This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

Context-aware Geographic Routing for Sensor Networks with Routing Holes

Jiaxi You, Dominik Lieckfeldt, Frank Reichenbach, and Dirk Timmermann
University of Rostock, Germany
{jiaxi.you, dominik.lieckfeldt, frank.reichenbach, dirk.timmermann}@uni-rostock.de

***Abstract*— Modern sensor networks are deployed in various terrains of interest. As the complexity of their deployed areas is growing, existing geographic routing algorithms are facing challenges. Holes in networks often cause failures in message routing. Energy consumption, scalability, and routing efficiency are also key design challenges. In this paper, we propose a novel geographic routing algorithm called HOle-BYpassing routing with Context-AwareNess (HobyCan). Our approach locally sets up multiple detour paths to bypass almost all kinds of holes. Therefore, contours of holes are extended with multiple detour paths. According to various context information of a sensor network, such as the size of holes or the remaining energy of nodes, disjoint detour paths can be used alternatively to achieve optimal routing paths or load balance of the network. Simulation results demonstrate the performance of our algorithm, as well as the significance of context information as routing parameters.**

***Key words*— context-awareness, geographic routing, hole-bypassing, wireless sensor networks.**

I. Introduction

Sensor nodes are smart devices that are capable of sensing physical parameters, processing data and communicating wirelessly [1]. Wireless Sensor Networks (WSNs) are composed of many sensor nodes deployed in an area of interest. Among various routing protocols [2], geographic routing (georouting) with *greedy forwarding* is attractive [4][5] for WSNs. In a basic geographic routing algorithm [6], a node communicates with its direct neighbors only. The neighbor nearest to the sink will be selected as the next hop. Such an approach is effective and can be dynamically adapted to changes, which only requires location information of sensor nodes.

As WSNs are becoming widely used in various applications, their deployments are becoming rather complicated. In many applications, sensor nodes are simply scattered in an area and left unattended. Thus, voids in deployment or node failure can cause *routing holes* [3] in the network, which often cause traditional geographic routing algorithms to fail. The reason is the *local minimum* phenomenon illustrated in Fig. 1. When using *greedy forwarding*, packets get stuck at node *A* since there is no neighbor node closer to the destination than node *A* itself. Another critical design issue is the lifetime of WSNs. The sensing and communication activity is typically non-uniformly distributed over sensor nodes, which implies non-uniform power consumption in the network. Depletion of energy on some nodes can reduce the coverage and connectivity of a WSN. To prolong the lifetime of WSNs, routing protocols should also consider balancing of energy consumption between sensor nodes.

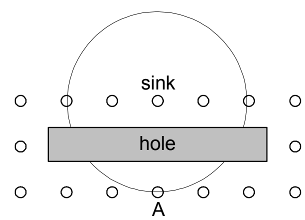


Fig. 1. An example of local minimum.

We propose a novel geographic routing protocol "HOle-BYpassing routing with Context-AwareNess (HobyCan)". In our approach, sensor nodes locally set up multiple detour paths to bypass *routing holes* in WSNs. Detour paths are disjoint from each other, while packets can be transferred from one detour path to another based on the context of the network. Such mechanism aims at finding optimal routing paths, as well as balancing of routing load between sensor nodes.

Context is any information that can be used to characterize the situation of an entity [11]. When referring to WSNs, context information mainly represents power profiles of sensor nodes, their storage and processing capability, network topology, as well as communication traffic. We take this work as the first step to integrate various kinds of context information in our routing protocol. In this paper, the context-information whether a node is a *local minimum* and its remaining energy are used as routing parameters. The main contribution of this paper is the construction and refinement of multiple detour paths for packets to bypass *routing holes* in WSNs. Furthermore, the approach effectively prevents fast energy depletion on nodes near holes by distributing load to additional nodes around a routing hole.

This work is financed by the German Research Foundation (DFG) (post graduate program MuSAMA, GRK 1424).

