A Mechanism for Void Avoidance in Real-Time Routing oriented Medical Applications

Mohamed Aissani, Abdelhamid Mellouk, and Nadjib Badache

Abstract: **To avoid the negative impact of void areas (i.e. holes) on medical application routing efficiency, we propose a new oriented void avoidance mechanism for wireless sensor networks embedded in medical environment. To choose the forwarding region (clockwise or anticlockwise) around the void, proposed mechanism is guided by the destination location with respect to the void. Our mechanism uses the right-hand rule to discover boundary nodes of the void and geometric formulas to obtain the forwarding region of a source node near the void. This node reduces its forwarding candidate set according to its already obtained forwarding region. Proposed approach is simple to implement, economic and could incorporate various other optimizations studies. Simulation results showed the effectiveness of the proposed mechanism which gives better performance compared to traditional schemes.**

Index Terms: **communication voids, sensor networks, real-time medical application routing, avoiding voids.**

I. Introduction

Recent advances in wireless technologies have opened up new possibilities that can revolutionize traditional health care delivery. Wireless connectivity provides the infrastructure and the mobility support for ubiquitous real time patient monitoring and tracking systems for emergency response. These are few examples of how wireless technologies can be applied in medical environments, but the opportunities are countless. The development of wireless sensor networks (WSNs) can provide a ubiquitous network that can be exploited to improve health care efficiency and deliver medical services to a broader range of users.

Some applications of WSNs imply deployment in remote or hostile environments to assist in tasks such as medical monitoring. It is often necessary for the underlying communication infrastructure to meet real-time constraints. The appearance of communication voids changes global network topology, and imposes additional difficulties in organizing the network. Thus, these constraints are not be able to support medical applications with real-time for a large number of users. Therefore, the integration of adaptive routing mechanisms for avoiding voids in WSNs can be used to provide connectivity to medical and non-medical applications in scenarios such as hospitals and medical clinics, and even at home or residential care centers.

Voids can be formed either due to sensor deployment or because of failure of sensor nodes due to various reasons such as malfunctioning, battery depletion or an external event. Voids can also exist due to local minimum phenomenon often faced in geographic greedy forwarding [1].

Geographical greedy forwarding, a simple, efficient and scalable strategy, is a promising routing scheme for large-scale sensor networks when sensor locations are available [2]. In geographical greedy forwarding, a packet is forwarded to a 1-hop neighbor who is closer to the destination than the current node. This process is repeated until the packet reaches the destination, or the packet is stuck at a node whose 1-hop neighbors are all farther away from the destination. The node where a packet may get stuck is called a *local minimum* or a *stuck node*, such as node *a* in Fig. 1.

Without an appropriate void-handling technique in place, some data packets may get lost in the network, wasting precious network resources as well as disabling communications between some pairs of nodes. In particular, such network behavior is highly undesirable in mission-critical wireless networks such as sensor networks, because a few failures to detect critical events may defeat the whole mission of sensor network applications [3].

Existing void-handling techniques can be classified into two principal categories: *right-hand rule* [2, 4, 5] and *backpressure rule* [6, 7]. However, some recent protocols, also based on the right-hand rule, bypass packets from both sides of the void [8, 9]. According to the right-hand rule, data packets tend to be routed along the boundaries of voids. The probability that the void boundary nodes are shared by several communication sessions is very high than other nodes. Thus excessive energy consumption and data collisions may occur in these nodes. According to the back pressure rule, data packets tend to be pushed back to upstream node and attempt to find another route to destination. For real-time protocols, backpressure rule violates packet desired progression speed toward the destination and many packets are dropped because routes are longer and then deadlines of packets are expired.

In this paper, we consider real-time routing approaches that are potential candidates for supporting medical applications in integrated sensor networks environment. For this, we focus on void avoiding problem in routing function for sensor networks. We present a new mechanism to solve the void problem faced by geographic forwarding mechanism in sensor networks for medical real-time applications. Our approach consists of two steps: *void discovery* and *forwarding neighbors selection*. The first step uses the right-hand rule to detect locations of the boundary nodes and estimates the center of the void. The second step uses geometric formulas to obtain the forwarding

Manuscript received Jun and revised August, 2008.

M. Aissani and A. Mellouk are with LISSI Laboratory - Paris 12 University, Vitry-sur-Seine, France.

N. Badache and M. Aissani are with USTHB University, Algiers, Algeria (e-mail: {aissani, mellouk}@univ-paris12.fr and {maissani@gmail.com, badache@mail.cerist.dz})

1845-6421/08/8144 © 2008 CCIS

region (clockwise or anticlockwise) of a source node near the void, according to locations of the destination node, the center of the void and the source node. Then the sender node reduces its forwarding candidate set according to its earlier computed forwarding region. Our mechanism is simple to implement, saves network resources and could be incorporated in various real-time routing protocols for medical applications.

