Efficient Non–Planar Routing around Dead Ends in Sparse Topologies using Random Forwarding

Paolo Casari†, Michele Nati*, Chiara Petrioli*, Michele Zorzi†

†Department of Information Engineering — University of Padova — Via Gradenigo 6/B, I-35131 Padova
*Department of Computer Science — University of Rome “La Sapienza” — Via Salaria 113, I-00198 Rome
E-mails: {casarip,zorzi}@dei.unipd.it, {nati,petrioli}@di.uniroma1.it

Abstract—Geographic forwarding in wireless sensor networks (WSN) has long suffered from the problem of bypassing “dead ends,” i.e., those areas in the network where no node can be found in the direction of the data collection point (the sink). Solutions have been proposed to this problem, that rely on geometric techniques leading to the planarization of the network topology graph. In this paper, a novel method alternative to planarization is proposed, termed ALBA–R, that successfully routes packets to the sink transparently to dead ends. ALBA–R combines nodal duty cycles (awake/asleep schedules), channel access and geographic routing in a cross–layer fashion. Dead ends are dealt with by enhancing geographic routing with a mechanism that is distributed, localized and capable of routing packets around connectivity holes. An extensive set of simulations is provided, that demonstrates that ALBA–R is scalable, generates negligible overhead, and outperforms similar solutions with respect to all the metrics of interest investigated, especially in sparse topologies, notoriously the toughest benchmark for geographic routing protocols.

I. INTRODUCTION

DISTRIBUTED sensing and seamless wireless data gathering are very attractive features for any monitoring application. By deploying a large number of inexpensive wireless sensor nodes (or motes), Wireless Sensor Networks (WSNs) have provided the unifying solution for both tasks: nodes perform their data harvesting duties unattended, and report back to a data collection point (called sink) via a multi-hop wireless path. Recent research has widely dealt with the definition of protocols for WSNs focusing mostly on MAC and routing layers [1]. Great interest has been devoted to geographic or location-based routing, where a node that has a packet to transmit selects a relay to send it to that greedily determines a positive advancement of the packet toward the sink. In other words, a relay (geographically) closer to the sink is chosen. Being almost state-less, fully distributed and localized geographic routing imposes little computation and storage needs on the sensor nodes. Current geographic routing schemes, however, fail to fully address the three design challenges concerning i) routing around connectivity holes (or dead ends), ii) resilience to localization errors, and iii) efficient relay selection. Connectivity holes are inherently related to the way greedy forwarding works. For instance, consider a connected network topology. In greedy forwarding relays are selected only among neighbors (eligible relays) that can advance the packet toward the sink. If a node does not have a neighbor closer to the sink it will ultimately discard all the packets that it generates or receives. This node is called a dead end, and it generates a connectivity hole. Solutions proposed to deal with dead ends either resort to making the network topology planar (in the sense that there are no links crossing each other, and it is always clearly possible to establish a sense of direction) or are able to only partially alleviate the impact of connectivity holes. Planarization and the related “face routing,” which is the only approach able to eventually guarantee packet delivery to the sink, do not work well in the presence of localization errors or in all those realistic scenarios in which the network topology cannot be modeled by a unit disc graph.

In this paper, we approach the problem of dealing with dead ends in a novel way that allows us to guarantee the delivery of packets to the sink without requiring the overhead and the inaccuracies incurred by “planar” methods. Our solution, termed ALBA–R for Adaptive Load-Balanced Algorithm, Rainbow version, is a simple, distributed scheme that is remarkably resilient to localization errors and independent of whether the network topology is modeled by a unit disk graph or not. The design of ALBA–R follows recent trends in geographic routing protocol design that demonstrated that remarkable performance improvements can be obtained by applying cross–layer techniques [2], [3] and [4], [5]. The availability and exchange of information among different possibly non-adjacent protocol layers, as well as the direct integration of the functionalities of multiple layers into a single one, allow a node to make more effective routing decisions based on a “wider view of the network” (more information is available), and to operate so to optimize the joint performance of the different layers. In particular, ALBA–R integrates MAC and routing design. Whenever a node has to forward a packet (a typical network layer duty) all its neighbors are addressed by a relay selection message (RTS-like, to use IEEE 802.11 terminology) which initiates a competition for electing the “best” next hop relay. Each eligible node can locally compute its own suitability to serve as a relay, based on a cross–layer parameter that reflects its current status (e.g., packet error rate, transmission bit rate, residual energy, current queue occupancy, capability of fast and reliable packet forwarding, and combinations thereof) and participates to the competition. The relay node is thus elected based on the value of this parameter that is based on values and methods typical of the PHY, MAC and routing (network) layers combined. This node then receives the data packet, and forwards it to the sink (if the sink is now directly reachable) or to the next best relay. As mentioned, a problem with this simple and very efficient mechanism occurs when a node is not able to find a relay, i.e., it is a dead end. The purpose and the contribution of our paper is to describe and demonstrate the “rainbow” (R) component of ALBA–R that obviates to
the dead-end problem and guarantees that every message is eventually delivered to the sink. The colorful name (Rainbow) derives from the way nodes are labeled with different colors, yellow being the color of nodes that lead a message to the sink using simple greedy forwarding (the road to success, or “yellow brick road”). Different colors indicate more or less suitability in being a relay toward the sink.

