2008 Second International Conference on Future Generation Communication and Networking

Geographic Routing in Wireless Multimedia Sensor Networks

Lei Shu[1*], Yan Zhang[2], Laurence T. Yang[3], Yu Wang[4], Manfred Hauswirth[1]
[1]Digital Enterprise Research Institute, National University of Ireland, Galway
[1]{lei.shu, manfred.hauswirth}@deri.org
[2]Simula Research Laboratory, Norway, email: yanzhang@ieee.org
[3]St. Francis Xavier University, Antigonish, NS, Canada, email: lyang@stfx.ca
[4]University of North Carolina at Charlotte, USA, email: wangyu@ieee.org

Abstract

In this paper, a Two-Phase geographic Greedy Forwarding (TPGF) routing algorithm is proposed in wireless multimedia sensor networks (WMSNs). TPGF is a pure geographic greedy forwarding routing algorithm, which does not include the face routing and does not use planarization algorithms. Simulation comparison in this paper indicates that TPGF is highly suitable for multimedia transmission in WMSNs.

1. Introduction

Generally, multimedia transmission in WSNs should consider the following three requirements: 1) *Multipath transmission*: Packets of multimedia streams are large in size. This requires that multipath transmission in WSNs should be used to increase the transmission performance. 2) *Hole-bypassing*: Dynamic holes may occur if several nodes in a small area overload due to multimedia transmission. Efficiently bypassing dynamic holes is essentially necessary. 3) *Shortest path transmission*: Multimedia applications generally have a delay constraint which requires that the multimedia streaming in WSNs should always use the shortest routing path which has the minimum end to end delay.

Multimedia transmission in WSNs requires a new routing algorithm to support these three requirements at the same time. This paper proposes a new Two-Phase geographic Greedy Forwarding (TPGF) routing algorithm for exploring one or multiple (near) shortest hole-bypassing paths in wireless multimedia sensor networks (WMSNs). The first phase is responsible for exploring the possible routing path. The second phase is responsible for optimizing the found routing path with the least number of hops. TPGF can be executed repeatedly to find multiple node-disjoint routing paths.

TPGF has the following features that make it be different from existing geographic routing algorithms [1-3]: 1) TPGF is a pure geographic routing algorithm. It does not include the face routing concept. 2) TPGF does not require the computation and preservation of the planar graph in WSNs. This point allows more links to be available for TPGF to explore more node-disjoint routing paths. 3) TPGF does not have the well-known *Local Minimum Problem* [1].

Research work in this paper has made practical contributions to understand the geographic routing in WMSNs: 1) *Supporting multipath transmission*: TPGF can find one routing path per execution and can be executed repeatedly to find more on-demand *node-disjoint* routing paths. 2) *Supporting hole-bypassing*: TPGF provides a better solution for hole-bypassing in both 2D and 3D WSNs than other related research work. 3) *Supporting shortest path transmission*: TPGF can find the shortest routing path (or near-shortest routing path when holes exist) for minimizing the end-to-end transmission delay.

In the rest of this paper: Section 2 presents the related work. Section 3 shows the network model and problem statement. Section 4 describes the algorithm and examples. Section 5 demonstrates simulation results, and section 6 concludes this paper.

2. Related work

2.1. Hole-bypassing in WSNs

Several research works on hole-bypassing routing in WSNs can be classified into: 1) *Hole-bypassing without knowing the holes information in advance***:** In [1], a geographic routing algorithm GPSR was

[*] This work was supported by the Lion project supported by Science Foundation Ireland under grant no. SFI/02/CE1/I131.

