Efficient Forwarding Approach on Boundaries of Voids in Wireless Sensor Networks

Mohamed Aissani, Sofiane Bouznad, Salah-Eddine Allia, and Abdelmalek Hariza
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
{s.alia, malik-abd}@hotmail.com

Abstract—**Geographical routing protocols are scalable, but they must handle voids appearing in wireless sensor networks. Existing void-handling techniques present limits, particularly in real-time applications. Consequently, we propose in this paper an efficient 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 sensor network or open located on the network boundary. To keep unchanged the size of each created void for a long time, the use of a 2-hop forwarding mode is privileged in our approach to preserve the limited energy of boundary nodes. The information needed for our mechanisms 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, and evaluated in several conditions, our proposal performs very well in terms of packet delivery ratio, end-to-end delay, energy consumption, control packet overhead and energy balancing.**

Keywords - Wireless sensor networks; geographical routing; void-handling techniques; closed voids; open voids.

I. Introduction

Wireless sensor networks (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 decision center. 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 sensor network. A geographical routing path towards the destination node (sink) can be failed due to lack of relay nodes because of a void.

As a contribution in resolving the problem of voids in geographical routing in WSNs, we propose in this paper an oriented 2-hop forwarding approach handling effectively all kinds of voids. 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. Since this paper is an extended version of our published conference paper [1], we incorporate pseudo-codes of the proposed algorithms and expanded experiments by evaluating the performance of the proposed void-handling approach when varying the packet data rate at sources, the packet deadline and the number of voids created in the simulation terrain.

Note that to handle the problem of voids in geographical routing, several solutions are proposed in literature [2]-[15], but they present some shortcomings presented in Section III, particularly in case of time-critical applications using WSNs.

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 boundary nodes. Section V evaluates performance of the proposed approach with several void radiuses, data packet rate at sources, packet deadlines and number of voids. 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 [2]-[7] and those using the backpressure rule [10]-[13].

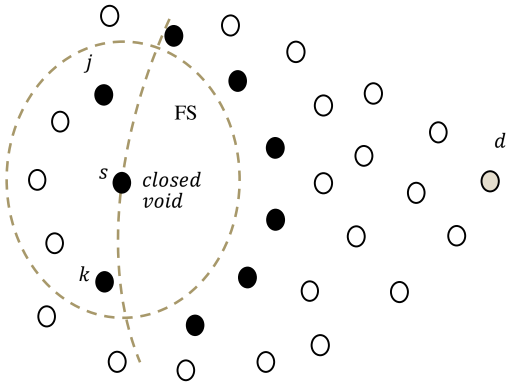


Figure 1. The void problem: the FS of sender s towards destination d is empty.

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 [2], the GPSR (greedy perimeter stateless routing) 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 [3]-[6]. However, it has been shown in [16] that the use of planarization algorithms, such as Gabriel graphs [2], 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. [10] describes the QoS routing protocol SPEED which provides 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 [11]-[13], 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, called OVA-vb (Oriented Void Avoidance on a closed void boundary), handles the closed voids within the network whereas the second one, called OVA-nb (Oriented Void Avoidance on the network boundary), handles open voids on the network boundary. The closed voids in a deployed 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. Note the present contribution improves our previous works [14][15] by handling both open and closed voids in geographical routing protocols in WSNs.

III. PROPOSED VBD AND VBM ALGORITHMS

Existing algorithms for discovery and maintenance of voids, such as BOUNDHOLE [7] and other algorithms based on the right-hand rule [8][9], 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. These algorithms also perform a periodical check of a void and rediscover the entire void if one boundary node fails, or it would be economic to discover locally only the changed segment. BOUNDHOLE [7] 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 updates any changes occurring 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 [2]. This graph is formed by neighbors of a boundary node where all intersections between edges are eliminated to avoid loops problem. 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 $\overline{\text{Min}\,\text{Max}}$, and its radius r 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 of a void in the opposite direction of the VD packet (Figure 2-d).

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. The pseudo-code of the proposed VBD algorithm is given in Figure 3.

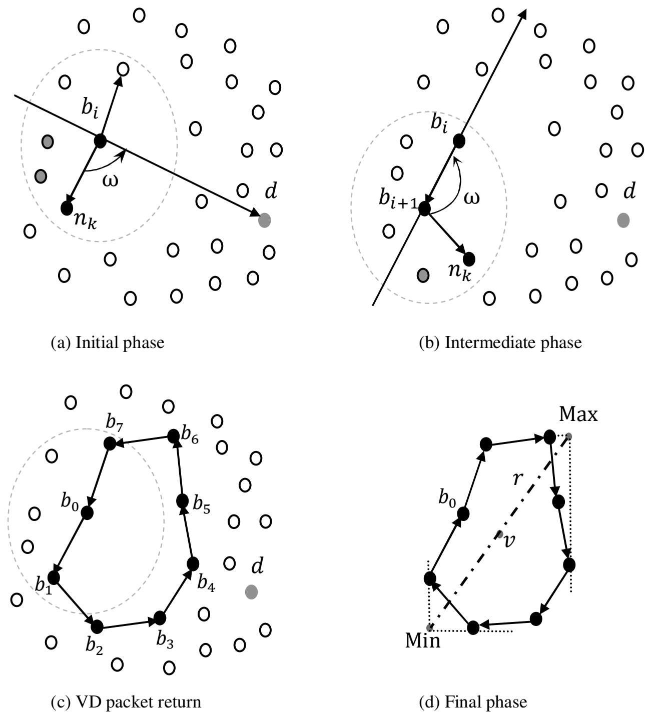


Figure 2. The void discovery process in the VBD algorithm.

