International Journal of Parallel, Emergent and Distributed Systems, 2013
Vol. 28, No. 2, 184–197, http://dx.doi.org/10.1080/17445760.2012.660151

Taylor & Francis
Taylor & Francis Group

Distributed load balancing mechanism for detouring schemes of geographic routing in wireless sensor networks

Fan Li[a], Jinnan Gao[a] and Yu Wang[b]*

[a]*School of Computer Science, Beijing Institute of Technology, Beijing 100081, P.R. China;* [b]*Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA*

(*Received 26 November 2011; final version received 18 January 2012*)

Well-known 'routing hole' problem of geographic routing is hardly avoided in wireless sensor networks because of various actual geographical environments. Existing geographic routing protocols 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 strategy can effectively balance the load around holes, thus prolonging the network life of sensor networks when an existing geographic routing protocol is used as the underlying routing protocol.

Keywords: geographic routing; load balancing; routing holes; wireless sensor networks

1. Introduction

Due to its wide applications, such as battlefield, emergency relief and environment monitoring, wireless sensor network (WSN) has recently emerged as a premier research topic. WSN usually consists of a large set of sensor nodes spreading over a geographical area. Routing in such large-scale WSNs is always a challenging task. One possible solution is geographic routing [4,14]. 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 neighbours, the message can be routed towards the destination without the knowledge of the whole network topology or a prior route discovery. This significantly improves the scalability of such routing protocols especially in large-scale sensor networks.

Greedy routing is the most popular and widely used geographic routing protocol. In greedy routing, packets are greedily delivered to the neighbour which is the nearest one

*Corresponding author. Email: yu.wang@uncc.edu

among the current node and all its neighbours to the destination. Greedy routing has been demonstrated very effective in large-scale WSNs and adaptive to topology changes. However, greedy routing fails to deliver the packet when it meets a node which cannot find a neighbour 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 WSN (such as at node *u* in Figure 1), thus it is also known as ‘routing 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.

Most geographic routing protocols have their own special methods to find a detour path when they encounter the *local minimum*. Those can be grouped into two categories: face routing methods based on right-hand rule [3,5,9,13] and methods based on back-pressure rule [7,8,22]. 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 methods based on back-pressure rule, data packets tend to be pushed back to upstream nodes for alternative routes. In this paper, we will focus on the geographic routing based on face routing.

The greedy-face-greedy (GFG) [3] or greedy perimeter stateless routing (GPSR) [9] adopts face routing based on the right-hand rule as the detour method for dealing with the local minimum. When encountering 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 Figure 1). Until the packets reach a node that can find a closer neighbour to the destination, it returns to greedy forwarding mode. This solution takes the advantages of both greedy forwarding (which intends to find the shortest path) and face routing (which guarantees the packet delivery), thus has been 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

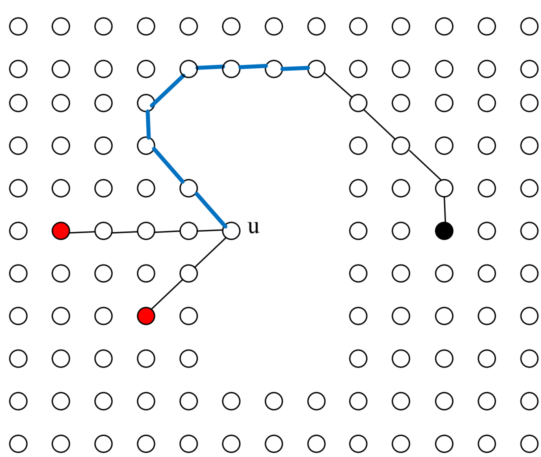

Figure 1. An example of detouring around a routing hole in geographic routing. Here, two red nodes are source nodes and the black node is the destination. Local minimum of greedy forwarding occurs at node *u*. Then face routing routes the packets along the boundary of the topology hole, denoted by blue curve. Clearly, this leads to higher traffic load on the boundary, since both routes use the same detour path.

