Distributed Load Balancing Mechanism for Detouring Routing Holes in Sensor Networks

Jinnan Gao* Fan Li* Yu Wang†

* School of Computer Science, Beijing Institute of Technology, Beijing, 100081, China.

† Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA.

***Abstract*—Well known "hole" problem is hardly avoided in wireless sensor networks because of various actual geographical environments. Existing geographic routing protocols (such as GFG [1] and GPSR [2]) use perimeter routing strategies to find a detour path around the boundary of holes when they encounter the "local minimum" during greedy forwarding. However, this solution may lead to uneven energy consumption around the holes since it consumes more energy of the boundary sensors. It becomes more serious when holes appear in most of routing paths in a large scale sensor network. In this paper, we propose a novel distributed strategy to balance the traffic load on the boundary of holes by virtually changing the sizes of these holes. The proposed mechanism dynamically controls holes to expand and shrink circularly without changing the underlying forwarding strategy. Therefore, it can be applied to most of the existing geographic routing protocols which detour around holes. Simulation results show that our new strategy can effectively balance the load around holes thus prolong the network life of sensor networks when using with GPSR.**

***Index Terms*—geographic routing, load balancing, routing holes, wireless sensor networks**

I. Introduction

Due to its wide applications, wireless sensor network (WS-N) has recently emerged as a premier research topic. A WSN usually consists of a large set of sensor nodes spreading over a geographical area. Routing in such large scale wireless sensor networks is always a challenging task. One possible solution is geographic routing [3], [4]. Geographic routing (also called georouting or position-based routing) relies on geographic position information to make routing decision at each sensor node. With position information of the destination and surrounding neighbors, the message can be routed to the destination without knowledge of the whole network topology or a prior route discovery. This significantly improves the scalability of such routing protocols.

Greedy routing is the most popular and widely used geographic routing. In greedy routing, packets are greedily delivered to the neighbor which is the nearest one among the current node and all its neighbors to the destination. Greedy routing has been demonstrated to be very effective in large scale wireless sensor networks and can be adapted to topology changes dynamically. However, greedy routing fails to deliver the packet when it meets a node which cannot find a neighbor closer to the destination than itself. We call this problem *local minimum phenomenon*. Such situation often happens at the boundary nodes of topology holes in a wireless sensor network (such as at node u in Fig. 1), thus it is also known as "hole" problem of geographic routing. Due to various geographical environments in real-life applications of WSNs, the local minimum problem is an inevitable problem existing in most geographic routing protocols.

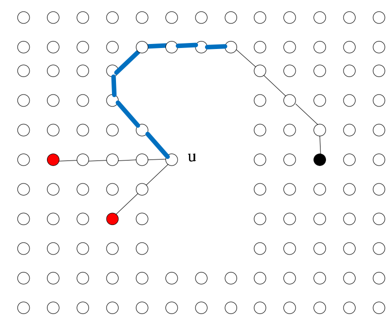


Fig. 1: **An example of detouring in geographic routing**. Red nodes are source nodes and black node is the destination. Local minimum of greedy forwarding occurs at node u. Then face routing routes the packets along the boundary of the topology hole, denoted by blue curve. Clearly, this leads to higher traffic load on the boundary.

Most geographic routing protocols have their own special methods to find a detour path when they encounter the "local minimum". Those methods can be grouped into two categories: face routing [1], [2], [5], [6] (based on right-hand rule) and back-pressure method [7]–[9]. In the methods with face routing, when greedy forwarding fails at a local minimum, data packets tend to be routed along the holes boundaries. In the back-pressure methods, data packets tend to be pushed back to upstream nodes for alternative routes. In this paper, we focus on the geographic routing based on face routing.

