ENERGY EFFICIENT GEOGRAPHIC ROUTING ALGORITHMS IN WIRELESS SENSOR NETWORK

TSENG-YI CHEN*, HSIN-WEN WEI†, CHE-RUNG LEE*, FU-NAN HUANG‡, TASN-SHENG HSU§ and WEI-KUAN SHIH

*Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
{tychen, cherung, wshih}@cs.nthu.edu.tw

†Department of Information Management, Tamkang University, Taipei, Taiwan
hwwei@mail.tku.edu.tw

‡Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan
simon@rtlab.cs.nthu.edu.tw

§Institute of Information Science, Academia Sinica, Taipei, Taiwan
tshsu@iis.sinica.edu.tw

Received 20 December 2012
Revised 20 January 2013

Over the past decade, energy efficiency has consistently been a critical research topic in the field of wireless sensor networks. In wireless networks, signal interference often leads to power waste in a sensor node. Several SINR-based routing algorithms designed for energy efficiency or interference avoidance had been proposed. However, they are either too complex to be useful in practices or may slow in routing computation speed. In this paper, two energy efficient geographic routing algorithms (EEGRA) for wireless sensor network are proposed to address the power consumption issue while considering the routing computation speed. The first algorithm take the value of interference into the routing cost function, and uses it in the routing decision. The second algorithm transforms the problem into a constrained optimization problem, and solves it by searching the optimal discretized interference level. We adopt four geographic routing algorithms: GOAFR+, Face Routing, GPSR, and RandHT, in EEGRA algorithms and compare them with three other routing methods in terms of power consumption and computation cost for the grid and irregular sensor topologies. The experimental result shows that the EEGRA algorithms reduce energy consumption by 30-50% comparing to geographic routing methods. In addition, the time complexity of EEGRA algorithms is similar to the geographic greedy routing methods, which is much faster than the optimal SINR-based algorithm.
Keywords: Geographic routing; Energy-efficient; Wireless sensor network, SINR, Interference power.

1. Introduction

Recently, wireless sensor networks have garnered increasing attention, and have widespread applications in monitoring and surveillance in the military, civil industries, home automation, and traffic control fields. However, the limited power resources of sensor nodes still restrict the capacities of these wireless sensor networks, and thus people usually use topology control techniques to effectively minimize the number of active sensor nodes and extend the lifetime of the network. Otherwise, the interference often disrupts signal transmissions between a source and a receiver, and causes a sensor node to resend the message. The interference becomes a major factor of power wasted in sensor nodes.

Undoubtedly, wireless connections are less stable than wired networks because the wireless network is prone to radio signals attenuation, signal interference, and signal noises. However, routing in wireless networks inevitably requires coping with these problems with low reliable link connections. Moreover, some scenarios assume that sensor nodes in the wireless network are mobile devices. Hence, the deployment of a wireless apparatus in the wireless network is more difficult than that in wired networks. Conventional routing algorithms that are applied to wired networks are not able to satisfy the requirements of a wireless network. Wireless routing algorithms should account for some potential factors such as signal disturbances and other dynamic characteristics.

The problem of an energy-efficient interference-based routing algorithm is to find out a feasible way that can effectively reduce the degree of interference and keep power consumption low in a sensor node. A good interference-based energy-efficient routing algorithm should have (a) robust interference resistance (b) a power-efficient routing design (c) a fast routing scheme (d) and a stable routing guarantee. There has been some solutions to solving interference and energy problems that arise from wireless sensor networks. First, communication that utilizes multi-channel capacity to reduce the signal interference was proposed in Ref. [4]. Nevertheless, this only applicable to the network supporting multi-channel and cannot be scalable well. Interference might still be generated if many communicating tasks transmit in the same channel. Second, algorithms using scheduling techniques to avoid the transmission of nearby SNs had been studied in Refs. [2, 3]. However, the scheduling needs to keep a lot of information such as sensor’s time slots and neighbor’s affection. It is not a dynamic method and not appropriate to practical WSNs environment. Third, in Ref. [5], the I2MR algorithm is proposed, in which the interference is characterized using the discrete graph model and avoided by cutting the adjacent edges of communicating SNs. Because of the removal of usable links, the transmission can be blocked. Last, in Ref. [6], Kwon and Shroff transformed the SINR to power consumption, and employed the shortest path algorithm to search the route with minimum energy cost. The method minimizes energy
consumption in WSNs that guarantees transmission quality, but the time complexity of routing calculation, solving the shortest path problem, is too high.