978-0-7695-3431-2/08 $25.00 © 2008 IEEE
DOI 10.1109/FGCN.2008.17

proposed and a *Local Minimum Problem* was identified. Before meeting the *Local Minimum Problem*, a node always chooses the next-hop node which is closer to the base station than itself. When it runs into a *Local Minimum Problem*, the face routing (*Right Hand Rule*) is adopted to solve the problem. Several other algorithms in [2-3], e.g., GOAFR+, and GPVFR were proposed subsequently. All these algorithms adopted the face routing to bypass holes. However, in [4], the authors reported that these geographic routing algorithms could not guarantee the delivery with arbitrary connectivity under realistic conditions. Furthermore, using the planarization algorithms, e.g., *GG* or *RNG*, to create a planar graph actually limits the useable links. However, in WMSNs, the number of usable links is not expected to be reduced since it has strong impact on the exploring result of multiple routing paths. 2) *Hole-bypassing with indentifying the holes information in advance***:** In [5, 6], the authors use graph theory to identify hole boundary nodes first, then use these identified boundary nodes to facilitate the hole-bypassing routing. Especially, in [6], every node is requested to identify twice whether it is a first-class node or a second-class node, which will consume a lot of energy. The actual routing algorithm executes after identifying these first-class and second-class nodes. In [7], the authors try to find an optimized hole-bypassing routing path by using hole geometric modeling after knowing the information of holes in advance. In this paper the hole information is obtained by using the algorithm proposed in [5]. All these algorithms can work correctly for identifying *static hole*s in WSNs, which can be formed by a set of dead sensor nodes due to energy exhaustion or damage. However, holes in WMSNs are more likely to be dynamic. Due to the large size of multimedia packet, transmission in WMSNs will generally use the maximum transmission capacity of each path. Any node that is transmitting multimedia data can hardly be reused for forming another path. When additional routing paths are needed for increasing the transmission performance, each new path should bypass the *dynamic hole* formed by the nodes of previous paths. In other words, the routing path nodes can enlarge the holes. Using the algorithms that proposed in [5, 6] to identify the hole/boundary nodes information in WMSNs after forming each new routing path is inefficient.

2.2. Geographic multipath routing in WSNs

Many multipath routing protocols have been studied [8]. However, most of the multipath routing protocols focus on load balance or fault tolerance, and they are the extended versions of DSR [9] and AODV [10]. Only a few research works adopt the geographic information to facilitate the on-demand disjoint multipath routing, e.g., [11, 12]. In [11], the authors proposed a Geography based Ad Hoc On demand Disjoint Multipath (GAODM) routing protocol in Ad Hoc networks. This GAODM uses the push-relabel algorithm to convert the Ad Hoc network as a flow network. The focus of this research work is how to use the push-relabel algorithm to find multiple node/edge disjoint paths based on the flow assignment. The routing paths found by GAODM are far from the optimal paths in terms of the end to end transmission delay. In [12], the authors proposed a node-Disjoint Parallel Multipath Routing algorithm (DPMR). This DPMR uses the algorithm proposed in [5] to identify the hole boundary first, then divides the identified hole into two regions (*clockwise region* and *unclockwise region*). When the *Local Minimum Problem* is met, the node always chooses a next hop only from either *clockwise region* or *unclockwise region*. Although, this research work breakes through the using of facing routing and planarization algorithms in geographic routing, it still has two key problems: it relies on the algorithm proposed in [5], and the restriction of using only either *clockwise region* or *unclockwise region* actually limits the number of routing paths. The found paths in [12] are also far from the optimal paths in terms of the end to end transmission delay.

3. Network model and problem statement

We consider a geographic WSN, which can be represented as a graph G (V, E), where $V = \{v_1, \ldots, v_n\}$ is a finite set of sensor nodes and $E = \{e_1, \ldots, e_n\}$ is a finite set of links. The base station can be randomly deployed. The locations of sensor nodes and the base station are fixed and can be obtained by using GPS. A finite set of nodes $V_{source} = \{v_{S1}, \ldots, v_{Sn}\}$ are source nodes. Each node has its transmission radius TR and M 1-hop neighbor sensor nodes. Each node is aware of its location and its 1-hop neighbor nodes' locations. Each node can have three different states: 1) *active and available*, 2) *active but unavailable*, and 3) *dead*. Each link can have two different states: 1) *available* and 2) *unavailable*. We assume that only source nodes know the location of the base station and other nodes can only know the location of base station by receiving the packet from source nodes. This assumption is the same with that used in [1-3]. A subset $V_{Static\_Hole} = \{v_{SH1}, \ldots, v_{SHn}\}$ of V are in the state of *dead*. The n^{th} routing path P_{nth} from a source node to the base station can be represented by a subset of the V as $P_{nth} = \{v_{Pn1}, \ldots, v_{Pnm}\}$, which results in that a subset $V_{Dynamic\_Hole} = \{v_{DH1}, \ldots, v_{DHn}\} = P_{1th} + \ldots, + P_{nth}$ of V are in the state of *active but unavailable* and a subset $E_{Hole} = \{e_{H1}, \ldots, e_{Hn}\}$ of E are in the state of *unavailable*. The available nodes and available links can be represented as $V_{available} = V - V_{Dynamic\_Hole} - V_{Static\_Hole}$ and $E_{available} = E - E_{Hole}$.