The Greedy-Face-Greedy (GFG) [1] or Greedy Perimeter Stateless Routing (GPSR) [2] adopts face routing based on the right-hand rule as the detour method for local minimum. When encounters a local minimum of greedy forwarding, it finds a detour path by using perimeter routing strategy where packets will be forwarded along the perimeter of the hole (as shown in Fig. 1). Until the packets reach a node that can find a closer neighbor to the destination, it returns to greedy forwarding mode. This solution takes the advantages of both greedy forwarding (intend to find the shortest path) and face routing (guarantee the packet delivery), thus has been

The work of F. Li is partially supported by the National Natural Science Foundation of China under Grant 60903151, Beijing Natural Science Foundation under Grant 4122070 and SRF for ROCS, SEM. The work of Y. Wang is supported in part by the US National Science Foundation (NSF) under Grant CNS-0915331 and CNS-1050398. F. Li (fli@bit.edu.cn) is the corresponding author.

978-1-4673-1881-5/12/$31.00

widely used in many geographic routing systems. However, such solution may also lead to a serious problem: uneven energy consumption around the holes, since it consumes more energy of the boundary sensors. Notice that sensor nodes in WSNs are usually powered by batteries, which have limited energy. The sensor network will be disconnected when the articulation nodes drain their energy. Therefore, to extend the lifespan of the network (the time until the first node is out of energy), the energy consumption of nodes should be balanced. However, geographic routing protocols with face routing as their backup method will suffer great energy depletion on the boundary nodes of holes since they route all traffic around the holes as shown in Fig. 1. When the energy of boundary nodes run out, the hole may expand and enter a vicious cycle. It becomes more serious when holes are shared by several communication sessions in a large scale sensor network.

In this paper, we propose a distributed strategy for existing geographic routing protocols to balance the traffic load on the boundary of holes. What attracts us is that we do not change the underlying forwarding strategy of existing geographic routing protocols (such as GFG [1] or GPSR [2]), packets are still forwarded along the boundary of the hole in perimeter mode. In our proposed mechanism, the sizes of these holes are virtually controlled, so packets are no longer always forwarded along the perimeter of the original holes. We dynamically control these virtual holes to expand and shrink circularly to set up multiple detour paths to bypass them. We propose two different ways to trigger and control the changing of the hole: one is based on a timer; the other is based on the count of packets forwarded. Our simulation results (Section IV) in *network simulator, ns2* [10] show that the proposed method can significantly balance the energy load around holes and prolong the life-time of the whole network.

II. Related Work

To deal with "local minimum" problem in geographic routing protocols, many detouring methods [1], [2], [5]–[9] have been proposed recently. A nice survey can be founded at [4]. Among these proposed methods, the most popular technique is detouring along the boundaries of holes which causes the local minimum of greedy forwarding. However, such solution may lead to over consumption of energy at the boundary nodes. Therefore, various new detouring strategies have been proposed in recent years to look for alternative detour routes.

In the detouring scheme by Jia *et al.* [7], if a packet gets to a local minimum node at the boundary of a hole, it marks that node as a "hole" node and also tells its neighbors. Then packets behind will not be sent to this node anymore. As the process goes on, at last all packets can avoid meeting a hole instead of bypassing it. However, such scheme does not solve the unbalancing problem, since it basically enlarges the boundary of the hole to eliminate the local minimum. But the nodes on the new boundary are still used by many routes. Yu *et al.* [11] and Tian *et al.* [12] used another idea to detour the packets around holes. First, they relied on existing hole boundary detection to detect existence of holes, then used either a virtual circle [11] or a virtual ellipse [12] to cover the hole completely. When packets sent by greedy forwarding reach a node on the boundary of virtual circle or ellipse, they will be forwarded along the tangent direction of its boundary for certain distance. You *et al.* [13] extend the work of [7] by setting up multiple detour paths around each hole instead of just a single detour path. These multiple detour paths are used alternatively in their proposed routing method (HOle-BYpassing routing with Context-AwareNess, HobyCan) to achieve load balancing. However, these paths need to be built after WSN deployment and kept maintained during its operation. Their path construction and maintenance methods are complicated.