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 Figure 1. When the energy of boundary nodes run out, the hole may expand and enter a vicious cycle. It becomes more serious when holes 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 attraction is that we do not change the underlying forwarding strategy of existing geographic routing protocols (such as GFG [3] or GPSR [9]), and 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 so that multiple detour paths are set up to bypass these holes. We propose two different ways to trigger and control the changing of the hole: one is based on a timer and the other is based on the count of packets forwarded. Our simulation results in *network simulator*, *ns2* [17], show that the proposed method can significantly balance the energy load around holes and prolong the lifetime of the whole network.

The rest of this paper is organized as follows. Section 2 discusses related work on detouring and load balancing approaches of geographic routing in WSNs. Section 3 provides our proposed method to balance the traffic load on the boundary of holes by smartly controlling virtual holes. Section 4 presents our simulation results in *ns2*, in which the performances of our proposed method with the original GPSR and another detouring algorithm HOle-BYpassing routing with Context-AwareNess (HobyCan) [21] are compared. Section 5 concludes this paper with a brief summary.

2. Related work

To deal with the *local minimum* problem in geographic routing protocols, many detouring methods [3,5,7–9,13,22] have been proposed recently. A nice survey on various detouring methods can be found at Ref. [4]. Among these proposed methods, the most popular technique is detouring along the boundaries of holes to get out of local minimum of greedy forwarding. However, such solution may lead to overconsume energy at the boundary nodes. Therefore, various new detouring strategies have been proposed in recent years to look for alternative detour routes.

Jia et al. [8] proposed a detouring scheme, called hole avoiding in advance routing (HAIR) protocol. 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 neighbours. 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 a hole. However, such scheme does not solve the unbalancing problem, since it basically enlarge the boundary of the hole to eliminate the local minimum. And the nodes on the new boundary are still used by many routes.

Yu et al. [15] and Tian et al. [18] used a similar idea to detour the packets around holes. First, they relied on existing hole boundary detection to detect the existence of holes and then used either a virtual circle [15] or a virtual ellipse [18] 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. Then the forwarding mode returns to the greedy forwarding.

You et al. [21] extended the work of Jia et al. [8] 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 (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. [1,2] proposed another on-demand routing scheme to detour the routing holes. In their method, 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. Again the hole detection, announcement and rerouting process are complicated and may lead to large control overhead of the routing protocols.

Yang and Fei [19] recently proposed a similar method called hole detection and adaptive geographical routing (HDAR). HDAR 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, the propagation of hole announcements may again lead to large amount of overhead messages.

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

Notice that load balancing in routing is not a new topic. It has been well studied in routing protocols for both *ad hoc* and sensor networks. Most existing methods [6,10,20] try to dynamically adjust the routes with the knowledge of current remaining energy or load distribution to achieve load balancing. Multi-path routing [16] has also been widely used for load balancing. Li et al. [11] and Li and Wang [12] recently also abandon shortest path routing or greedy routing to achieve better load balancing by using longer paths. However, all these load-balancing routing methods are not for avoiding holes in geographic routing.

3. 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 [21] 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 [3] or GPSR [9]) which makes it can 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 greedy mode reaches a local minimum, the routing algorithm changes to the perimeter mode and uses the right-hand rule to detour around the hole.

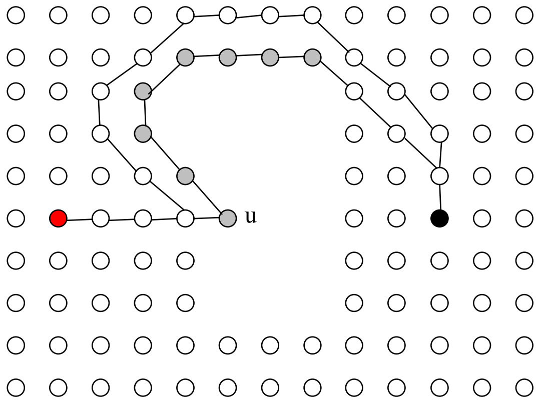

Figure 2. Expanding the virtual hole: when the first packet bypasses the hole along its boundary starting from at node *u* in perimeter mode, the virtual hole expands for one layer. The second packet then will bypass the hole along the new boundary of this virtual hole.