Note that to select its forwarding candidate nodes from a computed region, a sender node s uses the straight-line $\overline{sv}$ as shown in Fig. 7, but in our previous work [10], node s used the straight-line $\overline{sd}$. The two techniques are compared by simulation in this paper (see Section V).

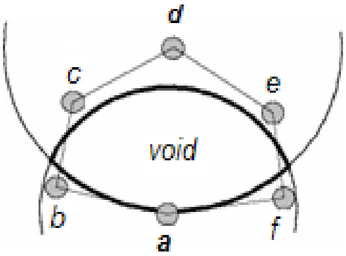


Fig. 1. Greedy forwarding failure (a local minimum).

The remainder of this paper is organized as follows. In Section II, we summarized the exiting real-time routing protocols that handle the void problem in wireless sensor networks and potentially used for medical environments. Void handling schemes used in SPEED [6] and FT-SPEED [8] protocols are discussed in Section III. Section IV elaborates proposed solution to handle void areas in wireless sensor networks for medical applications. Section V presents simulation results of the proposed mechanism associated to the protocol SPEED. Section VI concludes the paper and presents our future work.

II. Related Works

The majority of existing real-time medical routing applications protocols doesn't explicitly address the problem of voids in sensor networks. However, some protocols use specific rerouting mechanisms avoiding drops to packets by stuck nodes on borders of voids [5, 6, 8, 11, 12].

RAP [5] is one of the first real-time routing protocols in sensor networks. It provides service differentiation in the timeliness domain by velocity-monotonic classification of packets. It uses the right-hand rule to route packets around the perimeter of void areas. However, it works only when most traffic is periodic and all periods are known a priori. Also, it is not adaptive to dynamics of sensor networks. SPEED [6] is designed to provide soft end-to-end deadline guarantees for real-time packets in sensor networks. It ensures a network wide speed of packet delivery for real-time guarantee. It handles voids in the same way as it handles congested areas (see Fig. 2). It has been admitted in SPEED that the void avoidance scheme is not guaranteed to find a path if there is one as in GPSR [4], but it is guaranteed to find a greedy path if one exists. RPAR [11] integrates power control and real-time routing for supporting energy-efficient real-time communication. It combines feedback control and nondeterministic QoS-aware geographic forwarding. It incorporates face routing mechanisms [13] to route packets around large voids in the network. MM-SPEED [12] extends SPEED to support different delivery velocities and levels of reliability. It provides QoS differentiation in two quality domains, namely, timeliness and reliability, so that packets can choose the most proper combination of service options depending on their timeliness and reliability requirements. When voids occur in network, MM-SPEED uses the SPEED avoid avoidance scheme to reroute packets at stuck nodes (see Fig. 2). FT-SPEED [8] borrows the idea of the protocol SPEED to handle the real time packet delivery in greedy forwarding. It has the same components with SPEED except the void avoidance scheme. FT-SPEED proposes an alternative scheme to resolve the local minimum problem. To reach their destinations, packets at suck nodes are delivered using only void boundary nodes (see Fig. 3).

Because our proposal has a relationship with both SPEED and FT-SPEED void-handling schemes, we discuss them in the next section.

III. Void-Handling Techniques in SPEED and FT-SPEED Protocols

SPEED handles rerouting around a void in the same way as it handles congestion. As shown in Fig. 2, when a packet arrives at the stuck node 2, it will then drop it and send a backpressure beacon to node 1, and node 1 will stop sending packets to node 2. If node 3 doesn't exist, other packets will be dropped and further backpressure will occur until a new route is found.

However, when node 1 receives the backpressure beacon form node 2, SPEED will choose another node which satisfied the speed requirement as its next hop. In this case, node 3 can select the node 2 as its next hop. As described above, all packets received by node 2 will be dropped and node 1 has no idea about this. The scarce network energy resources are drained away for nothing, and the network lifetime is decreased. Also, it has been admitted in SPEED that the void avoidance scheme is not guaranteed to find a path if there is one as in GPSR [4], but it is guaranteed to find a greedy path if one exists.