II. Related Work

Various geographic routing protocols [3] have been proposed in recent years to address the *routing hole* problem of WSNs. Most of the routing protocols start with *greedy forwarding* and recover from *local minimum* with different strategies. The objective of those protocols is to navigate routing paths around holes in WSNs.

The Greedy Perimeter Stateless Routing (GPSR) is one of the fundamental protocols based on *planar graphs* [8]. A *planar graph* represents the same connectivity as the original network with non-crossing edges. GPSR uses *greedy forwarding* and switches to *perimeter routing* mode when a *local minimum* is reached. The *right-hand rule* is employed in the *perimeter routing* mode, where packets are forwarded along the edge counterclockwise on the face of a *planar graph*. Bose et al. proposed the FACE-1 and FACE-2 [9] algorithms that use the perimeter of the *planar graph* formed at each node. Such approaches have high success rate, but high extra cost due to the maintenance of the *planar graph* information on sensor nodes.

In [7], Jia et al. presented the idea of Hole Avoiding In advance Routing protocol (HAIR) to bypass holes in advance. In WSNs, packets are typically routed from sensor nodes to a data sink. The protocol takes advantage of such "many-to-one" communication characteristic. At the first stage of HAIR, when a node recognizes itself as a *local minimum*, it asks its neighbors to mark itself as a *hole node*. Packets are sent to *non-hole nodes* when possible. As the process goes on, more nodes are marked as *hole nodes* and the range of *routing holes* are propagated as illustrated in Fig. 2. In contrast to other algorithms that start to bypass a hole only when a hole is met, HAIR achieves shorter routing paths, and thus reduces energy consumption.

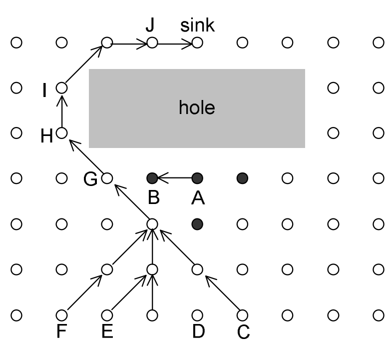


Fig. 2. An example of the HAIR routing protocol.

In the protocols mentioned above, all routing paths leading to a *local minimum* make a detour path along the contour of the hole. In Fig. 2, the nodes along the single detour path (*G-H-I···Sink*) suffer great energy depletion since it relays half the routing paths that are blocked by the hole. Nodes on such paths will soon run out of energy, which causes partial breakdown of the network. Besides, the hole becomes bigger. A new detour is required at this point, which is not considered in the above-mentioned protocols.

III. Protocol Design

The proposed HobyCan protocol addresses the load balance among sensor nodes, especially the nodes on the boundary of a hole. As mentioned above, the nodes on a single detour path fast drain their energy due to the heavy routing efforts around a hole. Therefore, we propose to construct multiple detour paths for a hole. For packet routing around the hole, a suitable path can be dynamically determined from the set of detour paths. As a result, the energy consumption is fairly distributed with more nodes on extra detour paths. Nodes on detour paths are called *detour nodes* in HobyCan.

To prevent the exhausting of energy on *detour nodes*, we introduce a threshold of the remaining energy of *detour nodes*, call *E_THR*. Specifically, when the remaining energy of a *detour node* is reaching a predefined threshold value, it will be recognized as energy-critical and be released from routing activities. Furthermore, we also include the possibility of establishing new detour paths, when all the existing detour paths are energy-critical.

HobyCan protocol starts from a WSN after its deployment, where sensor nodes are assumed to be static. In HobyCan, routing paths are dynamically constructed during regular forwarding of packets. Therefore, we avoid a dedicated phase of detour path construction, and reduce the cost of control message overhead. For illustration purpose, Fig. 3 shows a uniform network where routing paths are constructed via packets from node *X* to the sink.