3.1 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 neighbours. Once receiving the message, all its neighbours will set the *On_Off* flag of *u* to *Off* in their neighbour lists. This will disable node *u* as possible relay node in the following rounds. In other words, after the first packet bypasses the hole in original perimeter mode, a part of the boundary of the hole is expanded and a virtual hole with larger size is formed as shown in Figure 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 of GPSR. 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 with 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. Then the hole may get into the expanding stage again. This process will repeat to utilize multiple detouring routes in turn. This eventually balances the load among nodes in the perimeter model.

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 the timer. Note that N should not be too large, and usually 2–4 is already good enough as shown by our simulations in Section 4.

3.2 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*.

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 = shirking_mode
end if
end if
when receive a packet in shirking_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 message Not_Send_To_Me(k) from v:
if layer_number = NULL then
  layer_number = k
  mode = expanding_mode
  set On_Off flag of v = Off
end if
when receive a message Wake_Up(k) from v:
if layer_number = k then
  mode = shirking_mode
  send Send_To_Me()
end if
when receive a message Send_To_Me from v:
set On_Off flag of v = On
```

The DLBM starts with *expanding mode*. In expanding mode, when a node *u* receives a packet forwarded in perimeter forwarding mode, its packet counter increases after *u* forwarded the packet. 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 the current layer number of *u*) to its neighbours. 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 was forwarded at the current node. When the counter reaches zero, the node will broadcast a *Wake_Up* message (including the notification of shrinking mode and the next layer number which is its layer number minus 1) to its neighbours. 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 broadcasts a message *Send_To_Me* message to its neighbours 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 are taking turns to balance the load which bypasses holes. Algorithm 2 shows the detail operations in this version of DLBM at a node *u*.