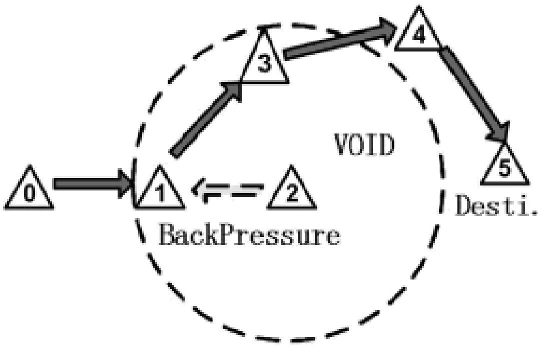


Fig. 2. Void avoidance scheme of SPEED [6]

FT-SPEED has the same components with SPEED except the void avoidance scheme. In FT-SPEED, when a first data packet for target reaches the stuck node, a void location and identification step will start. The received packet is sent to the following void edge node by the famous void finding algorithm BOUNDHOLE [2] in the anticlockwise order. When the packet reaches a nearer node, it could be sent with greedy forwarding mechanism again. Then when void edge node receives a packet not from the boundary nodes, it will bypass the void on both sides to deliver the packet to the destination node (see Fig. 3). FT-SPEED also introduces the periodical maintenance method to update the void information. The stuck node will periodically redo the void location and identification step by sending a control packet around the discovered void. Then if there are any changes about the void, all of the edge node will know the change about its edge neighbors soon [8].

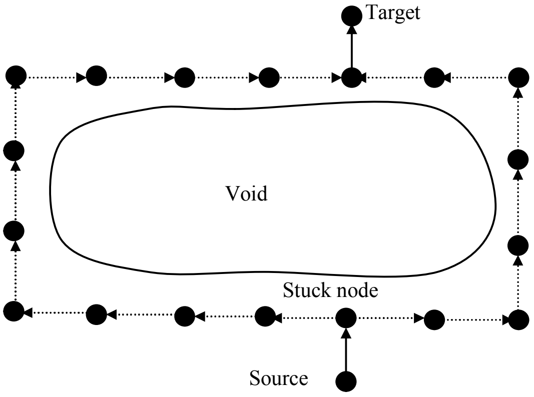


Fig. 3. Example of void bypass in FT-SPEED [8]

FT-SPEED encounters three problems too. First, when data packets reach the stuck node, FT-SPEED will no longer take the speed requirement as the criteria to choose the next hop node. Secondly, for a same data packet, FT-SPEED uses a two sides bypassing void method that consumes network resources uselessly (see Fig. 3). Finally, because FT-SPEED uses the void edge nodes to deliver data packets and control packets to maintain the void fresh information, data collisions may occur in these nodes and the battery of these nodes will be depleted soon.

IV. Oriented Void Avoiding Mechanism

To resolve the above-mentioned problems of SPEED and FT-SPEED protocols when meeting voids (see Section III), we propose an oriented void avoidance. It takes into account the locations of the destination (i.e. target), the center of the void and the source node. We assume in this work that each node can get the location information of itself, either by GPS or other location services [14], and location information of its 1-hop neighbors by a periodic beaconing [6]. We consider the topologies where the wireless nodes are roughly in a plane and all sensor nodes can move slowly. Finally, we assume that a source node marks packets with their destination's location. Thus, a location registration and lookup service that maps node ID to location [11] is required.

Our void avoiding mechanism consists of two steps: void discovery and forwarding neighbors selection.

A. Void Discovery Step

A node b_0 which firstly detects the void (i.e., its forwarding candidate set is empty), when receiving packet to the destination node, broadcasts a void-beacon packet to its 1-hop neighbors notifying them that it can't forward any more. The received data packet is dropped to guarantee a positive delivery speed as in [6]. Then node b_0 initiates a Void Discovery (VD) packet marked with its ID and forwards it to the void boundary node b_1 by the right-hand rule. Node b_1 inserts its location information into the received VD packet and forwards it to node b_2 by right-hand rule and so on. This process is repeated until the VD packet has traveled around the void and eventually been received by the initiator node b_0, as shown in Fig. 4. Node b_0 extracts the location information of all boundary nodes of the void from the received VD packet. Then node b_0 selects two nodes $b_i(x_i, y_i)$ and $b_j(x_j, y_j)$ from the set of boundary nodes $\{b_0, b_1 \ldots b_n\}$ so that the distance between b_i and b_j is the longest distance among the distances between any two nodes in $\{b_0, b_1 \ldots b_n\}$. Then node b_0 calculates the midpoint $v(x_v, y_v)$ of the line-segment $\overline{b_i b_j}$ (see Fig. 5) as follows:

$$x_v = (x_i + x_j)/2 \quad \text{and} \quad y_v = (y_i + y_j)/2 \tag{1}$$

To communicate v's coordinates to all boundary nodes, node b_0 initiates a second travel around the void. Then b_0 is a responsible node of the void. Note that v is an estimated point of the void center with respect to the network chip origin.

Due to the dynamics of sensor networks, a void may enlarge or change shape during the lifetime of the network. Therefore, responsible node b_0 will periodically redo the void discovery step. Then if there are any changes about the void, all boundary nodes will know changes about its edge neighbor nodes.

