Self-organization supported algorithms for wireless sensor networks

Jian Zhong
School of Computer Science and Information Technology
Royal Melbourne Institute of Technology
Melbourne, Australia
E-mail: jian.zhong@rmit.edu.au

Peter Bertok
School of Computer Science and Information Technology
Royal Melbourne Institute of Technology
Melbourne, Australia
E-mail: peter.bertok@rmit.edu.au

***Abstract*—*Self-organization is an important issue in wireless sensor networks because of the inherent unreliability of the network. Besides, variable threats in the networks can not be ignored. Extending battery life and enhancing robustness under variable threats are two essential aspects which need to be considered when a self-organization scheme is explored. In order to address these issues, a Redundant Nodes Selection scheme and a variable Threats Probability Estimation scheme are proposed in this paper. RNS is able to select redundant nodes that can be switched off without affecting overall sensing coverage. TPE is able to help a sensor node to choose the most suitable path and avoid high-threat neighbors in order to reduce packet loss. The scenario with RNS extended battery life by 30% to 50%, and postponed the occurrence of first partitioning in the network by 27% to 140%. TPE decreased packet loss by 225% to 400% when a high threat level was involved.*

Keywords-wireless sensor networks; self-organization; variable threats; battery life; robustness

I. INTRODUCTION

For the constraint of wireless sensor networks (WSNs), some threats can not be ignored, such as environment changes, sensor damage, information lost and sensor attacks etc. There are some key areas which need to be explored, such as [2] network organization, routing, security, node localization, clock synchronization, power management and key management etc. This focuses on network self-organization.

For wireless sensor networks, organizing typically begins with neighbor discovery [3]. Nodes send rounds of messages (packets), build local neighbor tables and organize clusters centered around a cluster head. The tables include information on each neighbor's ID and location. However, during operation some sensors become inactive due to battery exhaustion which may result in network partitioning, and packets can be lost due to various threats. Extending battery life, postponing the occurrence of first partitioning and reducing packet loss are significant aspects of self-organizing.

A self-organization scheme supported by a redundant nodes selection algorithm (RNS) and variable threats probability estimation (TPE) is proposed here to extend battery life and reduce packet loss. RNS is designed to scan all sensor nodes and select redundant nodes that can be switched off so that the whole area will still be covered. The redundant nodes will be used as backups and replacements to extend the effective network lifetime without any coverage loss.

The second method TPE, improves the scheme originally proposed in [7], by allowing nodes to choose a more reliable neighbor as a default path to the data sink and blocking high-threat nodes to reduce packet loss. The whole proposed scheme provides a solution for WSNs to extend battery life and avoid variable threats.

This paper is based on the conference article "A Variable Threats Based Self-Organization Scheme for Wireless Sensor Networks", presented at the SENSORCOMM 2009 conference.

II. BACKGROUND

Due to the physical constraints of wireless sensor networks, sensors organization, resilience to node capture attack and power-saving are essential aspects. This chapter discusses the work done by some of the researchers on wireless sensor network self-organization, coverage exploration, static attack probability and related aspects. In the first section, deployment and topologies will be presented. Then the self-organization issue will be discussed. Furthermore, previous works related to power saving and resilience will be detailed. In the end of this chapter, a summary of the essential literature is given.

A. Topologies and Network Architecture

For a wireless sensor network, the topology and network architecture always need to be considered first. In the literature, there are some wireless sensor network topologies and architectures proposed for the uniform and non-uniform deployment. The most common topologies and architectures are described by Bhaskar Krishnamachari [13], which are shown in Fig. 1.

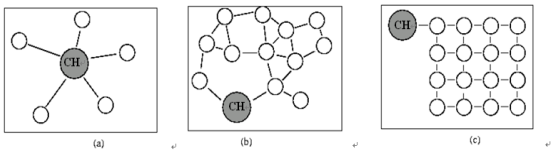

Figure 1. Different topologies in wireless sensor networks [13]

Fig. 1(a) shows the simplest topology, in which all sensors directly report the collected data to the cluster head

(CH). Fig. 1(b) shows a tree topology and the collected data is sent to the data sink via different paths depending on some factors, such as power-save path, most-secure path, reputation-based path etc. Fig. 1(c) shows a grid topology which is also used in an experimental model [7]. A more complex scenario is depicted in [13] with two-tiered architecture, as shown in Fig. 2.

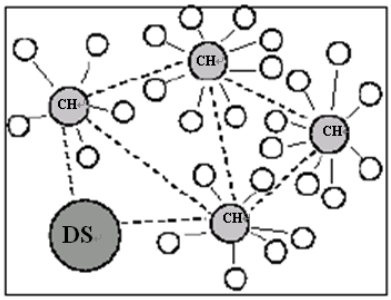

Figure 2. Topology in WSNs [13]

The biggest circle denotes the data sink (DS), which gathers reports from the cluster heads (CH). The small white circles indicate wireless sensors, which can collect data from their sensing ranges. In this typical topology, sensor nodes can directly connect to a cluster head which acts as a group leader.

In our research, we consider a clustered organization, when individual nodes are connected to a cluster head, and data from the cluster is relayed by the head towards the destination. However, in a number of cases there is no guarantee that all cluster heads can directly connect to the data sink or all sensor nodes can directly connect to a cluster head. For such cases, a random non-uniform architecture has been mentioned in [9], which is shown in Fig. 3.

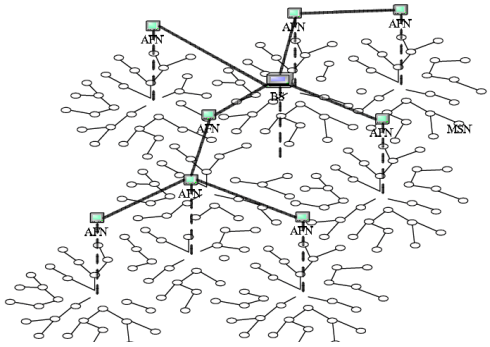

Figure 3. A two-tiered architecture [9]

In Fig. 3 [9], a small number of high-end nodes, called *Aggregation and Forwarding Nodes* (AFNs), are deployed together with numerous low-end sensor nodes, called *Micro Sensor Nodes* (MSNs). In addition, the network includes a globally trusted *base station* (BS), which is the ultimate destination for data streams from all the AFNs. The BS has powerful data processing capabilities, and is directly connected to an outside network. Each AFN is equipped with a high-end embedded processor, and is capable of communicating with other AFNs over long distances.

The deployment and topology in [9] are more reasonable for random scattering, and foremost, this architecture can adjust to the changes in topology during runtime, i.e. new nodes can be added into the network or some working nodes can be compromised.

Accordingly, a two-tiered wireless sensor network model will be used in our proposed method, and AFN will be called Cluster Head (CH), BS will be called Data Sink (DS) and MSN will be called Sensor Node (SN) in the rest of the paper.

B. Threats and Threat Model

There are many kinds of threats for WSNs, such as mentioned in [20]. In this paper, node failure will be discussed, including node capture, physical damage, battery exhaustion and any condition making the sensors unavailable. My proposed attack model does not include the scenario in which adversaries not only steal the data stored in the sensor nodes but also put the captured sensor nodes back into the WSNs as agents for collecting messages.

C. Self-Organization

After a sensor network has been deployed, self-organization comes. It will be separated into four different aspects: clustering and neighboring, power consumption, resilience, and sensor addition.

1) Clustering and Neighboring

To organize wireless sensors, Falko Dressler et al. [14] described a solution which involved a mobile robot that helps to organize the network. The model is shown in Fig. 4.

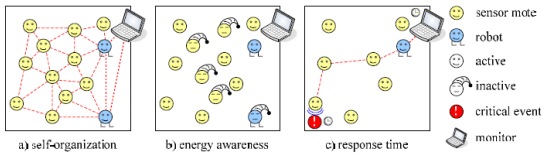

Figure 4. Challenges in the sensor networks [14]

Fig. 4(a), Self-organization, shows that the active sensor nodes can transfer their collected data to the robot via different paths and the monitor merely need to receive data from two robots. Fig. 4(b) shows that, for energy awareness, some sensors are switched to an inactive mode. A critical event is still can be gathered and transferred to the monitor, which is illustrated in Fig. 4(c). The method relies on mobile robots to maintain the whole network. However, in many cases, a mobile robot is not available.

