An Effective Mechanism for Handling Open Voids in Wireless Sensor Networks

Mohamed Aissani, Sofiane Bouznad, Abdelmalek Hariza, and Salah-Eddine Allia
Laboratory of Research in Artificial Intelligence, Polytechnic Military School (EMP)
P.O. Box 17, Bordj-El-Bahri 16111, Algiers, Algeria
{maissani, bouznad.sofiane}@gmail.com, {malik-abd, s.alia}@hotmail.com

***Abstract*—Open voids are often formed on the boundary of a deployed wireless sensor network (WSN). Geographical routing protocols must handle these voids where packets fall into local minima. To contribute on resolving this problem, we propose in this paper an effective mechanism for handling this kind of voids. It uses two simple and effective algorithms ensuring discovery and maintenance of the network boundary. Contrary to existing void-handling techniques, our proposal uses the information about this boundary and the destination node for better directing data packets in optimal paths. Thus, open voids are avoided with great efficiency. The proposed mechanism has good performances in terms of packet delivery ratio, average routing path length, boundary energy consumed per delivered packet and average residual deadline of all delivered packets.**

Keywords*—Sensor networks; geographical routing; open voids; void-handling techniques.

I. INTRODUCTION

The mission of a WSN is generally to supervise a phenomenon, to take measures regularly and to send alarms to a sink node. Many applications using WSN exist in different fields such as defense, safety, health, agriculture and smart houses. Due to several economic and deployment considerations, sensor nodes have small size with limited resources of storage and computation. They use batteries, thus energy conservation becomes a big challenge.

Since they communicate by radio with short range, the multi-hop routing becomes necessary so that captured information reaches the sink node. A simple approach would be to use the geographical routing, which guarantees a good scalability and a positive progression of forwarded packets towards the sink node. Each sensor node forwards the current data packet to its neighbor, which is nearest that itself to the sink node. The fact that no routing information is to maintain in a network, other than tables of neighbors, routing paths of data packets adapt to any topological change.

Nevertheless, the geographical routing has two problems. Firstly, it is not applicable when sensor nodes do not have the possibility of knowing their geographical locations. Virtual coordinates systems, such as NoGeo [1], GEM [2], and BVR [3], can be used in this case. These coordinates require nodes to know the distances from its neighbors to certain points of reference by using periodic messages. Secondly, there can be voids between a source node and a sink. A void is an area without any active node. It can be located inside the network (closed void) or on the network boundary (open void). A geographical routing path towards a sink is interrupting when relay nodes for avoiding voids are absent. Existing solutions present insufficiencies in handling open voids [4-17], so we propose in this paper an effective mechanism for this kind of voids.

The rest of this paper is organized as follows. Section II presents the problem of open voids. Section III describes two algorithms that we propose to discover and maintain open voids on a deployed WSN boundary. Section IV presents the proposed mechanism for handling open voids in WSNs. Section V evaluates performances of our void-avoidance mechanism. Section VI concludes the paper.

II. OPEN-VOID PROBLEM

A void is an area where sensor nodes are unable to route packets or straightforwardly inalienable. It appears when using a random deployment of nodes or because of node breakdown due to various reasons, such as circuit breakdown, destruction or energy exhaustion of some nodes. The problem of geographical routing is that stuck nodes, located on a void boundary, can receive packets destined to the sink. Let us consider the example in Figure 1, where black nodes are located on the void boundary and node i must forward a packet to the destination node d. In this case, node i is stuck because there is any forwarding neighbor closer to node d. Once received by node i, the packet cannot have a positive progression towards node d. This packet will be directed towards node j (or node k) in a negative progression around the void. The node where a packet may get stuck is called a local minimum.

Without an efficient void-handling mechanism, data packets are dropped, wasting the network resources and communications can be lost between a few pairs of nodes. Such a behavior is strongly undesirable in WSNs and the loss of some critical information can harm the network mission.