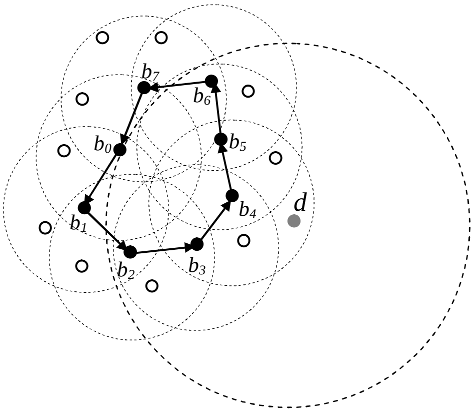


Fig.4. A travel of a void discovery packet

Expire-time is used in a proposed mechanism to timeout coordinates of v in all sensor nodes. If a boundary node is not refreshed after a certain timeout, its v's coordinates will be reinitialized. Note that v's coordinates in each sensor node are initialized with (0,0) values at the deployment of the wireless sensor network.

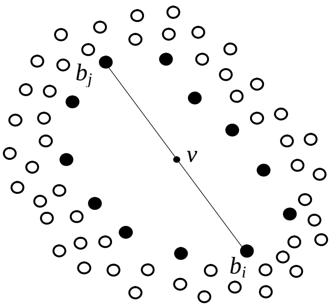


Fig. 5. Boundary nodes and void center estimation

B. Forwarding Neighbors Selection Step

Assume that our mechanism is used by the protocols SPEED [6] and MMSPEED [12]. As shown in Fig. 7, these protocols define $FS_s(d)$, i.e. the forwarding candidate set of source node s to reach destination node d, as the set of nodes that belong to NS_s and closer to destination node d. Where, NS_s is defined as the set of nodes within the radio range of node s.

When node s receives a packet for destination d, it will check and scan its $FS_s(d)$. If a boundary node is belonging to the $FS_s(d)$, i.e. v's coordinates of this node are not null, node s will execute the following rules:

- Node s extracts v's coordinates from its neighbor table and computes the forwarding orientation θ_{ds} (shown in Fig. 6) by using the formula (2). Then node s removes all boundary nodes from its $FS_s(d)$. Note that to make θ_{ds} always positive, we increment it by 2π if it is negative.
- If $\theta_{ds} \leq \pi$, node s selects its forwarding neighbors from the clockwise region. Thus, node s removes from its $FS_s(d)$ all nodes verifying either the inequality (6) if $x_s \leq x_v$ or the inequality (7) if $x_s > x_v$.
- If $\theta_{ds} > \pi$, node s selects its forwarding neighbors from the anticlockwise region. Thus, node s removes from its $FS_s(d)$ all nodes verifying either the inequality (7) if $x_s \leq x_v$ or the inequality (6) if $x_s > x_v$.
- If $FS_s(d)$ is empty, node s drops the data packet and sends a *void-beacon* (with v's coordinates) to the upstream node which will direct further packets in the appropriate forwarding region. Note that data packets are dropped to guarantee a positive delivery speed required by SPEED [6]. In other protocols, node s can push back the packet to upstream node which will try to find another route to destination.

In the example showed in Fig. 7, source node s has 6 forwarding candidate nodes. It extracts v's coordinates from the single boundary node, calculates θ_{ds} and then removes the single boundary node from its $FS_s(d)$. In this example, we have $\theta_{ds} > \pi$, so anticlockwise forwarding. Thus node s removes from its $FS_s(d)$ 2 other neighbor nodes located in the clockwise region. Finally, node s has reduced its $FS_s(d)$ to 3 forwarding neighbor nodes and they are located in the anticlockwise region (i.e. the three hatched nodes in Fig. 7).

As shown in Fig. 6, the forwarding orientation θ_{ds} for node s is obtained by the following formula:

$$\theta_{ds} = \theta_s - \theta_d \tag{2}$$

where the angles θ_s and θ_d are obtained by:

$$\theta_s = atan2(y_s - y_v, x_s - x_v)$$

$$\theta_d = atan2(y_d - y_v, x_d - x_v)$$

The function $atan2(y, x)$ is a C++ function that returns $atan(y/x)$ with respect to the quadrant where a node, having the coordinates x and y, is situated, such as:

$$atan2(y,x) = \begin{cases} atan(y/x) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ atan(y/x) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ atan(y/x) & \text{if } x > 0 \\ \pi/2 & \text{if } x = 0 \text{ and } y > 0 \\ -\pi/2 & \text{if } x = 0 \text{ and } y < 0 \end{cases}$$