Note that since each node has a predetermined threshold *C* for the packet counter, we can smartly control these thresholds to adjust the speed of expansion or shrink. One possible way is to give larger threshold to the nodes with higher remaining energy, so that they can serve the detour longer than those nodes with lower energy. We leave such study as one of our future work.

4. Simulation results

To evaluate the performance of our proposed DLBM, we implement both two DLBM versions (with a timer and with a packet counter) in *network simulator*, *ns2* [17], over GPSR [9]. We conduct extensive simulations over large-scale sensor networks to compare their performances with the standard GPSR. In addition, we also implement HobyCan protocol [21], 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.

We first use a 400-node sensor network, in which sensors are evenly deployed in a $400 \times 400\,\text{m}^2$ region excluding a huge rectangular hole (similar to the one in Figure 1), as the default network. But later we also test a larger sensor network with multiple holes and multiple random sensor networks. The communication radius of each sensor is 25 m. Each sensor has its initial energy at 300 units, and forwarding a *data packet* costs one unit of energy at each node. In this study, we do not consider the energy consumption for sending *control messages*, since the size of a control message is usually ignorable compared with the size of a data packet.

We employ the following metrics to evaluate the performance of all schemes:

(1) *Total energy consumption*: the summation of energy cost at all involved sensors.
(2) *Average energy consumption*: the average of energy cost at each involved sensor.
(3) *Maximum energy consumption*: the maximum energy cost at each involved sensor.
(4) *Average delay*: the average delay of all delivered packets.

Notice that since we assume reliable data transmission at link layer, all these geographic routing methods have perfect delivery ratio. For energy consumption at each node, we only consider those of nodes who participate forwarding of data packets. Both average and maximum energy consumption metrics can reflect the level of load balancing among sensors. The smaller the metric, the better the load balancing is. If the lifetime of a sensor network is defined as the time when the first sensor runs out of energy and the initial energy of each sensor is the same, then smaller maximum energy consumption will directly lead to longer lifetime.