The first sub-problem of this paper is to find the subset $P_{nth} = \{v_{Pn1}, \ldots, v_{Pnm}\}$ inside the graph $G_{available}$ $(V_{available}, E_{available})$ from one of the source nodes to the base station, which means to find a successful path while bypassing holes. The second sub-problem of this paper is to find the subset $P_{nth\_optimized} = \{v_{OPn1}, \ldots, v_{OPnm}\}$ $(P_{nth\_optimized} \subseteq P_{nth})$ to optimize the found routing path P_{nth} with the least number of nodes $N_{optimized}$ in $P_{nth\_optimized}$. We propose a Two-Phase geographic Greedy Forwarding routing algorithm to solve these two sub-problems in the following section.

4. Algorithm and examples

TPGF consists of two phases: 1) *Geographic forwarding*; 2) *Path optimization*.

4.1. Geographic forwarding

This first phase is responsible for solving the first sub-problem: exploring a delivery guaranteed routing path while bypassing holes in WMSNs. The *geographic forwarding* consists of two methods: *greedy forwarding* and *step back & mark*. The *step back & mark* is used in the situation when *greedy forwarding* cannot find the next-hop node.

4.1.1. Greedy forwarding. The principle for *greedy forwarding* in this paper is: *a forwarding node always chooses the next-hop node which is closest to the based station among all neighbor nodes, the next-hop node can be further to the base station than itself.* This *greedy forwarding* principle is different from the *greedy forwarding* principle in [1-3]: a forwarding node always chooses the 1-hop neighbor node that is closer to the base station than itself. And, the *Local Minimum Problem* does not exist in TPGF. The forwarding decision is based on the comparison among the distance of each neighbor node to the base station.

4.1.2. Step back & mark. There is a worst *block situation* for this new *greedy forwarding* principle. For any node, during the exploration of a routing path, if it has no next-hop node that is available for transmission except its previous-hop node, this node is defined as a *block node*, and this kind of situation is defined as a *block situation*, e.g. Fig.1. To handle the *block situation*, we propose the *step back & mark* approach**:**

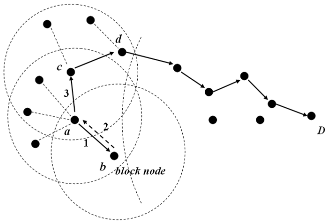

Figure 1. *Block node and block situation*: *b* is a *block node* since it has no 1-hop neighbor that is available to be the next-hop node except node *a*, which is the previous-hop node of *b*. This kind of situation is a *block situation*.

When a sensor node finds that it is a block node, it will step back to its previous-hop node and mark itself as a block node. The previous-hop node will attempt to find another available neighbor node as the next-hop node. Marking the *block node* is to forbid the loop. The *step back & mark* will be repeatedly executed until a node successfully finds a next-hop node which allows the path exploration to change back to *greedy forwarding*.

4.2. Path optimization

This second phase is responsible for solving the second sub-problem: optimizing the found routing path with the least number of nodes. The *path optimization* includes one method: *label based optimization*.

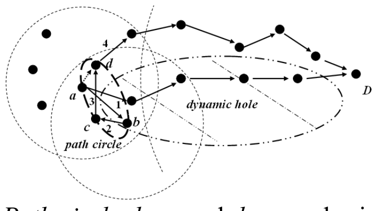

Figure 2. *Path circle*: *b*, *c*, and *d* are nodes in the path, and all of them are neighbor nodes of *a*. The *path circle* is formed by nodes *a*, *b*, *c*, and *d*. Actually, *a* can directly transmit packets to *d* as the dotted line.