B. *Proposed VBM algorithm*

Some boundary nodes of a closed void in a sensor network may stop working for various reasons. Also, new nodes can be deployed within a closed void to repair it. The proposed VBM algorithm handles these cases 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 5-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 5-a) runs the final phase of the VBD algorithm updating the void information in fields of the boundary nodes. The pseudo-code of the VBM algorithm handling the case of boundary-node failure is given in Figure 4.

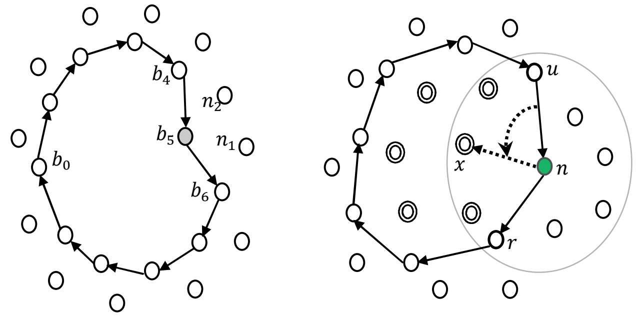
```
Node b treat packet p:
IF (p.type = VD) THEN                          /* void discovery or maintenance */
  Update the fields of packet p;
  IF (p.VIdent = b.ID) AND (p.NodeUp ≠ b.ID) THEN
    Extract Min and Max from packet p;
    Calculate VCenter and VRadius of the void;
    Use packet VU to update the fields of the void;
    Drop packet ;
  ELSE
    IF (b.V1Up ≠ 0) THEN                       /* old segment of the void */
      NextHop ← {n , such as: n.ID = b.V1Up };
    ELSE
      Broadcast packet VP to inform all neighbors;
      Build the sets R and L;
      IF (R ≠ ∅) THEN
        NextHop ← {n , such as: cos ω maximal in R };
      ELSE
        NextHop ← {n , such as: cos ω minimal in L };
      ENDIF
    ENDIF
    Update the fields of boundary node ;
    Forward packet p to successor boundary node in NextHop;
  ENDIF
ENDIF
IF (p.type = VU) THEN                          /* to update the fields of a void */
  Update the fields of packet p;
  Update the fields of boundary node b;
  IF (p.VIdent = b.ID) AND (p.NodeUp ≠ b.ID) THEN
    Drop packet p;
  ELSE
    Forward p to boundary node identified by b.V1Down;
  ENDIF
ENDIF
```

Figure 3. Pseudo-code of the proposed VBD algorithm.

```
Node n detect absence of a neighbor x:
IF (NT.ExpireTime = 0) THEN                    /* NT: table of neighbors of n */
  IF (n.VBorder = 1) THEN
    IF (n.V1Up = x.ID) THEN                    /* x is upstream boundary node */
      Delete neighbor x from NT;
      Execute the VBD algorithm to update the void (Figure 3);
      EXIT;
    ENDIF
  ENDIF
    Delete neighbor x from NT;
ENDIF
```

Figure 4. Pseudo-code of the VBM algorithm when a node fails on boundary of a closed void.

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 5-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. The pseudo-code of the VBM algorithm used when nodes are deployed within a closed void is given in Figure 6.



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

Figure 5. The void-maintenance cases in the VBM algorithm.

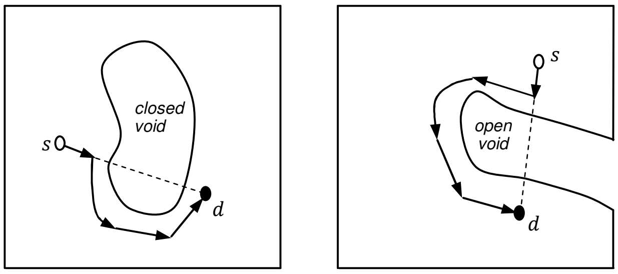
```
Node n receives packet p from neighbor x:
IF (p.type = LOC) THEN                         /* LOC : location packet */
  IF (x ∉ NT) THEN                             /* NT: table of neighbors of n */
    IF (n.VBorder = 1) THEN                    /* n is a boundary node */
      Insert neighbor x in NT;
      Calculate the angle formed by nodes u et r (Figure 5-b);
      IF (unx^ < unr^) THEN                    /* x is within a closed void */
        Execute VBD algorithm to update void (Figure 3);
      ENDIF
    ENDIF
  ELSE
    Update information about neighbor x in NT;
  ENDIF
  Drop packet p;
ENDIF
```

Figure 6. Pseudo-code of the VBM algorithm used when nodes are deployed within a closed void.

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 7. When a sender node s has to forward a packet p towards destination d, it forms its FS 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 closed void.

1) Sender node s has no information about the 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 8). The forwarder n is selected according to the protocol routing metric, such as the relay speed used in SPEED [10].



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

Figure 7. Packet orientation at a boundary node in our approach.

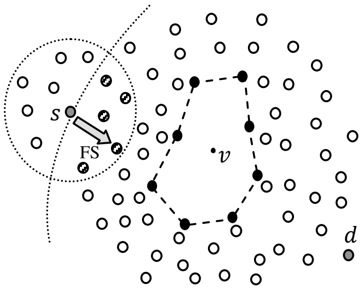
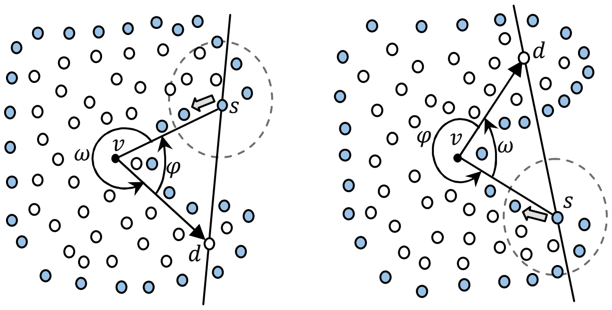


Figure 8. Case 1: sender s has not information about voids.