Extensive ns2-based simulation results demonstrate the effectiveness of ALBA–R with respect to relevant metrics of interest, which include packet delivery ratio, end-to-end packet latency, energy consumption, and overhead. We observed that ALBA–R is always able to deliver all packets to the sink, independently of network density. It is able to satisfactorily function at medium-high traffic loads, it imposes little overhead, and the nodal energy consumption is never higher than 40% of the energy consumption of nodes following the nominal duty cycle.

The remainder of the paper is organized as follows. In Section II we describe previous work on geographic routing, focusing on GeRaF and ALBA, two greedy forwarding protocols based on random selection of the next hop. Section III takes care of describing the details of the Rainbow component of ALBA and its correctness. ALBA–R is evaluated via ns2 simulations in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

Many routing techniques have been designed with specific WSN scenarios in mind. According to the taxonomy presented in [1], [6] these techniques can be classified as hierarchical or flat. In hierarchical schemes, nodes are divided into ordinary nodes and backbone nodes by executing a distributed clustering and backbone formation algorithm. Only the backbone nodes are involved in route setup and maintenance. The second class of routing protocols, the so called flat approaches, does not make any distinction in the role played by the nodes. All nodes are involved in routing and forwarding. The routing protocols differ for the criteria used to select the routes. Flat schemes can be QoS-aware, can favor multipath routing, can select relays so as to minimize the overall energy consumption, and so on.

Among the flat routing approaches recently proven to be particularly well suited to the energy conservation needs of WSN, geographic routing is certainly one of the most promising. The reasons are mainly to be found in the low routing overhead, especially due to the fact that full-scale route discovery, typical of many reactive ad hoc routing protocols [7], [8], is not needed in the case of geographic routing. The only thing that is needed is that each node is aware of its own location, a piece of information that can be obtained either via extra hardware (e.g., GPS) or via a distributed localization protocol. Since WSNs are typically static and the communication is from the sensors to a few sinks (and viceversa) the overhead associated to localization is almost negligible. The typical geographic routing protocol works like this. Once each node has an estimate of its own position, a rule (usually greedy) is used to choose the next hop relay. The neighbor that leads to the maximum advancement toward the sink is typically selected. It has been shown that greedy forwarding works very well in dense topologies, because of the greater number of relays in the direction of the sink. However, it has also been observed that the rate of successful packet delivery decreases in sparser topologies, where dead ends may occur more frequently: A packet is relayed to a node that has no neighbors in the direction of the sink (a dead end), and as such it is discarded, thus detrimentally affecting the protocol and network performance.

Starting from early work by Stojmenovic et al. [9], and then by Karp and Kung [10], solutions have been proposed to deal with dead ends by means of making the network topology graph planar and by using “face routing.”

The network topology is usually modeled as a graph \( G = (\mathcal{N}, \mathcal{E}) \), where \( \mathcal{N} \) is the set of nodes and \( \mathcal{E} \) is the set of edges, i.e., pairs \((i, j)\) identifying the link joining nodes \(i, j \in \mathcal{N}\). According to the unit-disk graph model, the edge \((i, j)\) is assumed to be in \( \mathcal{E} \) if and only if the Euclidean distance between nodes \(i\) and \(j\) is less than or equal to the node transmission range. Solutions proposed in [9] and [10] first perform planarization to remove links crossing each other. Planarization generates a new graph \( G' = (\mathcal{N}, \mathcal{E}') \), with \( \mathcal{E}' \subseteq \mathcal{E} \). The most used planarization algorithms are the Reduced Neighbor Graph (RNG) and the Gabriel Graph (GG) [10]. In both algorithms, a pair of nodes with an edge in \( \mathcal{E} \) check whether a third node (or witness) is located inside the intersection of their coverage areas, or inside part of it. If a witness is detected, the link between the inquiring nodes is removed. The resulting planar graph can be seen as a set of (possibly concave) polygons sharing some faces. GPSR, for instance, computes the planar graph. Whenever a connectivity hole is found, GPSR forwards packets along the faces of the polygons in \( G' \), changing face whenever a link would cross the line connecting the source and the destination [10]. Whenever the connectivity hole is bypassed, basic greedy forwarding operations are resumed. Other contributions to face routing are reported in [11], where adaptive worst-case optimal face routing algorithms are proposed.

These solutions are effective in bypassing dead ends. However, they rely on the assumption that the network topology graph is a unit disk graph. This is seldom the case in realistic scenarios due to the way radio signals propagate. Realistic radio coverages and localization errors could potentially hamper planarization techniques, preventing their correct operations. Another substantial drawback of face routing is the difficulty of updating the planar graph in case of network changes. Nodal mobility and/or nodes alternating between awake and asleep state for energy conservation lead to frequent topology changes, requiring to recompute the face links and resulting in higher overhead.