In this paper, we propose two energy efficient geographic routing algorithms. They are based on geographic routing algorithms Refs. [7, 8, 9, 10], because their low computation and storage requirements fit the WSN environments. The interference model used in the algorithms follows the approach proposed in Ref. [6], which measures the power consumption by signal-to-interference-and-noise (SINR). We implemented our algorithms by integrating several geometric routing algorithms, and the results show the proposed algorithms can reduce power consumption up to 50%, comparing to general geographic routing algorithms without any energy awareness. In addition, the time complexity of our algorithms is similar to the geographic routing algorithms, which is much faster than the existing optimal SINR-based algorithm. Moreover, we compared our algorithms with the I2MR algorithm, and measured routability by the number of available paths. The result indicates that our methods have much better routability than the I2MR algorithm.

The rest of this paper is organized as follows. In Section 2, we provide the background of the problem and algorithms, including problem formulation and related works. In Section 3, we introduce the interference model and a practical computational method for it. Section 4 presents the proposed energy efficient routing algorithms with analysis. In Section 5, the experimental setting and results are illustrated. The concluding remarks are given in Section 6.

2. Background

In this section, we introduce the problem formulation, the theoretical power model, and related work.

2.1. Problem formulation

In this paper, a wireless sensor network is modeled as a graph, denoted by $G(V,E)$, in which $V=\{v_1, v_2, ..., v_n\}$ is the set of sensor nodes and $E=\{e_1, e_2, ..., e_m\}$ is the set of transmission edges. Every transmission edge $e_i$ is defined by two sensor nodes $(T_{e_i}, R_{e_i})$ if they are in each other’s communication range. More precisely, two nodes can communicate directly if each can receive other’s signal with strength larger than some threshold $\delta_0$. The signal strength is measured by SINR, which implies that when there are interferences, stronger signals need be emitted to maintain the same SINR. We assume each SN can adjust their power to meet the required signal strength upon some limits.

The problem is to find a route of a given pair of the source node ($s$) and the destination node ($d$) such that the energy consumption is minimized. The problem formulation is given in (2.1), which minimizes the total energy consumption of a multi-hop message transmission,
where $R(s, d)$ is a route from the source node $s$ to the destination node $d$, $\Delta p(T_{e_i})$ the power consumed by transmitter node $T_{e_i}$ for each edge $e_i$ on the route $R$, $\theta(e_i)$ is the signal strength (SINR) of edge $e_i$, and $\theta_0$ is the required SINR. This formulation is the same as the one defined in 6. The relation of $p(e_i)$ and $\theta(e_i)$ will be explained in Section 2.2.

There are three factors to influence the power consumption. First, the routing distance from node $s$ to node $d$. However, the shortest one may not be the best. Figure 1 shows an example, which has three paths from $s$ to $d$. Path 1 (dashed lines) is the shortest, but the distance of each hop is double of that of Path 2 (solid lines). Because the power consumption is proportional to the square of distance for the same signal strength [11], the power consumption of Path 1 is actually double comparing to that of Path 2, although the number of hops of Path 2 is twice as many as those of Path 1. The hop distance of Path 3 is slightly shorter than that of Path 2, but its number of hops is much more than that of Path 2. Therefore, the path with minimum power consumption is Path 2.

Fig. 1. Three paths from node $s$ to node $d$ with different distance, routing power and interference affection.
2.2. The theoretical power model

Signal-interference-noise ratio (SINR) which measures the ratio of transmission signal strength to the interference and noise is used in our power model, because it considers the power spent on message transmission and interference together. For a transmission edge $e_i$, the sum of noise and interference at the receiver node $R_{e_i}$ can be expressed by the following equation,

$$I_f(e_i) = \sum_{m \neq i} G(T_{e_m}, R_{e_i})p(T_{e_m}) + \eta_i$$ \hspace{1cm} (2.2)

in which $G(T_{e_m}, R_{e_i})$ is the path gain between $T_{e_m}$: the transmitter on edge $e_m$, and $R_{e_i}$: the receiver on link $e_i$; $p(T_{e_m})$ is the transmission power of transmission nodes $T_{e_m}$; and $\eta_i$ is the ambient noise around receiver $R_{e_i}$. The path gain $G(T_{e_m}, R_{e_i})$ is usually a function proportional to the reciprocal of the square of distance between $T_{e_m}$ and $R_{e_i}$. Thus, the farther $T_{e_m}$ and $R_{e_i}$, the smaller $G(T_{e_m}, R_{e_i})$. According to (2.2), the SINR at edge $e_i$ is defined as
From (2.2) (2.3), one can see that when \( I_f(e_i) \) increases, to maintain the same SINR, the power \( p(T_{e_i}) \) needs to increase accordingly. Nevertheless, the increase of \( p(T_{e_i}) \) also enlarges the interference of other links. Therefore, other links also need to boost their power to maintain the same signal quality. The minimum power of each edge to maintain the required SINR can be obtained by solving the linear equation,

\[
(I - F)p = b,
\]

in which

\[
F(i, m) = \begin{cases} 
G(T_{e_m}, R_{e_i})c(e_i) & m \neq i, \text{and} \\
0 & m = i 
\end{cases}
\]

\[
b(i) = \frac{c(e_i)\eta_i}{G(T_{e_i}, R_{e_i})},
\]

where \( c(e_i) \) is the required SINR at edge \( e_i \).