Aissani *et al.* [14], [15] proposed another on-demand routing scheme to detour the routing holes. When a message encounters the local minimum at a node, the node initiates the hole detection procedure to detect the boundary of the hole and then announces the information to all nodes located n-hops away from the boundary. These nodes then launch a preventive rerouting process to select the appropriate forwarding region around the hole, to forward each data packet before reaching the boundary nodes. Yang and Fei [16] recently proposed a similar method which first applies a heuristic algorithm to detect possible routing holes and then represents them as simple segments. Such hole information is announced to all nodes within the vicinity who may be affected by the holes, so that those nodes can choose the endpoints of segments as relaying destination to bypass the holes. However, in all these methods, the hole detection, announcement, and rerouting process are complicated and may lead to large control overhead.

Above detouring methods either heavily rely on complex hole detection algorithms and static detouring paths or need to propagate large amount information to all nodes affected. In this paper, instead we focus on how to relieve the traffic pressure of the perimeter of the hole in geographic routing by adding a simple distributed mechanism without any complex detection algorithm or rerouting algorithm. Similar to HobyCan protocol, our method uses multiple detour paths to balance the energy around holes.

III. Distributed Load Balancing Mechanism

To overcome the problem of uneven load distribution at the boundary nodes along routing holes, we propose a new Distributed Load Balancing Mechanism (DLBM) to detour the routes by actively changing the boundary of each hole (forming a dynamic virtual hole). Instead of explicitly setting up multiple fixed detour paths for each hole as [13] did, our solution does not rely on any explicit detouring algorithm but simply changes the boundary of each virtual hole and still uses the underlying geographic routing algorithm for routing packets along the virtual hole. Therefore, the proposed method does not change the underlying forwarding strategy of existing geographic routing protocols (such as GFG [1] or GPSR [2]) which makes it possible to be applied widely. In this paper, we use GPSR as the example geographic routing protocol. Packet header of GPSR packet includes a flag field indicating whether the packet is in greedy mode or perimeter mode. In GPSR, when a data packet in the greedy mode reaches a local minimum, it changes to the perimeter mode and uses the right hand rule to detour around the hole.

A. DLBM with a Timer

Our mechanism is triggered when GPSR encounters the "local minimum" at a node u and the routing mode turns to the perimeter mode. Node u will forward the packet based on right hand rule along the boundary of the hole. After forwarding the packet, u then broadcasts a *Not_Send_To_Me* message to its neighbors. Once receiving the message, all its neighbors will set the *On_Off* flag of u to *Off* in their neighbor lists. This will disable node u as possible relay node in the following rounds. In other words, after the first packet bypass the hole in original perimeter mode, part of the boundary of the hole is expanded, a virtual hole with larger size is formed as shown in Fig. 2. Due to right-hand rule in GPSR's perimeter mode, the next packet will detour the hole on the new boundary of this virtual hole. In this way, every packet will choose a different path to bypass the hole in perimeter mode. Load on the boundary of the original hole is apportioned by the nodes on the outer layers.

When the virtual hole expands, the path formed by perimeter forwarding becomes longer than before. This leads to more energy consumption and longer delay from the global point of view. Thus, we do not want to expand the hole too far from the original boundary. In this version of DLBM, a timer is set to stop the expanding process. When the predetermined timer expires, all the *On_Off* flags will be reset to *On*, thus the hole immediately returns to its original shape. The hole may get into the expanding stage again. This process will repeat to utilize multiple detouring routes in turns. This eventually balances the load among nodes in the perimeter mode.

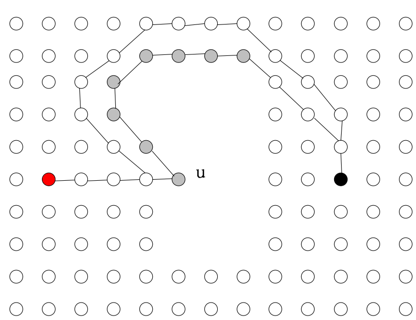


Fig. 2: **Expanding the virtual hole**: When the first packet bypasses the hole along its boundary starting from at u in perimeter mode, the virtual hole expands for one layer. The second packet then will bypass the hole along the new boundary of this virtual hole.

