transmit the broadcast message. The base station will receive a collection of messages MG from all leaf nodes. In this case, the number of ACKs is only the number of leaves on this *BFS* tree. At last, the base station will identify all victim nodes by calculating the union MG of all broadcast messages mg (i.e., the entry i is 1 *iff* there is one message $mg \in MG$ with the entry i equal to 1, where $1 \leq i \leq n$.) By constructing a diagonal matrix $M = diag(1, 2, \ldots, n)$, $MG \cdot M$ is the set of victim nodes.

4.2 The *GVN-MCDDC* Algorithm

In this section, we devise ***GVN-MCDDC*** algorithm *(Group Victim Nodes Based on Minimum Collection of Disjoint Disk Covers)* to group victim nodes so as to simultaneously test as many interference-free victim nodes as possible in each cover.

Algorithm 1 The *DVN-BFS* Algorithm

1: **Input**: WSN $G(V, E)$
2: **Output**: All victim nodes W
3: $mg \leftarrow (0, 0, \ldots, 0)$
4: Construct the *BFS* tree on G
5: Calculate $\{mg_1, \ldots, mg_k\}$ for each path on *BFS* {The number of leaf nodes on *BFS* is k}
6: {Calculate W}
7: $mg \leftarrow \bigcup_{i=1}^{k} mg_i$
8: Construct a diagonal matrix $M \leftarrow diag(1, 2, \ldots, n)$
9: $W \leftarrow mg \cdot M$
10: **return** W

4.2.1 Algorithm Description

The basic idea of this algorithm is: For each victim node v, construct two disks D_v^1 and D_v^2 centered at v with radius $(R - r)$ and $(3R - r)$ respectively. The objective is to find out a minimum collection of disjoint disk covers, where each cover is a set of disjoint disks such that victims nodes within each disjoint disks can be tested simultaneously without mutual interference each others. We adopt a greed method for this by selecting a node v whose corresponding $R - r$ disk covers the maximal number of victim nodes, and then select another node u similarly after removing all the nodes within $3R - r$ from v from the graph. Iterate this until no nodes are left in the graph.

Algorithm 2 The *GVN-MCDDC* Algorithm

1: **Input**: All left victim nodes W_{i-1} after cover $i - 1$
2: **Output**: The collection of groups in all covers $G_{i1}, \ldots, G_{ij}$, $1 \leq i \leq \mathcal{C}$, $1 \leq j \leq \sqcup_i$
3: $i \leftarrow 1$
4: **while** $|W| \neq 0$ **do**
5: ▷ Construct double disks for each victim node
6: **for** $w \in W_{i-1}$ **do**
7: Construct D_w^1 and D_w^2
8: **end for**
9: $k \leftarrow 1$
10: $W_i \leftarrow \emptyset$
11: **while** $|W_{i-1}| \neq 0$ **do**
12: Choose $w \in W_{i-1}$ to maximize $\kappa(D_w^1)$
13: $G_{ik} \leftarrow D_w^1$
14: $W_{i-1} \leftarrow W_{i-1} \setminus D_w^2$
15: $W_i \leftarrow W_i \cup \{D_w^2 \setminus D_w^1\}$
16: $k \leftarrow k + 1$
17: **end while**
18: $W \leftarrow W_i$
19: **end while**

4.2.2 The Analysis

LEMMA 1. *Any two nodes with the distance larger than $R + r$ are interference-free nodes.*

PROOF. Assume that any two nodes u and v with distance $\delta(u, v) > R + r$ are not interference-free. Then there exists a jammer J such that J can interfere both u and v. Without lost of generality, we can assume that node v activates J. Thus $\delta(v, J) \leq r$. Plus, $\delta(v, J) \leq R$ and $\delta(u, J) \leq R$, then $\delta(u, v) \leq R + r$, which contradicts to our assumption. □

LEMMA 2. *For each set of victim nodes in disk D_v^1, the center node v can be used as the collector. That is u can sense the noises from any jammers triggered by any nodes within the distance $R-r$ from v.*

PROOF. The proof is straightforward. Assume that a center node v in disk D_v^1 cannot sense the noise from a jammer J, which is activated by a node u in the disk D_v^1. Then, we have $\delta(u,J) \leq r$ and $\delta(v,J) > R$. Therefore, $\delta(u,v) > R-r$, contradicting to the fact that u is in D_v^1. □

LEMMA 3. *The number of victim nodes covered by cover i $\mathcal{N}_i$ is at least w_i/Δ_i, where w_i refers to the number of the uncovered victim nodes at the beginning of cover i.*