Like most of the related geographic routing protocols, HobyCan starts with *greedy forwarding* and turns to the *perimeter mode* when approaching a hole. In the *perimeter mode*, packets are routed using multiple detour paths. In the

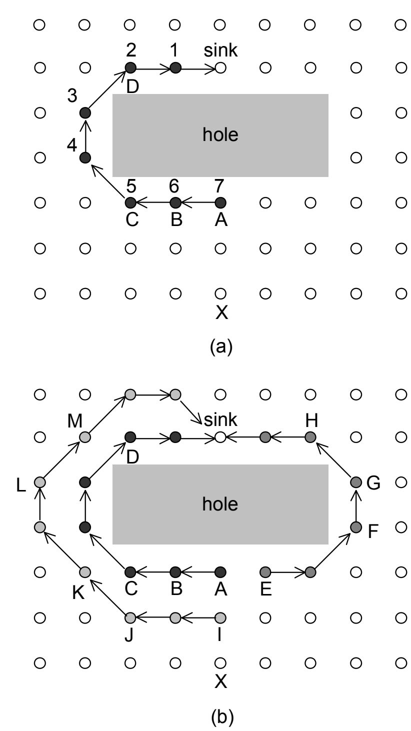


Fig. 3. An example of the HobyCan routing protocol.

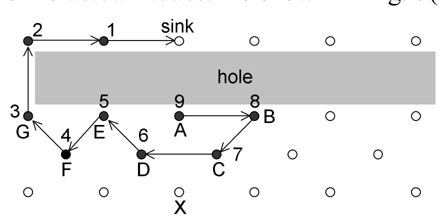
following part of this section, we will describe how to construct and use multiple detour paths with awareness to the context information of a WSN.

A. Construction of Paths

In the early phase of HobyCan, a predefined number of detour paths around hole(s) are constructed. We partly use the technique described in [7] to find detour paths around a hole. In contrast to [7] that simply propagate the dimension of a hole, our protocol constructs multiple detour paths, and uses these paths alternatively based on their context during packet routing. As shown in Fig. 3(a), HobyCan starts to construct a new detour path as soon as a *local minimum* (node *A*) is met. When there is no neighbor that is closer to the data sink, node *A* becomes a *detour node* and marks itself as the starting point of the new detour path. Node *A* selects one unmarked neighbor (node *B*) which is the nearest to data sink to forward the packet. During the construction of detour paths, packets are not sent to the marked *detour nodes*. The process continues until the sink is reached. As a result, a detour path (*A-B-C-D···Sink*) in Fig. 3(a) is established as the first detour path for the routing hole.

On the arrival of the new detour path (*A-B-C-D···Sink*), the sink assigns a unique path ID to it. A *CONSTRUCTION* message with incrementing hop counter is sent from the sink and routed backwards along the new detour path until the starting point (node *A*). In that way, nodes on the path are assigned with their hop distance to the sink. For instance, the *distance labels* of node *A*, *B*, *C* and *D* are 7, 6, 5 and 2 respectively. Such distance labels depend on the relative position of the hole and the destination of packets. The proposed HobyCan protocol is designed for the common "many-to-one" communication model of WSNs, where packets are sent from sensor nodes towards a single data sink. Therefore, the *distance labels* of *detour nodes* can be used for any pair of source and destination (the sink) addresses.

During the phase of path construction, packets are not forwarded to the *detour nodes*. As shown in Fig. 3(b), when a



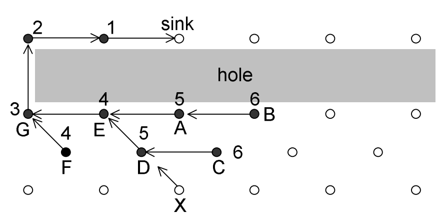


Fig. 4. An example of *path refinement* of HobyCan.

local minimal (node *E*) is met, construction of a new detour path can be started. New constructed detour paths for the same hole are indicated with incrementing numbers. When the predefined number of detour paths is 3, disjoint detour paths (*A-B-C-D···Sink*, *E-F-G-H···Sink* and *I-J-K-L-M···Sink*) are obtained using the same method mentioned earlier.