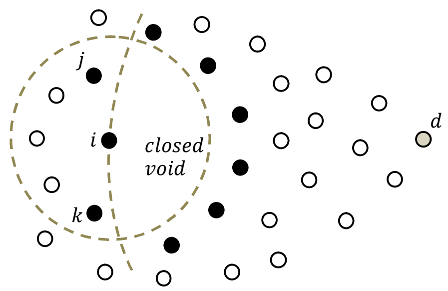
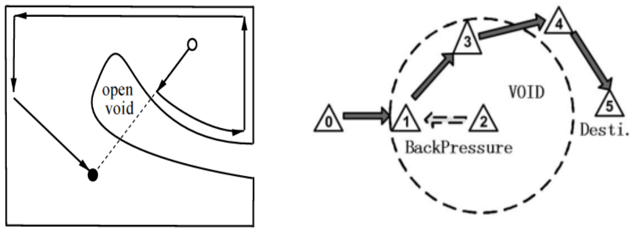


Figure 1. The void problem: i is a stuck node.

Open voids are located on the boundary of a deployed WSN. In order to reduce their negative impact on the routing effectiveness, particularly in case of real-time applications, several void-handling techniques exist in the literature. They

 ISBN: 978-1-61208-144-1

gather in two classes (Figure 2): right-hand rule [4-13] and backpressure rule [14-17].



(a) Right-hand rule [4] (b) Backpressure rule [14]

Figure 2. Classes of techniques handling open voids.

The techniques belonging to the first class use boundary nodes to route any stuck packet towards its destination. In [4], the geographical routing algorithm GPSR is proposed. On a non-stuck node, the packet is forwarded by GPSR to the nearest neighbor to the destination node (greedy forwarding mode). Consequently, the destination is approximate hop by hop until reached by the packet. When this mode fails, the current node uses the face routing to overcome the meted void (perimeter forwarding mode). Boundary nodes apply the right-hand rule until the packet arrives at a node closer to the destination. Several other algorithms using the face routing were proposed later [5-9]. However, [18] showed that planarisation algorithms used to obtain a planar graph, such as Gabriel graph [4], reduced the number of usable links in a network. However, sensor networks deployed for real-time applications cannot admit this reduction because of its negative impact on exploring multiples paths towards the packet destination (load balancing and network fluidity).

On the other hand, the techniques belonging to the second class exploit the backpressure beacons broadcasted by the boundary nodes. When receiving these messages, upstream neighbors get alternative paths around the met void for next data packets. SPEED [14] is a spatiotemporal communication protocol proposed for WSN. It assures an end-to-end soft real-time for data parquets, requires each node to maintain information on its neighbors and employs the geographical forwarding to choose routing paths.

Moreover, SPEED maintains a desired delivery speed across sensor networks with a two-tier adaptation included for diverting traffic at the networking layer and locally regulating packets sent to the MAC layer [14]. It considers a routing void as a permanent congestion. In SPEED, a stuck node drops the received packet and sends out a backpressure beacon informing its neighbors about its final incapacity to forward the next packets. When its forwarding neighbors are stuck nodes, the current node drops the packet and broadcasts a backpressure beacon. This process is repeated until an alternative path is found or the source node reached by the beacons. To improve QoS guarantees, former works [15][16] proposed extensions to SPEED but they not changed the technique for handling routing voids.

The right-hand rule is less effective when handling open voids. It excessively uses boundary nodes and consumes rapidly their energy. In this case, several sessions can use a same boundary, where the problems of collisions and delays of packets. In the same way, the backpressure rule generates not only many control packets but also drops data packets in concave zones of voids. Routing paths are long because of backpressure beacons, from where links are overloaded and packets delayed. These packets will be dropped after their deadline expires, a non-desirable situation in case of real-time application.

To mitigate these insufficiencies, we propose an effective mechanism for handling open voids in WSNs. Called OVA-nb (*Oriented Void-Avoidance on network boundary*), the proposed mechanism orients each stuck packet on the network boundary towards its destination node. It uses the geographical coordinates of the current node, those of the network center and those of the packet destination node to compute the packet orientation around an open void. It is based on two simple and effective algorithms: NBD (*Network Boundary Discovery*) and NBM (*Network Boundary Maintenance*). The first algorithm identifies nodes forming the network boundary just after its deployment and the second one maintain this boundary in reactive manner. Unlike existing techniques using long routing paths (Figure 2), OVA-nb uses short paths to avoid open voids (Figure 3).