All these problems have been investigated in details in [12]. In particular, their impact on the needed planar structure of the network topology has been evaluated. Results show that partitions and cross links may arise in the planar graph, which render face routing useless. The solution proposed in [12], termed mutual witnessing, allows two neighbors to remove the link between them only if both can sense the presence of a certain witness. This process, however, requires even more exchanges of control packets among neighbors, and therefore induces high overhead.
A different approach to location-based routing is based on random geographic forwarding. Instead of removing links for making the network planar, random forwarding exploits the “route diversity” offered by multiple neighbors by selecting as relay a node offering sufficient gain with respect to a given metric of interest. We have recently proposed some approaches specifically designed based on this concept. GeRaF [4] and ALBA [13] are flat, location-based, cross-layer routing protocols for WSNs which exploit local node status information together with geographic locations for converge-casting packets toward the sink. However, although quite efficient with respect to data latency and generated overhead, neither GeRaF [4] nor ALBA [13] are able to cope with dead ends. Furthermore, in sparse scenarios they both experience severe performance degradation.

This paper aims at making random geographic forwarding resilient to dead ends. We extend ALBA [13] with a fully distributed and localized mechanism that enables it to route packets around connectivity holes. The resulting protocol, ALBA-R, retains the best features of the solutions of this class of geographic routing protocols, i.e., it remains as simple, localized, fully distributed as GeRaF and ALBA. As GeRaF and ALBA, ALBA-R does not require planarization or any other overhead-generating operations. Differently from previous solutions, however, it is fully capable of dealing with dead ends.

In the following we briefly describe GeRaF [4] and ALBA [13] since they are the two schemes ALBA-R is built upon.

A. Geographic Random Forwarding (GeRaF)

The detailed operation of GeRaF is as follows. Any node operates according to a duty cycle with parameter $d$. Nodes awake/asleep schedules are asynchronous. The forwarding area of each node is divided into $N_r$ regions, 1 through $N_r$, such that any node in region $i$ is closer to the sink than any node in region $j$, $\forall i < j$. Whenever a node wants to transmit a packet it first senses the channel sense for a time long enough to detect ongoing handshakes. It then sends a Request-to-Send (RTS) message that silences nodes offering negative advancement and serves as a polling message for region 1. The awake nodes in this region report back using a Clear-to-Send (CTS) message. If more than one respond, the sender issues a COLLISION message to solicit the choice of a single node. Each node chooses whether to respond to the COLLISION packet with probability $0.5$. If multiple relays respond, only those nodes which have participated to the round (i.e., they have answered the CTS) toss the coin again. The process eventually terminates, determining the next relay. If region 1 is empty (e.g., because all nodes located therein are sleeping) the transmitter sends a CONTINUE packet to poll the following region. When the first nonempty region is found, a contention among nodes is originated as specified before. After the identification of the next hop, GeRaF sends a data packet, waits for an ACK, and then lets the sender go back to sleep, so that the relay finds the channel free and can forward the packet. If no relay is found in any region or the channel is sensed busy, the transmitter backs off and reschedules a later attempt. Note that, in GeRaF, nodes are not required to know the position of any other sensor but their own and the sink’s, since any other information is exchanged through RTSs and CTSs. Also nodes do not need to know their neighbors and their wake up schedule: a relay is selected among the awake neighbors. These two features are also typical of ALBA and ALBA-R.

B. Adaptive Load–Balanced Algorithm (ALBA)

ALBA is a greedy forwarding protocol for WSNs first introduced in [13]. It is designed to take congestion and traffic load balancing into consideration, other than just advancement as in [4]. All the eligible relays of a node compute two values, namely, the Geographic Priority Index (GPI), i.e., the index of the region the node would belong to in the GeRaF framework, and the Queue Priority Index (QPI), which is a measure of forwarding effectiveness as perceived by the relay. The QPI is computed as follows. Suppose that after gaining access to the channel, a node always tries to send a burst of up to $M_B$ data packets. Let then $Q$ be the queue occupancy of the relay and $N_B$ be the length of the burst as requested by the sender. Let also $M$ be the average length of a burst that the relay expects to be able to correctly transmit back to back. The QPI of that relay is calculated as $\lceil (Q + N_B) / M \rceil - 1$, and is smaller for smaller queue size, shorter requested burst length, and larger correct burst size. The QPI then captures how fast the packet will be further advanced if sent to that neighbor.

To solicit the election of a forwarder, the source first senses the channel and then issues RTS messages to scan the QPI values in increasing order, just as GeRaF scans geographic regions. In general, if more than one nodes answer the RTS message soliciting a given QPI, the sender knows that some relays offering that QPI are available. Among those, it tries to select the one with the best GPI, and to do that, it starts a second contention, using signaling messages to query nodes with progressively higher GPIs, à la GeRaF. This process eventually ends with the selection of a single relay. Note that the QPI is optimized first because we found that selecting routes with low congestion improves the performance of greedy routing much more than simply optimizing the advancement [13]. If no relay is found, the sender backs off.