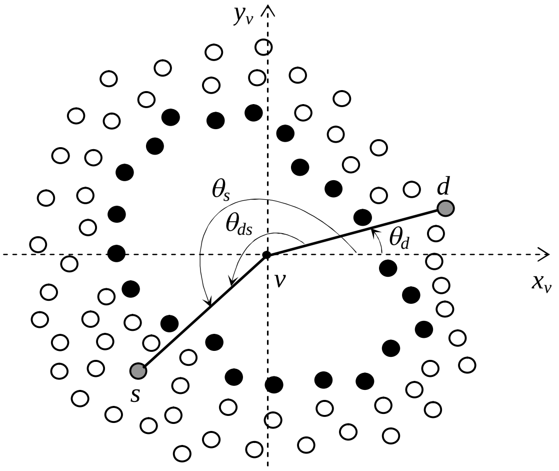


Fig. 6. An oriented geographic forwarding for avoiding voids in sensor networks

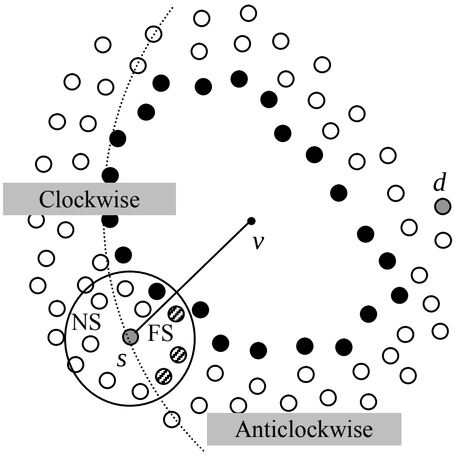


Fig. 7. Example of NS and FS in geographic forwarding

In Fig. 7, we notice that the forwarding regions (clockwise or anticlockwise) of node *s* are separated by the straight-line $\overline{sv}$ defined by the linear equation (3). Where m is the slope, given by the formula (4), and c the y-intercept, given by the formula (5), of the straight-line.

$$y = m \cdot x + c \qquad (3)$$

$$m = (y_v - y_s)/(x_v - x_s) \qquad (4)$$

Putting *v* and *m* in (3), we obtain the value of *c* by:

$$c = y_v - (y_v - y_s)/(x_v - x_s) \cdot x_v \qquad (5)$$

A node *i* belonging to the $FS_s(d)$, with coordinates x_i and y_i, is located in a clockwise forwarding region if it verifies either the inequality (6) in the case where $x_s > x_v$ (i.e. node i is located in the lower area of the line $\overline{sv}$), or the inequality (7) in the case where $x_s \leq x_v$ (i.e. node i is located in the above area of the line $\overline{sv}$). On the other hand, the node *i* is located in the anticlockwise forwarding region if it verifies either the inequality (7) when $x_s > x_v$ or the inequality (6) when $x_s \leq x_v$.

$$y_i < (m \cdot x_i + c) \qquad (6)$$

$$y_i > (m \cdot x_i + c) \qquad (7)$$

Where, parameters m and c are given by the formulas (4) and (5) respectively.

V. Simulation Results

We have implemented and conducted extensive simulations of SPEED protocol by using ns2 network simulator [15]. Then we associated the proposed mechanism to SPEED and the resulting protocol, called SPEED-sv, is compared by simulation to SPEED and SPEED-sd protocols (SPEED-sd is the protocol SPEED-1 proposed in our previous work [10]).

Note that SPEED-sv has the same components with SPEED-sd except the forwarding neighbors selection step. To select its forwarding candidate nodes from a computed region, a sender node *s* in SPEED-sd uses the straight-line $\overline{sd}$ (not the straight-line $\overline{sv}$ like in SPEED-sv as shown in Fig. 7), which is defined by the following linear equation:

