ALBA-R: Load-Balancing Geographic Routing Around Connectivity Holes in Wireless Sensor Networks
SUPPLEMENTAL MATERIAL

Chiara Petrioli, *Senior Member, IEEE,* Michele Nati, *Member, IEEE,* Paolo Casari, *Member, IEEE,* Michele Zorzi, *Fellow, IEEE,* and Stefano Basagni, *Senior Member, IEEE*

Abstract—This paper presents ALBA-R, a protocol for *convergecasting* in wireless sensor networks. ALBA-R features the cross-layer integration of geographic routing with contention-based MAC for relay selection and load balancing (ALBA) as well as a mechanism to detect and route around connectivity holes (Rainbow). ALBA and Rainbow (ALBA-R) together solve the problem of routing around a dead end without overhead intensive techniques such as graph planarization and face routing. The protocol is localized and distributed, and adapts efficiently to varying traffic and node deployments. Through extensive ns2-based simulations we show that ALBA-R significantly outperforms other convergecasting protocols, especially in critical traffic conditions and low density networks. The performance of ALBA-R is also evaluated through experiments in an outdoor testbed of TinyOS motes. Our results show that ALBA-R is an energy efficient protocol that achieves remarkable performance in terms of packet delivery ratio and end-to-end latency in different scenarios, thus being suitable for real network deployments.

Keywords—Wireless sensor networks, cross-layer routing, connectivity holes.

1 INTRODUCTION

This supplemental material contains further results concerning the investigation of the ALBA protocol and of its Rainbow mechanism presented in [1]. We start by describing the process that led us to choose the values of some of the key parameters of ALBA forwarding. We then introduce the protocols for convergecasting whose performance we have compared to that of ALBA-R via ns2-based simulations (Section 3.1). Some of the simulations results are also shown in Section 3.2. We then show the performance evaluation of ALBA-R in an outdoor testbed of TinyOS motes (Section 4). In Section 5 we prove the correctness of Rainbow.

2 ALBA PARAMETER TUNING

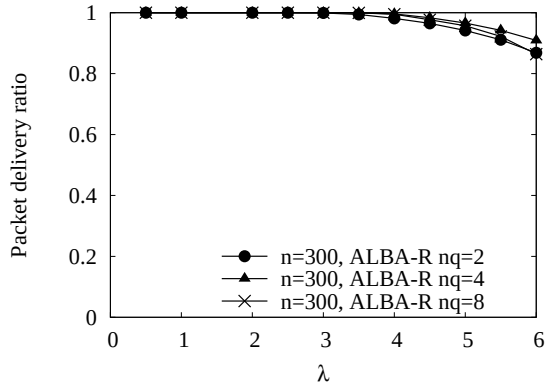
In this section we describe how we have tuned ALBA protocol parameters. Figures 1a to 1d display the performance of ALBA in terms of packet delivery ratio and end-to-end latency when varying $N_Q \in \{2, 4, 8\}$. The energy metric is not displayed since varying N_Q has a very limited impact on ALBA performance. The results shown here concern only simulations in networks with $n = 300$ and $n = 600$ nodes. We observed similar trends in networks with higher number of nodes.

The figures clearly show that the best performance trade-offs are obtained for $N_Q = 4$. Partitioning the nodes in too few QPI regions (case $N_Q = 2$) results into a up to 50% increase in the number of attempts of finding a relay. When the contention is longer it instead often happens that some nodes wake up and join it. When the number of regions increases the ability to select good relays also improves. However, if the number of QPI regions is too high ($N_Q = 8$) the advantages associated to a finer relay selection are overcome by the increased overhead. This motivates why $N_Q = 4$ is a good design choice. This is the value we used in all our experiments.

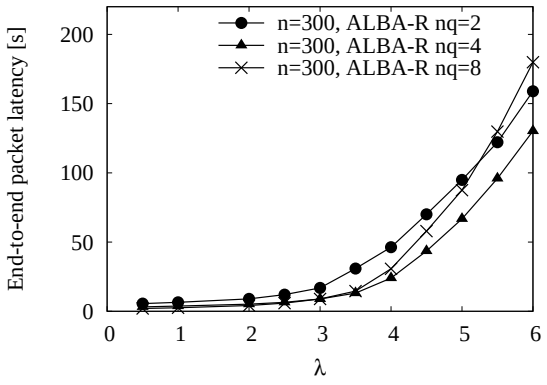
A similar way of reasoning was followed to tune the N_r parameter for ALBA GPI. This parameter, however, has a more limited impact on performance as GPI regions are used only as a tie breaker in case of relays with the same QPI. We found that $N_r = 4$ provides a good performance (see also [2]).