Like GeRaF, ALBA exchanges relevant data such as the geographic coordinates and the values of $N_B$ through RTSs and CTSs, requiring little information to be stored inside sensors, namely the value of $M$ experienced when transmitting toward any node in the forwarding area.

III. THE RAINBOW BACKTRACKING MECHANISM AND ALBA–R

The performance of both GeRaF and ALBA is significantly affected by dead ends. The greedy rule adopted in these protocols to select nodes closer to the sink has a fatal problem: a packet that reaches a dead end gets stuck, since greedy geographic forwarding does not allow its only chance of

In practice, $M$ is computed by a possible relay through a weighed average of its past transmission history, i.e., the observed values of $M$ over a set of attempts.

\footnote{In other words, collisions are resolved using a splitting tree technique.}
continuing the routing process, i.e., through neighbors offering a negative advancement.\(^3\) This represents a serious problem, especially in topologies that, though connected, are sparse. In the scenario depicted in Figure 4, for instance, a particularly unfortunate random deployment prevents 60 out of 100 nodes from reaching the sink through greedy forwarding, leading to unacceptable packet losses. In order to compensate for the absence of a direct path, an algorithm to route packets around connectivity holes is needed. In the following, we introduce Rainbow, a novel mechanism to route around dead ends, designed to be completely integrated in the ALBA protocol with no need for additional signaling packets.

Let us call \(i\) a node engaged in packet forwarding. Let us call \(F\) the portion of node \(i\)'s coverage area where relays offering positive advancement are located, and similarly let us call \(F^C\) the remaining part of its coverage area.\(^4\) Finally, let \(C_1, \ldots, C_h\) be a set of \(h\) colors that nodes assign to themselves according to their ability to forward packets. At the beginning, all nodes are labeled \(C_1\) (say, “yellow”). All yellow nodes follow ALBA basic operations as described in the previous section. If no connectivity hole is present, all nodes remain labeled as yellow. However, if \(i\) determines that it cannot forward any packet after a sufficiently high number of attempts,\(^5\) it considers itself a dead end. Accordingly, it turns to color \(C_2\) (say, “red”), and exponentially decreases its own likelihood to participate as an eligible forwarder in contentions initiated by other (yellow) nodes. This way, the nodes located on branches going to dead ends progressively realize that those branches are useless for routing, and hence they stop volunteering as relays for other nodes. Note that this behavior is softer as compared to an abrupt stop in offering as a relay. In fact, a node which is unable to find a relay may still have routes to the sink through neighbors which are currently asleep, thus only temporarily unavailable.

Red nodes handle the packets that they generate or receive according to a different rule: the packet is sent away from the sink selecting as relay a yellow or red node in \(F^C\). This process is repeated until a yellow node is reached. Starting from the yellow node, regular ALBA operation is resumed. Therefore, the packet is forwarded to the sink along a route which goes only through yellow nodes.

Now, if a red node is unable to find relays in \(F^C\) it progressively stops accepting packets from other red nodes, eventually switching to color \(C_3\) (say, “blue”). According to this method for assigning colors, blue nodes do not have yellow neighbors but could have a red neighbor in the \(F\) area. They will not candidiate themselves as relays for red or yellow nodes, but will just try to find a route for locally generated packets, asking other blue or red neighbors located in \(F\) to candidiate themselves as relay (giving priority to red nodes). If even blue nodes fail to forward data, they switch their color again, turning to \(C_4\) (say, “violet”). Similar to red ones, violet nodes search \(F^C\), trying to locate other violet or blue nodes, giving precedence to blue.

This mechanism may be generalized for an arbitrary number of colors \(h\). Any node with label \(C_k\), \(k < h\) always searches relays with color \(C_{k-1}\) or \(C_k\) in \(F^C\) for \(k\) even, and in \(F\) for \(k\) odd, with the exception of nodes with color \(C_1\), which always look for other \(C_1\)'s in \(F\). As a general rule, a greater number of colors allows more nodes to find a route toward the sink. Practically speaking, however, 4 colors are sufficient to handle most situations, as detailed in Section IV. It is possible to prove that \(i\) the Rainbow protocol is loop-free, \(ii\) nodes that have routes to the sink with \(h\) changes of direction (i.e., relays are alternately searched in \(F\) and \(F^C\) at subsequent forwarding steps) and that have no route to the sink with less than \(h\) changes of directions converge to color \(C_h\) in a finite time. The formal proof of these claims is reported in the Appendix.

Rainbow and ALBA are combined into a protocol called ALBA–R, \(h\) being the number of colors used in Rainbow. We wish to remark again that no specialized signaling is required for Rainbow operations, as Rainbow is designed to be transparently integrated into ALBA.