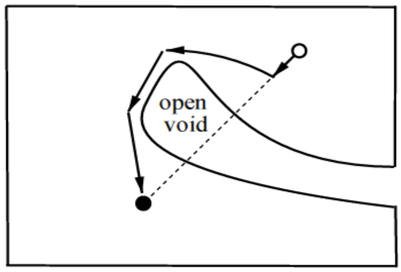


Figure 3. A short path used by our mechanism.

III. PROPOSED ALGORITHMS

Existing algorithms to discover and maintain voids, such as BOUNDHOLE [10] and the right-hand rule [11-13], inserts information about each boundary node in the VD (Void-boundary Discovery) packet, increasing the node memory requirements and reducing the algorithm scalability. Moreover, these algorithms periodically check an eventual failed node and rediscover the entire void if a boundary node fails. It would be interesting to rediscover only the affected local section of the void. The VD packet size grows whenever it moves forward on the boundary of a void to discover. Therefore, existing algorithms [10-13] deplete a significant portion of boundary nodes energy. The same drawback is true for the void maintenance procedure used by these algorithms. BOUNDHOLE [10] does not address the open void as a special case. The outside of the network deployment scope, including the open void shown in Figure 2-a, is considered as a great void. For each stuck packet on the network boundary, the algorithm uses a long routing path formed mainly by boundary nodes. At the same time, the right-hand rule does not consider an open void as a particular problem. It handles only the closed voids located inside a deployed sensor network.

To overcome these limits, we propose two simple and efficient algorithms. The NBD algorithm brings back all the nodes forming the boundary of a deployed WSN and then calculates and communicates its center. The NBM algorithm detects and then updates any topology change that can occur on the network boundary during its mission.

A. NBD algorithm

A designed sink (node c_i in the Figures 4 and 5) initiates the NBD algorithm when deploying a WSN. The algorithm operation is based on the GPSR protocol [4] to find the node closest to a virtual point located at one end of the network field. This node will complete the process of exploring the network boundary. The NBD algorithm takes place in three phases: initial phase, intermediate phase and final phase.

1) Initial phase: sink c_i selects the nearest border of a network field; i.e., the line which passes by one of the points B1, B2, B3 or B4 in Figure 4. Then node c_i projects its geographical location on the selected border. The resulting point (B1 in our example) represents the fictitious destination d_f used by the NBD algorithm to discover the nodes forming the network boundary.

2) Intermediate phase: the sink c_i sends to the fictitious node d_f a new packet ND (Network-boundary Discovery), whose header fields are summarized in TABLE I, to identify the fields Min and Max of the network boundary. The packet ND is routed by using greedy and perimeter modes of the GPSR protocol. When node b_i receives the packet ND, it launches the perimeter mode on the network boundary (Figure 5-a). During this process, the fields N1Up (1-hop upstream boundary node), N1Down (1-hop downstream boundary node) and N2Down (2-hops downstream boundary node) of each intermediate node are updated. When b_i (node that initiated the last perimeter mode) receives the packet ND for a second time, it deduces that it is the closer node to d_f. Thus, b_i execute the final phase of the NBD algorithm.

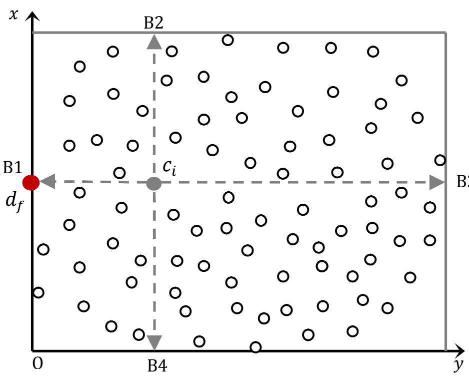


Figure 4. Fictitious destination d_f for the NBD algorithm.