Let \( p_T \) be the power consumption estimation made by the sensor node \( T_{e_i} \), and \( F_T \) and \( b_T \) the matrix and vector defined in (2.5). Let \( p'_T, F'_T \) and \( b'_T \) be the corresponding vectors and matrix for the case that assuming edge \( e_i \) is occupied. The increased power consumption for using edge \( e_i \) is

\[
\Delta p_T^{(i)} = p'_T - p_T \approx (I - F_T)^{-1}(\Delta F_T p_T + \Delta b_T).
\]

where \( \Delta F_T = F'_T - F_T \) and \( \Delta b_T = b'_T - b_T \).

Although this model is used in many related problems in wireless networks 61213, several difficulties will encounter when applying it directly. First, this model needs a matrix that keeps the information of path gains between all pairs of edges, which change dynamically and therefore frequently updates are required. Second, to obtain the solution, a big linear system needs be solved. Although distributed iterative methods for solving large linear systems are available, the required computation and communication are still too expensive for WSNs. Last, the goal of the power calculation is to estimate the impact of using a link for transmission, which has exponential many combinations. It is not practical to evaluate all possible combinations and to pick a route. A distributed and practical power estimation model will be presented in Section 3.

2.3. Related work

Routing algorithms in WSN have received massive research attention. From classical shortest path to geographic greedy forwarding, most algorithms focus on routability, computational cost, and routing distance, and only few of them are designed to minimize the power consumption or interference avoidance.
For interference avoidance WSN communication methods, diverse approaches had been proposed. In Ref. [47], the multi-channel technique is utilized to reduce the interference between the communication of different channel. The channel assignment is static and off-line, which may waste available bandwidth and restrict the possible routing paths. In addition, the number of channels is limited, which makes those methods poor in scalability.

In Refs. [2, 3, 18], authors proposed link scheduling methods that avoid the concurrent communication of nearby links. They can be viewed as another type of multi-channel method (time division). However, the problem is NP-hard, and the algorithms are not distributed and hard to satisfy the dynamic requirements. Although scheduling is essential to avoid concurrent sends and receives, practically, the timing control in WSN is very difficult. In this paper, we assume some scheduling method is applied, and the requested bandwidth of transmission has already taken the scheduling time into account.

Few interference avoid routing algorithms had been proposed. In Ref. [5], the communicating nodes and its neighbor nodes are blocked such that no concurrent communication can for communicating nodes and its neighbors. However, this method can cause low utilization of sensor nodes and may result poor routability. In Ref. [6], the authors obtained an optimal routing path based on the result of SINR transform. This algorithm resolves the problems occurred in Ref. [5] and reduces the power consumption in a sensor node. However, the computation of optimal routing path is based on the shortest path algorithm, which is expensive and unsuitable in WSN.

The proposed algorithms integrate several techniques mentioned above. First, the neighborhood identification of each sensor node is similar to the blocked nodes marking process proposed in Ref. [5]. However, we use this information in power estimation. Second, we referenced the power consumption and SINR formulation used in Ref. [6] to measure the link cost. Last, our routing algorithms are based on, but not limited to, the geographic routing methods. The following subsections reviews and illustrates the four routing algorithms, interference model, and topology model used in the experiments.

2.3.1. Geographic routing algorithm

Most greedy geographic routing algorithms decide a route from a source node to a destination node hop-by-hop. Each intermediate transmission node, including the source node, selects an adjacent sensor node that is proximal to the destination node. The distance is measured by the straight Euclidean distance from the transmission sensor node to the destination node. This greedy strategy works very well if the selected route does not have any traps, which are local minimums but not the destination.

Various geographic routing algorithms are designed to resolve the trapping problem. In Ref. [8], GPSR uses right hand rule to conquer local minima problem. The right hand rule delivers message to right hand neighbor when node is in a local minima. In Ref. [9], the face routing is introduced, which detects network graph feature and divides it into sub
network (face). Then, face routing algorithm transmits message face by face. It can avoid local minima and holes in WSNs. In Ref. [7], the GOAFR+ algorithm is proposed, which uses geographic greedy forwarding first, and resolves the trapping problem by using face routing. In Ref. [10], the RandHT algorithm is proposed to avoid hole problem. It uses geographic greedy forwarding in general situation. When encountering a local minimum, it splits the neighbored network into four stages and chooses landmark in each stage. Then, routing algorithm set the chosen landmark as temporary destination to detour round the hole in WSNs.