### IV. Simulation Results

In this section we illustrate the performance evaluation of ALBA–R. The protocol has been implemented in the VINT project network simulator ns2 \([14]\). We have considered different scenarios, corresponding to different nodal densities in order to demonstrate ALBA–R performance both in common scenarios (dense networks) and in harder ones (sparse networks), where its forwarding and rerouting capabilities are more frequently used. Section IV-A shows results for ALBA–R in denser networks, where the Rainbow mechanism is seldom invoked. In this way we evaluate ALBA–R to show that it scales well, impose low energy consumption and end-to-end packet latency, and has a good packet delivery ratio. The performance of ALBA–R is then tested in much sparser topologies (Section IV-B). Many solutions for data forwarding show quite a degradation in performance because of packets getting stuck (and discarded) in dead-ends. With the Rainbow protocol, ALBA–R performs quite well also in this scenarios. The number of colors required for achieving good performance is also investigated here.

Sensors are deployed in a square area with side \(L = 320\) m. The deployment area is divided into six squares. Three of them, randomly chosen, are low density zones, while the other three are high density zones. Most of the nodes (75\% of them) are randomly placed in the high density zones. The remaining 25\% are randomly placed in low density zones (Figure 4). This choice of nodal placement corresponds to the more realistic case where nodes are deployed in rough, uneven terrain that would make highly unlikely a completely random and uniform scattering. Furthermore, the corresponding heterogeneous nodal density makes for a tougher testing of ALBA–R operations.\(^6\)

The number of sensor nodes \(n\) ranges in the set \(\{100, 200, 300, 600, 800, 1000\}\). The coverage range of a node

\(^3\)Recall that ALBA does not consider nodes farther from the sink than the current forwarder, even when searching for neighbors with high QPI.

\(^4\)Nodes which are at the same distance from the sink as \(i\) are in \(F\) if and only if they have an ID which is smaller than \(i\). Otherwise they are in \(F^C\).

\(^5\)The optimal number of attempts has been derived to make false positives extremely unlikely when varying the scenario parameters in realistic ranges. No false positives were recorded in our experiments.

\(^6\)Experiments have also been performed when sensor nodes are randomly and uniformly deployed. Lack of space prevents a detailed discussion of those results, whose trend is, however, quite similar to what shown here.
is set to $r = 40$ m. The duty cycle of all nodes is set to $d = 0.1$. Packets are generated according to a Poisson process with rate $\lambda$ packets per second. All nodes have a buffer that can store up to 20 packets. Newly generated packets are randomly assigned a source node, which accepts the packet only if its buffer is not full, and discards it otherwise. Data packets are 250 Bytes long, whereas control packets are shorter (25 Bytes). The channel rate is 38.4 Kbps. The maximum length of a transmitted burst of packets with ALBA–R is set to 5. The number of different QPI as well as GPI values is 4.

The energy consumed by the nodes is calculated through the first-order energy model described in [15]. More specifically, the energy for receiving a bit, $E_{RX}$, is assumed constant, whereas the energy required to transmit one bit, $E_{TX}(r)$, is computed as follows:

$$E_{TX}(r) = E_{TXa} + E_{TXs}(r)$$

$$E_{TXs}(r) = \varepsilon_a \cdot r^2,$$

where $E_{TX}$ is the energy needed for the transmitter circuitry (and is set equal to $E_{RX}$), and $E_{TXs}(r)$ models the energy required to cover the transmission range $r$. The energy spend by a node in asleep mode is set to $1/1000$ of the cost of receiving.

All tested topologies are connected, i.e., there always exists at least one route joining any pair of nodes in the network. However, depending on the number of nodes, simple greedy forwarding may fail, in the sense that the forwarding area of some nodes may be empty. This require the application of the Rainbow algorithm for each node to determine what its color is. It takes a short time for a node to converge to its final color. However, since we are interested in the stationary behavior of ALBA–R, we do not average the performance metrics over the whole simulation time, but only after the initial transient phase. Note that such transient behavior is unavoidable, since the nodes are not topology-aware at the beginning, and therefore need some time to infer which routes can be followed to the sink, based on successful/unsuccessful transmission attempts.

A. Performance of ALBA–R in well-connected topologies

We test the performance of ALBA–R in dense networks, to show its scalability properties and its effectiveness in delivering packets. This is the kind of scenario where the use of Rainbow is seldom required.

Figures 1, 2 and 3 depict the average energy consumption per node, the end-to-end packet latency, and the protocol overhead, respectively. The energy is normalized to the consumption of a node just following the duty cycle. The overhead is defined as the ratio between ALBA–R per-packet channel occupancy and the time that would be required for packet forwarding in the best case (i.e., channel sense, RTS and data transmission, plus CTS and ACK reception). The packet delivery ratio is not explicitly shown here, since it is basically 100% for all traffic loads $\leq 4$. It then starts decreasing. When $\lambda = 6$ packet delivery ratio can be as low as 85% ($n = 1000$). The reason is that at high loads the funneling effect around the sink makes hard to successfully deliver packets. At these high loads the probability to sense the channel busy in the area close to the sink is very high. We observe that these values prove that ALBA–R performs well till the network saturates.