PROOF. Denote by W_i the set of center nodes for disjoint disks D_w^1 in cover i ($w \in W_i$), then the number of victim nodes covered $|C_i|$ is:

$$\begin{aligned}
|C_i| &= \sum_{w \in W_i} \kappa(D_w^1) \\
&= \sum_{w \in W_i} \frac{\kappa(D_w^1) \cdot \kappa(D_w^2)}{\kappa(D_w^2)} \\
&\geq \sum_{w \in W_i} \frac{\sum_{u \in \{D_w^2 \setminus w\}} \kappa(D_u^1)}{\kappa(D_w^2)} \\
&\geq \frac{1}{\Delta_i} \sum_{w \in W_i} \sum_{u \in \{D_w^2 \setminus w\}} \kappa(D_w^1) \\
&\geq \frac{1}{\Delta_i} \cdot n
\end{aligned}$$

□

Since the jammer noise range R is always larger than normal transmission range r, the trigger nodes must be included in the victim nodes. In the *GVN-MCDDC* algorithm, when constructing one $(R-r)$-disk D_w^1 for each left victim node according to LEMMA 2, the center node w_{ij} in each group j in cover i is certain to be able to hear the noises from the jammmers in each cover. Moreover, for each left victim node, we construct another $(3R-r)$-disk D_w^2, where $(3R-r)$ is from $2(R-r)+(R+r)$, where $R+r$ can guarantee the non-interference between groups according to LEMMA 1 and $R-r$ is as above.

THEOREM 1. *The number of covers $\mathcal{C}$ to cover all victim nodes is at most $\Delta(H)$, where $H(3R-r) = H_1(3R-r)$ is the graph at the beginning of the first cover.*

PROOF. Since for each cover a maximal collection of disjoint disks D_w^2 are selected, any uncovered victim node w has $\delta(w_1, w_2) \leq 3R-r$ for some disk center w. Therefore, by the end of cover i, after all these disk center nodes are removed from graph H_i, the degree of all the remaining nodes should be decreased by at least 1. Hence,

$$\Delta_{i+1} \leq \Delta_i$$

for any $i \in [1, \mathcal{C}]$.

According to LEMMA 3 we have, by any cover i at least n/Δ_i victim nodes can be covered. Therefore, let L_i be the number of uncovered victim nodes, we can iterate the algorithm as follows:

$$\begin{aligned}
C_1 &\geq \frac{n}{\Delta_1} \Rightarrow L_1 \leq n - \frac{n}{\Delta_1} = \frac{\Delta_1 - 1}{\Delta_1} n; \\
C_2 &\geq \frac{n}{\Delta_2} \Rightarrow L_2 \leq \frac{\Delta_2 - 1}{\Delta_2} L_1 \leq \frac{\Delta_1 - 2}{\Delta_1} n \\
&\vdots \\
C_{\mathcal{C}} &\geq \frac{n}{\Delta_{\mathcal{C}}} \Rightarrow L_{\mathcal{C}} \leq \frac{\Delta_1 - \mathcal{C}}{\Delta_1} n
\end{aligned}$$

Therefore, the iteration terminates at cover $\mathcal{C} \leq \Delta_1 = \Delta(H)$. □

4.3 The *DTN-NCGT* Algorithm

In this section, for each group j in cover i (a set of victim nodes in each disk D_v^1 in cover i), we devise the ***DTN-NCGT*** algorithm *(Detection of Trigger Nodes based on Non-Adaptive Combinatorial GT)* to detect all trigger nodes in these groups of victim nodes while all groups in the same cover can be tested at the same time.

Algorithm 3 The *DTN-NCGT* Algorithm on group j in cover i

Input: Victim nodes W_{ij} in one group, R, r
Output: trigger nodes U_{ij} in this group ($U_{ij} = d_{ij}$)
Construct $G_{ij}(W_{ij}, E_{ij})$, where $E_{ij} = \{(u,v) | \delta(u,v) \leq 2r, u, v \in W_{ij}\}$

▷ Find the upper bound D_{ij} of d_{ij}
$D_{ij} \leftarrow 0$
for $k = 1, 2, 3$ **do**
  Find the MAXIMUM CLIQUE $c(G_{ij})$ on graph G_{ij}
  $G_{ij} \leftarrow G_{ij} \setminus \bigcup_{w \in c(G_{ij})} w$
  $D_{ij} \leftarrow D_{ij} + |c(G_{ij})|$
end for
▷ Test by using NON-ADAPTIVE GT
Construct a D_{ij}-DISJUNCT MATRIX M_{ij}
Group the column in each row with entity 1 into one group
Test these groups simultaneously
Decode the testing result to find out all trigger nodes d_{ij}

4.3.1 Algorithm Description

In *DTN-NCGT* algorithm, by using GT based on d-disjunct matrix, all trigger nodes can be identified in $O(1)$ time in each cover. Consider construct a graph for each group $G_{ij}(W_{ij}, E_{ij})$, where $E_{ij} = \{(u,v) | \delta(u,v) \leq 2r, u, v \in W_{ij}\}$ based on the following LEMMA 6, we can find the upper bound D_{ij} of trigger nodes d_{ij}. Then we detect trigger nodes from victim nodes for each group based on D_{ij}-disjunct matrix, where D_{ij} is the upper bound of d_{ij} in LEMMA 2 in group j and cover i.

