Greedy geographic routing algorithms in a real environment

Milan Lukic[1,2], Bogdan Pavkovic[1,2]
[1]Faculty of Technical Sciences
University of Novi Sad
Serbia
{mrmot021, bogprs}@gmail.com

Nathalie Mitton[2]
[2]INRIA/CNRS
Univ. Lille 1
France
nathalie.mitton@inria.fr

Ivan Stojmenovic[1,3]
[3]SITE
University of Ottawa
Canada
ivan@site.uottawa.ca

***Abstract*—Existing theoretical and simulation studies on georouting appear detached from experimental studies in real environments. We set up our test environment by using WSN430 wireless sensor nodes. To overcome the need for significant number of wireless nodes required to perform a realistic experiment in high density network, we introduce a novel approach - emulation by using relatively small number of nodes in 1-hop experimental setup. Source node is a fixed sensor, all available sensors are candidate forwarding neighbors with virtual destination. Source node makes one forwarding step, destination position is adjusted, and the same source again searches for best forwarder. We compare three georouting algorithms. We introduce here Greedy geographical routing Algorithms in a REal environment (GARE) which builds a RNG by using $\frac{ETX(uv)}{|uv|}$ as edge weight ($ETX(uv)$ counts all transmissions and possibly acknowledgments between two nodes until message is received), and selects RNG neighbor with greatest progress toward destination (if none of RNG neighbors has progress, all neighbors are considered). Our experiments show that GARE is significantly more efficient than existing XTC algorithm (applying RNG on $ETX(uv)$) in energy consumption. COP_GARE selects neighbor with progress that minimizes $\frac{ETX(uv)}{|uv|}$, and outperforms both algorithms.**

***Index Terms*—wireless sensor network; routing; emulation; realistic environment.**

I. Introduction

Wireless ad hoc networks [1] , especially sensor networks, have received a lot of attentions in recent years due to their potential applications in various areas such as monitoring, security and data gathering. However, they have some limitations compared to wired infrastructure networks. Energy consumption and scalability are two challenging issues when designing sensor network protocols such as routing protocols, since they operate on limited capacity batteries while the number of deployed sensors could be very large. In addition, they have to face new issues related to the radio medium which induces unpredictable effects due to interferences, collisions, fading and shadowing phenomena. Being one of the ”hot” research topics nowadays, the problem of routing in wireless ad hoc networks has been a subject of significant number of studies. Routing in wireless sensor networks is a challenging task. Many different approaches have been proposed in the literature. We can identify three main classes of routing protocols: *(i)* proactive routing such as OLSR [2] *(ii)* reactive routing such as AODV [3] and *(iii)* geometric routing, or georouting. This latter approach is a memory-less and scalable approach, unlike the two others. Although existing routing algorithms are developed to deal with real world problems, a relatively small number of them is actually implemented and tested on hardware. Most existing solutions assume that an ideal physical layer and adjustable range.However, in a real environment, the medium is not reliable and message retransmission has to be considered. In addition, available hardware sensor nodes can adapt their range but not dynamically at each single transmission. Our goal is to narrow the existing gap between theoretical and experimental results. We introduce GARE (Greedy Algorithm in Real Environment), which are greedy cross-layer geographical routing algorithms whose main feature is to take into account in their routing decision the state of the medium. COP_GARE is an adaptation of existing algorithm from [4]. GARE and COP_GARE integrate a measure of the quality of the communication links (denoted as ETX, counting all retransmissions until reception at desired neighbor node) in their routing decisions. They are scalable and memory-less (no information needs to be stored), loop-free and localized (only information about neighbor nodes is needed). Moreover, they aim at reducing the number of message retransmissions and thus to reduce energy consumption. We evaluate and analyze their efficiency through implementations over a network test-bed composed of WSN430 wireless nodes [5]. Experiments show that GARE and COP_GARE outperform existing solutions. In most scenarios, COP_GARE performs better among the two. This paper also contributes toward emulating large scale sensor networks with small number of available sensors, thus providing also human processing efficiency with small sacrifice in validating obtained results. While the implementation of XTC in [6] uses all available 7 or 33 sensors for the full network, we use them just to emulate a single hop. That is, one of sensors is the source while 6 or 32 others are its neighbors, and destination is at some virtual position. The source selects the best neighbor. After doing it successfully, the routing actually continues from the same source node, by adjusting destination position for virtual vector between selected neighbor and source node in the previous hop. The position of virtual destination could be also rotated around the source node to provide more randomness with respect to the neighborhood.The remainder of this paper is organized as follows. Section II reviews some of the existing georouting algorithms. Section III describes our concept of RNG based on new weight definition. GARE and GARE_COP are then detailed in Section IV. Then, Section V details experiment setup and results and how the emulation is conducted. Section VI concludes this work and investigates future works.