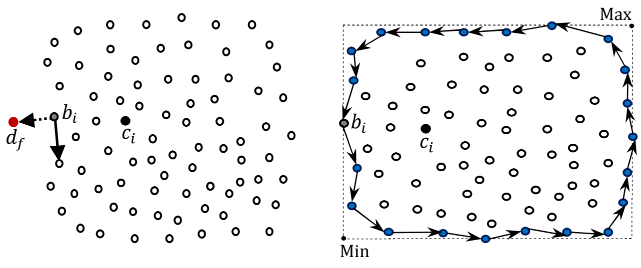
3) Final phase: when receiving the packet ND, node b_i computes the network center (the midpoint of $\overline{\text{Min Max}}$), drops the packet ND and sends a new packet NU (Network-boundary Update), marked by its identifier, to browse the network boundary in the opposite direction of the packet ND. The header fields of the packet NU are summarized in TABLE II. Each boundary node b_i that receives the packet NU updates its boundary information (NBorder=1 and NCenter=NU.NCenter) and verifies the field NodeUp of packet NU. If this field identifies a neighbor of b_i then node b_i updates its field N2Up by NodeUp, otherwise N2Up receives N1Up. Note that N1Up and N1Down are used to maintain the network boundary, N2Up (2-hops upstream boundary node) and N2Down to route packets using two hops on the network boundary. This routing technique reduces energy consumption and minimizes end-to-end delays of the routed packets.

TABLE I. THE HEADER FIELDS OF THE PACKET ND

Field	Mission/Content
PerimID	Identifier of the node having lance the last perimeter mode
DestID	Coordinates of the fitifious destination d_f
Mode	Forwarding mode of the packet ND: Greedy or Perimeter
Distance	Distance from d_f to the last node initiated a perimeter mode
Min	Coordinats of the minimum point on the network boudary
Max	Coordinats of the maximum point on the network boudary
NodeUp	Identifier of the boundary node having sent the packet ND

TABLE II. THE HEADER FIELDS OF THE PACKET NU

Field	Mission/Content
NCenter	Coordinates of the network center
NodeUp	Identifier of a node having sent the pauqet NU



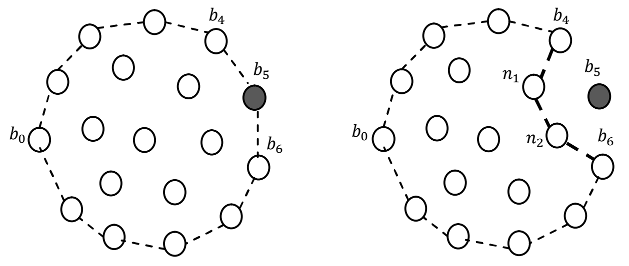
(a) Fictitious destination d_f. (b) Nodes forming the network boundary

Figure 5. Discovery process of the network boundary.

B. NBM algorithm

Some network-boundary nodes may stop working because of insufficient energy or hardware failure. The network boundary can also change shape following the redeployment of nodes on the outside the network. For information usable by any routing process, the algorithm NBM distinguishes two cases: (a) Failed node on the network boundary, (b) Redeployed node outside the network but near its boundary.

