

Intertwined Localization and Error-Resilient Geographic Routing for Mobile Wireless Sensor Networks

Imane Benkhelifa^{1,2} • Samira Moussaoui¹ • Ilker Demirkol³

Abstract Geographic routing in wireless sensor networks brings numerous inherent advantages, albeit its performance relying heavily on accurate node locations. In mobile networks, localization of the continuously moving nodes is a challenging task and location errors are inevitable and affect considerably routing decisions. Our proposal is in response to the unrealistic assumption widely made by previous geographic routing protocols that the accurate location of mobile nodes can be obtained at any time. Such idealized assumption results in underperforming or infeasible routing protocols for the real world applications. In this paper, we propose INTEGER, a localization method intertwined with a new location-error-resilient geographic routing specifically designed for mobile sensor networks even when these networks are intermittently connected. By combining the localization phase with the geographic routing process, INTEGER can select a relay node based on nodes' mobility predictions from the localization phase. Results show that INTEGER improves the efficiency of the routing by increasing the packet delivery ratio and by reducing the energy consumption while minimizing the number of relay nodes compared to six prevalent protocols from the literature.

Keywords Geographic Routing • Localization • Location-Prediction • Location Errors • Mobility Management • Delay-Tolerant Networks

✉ Imane Benkhelifa
ibenkhelifa@usthb.dz, i.benkhelifa@cerist.dz

Samira Moussaoui
smoussaoui@usthb.dz

Ilker Demirkol
ilker.demirkol@entel.upc.edu

¹ USTHB University of Science and Technology- Department of Computer Science, Algiers, Algeria

² CERIST Research Center, Algiers, Algeria

³ Universitat Politècnica de Catalunya- Department of Network Engineering, Barcelona, Spain

1 Introduction

In an application such as monitoring wild animals [19], localizing continuously moving animals and relaying the sensory information to a central system are one of the most important tasks necessary to understand the behavior of animals. In such applications, the sensed data is delay-tolerant, since collected data is generally to be studied and analyzed afterward, even though a reduced delay is preferable. Energy-efficient solutions is paramount for sensor nodes since catching wild animals in the purpose of recharging their sensor batteries is not an easy task and not practical. An application such as weather monitoring [21] is also a good showcase for which geographic routing can be employed. The most important data to be sent to forecast centers are the sensed time-stamped weather conditions and their locations. Volunteer vehicles, buses, bikes can be exploited to play the role of relay nodes especially those going to regions with low node densities. The unpredicted mobility of these nodes may result in an intermittently connected network that should not be neglected.

To date, there have been extensive studies on localization methods and on geographic routing algorithms for wireless sensor networks (WSNs), albeit mostly not covering both topics at the same time. Proposed geographic routing protocols often assume that nodes' positions are known either using Global Positioning System (GPS) or other localization methods and neglecting the possible localization errors of these approaches. In addition, the majority of the proposed algorithms target static networks, yet only few of them consider only the mobility of the sink [1, 2]. There have been few studies assuming the mobility of sensor nodes and only 9% of the surveyed routing algorithms in [3] consider the total mobility of nodes. Nevertheless, with the advent of Internet of Things (IoT) [20], mobile wireless sensor networks play an important role in this new technology, and in applications such smart cities, objects can be mobile, totally or partially, with regular or random mobility. Hence, there is a need to design localization methods and routing algorithms specific to such mobile sensor networks.

Geographic routing is a promising candidate in the mobile WSN field [4, 5] due to its real benefits such as its stateless nature, its distributed execution, and its low computational and storage resource requirements [6, 7]. It achieves high scalability due to its reasonable memory requirements since there is no need for a node to know beyond its neighbors [8]. In mobile WSN, topology changes frequently hence a fast route discovery and repair is required, which is enabled by geographic routing using only local topology information [3], [8]. In this way, it also saves considerable energy consumption and memory [9]. However, geographic routing relies strongly on location information and without prior knowledge of this information, nodes cannot decide about their next forwarders. In mobile networks, location information becomes increasingly imprecise because of the frequent change of nodes' positions, for which, a number of localization methods were proposed for mobile sensor networks [10-15]. Yet, all these localization methods give only estimations and errors are inevitable. Without considering these location errors in geographic routing, routing decisions can be totally erroneous thus leading to a high rate of packet loss. In addition, the mobility of nodes can cause intermittent connectivity that affects significantly the communication characteristics in the network [16]. Therefore, novel communication and routing techniques dealing with this lack of reliability are required.

In this paper, we propose a new approach to the problem of routing in mobile networks in the presence of location errors. Specifically, we propose an INTertwined localization and Error-resilient GEographic Routing (INTEGER) protocol, which combines mobility-prediction based localization, on-demand neighborhood discovery, relay weighted-selection and delay-tolerant geographic routing. The protocol is composed of two intertwined algorithms. Speed and Direction Prediction-based Localization (SDPL) for localizing mobile nodes which is an improvement of our works published in [17, 18] that gives an estimate of the location error bound to be considered in the routing phase. The other algorithm is a new geographic routing that uses on-demand neighborhood discovery and exploits the results of the improved localization algorithm to route packets considering the location-error, the mobility of nodes and the intermittence of the connections. To the best of authors' knowledge, INTEGER is the first method to include a joint prediction-based localization and geographic routing while considering the location error and the mobility of all nodes. Extensive ns2-based simulation experiments are performed to demonstrate how INTEGER deals with location errors of mobile nodes. We have compared the performance of INTEGER to that of six routing protocols from the literature under different network scenarios and parameter settings. The results show that when varying the speed of nodes INTEGER improves the energy efficiency by 33%, increases the packet delivery ratio by 24% and reduces the number of relay nodes by 42% while maintaining a reduced delivery delay.

This paper is organized as follows: Section 2 reviews the state-of-the-art of geographic routing for wireless sensor network, specifically those considering localization errors. Examples of the effect of nodes' mobility and location-errors are presented in Section 3. INTEGER is described in detail in Section 4.1 (Localization method) and Section 4.2 (Routing algorithm). Section 5 shows the performance evaluation of INTEGER under different network scenarios and a demonstration of the effectiveness of INTEGER in efficiently handling the localization errors under high mobility of nodes. Possible adaptation of INTEGER in Delay-tolerant networks is studied in Section 6. Finally, conclusion and future work are provided in Section 7.

2 Related Work

2.1 Works considering localization error

A number of works has investigated the effect of location errors on routing protocols but mostly for static networks.

The study presented in [22] and its improvement in [23] provides a detailed analysis of the effects of location errors on the correctness and performance of geographic routing in sensor networks. The analysis identified *information hiding* (when one of the nodes cannot get the information to construct a correct planar graph) as one of the main causes of incorrect (non-recoverable) behavior. The results show that even for realistic and relatively small location errors, the effects of location errors are noticeable. This work represents the first detailed micro-level analysis of pathologies for geographic face-based routing protocols in the presence of location errors.

Authors of [24] consider the impact of location errors on geographic routing in multi hop wireless networks. They propose a new algorithm called MER (Maximum Expectation within transmission Range) that mitigates the effect of noisy location information by explicitly considering the error probability when making routing decisions. An improvement of MER is presented in [25], which optimizes both the rate of failure and the expected progress towards the destination.

ELLIPSE Routing Protocol with Uncertain Positions [26] was proposed in order to reduce the number of redundant messages by using a region-based routing. After building an ellipse region based on the position of the source and the sink, all nodes in the ellipse region and those who receive messages, forward it with a probability " p ". Probability " p " defines a sub-set of nodes which will relay messages towards the destination. Note that neighbors of source and sink always forward messages even if they are out of the ellipse. This is to ensure that, even if the source and sink are not accurately located, the message can be received. However, this strategy leads to network congestion by the unnecessary involvement of many relay nodes in the routing process. In addition, authors did not focus on the broadcasting strategy inside the ellipse.

ALBA-R protocol was proposed in [27, 28], a localization error-resilience geographic routing based on nodal coloring mechanism for handling nodal connectivity holes. Authors stated that their new protocol is totally

error-resilient but tested it only when the localization error does not exceed the transmission radius.

2.1 Works considering mobility pattern

In mobile networks, some geographic routing protocols have been proposed that exploit the mobility of nodes.

A refined next-hop selection rule named Mobility-based Adaptive Greedy Forwarding MAGF was proposed in [30], authors came up with a concept named *motion potential* that combines node position with mobility patterns to select the next-hop forwarder. The neighbors of a forwarder node are divided into two regions namely the *progressive region* in which the selection is exactly as greedy forwarding and the *potential region* where the motion potential function is applied to select the best forwarder without using face routing. The potential region is used as an alternative if there is no node in the progressive region thus no node closer to the destination than the forwarder. The motion potential function considers the speed of mobile nodes and the angle they form with the destination. Exploiting the mobility pattern together with the carry-and-forward strategy is a promising idea, in delay-tolerant networks and intermittently connected networks. However, authors assume that the mobility pattern and the locations of nodes are supposed to be known beforehand thanks to GPS, which may not be adequate for some WSNs applications if they require little volume and communication and even GPS is not without errors [46].

Another mobility-based forwarding mechanism has been proposed in [31] called GPSR-MS for GPSR with Mobile Sensors. GPSR-MS defines new metrics for selecting the next-hop forwarder, which considers the moving direction, the moving speed and the distance to the sink. All these neighbors' data in addition to the node's position are supposed to be known, which is not always feasible nor without error. In addition, their objective function does not cover all the motion possibilities of nodes. For instance, a node is considered static when its current distance to the sink is the same as the one calculated in the previous timestamp. This assumption jeopardizes the reliability of the used objective function. A study on effect of mobility patterns is also crucial but has not been considered by authors. In section 5, we have performed this study for various networks parameters.

Authors in [32] proposed a scheme that relies on the use of artificial neural networks (NN) to predict the future location of neighbors inside a geographic routing for multimedia streaming in MANETs. The NN algorithm was implemented in GPSR [45] in the greedy mode-only. The used protocol was modified to store also the two most recent coordinates and their timestamps instead of only the current coordinates. However, NN may not be suitable for intermittently connected and delay-tolerant networks since WSN applications require mainly low computation complexity [33].

All reviewed geographic routing schemes assume that the mobility patterns and the exact positions of the mobile nodes are known and those which consider localization errors deal only with static nodes or only assume mobile sinks. Besides, those which consider the mobility

prediction and location error assume nodes are equipped with GPS to get their exact velocity at any time. No previously published work proposed a geographic routing for mobile sensor networks executed together with mobility prediction-based localization of all the mobile nodes. The novelty of our proposal is that it gathers many realistic characteristics of WSNs, namely:

- Being both geographic and GPS-free
- All nodes can be continuously mobile
- Using a localization method based on mobility-prediction
- On-Demand and Mobility-based Adaptive neighborhood discovery
- Considering jointly the distance, location-errors, mobility patterns of nodes and the reliability of links in routing decisions
- Can be also suitable for intermittently connected networks and Delay-tolerant networks.

Before presenting our proposal INTEGER, we will present the effect of nodes mobility and localization errors on the next forwarder selection and show how this affects routing decisions in the following section.

3 Effects of Node Mobility and Localization Errors on Next-Forwarder Selection

Since nodes are mobile (without loss of generality, they can change their directions and speeds, i.e., velocities, at any time), using traditional forwarding strategies such as Most Forwarder within Radius (MFR), Nearest Forwarding Progress (NFP) or Compass Routing (CR) [34] are no longer valid for mobile nodes. Take the example of MFR widely used in greedy-based and geographic routing. It consists on selecting the nearest to the sink amongst the neighbors. Applied in a mobile scenario, this includes the case where this nearest node is moving in the total opposite direction of the sink or moves with a slower speed compared to its neighbors but has been chosen because at the time of the selection, it was the most adequate according to MFR strategy. In addition, nodes are not aware of their exact locations even when using GPS; they can only have estimations and localization errors are inevitable. As a consequence, nodes declared to be close to the sink, in reality may not be. In the following, we present examples of erroneous decisions caused by MFR strategy when applied in mobile scenarios.

3.1 Effect of Node's Direction

The frequent change in nodes directions may lead to erroneous routing decisions. For example, nodes declared going toward the sink in reality are not and vice versa. The following example shown in Fig. 1 explains how the node's direction effects routing decisions.