2.3.2. Sender-centric interference model

There are several kinds of interference models, such as the receiver node model Ref. [21] or k-hop interference model Refs. [18], [21]. The receiver node model assumes that interference is generated from the sensor node that is responsible for collecting data. The k-hop interference model supposes that two nodes separated by k-hop distance cannot transmit data at the same time. Hence, this model is prone to transmission delay. To formulate a suitable interference model, we define a sender-centric interference model.

**Definition 3.3:** (Sender-centric interference model) Let T and R be a source and receive node and the radius of a circle consists of its link. Its circumference represents the range of the interference.

However, in our system model, we assume that a node is able to receive and forward messages and we map the signal interference to transmission links. Additionally, the node within the interference range can also pass message, but it must consume more power to complete the transmission job.

2.3.3. Topology model

Gabriel topology model is able to effectively eliminate the signal interference that occurs frequently in a communication. We assume that there are three sensor nodes (T, A, and R) in a planer graph and the range of the interference between T and R is e, T and A is e’, and R and A is e” as shown in Fig. 3. Moreover, we suppose that the sensor node T is ready to pass a message to sensor node R through path TR. This routing decision exposes more sensor nodes to interference from e because the interference range of e is the biggest in Fig. 3. In order to organize a low interference routing path, we refer to the characteristics of the Gabriel graph [22] and organize a Gabriel topology model.
3. Interference Power

The theoretical power model presented in Section 2.2 faces two major computational challenges. First, it requires global information, and second, it needs to solve a linear system. Here we present a more practical SINR model and a computational method to estimate the required power of a single hop transmission. Similar idea had been proposed in literature, such as in Ref. [6].

3.1. Local SINR model

The model we proposed is called the local SINR model which approximates the theoretical one derived in (4). The local SINR model of a link only considers the interference caused by the communications in its neighborhood. The reason is the path gain \( G(T, R) \) decays quickly for far separated nodes, and therefore their interference can be ignored.

Fig. 3 An example of Gabriel graph. The red node is a witness within two yellow sensor nodes’ transmission range.

Fig. 4 Solid node is current routing node. The calculation elements are dashed link for local SINR calculation model in grid.
We define the neighborhood of an edge $e_i$ by the neighborhood of its transmitter sensor $T_{e_i}$. Let $N(T_{e_i})$ be the sensor nodes in edges in $T_{e_i}$’s neighborhood. An edge $e_m$ is in $e_i$’s neighborhood $N(e_i)$ if and only if $T_{e_m}$ and $R_{e_m}$ are in $N(T_{e_i})$. The range of $T_{e_i}$’s neighborhood is defined operationally. If a sensor node $A$ can receive the signal transmitted from $T$ to sensor node $B$ with signal strength larger than a threshold $\delta$, then sensor node $A$ is in the neighborhood of $T$. In another word, the set of neighborhood is larger than the adjacent sensor nodes.

In a grid topology, the transmission range is assumed to be the same for every transmission link. Therefore, we define the neighborhood of a sensor node as the nodes that within two-hop distance, as shown in Figure 4.

For irregular topology, the neighborhood of a sensor node $i$ is defined as a set of sensor nodes within a fixed distance from $i$. Practically, this neighborhood set can be located by sending probing signals. Initially, each sensor node $i$ can send a signal to discover its neighborhood. If its nearby sensor nodes detect the probing signal with strength larger than some threshold, then they acknowledge this probing signal and join node $i$’s neighborhood. Figure 5 shows an example of the neighborhood of a sensor node in an irregular topology.

![Figure 5](image-url)

Fig. 5 The calculation elements are dashed links for local SINR calculation model in random distribution.

### 3.2. Power consumption estimation

For each edge $e_i=(T_{e_i}, R_{e_i})$, we estimate it power consumption by

$$\Delta p_{e_i} = \frac{\sum_{m: m \in A(T_{e_i})} g(e_m) \times p_{e_m} + \eta_i}{g(T_{e_i} R_{e_i})}.$$  \hspace{1cm} (3.1)

where $A(T_{e_i})$ is a set of edges $e_m$ which are in $T_{e_i}$’s neighborhood and is active. An edge $e_m$ is called active if there is a message passing from $T_{e_m}$ to $R_{e_m}$ at the moment that $e_i$ is requested. The value $p_{e_m}$ is the power consumption required for $T_{e_m}$ sending messages to $R_{e_m}$. To measure $p_{e_m}$, the sensor node $T_{e_m}$ runs a test route to get this information in the beginning and passes the value to its neighboring nodes.
To calculate (3.1), sensor node $T_{e_l}$ sums up the $G(e_m) \times p_{e_m}$ for its active neighboring edge $e_m$. This information of $G(e_m) \times p_{e_m}$ for neighboring edge $e_m$ is maintained in a table in $T_{e_l}$. The active information is obtained via a protocol, as stated follows. First, by the broadcasting nature and the property of the neighbor nodes, a transmission originated by a node can be received by all its neighbor nodes, and therefore neighbor nodes can use the ID (or coordinates) of sender and receiver to identify which link is used. When a transmission ends, the transmitter needs to broadcast a special signal to inform its neighbors to update the information again.