As described at the beginning of Section 4, the complete *ITN* algorithm is presented in Algorithm 4.3.1.

Algorithm 4 The *ITN* Algorithm

1: **Input**: WSN $G(V, E)$
2: **Output**: *TNLT* Broadcast Tree T
3: $W \leftarrow$ The set of victim nodes
4: $W_i \leftarrow$ The set of left victim nodes from cover $i-1$
5: $G_{ij} \leftarrow$ The group j in cover i
6: $U \leftarrow$ The set of trigger nodes
7: $U_i \leftarrow$ The set of trigger detected in cover i
8: $T \leftarrow$ The *TNLT* broadcast tree
9: $W \leftarrow \emptyset$, $U \leftarrow \emptyset$, $W \leftarrow$ Victims nodes based on the *DVN-BFS* Algorithm
10: $W_1 \leftarrow W$
11: $i = 1$
12: **while** $|W_i| > 0$ **do**
13: $G_{ij} \leftarrow$ Groups based on the *GVN-MCDDC* algorithm in cover i
14: $U_i \leftarrow$ trigger nodes based on the *DTN-NCGT* algorithm
15: $U \leftarrow U \cup U_i$
16: $i \leftarrow i+1$
17: **end while**

4.3.2 The Analysis

LEMMA 4. *The maximum jamming area with 3 jammers without a hole in it is* $(2\pi + 3\sqrt{3}/2)R^2$.

PROOF. This proof is straightforward. We prove, as in Figure 2, the jammers can be located as J_1, J_2, J_3. In this case, ΔABC is an equilateral triangle with the area $\sqrt{3}R^2/4$. The overlap area is $6\times(\pi R^2/6 - \sqrt{3}R^2/4) = \pi R^2 - 3\sqrt{3}R^2/2$. Therefore, the maximum jamming area with 3 jammers is $3\pi R^2 - (\pi R^2 - 3\sqrt{3}R^2/2) = (2\pi + 3\sqrt{3}/2)R^2$.
□

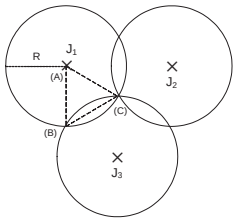


Figure 2: Estimated the Maximum Jamming Area

LEMMA 5. *Under our assumption that the jammers are deployed to jam as large area as possible, the estimated number of jammers triggered by the nodes in the same group is at most 3.*

PROOF. According the maximum jamming area discussed in LEMMA 4, the jammers are assumed to be deployed like Figure 2. In Figure 3, we consider the worst case that the center victim node of disk D_w^1 is at the cross point of the jammers' noise ranges (v is the victim nodes in Figure 3). In this case, since the radius of disk D_w^1 is $R-r$ so that at most only three trigger nodes u_1, u_2, u_3 can trigger at most three jammers J_1, J_2, J_2 respectively (shown in Figure 3) if u_1, u_2, u_3 exist. Let's say that even though the jammers are not so accurately distributed to maximize the jamming area, there might be a few more nodes than 3 which can trigger at most all these 3 jammers. Therefore, the estimated number of jammers triggered by the nodes in the same group is at most 3. □

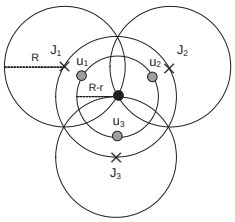


Figure 3: Upper Bound of trigger Nodes for One Jammer (Black nodes are victim nodes, Grey nodes are trigger nodes, Forks are jammers)

LEMMA 6. *The trigger nodes which trigger the same jammer must have less than distance* $2r$ *with each other.*

PROOF. We know only the nodes within the disk of radius r with the jammer as the center can trigger the jammer. Thus these trigger nodes have less than 2r distance with each other.
□