For a source node s , neighbor x_1 is the closest to sink d at time t_0 , however, neither x_1 nor x_2 are going towards the sink. Besides, x_3 that seems farther from the sink is going toward the sink and at time t_1 , x_3 will be the closest one to the sink. Nevertheless, MFR strategy chooses erroneously x_1 as the best forwarder of the sender s while in reality x_3

is. As a consequence, packets that should be sent to x_3 will be sent wrongly to x_1 since the sender s believes according to its neighborhood table built at t_0 that x_1 is the best candidate. This problem will be solved in next neighborhood discovery. Yet, the higher the discovery interval, the worse the performance will be.

3.2 Effect of Node's Speed

Mobile nodes either move with a constant velocity or change it frequently and variably. In the latter case, if the speed is not considered while selecting the best forwarder, a packet may be forwarded to a node that moves with a slow speed. This is because it has been chosen as a forwarder just because at the moment of forwarding, it was the closest one to the sink among the neighbors. However, other nodes may not be initially closer to the sink but can reach it quickly and before the assumed best forwarder. Fig. 2 provides an example of this case. Sender s has two neighbors closer to the sink than itself: x_1 and x_2 . However, x_2 is faster than x_1 . At time t_0 , x_1 is the closest node to the sink d so it will be chosen according to MFR. However, at time t_1 , x_2 becomes the closest one, but it might not be considered yet as the best forwarder if the neighborhood table of s is not updated by t_1 .

3.3 Effect of Localization Error

Geographic routing decision, particularly greedy forwarding, is based on the knowledge of the positions of nodes. However, knowing the accurate positions of nodes and especially of mobile nodes is an unrealistic assumption, and generally impossible in real world deployments. Thus, if a node communicates a wrong position to its neighbors, this can lead to incorrect (non-recoverable) behavior and noticeable degradation of performance [22]. A node believing being nearer to the sink in reality is not and vice versa. Consequently, the routing path may be much longer than what it should be and may result in loops. Fig. 3 illustrates an example of the effect of localization error on the next forwarder selection.

Let's consider node s the current packet holder, x_1 and x_2 are the real positions of its neighbors. x'_1 and x'_2 are their estimated positions communicated to node s respectively. In light of this information, s would choose node x'_1 as its best forwarder by applying MFR strategy since its position estimation is the closest to the sink d . However, in reality x_2 is the closest one.

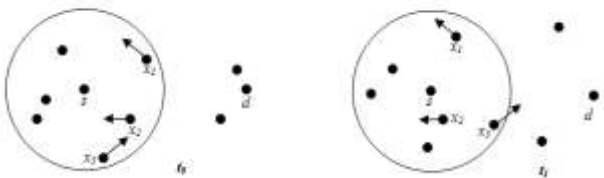


Fig. 1 Impact of the direction of mobile nodes on greedy forwarding

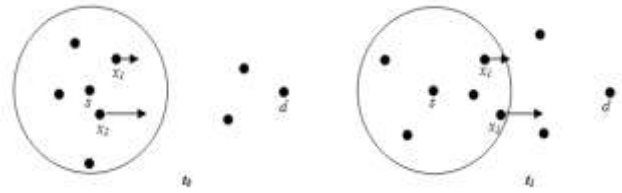


Fig. 2 Impact of the speed of mobile nodes on greedy forwarding

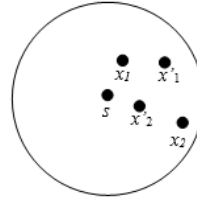


Fig. 3 Effect of localization error on forwarder selection

In light of the previous review about the effect of the direction and the speed of mobile nodes in addition to the localization error on next forwarder selection, and to overcome their related consequences, we present in the following section our intertwined localization and routing scheme.

4 INTEGER: Intertwined Localization and Routing Method

In this section, we propose a novel intertwined localization and routing scheme, namely INTEGER, for mobile sensor networks. We assume that sensor nodes have the same communication range and can devise neighbor distances based on the received signal strengths from their neighbors. The position of the destination node (i.e., sink) is known to all sensor nodes. For the case of mobile sink, we assume that it follows a predefined trajectory so that sensor nodes can know its current location at any time, thus relays can update the destination location of the packet before making routing decisions.

We propose a fully distributed localization and routing protocol. The protocol is composed of two intertwined algorithms; one for localization mobile nodes with an assisting mobile anchor, and then a geographic routing that uses results from the localization algorithm. At the beginning, an initialization phase of nodes' localization is necessary to serve as a preliminary step for the routing protocol. After relatively a stable time (generally after that the mobile anchor finish travelling letting nodes predicting their positions without the anchor assistance), the source nodes (randomly chosen and deployed) begin to send packets to the sink; and then the two algorithms continue simultaneously their execution. Note that mobile nodes estimate their positions periodically to be ready to communicate their location information when needed as detailed in the following sections.

4.1 Localization Algorithm

In this section, we describe our proposed localization algorithm Speed and Direction Prediction-based Localization (SDPL), which the preliminary version was previously published in [17]. The method allows localizing mobile sensor nodes with the assistance of a mobile anchor visiting the sensor area and following a path that ensures visiting the maximum number of nodes. Note that the mobile anchor is different from the sink as it is not permanently present in the network. In monitoring or disaster management applications [40], especially in hostile areas, the mobile anchor could be a UAV sent to monitor sensor nodes, take photos and videos and other tasks. The algorithm is fully distributed since mobile nodes estimate by themselves their location information and independently from each other, which makes the mechanism very suitable for intermittently connected networks. The mobile anchor initially provides an initial location reference to mobile nodes by sending location beacons periodically. When receiving anchor location beacons, nodes use the multilateration to estimate their positions. If nodes can no longer receive location beacons from the anchor, they continue estimating their positions independently from the anchor using our mobility-prediction scheme. The scheme exploits previous nodal locations to predict the speed and the direction a mobile node moves with. Due to space restrictions, we refer the interested reader to [17] and to its improvement published in [18] for further details.

The choice of using SDPL as a localization algorithm is motivated by its ability to provide an estimation of the speed V and the direction angle θ that a node is moving with and especially it provides an estimation of the localization error bound ε . These three parameters are very important for our forthcoming routing approach.

4.1.1 Location Prediction

Suppose that node i has estimated its position at time $t-1$, P_{t-1}^i , as (x_{t-1}^i, y_{t-1}^i) along with an estimation of its speed, V_{t-1}^i , and its angle of orientation, θ_{t-1}^i . Then, node i can predict its position P_t^i at time t , as follows:

$$P_t^i = P_{t-1}^i + \overrightarrow{V_{t-1}^i} \times \Delta T$$

$$\Leftrightarrow \begin{cases} x_t^i = x_{t-1}^i + V_{t-1}^i \times \cos \theta_{t-1}^i \times \Delta T \\ y_t^i = y_{t-1}^i + V_{t-1}^i \times \sin \theta_{t-1}^i \times \Delta T \end{cases} \quad (1)$$

where $\Delta T = T_t - T_{t-1}$ and where the speed is calculated as

$$V_{t-1}^i = \frac{\sqrt{(x_{t-1}^i - x_{t-2}^i)^2 + (y_{t-1}^i - y_{t-2}^i)^2}}{T_{t-1} - T_{t-2}} \quad (2)$$

and θ_{t-1}^i representing the angle between the x-axis and the speed vector $\overrightarrow{V_{t-1}^i}$ at time t is estimated as

$$\theta_{t-1}^i = \tan^{-1} \left(\frac{y_{t-1}^i - y_{t-2}^i}{x_{t-1}^i - x_{t-2}^i} \right) \quad (3)$$

4.1.2 Location Refinement

The idea of location refinement is to refine the estimated speed and direction angle based on the node's recorded historical information. In case node i has previously estimated n positions (n is a constant representing the number of previous estimated positions a node is allowed to store), the prediction could be more refined. Node i calculates then the speed between each couple of consecutive stored positions following formula (2) and takes the average as its predicted speed V_t^i . As for predicting the direction angle θ_t^i , it will be calculated as the angle formed between the x-axis and the linear regression line that best fits the n positions. Once the speed and the direction angle are predicted, node i continues to use formula (1) to predict its current position until receiving again location beacons from the anchor that allow it to refine its estimation; and previous stored estimated positions will be erased and their counter will be initialized.

Recall that the applications we are targeting by our approach such as wild animal monitoring and environmental surveillance do not require high location precisions even though a more accurate location would be more appreciated. Due to the computing complexity of Kalman Filter for tiny sensors, we have chosen to apply the polynomial regression with least squares in SDPL motivated by its simplicity to implement and its reduced time processing. In fact, Kalman filter was found of time complexity of $O(N^3)$ while the least square (LS) is of complexity of $O(N^2)$ [51] thus LS is faster than Kalman filter. In addition, polynomial regression fits a non-linear model to the data even though the regression is linear so it can be applied in larger scenarios. Thus, in polynomial regression, the errors don't have to be Gaussian; they only need to be uncorrelated [52].

It is also worth mentioning that we have improved the accuracy of SDPL, by an adapted selection function of the parameter n so that the linear regression line given by the least square approach fits with the adequate number of previous locations thus predicting more accurately the current location of a node.

The linear regression line of node i is defined by the line

$$y^i = a^i + b^i x^i \quad (4)$$

where

$$a^i = \frac{\sum_{j=1}^n y_j^i - b \sum_{j=1}^n x_j^i}{n} \quad (5)$$

$$b^i = \frac{n \sum_{j=1}^n (x_j^i y_j^i) - (\sum_{j=1}^n x_j^i)(\sum_{j=1}^n y_j^i)}{n \sum_{j=1}^n (x_j^i)^2 - (\sum_{j=1}^n x_j^i)^2} \quad (6)$$

where (x_j^i, y_j^i) are the coordinates of the location j among the n locations stored in node i .

4.1.3 Location Error

There is always an error ε_t^i between the real position and the estimated position. Since node i is not aware of its real position, it can only have an estimation of its location error.

Since nodes estimate their positions by themselves, their location errors are independent. The localization error of node i is defined as the distance between the estimated position and the real position of i and is calculated as:

$$\varepsilon_i = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (7)$$

Where (x_i, y_i) is the real position of i and (\hat{x}_i, \hat{y}_i) is its estimated position.

In static networks, most of works concerned by location error tackled only measurement-induced errors, specifically GPS-induced errors [48]. These works assume that the location error in each node is independent and is generally modeled by a Gaussian distribution $N(\mu, \sigma^2)$ with zero mean ($\mu = 0$) and finite standard deviation. The variance of the Gaussian error on x-axis and y-axis for each individual node are assumed to be equal. The Gaussian probability function is given by:

$$f(x) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (8)$$

The error ε_i of node i is supposed to follow a Rayleigh distribution with probability density function:

$$f(\varepsilon_i) = \frac{\varepsilon_i}{\sigma_i^2} e^{-\varepsilon_i^2 / 2\sigma_i^2} \quad (9)$$

However, in mobile networks, few works considered the location errors, mainly GPS-induced errors [49] while works using mobility prediction used simulation experiments to deduce then a location error model [23, 50]. With GPS-free localization such SDPL, modeling the localization error becomes even a very complex task as it depends on many factors including the sources of location errors such as the uncertainty in anchor beacons, the RSSI-induced error, the unpredictable mobility of nodes and the localization algorithm itself which makes the location error harder to predict/mitigate [6].

For these reasons, and as nodes are GPS-free and knowing that our scheme proposes the use of a mobile anchor to initially help localizing mobile nodes and to help in refining nodes' location estimations, modeling the location error differ from proposed models which led us to perform an experiment using simulation to study and to approximate the location error. Through this study, we observed that the location error of each mobile node depends upon two main parameters, its velocity and the location update time interval. When the velocity increases the location error increases too and when the time between two estimations increases, the location increases too. By analyzing the results, we propose the following approximation formula for predicting the location error bound. Node i can estimate its own location error bound at time t by:

$$\varepsilon_t^i = \frac{\pi}{2} \left| \vec{V}_t^i \right| \times \Delta T \times e^{-\frac{V_{min}}{\Delta T}} \quad (10)$$

where V_{min} is the minimum speed of nodes.

4.2 Routing Algorithm

In this section, we describe our proposed routing algorithm. It is mainly composed of two phases namely the On-Demand Mobility-based Adaptive Neighborhood Discovery and the Best Forwarder Selection. The neighborhood discovery is launched only when the packet holder needs to forward the packet, thus allowing saving more energy and wireless resources. The best forwarder selection relies on a new selection of metrics especially designed for mobile sensors taking into account the localization error, the speed and the direction of neighbors.

