ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

INTELLIGENT COMMUNICATION WITH GEOGRAPHIC ROUTING BY OVERCOMING UNREACHABILITY PROBLEM IN WIRELESS SENSOR NETWORK

[1]M.Berlin Mary,[2]T.Samraj Lawrence,[3]D.C.Joy Winnie Wise

Department of Cse(With spl.in network),Francis Xavier Engineering College,Tirunelveli,Tamilnadu,India

***Abstract*— To bypass voids in sensor networks, most existing geographic routing protocols tend to route packets along the boundary nodes. Generally, a packet will be either forwarded along a void boundary by the right-hand rule or pushed back to find another route when it encounters the void. The two techniques consume more energy of boundary nodes, drop many packets and may incur data collisions if multiple communication sessions share the same boundary nodes. We propose partial and full link reversal algorithms to bypass voids during geographic routing over duty-cycled wireless sensor networks. We propose a distributed approach that is oblivious to one-hop neighbor information. Upon termination of the algorithm, the resulting network is guaranteed to be destination-oriented. Further, to reduce the delays incurred under reactive link reversal, we propose the use of 'pseudo-events', a preemptive link reversal strategy, that renders the network destination-oriented before the onset of a real event. A simulation study of the effectiveness of pseudo-events is also provided. We propose in this paper an alternative and efficient void avoidance scheme. The proposed on-demand scheme consists of void discovery, void announce and packet rerouting steps. After discovering a void, a sender node inside the void's announce-area reroutes all data packets to get around the void in advance by selecting one appropriate forwarding side. The double objective of our scheme is to prevent data packets from traveling along the boundaries of voids and to avoid them the concave zones of voids. By achieving this objective we can reduce the energy consumption of boundary nodes and data collisions in these nodes. We can also reduce the packets rerouting overhead and the number of packets dropped by nodes on the boundaries of voids. Simulation showed the efficiency of our scheme.**

***Index Terms*—partial link, full link, pseudo events, sensor network, geographic routing, distributed algorithm, voids, Real-time routing, Sensor networks, avoid voids.**

I. INTRODUCTION

The presence of voids in wireless sensor networks imposes difficulties in routing data packets. Voids can be formed either due to sensor deployment or because of failure of sensor nodes; they can also exist due to local minimum phenomenon often faced in geographic greedy-forwarding strategy. The localized operation and the stateless feature of geographic routing make it simple and scalable. Geographic routing also enables a recasting service, which supports the delivery of packets to all nodes in a specified geographic region. In this strategy, a packet is forwarded to a one-hop neighbor who is closer to the destination than the current node. This process is repeated until the packet reaches the destination node, or the packet is stuck at a node whose one-hop neighbors are all farther away from the destination. The node where a packet may get stuck is called a *local minimum* or s*tuck node* (the node __in Figure 1). Existing void-handling techniques can be classified into two catteries: *right-hand rule* and *backpressure rule.* According to the right-hand rule, packets tend to be routed along the boundaries of voids. The probability that boundary nodes of a void are shared by several communication sessions is very high than other nodes. Thus excessive energy consumption and data collisions may occur in these nodes. According to the back pressure rule, packets tend to be pushed back to upstream node and attempt to find another route to destination. For real-time protocols, backpressure rule violates the desired packet positive progression toward its destination and many packets are dropped because routes are much longer. To contribute on resolving the above-mentioned void problems.

Particularly Geographic routing is composed of two basic elements: a positioning service and a geographic based routing algorithm. The main task of positioning services is determining the location of the destination, in response to a request from the source node. Then position obtained added to the packet header by the source node. Operation of Geographic forwarding algorithm runs in two modes: greedy mode and recovery mode. In greedy mode data packet is forwarded to one of the neighboring nodes that provide the greatest positive progress towards the destination. This selection process is performing