II. Related works

In *greedy* routing [7], S forwards the message to the node that is closest to destination D. Only neighbors closer to D are considered, otherwise forwarding fails. Greedy localized routing does not guarantee delivery, since a packet can be trapped in a local minimum. Simulations show that in dense networks the algorithm performs well with hop count as metric. However, several studies [8] have shown that greedy routing can perform poorly with power consumption as the metric. Nodes tend to choose the long links, to reach the destination in a minimum number of hops. Long links are less reliable as argued in [4], leading to an increase in retransmissions to reach particular neighbor.

Greedy georouting has then been enhanced in two directions, toward changing hop count to other metric, and toward providing guaranteed delivery [9], [10]. Power aware greedy routing algorithms were first studied in [11]. An algorithm with general cost metric was proposed in [12]. Cost over Progress based routing [12] (COP) is a localized metric aware greedy routing scheme. A node forwards a packet to the neighbor closer to destination D such that the ratio of the cost to reach neighbor to the progress made (measured as the reduction in distance to D) is minimized. Cost could be an arbitrary metric, such as (expected) hop count, power consumption, reluctance to forward packet, delay etc.

We make use here of a planar geometric structure called Relative Neighborhood Graph (RNG) [13]. A relative neighborhood graph with the distance metric is built as follows. The RNG of a graph $G = (V, E)$, is the set of all edges $uv \in E$ such that there is no vertex w such that $uw \in E$, $wv \in E$, $weight(uw) < weight(uv)$ and $weight(wv) < weight(uv)$. That is, RNG preserves links that are not the longest one in any triangle. To planarize a graph, RNG is used with the Euclidean distance as the weight. Planarization relies on the strong assumption of the Unit Disk Graph (UDG) [14], *i.e.* the transmission range area of a node is an exact disk. In a realistic environment, this assumption does not hold. Different pathologies during the planarization process in a realistic environment can be observed due to violation of unit disk assumption: partitioned planar subgraph, unidirectional links, crossing links. In [15] authors propose Cross-Link Detection Protocol (CLDP), which enables to overcome mentioned problems. Proposed protocol in combination with end to end geographical routing protocol GFG [9], [10] have been evaluated on real test bed consisted of up to 50 Mica2 nodes deployed in office building. The construction of needed structure however requires significant communication overhead.

Our work was motivated by the XTC algorithm [16]. XTC constructs RNG with the LQI (Link Quality Indicator) or RSSI (Received Signal Strength Indicator) as the edge weight. In our implementation of XTC, greedy routing is then applied over RNG links only. [6] implemented XTC with real sensors and described experiences. Authors first observed that packet loss increased drastically over time. They proposed then to use moving average as linear combination of current average and latest value. Their experiments show that received signal strength and link reliability are only loosely coupled. Moving average is then applied on packet loss to obtain a better link quality predictor. XTC is then applied to source routing on testbeds with 7 and 33 sensors. It prevents source routing from choosing unreliable links. It is not clear whether route discovery process was using acknowledgements, and whether sender node repeats transmissions. That is, actual protocol used is not specified.

The RNG based testbed used in [6] considers only close nodes and thus does not consider suboptimal links in terms of message consumption. The RNG graph in XTC is based on link qualities and prefers short links, thus making the routing path long in terms of number of hops needed to reach the destination. We will describe here a new weight definition of RNG link. Greedy routing in such RNG will select optimal paths from suboptimal links, instead of creating suboptimal paths from optimal links as in [6].

To cope with the changing nature of realistic radio channel, [17] propose an evaluation of hierarchical routing protocol HLR on a simple test bed of 10 Mica2 nodes.

III. ETX-based Relative Neighborhood Graph

We consider routing with retransmissions, such as in HHR [4]. In the HHR (Hop-by-hop retransmissions) model, a packet is retransmitted between two nodes until it is received and acknowledged correctly. Both the message and the acknowledgment are of equal length. In such a setup, it was convenient to use expected transmission count metric (ETX), introduced in [18], as a measure of link quality between two nodes. The ETX metric takes into account the effects of link lost ratios, and also detects and appropriately handles asymmetry by incorporating loss ratios in each direction. It is defined as the predicted number of data transmissions required to send a packet over that link, including retransmissions.

Because of static nature of our experimental environment, logical approach was to try to apply physical layer model introduced in [4], in order to determine ETX based just on distance between nodes, without any message exchange. This would consequently allow the computing of an optimal range (or a set of optimal ranges) for message transmission as in [19]. Unfortunately, preliminary experiments showed that expressing such a function is not feasible. It appeared that probability of reception is extremely dependent on orientation between the antennas (the difference in orientation of only few degrees makes a huge difference in link quality). For a reliable propagation model, the antennas would have to be omnidirectional, which was not the case with WSN430 nodes used for the experiments. We had to deal with ”potato” shaped radiation pattern instead of ideal circular shape. Thus, probabilities of reception had to be determined experimentally.

The authors of [6] construct an RNG graph using $ETX(uv)$ as a weight. This weight function favors high quality links (with a low ETX value), which are reliable and thus consume less energy to successfully transmit packets. For example, imagine simple testbed consisting of 11 nodes in straight line with distance of $1m$ between two successive nodes, as shown in Figure 1. The first node in row has to send (route) packet to the last one. In this topology, node currently holding the packet would choose next node in row (closest neighbor in direction to the destination) to forward the packet since it is the most reliable one. In this way, routing would be done by performing 10 consecutive high quality hops. The potential problem with this weight function is that the routing path becomes long in terms of total number of hops needed to reach the destination since RNG is only based on link qualities, thus preferring short links. This is not always energy efficient since a succession of a great number of short edges can require more energy than smaller number of less optimal hops [11].