4.2.1 On-Demand Mobility-based Adaptive Neighborhood Discovery

Since nodes are mobile, exchanging periodically beacons between neighbors without packet forwarding (which is generally the case in most of proposed geographic routing protocols) consumes unnecessarily their energy and wireless resources [35]. Moreover, information obtained by long periodic beacon exchanges may not remain valid due to the mobility of nodes. This information invalidation becomes higher when the speed of nodes is high. Thus, it is more efficient for a mobile node to collect the position information of its neighbors only when it needs to forward a packet. This strategy has been successfully adopted by some geographic routing such as [36] and proved to be energy-efficient. The novelty of our approach is in proposing a function to evaluate the most adequate neighborhood discovery time interval that maximizes the delivery ratio.

Choosing a suitable time interval for neighborhood discovery is very important and can affect heavily the routing process. In one hand, if this interval is too long, then many forwarding opportunities may be missed [33], that is, the packet holder may miss many undiscovered neighbors which may be really good candidates to forward the packet, also this will lead to a long end-to-end delay as shown in Fig. 4 (b). On the other hand, if this interval is too short, the packet holder will unnecessarily rediscover its previous neighbors since no much change happened but this will be done to the detriment of energy consumption of nodes when exchanging messages and will increase the overhead, which may lead to collisions and bandwidth wastage. To evaluate the impact of neighborhood discovery time on the routing efficiency, we conducted a study through simulation. Fig. 4 shows the performance of INTEGER when varying the discovery time using the default values described in table 2. Clearly, Fig. 4 (a) shows that the delivery ratio reaches the peak at a given interval. The discovery time depends upon three parameters. The communication range (r) that defines the neighborhood region, the average number of neighbors per node (N) and the average speed of nodes (v). When this speed is unknown, we consider $v = \frac{V_{max}}{2}$.

To calculate the suitable discovery time (T), we suggest the following.

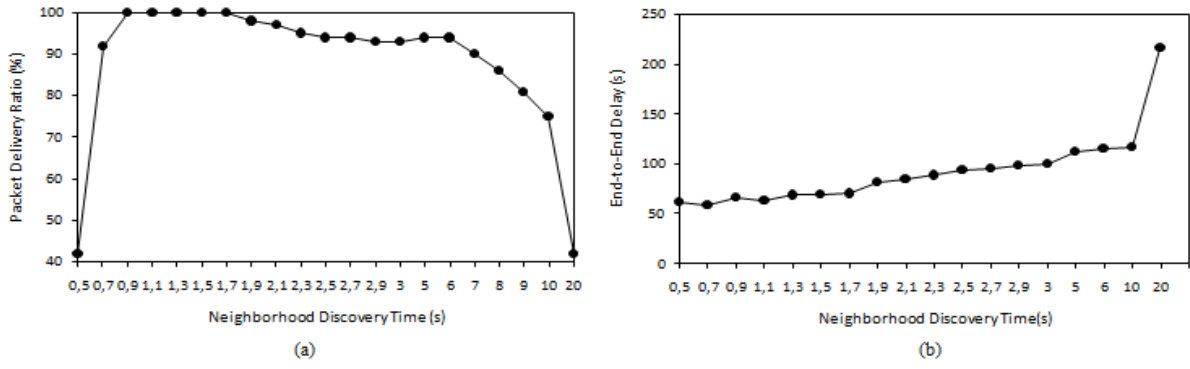


Fig. 4 Impact of Neighborhood Discovery Time

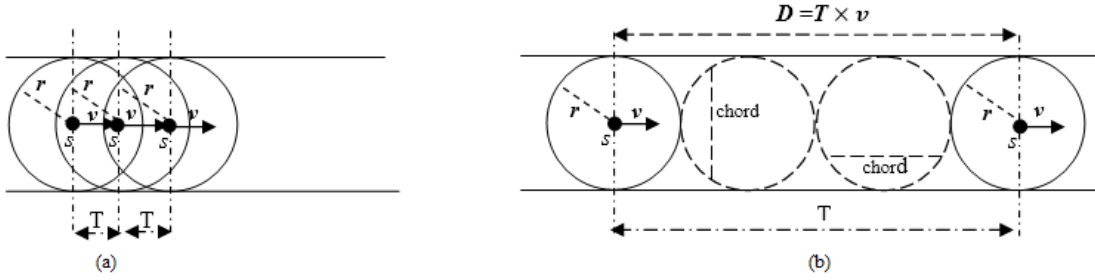


Fig.5 Overlapping and non-overlapping areas at successive neighborhood discoveries

To determine the lower and upper bounds of T , one considers the illustration in Fig. 5 that shows examples of overlapping and non-overlapping communication areas of node s at successive neighborhood discoveries. The discontinued circles represent virtual undiscovered areas between two neighborhood discoveries. To determine the upper bound, consider that the areas covered by s at each discovery do not overlap as shown in the example of Fig. 5(b), then the average number of missing neighbors during T is:

$$N_{miss} = N \times \frac{T}{T_{ch}} \quad (11)$$

where T_{ch} is the average time needed for a neighbor to pass the chord of the circle representing the communication area of s as represented in Fig. 5 (b). We have chosen the chord as a reference, as it is the average distance that a neighbor travels while being in the communication area of the packet holder.

The average length of the chord of a circle is given by

$$Chord\ Length = \frac{4r}{\pi} \quad (12)$$

$$\text{Thus, } T_{ch} = \frac{Chord\ Length}{v}, \text{ from (9), } T_{ch} = \frac{4r}{\pi v} \quad (13)$$

$$\text{From (11) and (13), } N_{miss} = \frac{N \times T \times \pi v}{4r} \quad (14)$$

$$\text{In order for } s \text{ to not miss any neighbor, } N_{miss} < 1 \quad (15)$$

$$\text{From (14) and (15): } \frac{N \times T \times \pi v}{4r} < 1 \rightarrow T < \frac{4r}{N \times \pi v} \quad (16)$$

To determine the lower bound of T , consider that the areas covered by s at each discovery overlap as shown in the example of Fig. 5(a). In order for s to discover at least one new neighbor in the next discovery, s should displace at least by $\frac{r}{N}$.

$$\text{Thus: } T \times v_{max} > \frac{r}{N} \quad (17)$$

$$\text{From (17): } T > \frac{r}{N \times v_{max}} \quad (18)$$

$$\text{From (16) and (18): } \frac{r}{N \times v_{max}} < T < \frac{4r}{N \times \pi v} \quad (19)$$

To select a new forwarder, the packet holder proceeds first to neighborhood discovery. To do so, it broadcasts a **Position Request** message to nodes within its transmission range for requesting neighbor position information. Once a neighbor node receives this message from the sender, it replies with a **Position Response** message containing its estimated position information including the estimated coordinates (x, y) with an error bound ϵ , its estimated speed V , and its estimated angle of orientation θ . Based on these data, the packet holder builds a temporary neighborhood table from which it selects its best forwarder.

4.2.2 Best Forwarder Selection

In this section, we describe the principle of our best forwarder selection and the parameters to consider when making routing decisions. As presented in section 3.1, nodes' directions can jeopardize the greedy forwarding. From this perspective, we propose to consider the direction of neighbors in the forwarder selection.

The best forwarder will be chosen among the neighbors that go in the direction towards the sink. Before sending a packet, the source node assigns a direction to the packet called *packet direction* so that only nodes having the same moving direction as the packet direction will be candidates to route the packet to its destination. To determine the packet direction, the source calculates the current angle θ that forms with the sink and assigns to the packet a direction number " ρ " among the four possible directions shown in Fig.6. Depending on the range of this angle, " ρ " is associated to the packet and will be considered as the *packet direction*. If the sink is static, the packet direction remains the same during the whole packet travel. If the sink is mobile, then each packet holder assigns a new packet direction depending on the current sink location and embeds it in the packet header.

The number " ρ " that identifies the packet direction is associated to each direction following function (20).

$$\left\{ \begin{array}{l} 0 < \theta \leq \frac{\pi}{2} \leftrightarrow \rho = 1, \\ \frac{\pi}{2} < \theta \leq \pi \leftrightarrow \rho = 2, \\ -\pi < \theta \leq -\frac{\pi}{2} \leftrightarrow \rho = 3, \\ -\frac{\pi}{2} < \theta \leq 0 \leftrightarrow \rho = 4. \end{array} \right. \quad (20)$$

Note that this identification of directions serves also to assign a direction number to mobile nodes so to make easy the comparison between the packet direction and the current direction of a candidate. In this case, θ will be the estimated angle of orientation of node x_i given by SDPL method at time t . The main difference between the packet direction and node direction numbers is that the packet direction is calculated considering the angle that a source forms with the sink at the time the packet generation independently from the current source direction. However, a node direction is calculated considering the angle of orientation that a node moves with, independently from the sink.

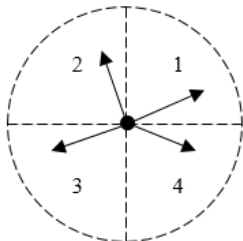


Fig. 6 The four possible directions

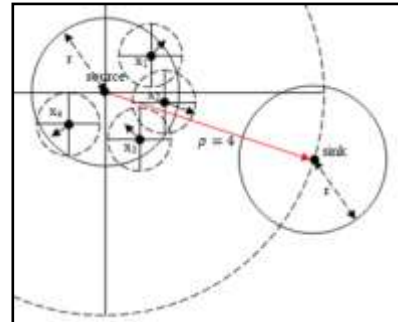


Fig. 7 Selection based on the packet direction and neighbors' directions

If the direction number of a neighbor x_i is the same as the packet direction, then x_i is a candidate, otherwise x_i is not going towards the sink. Note that the case when the angle between the source and the sink is very close to the lower or the upper bound of θ in function (20) is critical, as among nodes belonging to the previous or next direction may be good candidates. In this case, we apply the *right-hand-side* rule used in conventional geographic routing protocols to choose only one direction.

Fig. 7 shows an example of how a source node determines the packet direction and how to know the direction numbers of its neighbors. In this example, and according to the four possible directions, only x_2 will be a candidate since it has the same direction number as the packet direction number (which is number 4 in the example of Fig. 7). Once a sender has received *Position_Response* from its neighbors, it builds a temporary neighborhood table and associates to each neighbor a weight calculated based on the received location information from neighbors.

The weight measures a node's aptitude to forward efficiently a packet. The packet holder selects then the neighbor that has the same direction number as the packet direction number and that has the highest weight. The neighborhood table is as follows:

Table 1 Temporary Neighborhood Table

Estimated (x, y)	Estimated (V, θ)	Localization Error ϵ	Weight W	Direction Number ρ
--------------------	-------------------------	-------------------------------	------------	-------------------------

Note that neighbors that are not in the progressive area (grey area in Fig. 8) towards the destination could be also candidates even if they are farther to the destination than the packet holder, i.e. behind the sender. The main reason to not exclude these neighbors, as greedy forwarding does, is that among them, it may be a good forwarder going towards the destination and with higher speed than the current packet holder thus resulting in having a higher weight than the packet holder. Fig. 8 shows a similar case where x_4 , and despite being farther from the destination than the packet holder s , it moves with a higher speed than s and with the same direction as the packet direction (the length of the red arrows represents the velocity magnitude). Thus, it can forward the packet quicker than s , which makes it good candidate.

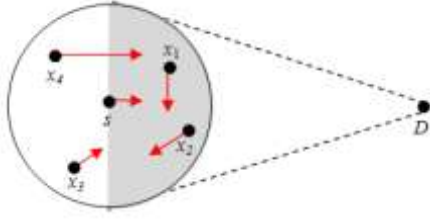


Fig. 8 Candidate selection favoring the speed

Thus we can formally define the best forwarder F as:

$$F = \{F \in N_s \mid (W_F = \max_{i \in N_s} W_i) \cap (W_F > W_S) \cap (\rho_F = k)\} \quad (21)$$

where N_s is the set of the neighbors of the sender node s and ρ_F is the direction number of neighbor F and k is the packet direction. The weight of each node is calculated using the following formula:

$$W_i = \begin{cases} \frac{V_i \times Drss_i}{ED_i \times \varepsilon_i}, & i \neq s \\ \frac{V_i \times R}{ED_i \times \varepsilon_i}, & i = s \end{cases} \quad (22)$$

where V_i is the estimated speed of a neighbor i , $Drss_i$ is the distance between node i and the sender based on the received signal strength, ED_i is the distance between the estimated position of the neighbor i and the Destination, ε_i is the localization error bound of a neighbor i . In order not to fall in a loop selection, the packet holder calculates its weight considering the $Drss$ as its radio range R as mentioned in formula (22).