The calculation of (3.1) can be done quickly since all the information can be gathered by table lookup. And because the number of neighboring nodes and edges is small, the cost to store the information and the cost to calculate (3.1) is cheap.

4. Energy Efficient Geographic Routing Algorithms

The problems of energy effective routing, defined in (2.1), can be viewed as multi-objective optimization problems, which need to minimize both the routing distance and the power consumption. Since the algorithms are distributed, all the information need be obtained from local. Section 3 describes how the power consumption information can be estimated locally. For the distance and routing information, we draw the support from the geographic routing algorithms, which utilize the Euclidean distance to measure the quality of next hopping node.

The remaining problem is how to combine those two kinds of information to achieve energy efficient routing. We introduce two algorithms, which are based on some geographic routing methods, to solve this multi-objective optimization problem.

4.1. EEGRA I

The first algorithm, called EEGRA I, merges two objectives by a weighted sum. For each edge, it defines a cost function by combining the distance to destination and the interference power:

$$w_i = \text{dist}(T_i, d) + \rho \Delta p_i,$$

where $\text{dist}(T_i, d)$ is the Euclidean distance from the transmitter node $T_i$ to the destination node $d$, $\rho$ is positive number, and $\Delta p_i$ is the increased power consumption defined in (3.1).

The EEGRA I algorithm is a distributed algorithm which employs greedy geographic routing algorithms to find the route based on the cost function defined in (8). The problem at each sensor nodes becomes to find a next hop with minimum $w_i$:

$$\min_{e_m \in A(T)} w_{e_m},$$

The procedure of how each sensor node responds after receiving a message is sketched in Algorithm 1.
Algorithm 1: Procedure of message handling for each sensor node of EEGRA I.

**Input:** A message containing the coordinate of the destination SN $d$.

**Output:** Next hopping node

**Algorithm:**

1. If $Current\_node \neq Destination\_node$
   1. Calculate link weight $w_i$ of the links around current node.
   2. Choose the node with the smallest weight as the next hopping.
   3. When it is trapped in a local minimum, resolve it by the geographic routing algorithm.
   4. Transmit message to next hopping node.

It can be seen that the procedure is just like most geographic routing algorithms, except the definition of cost function. Step 3 in Algorithm I varies for different used geographic routing algorithms.

The time complexity analysis depends on the used geographic routing algorithms, as well as the graph models they employed. Although we defined our graph by visibility, which is easier to define the power model, in routing other types of graphs, such as Relative Neighborhood Graph (RNG), Gabriel Graph (GG), or Restricted Delaunay Graph (RDG) Refs. [8, 19], may be used in the underlying geographic routing algorithms. Those graph models have better properties in the analysis of time complexity. To simplify our analysis, we give some assumptions on the power consumption, computed routes, and sensor node distribution.

**Assumptions:**

1. The maximum power consumption is bounded, $\Delta p_i \leq \varphi$.
2. In a route computed by EEGRA I, no cross links.
3. The length of edges on the path is normally distributed with mean $\ell$.

**Lemma 4.1:** If there is no local minimum in the route computed by EEGRA I, which means $w_1 > w_2 > \ldots > w_m$ for the cost functions along the computed route $\{s, t_1, t_2, \ldots, t_{m-1}, d\}$, and assumption (1) is satisfied, then $dist(t_i, d) < dist(s, d) + \rho \varphi$ for $i = 1, 2, \ldots, m-1$.

**Proof:** From the monotonic decreasing property of $w_i$, $w_i > w_{i+1}$, one has

$$dist(t_i, d) + \rho \Delta p_i > dist(t_{i+1}, d) + \rho \Delta p_{i+1}.$$  

If $dist(t_{i+1}, d) > dist(t_i, d)$, it can be written as

$$dist(t_{i+1}, d) - dist(t_i, d) < \rho (\Delta p_i - \Delta p_{i+1}) \leq \rho \varphi.$$  

Above relation can be extended to $dist(t_{i+k}, d) - dist(t_i, d)$ by induction if one only picks $t_j$ that makes $dist(t_j, d)$ increases,

$$dist(t_{i+k}, d) - dist(t_i, d) < \rho (\Delta p_i - \Delta p_{i+k}) \leq \rho \varphi.$$  

Setting $t_i = s$ in the above inequality, one can obtain the result. □

Lemma 4.1 shows if there is no local minimum, the distance of the nodes in the computed route to the destination is bounded. Thus, all the intermediate sensor nodes are
in the circle, centered at $d$, with radius $\text{dist}(s,d)+\rho \varphi$. Using the similar arguments in the proofs of 19, we have the following theorem.

