Oriented 2-hop Forwarding Approach on Voids Boundaries in Wireless Sensor Networks

Mohamed Aissani, Sofiane Bouznad, Abdelmalek Hariza, and Salah-Eddine Allia

Research Unit in Computer Science (UERI), Ecole Militaire Polytechnique (EMP)
P.O. Box 17, Bordj-El-Bahri 16111, Algiers, Algeria
{maissani, bouznad.sofiane}@gmail.com
{malik-abd, s.alia}@hotmail.com

Abstract — **Scalable geographical routing protocols suffer from voids that appear in Wireless Sensor Networks (WSNs). Several techniques are proposed in literature to handle this problem, but they present some limits, particularly in time-critical applications. Consequently, we propose in this paper a new 2-hop forwarding approach that orients any packet which arrives at a boundary node in the shortest path towards the sink. The handled voids can be either closed within a deployed WSN or open on the network boundary. To keep unchanged the actual size of a void for a long time, the use of a 2-hop forwarding mode is privileged to preserve the limited energy of boundary nodes. The information needed for our approach is provided by simple and reactive algorithms that we propose in this paper to discover and maintain the boundaries of voids. Associated with the SPEED real-time routing protocol, our proposal performs very well in terms of packet delivery ratio, control packet overhead, network and boundary nodes energy consumption.**

Keywords–Sensor networks; geographical routing; closed voids; open voids; void-handling techniques.

I. INTRODUCTION

WSNs can be deployed quickly in sensitive and/or difficult to access areas. Their mission is usually to monitor an area, to take regular measurements and to send alarms to the sink(s) of the deployed network. Many applications using WSNs are then emerging in several areas, such as defense, security, health, agriculture and smart homes. They generally used geographical routing ensuring scalability and allowing positive progression of packets towards the sink. However, geographical routing has two major problems. First, it is not applicable if a sender node has no opportunity to know its geographical locations. This problem can be solved by virtual coordinate systems. Second, there may be voids between a source node and a sink. These voids can be concave, convex, closed or open. Conversely to the closed voids that appear within a deployed WSN, the open voids are frequently formed on the boundary of this network. A geographical routing path towards the sink can be failed due to lack of relay nodes because of a void.

To handle the problem of voids in geographical routing, several solutions are proposed in literature [1-14], but they present some shortcomings, particularly in case of time-critical applications using WSNs. As a contribution in resolving this problem, we propose an oriented 2-hop forwarding approach handling effectively all kinds of voids in WSNs. To do so, we also propose four reactive algorithms to discover and then maintain each void that appear in a deployed WSN. Then each data packet received by a boundary node is forwarded towards its destination by using the shortest path and the minimum number of boundary nodes. This strategy aims to reduce the packet end-to-end delay, to economize the energy of boundary nodes and then to preserve for a long time the actual form of each discovered void. Note that the present work improves our previous work [13, 14] by handling both open and closed voids, using a 2-hop forwarding mode on the void boundary and maintaining dynamically each discovered void in a WSN.

The rest of the paper is organized a follows. Section II presents the problem of voids and discuses the existing void-handling techniques. Section III provides two efficient algorithms for discovery and maintenance of voids in WSNs. Section IV proposes an oriented 2-hop forwarding mode to use by each boundary node. Section V evaluates performance of the proposed approach. Section VI concludes the paper.

II. VOID PROBLEM IN GEOGRAPHICAL ROUTING

Routing voids are areas where nodes cannot forward data packets or completely unavailable. These voids are formed due to either the random deployment of nodes or the node failure because of various reasons, such as circuit failure, destruction or energy exhaustion. Therefore, packets to forward are often blocked in their positive progression towards their destination.

Suppose the example in Figure 1, where black nodes are boundary nodes and node s has to forward data packets to destination d. Node s is stuck because it has no neighbor so close to d to be selected as a forwarder node; i.e., the FS (Forwarding candidate neighbors Set) of node s is empty. Once received by node s, data packets cannot progress positively towards destination d. Thanks to a recovery mode, those packets will be forwarded to node j (or to node k) in a negative progression to bypass the void. This scenario, called the local minimum phenomenon, often occurs when a void appears in a WSN. We then say that s is a stuck (or a blocked) node.