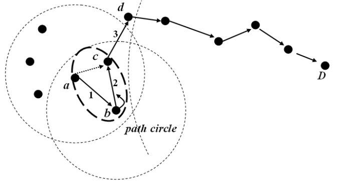

Figure 3. The *path circle* is formed by nodes *a*, *b*, and *c*. The *path circle* is caused by the face routing. Actually, *a* can directly transmit packets to *c* as the dotted line.

4.2.1. Path circle. For any given routing path in a WSN, if two or more than two nodes in the path are neighbor nodes of another node in the path, we consider that there is a *path circle*, e.g., Fig.2. A routing path that found by *geographic forwarding* in

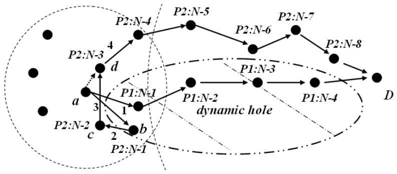

Figure 4. Each node in the routing paths is assigned a label which includes a path number and a degressive node number.

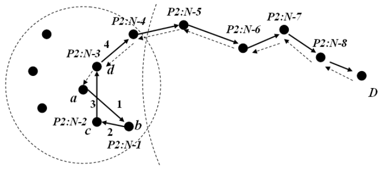

Figure 5. The dash line shows the reverse travelling in the found path. *b* and *c* are not used for transmission, and will be released. The *path circle* is eliminated, since *d* directly sends the acknowledgement to *a*.

TPGF can have *path circle*s, which actually can be eliminated for reducing the number of nodes in the routing path. *Path circle* also appears in the routing path of [1-3] due to the using of face routing, e.g., Fig.3. It is clear that the routing paths that found by TPGF and other algorithms can be optimized to have the least number of nodes by eliminating *path circle*s.

4.2.2. Label based optimization. To eliminate the *path circle*s, we propose the *label based optimization*, which needs to add an additional function in the *geographic forwarding* phase: *whenever a source node starts to explore a new routing path, each chosen node is assigned a label which includes a path number and a degressive node number*, e.g., Fig.4. In TPGF, whenever a routing path reaches the base station, an acknowledgement is sent back to the source node. During the reverse travelling in the found routing path, the *label based optimization* is performed to eliminate the *path circle*s. The principle of the *label based optimization* is: *Any node in a path only relays the acknowledgement to its one-hop neighbor node that has the same path number and the largest node number. A release command is sent to all other nodes in the path that are not used for transmission*, e.g., Fig.5. These released nodes can be reused for exploring additional paths.

4.3. TPGF algorithm

The flowchart of TPGF routing algorithm is shown in Fig.6. The inputs of TPGF are: 1) location of the current forwarding node; 2) location of the base station; 3) locations of 1-hop neighbor nodes. The outputs of TPGF are: 1) location of the next-hop node; 2) or successful acknowledgement; 3) or unsuccessful acknowledgement. It is worth noting that the inputs of TPGF are exactly the same as the inputs of the algorithms in [1-3]. The detailed description of TPGF routing algorithm is as follows: **Phase 1**: *Geographic forwarding* Step 1): The source node checks whether it has usable one-hop neighbor node. If no, the source node produces an unsuccessful acknowledgement and stops transmitting. If yes, then the source node checks whether the base station is in its one-hop neighbor nodes. If yes, then it builds up routing path. If no, then the source node tries to find the next-hop node which is the closet one to the base station among all its neighbor nodes which have not been labeled (occupied). A degressive number-based label is given to the chosen sensor node along with a path number. Step 2): The chosen sensor node checks whether the base station is in its one-hop nodes. If yes, then it builds up routing path. If no, then the chosen sensor node always tries to find the next-hop node which is the closest one to the base station among its all neighbor nodes which have not been labeled (occupied). A degressive number-based label is given to the found next-hop node along with a path number. When this sensor node finds that it has no neighbor node which is available for the next-hop transmission, which means the *block situation* is met, it will step back to its previous-hop node and mark itself as a *block node*. The previous-hop node will attempt to find another available neighbor node as the next-hop node. The *step back & mark* will be repeatedly executed until a sensor node successfully finds a next-hop node which has a routing path to the base station. **Phase 2**: *Path optimization* Step 3): Once the routing path is built up. A successful acknowledgement is sent back from the base station to the source node. Any sensor

