Greedy Geographic Routing with Path Optimization in Wireless Sensor Networks

Thanh Le Dinh
Dept. of Networks and Computer Communications
University of Engineering and Technology, VNU-HN
Hanoi, Vietnam
thanhld@vnu.edu.vn

Dai Tho Nguyen
Dept. of Networks and Computer Communications
University of Engineering and Technology, VNU-HN
Hanoi, Vietnam
nguyendaitho@vnu.edu.vn

***Abstract*—In this paper, we propose Greedy with Path Optimization Routing (GPOR), a novel geographic routing protocol for wireless sensor networks. GPOR finds initial routing paths by following a greedy with recovery strategy, then uses a follow-up technique to optimize the paths. An attempt is also made to create routing entries applicable to destination areas rather than individual nodes. Main advantages of GPOR are path optimization and void avoidance capacities. We implement GPOR in ns-2 and present simulation results.**

Keywords - greedy forwarding; geographic routing; wireless sensor networks; routing cache.

I. INTRODUCTION

Greedy forwarding [1] is the strategy that uses information on the position of the nodes to forward packets node by node towards the destination. In greedy forwarding, the neighbor closest to the destination and closer to the destination than the current node is chosen as the next hop. Greedy forwarding will fail at nodes locally closest to the destination. These nodes are referred to as local minima or dead-end nodes. The region within the radio range of a node and not containing neighbors closer to the destination than the current node is termed a *void* [2].

Greedy forwarding is *lightweight* in the sense that it requires only information on the position of neighboring nodes. Ás information on the position of neighboring nodes is updated quickly and efficiently, greedy forwarding can adapt very well to network changes. By maintaining only local topology information, greedy forwarding copes with increases in the number of network nodes without problems. Upon success, greedy forwarding produces nearly shortest paths. It rarely fails in dense networks. In short, geographic routing based on greedy forwarding promises an efficient, adaptive and scalable approach to wireless sensor networks.

Greedy geographic routing primarily uses greedy forwarding to forward packets towards destinations. When greedy forwarding fails, geographic routing switches to recovery state during which a backup recovery strategy is used in order to route packets to a node where greedy forwarding can be resumed, i.e. the node closer to the destination than the last local minimum. Many lightweight recovery strategies have recently been proposed such as face routing [2], [3], [4], [5], [6], [7], boundary detouring [8], [9]. As a result, geographic routing combining greedy forwarding with a recovery strategy can be very lightweight. Two typical protocols of this class are GPSR [2] and BOUNDHOLE [8]. The main disadvantage of these geographic routing protocols is that they produce long paths when packets manage to bypass the holes.

In this paper, we address the problems of nearly optimal path discovery and void avoidance of geographic routing and propose *Greedy with Path Optimization Routing* (GPOR) protocol. GPOR uses lightweight geographic routing to find initial routing paths, and then improves the paths with a follow-up technique. We create routing entries applicable to destination areas rather than individual nodes. The advantages of GPOR include path optimization and void avoidance. We implement GPOR in ns-2 [9] and present simulation results.

Related works are reviewed in Section II. GPOR is presented in Section III. Then, the evaluation of its performances is described in Section IV. Finally, Section V gives our conclusion and future works.

II. RELATED WORK

Greedy forwarding [1] is simple, scalable and efficient but suffers from local minima. To overcome this problem, greedy forwarding is combined with a recovery strategy. When greedy forwarding fails, the recovery strategy is used in order to route packets to a node where greedy forwarding can be resumed. Recovery strategies have been extensively studied. As a result, many recovery strategies have been proposed. These strategies can be classified into four main classes: *flooding*, *backtracking*, *face routing* and *boundary touring*.

A simple and inefficient way to deal with local minima is to use *flooding*. The protocol in [10] broadcasts the packet if it reaches a local minimum. The protocol in [11] initially uses greedy forwarding to forward packets towards their destinations. When greedy forwarding fails, a route discovery process is launched and managed in the depth first search manner. Routing information is cached, thus routing tables are accumulatively built up at the nodes. These routing tables are then used for conventional routing in place of greedy forwarding.