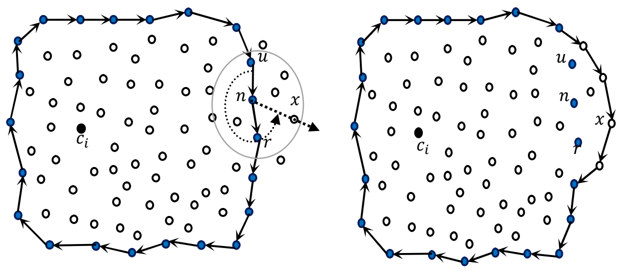
1) Failed node on the network boundary: through its field N1Up, each boundary node b_i can detect the absence of its direct upstream boundary node b_{i-1}. On expiry of the validity time of b_{i-1} in its neighbors table, node b_i discovers a new boundary segment to connect to the old one. Following the failure of boundary node b_5 in Figure 6-a, node b_6 discovers the new segment $b_6n_1n_2b_4$ that connects to the old segment $b_4b_0b_6$ of the network boundary (Figure 6-b). For this discovery, b_i considers b_{i-1} as fictitious destination, sets forwarding mode to perimeter in the packet ND and executes the intermediate phase of the algorithm NBD. The discovery of new nodes is completed in the first node encountered in the old boundary segment (node b_4 in Figure 6-b). This node is recognized by its field N1Up that is different from the default value. Once the two segments connected, the packet ND will continue its travel to restore the full information of the new network boundary. Upon receiving the packet ND, b_i (node b_6 in Figure 6-b) executes the final phase of the NBD algorithm updating fields of nodes on the network boundary.



(a) Boundary node b_5 failed. (b) Network boundary updated.

Figure 6. Network boundary updating after a node failure.

2) Redeployed node outside the network: upon receiving a location beacon from a neighbor x, boundary node n checks its neighbors table. If x is outside the network, node n sends a new packet NS (Network-boundary Suppression), marked by its identifier, on the actual network boundary. Its mission is removing the information concerning this boundary. When receiving the packet NS, each intermediate node b_i resets the fields concerning the network boundary (NBorder, N1Up, N2Up, N1Down and N2Down). At the end, node n drops the packet NS and executes the NBD algorithm to discover the new network boundary. Having the updated fields N1Up and N1Down, node n uses its 1-hop boundary neighbors u and r to perform the following rule: if $\widehat{unx} > \widehat{unr}$ then node x is outside the network (Figure 7-a).



(a) Node x deployed outside the network. (b) Network boundary updated

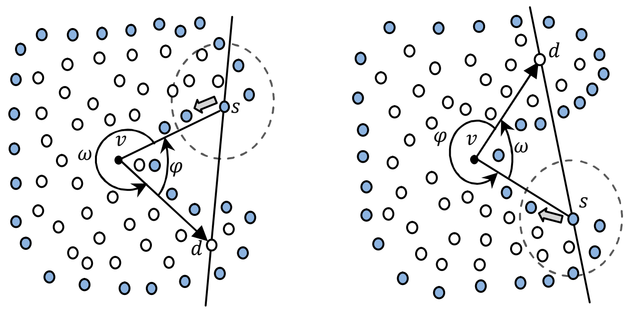
Figure 7. Network boundary updating after a node deployment.

IV. PROPOSED MECHANISM

The proposed OVA-nb mechanism orients towards the sink all packets arriving on the network boundary. Its role is to prevent these packets from drops by nodes located on boundaries of open voids. Having the network center and the updated fields NCenter, N2Up and N2Down, boundary node s (s.NBorder=1) forwards any received packet p to its destination node d by using the angles $\varphi = \widehat{dvs}$ and $\omega = \widehat{svd}$, shown in Figure 8. When receiving p, node s performs the following rules:

- If $\varphi < \omega$ (Figure 8-a) then p is forwarded at the right of the line (sd). Thus, node s updates the orientation field in p if necessary, constructs its set R (greedy forwarding neighbors of s located at the right of line (sd)) and executes the following rule: if R is empty the next-hop node n of p is identified by the field s.N2Down, otherwise n is chosen from R.
- If $\varphi \geq \omega$ (Figure 8-b) then p is to forward at the left of the line (sd). In this case, node s updates the orientation field of p if necessary, constructs its set L (greedy forwarding neighbors of s located at the left of line (sd)) and executes the following rule: if L is empty the next-hop node n of p is identified by the field s.N2Up, otherwise n is chosen from L.

Note that the next-hop node n is chosen from the set R (or L) according to the routing strategy of the implemented protocol that uses the mechanism OVA-nb. Associated with SPEED for performance evaluation, OVA-nb uses the neighbor delivery speed as a criterion to choose the next-hop of the packet p. Also, when s is not a network-boundary node (i.e., s.NBorder=0), it executes the routing strategy used by the implemented protocol for choosing the next-hop of each data packet p.