Without using an adequate void-handling technique, data packets can be removed in a WSN wasting the nodes resources and communications can be lost between some pairs of nodes. Such behavior is undesirable in a time-critical application because the loss of some captured information can interfere with the network mission. To reduce the negative impact of voids on the effectiveness of geographical routing, void-handling techniques are available in literature. They fall into two classes: those based on the right-hand rule [1-6] and those using the backpressure rule [9-12].

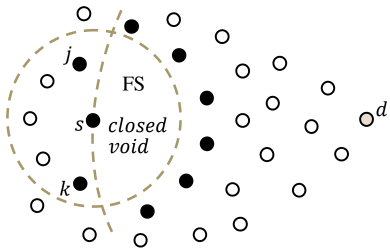


Figure 1. The void problem: the FS of sender s towards destination d is empty.

The techniques belonging to the first class use boundary nodes to route a stuck packet. In most cases, they use long recovery paths, especially in the case of open voids. Proposed in [1], the GPSR algorithm uses two forwarding modes: the greedy mode and the perimeter mode. When a sender node is not blocked, it forwards the current packet to the closest neighbor to the destination node (i.e.; greedy mode). As a result, the destination is approached hop by hop until reached by the packet. When the greedy mode fails, the packet is routed by using a face routing (i.e.; a perimeter forwarding on a planar graph) to bypass the void met. The right-hand rule is thus used on the void boundary until the packet reaches the closest node to the destination. Several other algorithms using the face routing were proposed later [2-5]. However, it has been shown in [15] that the use of planarization algorithms, such as Gabriel graphs [1], reduces the number of useful links in a WSN. This influences the exploration of multiple routing paths allowing load balancing, link-failure tolerance and network fluidity. This is not tolerable in WSNs dedicated to time-critical applications.

However, the techniques belonging to the second class uses the backpressure messages, that are broadcasted by stuck nodes near a void, to route the next packets in alternative paths. He et al. [9] describes SPEED; a QoS routing protocol providing a soft end-to-end real-time to all flows routed in a WSN. In this protocol, each node updates information on its neighbors and uses geographical routing to select paths. In addition, SPEED aims to ensure a certain delivery speed so that each application can estimate the packet end-to-end delay. It deals with a void as it handles a permanent congestion. When a packet is stuck, the sender node drops the packet and broadcasts a backpressure message informing its neighbors about the void met. Then the stuck node will not be considered by the neighbors in their future routing decisions. When neighbors of a node are all stuck, the actual packet is dropped and a backpressure message is broadcasted. This process is repeated until an alternative route is found or the source node is reached by the successive backpressure messages. Extensions to the SPEED protocol have been proposed later in [10-12], but the void-avoidance scheme of the protocol was not modified in these extensions.

Indeed, the right-hand rule is not effective in bypassing voids, especially in case of open voids. It requested a lot of boundary nodes and often used long paths on voids boundaries, resulting in excessive energy consumption of boundary nodes and delays packets due to the overload of these bypassing paths. Then the voids tend to expand rapidly due to energy depletion, complicating the sensor network mission. Similarly, the backpressure rule generates many control packets and removes data packets at stuck nodes in concave areas of some voids. Consequently, routing paths become long because of multiple backtrackings which overload links and delay packets. These packets might be removed in the sensor network after expiration of their deadline. This is again not desirable for time-critical applications. To overcome these weaknesses, we propose in this paper an efficient 2-hop forwarding approach that orients correctly towards the sink each packet received by a boundary node. The proposed approach uses two new mechanisms: the first one, is called OVA-vb (Oriented Void Avoidance on a closed void boundary), which handles the closed voids within the network whereas the second one, is called OVA-nb (Oriented Void Avoidance on the network boundary), and it handles open voids on the network boundary.

Note that the closed voids in a WSN are discovered by the VBD (Void-Boundary Discovery) algorithm and maintained by the VBM (Void-Boundary Maintenance) algorithm that we propose in the next section.