THEOREM 2. *The upper bound* D_{ij} *of the number of trigger nodes* d_{ij} *in one group is*

$$\sum_{k=1}^{3} |c(G_{ij} \setminus \bigcup_{s=0}^{k-1} \bigcup_{w \in G_{ij}^s} w)|$$

where $c(G)$ *is maximum clique on graph* G, $|c(G)|$ *is the size of clique* $c(G)$, G_{ij}^s *is the graph after removing vertices in* s *maximum cliques.*

PROOF. According to LEMMA 5, the nodes in one group can trigger at most 3 jammers. According to LEMMA 6, we know the trigger nodes to trigger the same jammer have less than distance 2r. In algorithm 3, we construct a unit disk graph $G_{ij}(W_{ij}, E_{ij})$ with disk radius 2r so that the nodes which trigger the same jammer must be a clique in graph G_{ij}.

In each iteration s, according to algorithm 3, we choose the maximum clique and remove all nodes in this clique and all number of them to D_{ij}, that is, $D_{ij}^s = |c(G_{ij}^s \setminus \bigcup_{w \in G_{ij}^s} w)|$.

After 3 iterations, we have $D_{ij}^s = \sum_{k=1}^{3} |c(G_{ij} \setminus \bigcup_{s=0}^{k-1} \bigcup_{w \in G_{ij}^s} w)|$.
□

THEOREM 3. *The number of testing covers to detect trigger nodes in each group of victim nodes* n_{ij} *is upper bounded by*

$$\lceil \min\{(2+o(1))\frac{D_{ij}^2 \log_2^2 n_{ij}}{\log_2^2(D_{ij}\log_2 n_{ij})}, n_{ij}\}/m \rceil$$

where $D_{ij} = \sum_{k=1}^{3} |c(G_{ij} \setminus \bigcup_{s=0}^{k-1} \bigcup_{w \in G_{ij}^s} w)|$.

PROOF. The best upper-bound of the number of rows for d-disjunct matrix is $\min\{(2+o(1))\frac{d^2\log_2^2 n}{\log_2^2(d\log_2 n)}, n\}$, using Du's construction [7, 6]. In WSNs, as we defined there are m radios so that at most m groups can be tested at the same time. According to THEOREM 2, d_{ij} are bounded by D_{ij} and n_{ij} is the number of victim nodes, we complete the proof. □

COROLLARY 1. *The total number of testing rounds in cover i is upper bounded by* $\max_j \lceil \min\{(2+o(1))\frac{D_{ij}^2\log_2^2 n_{ij}}{\log_2^2(D_{ij}\log_2 n_{ij})}, n_{ij}\}/m \rceil$.

THEOREM 4. *The Message Complexity per node w is* $O((2+o(1))\frac{D_{ij}\log_2 n_{ij}}{\log_2(D_{ij}\log_2 n_{ij})})$.

PROOF. In D_{ij}-disjunct matrix, the number of messages each node needs to transmit is the number of 1-entries in the corresponding column. As we mentioned above, Du's construction method [7, 6] for d-disjunct matrix, has the lowest upper-bound for the matrix size. It is trivial to find that, each column has exactly s 1-entries in the matrix constructed in that way, where

$$s = O((2+o(1))\frac{D_{ij}\log_2 n_{ij}}{\log_2(D_{ij}\log_2 n_{ij})})$$

hence the message complexity per node is the same. □

THEOREM 5. *The total testing time $\mathcal{T}$ is upper bounded by*

$$O(\sum_{i=1}^{\Delta(H)} \max_j \lceil \min\{(2+o(1))\frac{D_{ij}^2\log_2^2 n_{ij}}{\log_2^2(D_{ij}\log_2 n_{ij})}, n_{ij}\}/m \rceil)$$

where $D_{ij} = \sum_{k=1}^{3} |c(G_{ij} \setminus \bigcup_{s=0}^{k-1} \bigcup_{w\in G_{ij}^s} w)|$.

PROOF. According to THEOREM 3 and COROLLARY 1, the covers for all victim nodes are $\Delta(H)$ and the testing time for each cover is the maximum testing time among all groups, that is,

$$\max_j \lceil \min\{(2+o(1))\frac{D_{ij}^2\log_2^2 n_{ij}}{\log_2^2(D_{ij}\log_2 n_{ij})}, n_{ij}\}/m \rceil$$

where $D_{ij} = \sum_{k=1}^{3} |c(G_{ij} \setminus \bigcup_{s=0}^{k-1} \bigcup_{w\in G_{ij}^s} w)|$. We complete the proof. □

5. THE *TNLT-CDS* ROUTING ALGORITHM

One of the benefits for identifying the trigger nodes is to help construct a routing protocol which does not activate any reactive jammer. In this section, we propose a simple routing algorithm called ***Trigger Nodes Leaves Tree based on Connected Dominating Set*** (*TNLT-CDS*) which uses trigger nodes as only end receivers. Together with the *ITN* algorithm, *TNLT-CDS* will complete an efficient countermeasure for reactive jamming attacks.