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

by considering geographic location of current node, neighboring nodes and the destination node. Each node is aware of its own position using GPS receiver. Also, the positions of the other nodes are obtained through exchange periodic beacon packages. Thus, intermediate nodes can easily route packets according to geographic location of its own, neighboring nodes and the destination node. When the packet encountered a void, recovery mode is invoked. In this situation, the packet gets stuck at a node that called "void node". In recovery mode, the void node tries to forward the stuck packet surrounding the void since it is probability that there is a topologically valid route from the source to the destination node. The basic difference between various recovery techniques is their criterion to select the next hop. Without using an appropriate technique when dealing with voids, some of the packets in the network are likely to be discarded, and furthermore, much of the network resources are wasted. Hence, one of the major challenges for geographic routing protocols is the existence of communications voids problem and effectively and efficiently handling this problem. The most plain void-handling technique is flooding in which stuck packet is broadcasted to all neighbors by void node and every node that receives stuck packet at the first time. Through this technique, the packet will certainly reach its destination. So this technique has a good effectiveness, but on the other hand, this is not efficient method. Because large amounts of network resources, including network traffic capacity allocated to a single packet. In addition, may be a large number of unnecessary duplicate packets received by the destination node. When the stuck packet reaches at a node that is closer to the destination than the void node, routing mode returns to greedy forwarding. Note that should be in the attention is that a void-handling technique is invoked only when encountering a packet with voids.

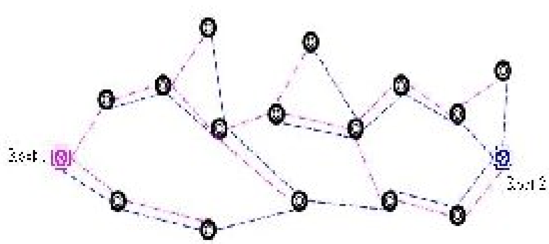


Fig 1.1 Geographic Routing

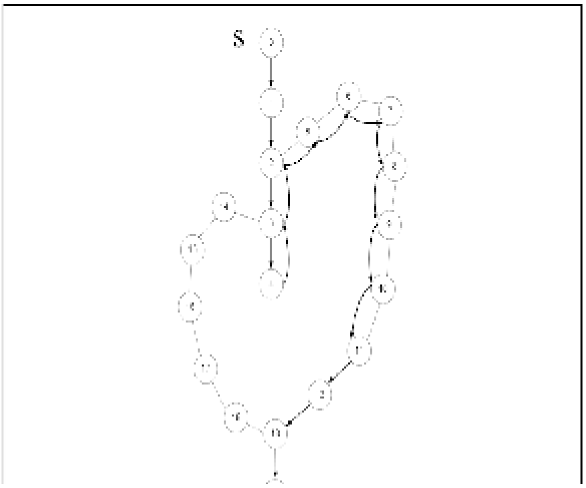
we propose in this paper an alternative and efficient real-time routing for sensor networks. The proposed on-demand void voidance scheme consists of three steps: *void discovery step*, *void announce step* and *packet rerouting step*. Based on the right-hand rule, the void discovery step detects all boundary nodes and then calculates the center _ of the void shown in Figure 4. during the void announce step, all nodes inside an announce area (the zone AA in Figure 5) will receive coordinates of for use during the packet rerouting step. After discovering a void, a sender node inside the announce area AA reroutes all packets to get around the void in advance by choosing a unique and appropriate forwarding side**.** Our scheme has two principal goals: one is to prevent data packets from traveling along the boundaries of voids; the other is to avoid them the concave zones of voids. By achieving the first goal we can not only reduce the energy consumption of boundary nodes of voids, but also reduce the data collisions in these nodes. The second goal can reduce the packets rerouting overhead and the number of packets dropped by nodes on the boundaries of voids. The rest of this paper is organized as follows. We summarizes the exiting real-time routing protocols that handle voids in sensor networks.