III. PROPOSED VBD AND VBM ALGORITHMS

Existing algorithms for discovery and maintenance of voids, such as BOUNDHOLE [6] and other algorithms based on the right-hand rule [7, 8], insert information on boundary nodes of a void in the VD (Void Discovery) packet, increasing both memory and energy requirements of these nodes and then reducing scalability. In addition, these algorithms perform a periodical check of a void and rediscovers the entire void if one boundary node fails, or it would be economic to discover locally only the changed segment. BOUNDHOLE [6] does not distinguish between an open void and a closed one. Indeed, the outside of a deployed WSN is considered as a great void and data packets that stuck on the network boundary will go on long bypassing paths. Also, the algorithms using the right-hand rule to discover a void do not consider an open void as a particular problem to be handled and they only discover the voids located inside the network. To alleviate these shortcomings, we propose below two effective algorithms (VBD and VBM). The VBD algorithm identifies all nodes forming the boundary of a closed void, calculates and then communicates the void information (i.e.; center and radius) to each discovered boundary node. The VBM algorithm detects and then updates any changes that occur on the boundary of a closed void that was already discovered in a WSN.

A. *Proposed VBD algorithm*

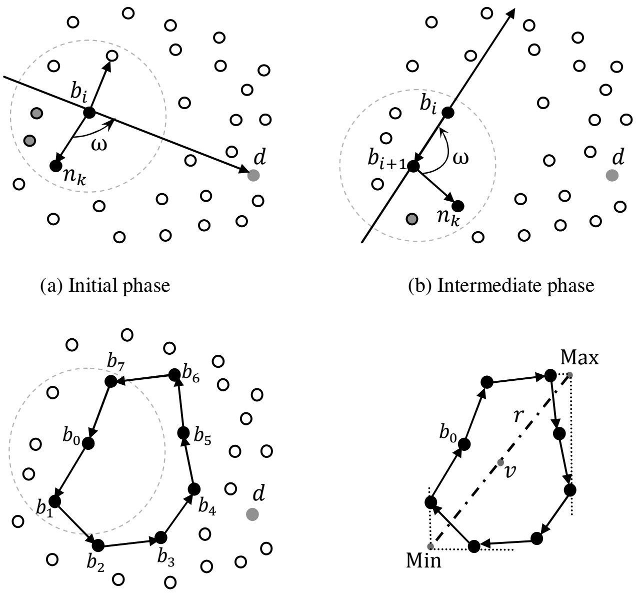
To discover the boundary nodes of a closed void, the VBD algorithm uses the right-hand rule on a Gabriel graph (GG) which preserves the network connectivity [1]. This graph is formed by neighbors of a boundary node where intersections between edges are eliminated to avoid loops. The VBD algorithm operates in initial, intermediate and final phases.

1) Initial phase: when a blocking situation is detected (i.e., FS=ϕ), node b_i performs the following tasks: (a) broadcasts a 1-hop VP (Void back-Pressure) packet announcing its non-availability for the time VT (Void Time-discovery), (b) drops the data packet to increase the network fluidity and (c) sends a

VD (Void-boundary Discovery) packet, marked by its ID, to next boundary-neighbor n_k located at right of vector $\overrightarrow{b_i d}$ (i.e., node n_k having the smallest ω shown in Figure 2-a).

2) Intermediate phase: when receiving the VD packet, the boundary node b_{i+1} broadcasts a VP packet and sends the VD packet to the next intermediate boundary neighbor n_k located at right of $\overrightarrow{b_{i+1} b_i}$ as shown in Figure 2-b. This process is repeated by each intermediate neighbor $(b_{i+2}, b_{i+3}, \dots)$ until the VD packet will be received by the initiator boundary node b_0 at the end of its trip around the void (Figure 2-c).

3) Final phase: by receiving the VD packet at the end of its trip, node b_0 performs the following tasks: (a) extracts from the VD packet the points Min and Max of the discovered boundary $\{b_0, b_1, \dots, b_n\}$, (b) calculates center v of the void which is the midpoint of the segment $\overline{\text{Min Max}}$, and its radius r which is given by: $r = Distance(\text{Min}, \text{Max})/2$, (c) drops the VD packet and then (d) sends a VU (Void-boundary Update) packet, marked by its ID, through the discovered boundary in the opposite direction of the VD packet (Figure 2-d).