4.1 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 the results are plotted in Figure 3.

Figure 3(a) shows the total energy consumption of all methods. GPSR-DLBM1 and GPSR-DLBM2 consume more energy than the original GPSR does, since packets bypass hole via longer paths. However, the increase is not significant and GPSR-DLBM1 has similar performances with GPSR-DLBM2. More importantly, Figure 3(b) shows that the average energy consumption of both DLBM methods is much lower than that of GPSR. This confirms that our proposed DLBM can significantly balance the traffic load on the boundary of holes. Notice that both the total energy consumption and the average energy consumption here are defined over all involved sensor nodes during the routing phase not over all sensor nodes in the network. Thus, even though our proposed DLBM methods have higher total energy consumption than GPSR due to more nodes involved, they do have lower average energy consumption than GPSR does. Similarly, as illustrated in Figure 3(c), the maximum energy of GPSR-DLBM1 and GPSR-DLBM2 is also much lower than that of GPSR. Therefore, the lifetime of the network (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.

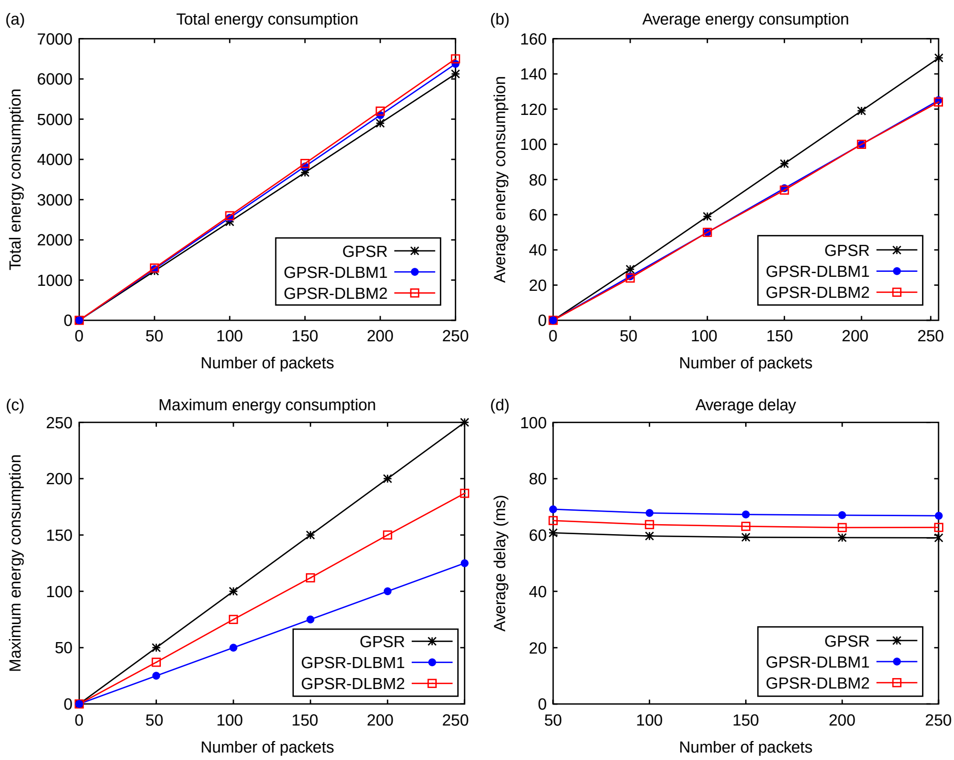

Figure 3. Performances of GPSR, GPSR-DLBM1 and GPSR-DLBM2.

Figure 3(d) shows that GPSR-DLBM1 and GPSR-DLBM2 have a higher delay than GPSR does. This is mainly due to bypassing the hole with longer detour paths.

4.2 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). Figure 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 (average energy and maximum energy both decrease with increased N). But larger N

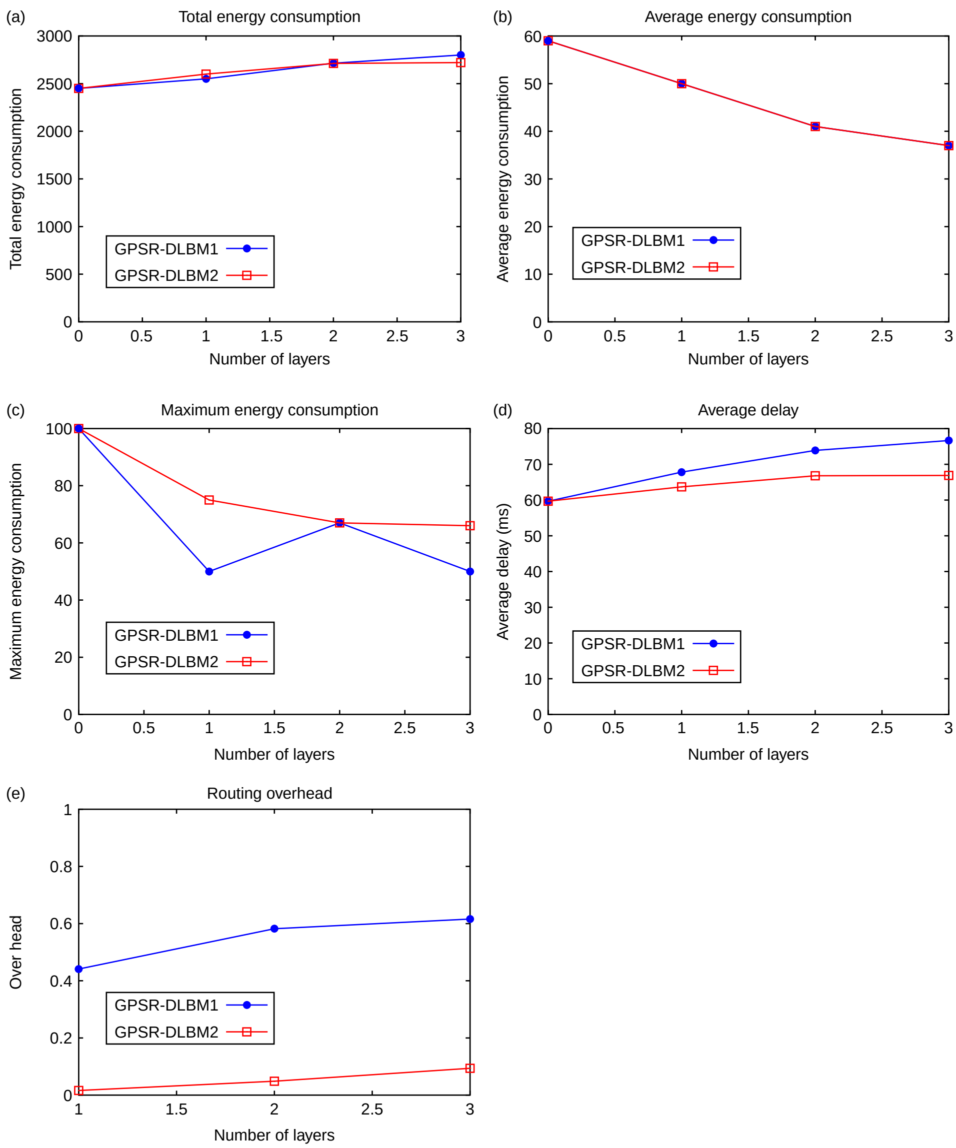

Figure 4. Performances of GPSR-DLBM1 and GPSR-DLBM2 with different maximum number of layers during the expanding mode.

also lead 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 an increase in energy consumption at those nodes. Notice that this only happens for GPSR-DLBM1 since GPSR-DLBM2 uses packet counter to trigger the layer switch which prevents such situation. We also measure the routing overhead (the number of control messages per data packet) of DLBM and plot it in Figure 4(e). It is clear that the overhead increases with the maximum layer number. GPSR-DLBM2 has less control messages than GPSR-DLBM1 does, because GPSR-DLBM1 switches layer more frequently.

4.3 Comparison with HobyCan