The initial value T of the predefined timer controls the number of layers during the expansion (*i.e.* the number of multiple detouring paths). Let N be the maximal layer that DLBM wants to expand and R be the data rate of the routing source. Then the duration of the expanding process T can be estimated as follows:

$$T = R \times N \tag{1}$$

In other words, if the packet generation rate is known, we can easily decide the initial value of timer. Note that N should not be too large, and it turns out that usually 2 to 4 is already good enough as shown by our simulation in Section IV.

B. DLBM with a Packet Counter

We can also control the expansion of virtual holes by using the number of packets that have been forwarded at each node. In this version of DLBM, every node has a counter to record how many data packets have been forwarded in the current session. Nodes need to maintain some additional information, including which layer they are on and whether the virtual hole is in *expanding mode* or *shrinking mode*.

The DLBM starts with *expanding mode*. In expanding mode, when a node u forwards a packet in perimeter forwarding mode, its packet counter increases. If the counter reaches a predefined threshold (C), u will broadcast a *Not_Send_To_Me* message (including the notification of expanding mode and the next layer number which is one plus current layer number of u) to its neighbors. When a node v receives this message, it turns off the *On_Off* flag of u and updates its layer number and mode. This process repeats until a node detects that it is in the outermost layer (its layer number is equal to a predefined number N) and its packet counter reaches the threshold C. DLBM then turns to *shrinking mode*.

In shrinking mode, packet counter decreases when a packet is forwarded at the current node. When the counter reaches to zero, the node will broadcast a *Wake_Up* message (including the notification of shrinking mode and the next layer number which is its current layer number minus one) to its neighbors. When a node u receives the *Wake_Up* message and its layer number is equal to the next layer number in the message, it enters to shrinking mode and broadcast a message *Send_To_Me* message to its neighbors so that they can turn on u's *On_Off* flag in their lists. This process repeats until a node detects that it is in the innermost layer and its packet counter reaches zero, the network recovers the original topology and DLBM turns back to expanding mode.

The expanding mode and shrinking mode take turns to balance the load which bypasses holes. Algorithm 1 shows the detail operations in this version of DLBM at a node.

IV. Simulation Results

To evaluate the performance of our proposed DLBM, we implement both two DLBM versions in *ns2* [10] over GPSR [2]. We conduct extensive simulations over a large scale sensor network to compare their performances with the standard GPSR. In addition, we also implement HobyCan protocol [13], which is an existing solution with multiple detour paths for comparison. Hereafter, we use GPSR-DLBM1 and GPSR-DLBM2 to denote the DLBM with a timer over GPSR and the DLBM with a packet counter over GPSR, respectively. The underlying sensor network is formed by 400 sensor nodes, which are evenly deployed in a $400 \times 400 m^2$ region excluding a huge rectangular hole. The communication radius of each sensor is $25m$. Each sensor has its initial energy at 300 units,