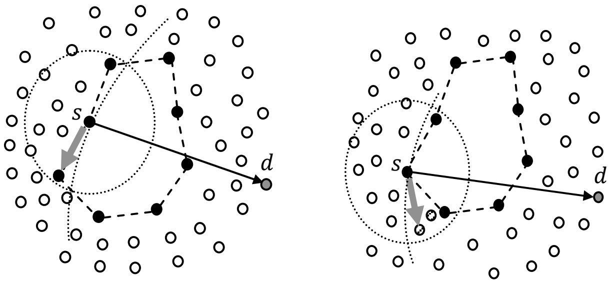
2) Sender node s is located on the network boundary ($s.NBorder$=1): the sender s uses the OVA-nb mechanism that we proposed in [18] to orient p towards its destination node d by using a 2-hop forwarding mode on the network boundary. Thus, sender s uses the angles $\varphi = \widehat{dvs}$ and $\omega = \widehat{svd}$ (Figure 9) to select the next forwarder n. If φ<ω (Figure 9-a) then sender s selects n from its neighbors located at the right of line (sd), else (Figure 9-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 [18].



(a) (b)

Figure 9. Case 2: sender s is on the network boundary [18]. The next forwarder is located right (a) or left (b) of line (sd).

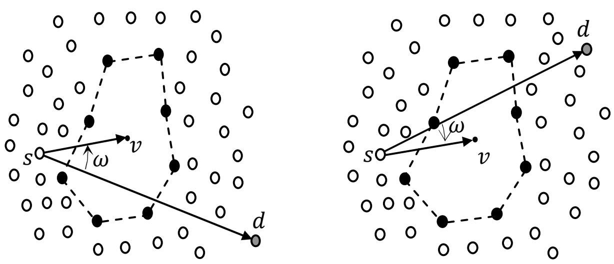
3) Sender node s is on boundary of a closed void ($s.VBorder$=1): the sender s uses the OVA-vb mechanism based on a 2-hop forwarding mode on the void boundary. Thus, packet p is oriented in the correct direction around the void by using a non-boundary node as next forwarder as soon as possible, to preserve the actual form of the void for a long time. If sender s have to route on the void boundary (Figure 10-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 10-b), sender s forwards p to a neighbor n selected from its RFS (reduced FS) which is formed by the hatched nodes in Figure 10-b.The selection of n is made according to the implemented protocol metric, such as the relay speed used in SPEED [10]. Note that to orient p arround a closed void, sender s uses the angle ω shown in Figure 11. If $sin(\omega) > 0$ (Figure 11-a) then the packet orientation must be at right of $\overrightarrow{sv}$ (i.e., $p.$Orient=1). If not (Figure 11-b) then packet orientation must be at left of $\overrightarrow{sv}$ (i.e., $p.$Orient=0). By using field Orient in p, sender s forms its RFS by neighbors in FS located either at right of $\overrightarrow{sd}$ when $p.$Orient=1 or at left of $\overrightarrow{sd}$ when $p.$Orient=0.



(a) (b)

Figure 10. Case 3: sender s is on the boundary of a closed void.

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 [18]. The pseudo-code of the proposed void-handling approach is given in Figure 12.



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

Figure 11. Packet orientation updating in the OVA-vb mechanism.

```
Node s has to forward p toward destination d:
IF (s.NBorder = 1)  THEN                          /* s a is boundary node */
  Execute the OVA-nb mechanism that we proposed in [18];
ELSE                                              /* use of OVA-vb mechanism */
  Build FS (Forwarding candidate neighbors Set);
  IF (s.VList = ∅) THEN                           /* s not informed about voids */
    IF (FS = ∅) THEN
      Execute the VBD algorithm to discover the void (Figure 3);
    ELSE
      NextHop ←{ n,  such as: n∈FS };
      Forward packet p to the neighbor in NextHop;
    ENDIF
  ELSE                                            /* s has information about void(s) */
    Select from s.VList the nearest void (NearestVoid);
    Build the sets L and R from FS;
    IF (CurrentVoid = NearestVoid) THEN           /* same orientation */
(1):  IF (p.Orient = 0)  THEN
          IF (L = ∅)  THEN
            NextHop ←{ n,  such as:  n.ID = s.V2Down };
          ELSE
            NextHop ← { n,  such as:  n∈L };
          ENDIF
      ELSE
          IF (R = ∅)  THEN
            NextHop ←{ n,  such as:  n.ID = s.V2Up };
          ELSE
            NextHop ←{ n,  such as:  n∈R };
          ENDIF
      ENDIF
    ELSE                                          /* different orientation */
      IF (sin ω ≤ 0)  THEN                        /* d is left of CurrentVoid */
        p.Orient ← 0;
      ELSE
        p.Orient ← 1;
      ENDIF
      GOTO (1);
    ENDIF
    Forward packet p to the neighbor in NextHop;
  ENDIF
ENDIF
```

Figure 12. Pseudo-code of the proposed 2-hop forwarding approach detailing the OVA-vb mechanism.