(a) Initial phase (b) Intermediate phase

(c) VD packet return (d) Final phase

Figure 2. The void discovery process in the VBD algorithm.

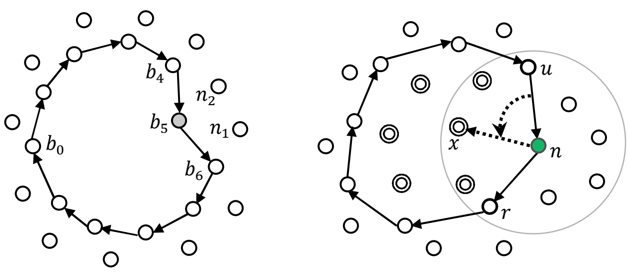
Note that before forwarding the VD packet, node b_i updates its field V1Up by the ID of its successor n_k and checks the field NodeUp in the VD packet. If this field identifies a neighbor then b_i updates its field V2Down (2-hop downstream boundary node) by NodeUp, else V2Down is updated by V1Down. Similarly, each node b_i that receives a VU packet updates its fields about the void and checks the field NodeUp in the VU packet. If this field identifies a neighbor then b_i updates its field V2Up by NodeUp, else V2Up receives V1Up. Note that the fields V2Up (2-hop upstream boundary node) and V2Down are used by the 2-hop forwarding mode of the OVA-vb mechanism which reduces both the node energy consumption and the packet end-to-end delay.

B. Proposed VBM algorithm

Some boundary nodes of a closed void in a WSN may stop working for various reasons. Also, new nodes can be deployed within a closed void to repair it. Thus the proposed VBM algorithm handles these two situations as follows.

1) Boundary-node failure: each boundary node b_i can detect the absence of its direct ascendant boundary neighbor b_{i-1} thanks to its field V1Up. When b_{i-1} expires in the neighbors table T of node b_i, the later discovers a new segment of nodes and connects it to the old segment of the void by running the VBD algorithm. When node b_5 fails in Figure 3-a, node b_6 discovers the new segment of nodes $b_6 n_1 n_2 b_4$ that connects to the old segment $b_4 b_0 b_6$ of the void. When the two segments are connected, the VD packet continues its trip to bring the full information about the new boundary of the closed void. Upon receiving the VD packet at the end, node b_i (i.e.; node b_6 in Figure 3-a) runs the final phase of the VBD algorithm updating the void information in fields of the boundary nodes.

2) Deployment of nodes within a closed void: by receiving a location beacon from a new neighbor x, boundary node n checks if x is located inside the void. Based on its updated fields V1Up and V1Down, node n uses its 1-hop boundary neighbors u and r (Figure 3-b) to execute the following verification: if $\widehat{unx} < \widehat{unr}$ then x is located inside the void. If so, node n sends a VS (Void Suppression) packet, marked by its ID, to visit the boundary of the repaired void. Upon receiving the VS packet, each boundary node removes from its list of voids (VList) the repaired void. Note that parts of a void may still exist due to repairing process, but they will be met later by packets and then discovered by the VBD algorithm.



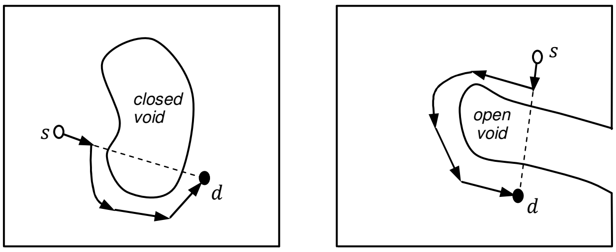
(a) Node b_5 fails. (b) New node x sends a beacon.

Figure 3. The void-maintenance cases in the VBM algorithm.

IV. PROPOSED 2-HOP FORWARDING APPROACH

The proposed 2-hop forwarding approach aims to orient towards the sink any packet that arrives at a boundary node by using an optimal path, as shown in Figure 4. When a sender node s has to forward a packet p towards destination d, it forms its FS (Forwarding candidate neighbors Set) and then distinguishes the three following cases: 1) sender s has no information about voids, 2) sender s is on the network boundary and 3) sender s is on the boundary of a closed void.