$$y = m \cdot x + c$$

where,

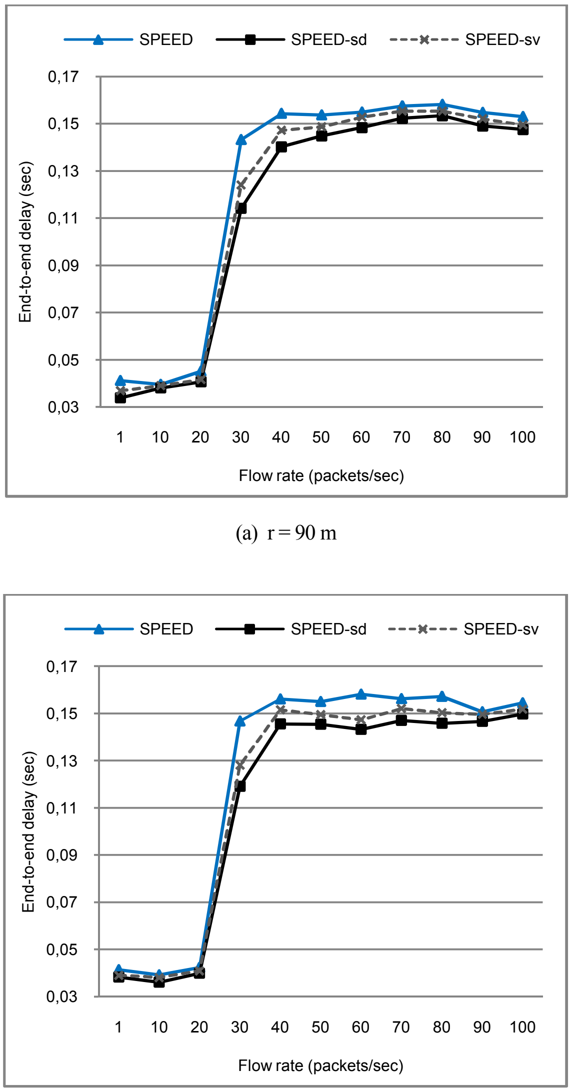
$$m = (y_d - y_s)/(x_d - x_s)$$

$$c = y_d - (y_d - y_s)/(x_d - x_s) \cdot x_d$$

The simulation scenario is in a squared area with some stationary sensor nodes placed uniformly. A void is created at the center of the area; i.e. an area is set with no sensor nodes inside to simulate a void in realistic environments. The sink is at the right side of the void and the source node at the left. To create several network loads, the flow rate of a source node is gradually increased until reaching 100 packets/sec. Table I gives the parameters settings used in the simulations.

TABLE I
SIMULATION ENVIRONMENT SETTINGS

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



(a) r = 90 m

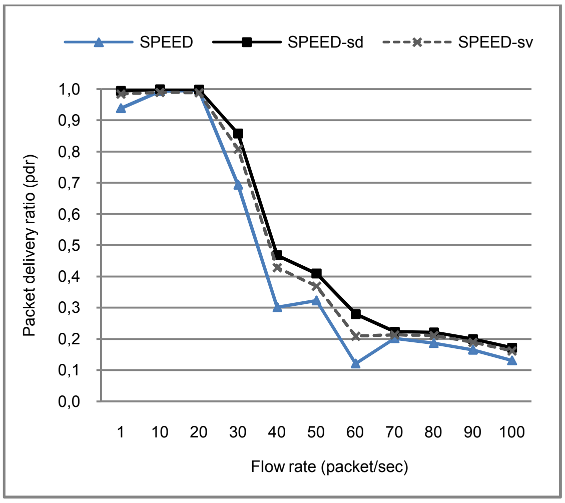
(b) r = 113 m

Fig. 8. Average end-to-end delay (e2e).

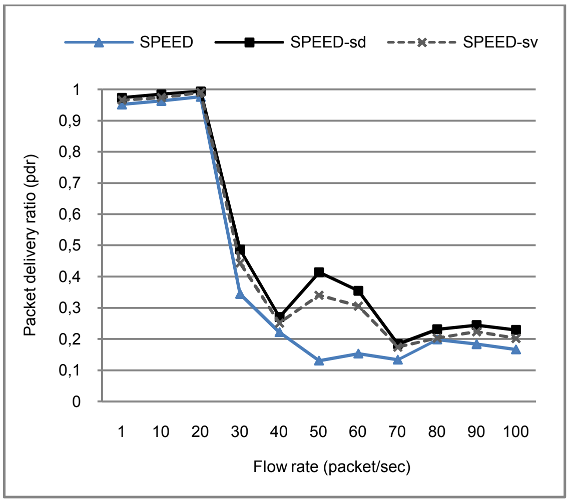
As shown in Fig. 8a (with void radius *r* equal to 90 meters) and Fig. 8b (with void radius *r* equal to 113 meters), the average end-to-end delay of each simulated protocol increases when the flow rate is increased, but the proposed protocols

SPEED-sd and SPEED-sv outperforms the original SPEED, particularly when the flow rate exceeds 20 packets/second. In the same time, we observe that SPEED-sd is more efficient than SPEED-sv because its selection step of forwarding neighbors, which is based on the straight-line $\overline{sd}$, converges more quickly towards a destination node d. This observation is also valid for the packet delivery ratio (see Fig. 9a and Fig. 9b) and the average number of hops forming a routing path (see Fig. 10a and Fig. 10b) of these protocols.

Compared to the protocol SPEED in Fig. 9a and Fig. 9b, the protocols SPEED-sd and SPEED-sv provide a better average packet delivery ratio, especially when the flow rate is greater than 20 packets/second. The number of packets dropped by boundary nodes is minimized by SPEED-sd and SPEED-sv protocols. We notice that SPEED-sd is more powerful than SPEED-sv because of its manner of selecting its forwarding candidate nodes.