**Theorem 4.2:** If there is no local minimum in the routing, and assumption (1)(2) are satisfied, then the length of the route computed by EEGRA I is of $O((\text{dist}(s,d)+\rho \varphi)^2)$.

With theorem 4.2 and assumption (3), we can get the expected number of hops for EEGRA I.

**Theorem 4.3:** If there is no local minimum in the routing, and all assumptions are satisfied, then the expected number of hops the route computed by EEGRA I is of $O((\text{dist}(s,d)+\rho \varphi)^2/\ell)$.

### 4.2. **EEGRA II**

The second algorithm, called EEGRA II, is to combine those two kinds of information by putting the power consumption in the constraints. Initially, we discretize the possible powers into several power-levels, and then guess a power level $P_{\text{max}}$ as the maximum power consumption in the routing paths, $P_{\text{max}} = \max_{e_i \in R(s,d)} \Delta p(T_{e_i})$. When the source sensor transmits a message, the information of $P_{\text{max}}$ is also sent with the message. Each intermediate sensor node $T$ only considers the edges whose power consumption is less than or equal to $P_{\text{max}}$, and chooses one feasible edge with the minimum distance to the destination. The procedure of how each sensor node responses is sketched in Algorithm 2.

**Algorithm 2.** Procedure of message handling for each sensor node of EEGRA II.

**Input:** A message containing the coordinate of the destination SN $d$ and the maximum allowed power consumption $P_{\text{max}}$.

**Output:** Next hopping node

**Algorithm:**

1. Choose the node with the smallest distance to destination $d$ whose $\Delta p(e_m) \leq P_{\text{max}}$.
2. When it is trapped in a local minimum, resolve it by the geographic routing algorithm.
3. Transmit message to next hopping node.

End If

This algorithm is similar to I2MR Ref. [5], which blocks edges that was affected by the interference or requires large power consumption. The difference is that when the routing fails, EEGRA II algorithm will try the next power level, which increases the maximum power consumption allowance, and the number of feasible communication edges. The high level description of EEGRA II is given in Algorithm 3.
Algorithm 3. The EEGRA II algorithm

**Input:** Source node $s$, destination node $d$, network graph $G(V,E)$, and power-levels $p_1 < p_2 < \ldots < p_k$.

**Output:** Energy-efficient routing path $r$

**Algorithm:**

1. For $p_{\text{max}} = p_1, p_2, \ldots, p_k$
   1. Block the edges whose power consumption are larger than $p_{\text{max}}$.
   2. Use the default geographic routing algorithm to find the routing with the use of feasible edges only.
   3. If a route is found, stop.

The routes found by the EEGRA II does not optimize the objective function (1), but the following one,

$$
\begin{align*}
\arg \min_{R \in (s,d) \subseteq E} \sum_{e_i \in R(s,d)} |e_i| \\
\text{subject to } & \theta(e_i) \geq \theta_0 \\
& \Delta p(T_{e_i}) \leq p_{\text{max}}
\end{align*}
$$

This objective function is not optimal in the global sense, which means it does not minimize the total power consumption of the entire sensor network. However, it makes more sense for a WSN, because it is to minimize the power consumption of each sensor node. Thus, the power consumption of each sensor node in a route can be constrained by $p_{\text{max}}$.

The feasibility of EEGRA II can be verified easily, because if we set $p_{\text{max}}$ equal to the maximum power $\Delta p$, which means no any restriction, EEGRA II goes back the default geographic routing algorithms it invokes. If we restrict $p_{\text{max}}$ to some very small values and only consider the interference, not the total power consumption, then it works like the I2MR algorithm, in which many links are blocked. The problem is how efficient this method is, in terms of time complexity and power saving.

**Theorem 4.4:** Let $O(T(n))$ be the time complexity of the geographic routing algorithm used in EEGRA II, and $k$ be the number of energy levels defined in Algorithm 3. The time complexity of EEGRA II is $O(kT(n))$.

**Proof:** The proof is straightforward from Algorithm 3. Since EEGRA II may try all possible energy levels, and for each energy level, it run the underlying geographic routing algorithm, the worst case time complexity is $O(kT(n))$. □

The performance of energy saving of EEGRA II really depends on the used geographic routing algorithm. Take the case in Figure 1 as an example. If the Face routing algorithm is used, it will choose Pass 3, which although has the smallest $\Delta p$ for each hop, but the total power consumption is the highest. We will validate the
effectiveness of power consumption for each accompanied geographic algorithms empirically in Section 5.2.2.