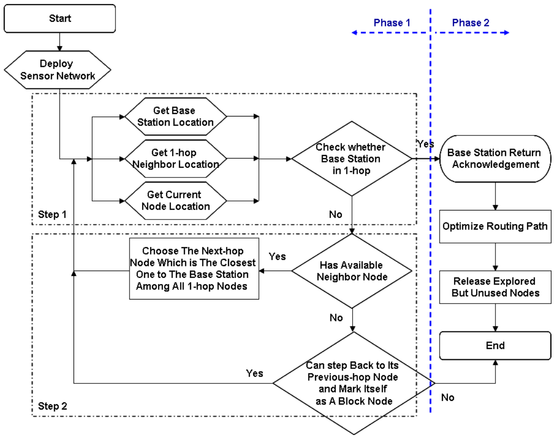
Start
Phase 1
Phase 2
Deploy Sensor Network
Get Base Station Location
Get 1-hop Neighbor Location
Get Current Node Location
Check whether Base Station in 1-hop
Yes
Base Station Return Acknowledgement
Step 1
No
Optimize Routing Path
Choose The Next-hop Node Which is The Closest One to The Base Station Among All 1-hop Nodes
Yes
Has Available Neighbor Node
Release Explored But Unused Nodes
No
End
Yes
Can step Back to Its Previous-hop Node and Mark Itself as A Block Node
No
Step 2

Figure 6. The flowchart of TPGF routing algorithm

node which belongs to this path only relays packets to its one-hop neighbor node which is labeled in Step 2) with the same path number and the largest node number. A release command is sent to all other one-hop neighbor nodes which are labeled in Step 2) but are not used for transmission. After receiving the successful acknowledgement, the source node then starts to send out multimedia streaming data to the successful path with the pre-assigned path number.

The way of finding multiple paths in TPGF is: *repeatedly using the TPGF in the same WMSN with the guarantee that any node will not be used twice*, which is the same with that of [12] by repeatedly using the DPMR algorithm.

5. Simulation and evaluation

The goals of the simulation include: 1) prove that TPGF can find more routing paths than that of GPSR; 2) prove that TPGF can have shorter average path length than that of GPSR. The end-to-end transmission delay is defined as: The average delay of each hop is D_{hop} + $D_{otherfactors}$, the end-to-end transmission delay D_{e2e} is defined as $D_{e2e} = k * (D_{hop} + D_{otherfactors})$, where k is number of hops, D_{hop} is the delay for transmission and $D_{otherfactors}$ stands for the delay contributed by all other factors, such as MAC layer delay and queuing delay. In this paper, we consider the average delay of each hop $D_{hop} + D_{otherfactors}$ as a fixed value.

To evaluate the TPGF routing algorithm, we use a new sensor network simulator NetTopo [13]. The network size in simulation is fixed as 600 × 400 (1 pixel on the canvas is considered as 1 meter). For each fixed number of sensor nodes and transmission radius, the average number of paths and the average path length are computed from 100 simulation results using 100 random seeds for network deployment. Then, we change the node number (from 100 to 1000) and transmission radius (from 60 to 105) to obtain different values. Figs 7, 8, 9 are the simulation results on the average number of paths that found by applying TPGF and GPSR respectively. By comparing the average number of paths in three figures, we can easily see that TPGF can find much more number of paths than that of GPSR on both GG and RNG planar graphs. Fig.10 is the simulation results on the average path length of TPGF before applying optimization and Fig.11 is the simulation results on the average path length of TPGF after applying optimization. It is easy to conclude that after optimization the average path length of TPGF is much shorter. Figs 12 and 13 are the simulation results of GPSR on the average path length on both GG and RNG planar graphs. Comparing Figs 11, 12, and 13, it is proved that TPGF can have shorter average path length than that of GPSR.