We will utilize the Connected Dominating Set (CDS) to construct our *TNLT-CDS* as CDS has been shown as one of the most efficient methods for constructing a broadcast protocol. Again, consider network $G = (V, E)$ with $U \subset V$ as a set of trigger nodes identified by *ITN*. We will construct a directed graph $G' = (V, E')$ by changing all the undirected edges $(u, v) \in E$ where $u \in V \setminus U$ and $v \in U$ to the directed edge (u, v). We then deploy any CDS algorithm in directed graph [15] on G'. It is easy to see that the obtained CDS S will not consist of any node in U. Finally, we construct a broadcast tree T by connecting nodes in S to the rest using newly added directed edges.

6. SIMULATION

In this section, we evaluate the efficiency of our design through a series of simulations in terms of time latency, for sensor networks with different parameters. The results of these experiments show that this solution is timely efficient for identifying trigger nodes and defending reactive jamming attacks, for practical networks.

6.1 Simulation Setup

In order to simulate a general sensor network, we randomly distribute a total of N sensor nodes with one base station and J jammers to a square network field with width s. As has been mentioned above, the base station, sensor nodes and jammers have respectively transmission range, ρ, r and R. In order not to exaggerate the power of the base station, we assume $\rho = r$ in this simulation, while larger ρ would make this solution more efficient.

We have in total six benchmarks in the simulations with different input parameter teams. On one hand, we study the average *number of disk covers* T in the *GVN-MCDDC algorithm*, and the *maximum node degree* Δ to validate the bound of T proved in THEOREM 1. On the other hand, we show the *overall test length* (number of rounds t) analyzed in THEOREM 3. Moreover, we record the *number of victim nodes* n and the *total volume of communication messages* M between the sensors and the base station, to indicate the message complexity of this method. To investigate the effects of a series of network parameters, over the efficiency of this solution, we vary the values for the *number of jammers* J, *number of radios* m, *number of sensor nodes* N, *width of the square network region* s as well as *noise range ratio* α, hence the following five paragraphs and Figure 4(a)–(j) are the corresponding results and analysis. Note that for each parameter team, 100 network instances are investigated and the results were averaged.

6.2 Results and Analysis

6.2.1 Performance by the number of jammers J

Figure 4(a) and (b) explain our protocol performance based on the various numbers of jammers J in the network. In this test, we have $N = 1000$ nodes with $m = 3$ radios, on a 1500×1500 network field, where $J \in [1, 10]$ jammers are randomly deployed. Our protocol employs sophisticated technique to perform as many parallel testing as possible as shown in Algorithm 3, therefore the number of testing rounds, T, can be stable while the number of jammers J and victim nodes n increase. As shown in Figure 4(a) and (b), T increases a little while n can vary from 50 to 450 when J increases from 1 to 10.