II. RELATED WORKS

Some existing real-time routing protocols for sensor networks use specific rerouting mechanisms to avoid drops to packets by stuck nodes on boundaries of voids. To provide service differentiation in the timeliness domain, the real-time routing protocol RAP uses velocity-monotonic classification of data packets. It uses the right-hand rule to route packets around the perimeter of void zones. SPEED [6] is considered as the first efficient real-time routing protocol for sensor networks in the literature. It provides soft end-to-end deadline guarantees for real-time packets. It is based on the backpressure rule to handle voids in sensor networks. It has been admitted in SPEED that its void avoidance scheme is not guaranteed to find a path if there is one as in GPSR. The protocol FT-SPEED [8] borrows the idea of SPEED to handle the real time packet delivery in greedy forwarding. It has the same component with SPEED except the void avoidance scheme. FT-SPEED proposes an alternative scheme to handle the problem of voids. To reach their respective destinations, data packets which meet a void are delivered in FT-SPEED by using only nodes on the boundaries of voids. Consequently, these nodes will consume much energy and the void area will be able to increase. The protocol RPAR integrates power control and realtime routing for supporting an energy-efficient communication. RPAR uses face routing mechanisms to route packets around large voids in the network.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015



The small-world model corresponds to a phenomenon in a social network where any two people have “six degrees of separation,” which means that any two people can be connected through six intermediate acquaintances. More recently, it has been shown in that this phenomenon is pervasive in many natural and artificial complex networks, and is captured by two measurements: small average path length and high clustering coefficient (defined as the average fraction of pairs of neighbors of a node that are also neighbors of each other). Kleinberg defined an infinite family of random network models that seek a simple framework that encapsulates the paradigm of Watts and Strogatz—rich in local connections with a few long-range connections. Rather than using a ring as the basic structure, this model uses a 2D n grid and allows each node to have a directional long link to a remote contact with the distance in the r-exponent power-law distribution. It is proved in that there is a unique “navigable” model ðr ¼ 2Þ within the family for which decentralized algorithms are bounded by nÞ. An extension to the navigable hierarchical network is discussed. Terminode is based on the small-world model that does not always forward packets directly toward the destination. In order to optimize routing in case of voids in the network topology, a node finds a list of remote contacts distributed all over the network to which it maintains a good path. To find a route to the destination,a node asks its remote contacts which in turn ask their remote contacts, and so on. The right remote contacts found are added as a loose source path to the header of the data packets. Though Terminode finds short paths, it uses some sort of broadcast to discover routes.

Many routing algorithms have been proposed for terrestrial ad hoc and wireless sensor networks in recent years. Routing algorithm tailored for UWSNs is however in its infancy. Xie,Cui and Lao [1] propose Vector-Based Forwarding (VBF) algorithm. Every node is assumed to be aware of its location. The directional information from source to sink, represented by a vector, is stored in the packets. When a node receives a packet, it calculates its distance to the vector of this packet. If the distance is smaller than a pre- defined constant threshold, referred to as radius, the node broadcasts this packet, and discards it otherwise. As we have pointed out earlier, there is no adaptive scheme for setting broadcasting radius in VBF. Nikolaou *et al.* [2] propose a Hop-by-Hop Vector-Based Forwarding (HH-VBF) scheme in which a routing vector is used for each individual forwarder. Different from both Band HH-VBF, we have designed an adaptive scheme for setting the threshold in our proposed REBAR. Meanwhile, we extend REBAR to cope with potential routing voids in the network. Lee *et al.* [3] develop underwater sensor diffusion protocol similar to Directed Diffusion [4]. The sink and source establish shortest path through flooding. Community-based forwarding and unicast probe flows are exploited to cope with node mobility. Search and Tan [5] propose multi-path virtual sink architecture to deal with the adverse underwater link condition. Packet duplicates are delivered to multi-sink through multipath in order to improve the reliability. Peng, Winston and Pius propose packet cloning in [6] to overcome the high transmission loss and provide reliability. MaMagistretti *et al.*[7] propose a Delay Tolerant Dolphin approach, which exploits mobile nodes to collect data from other static sensors. This architecture is similar to the Data Mules proposed in [8], which can save energy because all the data transmissions happen between static nodes and their one-hop mobile neighbors.