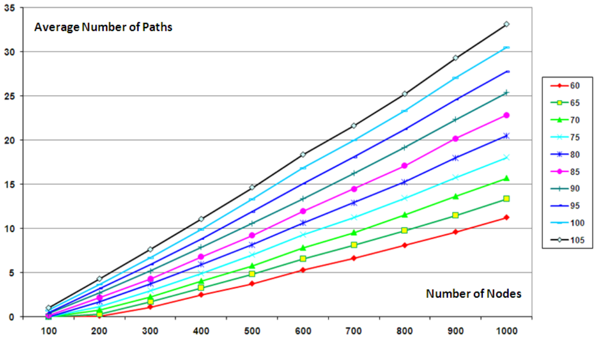

Figure 7. TPGF: average number of paths vs. number of nodes

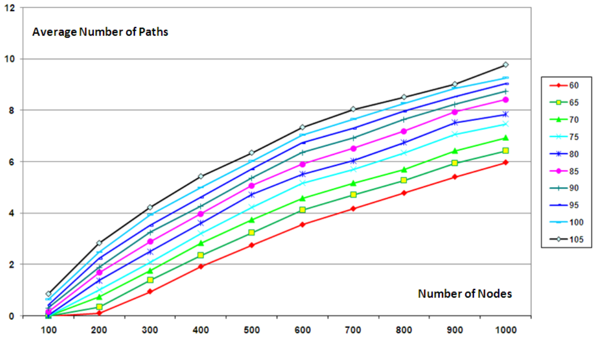

Figure 8. GPSR on GG planar graph: average number of paths vs. number of nodes

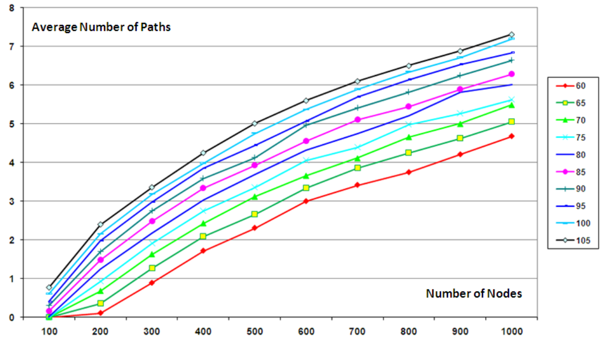

Figure 9. GPSR on RNG planar graph: average number of paths vs. number of nodes

6. Conclusion

Efficiently transmitting multimedia streaming data in WSNs is a basic requirement. In this paper, a new Two-Phase geographic Greedy Forwarding (TPGF) routing algorithm is proposed to facilitate the multimedia streaming data transmission in WMSNs. TPGF does not adopt face routing to bypass holes,