B. Path Refinement

The above-mentioned algorithm can construct detour paths in a straight-forward way. However, we point out that in non-uniform deployments, the resultant paths are probably twisted and need to be resorted. We introduced the phase of *path refinement* in HobyCan. As illustrated in Fig. 4(a), node *A* finds itself a *local minimum* and selects node *B* as the next hop. Among all neighbors of node *B*, node *C* is the nearest to the sink. An indirect detour path *A-B-C-D-E-F-G···Sink* is then constructed as in Fig. 4(a). Here, the *distance labels* of node *A*, *B*, *C*, *D*, *E*, *F* and *G* are 9, 8, 7, 6, 5, 4 and 3 respectively.

On the arrival of the path, the *CONSTRUCTION* message is sent by the sink and routed backwards along the new detour path. Each *detour node* broadcasts the *CONSTRUCTION* message to its direct neighbors. If a neighbor is a *detour node* of the same detour path, it connects itself to the neighboring *detour node* with the smallest *distance label*. After the *path refinement*, a packet at node *X* joins the detour path by choosing the nearest *detour node* to the sink, namely the *detour node* with the smallest *distance label* (node *D*). The mechanism applies also to the packets on *detour nodes*. In Fig. 4(b), a packet appearing at *detour node A* chooses neighboring *detour node E* as the next hop since the *detour node E* is the neighbor has the minimal *distance label*. After refinement, a detour path may have a tree-like structure as in Fig. 4(b).

C. Use & Maintenance of Paths

In the *perimeter mode* of HobyCan, routing of packets can be switched among different detour paths. A *detour node* with the minimal *distance label* and non-critical remaining energy is selected as the next hop. The amount of detour paths for the same hole is defined by the context of the network. One aspect regarding context is the size of a hole. More detour paths will be needed when a hole shadows a great number of nodes from the perspective of the sink. An advantage of this scheme is that the contour of a hole can be propagated with multiple detour paths as in Fig. 3(b). Beside, packets can be forwarded with fewer hops because the extended contour propagates the information of the existence of a hole in advance.

Another advantage is that load balance among nodes is achieved by choosing next-hop nodes with non-critical remaining energy from other detour paths. Since detour paths are likely to be close to each other on the border of a routing hole, a *detour node* may have *detour nodes* of other detour paths as its direct neighbors. The selection of detour path is based on the network context, such as distance to the sink or remaining energy of the next-hop node. For example, as illustrated in Fig. 3(b), a packet at node *J* will be forwarded to node *C* to achieve a shorter routing path. If the battery capacity of node *C* is under the predefined threshold of remaining energy *E_THR,* node *K* will be selected as the next hop to prevent the energy depletion of node *C*. A *detour node* needs to periodically refresh the current context information of its direct neighbors. In our current design, *detour nodes* reaching the threshold *E_THR* would broadcast a NOT_SEND_TO_ME message to their direct neighbors. *Detour nodes* receiving such a message will not consider its sender as the next hop. Therefore, the control overhead of using context information is limited to *O(N)*, where *N* is the number of nodes in a WSNs.

The amount of detour paths can also be limited by the number of neighbors of the data sink. In Fig. 3(b), the sink can be reached via its 5 neighbors, which implies the maximal number of detour paths which are disjoint from each other. Such limit is addressed in HobyCan by setting a number of nodes around sink as *joint points*, where detour paths can join each other. Another issue is the high energy consumption of nodes near the sink, which is addressed in other works [10] but is out of the scope of this paper.

After the phase of *path construction* and *path refinement*, most nodes are expected to find routing paths to the sink. However, since the construction of detour paths is triggered by unscheduled packet routing, there might remain unexplored *local minimums* in the network. New detour paths can be set up starting from the unexplored *local minimums* until they reach either the sink or any existing detour paths. If necessary, more detour paths can also be added during the run time of packet routing. When existing paths are all energy-critical (reaching the threshold of remaining energy *E_THR*), setup of new paths can be carried out with the steps described above.