(a) Forwarding at the right of (sd) (b) Forwarding at the left of (sd)

Figure 8. Packet orientation by the mechanism OVA-nb.

V. PERFORMANCE EVALUATION

Since we are interested by critical applications using WSNs, we first implemented the well-known real-time routing protocol SPEED by using the network simulator ns-2 [19]. For better performance, we associate with SPEED the mechanism OVA-nb, to handle open voids, and the resulting protocol is called SPEED-nb.

We compare SPEED-nb performance to those traditional protocols SPEED and GPSR. We use simulation scene with a grid distribution of nodes and the parameters summarized in TABLE III. Our objective is to show the inadequacy of existing techniques in handling open voids. This scene has a size of 800m×800m and contains 925 nodes. It contains an open void with 120m as radius, located on the right boundary of the scene. Six source nodes, selected randomly and located at the top of the void, periodically send data packets to a destination node located at the bottom of the void. Note that to enable a minimum of forwarded packets to the same destination node by the evaluated protocols, two other source nodes are selected from the left side of the void.

We evaluate performance of the protocols SPEED-nb, SPEED and GPSR at packet rate of 2 p/s. We vary the packet deadline between 50ms and 300ms. At the end of each simulation and for each protocol, we measure the packet delivery ratio, the average routing-path length, the average boundary-energy consumed and the average gain in deadline for each received packet. Each point in our graphs represents the average results of 15 simulations, with random source nodes for each simulation, performed under same conditions and during 221s.

TABLE III. SIMULATION PARAMETERS

MAC layer	IEEE 802.11
Radio model	RADIO-NONOISE
Propagation model	TwoRayGround
Antenna model	OmniAntenna
Queue model	Queue/DropTail/PriQueue
Size of the queue	50 packets
Canal de transmission	WirelessChannel
Wireless interface	WirelessPhy
Bandwidth	200 Kb/s
Size of data packets	32 bytes
Energy model	Energymodel of ns-2
Radio range	40 m
Transmission power	0.666 w
Reception power	0.395 w

Figure 9 shows that 75ms of packet deadline is sufficient for SPEED-nb to route successfully all the packets because it proposes to take a short path toward the destination node as shown in Figure 10. To reach the same performance, GPSR needs 300ms as packet deadline. This is because the face routing of GPSR which uses many boundary nodes before reaching the sink. Therefore, too long and busy routing paths are used by GPSR, as shown in Figure 10, and many packets are dropped because their deadline expires.

SPEED also removes many data packets, as shown in Figure 9, because backpressure beacons that generate stuck nodes delay the next packets in their progression and block definitely some source nodes. For this reason, packet delivery ratio of SPEED remains weak despite the growth of packet rate (Figure 9). For packet deadline exceeding 250ms, GPSR uses long routing paths to deliver the maximum number of packets, but the protocols SPEED and SPEED-nb use short paths for all delivered packets (Figure 10).

Figure 11 show that excessive use of boundary nodes in GPSR has led to large energy depletion in these nodes, but SPEED-nb consumed less energy of boundary nodes because it uses short routing paths. Consequently, it delivers data packets with significant residual deadline (i.e., reduced end-to-end delays), as shown in Figure 12.

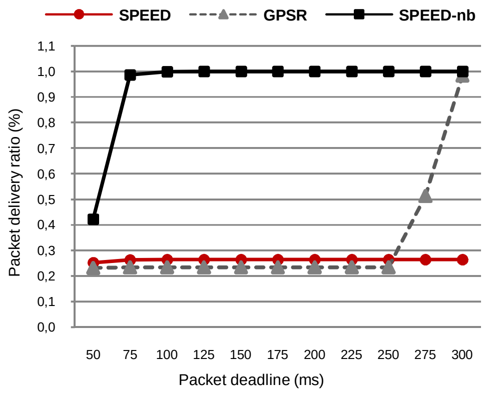


Figure 9. Success rate in delivering data packets.