Algorithm 1 DLBM scheme at node u

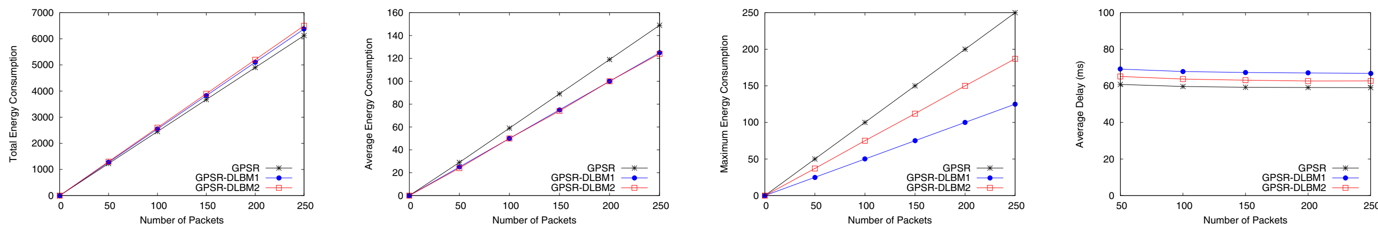
Initialization:
$mode = expanding\_mode$, $layer\_number = NULL$
when receive a packet in *expanding_mode*:
forward the packet
$packet\_count = packet\_count + 1$
if $packet\_count = C$ **then**
  if $layer\_number < N$ **then**
    send $Not\_Send\_To\_Me$ ($layer\_number + 1$)
  else
    $mode$ = $shrinking\_mode$
  end if
end if
when receive a packet in *shrinking_mode*:
forward the packet
$packet\_count = packet\_count - 1$
if $packet\_count = 0$ **then**
  if $layer\_number > 0$ **then**
    send $Wake\_Up$ ($layer\_number - 1$)
  else
    $mode = expanding\_mode$
  end if
end if
when receive a msg $Not\_Send\_To\_Me(k)$ **from** v:
if $layer\_number = NULL$ **then**
  $layer\_number = k$
  $mode = expanding\_mode$
  set On_Off flag of v to Off
end if
when receive a message $Wake\_Up(k)$ **from** v:
if $layer\_number = k$ **then**
  $mode = shrinking\_mode$
  send $Send\_To\_Me()$
end if
when receive a message $Send\_To\_Me$ **from** v:
set On_Off flag of v to On

and forwarding a data packet costs one unit of energy at each node. We employ the following metrics to evaluate the performance of all schemes: (1) *total energy consumption*, the summation of energy cost at all involved sensor nodes; (2) *average energy consumption*, the average of energy cost at each involved sensor node; (3) *maximum energy consumption*, the maximum energy cost at each involved sensor node; (4) *average delay*, the average delay of all delivered packets.

With or without DLBM: In the first set of simulations, we place two source nodes on the left side of the rectangle hole and two sinks on the other side of the hole. Each source node sends multiple packets (up to 125) with one of the sink as its destination. The maximum layer to expand (N) is set to 2. We compare the performances of GPSR-DLBM1 and GPSR-DLBM2 with GPSR and results are plotted in Fig. 3. Fig. 3(a) shows the total energy consumption of all methods. GPSR-DLBM1 and GPSR-DLBM2 consume more energy than the original GPSR, since packets bypass hole via longer paths. However, the increasing is not significant and GPSR-DLBM1 has similar performance with GPSR-DLBM2. More importantly, Fig. 3(b) shows that the average energy consumption of both DLBM methods is much lower than the one of GPSR. This confirms that our proposed DLBM can significantly balance the traffic load on the boundary of holes. Similarly, as illustrated in Fig. 3(c), the maximum energy of GPSR-DLBM1 and GPSR-DLBM2 are also much lower than the one of GPSR. Therefore, the lifetime of the network (the time until the first sensor node runs out of energy) is prolonged by using DLBM over GPSR. DLBM1 has a smaller maximum energy than DLBM2 does, since it has a shorter cycle which leads to more evenly distributed load. Fig. 3(d) demonstrates that GPSR-DLBM1 and GPSR-DLBM2 have a longer delay than GPSR does. This is mainly due to bypassing the hole with longer detour paths.