The energy consumption (Figure 1) and the end-to-end packet latency (Figure 2) show an inverse trend with respect to the packet delivery ratio. A more congested network means higher time and energy to find relays, and therefore higher end-to-end latencies. A similar trend is shown by the overhead (Figure 3). We observe that ALBA–R has a very limited
overhead per packet. When \( \lambda = 0.5 \) it is only from 18 to 21\% higher than the minimum when \( n = 600, 800, 1000 \). As the traffic load increases, ALBA maintains good performance in terms of overhead, mainly due to its light relay selection scheme (only a few packets need to be exchanged for each contention) and to the fact that the overhead for relay selection is shared among a burst of packets (since transmissions take place in bursts). When \( \lambda = 4 (\lambda = 6) \) the normalized overhead per node is only 1.37 (1.41) in large networks (\( n = 1000 \)). The different behavior observed in networks with \( n = 300 \) depends on the combination of two effects. On one hand, being the topology more constrained (i.e., traffic has to go though fewer nodes) the queues of the relay nodes build up at medium load allowing ALBA–R to exploit the possibility to transmit packets in bursts. This explains the initial decrease in the overhead. On the other hand, with fewer eligible relay nodes the network soon becomes congested. When the eligible relays queues are full such nodes do not offer themselves as relays, making it more time and overhead consuming to successfully forward a packet to the next hop.

**B. Performance of ALBA–R in sparse topologies**

The most interesting validation of the effectiveness of Rainbow concerns the performance of ALBA–R in sparse scenarios (\( n = 100, 200 \)). In this case, there is a significant percentage of nodes (25\% on average when \( n = 100 \), and an average of 6\% when \( n = 200 \)) which are unable to find a route to the sink. The worst case scenarios for networks with \( n = 100 \) and \( n = 200 \) nodes are displayed in figures 4 and 5.

A comparison of ALBA and ALBA–R is depicted in figures 6, 7 and 8. The number of colors for ALBA–R has been set to 4. All nodes converge to their correct color in any topology. At \( n = 200 \) ALBA–R delivers all the generated packets while ALBA delivery ratio is only from 79\% to 83\%. This remarkable improvement is even more evident in the sparser case (\( n = 100 \)), where ALBA–R delivers 98\% of the packets while ALBA only bring home from 56.8\% to 67\% of them. Note that the percentage of the packets discarded by ALBA is higher than the percentage of non-yellow nodes. Packets generated by non-yellow nodes are discarded. ALBA load balancing feature makes nodes seldom select as relays nodes which are on branches leading to dead ends. However, choosing next hop relays based on the QPI index does not guarantee that packets generated by yellow nodes are always forwarded to yellow relays. Some of these packets indeed reach nodes on the way to dead ends and are lost. The main drawback in the use of ALBA–R is represented by the more intense signaling and data traffic, which translates into higher packet latencies, as seen in Figure 7 and 8. Only the delay experienced by packets generated by yellow nodes is reported in Figure 7, since packets generated by other nodes are lost in ALBA. As traffic grows, ALBA selects more frequently relays offering a low QPI, but not necessarily a high GPI. This results in longer routes (up to 6\% when \( n = 200 \) and up to 11\% when \( n = 100 \)) and consequently higher latencies. Higher network loads also make the relay selection procedure more time-consuming, as nodes are often forced to back off, mainly after having sensed a busy channel. Both facts have a detrimental effect on latency. Figure 9 compares the overhead generated by ALBA and ALBA–R. Quite surprisingly ALBA–R experiences a lower overhead than ALBA. This happens for two reasons. ALBA–R does not pay the toll of the many retransmission for the packets that have reached nodes on dead ends. ALBA–R also makes good use of the transmissions of packets in bursts.

In order to broaden our understanding of ALBA–R performance in sparse scenarios, we have also varied the number of colors used by ALBA–R and observed how this affects its performance. We call ALBA–R\(_h \), the version of ALBA–R that uses \( h \) colors. In particular, ALBA–R\(_1 \) is ALBA (Section II-B). Increasing \( h \) improves re-routing, and increases the probability that a packet reaches the sink. Apart from rare, very congested situations, a packet is successfully delivered to the sink whenever it has a path to the sink which changes direction (toward/away from the sink) at most \( h \) times. Therefore, the higher \( h \), the better the delivery ratio (Figure 10).

Figure 11 shows the average end-to-end latency incurred by the packets generated by non-yellow nodes. As a rule of thumb, higher values of \( h \) induce longer routes and allow farther sources to reach the sink, although paying the price of multiple changes of direction (alternations). Correspondingly, the latency increases.

**V. Conclusions**

In this paper, we have described Rainbow, a novel protocol for routing packets around connectivity holes in WSNs. Our solution does not require the planarization of the network topology graph, thus generating less overhead than previous solutions for geographic face routing. The Rainbow scheme is integrated into ALBA, a cross–layer optimized MAC and routing solution that we previously proposed. The resulting protocol suite, termed ALBA–R, is an effective, complete scheme for convergecasting in sensor networks.

ALBA–R has been formally proven to be loop-free. A thorough ns2 based performance evaluation has shown the validity and feasibility of our approach. In particular, we tested the protocol performance on sparse topologies (prone to connectivity holes), and observed that ALBA–R performs remarkably well even in this case. ALBA–R is a simple, completely distributed, low overhead protocol which is natively cross–layer. Given its implementation simplicity and its overall good performance, it turns out to be an attractive solution for realistic WSNs implementation.