In [12], *backtracking* is used for recovery purpose. If a node becomes a local minimum, it reports a message “I am a hole-node” to its neighbors then sends packets to the neighbor closest to the destination. Upon receiving a report message, the receiver marks the sender of the report message as a hole-node in its list of neighbors; if all neighbors of the receiver have declared being hole-node, it also declares itself as a hole-node. The procedure is repeated until the packet reaches the destination or is discarded because of timeout.

Face routing is a recovery strategy that has been studied extensively. This strategy is twofold. It proactively extracts a connected planar sub-graph from the underlying communication graph and embeds this sub-graph to the nodes. When greedy forwarding fails, it uses a traversal technique on the embedded graph to route the packet to a node where greedy forwarding can be resumed. GFG [3] uses Gabriel graph and face routing [4] for recovery purpose. GPSR [2] uses Relative Neighborhood graph and the right-hand rule to route packets along the perimeter of holes. Many other face routing protocols have been proposed [4], [5], [6], [7]. The main problem with this family of routing protocols is that existing planar graph extraction methods may produce incorrect planar graphs in case of non-uniform radio patterns and thus can cause the routing to fail.

The fourth technique used in recovery state is *boundary detouring*. When a packet reaches a local minimum on a boundary, BOUNDHOLE [4] uses sweeping to route the packet along the boundary to a node that is closer to the destination than the last local minimum. GRIC [13] is another detouring technique. It is based on inertia principle. The core of the authors' argument is that while packets are attracted to their destination, their movement is also affected by inertial forces. Movement in the direction of the destination naturally ensures optimal performance. Meanwhile, inertial forces allow the packets to move along the current direction and follow the perimeter of obstacles in order to bypass them.

Face routing and boundary detouring are both lightweight. Their main disadvantage is that they produce long paths when packets manage to bypass holes.

Protocols in [14] and [15] provide a trust-based path optimization and void avoidance scheme. Nodes are initially provided with equal reputation. When a node becomes a local minimum, its reputation is reduced. Nodes with low reputation are not considered when choosing the next hop. Routing paths slowly converge to the optimal ones. At the same time, voice avoidance capacity is built up.

In [16], another path optimization scheme is proposed. Obstacles on the way to the destination are marked by beacon nodes which guide packets to bypass them. As the areas marked in front of obstacles gradually get wider, data paths increasingly get closer to the optimal ones.

In the next section, we describe our proposed GPOR protocol for wireless sensor networks.

III. Greedy with Path Optimization Routing

GPOR is a hybrid routing protocol. It uses guided forwarding based on routing cache ahead of greedy forwarding for making routing decisions. In this section, the routing cache, guided forwarding and routing based on routing cache, the maintenance of routing cache, and the advantages of GPOR compared to that of other protocols are presented, consecutively.

A. *Routing Cache*

Each node has a cache for routing entries whose format is described in Table I.

TABLE I. Format of Routing Entries

Field	Description
pos	Targeted position
next	Identifier of the neighbor that may be chosen as the next hop if the distance from the destination to the position *pos* is not greater than *r*, where *r* is the radio range of the nodes.
ttl	Time to live

An entry is created as a node forwards a data packet to a destination. Its *pos* is set to the position of the destination of the data packet and its *next* is set to the next hop. Note that unlike that of topological routing, each GPOR's routing entry is not for a particular destination, but a group of destinations that are geographically close to a defined position. In other words, though the *pos* is the position of a node, an routing entry is applicable to every node that is close to its *pos*. We refer to this property of routing entries with the term "*area applicability*". Area applicability gives opportunity for efficient path optimization and void avoidance as presented in the Section III.D.