To cope with this problem, GARE and COP_GARE use another weight function which chooses those links which have optimal ratio between link quality (ETX) and length traversed in one hop (distance between nodes). This way, less reliable links (higher ETX value) with greater length which consume more energy for one hop are used. From the aspect of overall routing, we expect a decrease in total number of hops needed to reach the destination and thus in the overall energy consumption. If we imagine the same simple test topology (Figure 1), this new approach will tend to use longer edges (green links on Figure) or even direct links to destination if possible, rather than just high quality links between closest neighbors (red links on Figure). The underlying idea behind this assumption is that such an approach would decrease overall hop count and thus overall energy consumption.

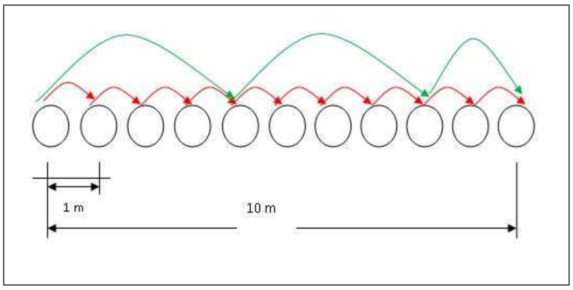


Fig. 1. Difference in routing strategy between GARE and XTC

GARE builds a RNG by using $\frac{ETX(uv)}{|uv|}$ as edge weight. For every triplet of nodes in graph, the edge with the largest weight is removed. This preserves edges with better cost $ETX(uv)$ over progress uv ratio, and in that way achieving more energy efficient routings.

Figure 2 shows an example of resulting graphs by applying RNG with that weight (Fig. 2(c)) on the original graph plotted on Fig. 2(a). Fig. 2(b) shows the resulting graph by using the ETX as weight [6].

For all experiments, we assume that nodes are aware of their position. This can be achieved thanks to a locating device such as GPS[1] or Galileo[2], or by any localization algorithms.

We consider a graph $G = (V, E)$ where V represents the set of sensor nodes in the network and there is an edge $e = (uv) \in E$ between nodes u and v if and only if a message from u to v can be received in some fixed number of attempts. We denote the set of neighbors of node u as $N(u)$, *i.e.* the set of nodes v such that $uv \in E$. Each edge $e \in E$ is assigned its ETX value $ETX(e)$. Applying an appropriate weight function over graph $G = (V, E)$ results in a reduced graph $RNG(G) = (V, E_{RNG})$. We denote by $N_{RNG}(u)$ the set of RNG neighbors of node u, *i.e.* the set of nodes v such that $uv \in E_{RNG}$.

IV. GARE and COP_GARE

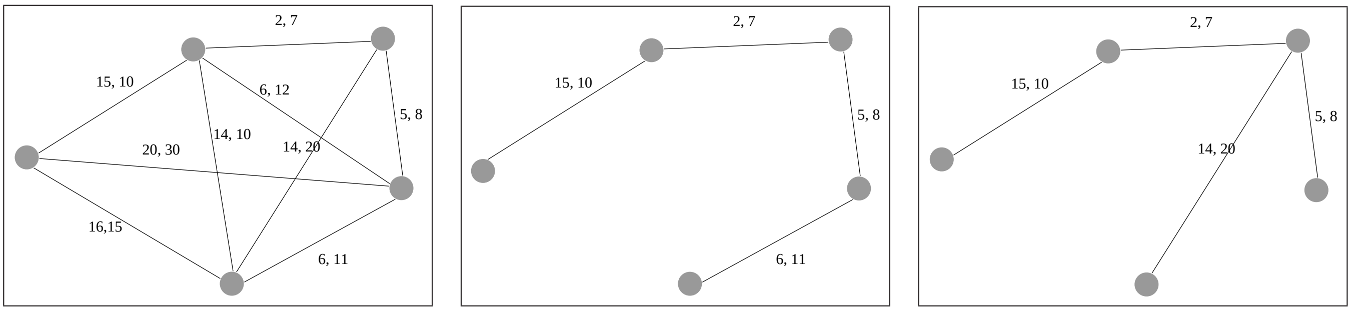
In this section we describe two greedy algorithms GARE and COP_GARE. The former one is inspired by [6] while the latter one is a variation of the COP algorithm [12] described in Section II, and is a general form of few algorithms described in [4]. They both use the ETX measure, a measure of the quality of the medium, in their decisions.