In formula (22), the speed is in the numerator so that neighbors with higher speed will be favored. The converted distance from the RSSI is also in the numerator to favor the farthest neighbors of the sender to make significant progress towards the destination. Even if the farthest neighbors of the sender might not be the closest to the destination but favoring also the shortest estimated distance between a neighbor and the destination can balance the total weight of a node. The localization error is in the denominator to favor candidates with low localization error thus their communicated data are more trusted. Because all the other metrics are estimated, the real distance between the sender and candidate i represented by $Drss_i$ can balance the formula by bringing certain exactitude to the formula. In formula (22), $\frac{V_i}{ED_i}$ can be seen as $\frac{1}{ET_i}$ where ET_i is the estimated delay for a packet to virtually travel the direct link between node i and the destination. This delay can be considered as a *Quality of Service (QoS) factor* for node i . The smaller this delay, the higher is the weight of a node.

Algorithm 1: Best Forwarder ($i, N_i, k, Destination$)

```

//  $i$  is the packet holder,
//  $N_i$  is the set of neighbors of node  $i$ 
//  $k$  is the packet direction number
1: Begin
2:    $F \leftarrow i$ 
3:    $W_i = \frac{V_i \times R}{D(i,d) \times \varepsilon_i}$ 
4:   For each  $N \in N_i$  do
5:      $\theta \leftarrow$  SDPL direction angle ( $N$ )
6:      $\rho \leftarrow$  Direction Number ( $\theta$ ) //Calculated according to formula (17)
7:      $W_N = \frac{V_N \times Drss(N,i)}{D(N,d) \times \varepsilon_N}$ 
8:     if ( $W_N > W_i$  and  $\rho = k$ )
9:        $F \leftarrow N$ 
10:       $W_i \leftarrow W_N$ 
11:    end if
12:  end do
13:  return  $F$ 
14: End

```

$\frac{Drss_i}{\varepsilon_i}$ can be considered as a *link reliability factor* for node i . The higher this factor is, the higher is the weight.

Under equal conditions for example: $ED_1 > ED_2$ and $\varepsilon_2 > \varepsilon_1$, the reliability factor $RFi = \frac{Drss_i}{\varepsilon_i}$ from the sender perspective can determine which one is favored. The one having higher reliability factor is chosen as the best forwarder. In general, if two candidates have equal weighs, priority is given to the candidate with less location error because its location information is more reliable thus the candidate is trustful. If they have the same location error, priority is then given to the one with higher speed in order to speed up the packet delivery. If all parameters are equal, then the one having an angle of orientation closer to the packet direction angle is favored thus to favor the candidate whose direction converge toward the destination.

Algorithm 1 summarizes the best forwarder selection paradigm and Fig. 9 shows an example of selecting the best forwarding considering the different neighboring parameters. Let s be the sender node, x_1 and x_2 are its neighbors that have the same direction as the packet direction (red arrows in Fig. 9 show that they are in direction 1 as $s-D$ link). In other words, they are candidate to be forwarders. (s, x_1, x_2) are their real positions respectively while (s', x'_1, x'_2) are their estimated positions that they communicate to the packet holder. ($\varepsilon_s, \varepsilon_1, \varepsilon_2$) are their location-error bounds. sx_1, sx_2 are distances between the sender and x_1 and x_2 respectively. These distances are converted from the received signal strength indicator while receiving position response from the neighbors. ED_1 and ED_2 are the estimated distances between the neighbors and the sink. (V_s, V_{x_1}, V_{x_2}) are their estimated speeds respectively. The packet holder compares between the neighbors' weights and its self-weight. The

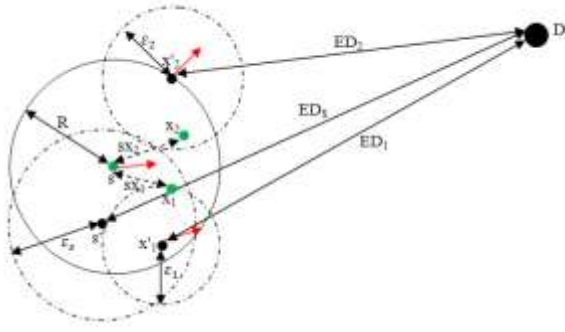


Fig. 9 Candidate selection considering different parameters

neighbor that has the highest weight with the same direction number as the packet direction will be selected as the best forwarder.

Numerical Example: In the previous example, let the estimated speed of node x_1 be 7m/s and that of x_2 be 5m/s. let the estimated distance between x_1 and the sink be 75m and that between x_2 and the sink be 60m. Accordingly, x_1 is far from the sink than x_2 . The D_{rss} that s has converted from the received signal of x_1 is 15m and that from x_2 is 20m. The localization error bound of x_1 is estimated to be 10m while that of x_2 is 20m. The location information of s is as follows: the estimated speed is 3m/s, the estimated distance between s and the destination is 85m and its error bound is 25m. The communication range is 30m. According to MFR strategy, s believes that x_2 is its best forwarder. However, by applying Formula (19), $W_1 = 0.14$ and $W_2 = 0.08$. $W_s = 0.04$. The sender forwards then the packet to x_1 and destroys the neighborhood table after receiving an acknowledgment from x_1 to save its memory.

We give priority to nodes with higher speed values and with minimal error bound. Consider the following example: Let x_1 and x_2 have the same speed 6m/s. $W_1 = 0.12$ and $W_2 = 0.10$. Then, x_1 will be chosen since it has the lowest error bound because it is more credible even if it believes that it is farther from the sink than x_2 .

Equally, if the x_1 and x_2 have almost the same error bound 15m but with different speeds, x_1 with 7m/s and x_2 with 10m/s.

$W_1 = 0.09$ and $W_2 = 0.22$. Then, x_2 will be chosen since there is high probability that it reaches the sink before the x_1 even though they have the same location error bound.

If the packet holder finds no neighbor that goes towards the sink or no neighbor that has higher weight than its weight, i.e. no neighbor fulfills condition (18), then it applies the *carry-and-forward* strategy, that is, it keeps holding the packet while continuing its basic trajectory hoping to across a neighbor that goes toward the destination or being in a more adequate neighborhood.

In the previous example, if s moves with 12m/s which higher than the speeds of its neighbors, then $W_s = 0.16 > W_1 = 0.14 > W_2 = 0.08$. Thus, s has the highest weight, which makes it more appropriate to route quickly the packet.

Same case if the location error bound of s is smaller than the errors of its neighbors. Let it be 5m, then $W_s = 0.21 > W_1 = 0.14 > W_2 = 0.08$. Thus, s has the highest weight because its location data are more trustful. Therefore,

Algorithm 2: Forward (i, pk, Destination)

```

// Initialization executed at the time of packet generation
a: if (i ∈ S) // S is the set of source nodes
b:   Pk ← Packet
c:   φ ← Current angle between node i and Destination
d:   k ← Direction Number (φ) // Calculated according
      to formula (17)

e: end if
1: Begin
2: Ni ← Neighborhood Discovery (i)
3: F ← Best Forwarder (i, Ni, k, Destination)
4: if (F = i)
5:   i keeps pk for T
6:   go to (2)
7: else
8:   Send Packet Pk to F
9:   Destroy Table of neighborhood
10:  Wait for Ack from F
11:  if no ack received from F
12:    go to (6)
13:  end if
14: end if
15: End

```

keeping the packet is wiser than forwarding it. In this case, while keeping the packet, the packet holder checks then for a new neighborhood each time T (as calculated in formula 19). Once a best forwarder has been selected, the packet holder sends the packet to this selected node, keeps a copy of the packet and waits for the reception of an acknowledgment.

Algorithm 2 explains the behavior of a packet holder i during the forwarding process.

4.2.3 Local Maximum Handling

Sometimes, for some reasons, such as collisions, the selected best forwarder cannot receive the packet, thus the sender will not receive any acknowledgment. In this case, once the duration for receiving the acknowledgment expires, the sender anew launches the neighborhood discovery and the best forwarder re-selection, which may result in choosing the same forwarder or another forwarder due to mobility change. The process of forwarding is repeated until the packet reaches the destination. So instead of just sending a packet to its best forwarder, in our approach, the sender makes sure the packet is well received by the best forwarder otherwise it tries to find another forwarder able to receive the packet. This could be very efficient especially in highly dynamic networks where nodes that were not considered best forwarders may become it in the next neighborhood discovery due to their mobility change, which may play in their favor. This strategy of keeping the packet and re-selecting another forwarder allows saving packets from being dropped or lost. Our simulations confirmed this gain. When considering default values in table 2, we found out that this re-selection strategy increases the delivery ratio by up to 30% as shown in Fig. 10.

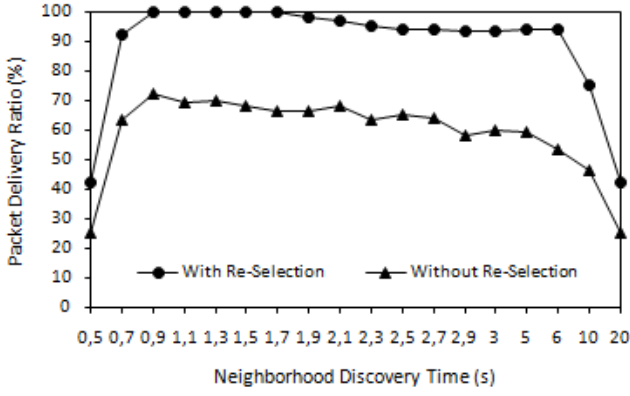


Fig. 10 Impact of re-selection strategy on Delivery Ratio

4.2.4 Loop Handling

Note that it is possible that a given node x_i can be chosen as the best forwarder even though it has previously forwarded the same packet. This is because of x_i 's mobility change. x_i can move to a more suitable new location or increase its speed that may be considered by the current packet holder as favorable using our forwarding scheme. Unlike the traditional loop formed in static networks, we believe this case of *re-use* can be productive as long as this reuse helps achieving progress to forward the packet towards the packet destination.

4.3 Properties and Comparison

In this section, we present the main properties of INTEGER. As for comparison, we decided to compare INTEGER with MAGF [30], GPSR-MS [31] and GPSR [45]. The choice of MAGF was motivated by its utilization of the store and forward strategy. When a node cannot find a better forwarder than itself, it carries the packet until finding a better neighbor to advance the packet towards the destination. In addition, MAGF exploits the mobility pattern of nodes to select the best next-forwarder as INTEGER does. The main differences between MAGF and INTEGER are that the former supposes that the exact location of nodes and their mobility including their speed and direction are known, which is not the case for INTEGER that relies on the prediction of nodes locations including predicting their speeds and directions. MAGF applies the objective function as a back up and only for neighbors behind the sender while the objective function of INTEGER applies to all neighbors. GPSR-MS also exploits the node mobility pattern and involves the current speed of nodes and evolution in distances in the objective function. GPSR was chosen as a baseline reference.

Property 1. *INTEGER selects more reliable candidates than its concurrents.*

Proof: consider the objective function of each proposal:

$$OF_{GPSR} = D(i, d) \quad (23)$$

$$OF_{GPSR-MS} = \frac{D_p(i, d) \times D_p(i, d)}{D_n(i, d) \times v_i} \quad (24)$$

Where $D_p(i, d)$ is the previous distance between node i and destination d , while $D_n(i, d)$ is the new one. The fraction of these two distances is considered by the authors as the direction of node i .

$$OF_{MAGF} = \begin{cases} D(i, d) & \text{if } D(i, d) < D(s, d) \\ MP(i, d) & \text{otherwise} \end{cases} \quad (25)$$

Where

$$MP(i, d) = \begin{cases} 1 + \frac{v_i}{\cos \theta \times D(i, d) - \sqrt{R^2 - (\sin \theta)^2 \times D(i, d)^2}} & \text{if } 0 \leq \theta < \sin^{-1} \frac{R}{D(i, d)} \\ \frac{\theta}{2} & \sin^{-1} \frac{R}{D(i, d)} \leq \theta \leq \pi \end{cases} \quad (26)$$

Where θ represents the angle formed by the line segment connecting node i and destination d , and the current motion vector of node i .