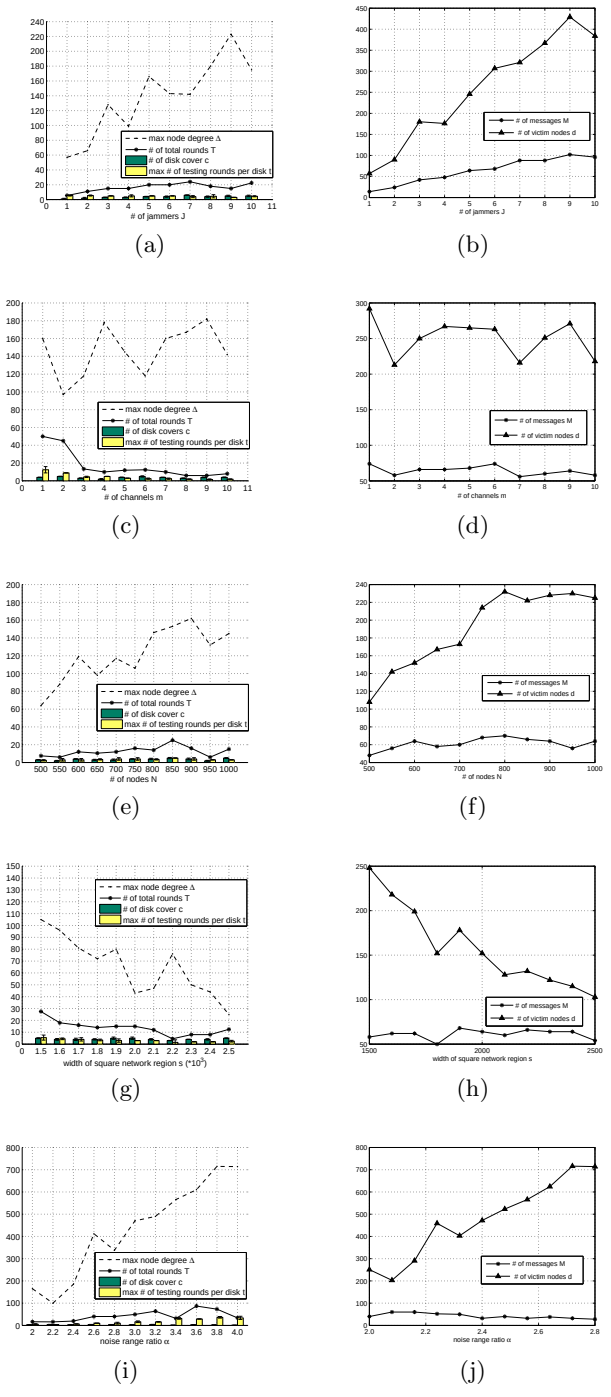


Figure 4: Experimental Results by Various Parameters

More specifically, the number of disk covers c and maximum number of testing rounds per disk t are smaller than 10, where the latter is much smaller than maximum node degree Δ. This contributes to dramatically small number of overall rounds T, which is no larger than 30 and stable for increasing J.

Moreover, since each $(R-r)$-disk in our tests needs only one sensor node to send the result back to the base station, the message complexity M is also much smaller (less than 100) than the number of victim nodes n. Note that in individual testing method, M should be as high as $O(n)$.

Therefore, our solution can promptly defend a jamming attack with increasing number of jammers, in terms of time complexity and message complexity.

6.2.2 *Performance by the number of radios m*

During the series of tests within $(R-r)$-disk, we accelerate the overall testing latencies by employing the multiple radios m since with a given *d-disjunct* matrix, m number of rows can be tested simultaneously. The parameters for this simulation are randomly distributed N=1000 nodes and J=5 jammers in a 1500×1500 network area, where $m \in [1, 10]$. In other words, the number of tests within an $(R-r)$-disk can be reduced by the factor m, which also shortens the *overall test length*. As illustrated in Figure 4(c), the maximum testing rounds per disk decrease as the radio size increase, which assists to drop the total rounds, T, drastically. Especially, when m=2 from m=1 in the Figure 4(c), the overall total rounds drop rapidly. We show the performance of our system based on the various numbers of radios, m, on a node. In conclusion, we learn that the radio size can highly benefit the *overall test length* of our protocol.

6.2.3 *Performance by the number of nodes N*

The number of nodes in the network is one of the critical aspects to consider a mobile network solution in terms of scalability. For instance, the countermeasure using DS-CDMA technique suffers from the number of codes for encoding/decoding messages in each node of the network, since the newly joined nodes trigger the creation of additional codes for the victorious battle against jammers who try to corrupt the messages by pilfering the codes. However, the fact that our solution is a disk-based test approach relieves the scalability problem fairly, and the performance shows somehow constant movement where $N \in [500, 1000]$ in Figure 4(e) and (f). As shown in Figure 4(e) and (f), the victim quantity increases obviously as the number of nodes increases, but the number of messages is quite constant. Moreover, the total testing rounds increase slowly. This figure shows how our system efficiently operates when the number of nodes varied from N=500 to N=1000 with $m = 3$ and J=5 jammers in a 1500×1500 network area. From this evaluation, we can conclude that our model is also a very suitable security solution for the majority of sensor networks in various areas.

6.2.4 *Performance by the density of the network*

Now, we will show how the protocol we proposed reacts in the various network densities where the network field size broadens. With the given number of jammers and the increase of the network field, it is clear that the number of victim nodes decreases where the system tries to deploy the nodes in order to cover the network field as much as possible. As we discussed before, due to the fact that our approach is disk-based classification of the nodes, sparse network would mainly help to reduce the number of victim nodes, especially Δ, and then reduce the overall testing rounds as well. Figure 4(g) and (h) shows the various simulation results with the increasing network field size from 1500×1500 to 2500×2500 where N=1000 with $m = 3$ and J=5 jammers. As the network is sparse, the number of victim nodes decreases as Δ gets smaller in this figure as we discussed.