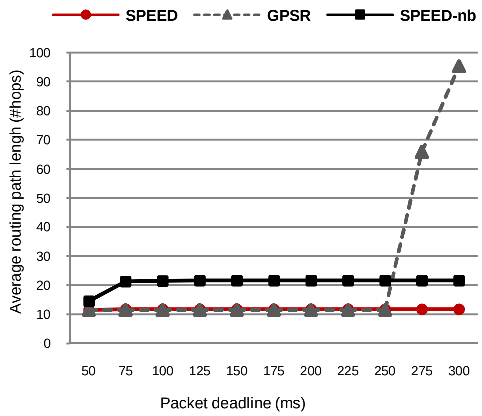


Figure 10. Average routing path length of delivered packets.

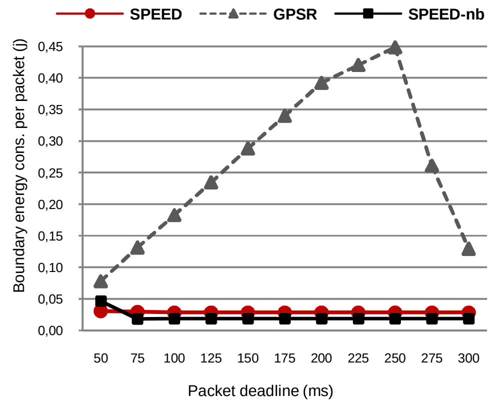


Figure 11. Boundary energy consumed per delivered packet.

VI. CONCLUSION

We proposed the mechanism OVA-nb whose role is to orient each stuck packet from the network boundary towards its destination node. We also proposed two simple and

 ISBN: 978-1-61208-144-1

effective algorithms used by OVA-nb to discover and maintain all boundary nodes of a deployed sensor network; where open voids are frequently formed. To evaluate the OVA-nb performances, we associated it with the well-known protocol SPEED. Evaluated by simulation, obtained protocol SPEED-nb outperformed the traditional protocols SPEED and GPSR in terms of packet delivery ratio, average routing path length, boundary energy consumed for each delivered packet and average residual deadline of delivered packets. Our mechanism OVA-nb resolved the insufficiencies of existing techniques in handling open voids. It is simple to implement, effective in handling open voids and can be easily associated with any geographical routing protocol.

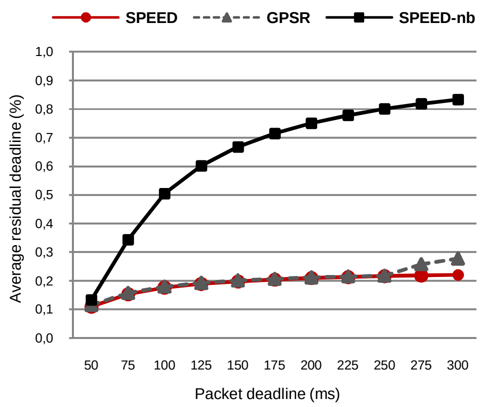


Figure 12. Average residual deadline of delivered packets.

Using the same approach, our current work is to propose a novel mechanism to deal with closed voids in WSNs, which will improve performances of the void-avoidance mechanism that we already proposed in [20][21]. We also plan to implement our proposals in a real scenario based on Imote2 sensor nodes.

REFERENCES

[1] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information," Proc. of 9th Annual Int'l Conference on Mobile Computing and Networking, pp. 96-108, San Diego, CA, September 14-19, 2003.

[2] J. Newsome and D. Song, "GEM: Graph Embedding for Routing and Data-Centric Storage in Sensor Networks without Geographic Information," Proc. of the 1st Conference on Embedded Networked Sensor Systems, pp. 76-88, Los Angeles, USA, Nov. 5-6, 2003.

[3] R. Fonseca, S. Ratnasamy, J. Zhao, C. Tien Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon Vector Routing: Scalable Point-to-Point in Wireless Sensornets," Proc. of the 2nd Symposium on Networked Systems Design and Implementation (NSDI), pp. 329-342, Boston, Massachusetts, USA, May 2-4, 2005.