Unlike its concurrents, INTEGER considers in the candidate weight the distance between the sender and a candidate; this is to ensure the link quality during data transfer especially in high dynamic scenarios. While the other approaches do not consider such metric. In addition to the consideration of location errors in the forwarder selection which is unique to INTEGER. As mentioned earlier, $\frac{Dr_{ss_i}}{\epsilon_i}$ is considered as reliability factor.

Property 2. *INTEGER makes efficient progress toward the destination.*

Proof: INTEGER selects only candidates going toward the destination and with higher speeds while GPSR doesn't consider the moving speed at all, GPSR-MS favors also higher speeds but the approach how a sender perceives the direction a candidate is moving with, is not reliable. Authors considered a node is going toward the destination if its new distance to the destination is smaller than the previous one; which is not always the case as a node may move to a position closer to the destination but is going completely toward other direction. In addition, authors consider a node is static if the new distance to the destination is equal to the previous one, which is not always the case as shown in figure 11.

As for MAGF, the motion potential objective function that considers the speed and the direction of a candidate is applied only as a back up if the greedy forwarding fails which means in the default case only the distance between a candidate and the destination is considered to select the forwarder.

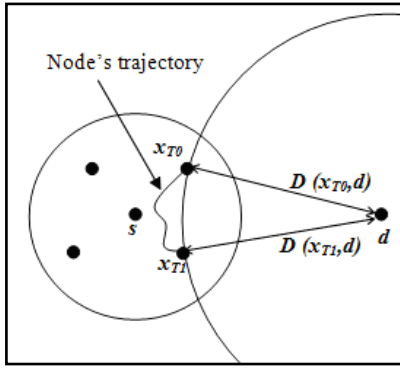


Fig.11 a mobile node moving with the same distance

Property 3. INTEGER guarantees the delivery of the packet.

Proof: thanks to its strategy of re-selection of the best forwarder in case of failure of reception by the chosen forwarder, INTEGER guarantees the delivery of a packet. The main concern becomes then when the packet arrives at its destination and the energy needed to deliver it successfully. While for the concurrent approaches, a packet is dropped if a neighbor failed to receive the packet.

5 Performance Evaluation

5.1 Simulation Scenarios and Metrics

To evaluate our proposed protocol INTEGER, we have implemented it in NS2 simulator [37] which is widely used in academic network researches.

We consider networks with n nodes, where $n = 100$ by default. Initially, the sensors are randomly and uniformly deployed in a square area of size 200×200 m². The default transmission range r is set to 30m. Therefore, the network degree d is around 7 according to defined in [38]:

$$d = \frac{n \pi r^2}{A} \quad (27)$$

where A is the sensor communication area.

We decided to show results when the number of sources is set to 10 sources (10% of nodes) because using less than this number of sources, the packet delivery ratio of our proposal was observed to be always 100% for all scenarios. Thus, the comparison with other protocols would not be significant. Data traffic is generated according to a Poisson process of intensity λ packets per second over the whole network. The traffic rate λ varies from 0.25 to 2 packets per second. Data packets are all 256B long. We randomly choose 10 source nodes while we keep the sink static and located at the center of the area. Mobile nodes follow the Random Waypoint (RWP) mobility model with no pause time. The maximum speed of nodes ranges in $\{3, 5, 10, 15\}$ m/s while the minimum speed is 0.05 m/s to ensure that all

nodes are totally mobile. The mobile anchor that assists the localization phase follows WAVES trajectory [39] with a resolution of 20m and has a fixed speed of 20m/s emulating UAV average speed. Each 6 seconds, nodes independently estimate their positions. The routing phase starts after 100s of the execution time to let nodes use the localization prediction. All our results have been obtained by averaging the outcomes of 50 independent simulations; each running for 1000 s. Table 2 summarizes the parameters ranges and the default values.

To analyze the simulation results, the main metrics chosen are the packet delivery ratio, the packet delivery delay, the number of relay nodes per packet and the consumed energy per packet. The packet delivery ratio, defined as the fraction of packets that are successfully delivered to the sink; and the delivery delay, defined as the time from packet generation until its delivery to the sink. The number of relay nodes is the number of nodes participating in the routing process to deliver successfully a packet to the sink.

The per packet energy consumption, defined as the average amount of energy expended by all nodes to successfully deliver a packet to the sink and is calculated by

$$Energy = E_{Idle} + E_{Sleep} + E_{Tx} + E_{Rx} + E_{Trans} \quad (28)$$

where E_{Idle} , E_{Sleep} , E_{Trans} is the energy spent during the idle, sleep and transition states respectively. E_{Tx} , E_{Rx} is the energy necessary for transmitting and receiving packets respectively.

Table 2 Simulation Parameters

Parameters	Range	Default value
Area Edge Length	{100,150,200,250,300}	200m
Number of Nodes	100	100
Transmission range r	{20, 25, 30, 40, 60}m	30m
Nodes Degree d	[3.14- 28.26]	7.07
Packet generation rate PGR	[0.05- 2]	0.1
Mobility Model	RWP	RWP
Pause Time	0s	0s
SDPL Location Estimation Interval ΔT	6s	6s
Maximum speed of nodes V_{max}	{3, 5, 10, 15}m/s	5m/s
Minimum speed of nodes V_{min}	0.05 m/s	0.05 m/s
Data Packet Size	256B	256B
Number of sources	10	10
Simulation time	1000s	1000s

The latter metrics are computed only for successfully delivered packets. When varying the localization error, we consider the average localization error of all sensors over the whole execution time. The mean error of all nodes is calculated each 6 seconds and the localization error is averaged at the end of the simulation.

We conducted a series of tests. The first set concerns the study of the impact of the nodes' speed on the delivery ratio, the end-to-end delay and the average number of relay nodes as well as the consumed energy on the compared protocols. In the second, we vary the packet generation rate and we study its impact on the above metrics. The variation of the localization error is also studied to proof the error-resilience nature of our approach. We finish the tests with studying the effect of the network degree on the compared protocols.

5.2 Performance Comparison

For simulation comparison, we compare INTEGER to its concurrents namely MAGF, GPRS-MS and GPRS. Note that none of the compared protocols basically used the mobility-prediction based localization; all considered the exact location without mentioning how this exact location is obtained. In contrast, the mobility-prediction based localization is a main component of our approach and considering the localization errors in routing decisions together with predicted mobility patterns is part of our contribution. For this reason and in purpose of doing an objective comparison between the four protocols, we

associated our location-prediction scheme and the carry and forward strategy as well as the re-selection approach to the concurrent protocols. Thus, the comparison will be judged according to how successful the selection of the next-forwarder is and the impact of the consideration of the localization error in routing decisions. As for MAGF, the cache time was set to a value that ensures for the packet holder to carry the packet until finding a suitable neighbor.

5.3 Simulation Results

5.3.1 Impact of the speed

In applications such as monitoring animals or weather monitoring relying on people/vehicles, the speed of individuals may vary without knowing the exact moving speed but only their maximum speed. The maximum speed can be derived from their biological/industrial nature or the congestion of roads taken. In this test, we vary the maximum speed of nodes between 3 m/s which represents the average human/animal walking speed and 15 m/s which represents the average driving speed of vehicles in urban agglomeration or the average running speed of wild animals.

Fig.12 (a) shows the packet delivery ratio when the maximum speed of nodes varies. Clearly INTEGER outperforms the other protocols.

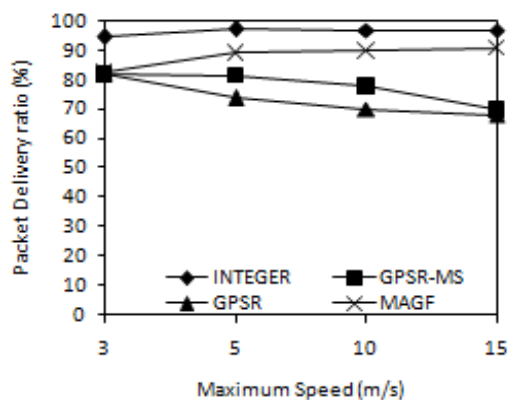


Fig.12 (a) Packet Delivery Ratio vs. Maximum Speed

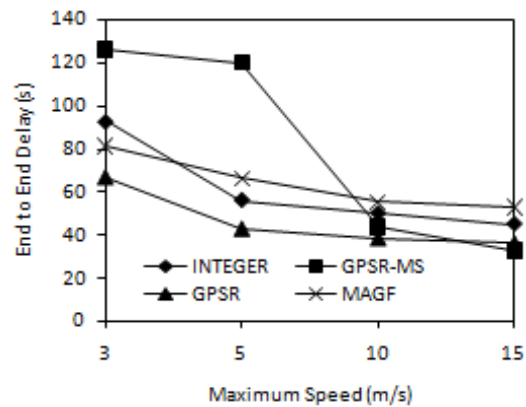


Fig.12 (b) End-to-End Delay vs. Maximum Speed

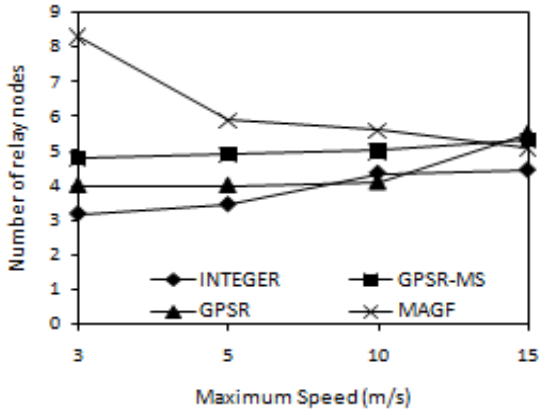


Fig.12 (c) Number of Relays vs. Maximum Speed

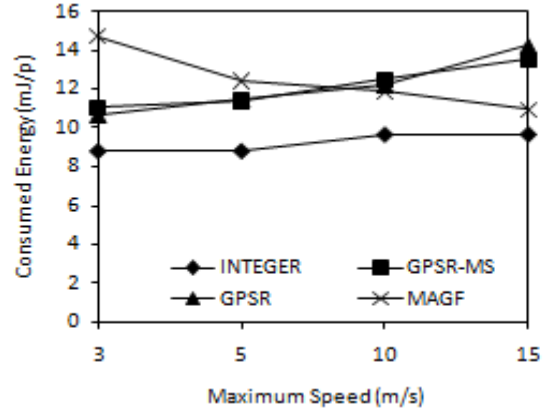


Fig.12 (d) Consumed Energy vs. Maximum Speed

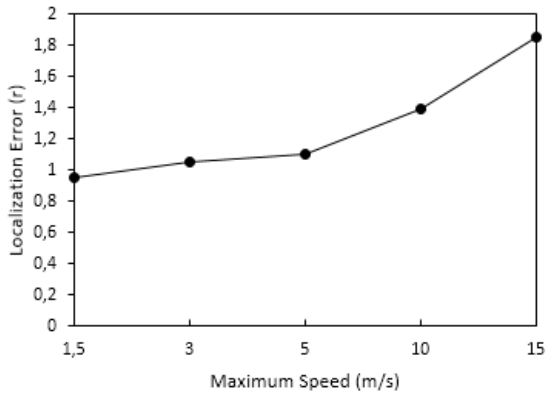


Fig. 13 Localization Error vs. Maximum Speed

We notice a tendency towards stability, thus less affected by the speed increases. This is due to its strategy that supports the speed variations and localization errors caused by the increase of nodes' speeds. The improvement brought by INTEGER compared to MAGF is between 7% and 12%, while compared to GPSR-MS and GPSR the improvement is in the range of 12% and 28%. It is important to mention that INTEGER and MAGF have both a behavior different from the one of GPSR-MS and GPSR when varying the speed of nodes. While the delivery ratio increases for the two first protocols when increasing the speed, it decreases for the two last protocols. This attests that considering efficiently the mobility pattern of mobile nodes can improve much the routing efficiency by selecting the best next-forwarder. It is worth to mention that with the increase of nodes' speed, the localization errors increase too as shown in Fig. 13.