B. *Guided Forwarding*

We say that a routing entry <pos_x, $next_x$, ttl_x> is applicable to the destination *d* if its ttl_x is greater than 0 and the distance from *d* to the position pos_x is not greater than *r*, where *r* is the radio range of the nodes. *Guided forwarding* is described as follows: The current node looks up its routing cache for entries that are applicable to the destination. Then, it selects the applicable entry whose *pos* is closest to the destination, and forwards the packet to the *next* of the selected entry.

In the next sub-section, that guided forwarding is used in concert with lightweight geographic routing is presented. At the same time, the scheme which makes use of guided forwarding, greedy forwarding and recovery forwarding to provide good path optimization and void avoidance is proposed.

C. *Routed and Routing Cache Buildup*

Initial routing paths are discovered on demand then the route-follow-up technique is used in order to efficiently build better paths. To accomplish this, *k* last traveled hops p_k, p_{k-1}, …, p_1 are recorded in the header of data packets and route follow up (RF) control packets are used.

Like other geographic routing protocols, GPOR (see Figure 1) has two modes of data packet forwarding: *greedy* mode and *recovery* mode. Unlike other geographic routing protocols, GPOR forwards greedy data packets by guided forwarding firstly, if guided forwarding fails then greedy forwarding is used. A data packet is generated with a greedy mode, is set to recovery mode at local minima where guided forwarding fails, and is set to greedy mode again at nodes closer to the destination than the last local minimum where guided forwarding fails.

The route-follow-up technique is used for the buildup of routing cache. This technique is described as follows. After forwarding a data packet whose destination is *d* to the next hop *n* in recovery mode or by guided forwarding, the current node (1) adds entry <*d.pos*, *n*, TTL> to its routing cache, where TTL is a constant indicating the time to live of this entry, and (2) generates *k*-1 RF packets <RF, p_k, *d.pos*, TTL>, <RF, p_{k-1}, *d.pos*, TTL>, …, <RF, p_2, *d.pos*, TTL> whose destinations are p_k, p_{k-1}, …, p_2, respectively, then sends these packets to defined destinations by greedy forwarding. RF packets are routed to their destinations solely by greedy forwarding. On receiving a RF packet <RF, *t*, *pos*, *ttl*> from neighboring node *q*, the receiver adds entry <*pos*, *q*, *ttl-1*> to its routing cache, reduces the *ttl* of the packet by 1 then forwards the packet to the neighbor closer and closest to *t*.

Upon receiving a data packet:

If I am the destination of the data packet then

Send the data packet to the upper network layer

Else

Update the list of k last traveled hops in the header of the data packet

If the data packet is in greedy mode then

Call *Guided-Greedy*

Else

Call *Recovery*

Upon receiving a route follow up packet <RF, t, pos, ttl> from neighboring node q:

Add entry <*pos, q, ttl-1*> to my routing cache

Reduce the *ttl* of the packet by 1 then forward the packet to the neighbor closer and closest to *t*.

Guided-Greedy:

If there are applicable routing entries in my cache then

Remove all routing entries for *d.pos* then add <*d.pos, n,* TTL> to my routing cache, where *d* is the destination of the data packet and *n* is the next hop specified by the applied routing entry.

Forward the data packet to *n*

Send <RF, p_k, *d.pos*, TTL>, <RF, p_{k-1}, *d.pos*, TTL>, …, <RF, p_2, *d.pos*, TTL> to p_k, p_{k-1}, …, p_2, respectively, where p_k, p_{k-1}, …, p_2, p_1 are the *k* last traveled hops recorded in the header of the data packet.

Else

Forward the data packet by greedy forwarding

If greedy forwarding fails then

Record me to the header of the data packet as the last local minimum

Set the data packet to recovery mode

Call *Recovery*

Recovery:

If I am closer to the destination than the last local minimum reached by the data packet then

Set the data packet to greedy mode

Call *Guided-Greedy*

Else

Remove all routing entries for *d.pos* then add <*d.pos, n,* TTL> to my routing cache, where *d* is the destination of the data packet and *n* is the next hop chosen by the used lightweight geographic routing.