IV. Simulation Results

We simulated the HobyCan protocol with Matlab. The simulated network had a dimension of 500 m × 500 m, where 500 sensor nodes were randomly deployed excluding a rectangular area which represented the hole in the network. A single data sink was located on the right side of the square plane. A hole with the size of 100 m × 200 m was located in the middle of the network. We used a simple *disc-communication model*: sensor nodes in the transmission range can receive signals from a transmitter without loss. To further simplify the network model, we assumed that each transmission of signal consumes 1 unit of energy.

In our demonstration, the value of *distance labels* (the remaining hop count to the sink on *detour nodes*) and the remaining energy of nodes were used as routing parameters in the *perimeter mode* of HobyCan. According to our protocol, nodes with smallest *distance labels* were selected first as next-hop nodes. Nodes are initialized with energy of 100 units. When the depletion of energy on a node reached 20 units (*E_THR* =80), the node was released from routing tasks. This applied only to the *detour nodes*. We simulated the HobyCan protocol with 1, 3, 5 detour paths constructed in the early phase. As mentioned above, additional detour paths can be established when existing paths are all energy-critical (20 units of energy consumption).

The objective of the experiment is to demonstrate the advantage of multiple detour paths, and the impact of selected context information as routing parameters. The results were compared with the HAIR protocol regarding success rate and load balance. Since such routing protocols emphasize on the performance when bypassing hole(s), we only sent packets from the left side of the network to the right side (the sink). Therefore, all packets would bypass the hole.

We use the following metrics to evaluate the performance of the simulated routing protocols:

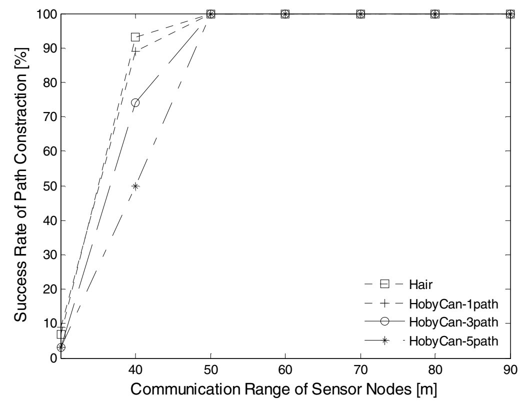


Fig. 5. Success rate of path construction with different communication ranges.

Success rate of path construction: Defined as the rate of successful path construction from the selected source node to the data sink. This metric illustrates the hole-bypassing ability of the simulated protocols.

Fig. 5 shows the percentage of successful path construction of the simulated protocols. *HobyCan-3path* and *HobyCan-5path* have lower success rates in this case, because more nodes are required to construct multiple detour paths. When provided with enough nodes for construction of detour paths (when communication range is bigger than 60m), HobyCan has the same success rate as HAIR, since they share the same algorithm for finding detour paths.

Average length of routing paths: Defined as the average hop counts of packets from the source node to the sink. This metric

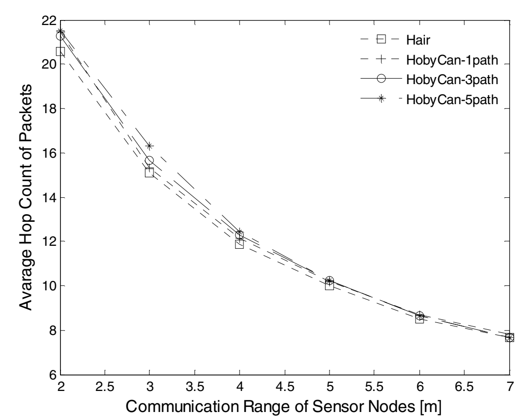


Fig. 6. Average hop count with different communication ranges.