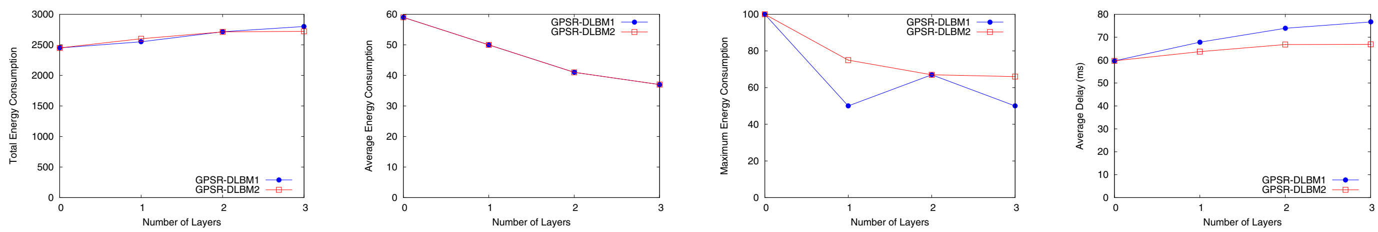
Number of Layers Expanded in DLBM: In the second set of simulations, we keep the same setting but test various value of N (the maximum number of layers expanded in DLBM). Fig. 4 provides the results with $N = 0$, 1, 2 and 3. Notice that when $N = 0$, DLBM regresses to GPSR. Here, each source sends 50 packets. Clearly, with more layers expanded, the load is more balanced among nodes (both average energy and maximum energy decrease with increased N). But larger N also leads to more total energy consumption and longer average delay due to longer detours. Notice that in Figure 4(c) the maximum energy of GPSR-DLBM1 has an unusually jump when $N = 2$. This is mainly due to our underlying layout of the network. With multiple layers, some routes may pass certain nodes twice in both greedy and perimeter forwarding modes. This may cause the increasing of energy consumption at those nodes. Notice that this only happens to GPSR-DLBM1 since GPSR-DLBM2 uses packet counter to trigger the layer switch which prevents such situation.

Comparison with HobyCan: In the last set of simulations, we let both source nodes on the left side of the hole send packets to just one sink on the other side. N is set to 2. We compare our DLBM methods with HobyCan [13]. In HobyCan, detouring over additional path is triggered when the remaining energy of current node on the routing path reaches a predefined threshold (E_{THR}). Here, we set E_{THR} to 80%. Fig. 5 shows the results. First, the total energy consumption of our DLBM schemes is similar with the one of HobyCan as shown in Fig. 5(a). Second, in Fig. 5(b) HobyCan has a higher average energy consumption than our DLBM methods at the beginning when the number of packets forwarded is small. As traffic increases, HobyCan has very similar average energy consumption with our DLBM methods. This is due to that multiple detour paths are used after E_{THR} is reached in HobyCan. Third, Fig. 5(c) demonstrates that GPSR-DLBM1 always has smaller maximum energy consumption than GPSR-DLBM2. Again a shorter cycle leads to a better performance in maximum energy consumption. HobyCan has smaller maximum energy consumption than GPSR-DLBM2



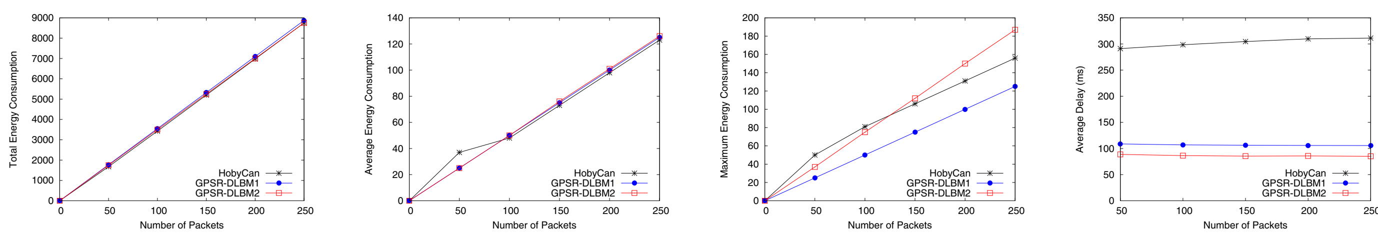
(a) total energy (b) average energy (c) maximum energy (d) average delay

Fig. 3: Performances of GPSR, GPSR-DLBM1 and GPSR-DLBM2.