V. Performance Evaluation

In order 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 [17]. We compare performance of the resulting protocol, called SPEED-vb, with the performance of the GPSR and SPEED traditional protocols. Note that to handle voids SPEED uses the backpressure rule and GPSR the right-hand rule. We use the two terrains shown in Figure 13 and we vary the void radius (Section V.1), the data packet rate at sources (Section V.2), the packet deadline (Section V.3) and the number of voids created in terrain (Section V.4). The deployed nodes are organized in a grid and the distance between two successive sensor nodes in each terrain is set to 25 meters. For each simulation, we measure packet delivery ratio, control packet overhead, network energy consumption per delivered packet, node energy balancing and boundaries energy consumption per delivered packet. Each point in our graphs, presented in this section, is the average results of 15 simulations performed under the same conditions, except that source nodes are chosen randomly for each simulation. Simulation parameters are given in TABLE I.

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 packets
Transmission channel	WirelessChannel
Wireless Interface	WirelessPhy
Bandwidth	200 Kbps
CBR Packet Size	32 bytes
Energy Model	energyModel de ns-2
Communication Range	40 meters
Transmission Power	0.666 w
Reception Power	0.395 w

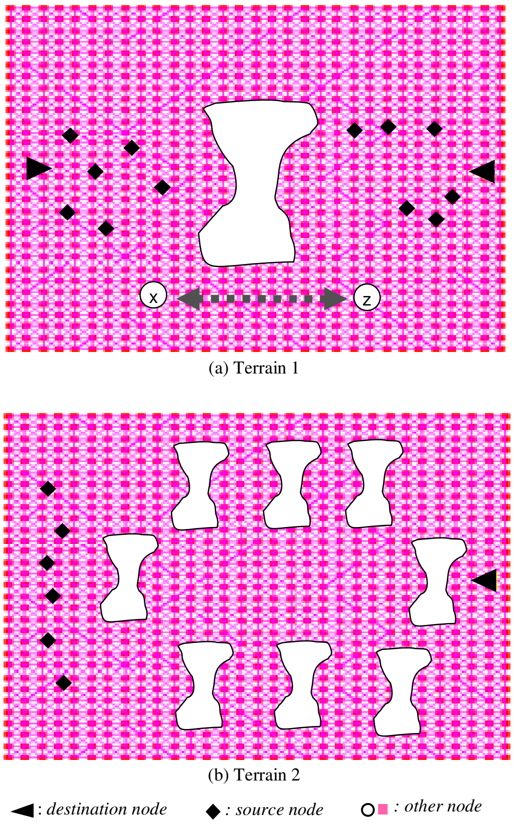


Figure 13. The used simulation terrains.

Terrain 1 shown in Figure 13-a, with a size 800×800 meters and 961 nodes, is used principally to measure the impact of the void radius on the routing performance. We create at the center of this terrain one void with a radius varying between 60 and 200m (meters). Six sources selected randomly from the left side of the void generate periodic CBR (Constant Bit Rate) 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 source rate is set to 1 pps (packet per second) and the desired delivery speed (the $S_{setpoint}$ defined in [10]) is set to 600 mps (meter per second) which leads to an end-to-end packet deadline of 100 ms (milliseconds). To measure the routing performance with the presence of congestion, two nodes x and z, placed under the void in Figure 13-a, exchanged packets with a rate of 10 pps during the simulation time which is set to 224 seconds. The two nodes are enough for congestion and there is no additional traffic excepting the traffic generated by sources.

But Terrain 2 (Figure 13-b), with a size 1240×800 meters and 1296 deployed nodes, is used to measure the impact of the number of voids created in the network on the routing performance. A void with a radius 100 meters is duplicated gradually between the sources and the destination node up to 8 voids in the network. Six sources, selected randomly from the left side of the terrain, periodically send packets to a destination node located on the right side of this terrain. The duration of each simulation using Terrain 2 is fixed to 264 seconds.

1) Performance when varying void radius: we use Terrain 1 (Figure 13-a) in which the source rate is set to 1 pps and the packet deadline to 100ms. We vary the void radius from 60m to 200m and we obtain the results shown in the figures 14-17 where 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 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 14 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 16) 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 14). For some delivered packets, SPEED consumes much energy of both network (Figure 15) and boundary nodes (Figure 17). This is due to excessive control packets generated by SPEED and its useless routing of delayed packets in the network.

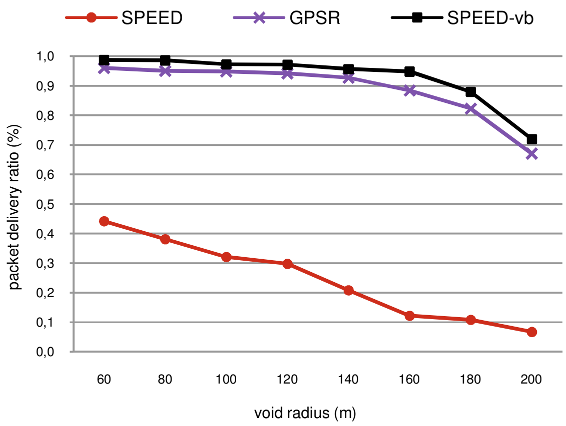


Figure 14. Packet delivery ratio *vs.* Void radius.

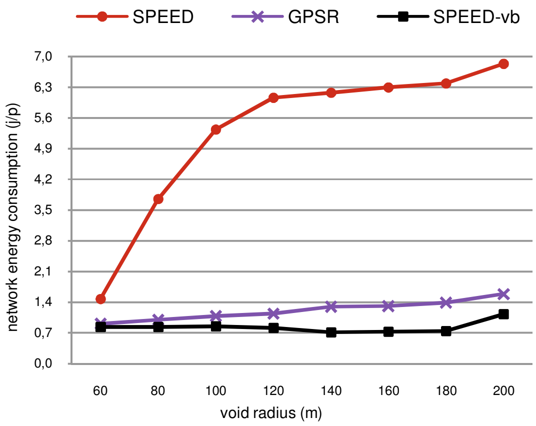


Figure 15. Network energy consumption *vs.* Void radius.