A. GARE

GARE is a greedy algorithm which makes the routing decision based on the progress. It is simple and localized since it needs only information about its 1-hop neighborhood. GARE runs over a ETX-based RNG graph, which is constructed by taking into account not just link quality, but also considering the link length. The RNG construction is local. It allows GARE to take into account the average state of the medium. The underlying goal pursued behind the use of a graph reduction based on ETX is to decrease overall hop count, messages retransmissions and thus overall energy consumption. Indeed, short links are the most reliable ones but some longer links, even if less reliable, still offer sufficient reliability. Since they are longer, the overall routing needs to follow less links and at the end spends less energy from end to end in retransmissions. Longer links induce less reliability but less hops on the final path. There is thus a trade-off between the length of links and the efficiency of path to follow.

In GARE, each node u currently holding the packet considers, among its RNG neighbors $N_{RNG}(u)$, the nodes v closer to the destination d than itself ($|vd| < |ud|$), *i.e.* nodes which provide a positive progress. Packet is forwarded to the node that is closest to the destination among RNG neighbors. If no RNG nodes with positive progress exist, routing will fail. We can overcome this with the following strategy. If no RNG edge with positive progress exists, the next node will be chosen among all other nodes, with the same criteria of minimal distance to destination. In this way, the routing fails if no node among neighbors is closer to the destination than source

[1] http://www.gps.gov/index.html
[2] http://ec.europa.eu/transport/galileo/index_en.htm



(a) Underlying graph (b) RNG with ETX as a weight (c) RNG with GARE weight

Fig. 2. Illustration of RNG with different edge weights. Labels on edges are arbitrary numbers for ETX and length, respec.

node itself. Other routing failure may be due to a failing link on which the packet has been lost. GARE is memoryless and scalable since no routing information is needed to be embedded in the message. It is also loop-free since at each step, since the message is always forwarded to nodes that provide a positive progress towards the destination.

GARE is formally described in Algorithm 1.

Algorithm 1 GARE(s, d) from source s to destination d

```
$u = s$
while not $u = d$ do
    build RNG by $\frac{ETX(uv)}{|uv|}$ as edge weight.
    $A \leftarrow N_{RNG}(u) \bigcap \{v\ s.t. |vd| < |ud|\}$
    {RNG neighbors with positive progress}
    if $A = \emptyset$ then
        $A \leftarrow N(u) \bigcap \{v\ s.t. |vd| < |ud|\}$
        {If no RNG neighbors with positive progress, consider every neighbor with positive progress.}
        if $A = \emptyset$ then
            {GARE failed}
        else
            $v \leftarrow p \in A$ which minimizes $|vd|$
        end if
    else
        $v \leftarrow p \in A$ which minimizes $|vd|$
    end if
    $u \leftarrow v$
end while
```

B. COP_GARE

COP_GARE algorithm also considers the state of the medium through the use of the ETX value. It is based on the following idea. Longer links induce less reliability but less hops on the final path. COP_GARE is based on the COP algorithm [12], described in Section II. It has been adjusted by using ETX as a cost function. In COP_GARE, the node u, currently holding packet, first selects its neighbors providing a positive progress towards the destination d ($\mathbf{N(u)} \cap |vd| < |ud|$). Among them, u selects the node v which minimizes the ratio of $ETX(uv)$ over the progress towards the destination (v minimizes $\frac{ETX(uv)}{|ud|-|vd|}$).

Algorithm 2 formally describes COP_GARE. Note that COP_GARE algorithm is general form of few algorithms described in [4]. In [4], few specific instances of that algorithm, corresponding to symmetric links between two nodes, with and without considering acknowledgements, were presented, with ETX metric being specifically replaced by corresponding values via packet reception probabilities, expressed as function of distance between two nodes. We extend here to values of $ETX(uv)$ measured from experimentation, and express the metric for the case of asymmetric links.

Algorithm 2 COP_GARE(s, d) from source s to destination d

```
$u = s$
while not $u = d$ do
    $A \leftarrow N(u) \bigcap \{v\ s.t. |vd| < |ud|\}$
    {neighbors with positive progress}
    if $A = \emptyset$ then
        {COP_GARE failed}
    else
        $v \leftarrow p \in A$ which minimizes $\frac{ETX(uv)}{|ud|-|vd|}$
        Return $\{v\}$
    end if
    $u \leftarrow v$
end while
```

V. Experiments

In this section, we describe our experiments the results we have obtained. We first describe our experimental set up. Then, we detail how we experimentally computed the ETX values on which every algorithm is based. We decided to evaluate our two algorithms by comparing them to XTC, the only algorithm from the literature with same assumptions. We analyze two values, the number of transmissions and the number of hops used along the routing paths.

A. Emulating large network with small network

Realistic experiment in wireless sensor networks may require a large set of devices available to construct a network of appropriate size. However, there might not be enough nodes available to conduct such an experiment. Time and human resources needed to maintain such a network is also a bottleneck. To cope with this problem we propose a novel approach. Instead of constructing a large scale network, we **emulate** such a network using a much smaller environment.