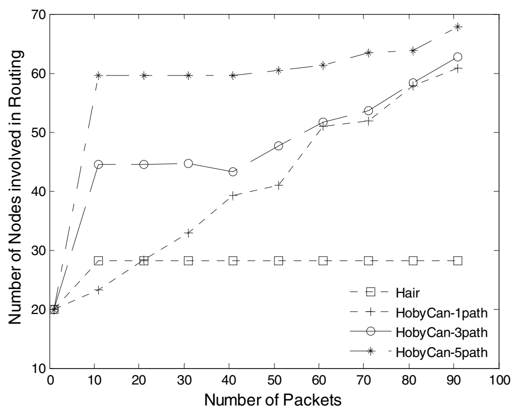


Fig. 7. Number of nodes involved in routing during simulation.

reflects the routing efficiency of the compared protocols.

As illustrated in Fig. 6, the length of routing paths decreases with the increasing of the communication range. HAIR is supposed to have short paths because it avoids hole(s) "in advance". However, we notice that in random deployments, a packet usually needs to try a large number of nodes in the area near a *local minimum*, before it can find a way to bypass the hole. This is due to the twisted route of finding detour paths (Fig. 4(a)). In HobyCan, such a process marks all the visited nodes as *detour nodes*. The refinement of such a detour path (Fig. 4(b)) results in a near-optimal routing path along the contour of the visited area. Therefore, the resultant routing paths in HobyCan achieve similar average path length compared with HAIR.

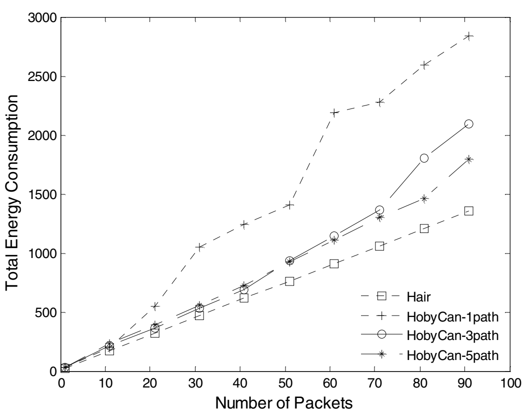


Fig. 8. Total energy consumption during simulation.

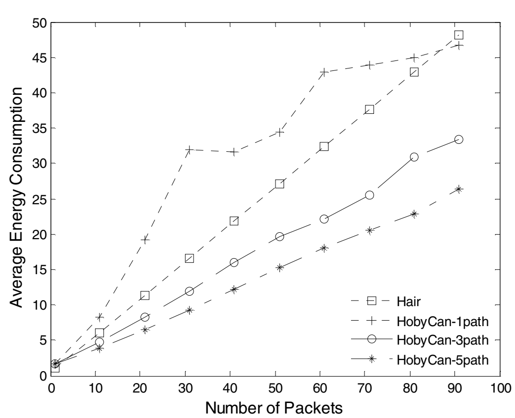


Fig. 9. Average energy consumption during simulation.

Number of nodes involved in routing: In our simulation, packets are always sent through the network from the left side to the right side. The number of nodes involved in routing activities along with the sending of packets can map the amount of routing paths created in the network. This metric shows the features of task distribution of different protocols. The communication range of sensor nodes is set to 50m here.

Fig. 7 depicts the results regarding the number of nodes involved in routing against the number of packets send. Because HAIR assigns the nodes static routing paths, it employs a fix number of nodes after the construction of a single detour path. In contrast, HobyCan constructs multiple detour paths around the routing hole. Compared to HAIR, *HobyCan-3path* and *HobyCan-5path* involve more nodes in their routing paths from the start. As the routing process continues, nodes on the detour paths of *HobyCan-1path* reach the threshold of energy consumption (20 units). Construction of new detour paths is then carried out for *HobyCan-1path*. Similarly, more and more nodes in HobyCan are recruited for detour paths as shown in Fig. 7.

Average energy consumption: Defined as the total energy consumption in the network divided by the number of nodes involved in routing. This metric reflects the performance of the simulated protocols in terms of load balance. The communication range of sensor nodes is set to 50m here.