(a) total energy (b) average energy (c) maximum energy (d) average delay

Fig. 4: Performances of GPSR-DLBM1 and GPSR-DLBM2 with different maximum number of layers during the expanding mode.



(a) total energy (b) average energy (c) maximum energy (d) average delay

Fig. 5: Performances of GPSR-DLBM1, GPSR-DLBM2, and HobyCan.

as the traffic increases, but it is always worse than GPSR-DLBM1. Last, Fig. 5(d) shows that DLBM has smaller delay than HobyCan. Therefore, overall, DLBM enjoys much better load balancing and smaller delay over HobyCan.

V. Conclusion

In this paper, we propose a novel distributed strategy to balance the traffic load on the boundary of holes for existing geographic routing protocols by virtually changing the sizes of holes. Compared with existing detouring methods, our DLBM scheme has some prominent advantages: (1) it does not change the underlying forwarding strategy of existing geographic routing protocols and is easy to be implemented; (2) it can balance the routing load among nodes near routing holes by dynamically controlling these holes to expand and shrink circularly. Our simulation results in *ns2* show that the proposed method can significantly balance the energy load around holes and prolong the life-time of the whole network.

References

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *ACM/Kluwer Wireless Networks*, 7(6):609–616, 2001.

[2] B. Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of ACM MobiCom*, 2000.

[3] M. Mauve, J. Widmer, and H. Harenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Net. Mag.*, 15(6):30–39, 2001.

[4] D. Chen and P. K. Varshney. A survey of void handling techniques for geographic routing in wireless networks. *IEEE Communications Surveys and Tutorials*, 9(1-4):50–67, 2007.

[5] C. Lu, B.M. Blum, T.F. Abdelzaher, J.A. Stankovic, and T. He. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *Proc. of IEEE RTAS*, 2002.

[6] O. Chipara, Z. He, G. Xing, et al. Real-time power-aware routing in sensor networks. In *Proc. of IEEE IWQoS*, 2006.

[7] W. Jia, T. Wang, G. Wang, and M. Guo. Hole avoiding in advance routing in wireless sensor networks. In *Proc. of IEEE WCNC*, 2007.

[8] T. He, J.A. Stankovic, C. Lu, and T.F. Abdelzaher. A spatiotemporal communication protocol for wireless sensor networks. *IEEE Trans. on Parallel and Distributed Systems*, 16(10):995–1006, 2005.

[9] L. Zhao, B. Kan, et al. FT-SPEED: A fault-tolerant, real-time routing protocol for wireless sensor networks. In *Proc. of WiCom*, 2007.

[10] The Network Simulator. (ns-2). http://www.isi.edu/nsam/ns.

[11] S. Park F. Yu, Y. Tian, et al. Efficient hole detour scheme for geographic routing in wireless sensor networks. In *Proc. of IEEE VTC*, 2008.

[12] Y. Tian, F. Yu, Y. Choi, S. Park, E. Lee, M. Jin, and S.-H. Kim. Energy-efficient data dissemination protocol for detouring routing holes in wireless sensor networks. In *Proc. of IEEE ICC*, 2008.

[13] J. You, D. Lieckfeldt, et al. Context-aware geographic routing for sensor networks with routingholes. In *Proc. of IEEE WCNC*, 2009.

[14] M. Aissani, et al. An efficient real-time routing with presence of concave voids in wireless sensor networks. In *Proc. of ICWCMC*, 2009.

[15] M. Aissani, et al. A preventive rerouting scheme for avoiding voids in wireless sensor networks. In *Proc. of IEEE Globecom*, 2009.

[16] J. Yang and Z. Fei. HDAR: Hole detection and adaptive geographic routing for ad hoc networks. In *Proc. of IEEE ICCCN*, 2010.