To emulate a high density network, we constructed a 1-hop neighborhood of node S currently holding packet with all our available nodes. Hence destination node D is not a real device but a virtual node positioned initially at distance much greater than radius of 1-hop neighborhood. Node S, which is always the same central node positioned in the coordinate origin, forwards packet to one of its neighbors (node B) which is selected by routing criterion. After this step has been successfully performed according to HHR model (acknowledgment message has been received at node S), virtual destination D changes its relative position according to node S. The current position of virtual destination is now translated by vector $-\vec{SB}$ to the new position, denoted as D', as illustrated by Fig. 3. Then, the routing continues using the same source node S again. This process is repeated until the position of virtual destination moves within the transmission range of source node S, which is considered as a successful routing, or when there is no node in the neighborhood suitable for routing, in which case the routing attempt is considered as a failure. Process can be explained as if destination is static. This does not fully correspond to realistic situation since the neighborhood of the transmitting node always looks the same, but it could be used as an appropriate model that allows us to emulate huge networks with large number of devices by using a relatively small number of nodes. It already gives a fair overview of different protocols and provides fair comparisons since various algorithms can run by using the same test bed and assumptions.

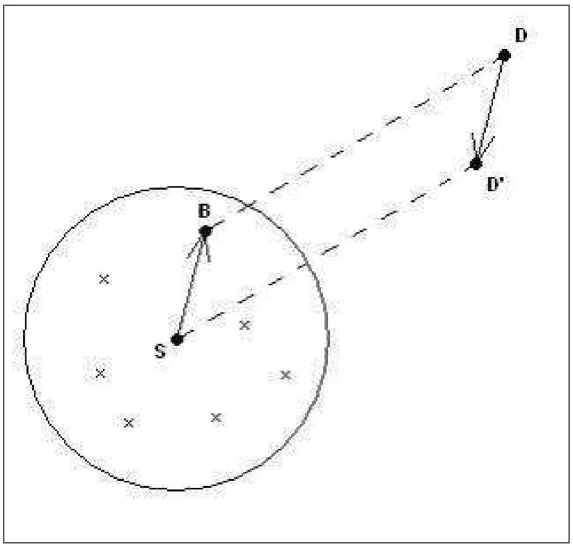


Fig. 3. 1-hop environment

We established 1-hop environment with total number of 42 WSN430 nodes [5]. WSN430 sensor node is ideal fit for low power, low cost, open source platform designed for deploying autonomous large scale sensor network applications. WSN430 sensor node is designed to operate at 315/433/868 and 915 MHz ISM/SRD band with very low power consumption allowing long life time and fast wakeup from sleep mode.

Node currently holding packet (same node all the time) is placed in coordinate origin while other 41 nodes are randomly and uniformly deployed within circle with radius of $5m$ as shown in Fig. 5(a) and 5(b). Topology was created using WSNet/Worldsens topology generator [20]. Radius for circle is chosen in such a fashion that the central node can reach all other nodes in one hop with constant transmission power. We have chosen to route to 16 virtual destinations placed around a circle with radius of $50m$. Between every two successive virtual destinations, there is an angular difference of 22.5 degrees as shown on Fig. 4.

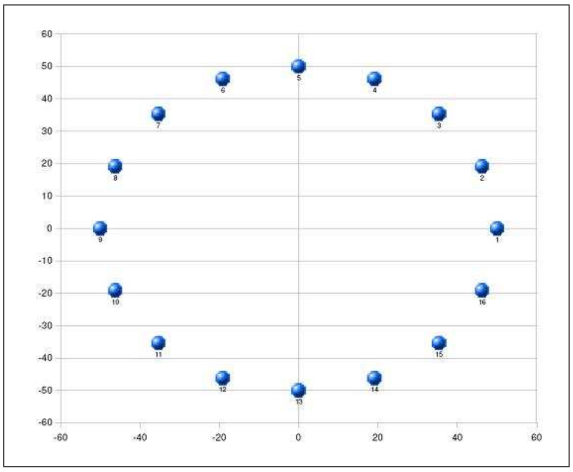


Fig. 4. Virtual destination placement

Our emulation consists of three steps:

1) Determining ETX as a metric of link quality, for every possible link in our 1-hop environment.
2) Constructing RNG based on one of the two different weight functions for comparing GARE (weight = $\frac{ETX(uv)}{|uv|}$) to XTC [16] (weight = $ETX(uv)$).
3) Running different routing algorithms using strategy described above, and evaluating performance statistics for each of them.

B. Determining ETX values

Version of the HHR protocol [4] that we used, assumes that node u sends the message and waits for acknowledgment from node v. If no acknowledgment is received by node u, the message is retransmitted again, until reception of proper acknowledgment. Thus, if $p(uv)$ is the probability that a message sent by u is properly received by v, and if we consider the link to be symmetrical, the probability that u receives an acknowledgment is $p^2(uv)$. In this case, we consider that expected number of retransmissions at node u is $\frac{1}{p^2(uv)}$, and we denote this value as $ETX(uv)$. We only take into account the number of transmissions at node u, while disregarding the number of acknowledgment attempts made by node v, in order to simplify the analysis. In case the links are asymmetrical ($p(uv) \neq p(vu)$), we consider that $ETX(uv) = \frac{1}{[p(uv)*p(vu)]}$. This perhaps makes the above mentioned simplification even more justified, because this definition of ETX eliminates effects of link asymmetry.