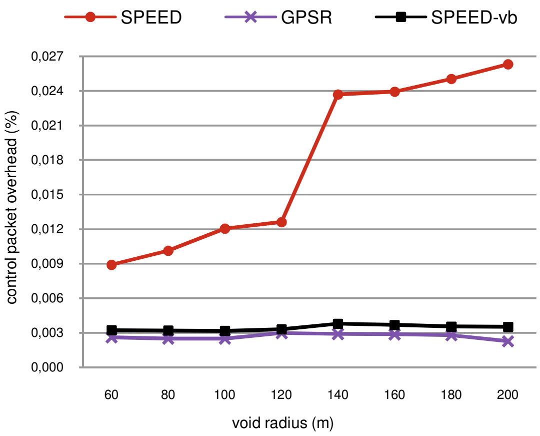


Figure 16. Control packet overhead *vs.* Void radius.

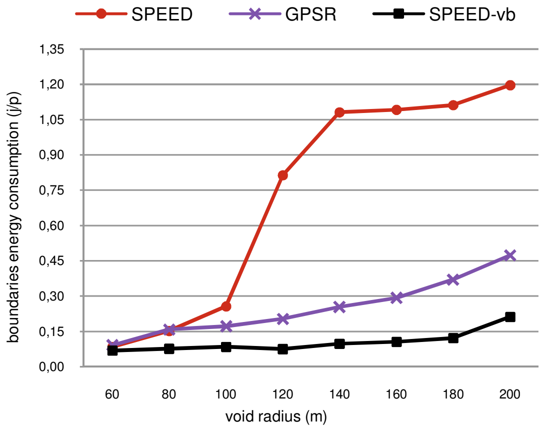


Figure 17. Boundaries energy consumption *vs.* Void radius.

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 17). For these large voids, GPSR routes most packets on the long parts of the boundary. On the other hand, our SPEED-vb protocol achieves the best tradeoff between the packet delivery ratio and the energy consumption (Figure 15). Since GPSR always uses a unique path connecting a source to the sink, it does not achieve a good node energy balancing.

2) *Performance when varying the source rate:* we use Terrain 1 (Figure 13-a) with 120m as radius of the void created at the center of the terrain and 100ms as deadline of the generated packets. Each source node generates one CBR flow with a rate increased step by step from 2 to 12 pps. For each source rate, we obtained the results shown in the figures 18-21. Thanks to the proposed void-handling mechanisms, SPEED-vb performs better than both GPSR and SPEED for all measured metrics. Indeed, Figure 18 shows that SPEED removes many data packets either by stuck nodes on the void boundary or by other nodes when the packet deadline expires. The deadline expiration is due to network congestion caused by both the backpressure messages broadcasted by sensor nodes near the void (Figure 20) and the use of alternative paths too long around this void. In SPEED, some source nodes located in concave areas of the void are permanently blocked toward the sink after receiving a backpressure message from each forwarding candidate neighbor. This further weakens the performance of SPEED in delivering packets. For few packets delivered and many backpressure messages generated, SPEED has a high energy cost (Figure 19) and consumes unnecessarily the energy of boundary nodes (Figure 21) maximizing the chances of expanding the void rapidly. As the flow generated by a source to a destination in GPSR uses the same routing path, the later will be overloaded mainly when the rate is greater than 3 pps, as shown in Figure 18. Therefore, most of these packets are delayed and then they are dropped after expiration of their deadlines. Since GPSR uses the face routing to bypass the void, it consumes more energy of boundary nodes than the proposed SPEED-vb protocol as shown in Figure 21.

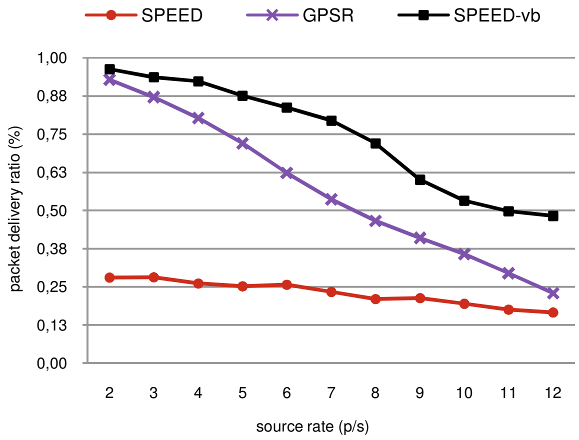


Figure 18. Packet delivery ratio *vs.* Source rate.

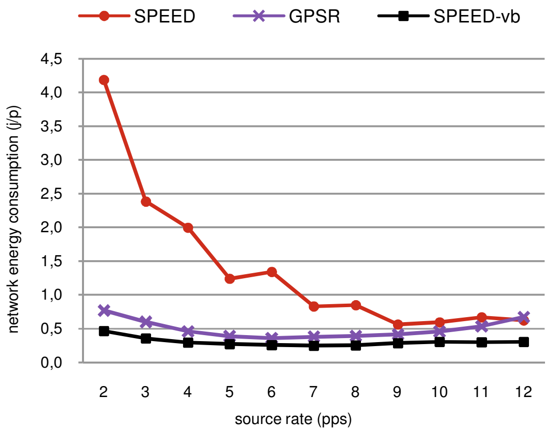


Figure 19. Network energy consumption *vs.* Source rate.