1) Sender s has no information about voids ($s.VList$=ϕ): if FS is empty then sender s runs the VBD algorithm to discover the void met, else it forwards packet p to its neighbor n in FS (i.e., one of the hatched nodes in Figure 5). The forwarder n is selected according to the protocol routing metric, such as the relay speed used in SPEED [9].



(a) OVA-vb mechanism (b) OVA-nb mechanism

Figure 4. Packet orientation at a boundary node in our approach.

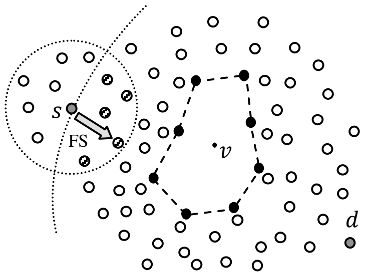
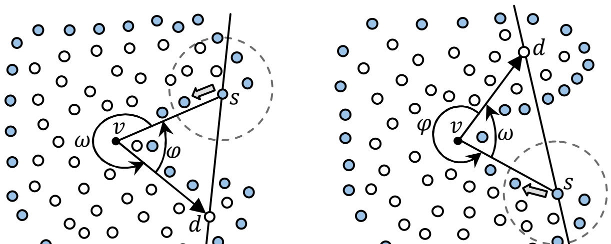


Figure 5. Case 1: sender s has not information about voids.

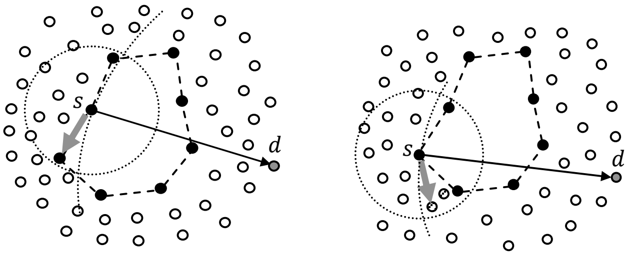
2) Sender s is on the network boundary ($s.NBorder$=1): the sender s uses the OVA-nb mechanism that we proposed in [17] to orient p towards its destination node d by using a 2-hop forwarding mode on the network boundary. Thus, sender s uses the angles $\varphi = \widehat{dvs}$ and $\omega = \widehat{svd}$ (Figure 6) to select the next forwarder n. If φ<ω (Figure 6-a) then sender s selects n from its neighbors located at the right of line (sd), else (Figure 6-b) n is selected from the neighbors of s that are located at the left of line (sd). More details about OVA-nb are given in [17].



(a) (b)

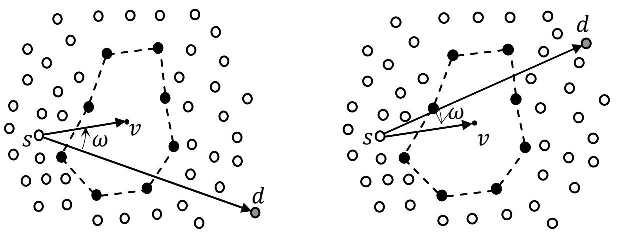
Figure 6. Case 2: sender s is on the network boundary [17]. The next forwarder is located right (a) or left (b) of line (sd).

3) Sender s is on boundary of a closed void ($s.VBorder$=1): the sender s uses the OVA-vb mechanism based on a 2-hop forwarding mode on the void boundary. Thus, packet p is oriented in the correct direction around the void by using a non-boundary node as next forwarder as soon as possible, to preserve the actual form of the void for a long time. If sender s have to route on the void boundary (Figure 7-a), it forwards p to its 2-hop upstream node identified by V2Up (or 2-hop downstream node identified by V2Down) depending on the packet orientation (i.e., right or left of $\overrightarrow{sv}$). If not (i.e., there is at least one non-boundary node in FS as shown in Figure 7-b), sender s forwards p to a neighbor n selected from its RFS (reduced FS) which is formed by the hatched nodes in Figure 7-b. The selection of n is made according to the implemented protocol metric, such as the relay speed used in SPEED [9]. Note that to orient p arround a closed void, sender s uses the angle ω shown in Figure 8. If $sin(\omega) > 0$ (Figure 8-a) then the packet orientation must be at right of $\overrightarrow{sv}$ (i.e., $p.$Orient=1). If not (Figure 8-b) then the packet orientation must be at left of $\overrightarrow{sv}$ (i.e., $p.$Orient=0). By using the field Orient in p, sender s forms its RFS by neighbors in FS that are located either at right of $\overrightarrow{sd}$ when $p.$Orient=1 or at left of $\overrightarrow{sd}$ when $p.$Orient=0.