Another important factor that influences the power consumption is the number of power level. Intuitively, the more level of power constraints, the better power saving we can obtain because it releases just enough edges for data transmission. However, the real situation may be more complicated. For finer energy levels, the more number of iteration needs to run to find a feasible route. This does not increase the power on routing, but also the power in computation. In fact, according to our experiments, as reported in Section 5.3, sooner after the number of power-level exceeds a threshold, the power saving would not be improved.

5. Experimental Results

In this section, we present the results of experiments to scrutinize the routing path selection based on the EEGRA algorithms. Furthermore, we use simulations to compare the effectiveness of EEGRA algorithms with existing ones, in terms of time complexity, power consumption, and packet received ratios.

We implemented the EEGRA algorithms based on four geographic routing algorithms: GPSR Ref. [8], GOAFL Ref. [7], Face Routing Ref. [9] and RandHT Ref. [10], and other algorithms for comparison, including SINR-based Ref. [6], I2MR Ref. [5] and the four geographic forwarding algorithms Refs. [7, 8, 9, 10]. All the experiments and simulations are integrated NetSim2 and our own codes are developed in C.

5.1. Computing routing paths

In this subsection, we compared the routing paths of three types of routing algorithms: the EEGRA algorithms, the SINR-based algorithm [6], and the pure geographic greedy algorithms [7], in the case when there are some ongoing transmissions in the neighbor.

Figure 6 shows the routing paths calculated by four algorithms, in which there is an ongoing transmission, marked in the red solid line. And a new transmission is requested from the source node (solid square) to the destination node (solid circle). As can be seen, both EEGRA algorithms and SINR-based algorithm detour to avoid the ongoing transmission, Figures 5(a)-5(c), while the pure geographic greedy algorithm forwards messages to the nearest neighbor nodes.
5.2. Simulation

The experiments in this section aim to verify the performance, power-efficiency and routability of EGRA algorithms. Hence, we engage in some experiments that includes routing speed, power consumption, the integrity of received packets, the numbers of transmission tasks in network topology at same time.

The simulations consist of a random distribution network with 50 ~ 5×10^6 sensor nodes on a 1000 × 1000 unit plane. Moreover, we set the system parameters based on ZigBee™ IEEE802.15.4 20 specification, such as: 2.4 GHz frequency and 250 Kbps transmission bandwidth and some parameters used in Ref. [11]. We set ambient noise to 5×10^-8 and path loss exponent equal to 2. The numbers of routing task are ranged from 10^2 to 10^6 and every transmit task size is fixed at 100 KB, and the minima required SINR $\theta_0$ is fixed at 0.1.

5.2.1. Routing efficiency

In general, the EEGRA and the SINR-based algorithms enable to obtain a power-efficient path to forward messages. However, the SINR-based algorithm has time complexity $O(n^3)$ and requires global information. In contrast, the EEGRA algorithms, inherited from pure...
geographic routing algorithms, are fully distributed and of time complexity is similar to the used geographic routing algorithms.

Figure 7 compares the running time of 13 algorithms for different numbers of transmission tasks. The algorithms in comparison include four geographic routing algorithms: GPSR, RandHT, Face routing and GOAFR+, four EEGRA I algorithms with four geometric routing algorithms, four EEGRA II algorithms with four geometric routing algorithms, and the SINR-based algorithm. The number of sensor nodes is 1000, and the number of transmission varies from 100 to 1000.

The results show the running times of geographic algorithms and the EEGRA algorithms are similar, and grow almost linear with the number of transmissions. On the other hand, the SINR-based algorithm is the most time consuming algorithm and it scales poorly even with the number of transmissions.

In pure geographic routing algorithm, GOAFR+ is the fastest one in general. In Refs. [7] [20], GOAFR+ routing time complexity is optimal on the Gabriel Graph model in the proof. Face routing is the slower geographic routing algorithm in our experiment. Face routing time complexity is \(O(n)\) for \(n\) sensor nodes. Our experiments, with and without energy-aware, conform this theoretical result.

![Figure 7. Simulation time for routing algorithm the number of node = 1000.](image)

5.2.2. **Power-efficiency**

This experiment verifies the power efficiency of 13 routing algorithms, same as those in Section 5.2.1, with \(10^2\) to \(10^3\) transmission tasks for 1000 sensor nodes. The results are shown in Figure 8, by which one can see that the power consumptions of the routes...
calculated by four pure geographic greedy routing algorithms (they are overlapped in the figure) grow more rapidly, and reach 7500 mW for 1000 tasks. One the other hand, the power consumption of the route calculated by the EEGRA algorithms and the SINR-based algorithm slowly increased with the number of transmission tasks, and reach 4000 mW to 4800 mW for 1000 tasks.