One of the important things we have noticed during our failed attempts to verify physical layer model introduced in [4], is that in case the nodes are static and there are no moving objects between them, the probability of reception does not change significantly over time. This property allows us to evaluate probabilities separately for each link only once in the initial phase of the experiment, and then to use that knowledge later on. No supplementary link quality evaluation is needed since links are stable.

For this part of experiment, we have set up a test bed consisting of 42 WSN430 nodes. The nodes were set to transmit 8 series of 128 messages. In every series, all nodes were transmitting 128 messages successively one node at a time, while the other nodes were counting the number of received messages from every node separately. Total number of messages sent by each node was 8×128 = 1024. After we have read the statistics from referent node (central node S), we have calculated standard deviation in number of messages received in separate series, from each of the remaining nodes. Mean standard deviation for all nodes was 4.61%, which justifies our point about the link stability. From these measurements we compute:

$$p(uv) = \frac{MSG_{u\leftarrow v}}{1024}, \qquad (1)$$

where $MSG_{u\leftarrow v}$ stands for the number of messages node v has successfully received from node u,

$$p(vu) = \frac{MSG_{v\leftarrow u}}{1024}, \qquad (2)$$

and we deduce

$$ETX(uv) = \frac{1}{[p(uv) \times p(vu)]}. \qquad (3)$$

C. Construction of RNG

To compare XTC, GARE and COP_GARE, two different weight functions are used for generating RNG graphs:

1) $w_1 = ETX(uv)$.
2) $w_2 = \frac{ETX(uv)}{|uv|}$.

Weight function (w_1) was chosen as analogy to one used to construct RNG in [6]. w_2 is the weight used by GARE.

Figure 5(a) shows RNG edges that include central node, obtained by using weight function w_1, while Figure 5(b) shows RNG edges obtained by using w_2.

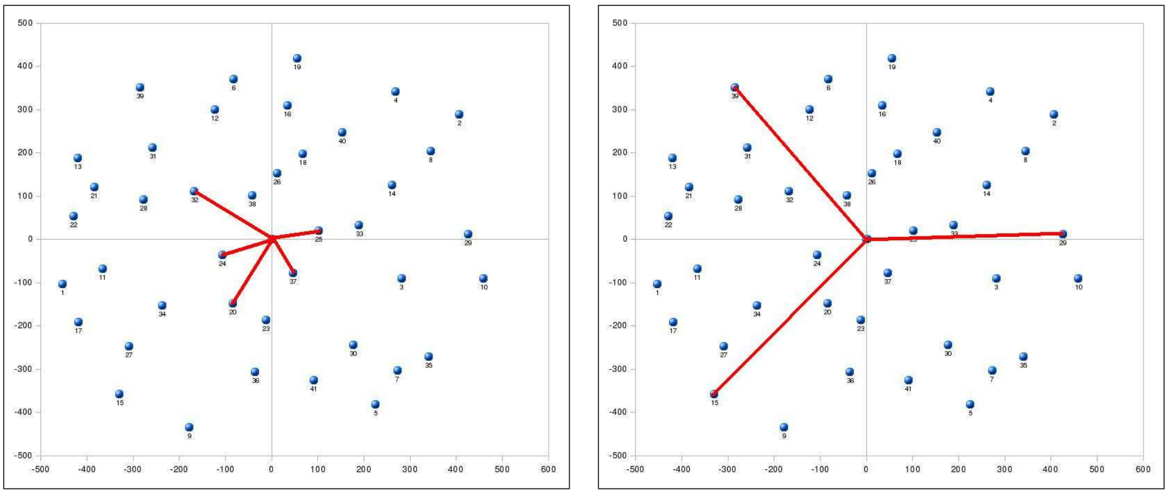
D. Experimental results

We compared the performance of three different routing protocols (GARE, COP_GARE and XTC) by performing series of emulations in 1-hop environment. We measured transmission count and hops for one successful routing. Transmission count represents the total number of messages needed to reach destination according to HHR model (message delivery with acknowledgment) and hop count represents the total number of traversed nodes to perform the same task. Results have been averaged over 100 performed routings to every of 16 virtual destinations. Fig. 6(a) and 6(b) show graphical comparison of performances of algorithms, regarding needed number of transmissions and hop counts per routing.

Success rate of all algorithms is 100% due to high node density of emulated network. Sender node degree is 41 (with the source being in the center, the effective degree, which is number of neighbors closer to current virtual destination, was always about 10 which was dense enough to make progress), and neighbors are uniformly positioned, so the routing algorithms were always able to find a node with positive progress.