Forward the data packet to *n*

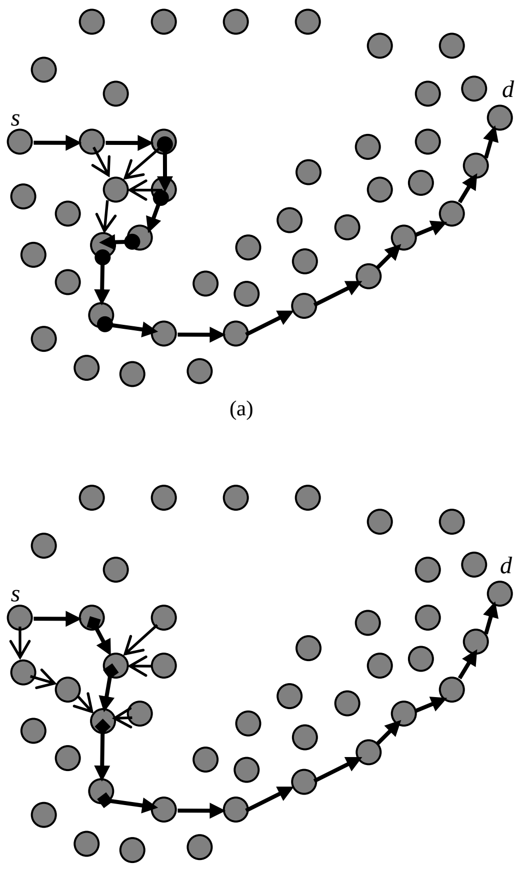
Send <RF, p_k, *d.pos*, TTL>, <RF, p_{k-1}, *d.pos*, TTL>, …, <RF, p_2, *d.pos*, TTL> to p_k, p_{k-1}, …, p_2, respectively, where p_k, p_{k-1}, …, p_2, p_1 are the *k* last traveled hops recorded in the header of the data packet.

Figure 1. The GPOR, code for node *p*.

For memory efficiency, every node stores at most one entry for a targeted position in its cache. Therefore, on adding a new entry to routing cache, nodes discard entries for the same targeted position that do not have the best *ttl*. Additionally, every node periodically cleans its routing cache: reduces the *ttl* of each of its routing entries by 1 and removes entries with zero *ttl*.

D. Advantages

Path optimization is the main advantage of GPOR. Path optimization is gained in both single traffic flow and multiple concurrent traffic flows. Figure 2 gives an example of path optimization in a single traffic flow. In this example, nearly optimal path is constructed when three data packets and their associated RF packets finish their journeys.



(b)

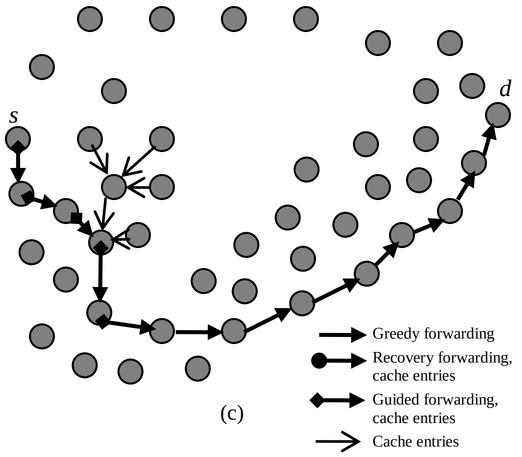


Figure 2. Routing paths constructed by
(a) a data packet and its associated RF packets.
(b) two data packets and their associated RF packets.
(c) three data packets and their associated RF packets.

In the example given in Figure 3, two concurrent traffic flows are established and the two destinations d_1, d_2 are close to each other. Area applicability of routing entries becomes effective. Routing entries that are applicable to d_1 are also applicable to d_2. Thus, the routing path from s_2 to d_2 can successfully avoid a void.