Another self-organization mechanism for both uniform and non-uniform was described. The location-based mechanism [3] is relying on a special node named “server node”. This will raise the cost of the whole network and if this server node is compromised or damaged, all nodes under its control will be affected. If this function is embedded in each cluster head, the power consumption will increase, due to most of the key management and delivery being

performed by the server node. To reduce the memory overhead as well as maintain security for the network, a new approach is proposed in [9], which is called Survivable and Efficient Clustered Keying (SECK). However, a new issue, high-threat networks cannot be ignored in the path selection algorithms.

2) Power Consumption

To reduce WSN power consumption switching off some nodes has been introduced. Nevertheless, when switching off redundant nodes, how to maintain wireless network coverage and rerouting existing connections become new issues.

In [6], the proposed notion is that "if any parameter on a point can be reliably estimated, then this point can be claimed to be information covered".

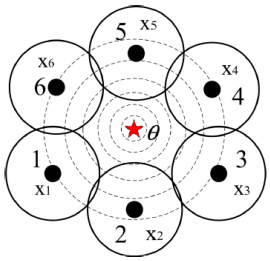

Figure 5. Illustration of physical and information coverage [6]

As shown in Fig. 5, although the sensing area of the node marked by a star is not covered, it may be "information covered" as long as the estimation error is small enough. The information coverage is based on parameter estimation, which means that for an unknown or uncovered point/area θ, there is a set of K sensor nodes for estimation. Each related sensor node has estimation for the non-physical covered point, which is (1),

$$x_k = \frac{\theta}{d_k^{\alpha}} + n_k, k = 1,2,\ldots K. \quad (1)$$

Where x_k denotes the output of the estimation; θ denotes the parameter of the non-physical covered point; d_k denotes the distance between a sensor k and a location with parameter θ; d_k^{α} denotes the attenuation with the distance where α denotes the attenuation exponent; n_k denotes the additive noise. The *mean squared error* (MSE) is used for the evaluation of the estimation for K related sensor nodes.

To try to achieve the sleep/wake up scheme, the authors [15] provided an information coverage estimation for redundant nodes selection named *Distributed Node and Rate Selection* (DNRS). In the proposed scheme, the redundant nodes can be switched off to save power by measuring the distortion. Distortion is any undesired change in the data transmitted such as signal strength and signal format etc. As described in the paper [15], the proposed scheme focused on an area. However, for some applications, such as tracking etc, the proposed scheme will not work properly. Besides, if two sensors are both information covered and rely on each other, which one will be switched off should be discussed.

These proposed methods [6, 15] are based on information coverage and can reduce the impact from transmission distortion. For large density of sensing nodes, a number of adjacent nodes may be available, but the computation overhead will rise. How many samples are enough for one event and how to choose the reporting sensors need to be considered. Besides, in [6] the balance of power-saving and information distortion should be carefully evaluated. In my proposed method, physical coverage will be considered and this situation will be improved.

3) Resilience

Being deployed in a hostile environment, failure/attack probability can not be ignored. Besides, researchers have pointed out that there is no sure and efficient way to readily detect a node capture [16, 17]. Thus, threat probability should be concerned.

In [7], the authors proposed a non-uniform sensor deployment algorithm based on static attack probability to improve the resistance of sensor nodes to node capture. The main algorithm is for estimating the number of keys, called the degree of each sensor node. Let $d_{i,j}$ denote the sensor nodes in the deployment group $G_{i,j}$. In the proposed method [7], the authors claimed that "the higher probability that a deployment group to be attacked implies that $d_{i,j}$ should be set to be a higher value". In other words, the degree of a node in each deployment group should be proportional to its attack probability. Thus, $d_{i,j}$ calculation is the core algorithm of the proposed method. In [7], $D_I(p_{i,j})$ denotes the inner group degree determination function used to calculate the degree of a node in group $G_{i,j}$, which means $d_{i,j} = D_I(p_{i,j})$. When $D_I(\cdot)$ takes input as $p_{i,j}$, it maps $p_{i,j}$ into one of $(\Omega+1)$ values. Formally, $D_I(\cdot)$ can be represented as

$$D_I(p_{i,j}) = \begin{cases} \pi_1, & \text{if } p_{i,j} < \varpi_1 \\ \pi_2, & \text{if } \varpi_1 < p_{i,j} < \varpi_2 \\ \vdots & \vdots \\ \pi_{\Omega+1}, & \text{if } \varpi_\Omega < p_{i,j}. \end{cases} \quad (2)$$

Therefore, if $p_{i,j} \le p_{i',j'}$ holds then $d_{i,j} \le d_{i',j'}$ holds. In [7], $D_I(p_{i,j})$ was designed to be a threshold function. $\{\varpi_1, \varpi_2, \cdots, \varpi_\Omega\}$ is a set of threshold values and $\{\pi_1, \pi_2, \cdots, \pi_{\Omega+1}\}$ is a set of $(\Omega+1)$ values. The values $\{\pi_1, \pi_2, \cdots, \pi_{\Omega+1}\}$ are given such that after the assignment of keys based on the setting, sensor nodes can resist attacks in the corresponding groups. These values are assigned based on experience. In the formula (2), $p_{i,j}$ denotes the normalized attack probability with respect to a deployment group $G_{i,j}$ and $p_{i,j}$ is defined as:

$$p_{i,j} = \frac{\omega_{i,j}}{\sum_{i',j'} \omega_{i',j'}} \quad (3)$$

In equation (3), $\omega_{i,j}$ denotes the attack coefficient associated with $G_{i,j}$. It is a value by considering all of factors and can be calculated as:

$$\omega_{i,j} = b_{i,j} + b_{i',j'} + \sum_{\rho=1}^{\alpha} \sum_{(b_{i',j'}, \tilde{b}_{i',j'}) \in \Psi(G_{i,j}, \rho)} (b_{i',j'} \times g_\rho + \tilde{b}_{i',j'} \times \tilde{g}_\rho) \quad (4)$$

where $\Psi(G_{i,j}, \rho)$ is a set of pairs of the base coefficient $b_{i',j'}$ for $G_{i',j'}$ and base coefficient for data sink $\tilde{b}_{i',j'}$ satisfying that $G_{i,j}$ and $G_{i',j'}$ are at a distance from ρ deployment groups. In [7], g_ρ denotes attack influence factors for sensor nodes and $\tilde{g}_\rho$ denotes attack influence factors for the data sink.

Also, $b_{i,j}$ denotes the basic attack coefficient for sensor nodes, representing the threat from adversaries and $\tilde{b}_{i,j}$ is that for the data sink. Accordingly, the coefficient $b_{i,j}$ and influence factors g_ρ will be used for computing the parameter $d_{i,j}$ that was specified at the beginning of this paragraph.

In [7], $b_{i,j}$ and g_ρ are both static. The proposed method had a good result when attack probability remained unchanged. An obvious improvement was presented not only on memory overhead but also on connectivity maintenance. For the algorithm, the higher attack the probability of the deployment group, the more keys were kept by group members to maintain connectivity. Although it had good performance in the described scenario without node failures, the paper did not consider the effect of nodes being compromised. Besides, static attack probability is not good enough for real scenarios, variable attack and failure probability should be considered. Thus, threat probability should be a concern.

D. Assumptions: The model considered in this thesis

In this paper, the deployment area is set to 2-dimensional and all sensors are randomly scattered. There are one data sink and three cluster heads with fixed location. In the model, a point *u* is covered (monitored) by a node *v* if their *Euclidian distance* is equal or less than the sensing range *R*. Sensor density (*D*) is defined as $D = m/(\pi R^2)$. Assume there are *m* nodes on average in each sensor's signal range and *R* is the radius of the signal range [18].

High threat probability estimation indicates nodes with "high threat" status, which have a high probability to lost packets and may affect their neighbors. In the model, four threat levels are used which are "no threat", "low threat", "high threat" and "compromised". Node attacks may raise the sensors' threat level. A "no threat" node will pass all packets without any packet loss. A "low threat" node may lose packets in a very low probability while sensors with "high threat" may lose packets in a very high probability. "Compromised" sensors will not forward any packets. Also, a sensor connecting to a "high threat" node may raise its threat level. Besides, no transmission error is included in the model.