[4] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," Proc. of the ACM/IEEE Conference on Mobile Computing and Networking (MobiCom), pp. 243-254, Boston, Massachusetts, USA, August 6-11, 2000.

[5] B. Leong, S. Mitra, and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information", Proc. of the 13th IEEE International Conference on Network Protocols (INCP'05), Boston, MA, USA, November 6-9, 2005.

[6] L. Moraru, P. Leone, S. Nikoletseas, and 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, Chania, Crete Island, Greece, Oct. 22-26, 2007.

[7] L. Moraru, P. Leone, S. Nikoletseas, and J. Rolim, "Geographic Routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks," Lecture Notes in Computer Science (LNCS), Vol. 5198, pp.148-161, 2008.

[8] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks," IEEE Transactions on Networking, Vol. 16(1), pp. 51–62, 2008.

[9] 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," Lecture Notes in Computer Science (LNCS), Vol. 5804, pp. 102–115, 2009.

[10] Q. Fang, J. Gao, and L.J. Guibas, "Locating and bypassing routing holes in sensor networks," Journal of Mobile Networks and Applications, Vol. 11(2), pp. 187–200, Oct. 2006.

[11] F. Yu, E. Lee, Y. Choi, S. Park, D. Lee, Y. Tian, and S.H. Kim, "A modeling for hole problem in wireless sensor networks," Proc. the ACM International Conference on Wireless Communications and Mobile Computing, pp. 370-375, Amman, Jordan, Sept. 6-8, 2007.

[12] F. Yu, Y. Choi, S. Park, E. Lee, Y. Tian, M. Jin, and S.-H. Kim, "Anchor node based virtual modeling of holes in wireless sensor networks," Proc. of the IEEE International Conference on Communications, pp. 3120-3124, Beijin, China, May 19-23, 2008.

[13] F. Yu, S. Park, E. Lee, and S.H. Kim, "Hole Modeling and Detour Scheme for Geographic Routing in Wireless Sensor Networks," Journal of Communications and Networks, pp. 327-336, 2009.

[14] T. He, J.A. Stankovic, C. Lu, and T. Abdelzaher, "A Spatiotemporal Communication Protocol for Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, Vol. 16(10), pp. 995-1006, October 2005.

[15] E. Felemban, C.G. Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 5(6), pp. 738-754, June 2006.

[16] W. Cheng, L. Yuan, Z. Yang, and X. Du, "A real-time Routing Protocol with Constrained Equivalent Delay in Sensor Networks," Proc. of the 11th IEEE Symposium on Computers and Communications, pp. 597-602, Sardinia, Italy, June 26-29, 2006.

[17] W. Jia, T. Wang, G. Wang and M. Guo, "Hole Avoiding in Advance Routing in Wireless Sensor Networks," in Proc. of the IEEE Wireless Communications and Networking Conference, pp. 3519-3523, Hong Kong, China, March 11-15, 2007.

[18] K. Seada, A. Helmy, and R. Govindan, "Modeling and analyzing the correctness of geographic face routing under realistic conditions," Ad Hoc Networks, Vol 5(6), pp. 855–871, 2007.

[19] Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, "The ns Manual", available at: *http://www.isi.edu/nsnam/ns/, January 12, 2011.*

[20] M. Aissani, A. Mellouk, N. Badache, and B. Saidani, "Oriented Void Avoidance Scheme for Real-Time Routing Protocols in Wireless Sensor Networks," Proc. of the IEEE GLOBECOM Conference, pp. 83-87, New Orleans, LA, USA, 30 Nov. - 04 Dec. 2008.

[21] M. Aissani, A. Mellouk, N. Badache, and M. Boumaza, "A Novel Approach for Void Avoidance in Wireless Sensor Networks," Int'l Journal of Communication Systems (IJCS), John Wiley & Sons Editions, Vol. 23(8), pp. 945–962, 2010.