(a) r = 90 m

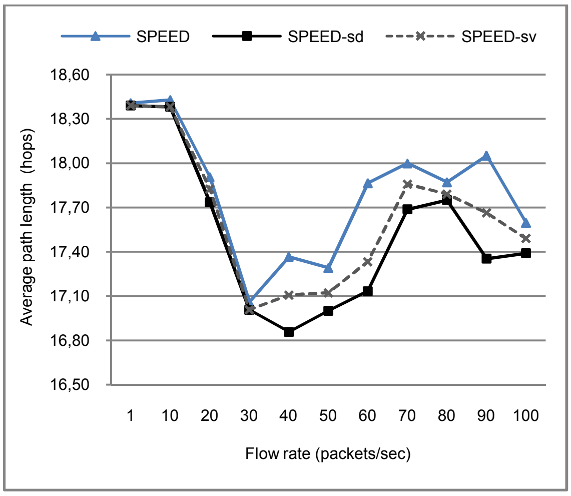


(b) r = 113 m

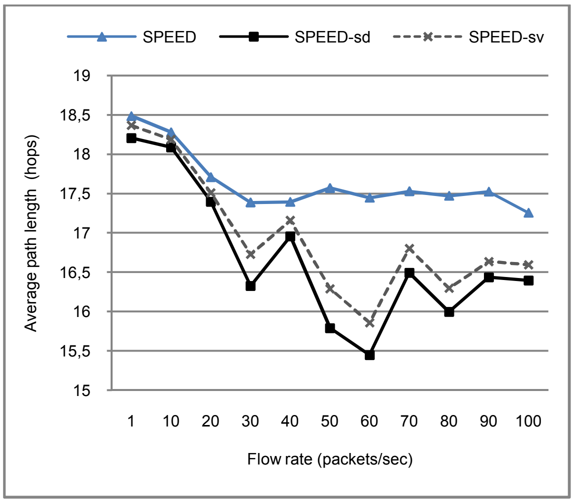
Fig. 9. Packet delivery ratio (pdr).

Fig. 10a and Fig. 10b show a good performance of the protocols SPEED-sd and SPEED-sv in term of routing path length (average number of hops) compared to the protocol SPEED, especially when the flow rate is greater than 25 packets/second. Thanks to our void avoiding mechanism, SPEED-sd and SPEED-sv reduce the number of end-to-end hops when routing packets around a discovered void.

The good performance of SPEED-sd and SPEED-sv protocols can be improved in our future works. During conducted simulations, we have observed that some data packets are dropped because of full queues of nodes inside a forwarding region. So, we can envisage a threshold of filling of these queues. When this threshold is reached, a sender node will direct the packets in the opposite forwarding region.



(a) r = 90 m



(b) r = 113 m

Fig. 10. Average path length (number of hops).

VI. Conclusion and Future Work

The existing void-handling schemes based on the right-hand rule or the backpressure rule still have problems. If multiple communication sessions need to bypass the void simultaneously, the probability is very high that the nodes on the boundary of the void are shared by several communication sessions by the right-hand rule. Thus data collisions may occur in these nodes and the battery of these nodes will be depleted soon. Also, for medical real-time routing protocols, backpressure rule violates desired speed progression toward the destination and many packets are dropped because routes are longer and packet deadlines are expired.

To solve the void problems faced by geographic forwarding mechanisms in sensor medical networks, we proposed an alternative mechanism, simple to implement, saves network resources and could incorporate various real-time routing protocols for medical applications. A merit of our approach is that a data packet is sent only in the appropriate forwarding region according to its destination location with respect to both the void and source node. In contrast to existing void avoiding techniques, boundary nodes in our mechanism don't deliver many packets; they just maintain the void fresh information in order to save their energy.

Our future work is to develop other estimating methods taking in account complex forms of voids in wireless sensor networks. Resulting void avoidance mechanism will be associated to other existing real-time routing protocols and then extensive simulations will be conducted.

REFERENCES

[1] N. Ahmed, S. Kanhere and S. Jha, "The Holes Problem in Wireless Sensor Networks: A survey," ACM Sigmobile Mobile Computing and Communications Review, vol. 9, Issue 2, pp.4-18, Apr., 2005.

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

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

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

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

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

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

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

[9] Z. Wu, S. Li, "An Adaptive Multi-paths Algorithm for Wireless Sensor Networks," ICESS 2007, LNCS 4523, pp. 686-698, 2007.