In the third 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 [21]. 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%. Figure 5 shows the results.

First, the total energy consumption of our DLBM schemes is similar with the one of HobyCan as shown in Figure 5(a). Second, Figure 5(b) HobyCan has a higher average energy consumption than our DLBM methods at beginning when the number of packets

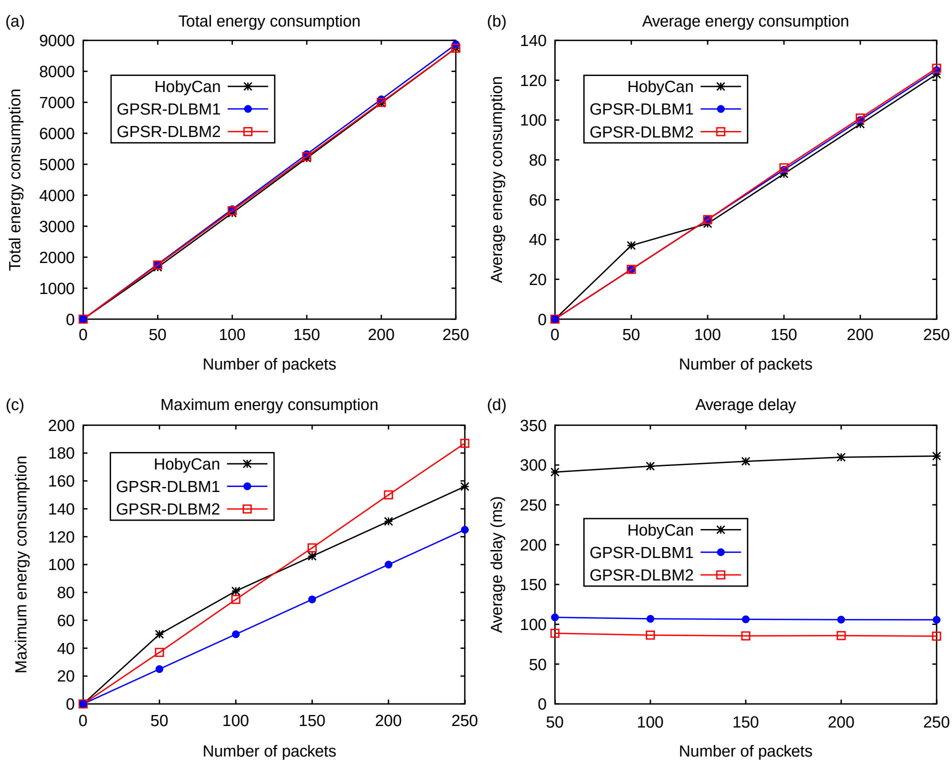

Figure 5. Performances of GPSR-DLBM1, GPSR-DLBM2 and HobyCan.

forwarded is small. As traffic increases, HobyCan has very similar average energy consumption with our DLBM methods. This is due to the reason that multiple detour paths are used after E_{THR} is reached in HobyCan. Third, Figure 5(c) shows 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 as the traffic increases, but it is always worse than GPSR-DLBM1. Last, Figure 5(d) shows that our DLBM algorithm has smaller delay than HobyCan. Therefore, overall, DLBM enjoys much better load balancing and smaller delay over HobyCan.

4.4 With multiple holes

In the fourth set of simulations, we create two holes in a larger sensor network (in which 800 sensors are evenly deployed in an $800 \times 400\,\mathrm{m}^2$ region) and all traffic needs to bypass both holes. The results are plotted in Figure 6. The same observations can be obtained, in which our proposed DLBM schemes can balance the energy consumption at each node and reduce the average energy consumption by slightly increasing the delay and the total energy consumption. Interestingly, the maximum energy consumption in this experiment is the same among all methods. This is due to the reason that the maximum energy consumption always happen at a node on the greedy forwarding path instead of the perimeter path around the holes.

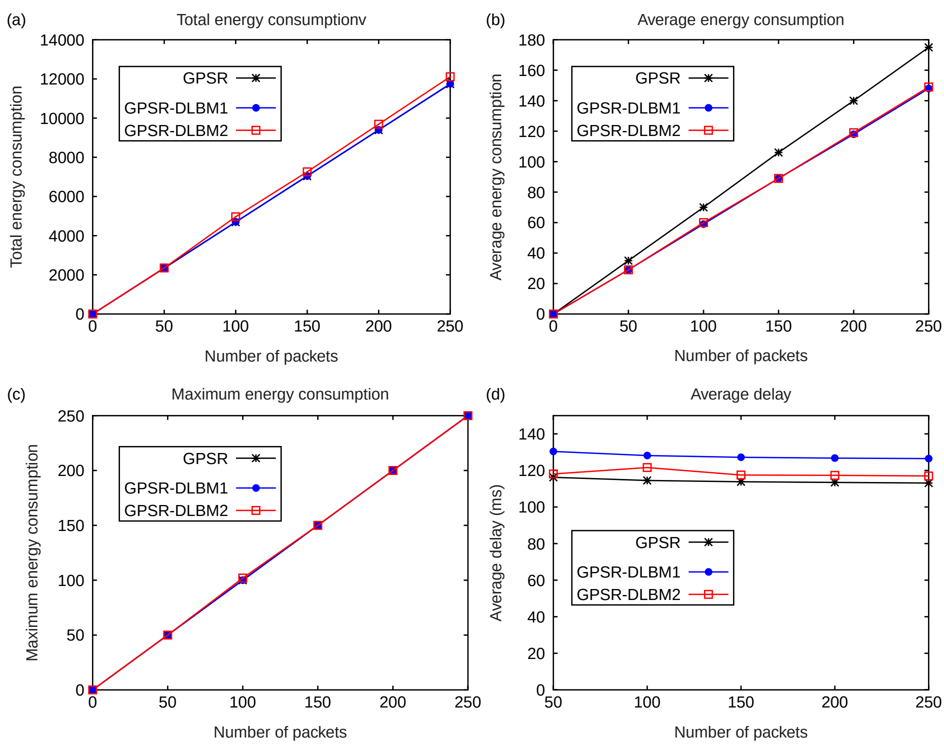

Figure 6. Performances of GPSR, GPSR-DLBM1 and GPSR-DLBM2 over a network with two holes.

4.5 Random networks

In the last set of simulations, we use randomly deployed sensor networks instead of the regularly distributed sensor network. In each network, 400 sensor nodes are randomly deployed in a $400 \times 400\,\mathrm{m}^2$ region. Two sources and sinks are placed on both sides of a single rectangle hole. The average is taken over 50 random networks. The results are plotted in Figure 7. The results are basically consistent with those from regularly deployed networks.

5. 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 the existing detouring protocols, our DLBM scheme has some prominent advantages:

(1) It does not change the underlying forwarding strategy of existing geographic routing protocols.
(2) It can balance the routing load among nodes near routing holes by dynamically controlling these holes to expand and shrink circularly.
(3) The proposed method is simple to implement and several parameters (such as packet counter C, layer number N and the timer T) can be controlled to adjust the performance of the proposed method to fulfil different environments or requirements.

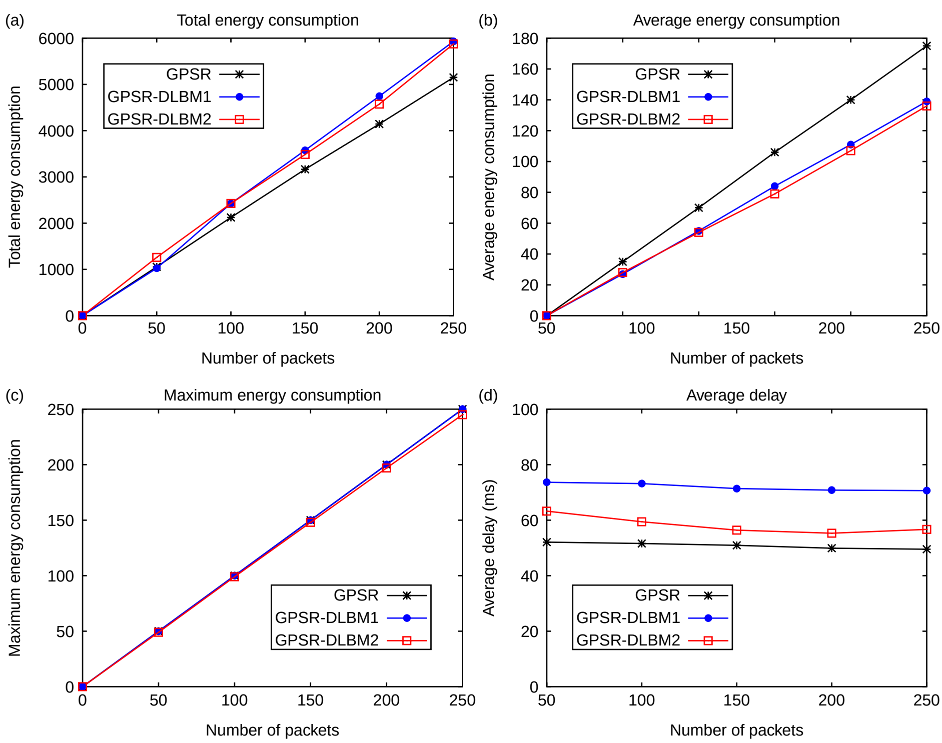

Figure 7. Performances of GPSR, GPSR-DLBM1 and GPSR-DLBM2 over random networks.

Our simulation results in *ns2* show that the proposed method can significantly balance the energy load around holes and prolong the lifetime of the whole network.

Acknowledgements

The work of F. Li is partially supported by the National Natural Science Foundation of China (NSFC) under Grant 60903151 and Beijing Natural Science Foundation under Grant No. 4122070. The work of Y. Wang is supported, in part, by the US National Science Foundation (NSF) under Grant No. CNS-0915331 and CNS-1050398.

References

[1] M. Aissani, A. Mellouk, N. Badache, and M. Djebbar, *A preventive rerouting scheme for avoiding voids in wireless sensor networks*, Proceedings of 2009 IEEE conference on Global telecommunications (GLOBECOM 2009), 2009.

[2] M. Aissani, A. Mellouk, N. Badache, and B. Saidani, *An efficient real-time routing with presence of concave voids in wireless sensor networks*, Proceedings of International Conference on Wireless Communications and Mobile Computing (IWCMC 2009), 2009.

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

[4] D. Chen and P.K. Varshney, *A survey of void handling techniques for geographic routing in wireless networks*, IEEE Commun. Surv. Tutor. 9(1–4) (2007), pp. 50–67.

[5] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J.A. Stankovic, and T.F. Abdelzaher, *Real-time power-aware routing in sensor networks*, Proceedings of 2006 IEEE International Workshop on Quality of Service (IWQoS 2006), 2006.

[6] H. Hassanein and A. Zhou, *Routing with load balancing in wireless ad hoc networks*, Proceedings of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2001), 2001.

[7] T. He, J.A. Stankovic, C. Lu, and T.F. Abdelzaher, *A spatiotemporal communication protocol for wireless sensor networks*, IEEE Transact. Parallel Distrib. Syst. 16(10) (2005), pp. 995–1006.