III. PROBLEM DESCRIPTION

A concave node is a node that has no neighbors closer to the destination other than itself [9]. The term ’closer’ is somewhat fuzzy, as different greedy algorithms have different closeness criteria. Since our ideas can be used with different greedy routing algorithms, we define a concave node as anode that has no neighbor that can make a greedy progress towards the destination (for the routing algorithm in use).Since position-based routing uses local information for forwarding decisions, a concave node can not be predicted inadvance, based on the position of its neighbor nodes. Usingthe 2-neighborhood information can indeed improve decisions made during the algorithm, but cannot avoid reaching concave nodes. Assuming one uses a recovery algorithm that switches back to greedy mode once recovered from the concave situation, the number of back-tracking packet transmissions required to switch back to greedy mode can vary between just a few hops to a very long retreat. Figure 1 shows an example path to a concave node that is reached only after numerous hops, only at this point the recovery process shown in the figure) begins. In addition, [16] reviews additional defeciencies of perimeter-based recovery algorithms: network disconnection due to graph planarization, nodes mobility causing routing loops, and

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

routing in the wrong direction causing error due to mobility or simply increasing the number of hops. We thus extend the definition of concavity. A concavity in the wide sense means that none of the node's neighbors towards the destination can eventually lead to the destination. A first degree concavity is the same as the general definition for concavity, a node that does not have a greedy next hop to the destination. A concavity of the nth degree, is a concavity where the smallest concavity degree of all neighbors is n-1. By identifying regions of concave nodes in the wide sense, our greedy algorithm can stop short of entering them, and thus avoid long retreats.

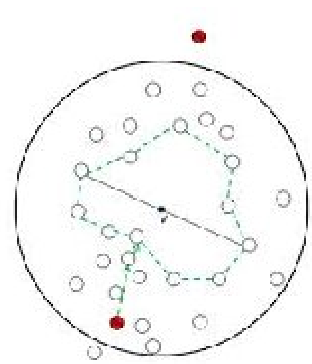


Fig 3.1 Routing Void Problem In Nodes

To contribute in resolution of problems encountered by real-time routing protocols with presence of concave voids in sensor networks, we propose an efficient void avoidance scheme. We assume in this work that each node can get the location information of itself and of its one-hop neighbors by a periodic beaconing [6]. Also, we assume that a sender node marks packets with their destination's location. Thus, a location registration and lookup service that maps node ID to location [9] is required. To prevent packets from concave areas of voids and then dropping by boundary nodes in sensor networks, we propose a new idea which consists in starting a void avoidance process with *n*-hops before packets reach boundary nodes of a discovered void. The avoidance of the void is done in only one direction by using either clockwise or anticlockwise forwarding. The proposed scheme consists of three steps: *void identification, void declaration* and *packet rerouting.*

IV. VOID IDENTIFICATION STEP

We formulate a plan and we define the absolute related to the origin of a network area (Figure 1). The void discovery step is made on-demand; it is started by the first data packet for a destination falling into a local minimum at a stuck node, as shown in Figure4.1. This step consists of three phases: First, middle and last phase.

A. The First Phase

We suggest a way to virtually reposition nodes in the network, so that greedy routing decisions can be wisely taken and recovery process can be significantly improved or avoided altogether. Node repositioning has several goals. The firstone is to identify and mark concave nodes. Identifying a concave node is simple, as every node can do so locally by analyzing its connectivity. When a first data packet fall into a local minimum (case of node in Figure4.1), node broadcasts a void-beacon packet to its one-hop neighbors notifying them that it can't forward any more. The received packet is then dropped, as in SPEED [6], in order to respect the required speed constraint of positive progression towards the destination. Then, the stuck node _creates a VD (void discovery) packet marked by its ID, whose mission is to collect the locations of all boundary nodes by using the right-hand rule, which is held as follows. The stuck node builds, from its (the set of all neighbor nodes), its *d*(the set of neighbor nodes inside the right-hand area of the vector *bi*)and__*d* (the set of neighbor nodes inside the left-hand area of *bdi*). Each neighbor node of an angle, as shown in Figure 4.1]0, _[for nodes inside forms with the vector *bi* and 2 for nodes inside .By using the vector product, we have

$(b_i b_k) = b_i b_k$ (1)

By taking the magnitude on both sides of (1), we obtain:

$b_i b_k = b_i . b_k$ (2)