In this paper, three different sensor distribution and three traffic distribution models are employed. Uniform distribution is used for normal environment monitoring. Normal distribution (*Gaussian* distribution) is used for some special usage such as monitoring forest fire. Zipf distribution is used for simulating some environment changing such as sensors are blown away by strong wind etc.

E. Summary

For WSNs organization, a self-organization mechanism for non-uniform distribution of sensor nodes is proposed in [14], and a location-based scheme for both uniform and non-uniform was described [5]. The mechanism in [14] offers good performance for non-uniform distribution, and an isolated part of the network can be reconnected by mobile sensors/devices. The method can find the optimum path to connect to the data sink efficiently. The solution in [14] relies on mobile robots to maintain connectivity in the network, but in some cases a mobile robot is not available. The location-based mechanism [5] is relying on a special node named "server node". This will raise the cost of the whole network and if this server node is compromised or damaged, all nodes under its control will be affected. If this function is embedded in each cluster head, the power consumption will increase, due to most of the key management and delivery being processed by the server node. To reduce the memory overhead as well as maintain security for the network, a new approach is proposed in [9], which is called *Survivable and Efficient Clustered Keying* (SECK). However, a new issue, default path selection becomes a problem in high-threat networks.

For the WSNs power consumption issue, an information coverage concept [6] has been proposed. In [6], a balance between coverage and sensor density has been explored. In some cases data can be estimated reliably, in other cases it cannot, and estimation cannot be a replacement of actual data.

For resilience, the attack probability estimation algorithm in [7] has a good experimental result for static attack probability estimation and connectivity maintenance. As mentioned before, a variable attack and failure probability is more realistic for WSNs.

For SN addition, a cluster and network-oriented scheme is proposed in [9]. However, the issue of connection between new sensors and existing cluster heads needs to be considered.

This paper focuses on the sensor node organization issues, and a neighbor-oriented self-organization mechanism will be proposed. Four aspects will be described respectively and then the integrative proposed scheme will be detailed.

III. PROPOSED METHOD

To extend the life for wireless sensor networks, redundant nodes can be switched off to save power, and later switched back on to replace failure nodes. Redundant nodes are those that can be switched off and the whole area will still be covered.

In this chapter, the proposed self-organization scheme is examined from three aspects, namely clustering and neighboring, battery life extension, network resilience and new sensor node addition. The self-organization mechanism addresses network maintenance, and is based on Redundant Nodes Selection scheme (RNS) and variable Threats Probability Estimation scheme (TPE). RNS is employed to select the redundant nodes in order to save power. TPE is used to help a sensor node to choose the most suitable path and avoid high-threat neighbors in order to reduce packet loss.

The proposed scheme not only extends battery life, but also enhances network robustness and maintains connectivity. In addition, the proposed scheme can be applied to both uniform and non-uniform distributions.

A. Redundant Node Selection (RNS)

As mentioned in the background chapter, the authors [13] employed a similar system named Distributed Node and Rate Selection (DNRS). The method in [13] is aiming to measure whether a node is redundant by calculating the distortion. However, this method has a limited effect when the objects are appearing random, or sensor density is high.

All in all, less than full coverage will bring in some new problems. For instance, when tracing an object, in the non-covered area we will lose track of the object. Although it may be estimated from related information, it is still not precise.

RNS algorithm has two steps. One is to select redundant nodes and the other is to check whether the redundant nodes can be switched o.

Notion 1: *It is a redundant node, if and only if there are no changes in the covered area when it has been switched to sleep mode.*

A simple sketch map is shown in Fig. 6. In the simulation, the sensor density is much higher. In the Fig. 6, SN_1 can be switched to sleep mode because its original sensing area is covered by other sensor nodes, namely by SN_2, SN_3, SN_4 and SN_5. Thus, even if SN_1 has been switched off, there are no changes in the covered area.

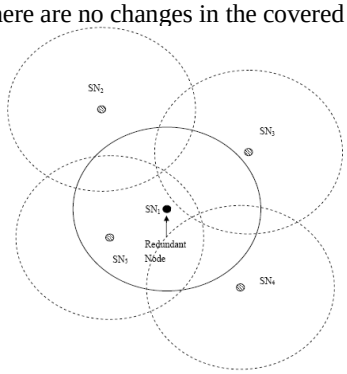

Figure 6. The redundant node

1) The RNS Algorithm and Proof

Assume that every node has its own location information which will be the coordinate in this algorithm and all radio range (radius) will be R. Assume that all sensor nodes are in the same 2-dimensional area. In this thesis, assume a point u is covered (monitored) by a node v if their Euclidian distance is equal or less than the sensing range, i.e., $|uv| \leq R$. Define the sensing circle C(u) of node u as the boundary of u's coverage region.

Notion 2: Let SN_x be a set of sensor nodes. A sensing area is fully covered if and only if for $\forall Z(x,y) \in C(SN_1)$, there exists at least one $C(SN_k)$ $(k > 1)$, that $Z(x,y) \in C(SN_k)$ is true.

Theorem 1: A sensing area of SN_1 is fully covered by a group of nodes if and only if, for $\forall Z(x,y) \in C(SN_1)$, there exists a set of nodes group $G_N = \{SN_2, SN_3, \cdots SN_k\}$ that $Z \in C(SN_2) \bigcup C(SN_3) \bigcup \cdots \bigcup C(SN_k)$ is true.

Proof: Assume for $\forall Z(x,y) \in C(SN_1)$, there exists a set of nodes group $G_N = \{SN_2, SN_3, \cdots SN_k\}$ that $Z \in C(SN_2) \bigcup C(SN_3) \bigcup \cdots \bigcup C(SN_k)$. Without loss of generality, there must be a $C(SN_j), 2 \leq j \leq k$ that $Z \in C(SN_j)$ is true.

The algorithm has the following steps:

1. Put a node SN1 at the origin of the coordinate system.
2. Assume there is a node SN2 in C(SN1) which means node SN2 is in node SN1's radio range, vice versa. Without loss of generality, let SN2 be on the X-axis, shown in Fig. 7.
3. The X-axis and circle C(SN2) intersect at P'. C(SN1) and C(SN2) intersect at Q' and Q".
4. To find the next circle.

4.1 If there is a node SN3 in the area Q'SN1W and SN2Q' ≤ R, then go to step 3 and replace SN2 by SN3. If SN2 could be the next circle, then node SN1 can be switched to sleep mode, as shown in Fig. 8.

4.2 If there is a node SN3 in the area Q'SN1W and SN2Q' > R, as shown in Fig. 9, there will be a small area A'B'Q' which is not covered. If there exists a node SNE that SNEA' ≤ R, SNEB' ≤ R, SNEQ' ≤ R, then return to step 3 and replace SN2 by SN3. If there is no such SNE, the node SN1 will not be switched off, as shown in Fig. 10.

4.3 If there is not any nodes in the area Q'SN1W: Assume that there is a node SNA which is outside the area Q'SN1W, as it is shown in Fig. 11, and SNASN1 < 2R, SNASN2 ≤ 2R, and SNAQ' < R. The C(SNA) and the C(SN1) intersect at C' and C' is in the area Q'SN1W. Then, go to step 4.3 and replace Q' and SN2 by C' and C". If Q" is in the next circle, then node SN1 can be switched to sleep mode. If there is no such node SNA, the node SN1 can not be switched off.