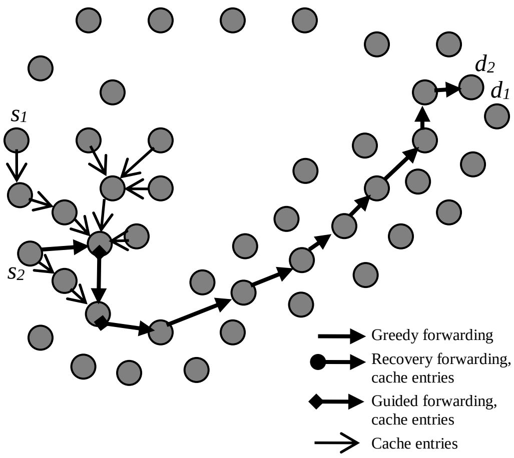


Figure 3. Path optimization and void avoidance in concurrent traffic flows. By applying routing entries for d_1, routing path from s_2 to d_2 successfully avoids a void.

The second advantage of GPOR is that it provides void avoidance capacity. Void avoidance can be illustrated by example given in Figure 3. In this example, every data packet whose destination is close to d_1 or d_2 passing nodes with non-empty routing cache will successfully avoid a void.

E. Memory Efficiency

Each node maintains a routing cache for guide forwarding. For each traffic flow, at most one entry is stored in a node. Thus, the maximal number of entries in a routing cache will not excess much the number of concurrent traffic flows because entries for past traffic flows are discarded by *ttl* condition.

F. Comparison to other Protocols

Protocols in [14] and [15] provide slower path optimization in comparison to GPOR. These protocols take time to form areas of nodes with bad reputation. On the other hand, scheme in [16] suffers from hole problem in constructing beacon paths. If there are holes on the path from the first returning point to the source, for example, the beacon path may be blocked. Next, the constructing of beacon paths requires a certain amount of time, and the improved paths can only be used when the beacon paths are completely constructed. Additionally, a beacon path is applied to only one destination. As a result, the overall path optimization in multiple concurrent traffic flows is not as good as GPOR.

IV. SIMULATION

We implement GPOR in the open-source network simulator ns-2 (v.2.33) [9]. BOUNDHOLE [8] is used for initial routing path discovery. Then, extensive simulations are performed. Simulation results show that GPOR provides higher packet delivery rate in scenes with multiple concurrent traffic flows, and produces shorter paths in comparison with that of BOUNDHOLE [8].

We evaluate the performance of GPOR in three metrics: packet delivery rate, routing protocol overhead, and the average length of paths taken by successful data packets. Then, we compare the results with those of BOUNDHOLE [8].

In order to meet our simulation goal, we use scenes varying in the number of concurrent traffic flows and the network diameter. Nodes have the radio range of 250 m and beaconing interval of 1 second. Each simulation lasts for 900 simulated seconds and uses CBR traffic flows sending 64-byte packets at the rate of 2 Kbps. Each set of simulations (specified by a defined network diameter and the defined number of traffic flows) contains six simulations. We use the mean of each metric over these set of simulations. Tables II and III summarize the characteristics of simulations.

TABLE II. SCENES VARYING IN THE NUMBER OF TRAFFIC FLOWS

Nodes	Region	Density	CBR Flows
200	3000 m x 600 m	1 node / 9000 m^2	20
200	3000 m x 600 m	1 node / 9000 m^2	40
200	3000 m x 600 m	1 node / 9000 m^2	60
200	3000 m x 600 m	1 node / 9000 m^2	80

TABLE III. SCENES VARYING IN THE NETWORK DIAMETER

Nodes	Region	Density	CBR Flows
100	1500 m x 600 m	1 node / 9000 m^2	20
150	2250 m x 600 m	1 node / 9000 m^2	20
200	3000 m x 600 m	1 node / 9000 m^2	20
250	3750 m x 600 m	1 node / 9000 m^2	20

A. Effect of Concurrent Traffic Flows

Figure 4 shows the packet delivery success rate of GPOR and BOUNDHOLE in scenes with varying the number of concurrent traffic flows. Simulation results show that GPOR has a higher fraction of packet delivery success rate than BOUNDHOLE does. This result can be explained as follows. GPOR distributed traffic to non-boundary nodes while BOUNDHOLE routes all recovery data packets along the boundaries. As a result, congestion may occur more frequently at boundary nodes in BOUNDHOLE.