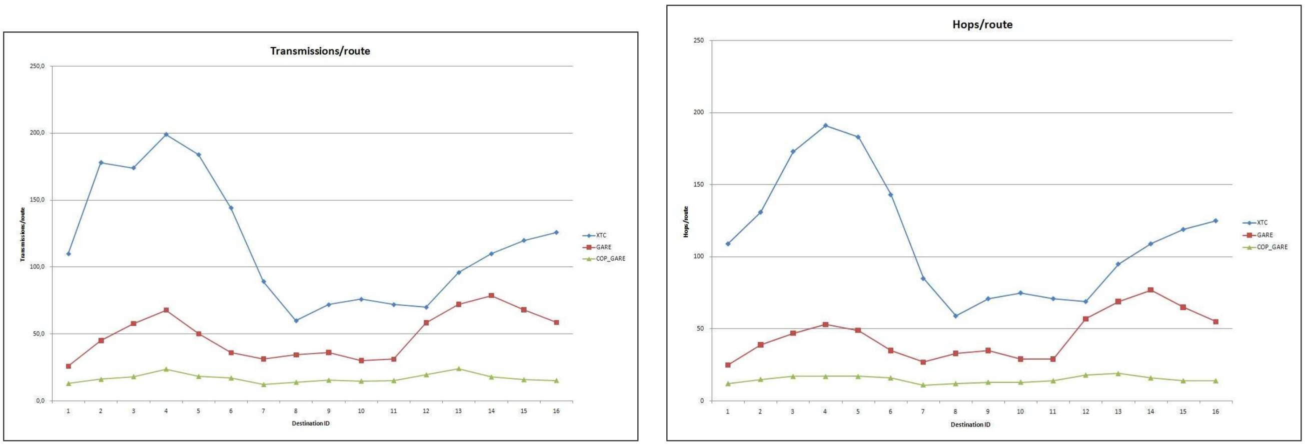
GARE achieves better results than XTC, which is roughly the same algorithm run over a different RNG. It uses fewer transmissions and hops to successfully reach various destinations and since every transmission uses same energy amount, it uses less energy for single routing. Since we assume that nodes cannot adjust their range, every transmission has the same energy consumption. Thus, under these conditions, since XTC induces more retransmissions, it is the algorithm which spends the most energy. GARE is on average case around 54% more efficient than XTC in transmission count, hence total energy usage for one successful routing. The reason is that XTC needs to perform more hops since it chooses only short links when longer ones are sufficiently reliable to be followed.

COP_GARE (marked with yellow color in figures) outperforms both GARE and XTC. It does not use RNG edges to make routing decisions; rather it has freedom in every hop to choose edge with the best cost over progress ratio. COP_GARE scored on average 62% better results than GARE. This can be explained as follows. GARE prefers RNG edges, but unfortunately there is a relatively small number of them (in our case, we have only 3 edges, as you may see from Figure 5(b)). So, while COP_GARE always chose the best edge among all possible edges, GARE is by its nature forced to use RNG edges which might have relatively small progress, increasing the total number of used nodes needed for routing. This point is supported by our experimental results, where we might notice that gap between GARE and COP_GARE performances is shrinking in cases when our virtual destination is in general direction of RNG edges (destinations 1, 7 and 11). This can be observed from Fig. 6(a) and 6(b). Obtained results for hop and transmission counts considerably vary in the case of GARE routing depending on the position of virtual destination. This can be explained as follows. RNG edges are not distributed uniformly in all directions. In cases of some virtual destinations (for example 3, 4 and 5, Fig. 6(a) and 6(b)) no RNG edge exists in their general direction so central node is forced to make poor routing decisions based on progress only, and disregarding the link quality, which consequently results in greater total length of route (more nodes and needed to reach destination). On the other hand, COP_GARE routing appears not to suffer from this problem because of its nature to always choose the best solution amongst all of the neighboring nodes, regarding ETX and progress. Same parameters vary very slightly around mean value. In arbitrary networks, it is better to use COP_GARE while in situations where RNG edges are



(a) $w1 = ETX(uv)$ (b) $w2 = \frac{ETX(uv)}{|uv|}$

Fig. 5. Emulation topology and RNG edges.



(a) Transmission Count (b) Hop Count

Fig. 6. Comparison of algorithm performances

well distributed, less nodes will be assailed using GARE for achieving similar results than when using COP_GARE.

During the course of our experiments, we have noticed one potential conceptual flaw in experimental strategy. When routing is performed towards some virtual destination, RNG graph tends always to preserve approximately the same orientation along the same route, which would of course not be the case in a real network. Thus, we have performed another series of experiments with slightly altered strategy. During each routing attempt, first we put virtual destination on random position, but with fixed distance from the source node ($50m$, same as in previous series of experiments). Then, we route using one of the algorithms, and after each hop we update the position of virtual destination as we did before. The key difference is that before we perform the next hop, the position of virtual destination is rotated by random angle around the central (source) node. In this way, we randomize the orientation of RNG graph in each hop, while preserving the distance from source. We have performed 100 routings for each algorithm using this altered strategy, and the results are shown in Table I.

Routing algorithm	Hit rate [%]	Nb.transm. (total)	Nb.hops (total)
XTC	100	6196	5820
GARE	100	4948	3103
COP_GARE	100	2162	1536

TABLE I
Comparison of algorithm performances with randomized graph orientation

We notice that even with these changes we have made to make emulations closer to a real case, the relationships between performances of routing algorithms are roughly the same. GARE is shown to be more efficient than XTC, while COP_GARE outperforms them both in most of situations.