In Figure 8, the SINR-based routing algorithm is of the best power efficiency. EEGRA algorithms can achieve similar power efficient routings. Among them, the EEGRA II algorithms perform better than the EEGRA I algorithm in terms of power efficiency.

![Figure 8. Power consumption (mW) for routing algorithm the number of node = 1000.](image)

Figure 9 presents how much energy saving is made by different energy efficient algorithms. The measurement is defined by the energy saving comparing to the power consumption of the geographic routing algorithms. Let $P_{gra}$ be the power consumption of a geographic routing algorithm and $P_{ee}$ be the power consumption of a energy efficient routing algorithm. To make comparison easier, we use relative energy saving, which is defined as

$$
\text{energy saving} = 1 - \frac{P_{ee}}{P_{gra}}
$$

(5.1)

We use the power consumption of GOAFR+ as the reference base, and compute the energy saving while using eight EEGRA algorithms and the SINR-based algorithm. As shown in Figure 8, the SINR-based algorithm is the best, which can save 40%-50% energy comparing to the pure geographic routings. EEGRA II algorithms, four of them,
can save 35% to 45% power consumption. EEGRA I algorithms are in the bottom, which can still save 25% to 35% energy.

In Figure 9, the most power ineffective geographic routing algorithm cooperated with EEGRA I and EEGRA II, is the RandHT. This phenomenon can be reasoned as follows. In energy efficient comparison, there are two factors: the routing distance and the interference, as shown in the example of Figure 1. The RandHT algorithm uses some congestion control technique to avoid congestion zone, which gives better routability, but the routes it computes may be longer than others. Since our algorithms avoid interference of transmitting nodes, which also serves similar functionality as congestion control, the advantage of the RandHT is not so effective. Therefore, the algorithms that choose shorter paths could save more power consumption.

5.2.3. The reliability of the routing

The routability of routing algorithms is the ability to find routes for given sources and destinations. In this experiment, we use packet arrival ratio to measure the routability of routing algorithms, which is defined as follows,
There are several reasons that make the sensor nodes unavailable for data transmission. The first reason is each sensor node cannot physically process more than one data transmission at the same time. Therefore, when there are more data transmission requested simultaneously, the number of available sensor nodes will drop. The second reason is caused by interference-avoid routing algorithms, such as I2MR, which will block some sensor nodes that near the working sensor nodes to avoid the interference. EEGRA II algorithm also blocks some sensor nodes according to the desired interference levels.

Figure 10 shows the results of various routing algorithms for different number of transmission tasks. The result can be divided into three groups. The first group is the pure geographic routing algorithms. The second group includes the EEGRA algorithms and the SINR-based algorithm. The third group only has one algorithm, which is the I2MR algorithm 5. As can be seen, the packet arrival ratio of the first group is about 85%, and is about 80% for the second group, but is only around 70% for the I2MR algorithm.

5.3. Power-level affection of EEGRA II

In EEGRA II, each routing task is assigned a minimal power-level $p_{\text{max}}$, and only the links with $\Delta p$ less than $p_{\text{max}}$ is usable. In this section, we experiment different power-level settings of EEGRA II algorithm to explore its affection of energy saving.
Figure 11 shows the experimental result of power consumption and running time of EEGRA II for using different numbers of energy levels. The number of sensor nodes and the number of tasks are fixed to 1000 and 1000 for all the different energy levels. The routes for different energy-level setting may be varied. As shown in the figure, the power consumption is decreasing as the number of energy-level increases, and converges after the number of power-level exceeds 10. The running time is increasing gently with the power-level. If one adds the power consumption of computation to the entire power measurement, there will be an obvious minimum of the power consumption curve.

6. Conclusion

This paper proposes two energy efficient routing algorithms, called EEGRA, which consider all three factors that affect the power consumption of routing: routing distance, interferences, and computational cost. The basic routing of EEGRA is driven by geographic routing algorithms, and the power term is added to the objective function and constraints for those two algorithms respectively. The experimental results show the EEGRA algorithm uses similar time as the geographic routing algorithms, and can also achieve similar power savings as the optimal SINR-based routing algorithm. The result shows that the EEGRA algorithms reduce energy consumption by 30-50% comparing to geographic routing methods. In addition, it has better routability comparing with the I2MR algorithm.
Acknowledgments

We would like to thank the National Science Council of the Republic of China (Taiwan) for financial support of this research under grant NSC101-2219-E-007-007, NSC100-2221-E-001-011-MY3 and NSC101-2221-E-032-067.

References