The consideration of localization errors by INTEGER when making routing decisions proves its eligibility and should not be ignored or taken lightly. Since GPSR is based only on geographic advancement, the nodes tend to pick less reliable relays, which explains why GPSR gives the worst results. As for the end-to-end delay, clearly, Fig.12 (b) shows that when nodes move with low speeds,

this influences the packet delivery delay because of the carry and forward strategy. In other words, the packet speed depends on nodes' speeds. The higher the speed of the nodes, the lower the end-to-end delay. In addition, INTEGER tries always to choose the best combination of the most influencing factors (speed, direction, location error and distance toward the destination) to select the best forwarder. Before achieving this refined selection, the packet holder keeps carrying the packet with its own moving speed. Besides, GPSR and GPSR-MS have less constraints choosing the next-forwarder which leads them to make quick decisions but to the detriment of the packet delivery success. Fig.12 (c) shows the average number of relay nodes per successfully delivered packet. Our approach of considering the mobility pattern together with localization errors allows INTEGER to use less and well-chosen relay nodes to transmit packets. The number of relay nodes participating in the packet transmission slightly increases when increasing the mobility speed because of the frequent change of the neighborhood thus meeting new candidates. The same behavior is observed for GPSR-MS. GPSR shows a steady behavior between 3m/s and 10m/s while its curve heightens for 15m/s as GPSR works bad in highly mobile networks. As for MAGF, a decrease in the number of relay nodes is observed when increasing the speed of nodes but still this number is higher compared to other protocols. The consumed energy per received packet is shown in Fig.12 (d). INTEGER shows a quite steady curve and widely outperforms other protocols evaluated while the energy consumed by MAGF decreases when the speed increases.

As shown in Fig.12 (b), the packet speed is proportional to the node's speed, and because INTEGER and MAGF give the better results in terms of packet delivery, they tend to deliver the overall amount of packets generated by the sources in a relatively short time compared to GPSR and GPSR-MS, which makes nodes take a transmission-break after finishing transmitting all the packets thus consuming less energy. However, it is not the case for GPSR and GPSR-MS where it was observed that the delivery ratio

lowers when increasing the nodes' speed because lot of packets travel the network hopping from node to node without reaching the destination thus consuming more energy.

5.3.2 Impact of the packet generation rate

Now we set the speed to 5m/s which represents a moderate speed and we vary the packet generation rate PGR generated by each source node.

Fig.14 (a) shows that our proposed INTEGER widely outperforms the other protocols, especially when the packet traffic is higher. The improvement brought by INTEGER can reach 70%. All the curves show a decreasing behavior in the packet delivery ratio when increasing the traffic in the network. This is mainly because of the occurrence of collisions. In fact, the higher the traffic in the network, the more the collisions occur due to using only one channel.

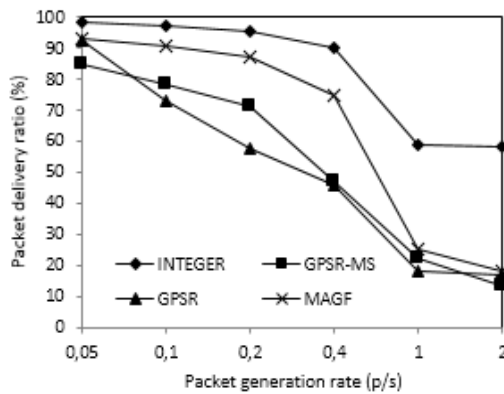


Fig.14 (a) Packet Delivery Ratio vs. PGR

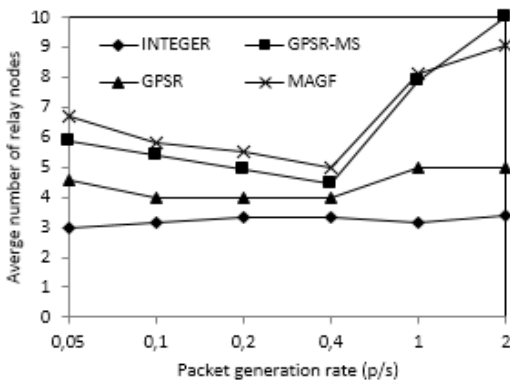


Fig.14 (c) Number of Relay nodes vs. PGR

Note that without our strategy of re-forwarding after collision applied to all protocols in case of no acknowledgment was received, the delivery ratio is worse especially for GPSR, GPSR-MS and MAGF. INTEGER keeps the highest ratio even when the traffic is high and with relatively good end-to-end delay compared to the ones given by MAGF and GPSR as shown in Fig.14 (b) while using the minimum number of relay nodes as shown in Fig. 14 (c). The per packet consumed energy of INTEGER as shown in Fig. 14 (d) is the lowest one among the four protocols. All these results prove that our best forwarder selection scheme improves much the efficiency of the routing process and allows selecting the most adequate forwarder. As for the other protocols, nodes consume more energy and higher delay when trying to re-launch the forwarding selection and re-forward the packet to the new selected forwarder even if the latter is not available to receive more packets due to traffic congestion in the network.

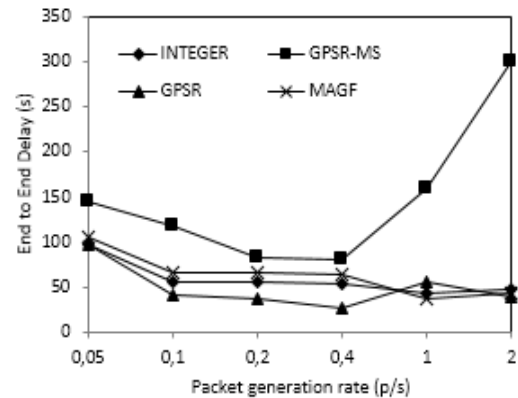


Fig.14 (b) End-to-End Delay vs. PGR

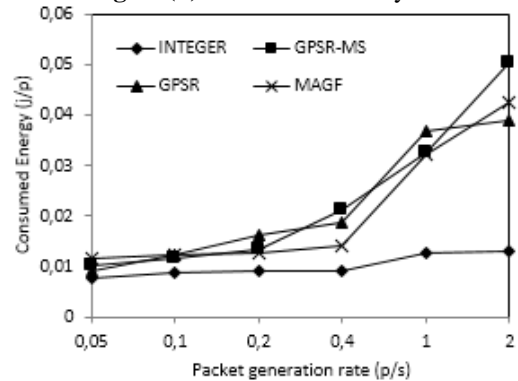


Fig.14 (d) Consumed Energy vs. PGR

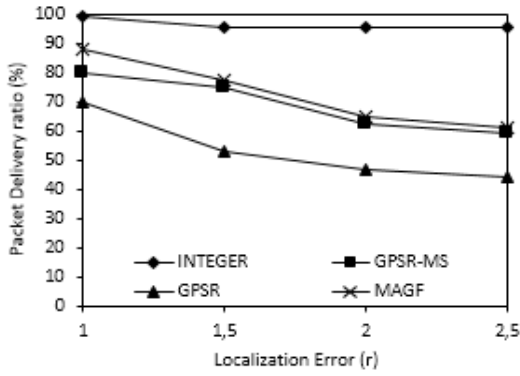


Fig.15 (a) Packet Delivery Ratio vs. Error

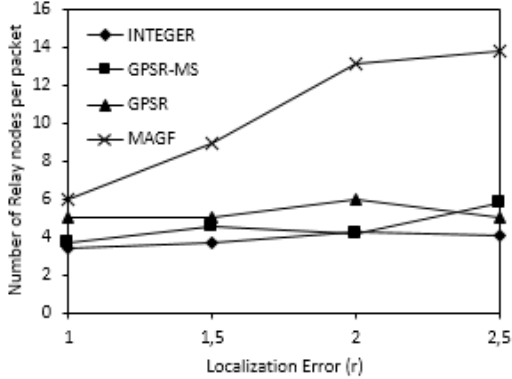


Fig.15 (c) Number of Relay nodes vs. Error

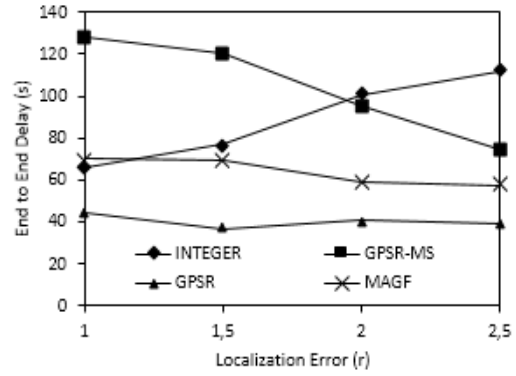


Fig.15 (b) End-to-End Delay vs. Error

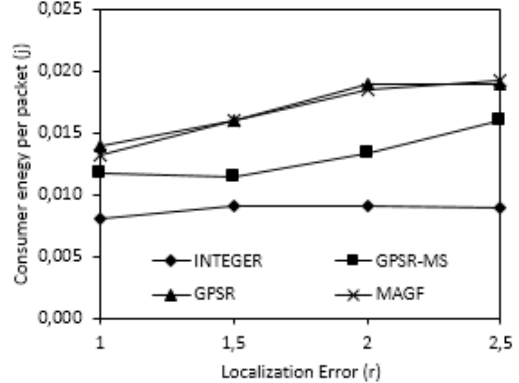


Fig.15 (d) Consumed Energy vs. Error

5.3.3 Impact of the localization error-bound

In this test, we intentionally vary the localization error-bound resulted from the location-prediction to study its effect on the proposed geographic routing. The default values are applied.

As we can notice, INTEGER gives the best results for most of the metrics. This is mainly because INTEGER considers the localization error while making routing decisions. Nodes with less location error are always favored to be next forwarders because they are more reliable and trusted. The degradation in the delivery ratio is minimal as shown in Fig. 15 (a) while the average number of relay nodes is slightly increasing and so the consumed energy. However, Fig. 15 (b) shows that the end-to-end delay per successful packet is very sensitive to the location-error. We explain that by the need for nodes to carry the packet for additional time until finding a suitable forwarder with less location-error. As for MAGF, the delivery ratio decreases when increasing the location-error and the degradation is noticeable while the consumed energy and the number of relay nodes increase sharply because nodes try hopelessly to find suitable forwarders but because of the high location-error, those forwarders are erroneously declaring being suitable and so will be erroneously selected which leads sometimes to distant the packet from the destination. Consequently, nodes will be obliged to reselect and find

other more suitable forwarders. Because many nodes participate in the routing process in MAGF, this leads to more energy consumption. Meanwhile, we notice that the end-to-end delay given by MAGF slightly decreases. We explain that by the proportionality between the delay and the number of successfully delivered packets since the shown end-to-end delay is given for only successful delivered packets. Same explanation would be given for GPSR-MS and GPSR. Note that GPSR gives the lowest end-to-end delay but to the detriment of poor packet delivery ratio.

5.3.4 Impact of nodes' degree

We mean by the nodes' degree, the average number of neighbors per node. The node degree is a very important parameter since together with the mobility pattern, it determines how well connected nodes are. Note that according to formula (27), the obtained results are similar when changing the number of nodes in the network or the area length or the communication range. In fact, the results are equivalent if we vary the number of nodes in the set {45, 70, 100, 180, 400} or the length of the sensor area in {350, 250, 200, 150, 100}. For simulation simplicity, we chose to vary the communication range of nodes in the set {20, 25, 30, 40, 60} which will give a node degree in the set {3, 5, 7, 13, 28}.

In this test, we are more interested in studying the behavior of INTEGER when changing the mobility pattern. To do so, the maximum speed of nodes is varied from 5 to 15 m/s. In Fig.16 (a), when the speed of nodes is low (5m/s), this gives more link reliability, that is why the curve given by 5m/s outperforms those of 10m/s and 15m/s. Fig. 16 (a) shows also that the delivery ratio increases with the increase of the communication range. This is because a packet holder has more candidates in its neighborhood. This same reason explains why the end-to-end delay in Fig.16 (b) decreases when increasing the communication range. In fact, more nodes become reachable within a single hop. This saves considerably the time for discovering neighbors and minimizes doing frequent hop-to-hop transmissions. Because it is obvious that with the increase of the communication range, the number of relay nodes participating in the routing process decreases, we chose to show in Fig.16 (c) how this influences also the time that a node is obliged to carry a packet before forwarding it to the next forwarder. With shorter communication range, a node has less candidate neighbors and the packet holder may not find easily a neighbor with better weight than itself, which obliges it to keep the packet for a long time until crossing a suitable forwarder. Add to that, if its speed is low, then the time for its displacement and the time to its neighborhood to change will be also long. This time will be reduced when a packet holder has a larger choice among its neighbors and the number of hops to reach the destination will be reduced, which saves the routing time. As for the consumed energy, Fig.16 (d) shows that the variation is smooth. This is mainly because the energy wasted in discovering frequently new neighbors is saved in case of larger communication range. A larger number of neighbors could be discovered at once but there is a need for more energy to transmit and receive Position_Request and Position_Response messages, also to transmit a packet if the selected neighbor is far from the sender, which makes such balance in the consumed energy. An interesting observation is that with the increase of communication range, all the curves tend to converge even when increasing the speed of nodes. This behavior is explained by the fact that from a packet holder perspective, when the communication range increases largely, whatever the speed of the sender, there will not be much change in its neighborhood since it can already reach much neighbors within a single hop.