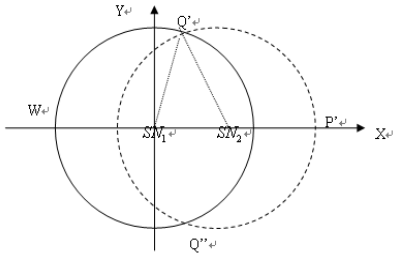

Figure 7. Put both nodes in the coordinate system

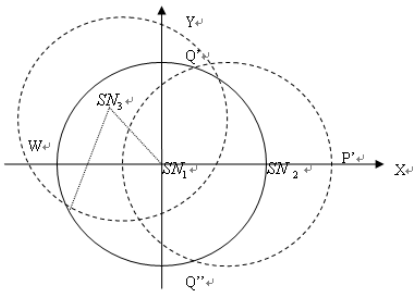

Figure 8. Step 4.1

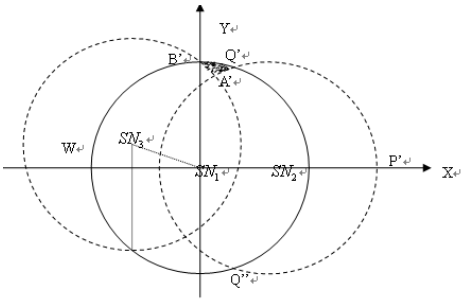

Figure 9. Step 4.2

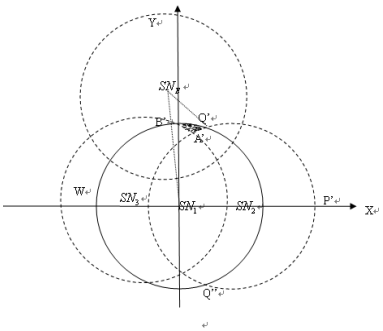

Figure 10. Step 4.2

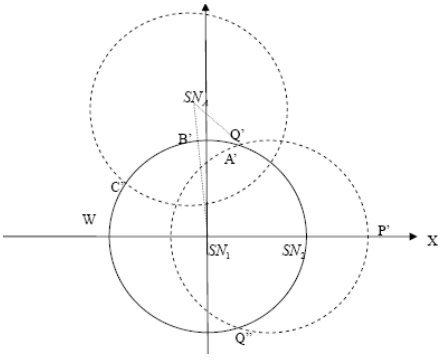

Figure 11. Step 4.3

Notion 3: If there exists a set of nodes $R = \{SN_{m1}, SN_{m2},\ldots SN_{mn}\}, n \in N$, for any $SN_{mk}(1 \le k \le n)$ that SN_{mk} is a redundant node, but if any node $SN_{ml}(1 \le l \le n)$ from among them is switched off, there will be at least one node $SN_{mp}(1 \le p < n)$ that is no longer redundant, then the node SN_{ml} and all SN_{mp} are defined as redundant related nodes (RR nodes or RRNs) and SN_{ml} is defined as redundant related seed (RRS).

Especially, there may be only one SN_{mp} related with SN_{ml} and both of them are RRS for each other. Then, these two nodes are defined as twin redundant related nodes (TRRNs), shown in Fig. 12.

Theorem 3: Only one of the TRRNs can be switched off.

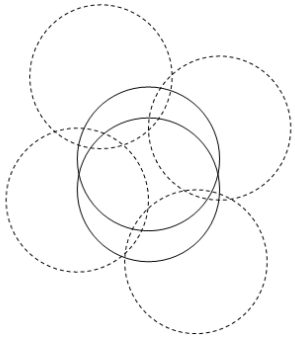

Figure 12. Twin redundant related nodes

The algorithm of switching the RR nodes is to separate them into TRRNs. First, select a RRS from RR nodes and search all of the SN_{mp} that whether there is a twin node for the RRS. If there is, switch one of them off and put the other one back. Besides, the selection algorithm, from step one to step four, does not cover all coverage probabilities because the computation overhead still need to be considered.

The selection for the TRRNs is based on the Variable Threats Probability Estimation algorithm which is specified in the following section. The higher threat probability estimation node will be switched off.

Besides, for any sleeping node SN_s, let a point in the deployment area be Z=(x, y). If there exists $\exists Z,\ |ZSN_i| > R, |ZSN_s| \le R, i \ne s$, SN_s will be switched on.

B. Variable Threats Probability Estimation

"There is no sure and efficient way to readily detect a node capture." The authors mentioned in [10] and [11]. Accordingly, a threats probability estimation algorithm is employed in the self-organization scheme.

The algorithm proposed here is an extension to the one originally described in [7], in which variable attack and failure probability will be involved. Consider that sensor nodes are deployed in groups, as shown in Fig. 3. The authors [7] proposed an algorithm based on static failure or attack probability for key predistribution. In their study, they also consider that a sensor node can play the role of data sink.

In our proposed method, a new deployment model is used. Compared with that in [7], the attack and failure probability does not focus on groups but on individual sensors. A new algorithm is proposed to keep the whole network connected and avoid key threats.

In the paper, we assume that the data sink is chosen before the network is deployed and will not be replaced. It is the same with cluster heads. Moreover, my proposed algorithm focuses on sensor nodes rather than on deployment groups which was proposed in [7]. The sensor nodes will not distinguish between an attack from a neighbor and one from an adversary. For security of Trusted Neighbors, I refer to *Reputation-based Framework for High Integrity Sensor Networks* [10].

In our proposed algorithm, the sensor nodes reporting to the same cluster head are defined as in one group. Assume that all deployment groups are in the same 2-dimensional area. Assume that all sensor nodes will automatically record threats' times and levels. Assume that only sensor nodes will be under threat. The algorithm for cluster heads can be derived similarly and will be discussed in the future work section.

In the following, the estimation of node threat level is described. It follows the general PID controller principle [19], and adjusts the estimate threat level according to three correction factors. The whole algorithm can be described as: if there are no threats detected by a node, the threat value of this node will drop gradually; if threats are detected by a node, there will be a correction value (derived from three correction factors) added on this node's threat value to estimate the threat level.

In my proposed algorithm, a sensor node $S_i (i \in N)$ is associated with a basic failure coefficient b_{S_i} representing the threat from the environment and b_{S_i} is predistributed value by experience. Let w be the variable attack and failure weight, which is obtained by experience. Let φ be threshold threat level and let D_φ be the detected attack level. Then w is defined as

$$w = \begin{Bmatrix} w_1 & , D_\varphi = \varphi_1 \\ w_2 & , D_\varphi = \varphi_2 \\ \vdots & \vdots \\ w_n & , D_\varphi = \varphi_n \end{Bmatrix}, n \geq 1 \tag{5}$$

Here n denotes the number of threat levels. Let $\{\varphi_1, \varphi_2, \cdots, \varphi_n\}$ be a set of threshold values and N_T shows how many times a certain threat level has been detected. For a certain threat level, the failure estimate is $F = wN_T$. The difference between the failure estimates at time k and that at time (k-1) can be defined as

$$e(k) = F_{S_i}(k) - F_{S_i}(k-1) \tag{6}$$

which is the first correction factor. Here $F_{S_i}(k)$ denotes the estimate failure of S_i at time k. Then the second correction factor M_c, relative threat value, can be defined as

$$M_c(k) = \theta \frac{F_{S_i}(k)}{F_{C(S_i)}(k)} \tag{7}$$

where θ is a constant coefficient. $F_{C(S_i)}(k)$ denotes the sum of failure estimate of S_i 's neighbors, which can be defined as

$$F_{C(S_i)}(k) = F_{S_i}(k) + \sum_{S_j \in C(S_i)} F_{S_j}(k) \tag{8}$$

The third correction factor is connectivity detection, which is the number of sensors that have at least an available path to a cluster head. It can be defined as

$$CD_{S_i}(k) = (1 - \frac{CNT_{S_i}(k)}{CNT_{S_i}})(CNT_{S_i} - CNT_{S_i}(k)) \tag{9}$$

Here $CNT_{S_i}(k)$ denotes the connectivity of S_i at the time k and CNT_{S_i} denotes the connectivity when the network first deployed. If $CD_{S_i}(k) > 0$, it means some nodes are compromised or unavailable at the time k.

The first correction factor measures the diversity between different time intervals. The second factor measures the relative threat. The third factor measures the connectivity for a sensor and its neighbors. The total correction measurement derived from (6), (7) and (9) can be defined as

$$M_A(k) = \omega \cdot (e(k) + M_c(k) + \gamma CD_{s_i}(k) + \frac{1}{\alpha}\int (e(k)) + M_c(k))dk + \beta \frac{d}{dk}(e(k) + M_c(k)) \tag{10}$$

Here α, β, γ are constants and will be set based on experience. Let δ , a constant, be the decreasing threat value, which means if no failure/attack problems are detected, b_{si} will gradually drop down. Then b_{si} can be defined as

$$b_{S_i}(k) = b_{Si}(k-1) - \delta + M_A(k) \tag{11}$$

If $b_{S_i}(k) < 0$, then let $b_{S_i}(k) = 0$. The $b_{S_i}(k)$ is the real time threat estimate and this will also be used in RNS algorithm for TRRNs.