(a) (b)

Figure 7. Case 3: sender s is on the boundary of a closed void.



(a) Orientation at right of $\overrightarrow{sv}$ (b) Orientation at left of $\overrightarrow{sv}$

Figure 8. Packet orientation updating in the OVA-vb mechanism.

Note that any changes that occur on the boundary (or inside) of a closed void will be immediately detected by a boundary node and then updated by this later after running the VBM algorithm. The reactive maintenance of the open voids on the network boundary is guaranteed by the NBM algorithm that we proposed in [17].

V. Performance Evaluation

To evaluate performance of the proposed 2-hop forwarding approach, we associate the proposed OVA-vb and OVA-nb mechanisms with the well-known SPEED real-time routing protocol by using the ns-2 simulator [16]. We compare the resulting protocol, called SPEED-vb, with the GPSR and SPEED traditional protocols. Note that to handle voids SPEED uses the backpressure rule and GPSR the right-hand rule. The parameters used in our simulations are given in TABLE I.

We used a terrain (scene) with a size of 800m×800m and 960 deployed nodes. For each simulation, we create a void in

 ISBN: 978-1-61208-199-1

the center of this terrain with a radius varying between 60m and 200m. Six sources selected randomly from the left side of the void generate periodic CBR packets to the first destination placed at right side of this void. Meanwhile, six other sources selected randomly from the right side of the void generate periodic CBR packets to the second destination placed at the left side of the same void. The rate of the sources is set to 1 packet/second and the desired delivery speed (the $S_{setpoint}$ defined in [9]) is set to 600m/s, which leads to an end-to-end packet deadline of 100ms. Each point in our graphs is the average of 15 simulations carried out in the same conditions, but with different sources selected randomly for each simulation. To measure the routing performance with the presence of congestion, two nodes located below the void exchanged packets with a rate of 10 packets/second during the simulation time which is set to 224 seconds.

TABLE I. SIMULATION PARAMETERS.

MAC Layer	IEEE 802.11
Radio Layer	RADIO-NONOISE
Propagation Model	TwoRayGround
Antenna Model	OmniAntenna
Queue Model	Queue/DropTail/PriQueue
Queue Size	50 paquets
Transmission chanal	WirelessChannel
Wireless Interface	WirelessPhy
Bandwidth	200 Kb/s
CBR Packet Size	32 octets
Energy Model	energyModel de ns-2
Communication Range	40 m
Transmission Power	0.666 w
Reception Power	0.395 w

We evaluate performance of protocols SPEED-vb, SPEED and GPSR. We vary the void radius and we measure the packet delivery ratio, the control packet overhead, the network and the boundaries energy consumption per delivered packet. The figures 9, 10, 11 and 12 show that the protocols' performance decreases each time the void radius grows because they use long paths around the void. Therefore, deadline of many packets expires before reaching their destination and then they are dropped in the network because we suppose a time-critical application. We also note that the proposed SPEED-vb protocol is the most efficient with the presence of both small and large voids in a WSN. This is due to the performance of the proposed mechanisms used by the boundary nodes.