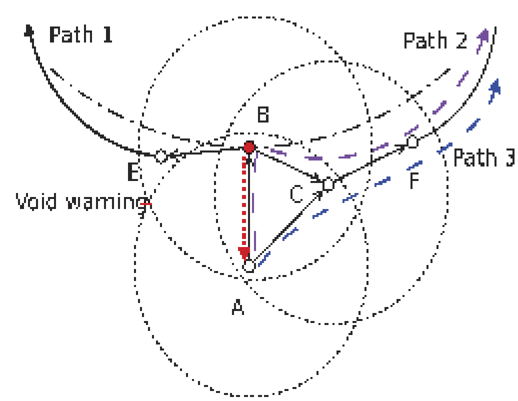


FIG 4.1 VOID IDENTIFICATION PROCESS

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

From (3), we deduce:

$$x_1 \cdot y_2 > x_2 \cdot y_1 \Leftrightarrow \sin\theta > 0 \Leftrightarrow n_k \in RN_i$$

$$x_1 \cdot y_2 \le x_2 \cdot y_1 \Leftrightarrow \sin\theta \le 0 \Leftrightarrow n_k \in LN_i$$

Finally, the sets RN_i and LN_i of a node b_i are given by:

$$RN_i = \{ n_k \,/\, n_k \in NS_i \ \text{ and } \ x_1 \cdot y_2 > x_2 \cdot y_1 \}$$

$$LN_i = \{ n_k \,/\, n_k \in NS_i \ \text{ and } \ x_1 \cdot y_2 \le x_2 \cdot y_1 \}$$

B. MIDDLE PHASE

A second goal of repositioning is to improve the greedy routing. Our greedy algorithm avoids using the floating nodes and thus does not get stuck in a concave area. This way we can avoid switching to recovery mode in many cases. This algorithm is distributed, and local, and is executed periodically at low cost - due to its local nature. We also execute avoid identification algorithm which is performed around the void. This algorithm is distributed as well, but it is executed by all nodes at the void edge, and possibly their neighbors.

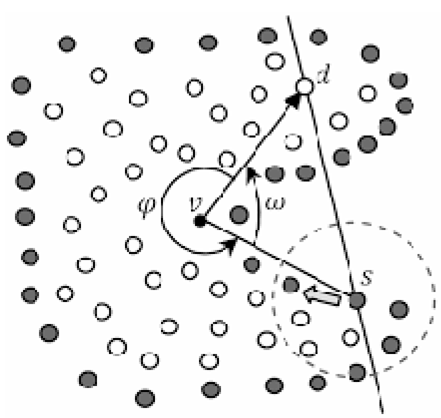


FIG 4.2 Nodes Receiving Packet

When receiving the VD packet (Figure 4.2), the node builds its and (the set of neighbor nodes inside the right-hand area of the vector *bi+1* the set of neighbor nodes inside the left-hand area of the vector *b* phase described above. If, in the same manner as in the initial *bi* is not empty, the node sends the VD packet to its neighbor node having the smallest angle) among all angles of nodes in Formally

$$Max\{\cos\varphi \,/\, n_k \in RN_{i+1}\}$$

$$\cos\varphi = \frac{\overrightarrow{b_{i+1}n_k} \cdot \overrightarrow{b_{i-1}b_i}}{|\overrightarrow{b_{i-1}n_k}| \cdot |\overrightarrow{b_{i+1}b_i}|} \quad (5)$$

$$\cos\varphi = \frac{x_1 \cdot x_2 + y_1 \cdot y_2}{\sqrt{(x_1^2 + y_1^2)(x_2^2 + y_2^2)}} \quad (6)$$

If it is empty, the node consults its not empty node and sends the VD packet to its neighbor Node, having the largest angle among all angles of nodes. Formally,

$$Min\{\cos\varphi \,/\, n_k \in LN_{i+1}\}$$

LN are both empty, the node ki+1 destroys the received VD packet and the void discovery process will stop (i.e., an absence of a void).

C. The Last Phase

Each intermediate node will repeat the intermediate phase described above until the VD packet has traveled around the void and eventually been received by the initiator node (Figure 4.3).At the end of the final phase, node extracts from the received VD packet geographical positions of all boundary nodes. Node _selects two nodes and from the set of boundary nodes so that the distance between is the longest distance among the distances between any two nodes. Then, the node estimates the void's center represented by the midpoint of the segment b.