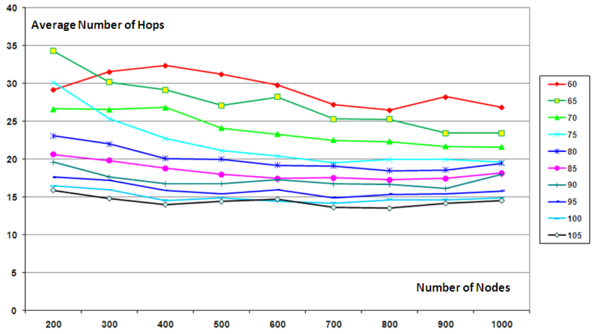

Figure 10. TPGF: average number of hops before optimization vs. number of nodes

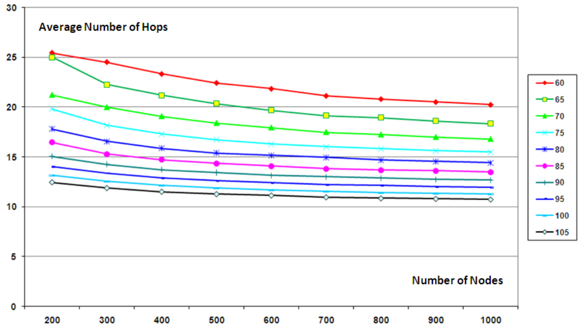

Figure 11. TPGF: average number of hops after optimization vs. number of nodes

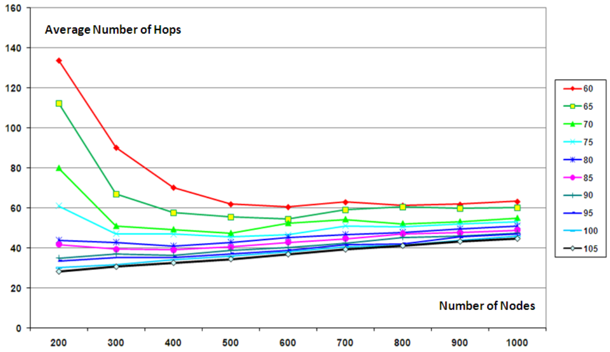

Figure 12. GPSR on GG planar graph: average number of hops vs. number of nodes

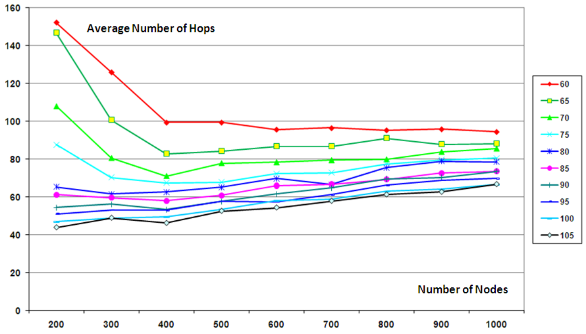

Figure 13. GPSR on RNG planar graph: average number of hops vs. number of nodes

which makes TPGF be different from many existing geographic routing algorithms. Simulation comparison in this paper shows that TPGF is more suitable for transmitting multimedia streaming data than other geographic routing algorithms in geographic WMSNs.

7. References

[1]. B. Karp, H.T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, in Proc. of the Annual International Conference on Mobile Computing and Networking, Boston, August, 2000.

[2]. F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, “Geometric Ad-Hoc Routing: Of theory and practice”, in Proc. of the 22nd ACM International Symposium on the Principles of Distributed Computing, 2003.

[3]. B. Leong, S. Mitra, B. Liskov, “Path Vector Face Routing: Geographic Routing with Local Face Information”, in Proc. of the 13th IEEE International Conference on Network Protocols, 2005.

[4]. K. Seada, A. Helmy, R. Govindan, “Modeling and Analyzing the Correctness of Geographic Face Routing Under Realistic Conditions. Ad Hoc Networks”, doi:10.1016/j.adhoc.2007.02.008, pp. 855-871, 2007.

[5]. Q. Fang, J. Gao, L.J. Guibas, “Locating and bypassing routing holes in sensor networks”, in Proceedings of the 23rd Conference of the IEEE Communications Society (INFOCOM), Hong Kong, China, March 2004.

[6]. W. Jia, T. Wang, G. Wang, M. Guo, “Hole Avoiding in Advance Routing in Wireless Sensor Networks”, in Proceedings of the IEEE Wireless Communication & Networking Conference (WCNC). USA, March, 2007.

[7]. F. Yu, E. Lee, Y. Choi, S. Park, D. Lee, Y. Tian, S. Kim, “A Modeling for Hole Problem in Wireless Sensor Networks”, in Proc. of the International Wireless Communications and Mobile Computing Conference, Hawaii, USA, August, 2007.

[8]. Multipath routing in wireless ad hoc & sensor networks, http://snac.eas.asu.edu/snac/multipath/multipath.html, the latest access on March 13, 2008.

[9]. D.B. Johnson, D.A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, in Proc. of Mobile Computing, Imielinski, Korth, Eds. 1996.

[10]. C. Perkins, “Ad Hoc On-Demand Distance Vector (AODV) Routing”. RFC 3561, 2003.

[11]. K. Zeng, K. Ren, W. Lou, “Geographic On-Demand Disjoint Multipath Routing in Wireless Ad Hoc Networks”, in Proc. of Military Communications Conference, Atlantic city, USA, October 17-20, 2005.

[12]. S. Li, Z. Wu, “Node-Disjoint Parallel Multi-Path Routing in Wireless Sensor Networks”, in Proc. Of the Second International Conference on Embedded Software and Systems, China, December 16-18, 2005.

[13]. L. Shu, C. Wu, M. Hauswirth, “NetTopo: Beyond Simulator and Visualizer for Wireless Sensor Networks”, Technical Report of Digital Enterprise Research Institute, July, 2008.