6.2.5 *Performance by the α in transmission range of the jammers*

Now let's consider the interference range of the jammers. Since noise range is relatively larger than transmission range of sensor nodes, more messages of the sensor nodes will be jammed. In contrast, in our system, the larger jammers signal range may imply the increasing number of testing rounds even though this does not determine the damage area of the network. For instance, JAM [1] locks down the whole jammed region while our system minimizes the jammed region size by classifying the nodes more efficiency. In Figure 4(i) and (j), as the α gets larger, the number of victim nodes increases since a jammer can transmit farther and contaminates more nodes during the activation. Moreover, more victim nodes requires more testing rounds to cull out the trigger nodes among them. In our result, the number of rounds rises as α gets larger while the routing in the jammed region differs from the transmission range of the jammers through the classification process. However, the number of rounds is changing very slowly.

7. RELATED WORKS

There are many attack strategies to maximize effective damages to the network, such as constant jammer, deceptive jammer, random jammer and reactive jammer. Among these, the reactive jammer might have the most intelligent behavior by monitoring the communication channels, instead of sending out a radio signal based on its own decision. Thus the reactive jamming attack, which is the focus of our paper, is harder to detect than any other strategy.

The use of spread-spectrum communication has been considered one of the most well-known schemes to evade the jamming attacks. In the literature, there are two ways to implement this spread-spectrum system, frequency hopping (FH) [12] and code division multiple access (CDMA) [4]. Channel surfing [18, 17] motivated by FH technique is also well-known approach to discuss. Recently, there have been couple of new approaches against jamming attacks, and we will discuss regarding some of those solutions [1, 14] in this section.

The initial outline of the FH method in [12] is to keep switching the communication frequency into safe channels in order to avoid the jamming signal based on the global switching sequences in the networks. The distribution of the synchronized sequences could hurt the overall performance of the networks drastically since the sequences should be updated frequently and globally for every single node in WSNs. For this distribution problem, Strasser [13] proposed the Uncoordinated Frequency Hopping (UFH) technique for the anti-jamming point-to-point scheme to establish the secrete key between two communication parties. This scheme is based on probabilistic model, such that it wastes computational overhead even without jamming attacks in the networks, and the performance of the scheme is greatly effected by the number of channels. Importantly, FH technique has limitation against follower jamming attacks (also known as a repeater jamming attacks) since the follower jammers would neutralize the benefit of FH. Despite there have been several studies on the follower jamming attacks including [8, 11, 16], no complete solution on the problem of follower jamming attacks so far.

The channel surfing method [18] from MAC layer uses similar mechanism motivated by FH, but it switches the channels on demand. The critical issues for this solution are synchronization, latency and scalability from the coordinated channel switching procedures across the whole networks. Xu also introduced spatial retrieval method [17] with mobile nodes to escape from a jammed region. This method would help mobile nodes to move away from the jammed region , but the networks would be unbalanced and even isolated by the attacks with mobile jammers. On the contrary, the mitigation solution we proposed has small time and message complexity, which can overcome the deficiency of these methods.

With respect to the last approach CDMA among the spread-spectrum communication, one study [4] related to utilizing the direct sequence CDMA(DS-CDMA), resorts to high-power dynamic tree-remerging schemes, to maintain the small number of orthogonal codes in use, and avoid re-calculation for the codes. However, due to the variation in nodes for dynamic environments, this method suffers from additional maintenance overhead for join-in and leaving behaviors, especially, computation of orthogonal codes takes a huge amount of time. In this respect, our scheme needs no cryptographic key management and is scalable and stable to various dynamic networks. Direct-Sequence Spread Spectrum(DSSS) [5] is also common mitigation solution against jamming attacks, but it has key sharing problem since communication parties need to acquire the key in order to establish the safe communication beforehand. Additionally, DSSS has limitation against repeater jamming attacks [19, 10] as well since the repeater jammers try to acquire the code by monitoring the on-going traffic and garble the communication messages based on the acquisition of the code.

Recently, Tague *et. al* have presented a linear programming model [14] for a specific type of the jamming attack, but it mainly focuses on the flow based attack without the consideration of protocol based attacking model. Besides the defenses mentioned above, a mapping-based defense has been introduced in JAM [1]. In this system, the jammed nodes cooperatively map the jammed region. A deficiency of this approach would be the possible unnecessarily large jammed region built against reactive jamming attack. As a result, parts of the networks might be isolated. This is because many nodes in the exaggeratedly large jammed region may still be able to transmit without activating the jammers, yet they are isolated and the message delivery are interrupted. In our solution, no unnecessary trigger nodes would be mistakenly disabled, hence the defense are more efficient.

In this work, we introduce a novel countermeasure protocol against reactive jamming problem with minimum time latency and low message complexity for the sensor node with limited resources.