6 INTEGER in Delay-Tolerant Networks

Intermittently connected Delay-Tolerant Wireless Sensor Networks (ICDT-WSNs) are a new branch of Wireless Sensor Networks, which combine both characteristics of Wireless Sensor Networks and Delay-Tolerant Networks (DTNs). Their main characteristics include short communication range, narrow bandwidth, limited energy

and low computation capabilities in addition to the intermittent connectivity in which end-to-end paths between sources and destination do not always exist and if they do exist, most of the time are unstable and may break anytime during the routing process. Underwater WSNs, underground WSNs and Mobile WSNs –e.g. ZebraNet [44] – are well known examples of ICDT-WSNs. In our research, we are more interested in Mobile WSNs and their applications for monitoring individuals and animals. Thus, our focus will be devoted to Intermittently Connected Delay-Tolerant Mobile Sensor Networks (ICDT-MWSNs). The challenges of ICDT-MWSNs can be divided into two classes:

- **Link Challenges:** Intermittent and unpredictable connectivity, low or variable delay, asymmetric data rates and high error rates, sudden disconnection and link loss due to mobility of nodes.
- **Node Challenges:** Mobility of nodes, limited power, low processing capability, minimal storage, short communication range and low bandwidth.

In the very recent survey about geographic routing in DTNs [33], Cao et al. observed a real research vacancy in terms of proposed geographic protocols for DTNs in the literature despite being a very promising communication way in such intermittent networks. Only seven up-to-date geographic protocols have been reviewed in the literature representing only 11% compared to numerous topological ones [41]. Motivated by this lack of attention by the research community and our conviction that our proposal INTEGER suits well with delay-tolerant networks with intermittent connection scenarios, we propose to adapt INTEGER in the context of the delay-tolerant networks.

Cao et al. [33] identified several future directions that should be considered when designing geographic routing for ICDT-MWSNs that are already considered by INTEGER namely:

- *Handling the local maximum:* The local maximum problem happens when condition (21) is not fulfilled. In sparse networks, where opportunities to encounter adequate relay nodes are rare, this problem becomes more and more frequent resulting in drop of packets or long end-to-end delays. By adopting wisely the store-carry-forward strategy, INTEGER saves packets from being dropped by allowing the packet holder to keep the packet until finding a suitable relay.

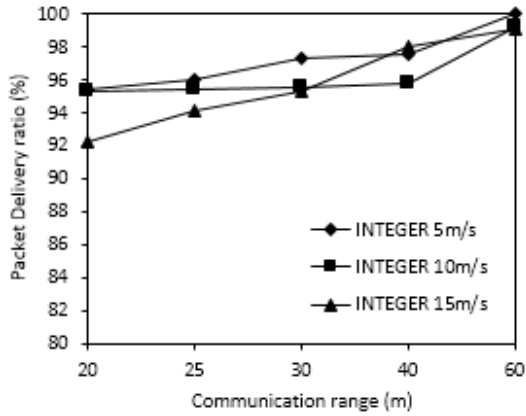


Fig.16 (a) Packet Delivery Ratio vs. Node Degree

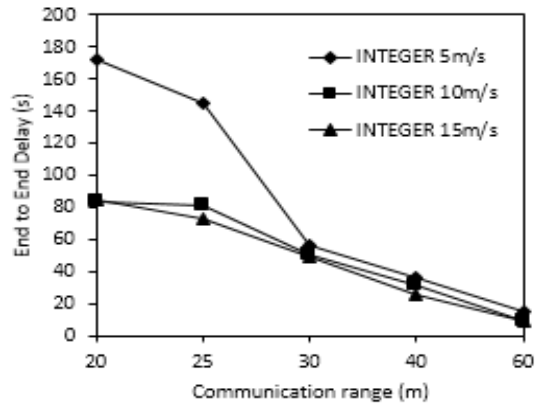


Fig.16 (b) End-to-End Delay vs. Node Degree

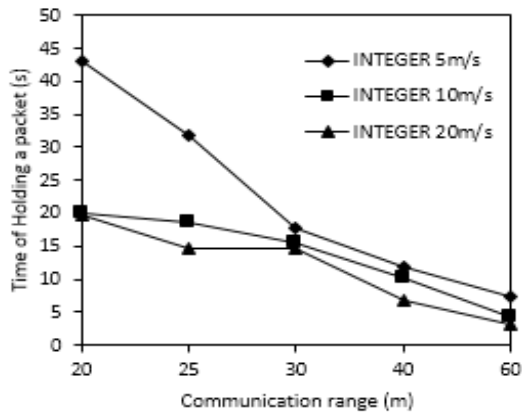


Fig.16 (c) Number of Relay nodes vs. Node Degree

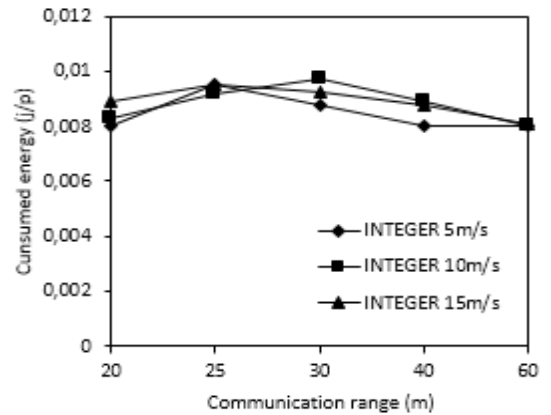


Fig.16 (d) Consumed Energy vs. Node Degree

- QoS consideration:* Knowing that frequent neighborhood discovery is energy costly and infrequent discovery may lead to missing many communication opportunities with neighbors that may be good candidates, INTEGER proposes an intelligent and mobility-based adaptive neighborhood discovery delay that maximizes the delivery ratio and minimizes the consumed energy.
- Assistance of additional infrastructure:* In INTEGER, the mobile anchor that moves with dedicated path serves basically as location reference for mobile nodes but can also serve as a message ferry. In fact, if the mobile anchor is a neighbor of the packet holder and is going in the direction of the destination, it will be favored to be a relay since it has the fastest speed and its location error is negligible. It can also bridge the communication gap between disconnected nodes.
- Combining MANET-based and DTN-based geographic routing:* This need comes from the variation of the network density in some application scenarios. In fact, when the network is dense, INTEGER applies the MANET communication mode by benefiting from more opportunities for the packet holder to choose the best forwarder among its numerous neighbors. In case of sparse network, INTEGER switches reactively to DTN mode by allowing the packet holder to keep the packet until meeting a suitable relay. Such intelligent switch allows reducing the packet end-to-end delay when being in MANET mode without the obligation to drop the packet when being in DTN mode.

Geographic routing protocols in DTNs have promising potential to be adopted by VANETs, UWSNs and ANs (Aeronautical Networks) scenarios because of their highly dynamic topologies. Because INTEGER responds to major DTNs characteristics, we aspire and look ahead to apply

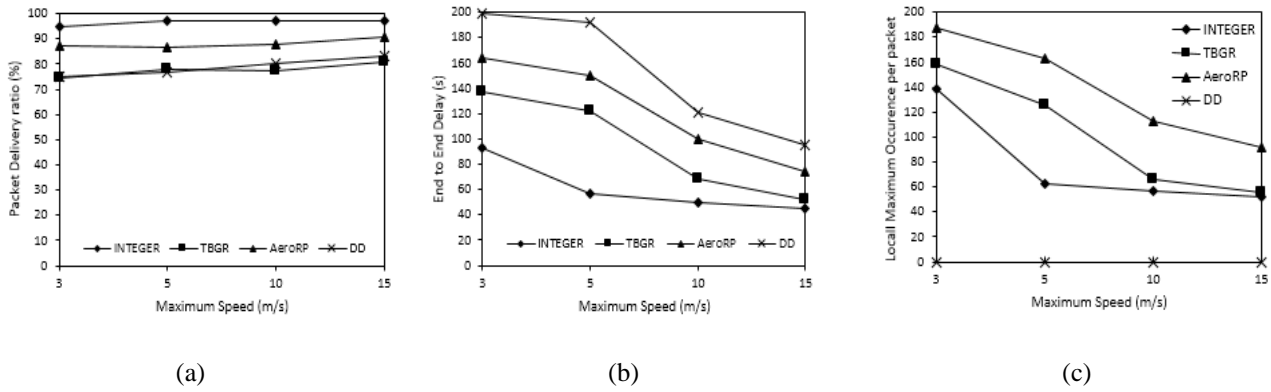


Fig. 17 Comparison with delay-tolerant protocols

INTEGER in such scenarios. To do so, we compare it with three other delay-tolerant geographic routing protocols, the very recent TBGR [35] with one copy mode, AeroRP [42] with the Ferry mode and DD [43]. We associate the re-forwarding strategy in case of collisions to all protocols. Note that the simulation scenarios and the simulation environment (NS-2) is the same with that of Section 5.1.

Since mobility of nodes is a significant challenge being one the main causes of intermittent connectivity, in the following tests, we vary the maximum speed of nodes in the set {3, 5, 10, 15}m/s to study its effect on the packet delivery ratio, the end-to-end delay and the local maximum occurrence per successfully delivered packet. Otherwise, the default values of Table 2 are applied.

Fig. 17 clearly shows that INTEGER outperforms other delay-tolerant protocols in terms of delivery ratio, end-to-delay and even local maximum occurrence. Both TBGR and AeroRP consider the velocity of nodes in the relay selection. Accordingly, we strongly believe that the favorable performance of INTEGER is thanks to the consideration of the localization error in the selection of the relay node in addition to its refined selection by considering the reliability and the QoS factors.

In terms of delivery ratio, Fig. 17 (a) shows that when the maximum speed of nodes is moderate (5m/s), the gain in packet delivery with INTEGER is up to 21% compared with DD, up to 18% compared with TBGR, and up to 10% compared with AeroRP. With the same speed, Fig. 17 (b) shows that INTEGER has the lowest latency which is explained by Fig. 17 (c) where the occurrence of local maximum problem is less for INTEGER than others. In this case, the time consumed while keeping the packet and discovering new neighborhood is saved by INTEGER.

The difference in the end-to-end delay between INTEGER and other protocols is important, up to 70% reduced compared to DD, 62% reduced compared to AeroRP and 54% reduced compared to TBGR. As for local maximum handling, INTEGER allows saving 60% and 55% compared to AeroRP and TBGR respectively as shown in Fig. 17 (c) while DD does not deal with this problem, as there is only one transmission between the sources and destination.

7 Conclusion

In this paper, we have proposed INTEGER, a novel geographic routing scheme intertwined with a localization algorithm specifically designed for mobile wireless sensor networks. The scope of this scheme is its location error-resilience nature despite being typically geographic. In fact, INTEGER efficiently exploits the mobility of nodes and considers not only the geographic location of nodes but also their speeds and directions in addition to their location error bound to select the best forwarder. These crucial data are obtained from SDPL method that jointly localizes mobile nodes by estimating their positions through predicting their mobility pattern.

To our knowledge, INTEGER is the first protocol that tackles localization and geographic routing for intermittently connected mobile sensor networks at the same time. The routing algorithm is composed of two main components namely the on-demand mobility-based adaptive neighborhood discovery and the best forwarder selection. The first consists on processing the discovery of neighbors by a node only if it has a packet to forward to save nodal energy. To do so, a mobility-based adaptive time interval has been proposed to maximize the delivery of packets considering different network parameters. As for the best next forwarder selection, we have proposed a new approach that allows choosing the most adequate forwarder by considering the position, the speed, the direction, the link reliability, and the localization error of neighbors when making routing decisions. If no neighbor fits the requirements or if the packet holder is isolated, INTEGER switches to the carry-and-forward strategy that it carrying the packet until finding a more suitable next forwarder. Thus, INTEGER could be well applied in delay-tolerant networks and intermittently connected networks.