TABLE I. PARAMETERS IN TPE

Parameters		Formula
0	Sensor failure coefficient	$M_c(k) = \theta \frac{F_{S_i}(k)}{F_{C(S_i)}(k)}$
ω	TPE coefficients based on experience	$M_A(k) = \omega \cdot (e(k) + M_c(k) + \gamma CD_{s_i}(k)$
β		$+ \frac{1}{\alpha}\int (e(k) + M_c(k))dk + \beta \frac{d}{dk}(e(k) + M_c(k)))$
γ		
ω	Amendment coefficient	

After real time threat estimate b_{si} is calculated, the degree of a sensor node can be derived. A degree of a sensor node denotes the number of available connections for a sensor node (number of shared keys with neighbors). Given a set of threshold $\{\varpi_1, \varpi_2, \cdots \varpi_n\}$, $n \geq 2$ for any sensor nodes, the degree can be calculated by

$$K(S_i) = \begin{Bmatrix} \kappa_1 & , b_{s_i} = \varpi_1 \\ \kappa_2 & , b_{s_i} = \varpi_2 \\ \vdots & \vdots \\ \kappa & , b_{s_i} = \varpi_n \end{Bmatrix}, n \geq 2 \tag{12}$$

Here $K(S_i)$ denotes the degree of a sensor node in a deployment group. In the formula (12), for $\forall i, j, i \neq j$, if $\varpi_i \geq \varpi_j$, then $\kappa_i \leq \kappa_j$. In my proposed method, differently from that in [5], the higher attack/failure probability a sensor has the fewer keys it has.

C. Organizing

In this section, a neighbor-oriented clustering and neighboring scheme is described which is supported by RNS and TPE. As it has been mentioned in the background chapter [13], the sensor nodes may be scattered randomly e.g. scattered from an airplane. Thus, there is no guarantee that all cluster heads can directly connect with the data sink or all group nodes can directly connect with their respective

cluster head. Routing with no threats is described in section *1)*, while in section *2)* variable threats are involved.

1) Normal Routing

In this section, the thesis considers the routing algorithm only between sensor nodes without any threats. To discover a primary cluster head (CH), each sensor node (SN) wants to discover the ID of the closest CH.

In our proposed scheme, let SNID be the ID of a SN, DP be Default Path, DD be the Depth of Default Path, DTD be Distance to Default Path, SLP be the status (sleeping or active), TL be Threat Level, NL be a Neighbor List. Then, we give each SN an expression (13):

$$SN_i = \{SNID, DP, DD, DTD, SLP, TL, NL(\omega)\} \quad (13)$$

Where ω is the index of SN's neighbors. SN's parameters are expressed with dotted notation, for instance, SN.DP denotes the SN's Default Path.

After deployment, the RNS algorithm (detailed in the next section) is activated. SNs which are redundant will be set to SLP = 1. At the same time, all SNs discover all their neighbors and store in NL. Then they will wait for connection from their neighbors which can connect to cluster heads and the algorithm is described as follows. At first, each CH will search SNs within its sensing range. Each SN within CH's sensing range will update its expression (14).

$$SN = update\{DP = CH, DD = 0, DTD, SLP = 0, NL(\omega)\} \quad (14)$$

Then these SNs continue to tell their neighbors they can communicate with a cluster head by sending a path message (15):

$$PathInfo = \{SNID = SN, DP = CH, DD = 0, DTD, SLP = 0\} \quad (15)$$

The SNs who receive the path message (15) will update their expression (16).

$$SN = update\{DP, DD, DTD, SLP)\} \quad (16)$$

If a SN receives more than one path message, it will calculate the power consumption (PC) on these different communication paths. As mentioned above, the proposed scheme is neighbor-oriented. The default path selection algorithm is described in (17).

$$SN = update\{PC(DP, DD, DTD)\} \quad (17)$$

Here PC denotes the power consumption function. Before give the expression for total power consumption, assume power consumption is proportional to the square of distance with a coefficient θ and each hop will consume λ. Thus, the power consumption function can be described as

$$PC(SN, NL(index)) = \theta \cdot (DTD^2 + NL(index).DTD^2) + \lambda \cdot DD \quad (18)$$

NL(index).DTD denotes the distance between a neighbor's default path and the neighbor. The paths from a SN to a CH are called communication links. For instance, SN_i need to send messages to SN_j, and SN_j need to forward to SN_k then finally to the CH, the link $i \to j \to k \to CH$ is called a communication link.

Theorem 2: For any SNs, if there exists $PC_1(SN,NL(index_1)) < PC_2(NL(index_2))$, the power consumption of the communication link on $NL(index_1)$ is less than that on $NL(index_2)$ is true.

Proof:

$$\begin{aligned} PC(SN, NL(index_1)) &= \theta \cdot (DTD^2 + NL(index_1).DTD^2) + \lambda \cdot DD \\ &= (\theta \cdot DTD^2 + \lambda) + (\theta \cdot NL(index_1).DTD^2 + (\lambda \cdot DD - 1)) \\ &= PC(SN) + PC(NL(index_1)) \end{aligned}$$

PC(SN) denotes the power consumption between the SN and its default path and $PC(NL(index_1))$ denotes the power consumption between the node of SN's default path and the node of SN's default path's default path. For any available communication links, we define a searching function (SF), that SF(SN) = $SN.NL(index_1)$ if for any n ($1 < n \leq N$, N is the number of SN's neighbor), there is $PC_1(SN,NL(index_1)) < PC_n(NL(index_n))$, where $SN.NL(index_1)$ denotes the SN's neighbor with $index_1$. We define $SF(SF(SN)) = SF^2(SN)$, then the most power-saving path will be the communication link $SN \to SF(SN) \to SF^2(SN) \to \cdots \to SF^{SN.DD+1}$. Thus, for any SNs, if there exist $PC_1(SN,NL(index_1)) < PC_2(NL(index_2))$, that the power consumption of the communication link on $NL(index_1)$ is less than that on $NL(index_2)$ is true.

Lemma 1: For any SNs, if there exists $PC_1(SN,NL(index_1)) < PC_n(NL(index_n))$,($1 < n \leq N$, N is the number of SN's neighbors), the $NL(index_1)$ is the minimum power-saving link for SN.

Based on Lemma 1, all SNs can find the minimum power path. If a sleeping node is on a minimum power path, then it will be switched on. An example of the normal routing is shown in Fig. 13.

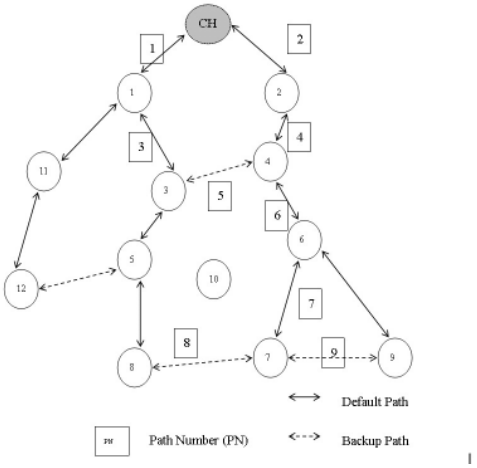

Figure 13. Normal routing

Fig. 13 shows the normal routing scheme. In RNS, SN3, SN4, SN5 and SN10 are candidates of redundant nodes. SN1 and SN2 are both in CH's sensing range and they have the default path DP = CH and DD = 0. Then, they start to send path messages to neighbors. Both SN3 and SN11 receive a path message from SN1 and a path message from SN4 is received by SN3 as well. Then based on Lemma 1, SN1 is set as the default path for SN3 and SN4 is set as backup. The rest may be deduced by analogy. After the default path searching, SN3, SN4 and SN5 are all on the minimum power link, thus, only SN10 will be switched off.

2) Routing with Variable Threats

In this section, variable threats are assumed, and attack threats and sensor failure will be discussed.

Based on TPE, each SN has a property called Threats Level (TL). In the routing scheme under variable threats, we assume there are three threat levels. Let level 0 be no threats, level 1 be low threats, level 2 be high threats. Thus, there are three different cases.

SN.TL = 0. Under this circumstance, the neighbor with TL = 0 and lowest power consumption is a priority selection. If there is no SN with TL = 0, the one with TL = 1 and lowest power consumption is a priority selection.

SN.TL = 1. Under this circumstance, the solution is the same with SN.TL = 0. For a more complex scenario, the balance between threats level and power consumption is mentioned in the future work.