**Acknowledgment**

This work was partially supported by the European 6th Framework Programme through the Integrated Project (IP) e-SENSE, contract number 027227.

**References**


Fig. 6. Packet delivery ratio, ALBA-R vs. ALBA, $n = 100, 200$.

Fig. 7. End-to-end delivery delay, ALBA–R vs. ALBA, $n = 100, 200$.

Fig. 8. Avg. norm. energy consumption per node, ALBA–R vs. ALBA, $n = 100, 200$.

Fig. 9. Avg. norm. overhead per node, ALBA–R, $n = 100, 200$.

Fig. 10. Packet delivery ratio for varying $h$.

Fig. 11. End-to-end delay incurred by back-tracked packets for varying $h$.

Proof: We start by showing that there are no cycles (loops) in routes only made up of nodes of the same color (say, all $C_1$ nodes). Let $x_0$ be a $C_1$ node and $x_1, x_2, \ldots, x_k$ a route through $k$ $C_1$ nodes. If a $k$-cycle existed (i.e., $x_1 = x_k$), then $d(x_1) = d(x_k)$, where $d(x)$ is the Euclidean distance from node $x$ to the sink. However, $C_1$-nodes always forward packets only if $d(x_0) \geq d(x_1) \geq \cdots \geq d(x_k)$. Then $d(x_0) = d(x_1) = d(x_k)$. But this is possible only if $x_k = x_{k-1} < x_0$. Hence a contradiction.

Now let us proceed by induction on $h$, the number of colors. Suppose that routes are loop–free for all nodes with colors $C_1, \ldots, C_h$. We have to prove that the theorem holds true for $h + 1$ colors. We consider the two cases corresponding to $h + 1$ being either odd or even. If $h + 1$ is odd, $C_{h+1}$-nodes search for relays in region $F$. Let us consider a route from a $C_{h+1}$-node $x_0$ to the sink $S$, $x_0, \ldots, x_k, x_{k+1}, \ldots, S$, passing through $C_{h+1}$-nodes, then $C_h$-nodes, and so on. Let $x_k$ be the first $C_h$-node. The route from $x_0$ to $x_{k-1}$ is clearly loop free, since all the nodes have the same color. We notice also that no route passes twice through $x_k$, because for such a cycle to exist the node $x_i$, $i < k$, that would close the cycle $(x_i = x_k)$ would have to change its color from $C_{h+1}$ to $C_h$, which is against Rainbow rules. Finally, $x_k, \ldots, S$ is loop-free by inductive hypothesis.

The case with $h + 1$ even is similar to the previous one, considering that the region where relays are searched for is $F^C$ instead of $F$ and that the direction of the inequalities becomes $\leq$.

The following theorem shows that if nodes are able to reliably decide whether they have neighbors in $F$ or $F^C$, then they will eventually become of the proper color. In the proof by alternation we indicate a change in the region, whether $F$.

**Appendix**

We formally prove the claims of Section III, regarding the properties of the Rainbow algorithm.

*Theorem 1 (Rainbow is loop–free):* The Rainbow extension to ALBA always finds loop–free routes to the sink.

**Proof:**

[Further details of the proof would be included here.]
or $F^C$, queried for relays (Section III).

**Theorem 2:** All and only the nodes that have routes to the sink with at least $h$ alternations take the color $h$ in finite time.

**Proof:** Let us denote with $F(x)$ and $F^C(x)$ the regions $F$ and $F^C$ as seen from node $x$. Let also $N_F(x)$ and $N_{F^C}(x)$ be the sets of nodes in the corresponding regions. (We assume that if at least a relay exists in $F(x)$ and $F^C(x)$, $x$ is able to eventually find it.)

In the proof we use the function $\Xi(x)$ that lists the network nodes in order of increasing distance from the sink, i.e., for the node $x$ closest to the sink $\Xi(x) = 1$, for the node $x$ second closest to the sink is $\Xi(x) = 2$, etc. (If two nodes $y$ and $z$ are at the same distance from the sink, and $y < z$ we stipulate that $\Xi(y) < \Xi(z)$.) We also use the function $\Xi'(x)$ that lists the network nodes in order of increasing number of alternations required to reach the sink. If two nodes show the same number of alternations, then the closer to the sink the lower its $\Xi'$. (Ties are broken as for $\Xi$.)

The theorem is proven by “double induction” on the number $h$ of alternations required to get packets from their source to the sink and on $\Xi(x)$ and $\Xi'(x)$.

Case $h = 1$. We start by proving that a node $x$ gets colored $C_1$ if there exists a route from $x$ to the sink where each relay is chosen in the $F$ region of its predecessor in the route (no alternation case). We proceed by induction on $\Xi(x)$. If $\Xi(x) = 1$, either the sink is in $N_F(x)$ or not. In the first case $x$ is a $C_1$-node because it finds the sink in $F$. If the sink is not in $N_F(x)$ then the network is disconnected, since the sink is out of the transmission range of its closest node. Since we assumed the network connected, this case never occurs.