Figure 9 shows that SPEED is the worst protocol in delivering packets, especially when a void radius is greater than 120m. This protocol overloads its upstream nodes by the backpressure messages generation near the voids. Following the spread of these messages, some sources are blocked and many packets are removed when their deadline expires in congested links. For an acceptable packet deadline (100ms), GPSR performs better than SPEED tanks to its face routing scheme used by boundary nodes. GPSR generates less control packets (Figure 11) that reduces the network congestion. With the adequate orientation of packets ensured by the proposed mechanisms, the SPEED-vb protocol uses the shortest and smoother routing paths compared to the SPEED and GPSR protocols. Therefore, the packet delivery ratio achieved by SPEED-vb is the highest (Figure 9).

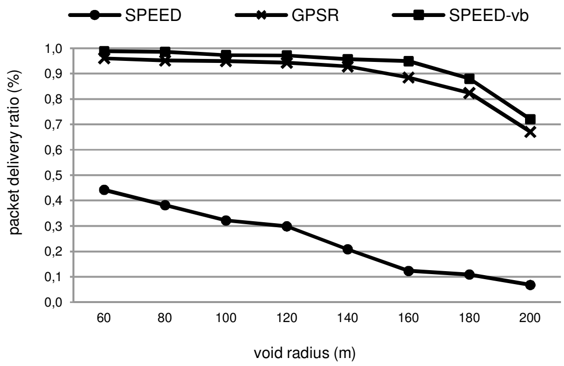


Figure 9. Packet delivery ratio *vs.* Void radius.

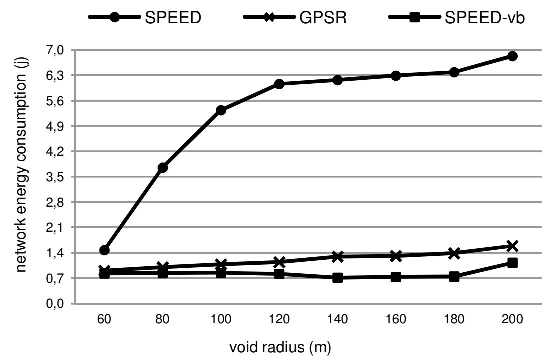


Figure 10. Network energy consumption per delivered packet.

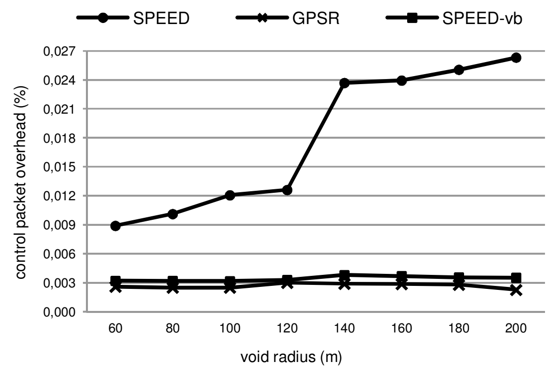


Figure 11. Control packet overhead.

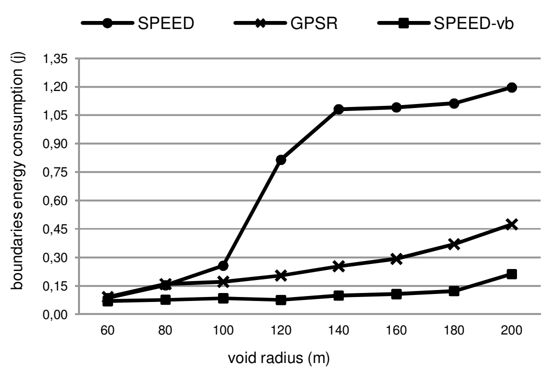


Figure 12. Boundaries energy consumption per delivered packet.

 ISBN: 978-1-61208-199-1

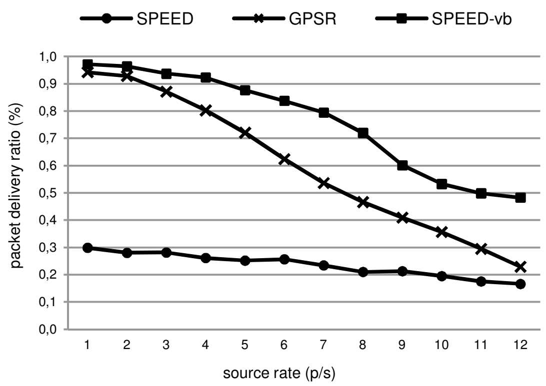


Figure 13. Packet delivery ratio *vs.* Source rate.