SN.TL = 2. Under this circumstance, the SN has a high probability of failure or of being compromised. In the proposed method, TPE, this "high threat" SN may be totally isolated because it will be disconnected from "low threat" or "no threat" neighbors and only neighbors with TL = 2 can be used in the default path to the sink.

Besides, a SN will delete neighbors with TL = 2 from SN.NL.

For a failed SN, its neighbor will delete its index from SN.NL. If a CH becomes unavailable, such as due to physical damage or battery exhaustion, the SNs in its group will join another group using a backup path. Fig. 14 shows the scheme under variable threats.

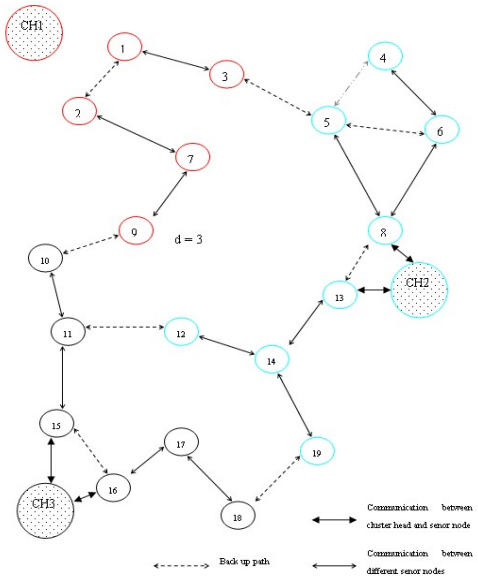

Figure 14. Routing with variable threats

In Fig. 14, we assume SN5 has a higher threat level than SN4, thus, SN4 will select SN6 as default path. If SN5 has a TL = 2, it will be isolated by SN4, SN3, SN6 and SN8 in order to avoid key and packet loss. If assume that CH1 is unavailable and SN5 has a normal threat level, then the backup path between SN1 and SN2, SN3 and SN5, SN9 and SN10 will be set as default path and all SNs in group 1 will join group 2 or group 3.

D. Sensors Addition

Throughout the lifetime of a WSN, it may be necessary to deploy additional SNs. In my proposed scheme, the network is flexible to receive additional SNs. We assume the additional SNs are randomly deployed in the monitored area. Based on the scheme specified and Lemma 1, new SNs will join a suitable group.

E. Proposed Scheme

The *3.A*, *3.B* and *3.C* will work as an integrative and the entire proposed scheme is outlined as follows: after scattering on the wild area, the RNS is activated. Each cluster head will search the data sink within its communication range. When a cluster head receives an available path to the data sink, 1) the cluster head will start to search other cluster heads within its sensing range (neighbors); 2) the cluster head will start to search sensor nodes around it to organize a group; 3) the sensor nodes will calculate the threshold $b_{S_i}(0)$. If a sensor node receives no available path or message to become a group member, it will be switched to sleep mode. If a redundant node is the only available path or the minimum power path for a sensor node, it will not be switched off.

If the environment changes: 1) if a cluster head becomes unavailable, all of its group members will join other groups via the algorithm specified; 2) if a sensor node SN has detected a high threat level, all sensor nodes connecting to SN will reroute to the first backup path; 3) if new sensor nodes join the network, they will follow *3.C*; 4) in case of a node pair (S_i and S_j), when only one of the pair can be switched to sleep mode, the following calculation is used. If S_i's $b_{S_i}(k)$ is higher than the sleep one S_j's, S_i will be switched to sleep mode and S_j will be waked up, vice versa.

Algorithm: Proposed Scheme

Sensors do RNS algorithm;
for $CH_i (1 \le i \le N)$ **do**
  search data sink;
  if CH_i find data sink **then**
    register at the data sink(CH_i);
    search neighbors(CH_i);
    search sensor nodes(CH_i);
  end if
end for
for $S_j (1 \le j \le M)$ **do**
  if S_j can access to a cluster head **then**
    search neighbors(S_j);
    organize network;
  end if
end for

for $S_j(1 \le j \le M)$ **do**

if RNS(S_j)is true and S_j is not the only path for another node **then**

switch(S_j,sleep);

end if

end for

if $CH_i(1 \le i \le N)$ cannot access to data sink **then**

switch(CH_i,sleep);

end if

if $S_j(1 \le j \le M)$ cannot access to cluster head **then**

switch(S_j,sleep);

end if

for $S_j(1 \le j \le M)$ **do**

if S_j default path is sleep or unavailable **then**

load the top of the stack and set as default path;

end if

end for

Thus, after the deployment, the network will be automatically organized and redundant nodes will be switched to sleep mode to save power. When some nodes become unavailable, some of the sleep nodes will be set as replacements and maintain the network.

IV. Analytical and Simulation Results

In this section, simulation results are given for total energy consumption and total packages lost in the network. The latter indicates network robustness, that is, the ability of the network to continue operating after variable threats.

In the simulation, a JAVA based wireless sensor network simulator was used. Compared with other simulators, such as OMNeT++, the JAVA simulator proved to be more flexible for environment configuration and implementation of the proposed solutions.

A. The Impact of Sensor Failure on Network Integrity

The proposed model, RNS can reduce partitioning in the network. An example illustrating the RNS algorithm when some sensor nodes are unavailable is shown in Fig. 16. In this example, there are 15 sensor nodes, the topology and deployment group is shown in Fig. 16. If the sensor nodes in the ellipse are unavailable, the default path will be blocked. However, the node which is switched to passive and sleep mode is not affected by the attack, and when the default path for the first forwarding sensor is blocked the sleeping node will be woken up to reconnect to the cluster head.

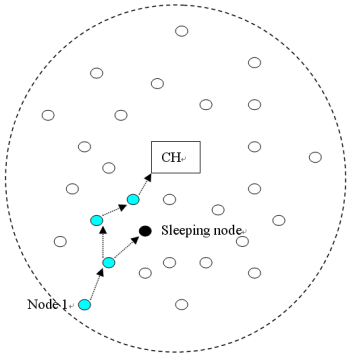

Figure 15. Message delivery from a sensor to the cluster head

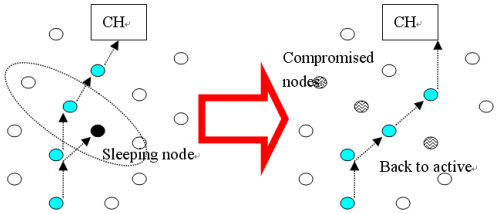

Figure 16. The impact of sensor failure

However, the problem of resistance to attacks becomes more delicate when the location of cluster heads is taken into consideration, and this will be detailed in the following section.

B. Resilience Against Node Failure

In this section, resilience against node failure between deployment groups is described.

An example illustrating the RNS algorithm in the case a cluster head fails is shown in Fig. 17. In this example, first, cluster head 1 is damaged and default paths for node 1 and 2 are blocked. In my proposed scheme, the isolated nodes will search for an available path to a new cluster head in order to join a new deployment group. As shown in Fig. 17, node 9 is switched to its backup path that connects to node 10, and node 7 and node 9 will join to cluster head 3, meanwhile, node 2, node 1 and node3 will join to cluster head 2. If cluster head 2 fails, all these sensor nodes will join group 3 via the backup path. Although this may raise communication overhead, it can maintain coverage over the whole monitoring region.

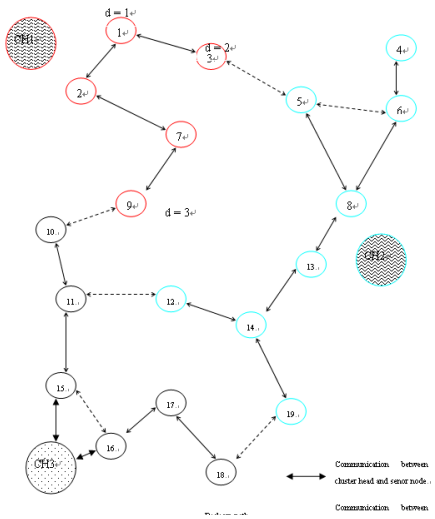

Figure 17. Resilience against node failure

C. Test Environment

In this section, simulation results are given for total energy consumption and total packages lost in the network. The latter indicates network robustness, that is, the ability of the network to continue operating after variable attacks.