[8] W. Jia, T. Wang, G. Wang, and M. Guo, *Hole avoiding in advance routing in wireless sensor networks*, Proceedings of 2007 IEEE Wireless Communications and Networking Conference (WCNC 2007), 2007.

[9] B. Karp and H.T. Kung, *GPSR: Greedy perimeter stateless routing for wireless networks*, Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), 2000.

[10] S.-J. Lee and M. Gerla, *Dynamic load-aware routing in ad hoc networks*, Proceedings of 2001 IEEE International Conference on Communications (ICC2001), 2001.

[11] F. Li, S. Chen, and Y. Wang, *Load balancing routing with bounded stretch*, EURASIP J. Wireless Commun. Netw. 2010 (2010), Article ID 623706, 16 pp.

[12] F. Li and Y. Wang, *Circular sailing routing for wireless networks*, Proceedings of 2008 IEEE International Conference on Computer Communications (INFOCOM 2008), 2008.

[13] 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*, Proceedings of 2002 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002), 2002.

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

[15] S. Park, F. Yu, Y. Tian, M. Jin, and S.-H. Kim, *Efficient hole detour scheme for geographic routing in wireless sensor networks*, Proceedings of 2008 IEEE Vehicular Technology Conference (VTC 2008), 2008.

[16] P.P. Pham and S. Perreau, *Performance analysis of reactive shortest path and multipath routing mechanism with load balance*, Proceedings of 2004 IEEE International Conference on Computer Communications (INFOCOM 2003), 2004.

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

[18] 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*, Proceedings of 2008 IEEE International Conference on Communications (ICC 2008), 2008.
[19] J. Yang and Z. Fei, *HDAR: Hole detection and adaptive geographic routing for ad hoc networks*, Proceedings of 2010 IEEE International Conference on Computer Communication Networks (ICCCN 2010), 2010.
[20] Y. Yoo and S. Ahn, *Ga simple load-balancing approach in cheat-proof ad hoc networks*, Proceedings of 2004 IEEE Global Communications Conference (GLOBECOM 2004), 2004.
[21] J. You, D. Lieckfeldt, F. Reichen, and D. Timmermann, *Context-aware geographic routing for sensor networks with routingholes*, Proceedings of 2009 IEEE Wireless Communications and Networking Conference (WCNC 2009), 2009.
[22] L. Zhao, B. Kan, Y. Xu, and X. Li, *FT-SPEED: A fault-tolerant, real-time routing protocol for wireless sensor networks*, Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007), 2007.