For some delivered packets, SPEED consumes much energy of both the network (Figure 10) and the boundary nodes (Figure 12). This is due to excessive control packets generated by SPEED and its useless routing of delayed packets in the network. GPSR is more efficient than SPEED in term of network energy consumption, but it consumes more energy of boundary nodes, especially when the void radius exceeds 100m (Figure 12). For these large voids, GPSR routes most packets on the long parts of the boundary. In the other hand, our SPEED-vb protocol achieves the best tradeoff between the packet delivery ratio and the energy consumption (Figure 10). Since GPSR always uses a unique path connecting a source to the sink, it does not achieve a good node energy balancing. Figure 13 shows these limits when the rate exceeds 3 p/s and a void with 120m as radius is created in center of the terrain. In the other hand, SPEED-vb delivers many data packets thanks to its void-handling mechanisms.

VI. Conclusion and Future Work

We have proposed an oriented 2-hop forwarding approach that provides to each packet received by a boundary node the shortest path towards the sink. Our void-tolerant approach uses two complementary mechanisms: the first one handles the open voids located on the network boundary and the second one handles the closed voids located within the network. These mechanisms use simple and reactive algorithms that we have proposed to discover and then to maintain each void that appears in a deployed WSN. We have associated them with the well-known SPEED routing protocol, designed for real-time applications, and the resulting protocol, called SPEED-vb, achieved the best performance compared to the traditional GPSR and SPEED protocols. The SPEED-vb protocol was able to respond to the shortcomings of the existing void-handling techniques, which are based either on the right-hand rule, such as GPSR, or on the backpressure rule, such as SPEED.

Since we are interested by time-critical applications based on WSNs, our future work will focus on the sequencing of data packets at a node based on the time remaining to reach the sink. The objective is to reduce the number of removed critical packets due to deadline expiration. We also plan to check how our idea can be applied to congested regions and or to the voids created due other problems, like intermittent connectivity.

References

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

[2] L. Moraru, P. Leone, S. Nikoletseas, and J.D.P Rolim, "Near optimal geographical routing with obstacle avoidance in wireless sensor networks by fast-converging trust-based algorithms," Proc. of the 3rd ACM Workshop on Q2SWinet, pp. 31–38, Greece, October 22-26, 2007.

[3] L. Moraru, P. Leone, S. Nikoletseas, and J. Rolim, "Geographical routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks," Lecture Notes in Computer Science, Vol. 5198(1), pp. 148-161, 2008.

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

[5] 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, Vol. 5804(1), pp. 102–115, 2009.

[6] 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, October 2006.

[7] 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 International Conference on Communications (ICC), pp. 3120-3124, Beijing, China, May 19-23, 2008.

[8] [10]F. Yu, S. Park, E. Lee, and S.H. Kim, "Hole Modeling and Detour Scheme for Geographical routing in Wireless Sensor Networks," Journal of Communications and Networks, Vol. 11(4), pp. 327-336, Aug. 2009.

[9] 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.

[10] 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.

[11] 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 (ISCC), pp. 597-602 Italy, June 26-29, 2006.

[12] L. Zhao, B. Kan, Y. Xu, and X. Li, "FT-SPEED: A Fault-Tolerant Real-Time Routing Protocol for Wireless Sensor Networks," Proc. of the International Conference on WiCom, Shanghai, pp. 2531–2534, China, September 21-25, 2007.

[13] 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.

[14] M. Aissani, A. Mellouk, N. Badache, and M. Boumaza, "A Novel Approach for Void Avoidance in Wireless Sensor Networks," International Journal of Communication Systems (IJCS), Vol. 23(8), pp. 945–962, 2010.

[15] 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, August 2007.

[16] Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, "The ns Manual", *http://www.isi.edu/nsnam/ns/*, last consultation in June 2011.

[17] M. Aissani, S. Bouznad, A. Hariza, and S.E. Allia, "An effective mechanism for handling open voids in wireless sensor networks," Proc. of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 24-29, Riviera, France, August 21-27, 2011.