In the simulation, a JAVA based wireless sensor network simulator was used. Compared with other simulators, such as OMNeT++, the JAVA simulator proved to be more flexible for environment configuration and implementation of the proposed solutions. The tests can be grouped into two parts. The first part describes the total energy consumption before and after implementing the Redundant Nodes Selection scheme (RNS). The second part depicts the robustness of the sensor network under variable threats with the proposed Threats Probability Estimation Scheme and with the shortest path first (SPF) scheme [9]. Robustness is examined with three different kinds of sensor distributions. Sensor Density in uniform distribution is 8 and in other distribution is 6, because in the simulator, the uniform distribution has a fixed template, and the number of sensors was the same in all simulation.

D. Power Consumption

Uniform sensor distribution and uniform traffic distribution was used when examining the total energy consumptions between before and after implementing proposed method. In the simulator, each hop costs 0.0005% battery life, around 0.00006% per simulation meter and 0.001% per working sensor.

Simulations with different parameters produced similar results. A typical set of results is presented here. There were 100 sensors with three cluster heads and sensor density is 8. The sensors were arranged in a 10 × 10 matrix. Fig. 18 shows the total battery consumption without proposed scheme and Fig. 19 shows the total consumption under the same circumstance but with self-organization scheme activated.

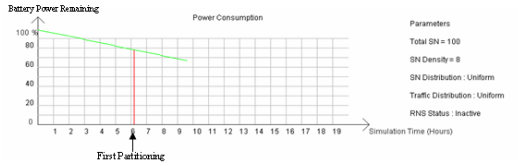

Figure 18. Total power consumption in uniform distribution without RNS

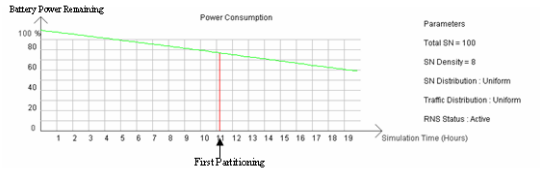

Figure 19. Total power consumption in uniform distribution with RNS

Then we look at the differences in normal distribution. Fig. 20 shows total power consumption without RNS scheme. Fig. 21 shows the total consumption under the same circumstance but with RNS activated. In the scenario with Zipf distribution, Fig. 22 and Fig. 23 show the improvement.

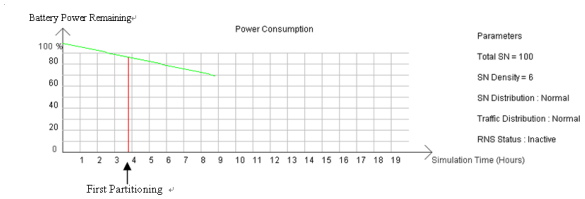

Figure 20. Total power consumption in normal distribution without RNS

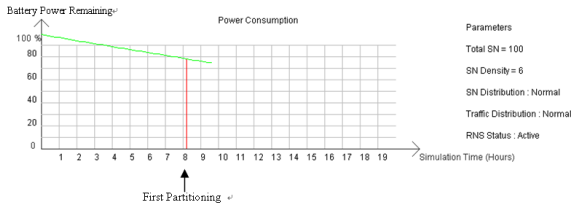

Figure 21. Total power consumption in normal distribution with RNS

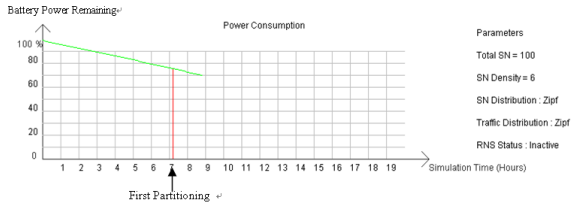

Figure 22. Total power consumption in Zipf distribution without RNS

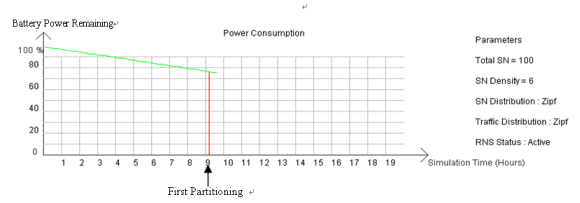

Figure 23. Total power consumption in Zipf distribution with RNS

The charts prove that the RNS scheme could reduce energy consumption of the network, and thereby extend the network's lifetime.

E. *Robustness*

Node captures in hostile environments are inevitable. Robustness is a kind of ability to help WSN recover from variable threats. This thesis describes a Threats Probability Estimation (TPE) scheme to support the key management method described in Chapter *3.B.* For comparison, the default key management scheme is shortest path first (SPF). There are total 100 sensors were deployed in such area and average sensor density is 6. One fourth of the sensors are set as "high threats" and random attacks are launched six times in each scenario. In the simulator the process of detecting an attack was not modeled, but rather the attack event was directly passed on to the sensors.

1) Normal sensor distribution with normal traffic distribution

First, we look at the scenario when sensors are deployed in a normal distribution (Gaussian distribution). The traffic distribution is also normal, to simulate centralized events, such as fire in a forest. The Fig. 24 shows the robustness of the network with SPF scheme and Fig. 25 shows that with TPE algorithm.

The curves illustrated above indicate the TPE scheme could enhance the robustness of the network, by reducing packet loss from 15% to 2%.

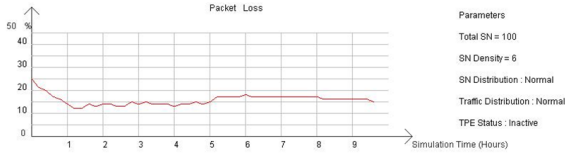

Figure 24. Robustness of the network with SPF in normal distribution

Figure 25. Robustness of the network with TPE in normal distribution

2) Uniform sensor distribution with uniform traffic distribution

Fig. 26 shows robustness of the network with SPF scheme when sensor deployed and network traffic follows uniform distribution. Fig. 27 shows the robustness of the same circumstance but with the same distribution, but with TPE activated.

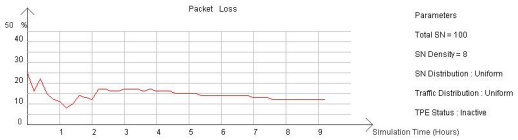

Figure 26. Robustness of the network with SPF in uniform distribution

Figure 27. Robustness of the network with TPE in uniform distribution

In uniform distribution, the TPE scheme improves the robustness by reducing packet loss from 14% to 4%.

3) Zipf sensor distribution with Zipf traffic distribution

Fig. 28 shows the robustness of the network with SPF when sensor deployed and network traffic follow Zipf distribution. Fig. 29 shows the robustness of the same circumstance but with the same distribution, but with TPE activated.

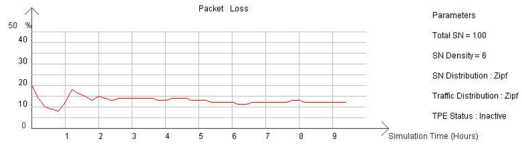

Figure 28. Robustness of the network with SPF in Zipf distribution

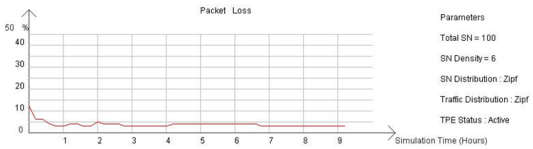

Figure 29. Robustness of the network with TPE in Zipf distribution

In Zipf distribution, the TPE scheme improves the robustness by reducing packet loss from 13% to 4%.

With high threats in the network, TPE can be considered as an effective scheme to improve the robustness by reducing packet loss.

4) Sensor Addition

In the scenario with proposed method, new sensors were successfully added to the network as long as there was at least one available "no threat" neighbor or "low threat" neighbor. Obviously, adding new sensors will improve network connectivity and coverage.

Discussion and evaluation will be detailed in the following Chapter.

V. DISCUSSION AND CONCLUSION

Power consumption and robustness are two important aspects of self-organization for wireless sensor networks. Throughout the lifetime of a wireless sensor network, a partitioning inevitably occurs after a certain period of time, because of finite battery power. Extending the battery life and postponing the occurrence of the first partitioning are the problems of the former aspect. The latter aspect describes the ability of a WSN to recover from different attacks.