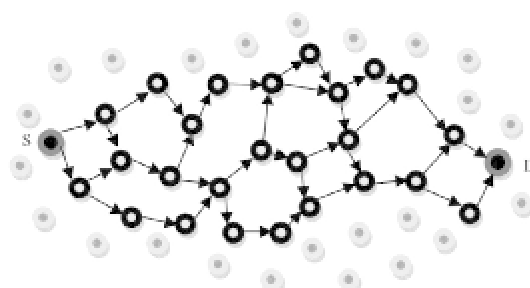


Fig4.3 Correct Estimation Of Void

V. VOID DECLARATION STEP

This section discusses the weight assignment for each of the two versions of RGF protocol *Version 1.* In the first version, as discussed in the previous section, we consider all

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

neighbors of node *A* to have equal probability to be chosen as the next hop. Assuming node *A* has *n*neighbors:

$w(1) = w(2) = ... = w(n)$

and following the formula for probability given above:

$P(1) = P(2) = ... = P(n) = 1/n$

Version 2. In the second version, we assign weights to each node relative to its distance from the destination. In this proposal, we suggest giving higher weights and thus higher probabilities to neighbors that are closer to the destination. This could be justified in that we detour less from the destination. Giving weights relative to the distance from the destination implies that node *A,*on switching to RFG, must perform a small calculation to assign the weights to its current neighbors. In other words, once node *A* determines that it must overcome a void by employing RFG, it will have to calculate the distance from the destination for each and every node of its neighbors. The coordinates of the destination exists in the packet that node *A* has received. Also, node *A* has already received the coordinates of all its neighbors as this information is updated in node *A'*s neighbor table upon reception of a Hello message. Assuming a total of *n* neighbors for node *A* and defining to be the distance of neighbor *i* from the destination, the weight for neighbor *i* would be as follows:

$w(i) = 1/d$

where $1 = i = n$

and following the formula for calculating the probability proposed in the previous section, we

would have:

$P(i) = (1/d_i) / \sum_{j=1}^{n} 1/d_j$ where $1 \le i \le n$

where $1 = i = n$

Once node *A* calculates the probabilities for each of its neighbors, it will perform a weighted random selection to choose one neighbor as the next hop so that it can recover from the void situation.

VI. REROUTING DATA PACKET

The packet rerouting step consists on selecting forwarding candidate nodes by a sender node _ inside the announce area showed in Figure 6.1. The purpose of this step is to choose forwarding nodes which avoid to data packets both direct contact with boundary nodes and concave areas of a void. The selection of forwarding candidate nodes by a sender node,

which is located at *n*-hops far from borders of a void (Figure 6.1), isdone after having determined coordinates of the point,

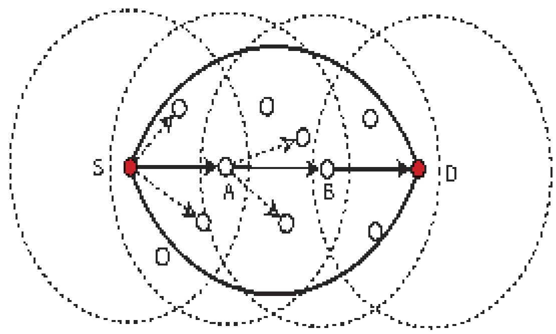


FIG 6.1 DATA REDIRECTED TO ANOTHER PATH ACROSS NODES

the

1. If the destination d is on the right of $\vec{sv}$ (i.e., $sin(\omega) > 0$ in Figure 6.1), node s builds its $FS_s(d)$ from its RN_i (nodes belonging to the hatched zone in Figure 8).
2. If the destination d is on the left of $\vec{sv}$ (i.e., $sin(\omega) \le 0$ in Figure 6.1), node s builds its $FS_s(d)$ from its LN_i.
3. If the packet arrives at one-hop near boundary nodes (Figure 7), the node s chooses from its $FS_s(d)$ the further node from the center v of the void as next-hop.
4. If $FS_s(d)$ of the node s is empty, a new void discovery process is started by the node s.