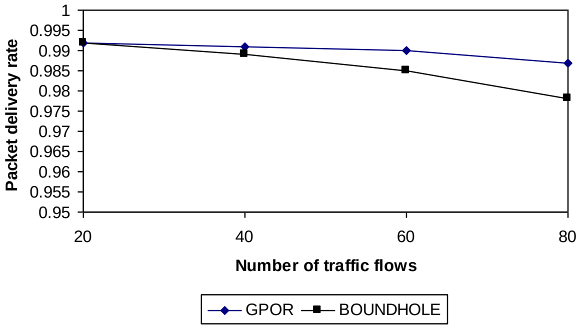


Figure 4. Packet delivery success rate with varying the number of concurrent traffic flows, *k* = 4.

Figure 5 shows the average number of hops traveled by GPOR's and BOUNDHOLE's data packets. Again, GPOR produces shorter paths than BOUNDHOLE does. With GPOR, as the number of concurrent traffic flows increases, the average path length is decreased. This simulation result proves the effectiveness of the area applicability of routing entries.

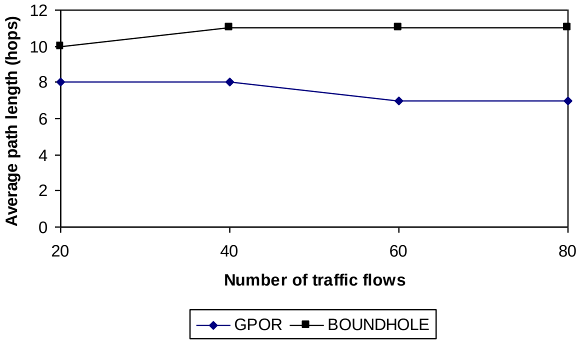


Figure 5. Average path length with varying the number of concurrent traffic flows, *k* = 4.

On the other hand, BOUNDHOLE is more efficient than GPOR. This can be ascertained by the fact that BOUNDHOLE uses solely beaconing control packets while GPOR uses additional RF packets. Figure 6 shows that the overhead of BOUNDHOLE is independent of the number of traffic flows while GPOR's overhead increases linearly with the increasing number of traffic flows.

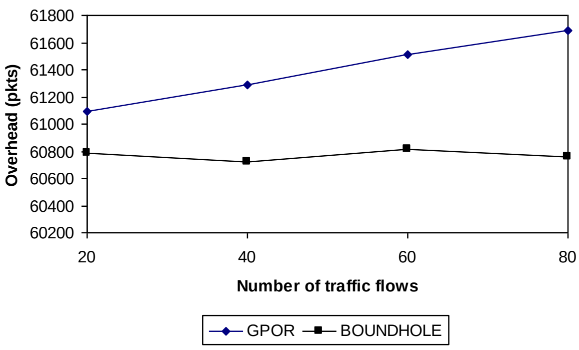


Figure 6. Overhead with varying the number of concurrent traffic flows, *k* = 4.

B. *Effect of Network Diameter*

Figure 7 shows that both BOUNDHOLE and GPOR provide quite stable packet delivery success rate while the network diameter changes. Figure 8 shows that overheads of BOUNDHOLE and GPOR increase linearly with the increasing network diameter. Last, Figure 9 shows that the larger network is, the better path optimization is gained.

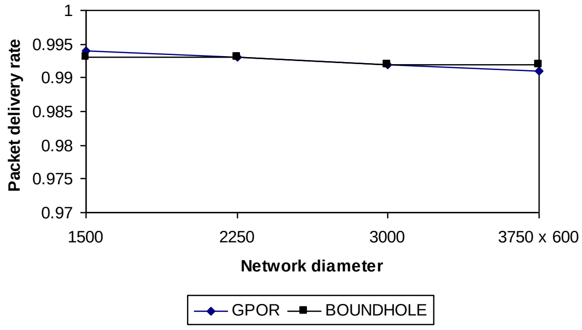


Figure 7. Packet delivery success rate with varying network diameter, *k* = 4.