In the proposed method, Redundant Nodes Selection scheme (RNS) and variable Threats Probability Estimation scheme (TPE) are designed to improve network performance on these two aspects. RNS is proposed to select redundant nodes and switch them off to save power, and set them back to active mode when a partitioning occurs. A node is redundant if after switching it off, its sensing area is still covered by neighbors. TPE is designed to enhance the routing algorithm and avoid high threats sensors in order to reduce packet loss.

The experimental results presented in the previous chapter showed that the battery life is extended dramatically. In uniform distribution, the total remaining battery power dropped to 80% after around 6 simulation hours without RNS, while under the same circumstance but with RNS activated, it took approximate 9 simulation hours, which represents a 50% improvement. The figures show that the battery consumption curve for the scenario with RNS has a gentler slope. Besides, from about 6.25 simulation hours, in the scenario without RNS a partitioning appeared because of sensor exhaustion. After that, connectivity and coverage also went down. However, in the scenario with RNS, sleeping nodes replaced the exhausted ones and maintained connectivity, and the first partitioning occurred at around 11.25 simulation hours, which is an 80% improvement.

Similarly, in normal distribution, the total remaining battery power dropped to 80% after around 5.75 simulation hours without RNS, while under the same circumstance but with RNS activated, it took approximate 7.5 simulation hours, which represents a 30% improvement. In addition, it took approximate 3.75 simulation hours to meet the first partitioning in the scenario without RNS. Compared with that, scenario with RNS has around 8.25 simulation hours without partitioning, which is a 140% improvement.

In Zipf distribution, the total remaining battery power dropped to 80% after around 6 simulation hours without RNS, while under the same circumstance but with RNS activated, it took approximate 8 simulation hours, which represents a 33% improvement. In addition, it took approximate 7.25 simulation hours to meet the first partitioning in the scenario without RNS. Compared with that, the scenario with RNS has around 9.25 simulation hours without partitioning, which is a 27% improvement.

For the implementation of RNS, redundant nodes are switched off to save power and then switched on to replace power-exhausted sensors. This can help the WSN to extend battery life and postpone the occurrence of first partitioning. The calculation for the redundant nodes was only executed once, after network deployment. Thus, there was not much computation added during runtime and we ignored the computation overhead in the simulation.

On the other hand, variable threats, such as node capture attacks and nodes failure in hostile environments are inevitable. A sensor with high threat level indicates a high packet loss probability when packets are received or sent. TPE scans all neighbors and helps sensors to avoid high-threat nodes to reduce packet loss. The scenario with my proposed scheme, TPE, is also showed a distinct improvement. In the simulation with sensor normal distribution, the scenario without TPE had approximate 15% packet loss on average while in the network with TPE, it had around 3% packet loss during simulation times, which is a 400% improvement.

Similarly, in uniform distribution, the scenario without TPE had approximately 14% packet loss on average while in the network with TPE, it had around 4% packet loss during simulation times, which is a 250% improvement.

In Zipf distribution, the improvement is also significant. I observed 13% packet loss in the scenario without TPE while under the same circumstance but with TPE, it had 4% packet loss, which is a 225% improvement.

In the simulation, the level of packet loss in the scenario without TPE usually stayed at a high level, although it fluctuated sometimes. On the other hand, in the scenario with TPE it always dropped quickly to a low level and stabilized, despite sometimes having a small rise at the beginning. TPE helps routing by avoiding higher threat neighbors, thus, a lower packet loss is obtained when variable threats are involved.

In my proposed method, the improvement by RNS depends on sensor density, the higher the density, the more improvement. Low sensor density networks will not benefit significantly from RNS. TPE is designed to counter variable threats and there will not be much improvement on the scenario without variable threats. Also, TPE may slightly raise the communication overhead and memory overhead because of rerouting to a safer neighbor. When a network was in low threat, the communication overhead was the same as the scenario without TPE. However, as the threat level goes up, the communication overhead was rising. In the simulation, the communication overhead was approximately 0% to 12% more than in the scenario without TPE, which was the price for significantly lower packet loss.

ACKNOWLEDGMENT

I would like to thank my parents and my friend Rui Rui Zhang for their understanding and support.

REFERENCES

[1] Jian Zhong and Peter Bertok, "A Variable Threats Based Self-Organization Scheme for Wireless Sensor Networks," 3rd Intl. Conf. on Sensor Technologies and Applications, 2009 (SENSORCOMM '09), pp. 327-332, doi: 10.1109/SENSORCOMM.2009.57

[2] J. A. Stankovic, "Self-organizing Wireless Sensor Networks in Action," J. of Networking and Mobile Computing, vol 3619/2005, doi: 10.1007/11534310_1

[3] B. Karp, "Geographic routing for wireless networks," PhD Dissertation, Harvard University, October 2000. URL http:// actcomm.dartmouth.edu/papers/karp:paper.pdf

[4] R. Di Pietro, L. V. Mancini and A. Mei, "Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks," Intl. J. on Wireless Netowrks, vol 12, pp 709-721, doi: 10.1007/s11276-006-6530-5

[5] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," In Proc. of the 10th Annu. Network and Distributed Sensor Networks (NDSS03) , 2003.

[6] B. Wang, K. C. Chua and V. Srinivasan, Wei Wang, "Sensor density for complete information coverage in wireless sensor networks," Intl. J. of EWSN2006,vol 3868/2006, pp. 69-82, doi: 10.1007/11669463.

[7] C. Yu, C. Li, C. Lu, D. Lee and S. Kuo, "Attack probability based deterministic key predistribution mechanism for non-uniform sensor deployment," In Proc. of the 27th Intl. Conf. on Distributed Computing Systems Workshops, pp 18, doi: 10.1109/ICDCSW.2007.24.

[8] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks," University of California at Berkeley, 2002. URL http://webs.cs.berkeley.edu/papers/sensor-route-security.pdf

[9] M. W. Chorzempa, "Key management for wireless sensor networks in hostile environments," Paper submitted to the Faculty of the Virginia Polytechnic Institute and State University, 2006. URL http://scholar.lib.vt.edu/theses/available/etd-05022006-171402/unrestricted/chorzempapaper_v2.pdf

[10] Saurabh Ganeriwal and Mani B. Srivastava, "Reputation-based Framework for High Integrity Sensor Networks," SASN'04, October 25, 2004, Washington, D.C., USA

[11] G. Jolly, M. Kusu, and P. Kokate. "A hierarchical key management method for low-energy wireless MSN networks," In Proc. of the 8th IEEE Symposium on Computers and Communication (ISCC), pp. 335–340. Turkey, July, 2003.

[12] S. Zhu, S. Setia and S. Jajodia. "LEAP: efficient security mechanisms for large-scale distributed MSN networks," In Proc. of the 10th ACM Conf. on Computer and Communication Security (CCS), pp. 62-72, Washington DC, October 2003.

[13] B. Krishnamachari, "An introduction to wireless sensor networks," Tutorial Presented at the Second Intl. Conf. on Intelligent Sensing and Information Processing (ICISIP), Chennai, India, 2005.

[14] F. Dressler, B. Krüger, G. Fuchs and R. German, "Self-organization in sensor networks using bio-inspired mechanisms," In Organic-computing, 2005. URL http://www.organic-computing.org/ conferences/arcs2005/dressler-article.pdf

[15] B. Atakan and Özgür B. Akan, "Immune system-based energy efficient and reliable communication in wireless sensor networks," In Next Generation Wireless Communications Laboratory, 2007. URL http://www.springerlink.com/index/14g2255603256jn6.pdf

[16] G. Jolly, M. Kusu, and P. Kokate, "A hierarchical key management method for low-energy wireless MSN networks," In Proc. of the 8th IEEE Symposium on Computers and Communication (ISCC), pp. 335–340. Turkey, July, 2003.

[17] S. Zhu, S. Setia and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed MSN networks," In Proc. of the 10th ACM Conference on Computer and Communication Security (CCS), pp. 62-72, Washington DC, October 2003.

[18] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," Cyber Defense Laboratory, Department of Computer Science, North Carolina State University, 2003

[19] *Documentation for the Bytronic Process Control Unit version 3*, Bytronic International Ltd., 2002.

[20] F. Xiujun, W. Fan, T. Bihua and L. Yuanan, "Wireless sensor networks security analysis," School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, 2007 URL http://www.paper.edu.cn/downloadpaper.php? serial_number=200704-597