It's clear that if the void happens at the source node, this node will be selected as trigger node and so source node will set forwarding mode as void handling without any other choice.

SIMULATION RESULTS

We run the simulation on our custom NS2 simulator. Our simulations are done in both static connected networks and networks with the random waypoint mobility. In each simulation with a static network, a connected graph with N(ranging from 150 to 450) nodes is randomly generated in a1,000 _1,000 square field. One hundred random networks a regenerated for each of the 13 network densities. We generate connected networks by randomly spraying nodes and discard

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

the networks that are not connected until a connected one is generated. After that, we let the simulator run for a period of time which is sufficient for the nodes to grow the VLLs. We selected this period of time to be 60 seconds. Then, we select 1,000 source-destination pairs by randomly selecting nodes in the network. For each pair of nodes, we generate message for each protocol. That is, for each routing protocol,1,300,000 different messages are created in our experiment. The static network density in our experiment ranges between two extremes. In the sparse extreme, the shortest path is usually

much longer than the Euclidean distance between the source and the destination, and the topology is more like a tree structure or a linear structure than a mesh. This density is critical for performance evaluation of routing algorithms, where finding a good path at a low cost becomes nontrivial task and a real challenge for position-based routing. In the dense region, the traditional greedy routing has a high success probability, and therefore, all algorithms have similar performance since they all degrade to greedy routing. The simulation scenario is in a squared area with stationary sensor nodes placed uniformly. A void is created at the center of the area;i.e. an area is set with no sensor nodes inside to simulate a void unrealistic environments. The sink is at the right side of the void and the source node at the left. To create several network loads, the source rate is gradually increased until reaching 100 pkts/sec.

Bandwidth	200 Kbps
Payload	32 bytes
Terrain	200m × 200m
Number of nodes	100
Node placement	Uniform
Radio Range	40 m
MAC Layer	802.11
Radio Layer	RADIO-NONOISE
Propagation model	TWO-RAY
Radius of the void	105 m
Time of simulation	200 sec

FIG 6.4 Simulation Settings for NS2

As shown in Figure 6.5, the average end-to-end delay of each protocol increases when the rate is increased, but SPEED is always better in performances than SPEED. The rerouting of packets in advance used in SPEED force each packet to converge quickly towards the destination node. Compared to SPEED in Figure 6.5,the protocol SPEED provides a better average packet delivery ratio, especially when the rate is greater than 20 pkts/sec. The number of packets dropped by the boundary nodes is reduced by SPEED. Figure 11 shows that SPEED outperforms SPEED in term of routing path length. Thanks to our proposed void avoidance scheme, SPEED reduces the routing distance with presence of voids in a sensor network. Even with varying the rate between 1 and 100 pkts/sec, the average path length of each protocol remains stable between 17 and 18 hops.

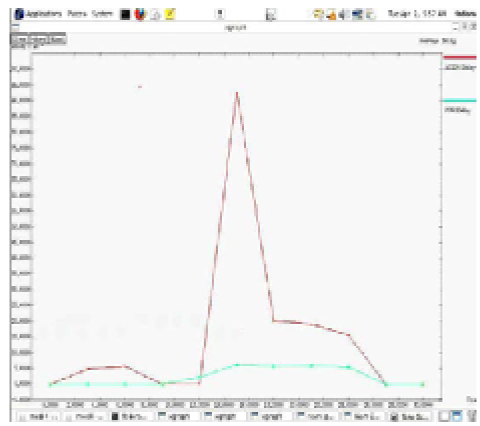


FIG 6.5 Simulation Result

To reduce the above-mentioned insufficiencies of existing void handling schemes in sensor networks, we proposed an efficient void avoidance scheme for real-time routing protocols under geographic routing application. A sender node, located at *n*-hops far from a discovered void, reroutes all data packets to get around the void in advance by choosing a unique and appropriate forwarding side according to the positions of the sender node, the destination node and le center of the void. The proposed scheme prevents data packets from traveling along the boundaries of voids and avoids them the concave zones of voids. Associated to the old protocol, our void avoidance scheme ensured an efficient routing with presence of concaves voids in wireless sensor networks. Compared by simulation, the obtained protocol provided good performance in terms of average end-to-end delay, packet delivery ratio and routing path length. Our future work is to conduct several simulations of the proposed scheme associated to various real-time routing protocols with varying radius of voids in a sensor network k. We are also working on a non-deterministic metric to use by a sender node in order to select its own forwarding side around a discovered void.