Simulation results have shown the efficiency of INTEGER and its resilience to localization error. It also manages well the different speeds of mobile nodes from low speeds to high speeds thus could be applied in a large number of scenarios where the speed of nodes could vary or be very high such as monitoring wild animals. It also deals with

networks with high traffic by keeping the highest delivery ratio thanks to the strategy of relay re-selection in case of collisions. The adopted strategy of carry and forward has proven its efficiency by giving chance to packet holders to keep the packet until finding other forwarders instead of dropping it. We further compared INTEGER with delay-tolerant protocols. Results showed that INTEGER outperforms its competitors by handling efficiently the local maximum problem while reducing the end-to-end delay and increasing the packet delivery.

Due to limited bandwidth especially in intermittently connected networks, the selection of packets to forward is also of a great importance, a strategy to forward only a limited number of packets during a contact should be developed. In addition to previous metrics, considering also the limited encounter time between neighbors and the buffer size in selecting the next forwarder is being explored by our ongoing works.

References

1. Tunca, C., Isik, S., Donmez, M. Y., & Ersoy, C. (2014). Distributed mobile sink routing for wireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials*, 16(2), 877-897.
2. Gu, Y., Ren, F., Ji, Y., & Li, J. (2015). The evolution of sink mobility management in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 18(1), 507-524.
3. Sara, G. S., & Sridharan, D. (2014). Routing in mobile wireless sensor network: A survey. *Telecommunication Systems*, 57(1), 51-79.
4. Torrieri, D., Talarico, S., & Valenti, M. C. (2015). Performance Comparisons of Geographic Routing Protocols in Mobile Ad Hoc Networks. *IEEE Transactions on Communications*, 63(11), 4276-4286.
5. Huang, H., Yin, H., Luo, Y., Zhang, X., Min, G., & Fan, Q. (2016). Three-dimensional geographic routing in wireless mobile ad hoc and sensor networks. *IEEE Network*, 30(2), 82-90.
6. Cadger, F., Curran, K., Santos, J., & Moffett, S. (2013). A survey of geographical routing in wireless ad-hoc networks. *IEEE Communications Surveys & Tutorials*, 15(2), 621-653.
7. Oppermann, F. J., Boano, C. A., & Römer, K. (2014). A decade of wireless sensing applications: Survey and taxonomy. In *The Art of Wireless Sensor Networks* (pp. 11-50). Springer Berlin Heidelberg.
8. Popescu, A. M., Tudorache, I. G., Peng, B., & Kemp, A. H. (2012). Surveying position based routing protocols for wireless sensor and ad-hoc networks. *International Journal of Communication Networks and Information Security*, 4(1), 41.
9. Zhu, C., Yang, L. T., Shu, L., Leung, V. C., Rodrigues, J. J., & Wang, L. (2014). Sleep scheduling for geographic routing in duty-cycled mobile sensor networks. *IEEE Transactions on Industrial Electronics*, 61(11), 6346-6355.
10. Han, G., Xu, H., Duong, T. Q., Jiang, J., & Hara, T. (2013). Localization algorithms of wireless sensor networks: a survey. *Telecommunication Systems*, 52(4), 2419-2436.
11. Zhu, H., Zhong, X., Yu, Q., & Wan, Y. (2013). A localization algorithm for mobile wireless sensor networks. In *IEEE Third International Conference on Intelligent System Design and Engineering Applications (ISDEA)*, 81-85.
12. Sheu, J. P., Hu, W. K., & Lin, J. C. (2010). Distributed localization scheme for mobile sensor networks. *IEEE Transactions on Mobile Computing*, 9(4), 516-526.
13. Zhang, S., Cao, J., Li-Jun, C., & Chen, D. (2010). Accurate and energy-efficient range-free localization for mobile sensor networks. *IEEE Transactions on Mobile Computing*, 9(6), 897-910.
14. Zhou, Z., Peng, Z., Cui, J. H., Shi, Z., & Bagtzoglou, A. (2011). Scalable localization with mobility prediction for underwater sensor networks. *IEEE Transactions on Mobile Computing*, 10(3), 335-348.
15. Chenji, H., & Stoleru, R. (2013). Toward accurate mobile sensor network localization in noisy environments. *IEEE transactions on Mobile Computing*, 12(6), 1094-1106.
16. Cha, S., Talipov, E., & Cha, H. (2013). Data delivery scheme for intermittently connected mobile sensor networks. *Computer Communications*, 36(5), 504-519.
17. Benkhelifa, I., & Moussaoui, S. (2012). Speed and direction Prediction-based Localization for mobile wireless sensor networks. In *The 5th International Conference on Communications, Computers and Applications (MIC-CCA)*, 2012 (pp. 1-6). IEEE.
18. Benkhelifa, I., Lamini, C., Azouz, H., Moussaoui, S., & Khokhar, A. (2014). Prediction-based Localization for Mobile Wireless Sensor Networks. In *17th International Conference on Network-Based Information Systems* (pp. 257-262). IEEE.
19. Dyo, V., Ellwood, S. A., Macdonald, D. W., Markham, A., Trigoni, N., Wohlers, R., ... & Yousef, K. (2012). WILDSENSING: design and deployment of a sustainable sensor network for wildlife monitoring. *ACM Transactions on Sensor Networks (TOSN)*, 8(4), 29.
20. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
21. Othman, M. F., & Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. *Procedia Engineering*, 41, 1204-1210.
22. Seada, K., Helmy, A., & Govindan, R. (2004). On the effect of localization errors on geographic face routing in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (pp. 71-80). ACM.
23. Son, D., Helmy, A., & Krishnamachari, B. (2004). The effect of mobility-induced location errors on geographic routing in mobile ad hoc sensor networks: analysis and improvement. *IEEE Transactions on mobile computing*, 3(3), 233-245.
24. Kwon, S., & Shroff, N. B. (2006). Geographic routing in the presence of location errors. *Computer Networks*, 50(15), 2902-2917.
25. Milocco, R. H., Costantini, H., & Boumerdassi, S. (2014). Improved geographic routing in sensor networks subjected to localization errors. *Ad Hoc Networks*, 13, 476-486.
26. Saad, C., Benslimane, A., Champ, J., & König, J. C. (2008). Ellipse routing: A geographic routing protocol for mobile sensor networks with uncertain positions. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference* (pp. 1-5). IEEE.
27. Basagni, S., Nati, M., & Petrioli, C. (2008). Localization error-resilient geographic routing for wireless sensor networks. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference* (pp. 1-6). IEEE.

28. Petrioli, C., Nati, M., Casari, P., Zorzi, M., & Basagni, S. (2014). ALBA-R: Load-balancing geographic routing around connectivity holes in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(3), 529-539.
30. Li, J., & Shatz, S. M. (2008). Toward using node mobility to enhance Greedy-forwarding in geographic routing for mobile ad hoc networks. *Proc. of MODUS*, 1-8.
31. Kouah, R., Moussaoui, S., & Aissani, M. (2013). Mobility-based Greedy Forwarding Mechanism for Wireless Sensor Networks. In *Int'l Conference on Networking and Services* (pp. 140-145).
32. Cadger, F., Curran, K., Santos, J., & Moffet, S. (2016). Location and Mobility-Aware Routing for Improving Multimedia Streaming Performance in MANETs. *Wireless Personal Communications*, 86(3), 1653-1672.
33. Wang, T., Cao, Y., Zhou, Y., & Li, P. (2016). A survey on geographic routing protocols in delay/disruption tolerant networks. *International Journal of Distributed Sensor Networks*, 2016, 8.
34. Al-Shugran, M., Ghazali, O., Hassan, S., Nisar, K., & Arif, A. S. M. (2013). A qualitative comparison evaluation of the greedy forwarding strategies in Mobile Ad Hoc Network. *Journal of Network and Computer Applications*, 36(2), 887-897.
35. Cao, Y., Wei, K., Min, G., Weng, J., Yang, X., & Sun, Z. (2016). A Geographic Multi-Copy Routing Scheme for DTNs With Heterogeneous Mobility. *IEEE Systems Journal*.
36. Lee, E., Park, S., Park, H., Lee, J., & Kim, S. H. (2009). Geographic routing based on on-demand neighbor position information in large-scale mobile sensor networks. In *2009 International Symposium on Autonomous Decentralized Systems* (pp. 1-7). IEEE.
37. NS-2 Simulator, <http://www.isi.edu/nsnam/ns/>
38. Akyildiz, I. F., & Vuran, M. C. (2010). *Wireless sensor networks* (Vol. 4). John Wiley & Sons.
39. Benkhelifa, I., & Moussaoui, S. (2011). Appl: Anchor path planning based localization for wireless sensor networks. In *The 4th International Conference on Communications, Computers and Applications (MIC-CCA 2011)* (pp. 48-53).
40. Benkhelifa, I., Nouali-Taboudjemat, N., & Moussaoui, S. (2014). Disaster management projects using wireless sensor networks: An overview. In *IEEE 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2014* (pp. 605-610)..
41. Cao, Y., & Sun, Z. (2013). Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications surveys & tutorials*, 15(2), 654-677.
42. Peters, K., Jabbar, A., Cetinkaya, E. K., & Sterbenz, J. P. (2011). A geographical routing protocol for highly-dynamic aeronautical networks. In *Wireless Communications and Networking Conference (WCNC)*, (pp. 492-497). IEEE.
43. Grossglauser, M., & David, N. C. (2002). Mobility Increases the Capacity of Ad Hoc Wireless Networks. *IEEE/ACM TRANSACTIONS ON NETWORKING*, 10(4), 477.
44. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., & Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *ACM Sigplan Notices* (Vol. 37, No. 10, pp. 96-107). ACM.
45. Karp, B., & Kung, H. T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 243-254). ACM.
46. Kwon, S., & Shroff, N. B. (2006). Geographic routing in the presence of location errors. *Computer Networks*, 50(15), 2902-2917.
47. Peng, B., & Kemp, A. H. (2010). Impact of location errors on geographic routing in realistic WSNs. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, (pp. 1-7). IEEE.
48. Popescu, A. M., Salman, N., & Kemp, A. H. (2014). Energy efficient geographic routing robust against location errors. *IEEE Sensors Journal*, 14(6), 1944-1951.
49. Peng, B., & Kemp, A. H. (2011). Energy-efficient geographic routing in the presence of localization errors. *Computer Networks*, 55(3), 856-872.
50. Zhu, Y., Jiang, R., Yu, J., Li, Z., & Li, M. (2014). Geographic routing based on predictive locations in vehicular ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 2014(1), 137.
51. Valade, A., Acco, P., Grabolosa, P., & Fourniols, J. Y. (2017). A Study about Kalman Filters Applied to Embedded Sensors. *Sensors*, 17(12), 2810.
52. Ostertagová, E. (2012). Modelling using polynomial regression. *Procedia Engineering*, 48, 500-506.



Imane Benkhelifa is a PhD student at the Department of Computer Science at USTHB University of Science and Technology as well as a research associate at CERIST Research Center, Algiers. Imane holds both Master's degree in Mobile Computing from USTHB University and State Engineer Degree in Computer Systems from ESI National School of Computer Science. Her research interests include localization, routing, mobility management in wireless sensor networks, remote sensing using UAVs, Internet of Things and Smart Cities.



Samira Moussaoui is Professor of Computer Science at the University of Sciences and Technologies (USTHB) in Algiers. She received the Doctor degree in Computer Science from USTHB University in 2007. She is Lecturer Researcher in the Department of computer science of USTHB, since 1988. Her research interests include mobile computing systems, wireless sensor networks and vehicular networks.



Ilker Demirkol, IEEE Senior Member, is a Ramon y Cajal Research Professor in Department of Network Engineering at the Universitat Politècnica de Catalunya. His research focus is on communication protocol development and performance evaluation of wireless networks. Demirkol received his B.Sc., M.Sc., and Ph.D.

degrees in Computer Engineering from the Bogazici University, Istanbul, Turkey. He has held several positions both in academia and industry, such as Network Engineer, System and Database Consultant, and Lecturer. Demirkol is the recipient of the Best Paper Award in IEEE ICC 2013 and Best Mentor Award at Univ. of Rochester, NY, in 2010.