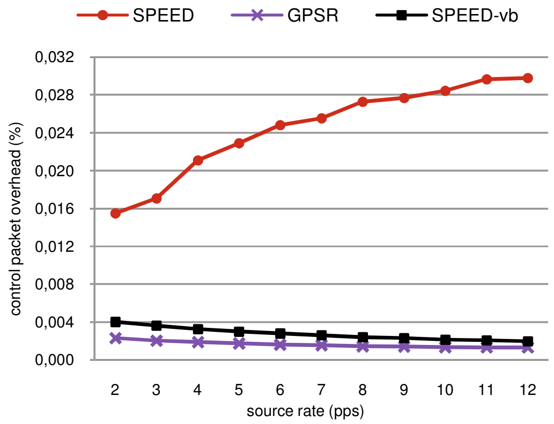


Figure 20. Control packet overhead *vs.* Source rate.

On the other hand, Figure 20 shows that the GPSR protocol generates less control packets compared to the two other evaluated protocols.

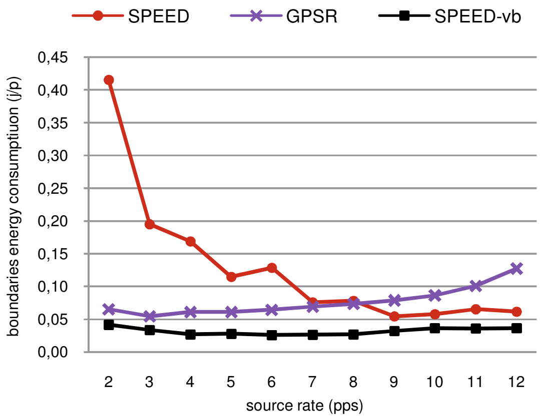


Figure 21. Boundaries energy consumption *vs.* Source rate.

3) Performance when varying packet deadline: we use Terrain 1 (Figure 13-a) with 120m as radius of the void created at the center of the terrain and 1 pps as rate of the generated packets. The packet deadline is increased step by step from 50 to 150ms and the obtained results are shown in the figures 22-25. These results show that SPEED-vb achieved the best performance compared to both GPSR and SPEED, especially for important packet deadlines. The proposed mechanisms effectively oriented many data packets around the open and closed voids and increased the links fluidity in these regions of the network. Figure 24 shows that GPSR is less efficient than other protocols in term of average gain in packet deadline, particularly when the later is less than 110ms. This is because GPSR do not balance the load between the nodes since it uses the same routing path connecting a source to a destination. For all deadlines greater than 100ms, GPSR outperforms SPEED, which generates many backpressure messages overloading nodes and unnecessarily consuming energy of boundary nodes as shown in Figure 25. But the figure proves that SPEED-vb delivers many packets in a shorter average end-to-end delay. Since it balances the load around voids and then increases the network fluidity in these areas, SPEED-vb delivers many packets (Figure 22) and saves more energy of nodes (Figure 23) compared to the protocols GPSR and SPEED. For packet deadlines less than 90ms, GPSR unnecessarily consumes the energy of the nodes because many packets are dropped in the network. These drops are due to frequent delays of packets in unique paths relating a source to a destination. Figure 22 shows that GPSR is equivalent to SPEED-vb in delivering packets when the packet deadline is greater than 100ms. This is due to the face routing applied on the void boundary which justifies the excessive energy consumption of boundary nodes in GPSR (Figure 25).

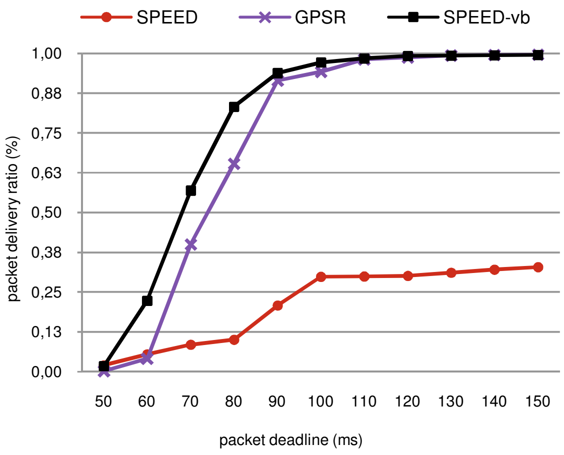


Figure 22. Packet delivery ratio *vs.* Packet deadline.

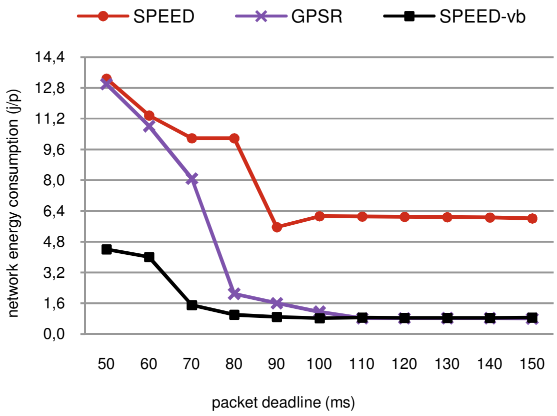


Figure 23. Network energy consumption *vs.* Packet deadline.

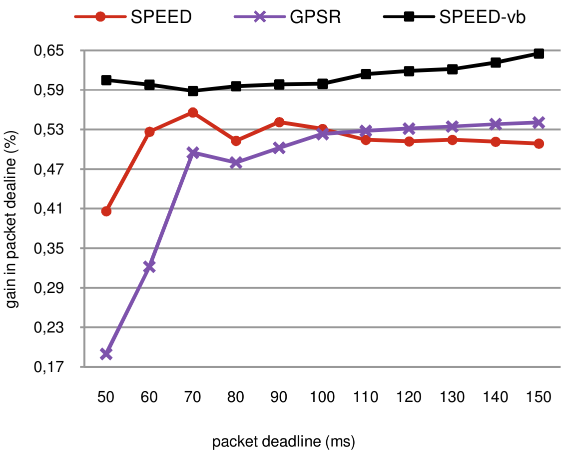


Figure 24. Gain in packet deadline *vs.* Packet deadline.