References

[1] Ahmed,S.KanhereandS.Jha,"TheHolesProbleminWireless Sensor Networks: A survey," ACM Sigmobile Mobile

Computing and Communications Review, vol. 9, Issue 2, pp.4-18, Apr., 2005.

[2] D. Chen, P. Varshney, "A survey of void handling techniques for geographic routing in wireless networks," IEEE Communications Surveys and Tutorials, pp. 50-67, 2007.

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

[4] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," MobiCom'00, Aug 2000.

[5] C. Lu, B.M. Blum, T.F. Abdelzaher, J.A. Stankovic, and T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02), September 2002.

[6] T. He, J.A. Stankovic, C. Lu and T.F. Abdelzahe r,"ASpatiotemporal Communication Protocol for Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems 16(10): 995-1006, 2005.

[7] .Jia,T.Wang,G.WangandM.Guo,"HoleAvoidinginAdvance Routing in Wireless Sensor Networks," IEEE Wireless Communications and Networking Conference (WCNC), pp. 3519-3523, 2007.

[8] L. Zhao, B. Kan, Y. Xu and X. Li, "FT-SPEED: A Fault-Tolerant, Real-Time Routing Protocol for Wireless Sensor Networks," International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), pp. 2531-2534, 2007.

[9] 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," IEEE International Workshop on Quality of Service, June 2006.

[10] J. Broch et al., "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols," Proc. 4th ACM/IEEE Int'l. Conf. Mobile Computing and Networking MOBICOM '98, Dallas, TX, USA, 1998, pp. 85–97.

[11] E. Kaplan, Understanding GPS, Artech House, 1996.

[12] S. Capkun, M. Hamdi, and J. Hubaux, "Gps-free Positioning in Mobile Ad Hoc Networks," Proc. Hawaii Int'l. Conf. System Sciences, Jan. 2001.

[13] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, Aug. 2001, pp. 57–66.

[14] Ko, Y., and Vaiyda, N. Location-aided routing in mobile ad hoe networks. In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98), Dallas, Texas, USA, Aug. 1998.

[15] Finn. Routing and Addressing Problems in Large Metropolitan-scale Internetworks. Technical Report ISI/RR-87-180, USC/ISI, March 1987.

[16] A. Nguyen, N. Milosavljevic, Q. Fang, J. Gao, and L. J. Guibas, "Land-mark selection and greedy landmark-descent routing for sensor networks,"in *Proc. IEEE INFOCOM*, 2007, pp. 661–669.

[17] T. Park and K. G. Shin, "Soft tamper-proofing via program integrity verification in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 4, pp. 297–309, 2005.

[18] P. N. Pathirana, N. Bulusu, A. V. Savkin, and S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Trans. Mobile Comput.*, vol. 4, pp. 285–296, 2005.

[19] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. IEEE/ACM MobiCom*, 2003, pp. 96–108.

[20] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A geographic hash table for data-centric storage," in *Proc. ACM WSNA*, 2002, pp. 78–87.

[21] H. Sabbineni and K. Chakrabarty, "Location-aided flooding: An energy-efficient data dissemination protocol for wireless sensor networks,"*IEEE Trans. Comput.*, vol. 54, pp. 36–46, 2005.

[22] K. Seada, A. Helmy, and R. Govindan, "On the effect of localization errors on geographic face routing in sensor networks," in *Proc.IEEE/ACM IPSN*, 2004, pp. 71–80.

[23] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. IEEE/ACM MobiHoc*, 2003, pp.201–212.

[24] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 13, pp.384–397, 2005.

[25] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Netw.*, vol. 8,pp. 153–167, 2002.