Let us now assume that for each $y$ such that $\Xi(y) \leq k$, $k > 1$, the claim holds true. Consider the node $x$ such that $\Xi(x) = k + 1$. If the route $x = x_0, x_1, \ldots, S$ from $x$ to the sink $S$ is formed of relays chosen in each node’s $F$ region, then $d(x_0) \leq \cdots \leq d(x_0)$, where $x_i \in N_F(x_{i-1})$. If $d(x_{i+1}) = d(x_i)$ then $x_{i+1} < x_i$. Therefore, $\Xi(x_i) < \Xi(x_{i+1})$. To each of the $x_i$, the inductive hypothesis applies, which means that every node $x_i$ stays colored $C_1$ (each node belonging to the route is a $C_1$-node). Now, $x = x_0$ has at least one relay in $F(x_0)$ (the node $x_1$), and since $x_1$ is $C_1$, itself, i.e., the node with $\Xi(x) = k + 1$, remains colored $C_1$.

Let us now consider the case where $x$ such that $\Xi(x) = k + 1$ does not have routes to the sink without alternations (the “only” part of the “all and only” claim). Since $d(y) < d(x)$ or $d(y) = d(x)$ and $y < x$, for each $y \in N_F(x)$, then $\Xi(y) < \Xi(x)$. As a result, by inductive hypothesis on $\Xi(x)$, all nodes $y \in N_F(x)$ will change their color until $N_F(x)$ has no more $C_1$-nodes. At this time, $x$ will no longer be able to forward in $F$, and will change its color as well. To sum up, for $h = 1$, the color of a node is $C_1$ if and only if the node exhibits a route toward the sink whose relays are always found in the region $F$ of its predecessor node.

By using the same reasoning, we now prove the following claim (case $h > 1$).

**Claim 1:** Assume that all nodes that have routes to the sink with $j \leq h$ alternations, $j > 1$, take color $C_j$ in a finite time. Then all nodes showing routes with $h + 1$ alternations take color $C_{h+1}$ in a finite time.

Let us assume at first that $h + 1$ is odd. We proceed by induction on $\Xi'(x)$. Base case: Let $x$ be the node closer to the sink and with routes to the sink with $h + 1$ alternations. Let also $m = \Xi'(x)$. By induction, Claim 1 is true for all nodes $y$ having $\Xi'(y) < m$, since these nodes have routes with at most $h$ alternations. Now, in any route with $h + 1$ alternations $x = x_0, x_1, \ldots, S$ from the sink $S$, $x_1 \in F(x_0)$, and moreover, $\Xi'(x_1) < \Xi'(x_0)$. Therefore, $x_1$ is a $C_{h}$-node. Node $x$ cannot be colored with any of the $C_h, \ldots, C_1$ colors because there is no route from $x_0$ to the sink with at most $h$ alternations. However, it can stay colored $C_{h+1}$ since it has a neighbors in its $F$ zone which is a $C_h$-node.

**Inductive step:** Assuming that Claim 1 is true for any $x$ such that $\Xi'(x) \leq h$, we prove that it is also true for $\Xi'(x) = k + 1$.

Let $x_0$ be node such that $\Xi'(x_0) = k + 1$ and let $y_0$ be the node such that $\Xi'(y_0) = k$. Either both $x_0$ and $y_0$ require $h + 1$ alternations, or $x_0$ requires more than $h + 1$ alternations. In the first case, a route $x_0, x_1, \ldots, S$ with $h + 1$ alternations leads to the sink, with $x_1 \in F(x_0)$, requiring $h$ or $h + 1$ alternations. Since $\Xi'(x_1) < \Xi'(x_0)$, node $x_1$ has color $C_h$ or $C_{h+1}$ by induction.

Therefore, $x_0$ will assume color $C_{h+1}$, due to the presence of $x_1$ in $F(x_0)$. Node $x_0$ cannot be colored with any of the $C_h, \ldots, C_1$ colors because there is no route from $x_0$ to the sink with at most $h$ alternations. In the second case, $x_0$ cannot assume any of the colors $C_{h+1}, C_h, \ldots, C_1$. In fact, by induction on $h$ it cannot be colored with $C_h, \ldots, C_1$. To be colored with $C_{h+1}$, $C_h$-nodes or $C_{h+1}$-nodes should be in $F(x_0)$. Such nodes would precede $x_0$ in the sorting given by $\Xi'$. Therefore, they would have routes to the sink with at most $h + 1$ alternations by induction on $\Xi'$. But then $x_0$ would exhibit at least one route with $h + 1$ alternations, which is a contradiction.

The care $h + 1$ even is similar to when it is odd. This time a different ordering function needs to be used, say $\Xi''(x)$, which lists all nodes in order of increasing number of alternations required to reach the sink. Differently from $\Xi'$, the node farther from the sink precedes the closer in case they exhibit an equal number of alternations.