8. CONCLUSION AND FURTHER DISCUSSION

To efficiently tackle reactive Jamming attacks in multiple-radio WSN, we devise a new mitigation for identifying trigger nodes, whose broadcasting triggers the jammers, and a routing protocol to switch trigger nodes to receivers so as to keep jammers idle. By utilizing an integration of Bread-First-Search tree, nonadaptive group testing scheme, dis-

joint disk cover method, and clique-based clustering, this countermeasure achieves low overhead in terms of time and message complexity, thus is practical for general WSNs. Besides the analytical complexity analysis, we also conduct a series of simulations to investigate the scalability and stability of this method to various WSNs. The outstanding performances showed strengthen the capability and potential of this ITN defense.

Throughout this paper, we assume no packet-loss during all the transmissions, while in real WSNs this is inevitable. In the case that all the broadcasting messages of one trigger node do not arrive the jammers nearby, due to the packet-loss, the test outcomes of the corresponding groups might have error and fail to identify all the trigger nodes out. Even in this extreme case, our method can also be adapted using error-tolerant GT techniques [7, 6] without large additional overhead.

For the future work, we would accomplish the distributed version of all these proposed algorithms, examine how these algorithms perform to locate the triggers in a fault-tolerant environment and further identify the jammers by the locations. We will also investigate the sufficient conditions for the existence of constructing routing algorithms where triggers as only receivers.

9. REFERENCES

[1] S. S. A. D. Wood, J.A. Stankovic. A jammed-area mapping service for sensor networks. *Proc. 24th IEEE Intl. Real-Time System Symposium*, pages 286–297, 2003.

[2] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.

[3] C. Bron and J. Kerbosch. Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.

[4] J. Y.-C. H. Chiang. Dynamic jamming mitigation for wireless broadcast networks. *INFOCOM*, 2008.

[5] Y. Desmedt, R. Safavi-Naini, H. Wang, C. Charnes, and J. Pieprzyk. Broadcast anti-jamming systems. *Networks, 1999. (ICON '99) Proceedings. IEEE International Conference on*, pages 349–355, Sept.-1 Oct. 1999.

[6] D.-Z. Du and F. Hwang. *Combinatorial Group Testing and its Applications(2nd ed.)*. World Scientific, Singapore, 1999.

[7] D.-Z. Du and F. Hwang. *Pooling Designs: Group Testing in Molecular Biology*. World Scientific, Singapore, 2006.

[8] E. Felstead. Follower jammer considerations for frequency hopped spread spectrum. *Military Communications Conference, 1998. MILCOM 98. Proceedings., IEEE*, 2:474–478 vol.2, Oct 1998.

[9] R. Gupta and J. Walrand. Approximating maximal cliques in ad-hoc networks. *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, 1:365–369 Vol.1, Sept. 2004.

[10] W. Hang, W. Zanji, and G. Jingbo. Performance of dsss against repeater jamming. *Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference on*, pages 858–861, Dec. 2006.

[11] A. A. Hassan and J. E. Hershey. On a follower tone-jammer countermeasure technique. *IEEE Transactions on communications*, 43(4), Mar 1995.

[12] O. Sidek and A. Yahya. Reed solomon coding for frequency hopping spread spectrum in jamming environment. *American Journal of Applied Sciences*, 5(10):1281–1284.

[13] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. IEEE, May 2008.

[14] P. Tague, D. Slater, R. Poovendran, and G. Noubir. Linear programming models for jamming attacks on network traffic flows. *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pages 207–216, April 2008.

[15] M. T. Thai, R. Tiwari, and D.-Z. Du. On construction of virtual backbone in wireless ad hoc networks with unidirectional links. *IEEE Transactions on Mobile Computing (TMC)*, 7(8):1–12, 2008.

[16] D. Torrieri. Fundamental limitations on repeater jamming of frequency-hopping communications. *Selected Areas in Communications, IEEE Journal on*, 7(4):569–575, May 1989.

[17] W. T. W. Xu, T. Wood and Y. Zhang. Channel surfing and spatial retreats: Defenses against wireless denial of service. *Proceedings of the 2004 ACM workshop on Wireless security*, pages 80–89, 2004.

[18] W. T. W. Xu, T. Wood and Y. Zhang. Channel surfing: Defending wireless sensor networks from namming and interference. *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 499–508, April 2007.

[19] H. Wang, J. Guo, and Z. Wang. Feasibility assessment of repeater jamming technique for dsss. *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2322–2327, March 2007.

[20] Y. Z. Wenyuan Xu, Wade Trappe and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. *International Symposium on Mobile Ad Hoc Networking and Computing*, pages 6–57, April 2005.