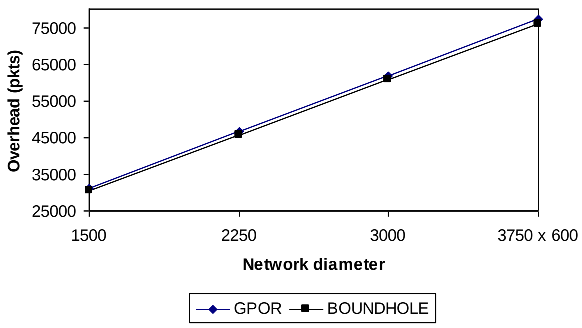


Figure 8. Overhead with varying network diameter, *k* = 4.

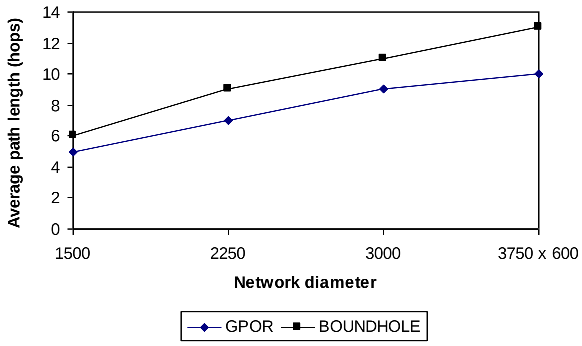


Figure 9. Average path length with varying network diameter, *k* = 4.

C. *Effect of the Number of Recorded Hops*

In order to set the best value to the number of hops recorded in the header of data packets, *k*, we use simulations varying in *k*. In more details, we vary the number of recorded hops *k* from 2 to 10 in simulations with the networks of 200 nodes, 20 CBR flows. Simulation results show that the number of recorded hops has no effect on the packet delivery success rate (Figure 10), the average path length is reduced as the number of recorded hops is increased (Figure 11), and the overhead is increased significantly as the number of recorded hops is increased (Figure 12). From above simulation results, we suggest that the used values for *k* should be 4, 5 and 6.

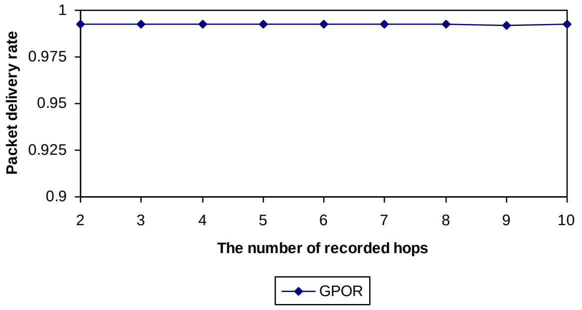


Figure 10. Pakcet delivery success rate with varying the number of recorded hops, 200 nodes, 20 CBR flows.

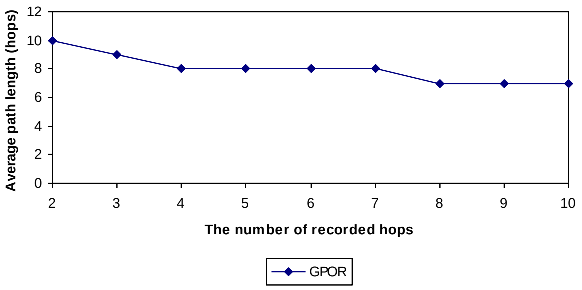


Figure 11. Average path length with varying the number of recorded hops, 200 nodes, 20 CBR flows.