[10] M. Aissani, A. Mellouk, N. Badache and Brahim Saidani, "Oriented Void Avoidance Scheme for Real-Time Routing Protocols in Wireless Sensor Networks," IEEE Global Communications Conference, New Orleans, LA, USA, 30 November-4 December 2008.

[11] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J.A. Stankovic and T.F. Abdelzaher, "Real-time Power-Aware Routing in Sensor Networks," IEEE International Workshop on Quality of Service (IWQoS'06), June 2006.

[12] 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, v.5 n.6, p.738-754, June 2006.

[13] Y.J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical, " In Proc. of USENIX/ACM Symposium on Networked System Design and Implementation (NSDI), May 2005.

[14] S. Ray, W. Lai and I.C. Paschalidis, "Statistical location detection with sensor networks". IEEE Transactions on Information Theory 52 52(6): 2670-268, 2006.

[15] Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. *The ns Manual*, http://www.isi.edu/nsnam/ns/, 2005.

Mohamed AISSANI is a Ph.D. candidate in Computer Science at Paris 12 University (France) and USTHB University (Algeria). Since September, 2007 he is head of LAI (Laboratory of Artificial Intelligence) at Polytechnic School of Algiers (EMP). He received his Diploma-Engineer in Computer Science from EMP School in 1992 and Diploma-Master of Sciences from USTHB University in 2001. His research interests include artificial intelligence, mobile ad-hoc networks and wireless sensor networks.

Abdelhamid MELLOUK (*mellouk@univ-paris12.fr*) (Senior Member IEEE). Head (1999-2007) of Networks and Telecommunication (N&T) Department of Paris 12-Val de Marne University, France, He graduated in Computer Network Engineering from the University of Paris Sud XI Orsay (with distinction) and received his PhD in informatics Engineering from the same university and the Doctorat of Sciences Diploma (Habilitation) in Networks Telecommunication from the University Paris 12. Abdelhamid MELLOUK was also founder of Licence level Diploma in Network & telecommunication, Responsible of the Professional Master in Networks Technology (IUP Paris 12), Vice-President of French National Council of Head of Network Departments and member of LMD (Licence-Master-Doctorat) sciences commission of University of Paris 12. He is currently a member of Scientific University Council and member of National French Teaching Commission for Network and Telecommunication specialities. His general area of research is in high-speed wired/wireless networking, switching and routing, quality of service, core networks and adaptive contextual services. Currently, he is working on multimedia and high-speed communications, routing and switching optimization in dynamic traffic networks; human and bio-inspired artificial intelligence approaches, real-time application; traffic engineering and shaping in wired, wireless, sensor and mobile communications. He investigates particularly the use of artificial neuronal intelligence together with biologically inspired techniques such as reinforcement learning, to control network behavior in real-time so as to provide users with the quality of service that they request, and to improve network provide robustness and resilience. He is Senior Member of IEEE, member of IEEE Communications Society and IEEE Computer Society. He has authored, co-authored or coordinated two books (Hermes Lavoisier and Wiley/ISTE Editors) and more than 80 refereed international publications in journals, conferences and books. He served or is serving also as International Steering Committee member, Co-Chair, Track Chair, or Technical Program Committee member of several major international conferences, including IEEE ICC, GlobeCom, LCN, Networking, ICN, MSWIM, ICIW, ACM SAC, etc. He served or is serving as Associate Editor, Editorial Board, Guest Editor, or Reviewer for *Wiley's Journal of Security and Communication Networks*, *ITSSA Journal, International Journal of Internet Protocol Technology, IEEE Communications Magazine, IEEE Wireless Communications Magazine, Wiley's International Journal of Communication Systems, Wiley Computer Communications*, etc. In particular during the last two years, he was a guest editor-in-chief of two special issues: *IEEE 2007 Communications magazine* and 2008 *Springer-Annals of telecommunications*. He has/is chaired (co-) more than 10 international conferences and symposiums including IEEE GLOBECOM'08 Communications Software and Services Symposium and The IEEE ICC'09 Wireless Networking Symposium.

Nadjib BADACHE was with CISTTT (Centre d'Information Scientifique et Technique et de Transfert Technologique) now CERIST (Centre de Recherche sur l'Information Scientifique et Technique) at Algiers, from 1978 to 1983, working on applied research projects in information retrieval, operating systems and library management. In 1983, he joined USTHB University of Algiers, as assistant professor and then professor, where he teached operating systems design, distributed systems and networking with research mainly in distributed algorithms and mobile systems. From 2000 to 2008, he was head of LSI (Laboratoire des Systèmes Informatiques of USTHB University) where he conducted many projects on routing protocols, energy efficiency and security in Mobile Ad-hoc Networks and Wireless Sensor Networks. Since March, 2008 he is director of CERIST and professor at USTHB University.