In the same time, as shown in Figure 25, the proposed void-handling mechanisms preserve more energy of boundary nodes by both using a 2-hop forwarding mode on the void boundary and orienting packets in the shortest paths around the void. Moreover, the results shown in Figure 22 show that SPEED delivers few packets and excessively consumes energy of nodes forming the network, especially when the packet deadline is less than 90ms. Indeed, many data packets are dropped in SPEED because their deadline expires in long routing paths that are also overloaded with the backpressure messages broadcasted around the voids.

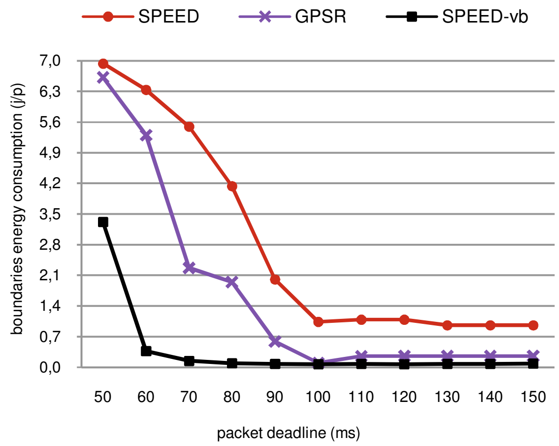


Figure 25. Boundaries energy consumption *vs.* Packet deadline.

4) Performance when varying number of voids: we use Terrain 2 (Figure 13-b) by fixing the source rate to 10 pps and the packet deadline to 100ms. We vary the number of created voids in the network from 1 to 8 and the obtained results are shown in the figures 26-29. These results show that SPEED-vb achieved the best performance compared to the GPSR and SPEED protocols. This is due to the efficiency of the proposed void-handling mechanisms. Since it uses the same routing path to deliver all packets of each flow generated by a source to a destination, the GPSR protocol delays the urgent packets because it overloads some nodes forming the used routing paths. Then many of these packets are dropped because of deadline expires (Figure 26) and energy depletion of nodes forming the routing path is then accelerated. Moreover, Figure 28 shows that the GPSR protocol have the worst node energy balancing. But the proposed SPEED-vb protocol provides the best node energy balancing thanks to its void-handling mechanism and its packet forwarding strategy inherited from the SPEED protocol. Moreover, SPEED is classed second after GPSR in term of node energy balancing as shown in Figure 28. By achieving the best node energy balancing, SPEED-vb certainly helps in extending the network lifetime. We note in Figure 26 that the number of packets delivered by SPEED is too low, especially when the number of created voids is important in the network. With several closed voids in the network, SPEED generates many backpressure packets and then accelerates the number of overloaded nodes.

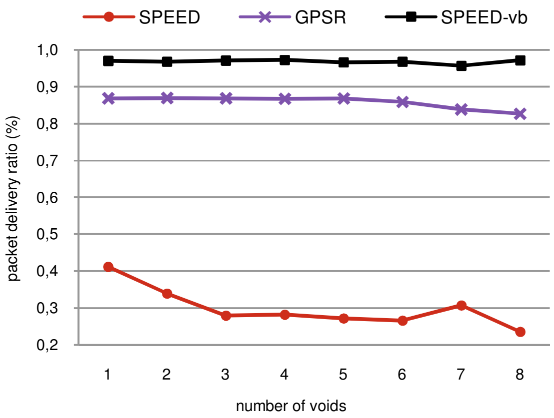


Figure 26. Packet delivery ratio *vs.* Number of voids.

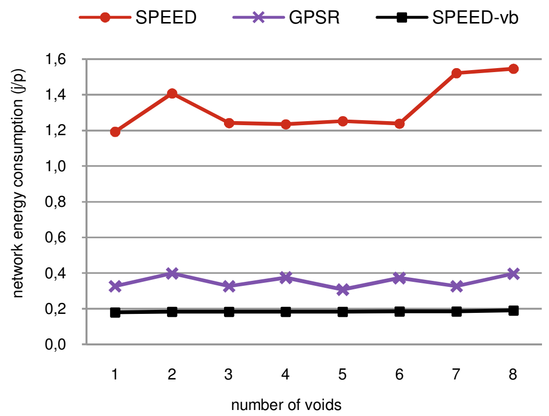


Figure 27. Network energy consumption *vs.* Number of voids.

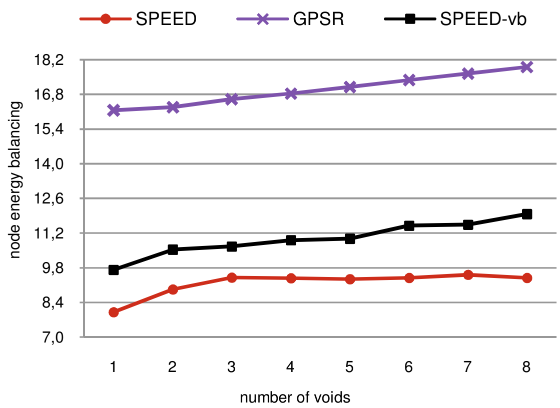


Figure 28. Node energy balancing *vs.* Number of voids.