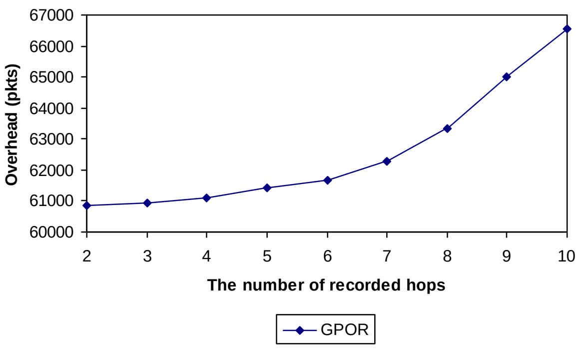


Figure 12. Overhead with varying the number of recorded hops, 200 nodes, 20 CBR flows.

V. CONCLUSION

We have introduced GPOR, a scheme for path optimization and void avoidance to geographic routing. While the area applicability provides the efficiently exploiting of routing entries, the route-follow-up technique creates useful routing entries on-the-flow, thus improves the paths continuously. Introducing these two advantages to lightweight geographic routing results in a new and better position-based routing paradigm.

We believe that position-based routing entries can be employed more efficiently than we have done in this work. So, developing rules for the more efficiently exploiting of position-based routing entries will be one of our future works. Besides, we intend to conduct a deeper research on route-follow-up techniques in order to make our position-based routing paradigm more efficient.

REFERENCES

[1] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetwork," Tech. Rep. ISI/RR-87-180, Information Sciences Institute, Mar. 1987.

[2] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless sensor networks," Proc. of Mobicom, pp. 243-254, 2000.

[3] S. Datta, I. Stojmenovic, J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," Proc. of the IEEE International Conference on Distributed Computing and Systems, pp. 461-466, 2001.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," Wireless Networks, 7(6):609-616, 2001.

[5] F. Kuhn, R.Wattenhofer, A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," Proc. of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Atlanta, Georgia, USA, pp. 24-33, 2002.

[6] F. Kuhn, R.Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," Proc. of PODC 2003, pp. 63-72, July 2003.

[7] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing: Geographic routing with local face information," Proc. of ICNP, pp. 147-158, 2005.

[8] Q. Fang, J. Gao, and L. Guibas, "Locating and bypassing routing holes in sensor networks," Mobile Networks and Applications, 11(2): 187-200, April 2006.

[9] The Network Simulator – ns-2. [Online]. Available: http://www.isi.edu/nsnam.

[10] I. Stojmenovic, X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," IEEE Transactions on Parallel and Distributed Systems, vol. 12, pp. 1023-1032, Oct. 2001.

[11] R. Jain, A. Puri, R. Sengupta, "Geographical routing using partial information in wireless ad hoc networks," IEEE Personal Communications, pp. 48-57, Feb. 2001.

[12] Zheng Guping, Wang Yu, "Advance detour strategy for geographic routing in wireless sensor networks," 2009 International Forum on Information Technology and Applications, pp. 296-299, 2009.

[13] S. Nikoletseas and O. Powell, "Simple and efficient geographic routing around obstacles for wireless sensor networks," Proc. of the 6th Workshop on Efficient and Experimental Algorithms, LNCS, Splinger-Verlag, pp. 161-174, 2007.

[14] L. Moraru, P. Leone, S. Nikoletseas, J. D. P. Rolim, "Near optimal geographic routing with obstacle avoidance in wireless sensor networks by fast-converging trust-based algorithms," Proc. of the 3rd ACM workshop on QoS and security for wireless and mobile networks, pp. 31-38, 2007.

[15] F. Huc, A. Jarry, P. Leone, L. Moraru, S. Nikoletseas and J. Rolim, "Early obstacle detection and avoidance for all to all traffic pattern in wireless sensor networks," ALGOSENSORS 2009, LNCS 5804, pp. 102–115, 2009.

[16] A. Koutsopoulos, S. Nikoletseas, J. D. P. Rolim, "Near-optimal data propagation by efficiently advertising obstacle boundaries," Proc. of the 6th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pp. 15-22, 2009.