VI. Conclusion and Perspective

In this paper, we have introduced two novel position-based greedy routing algorithms for realistic environment: GARE and COP_GARE. We also proposed a novel approach for evaluation in a real environment to overcome a small number of available hardware nodes. It is to setup test bed for measuring one hop performance repeatedly with using mobile virtual destination, instead of setting the whole network physically. GARE and COP_GARE outperform XTC in our test bed.

We intend to push further our experimentations. Based on the first feedbacks of these experiments, we will try to consider other physical parameter in routing decisions. Then, we intend to investigate algorithms for real environments that guarantee the message delivery to the final destination.

Our test bed has source node in the center, leading to approximately 10 nodes as neighbors closer to virtual destination. If we place the center in one of corners, we could emulate density about 40 with same sensors. Further, we observed that long edges were preferred in our RNG construction. To make more reliable conclusions, the distances between sensors should be increased so that RNG edges are approximately between short and long edges. Another option is to change the weight definition in RNG graph. We will consider more options. For instance, $w(uv) = ||uv| - r*|$, where $r*$ needs to be determined, and is expected to be around 10m. The same weight could be also defined to better reflect the approximate slope away from the optimal distance. We also intend to repeat experiments with different type of sensors, to investigate the impact of actual hardware characteristics. We plan to add some more algorithms for comparison, such as few more described in [4] (one of them maximizes the product $p(x)||vx| - |ux||$ of packer reception probability and advance toward the destination. Further extension is planned toward adding power consumption and network lifetime into the cost routing metric [21].

Acknowledgment

This research is partially supported by NSERC Discovery grant and NSERC Strategic Grant on sensor and actuator networks STPGP 336406-07.

References

[1] J. Wu and I. Stojmenovic, "Ad hoc networks," *IEEE Computer*, vol. 37, pp. 29–31, 2004.

[2] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol (OLSR)," 2003, RFC 3626.

[3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-demand Distance Vector Routing," 2003, RFC 3561.

[4] J. Kuruvila, A. Nayak, and I. Stojmenovic, "Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer," *IEEE Journal of Selected Areas in Communications (JSAC)*, vol. 23, pp. 1267–1275, 2005.

[5] A. Fraboulet, G. Chelius, and E. Fleury, "Wsn430, http://worldsens.citi.insa-lyon.fr/," 2007.

[6] N. Burri, P. von Rickenbach, R. Wattenhofer, and Y. Weber, "Topology control made practical: Increasing the performance of source routing." in *2nd Int. Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2006.

[7] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Information Sciences Institute (ISI), Tech. Rep. ISI/RR-87-180, 1987.

[8] E. H. Elhafsi, N. Mitton, and D. Simplot-Ryl, "End-to-End Energy Efficient Geographic Path Discovery With Guaranteed Delivery in Ad hoc and Sensor Networks." in *Proc. IEEE PIMRC*, Cannes, France, 2008.

[9] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. DIAL-M*, WA, USA, 1999, pp. 48–55.

[10] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *Proc. 12th Annual international conference on mobile computing and networking (MOBICOM)*, CA, USA, 2006.

[11] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE Trans. Paral. Distrib. Syst.*, vol. 12, no. 11, pp. 1122–1133, 2001.

[12] I. Stojmenovic, "Localized network layer protocols in sensor networks based on optimizing cost over progress ratio," *IEEE Networks*, vol. 20, pp. 21–27, 2006.

[13] G. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

[14] B. Clark, C. Colbourn, and D. Johnson, "Unit disk graphs," *Discrete Mathematics (DISCMATH)*, vol. 86, no. 1–3, pp. 165–177, 1990.

[15] Y.-J. Kim and al., "Geographic routing made practical," in *Proc. NSDI 05: 2nd Symposium on Networked USENIX Association ed Systems Design and Implementation*, 2005.

[16] R. Wattenhofer and A. Zollinger, "XTC: a practical topology control algorithm for ad hoc networks," in *Proc. 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, MN, USA, 2004.

[17] G. Fang Bian, Xin Li Ramesh and S. Shenker, "Using hierarchical location names for scalable routing and rendezvous in wireless sensor networks," UCLA, Tech. Rep., 2005.

[18] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing." in *Proc. 9th Annual international conference on mobile computing and networking (MOBICOM)*, 2003.

[19] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, "Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 7, pp. 536–547, 2006.

[20] A. Fraboulet, G. Chelius, and E. Fleury, "Worldsens: Development and prototyping tools for application specific wireless sensors networks," in *IPSN'07 Track on Sensor Platforms, Tools and Design Methods (SPOTS)*. Cambridge, Massachusetts, USA.: ACM, April 2007.

[21] J. Kuruvila, A. Nayak, and I. Stojmenovic, "Progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks," in *Proc. Third Int. Conf. on AD-HOC Networks and Wireless ADHOC-NOW* . Vancouver, BC, Canada: Lecture Notes in Computer Science 3158, 2004, pp. 294–299.