We also notice the excessive and wasteful consumption of the network energy in SPEED as shown in Figure 27. But the same figure shows the obtained SPEED-vb protocol consumes less energy according to its packet delivery ratio. Compared to both GPSR and SPEED in Figure 29, the simulation results show that SPEED-vb has a low use of the boundary nodes when forwarding packets towards the destination node and it orients many of these packets in optimal paths near the voids in the network. Note that when preserving the energy of the boundary nodes, the voids expanse slowly and thus contribute to the best operation of the deployed application.

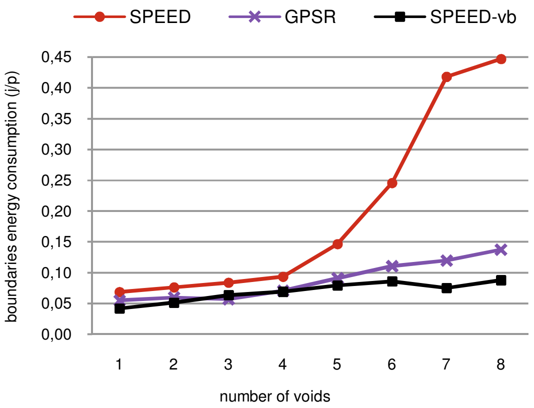


Figure 29. Boundaries energy consumption *vs.* Number of voids.

VI. CONCLUSION

We have proposed an oriented 2-hop forwarding approach that provides to each data packet received by a boundary node the shortest path towards the destination node. 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 sensor 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 wireless sensor network. 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. This comparison was done with several radius of created voids, flow rate at sources, packet deadlines and number of voids created in a same sensor network. The proposed approach, associated to the SPEED-vb protocol, was able to respond to the shortcomings of the existing void-handling techniques in terms of packet delivery ratio, control packet overhead, end-to-end delay, energy consumption and node energy balancing. Note that these techniques are based either on the right-hand rule used in the GPSR protocol or on the backpressure rule used in the SPEED protocol.

Since we are interested by sensor networks dedicated to real-time applications, our future work will focus on the sequencing of data packets at a node based on the time remaining to reach the destination node. The objective is to reduce the number of removed critical data packets due to deadline expiration. We plan to improve the proposed void-handling mechanisms by realizing a trade-off between packet delivery deadlines on short routing paths versus load balancing. We also plan to check how the propose approach can be applied to congested regions in a sensor network or to the voids created due other problems, like intermittent connectivity.

REFERENCES

[1] M. Aissani, S. Bouznad, A. Hariza, and S. E. Allia, “Oriented 2-hop Forwarding Approach on Voids Boundaries in Wireless Sensor Networks,” Proc. Eighth Advanced International Conference on Telecom. (AICT 2012), IARIA Press, May 2012, pp. 46-51.

[2] B. Karp and H. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” Proc. IEEE/ACM Conference on Mobile Computing and Networking, ACM Press, August 2000, pp. 243-254, doi: 10.1145/345910.345953.

[3] 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. 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks, ACM Press, Oct. 2007, pp. 31–38, doi:10.1145/1298239.1298246.

[4] L. Moraru, P. Leone, S. Nikoletseas, and J. Rolim, “Geographical routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks,” Proc. 7th international conference on Ad-hoc, Mobile and Wireless Networks, Springer-Verlag, Sept. 2008, pp. 148-161, doi: 10.1007/978-3-540-85209-4_12.

[5] 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), March 2008, pp. 51–62, doi:10.1109/TNET.2007.900372.

[6] 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. 5304, Dec. 2009, pp. 102–115, 10.1007/978-3-642-05434-1_11.

[7] 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), October 2006, pp. 187–200, doi: 10.1007/s11036-006-4471-y.

[8] 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. International Conference on Communications, IEEE Press, June 2008, pp. 3120-3124, doi:10.1109/ICC.2008.587.

[9] 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), August 2009, pp. 327-336, doi: 10.1145/1298239.1298246.

[10] 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), October 2005, pp. 995-1006, doi: 10.1109/TPDS.2005.116.

[11] 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), June 2006, pp. 738-754, doi: 10.1109/TMC.2006.79

[12] W. Cheng, L. Yuan, Z. Yang, and X. Du, “A real-time Routing Protocol with Constrained Equivalent Delay in Sensor Networks,” Proc. 11th IEEE Symposium on Computers and Communications, IEEE Press, June 2006, pp. 597-602, doi: 10.1109/ISCC.2006.16.

[13] L. Zhao, B. Kan, Y. Xu, and X. Li, “FT-SPEED: A Fault-Tolerant Real-Time Routing Protocol for Wireless Sensor Networks,” Proc. International Conference on Wireless Communications, Networking and Mobile Computing, IEEE Press, October 2007, pp. 2531–2534, doi: 10.1109/WICOM.2007.630.

[14] M. Aissani, A. Mellouk, N. Badache, and B. Saidani, "Oriented Void Avoidance Scheme for Real-Time Routing Protocols in Wireless Sensor Networks," Proc. IEEE Global Communications Conference, IEEE Press, December 2008, doi: 10.1109/GLOCOM.2008.ECP.24.

[15] 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), August 2010, pp. 945–962, doi: 10.1002/dac.1079.

[16] 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), August 2007, pp. 855–871, doi: 10.1016/j.adhoc.2007.02.008.

[17] Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, "The ns Manual", *http://www.isi.edu/nsnam/ns/*, accessed 15 June 2011.

[18] M. Aissani, S. Bouznad, A. Hariza, and S.E. Allia, "An effective mechanism for handling open voids in wireless sensor networks," Proc. Fifth International Conference on Sensor Technologies and Applications, IARIA Press, August 2011, pp. 24-29.