The proposed HobyCan protocol has additional control overhead: when a new detour path is constructed, a *CONSTRUCTION* message should be routed backwards along the path to set up the *distance labels* of the *detour nodes*. Therefore, the total energy consumption of HobyCan is higher compared to HAIR. However, in Fig. 8, we observe that only the energy consumption from *HobyCan-1path* is significantly higher than HAIR's. This is due to the frequent search of additional detour paths. For the HobyCan protocol with more paths constructed in advance (e.g. *HobyCan-3path* and *HobyCan-5path*), the control overhead is rather small compared to the energy consumption of routing efforts.

As illustrated in Fig. 9, the average energy consumption of *HobyCan-3path* and *HobyCan-5path* is lower than HAIR. The advantage becomes more significant as more and more nodes are involved in routing efforts. Due to its control overhead, *HobyCan-1path* results in higher average energy consumption than HAIR at the beginning. Since the maximal energy consumption on *detour nodes* is 20 units, *HobyCan-1path* establishes new detour paths and distributes the routing task to new *detour nodes* from time to time. As the process continues, *HobyCan-1path* finally exceeds HAIR in terms of average energy consumption.

V. Conclusions and Future Work

In this paper, we proposed a novel geographic routing algorithm "HOle-BYpassing routing with Context-AwareNess

(HobyCan)". The proposed protocol dynamically constructs multiple detour paths around hole(s), and uses them alternatively for routing. Context information of WSNs is utilized during construction of detour paths and routing itself. HobyCan has a high success rate and short length of routing paths. The highlight of HobyCan is its ability to balance the routing load among nodes near routing holes.

Simulation results showed that the number of detour paths constructed in the early phase has a strong impact on the performance of HobyCan. In future, we intend to study the relationship between the number of detour paths in HobyCan and the deployment of WSNs. Depending on specific application scenarios, various kinds of context information (e.g. buffer size of sensor nodes, link quality, priority of events, etc.) can be utilized in HobyCan. We plan to consider more kinds of context information in our protocol, as well as their impacts as routing parameters.

References

[1] B. Warneke, M. Last, B. Liebowitz, Kristofer and S. J. Pister, "Smart Dust: Communicating with a Cubic-Millimeter Computer", Computer Magazine, Vol. 34, No. 1, pp 44-51, Jan. 2001.

[2] K. Akkaya and M. Younis, "A survey of routing protocols for wireless sensor networks", Elsevier Ad Hoc Network Journal, 2005, Vol. 3/3, pp. 325-349.

[3] Nadeem Ahmed , Salil S. Kanhere , Sanjay Jha, "The holes problem in wireless sensor networks: a survey", ACM SIGMOBILE Mobile Computing and Communi-cations Review, v.9 n.2, April 2005.

[4] Holger Karl and Andreas Willig. "Protocols and Architectures for Wireless Sensor Networks", Wiley, 2005.

[5] Fengh Zhao and Leonidas Guibas. "Wireless Sensor Networks, an Information Processing Approach", Elsevier, 2004.

[6] G. G. Finn. "Routing and addressing problems in large metropolitanscale internetworks", Technical Report ISI/RR-87-180, Information Sciences Institute, Mars 1987.

[7] W. Jia, T. Wang, G. Wang, M. Guo, "Hole Avoiding in Advance Routing in Wireless Sensor Networks", in Proceedings of WCNC 2007.

[8] Brad Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks", in Proceedings of the ACM/IEEE MobiCom '2000, pages 243-254, 2000.

[9] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks", Wireless Networks, 7:609-616, 2001.

[10] Dali Wei; Chan, H.A.; Kameri, K.V.N., "Circular-Layer Algorithm for Ad Hoc Sensor Networks to Balance Power Consumption", Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. vol.3, no., pp.945-950, 28-28 Sept. 2006.

[11] A. K. Dey, D. Salber and G. D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", Human-Computer Interaction, 16(2-4): pp. 97-166, 2001.