Finally, the number of retransmission attempts has been experimentally tuned so that nodes very seldom discard packets due to the fact that the N_{Att} or the N_{Boff} thresholds has been exceeded.

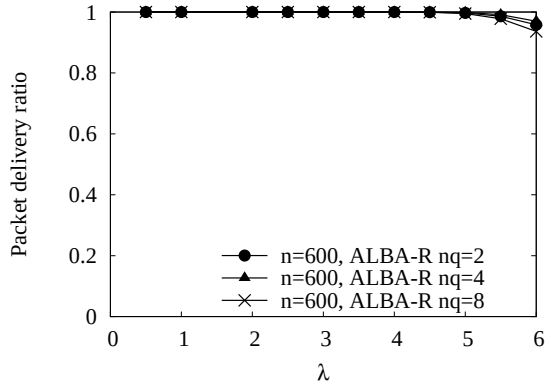
- *Chiara Petrioli is with the Department of Computer Science, University of Rome "La Sapienza," Rome, Italy. E-mail: petrioli@di.uniroma1.it.*
- *Michele Nati is with Centre for Communications Systems Research, University of Surrey, Guildford, U.K. E-mail: M.Nati@surrey.ac.uk.*
- *Paolo Casari and Michele Zorzi are with the Department of Information Engineering, University of Padova, Padova, Italy. E-mail: {casarip,zorzi}@dei.unipd.it.*
- *Stefano Basagni is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, U.S.A. E-mail: basagni@ece.neu.edu.*



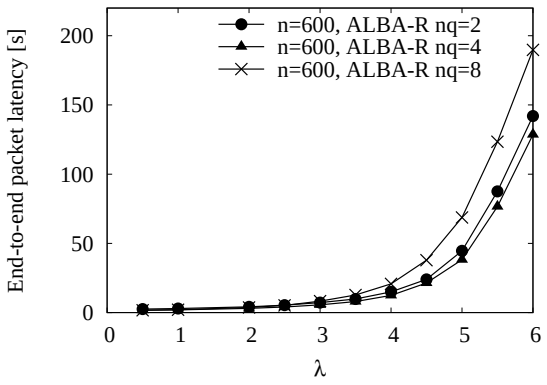
(a) Packet delivery ratio, $n = 300$.



(b) End-to-end packet latency, $n = 300$.



(c) Packet delivery ratio, $n = 600$.



(d) End-to-end packet latency, $n = 600$.

Figure 1. Determining the best value of N_q for ALBA QPI.

3 GeRaF and IRIS: Description and Comparison with ALBA

3.1 Description of the protocols

3.1.1 Geographic Random Forwarding (GeRaF)

GeRaF [3] is one of the first cross-layer solutions for convergecasting in WSNs. It reduces energy consumption by making nodes alternate between awake/asleep states according to a schedule with fixed duty cycle d. The schedules are asynchronous among different nodes. Packet transmissions are preceded by a contention to identify a relay among awake neighbors, preferring those which offer the best advancement towards the sink. In order to identify relays and enforce efficient channel access during the selection phase, the forwarding area[1] of the transmitter is divided into N_r regions, so that any node in a given region i is closer to the sink than any other node in region j, for all $i < j$. This is shown in Figure 1 of the paper where node A is the one providing source node S with the highest positive advancement toward the sink, residing in GPI region 0. Consequently, nodes B and C, also located in the region of positive advancement but providing smaller progress are located in GPI region 2. Node D is the node closest to the source S falling in the last GPI region 3. Before initiating packet transmission the channel is sensed for a time long enough to allow a node to discover ongoing handshakes [4]. If the channel is sensed free, the transmitter broadcasts a Request-to-Send (RTS) packet carrying the index of region 0. This silences nodes offering negative advancement and polls all awake nodes in region 0, which reply with a Clear-to-Send (CTS) packet. (Note that while the RTS packet is broadcast to all the sender neighbors, the CTS packet is unicast back to it.) Now, if only one neighbor is currently available, it sends the CTS, is immediately chosen as a relay, and is sent the data packet. Upon correct reception, it replies back with an ACK, thereby concluding the packet exchange. If more than one neighbors respond, their CTS will likely collide. From the reception of a garbled packet, the sender can detect the collision and arbitrate a contention among the multiple relays, by broadcasting an RTS that carries the index of region 0 again, along with a collision flag. Upon receiving a "collision RTS," nodes decide whether to send another CTS with probability $p = 0.5$. Silent nodes are excluded from further collision resolution rounds. In case no node responds, a further attempt is made involving the same set of candidate relays This collision resolution process is carried out until only one relay sends the CTS. This

1. The portion of the coverage area where relays offering a positive advancement are located.

node is chosen as the relay and it is sent the data packet. If region 0 is empty due to lack of active nodes, the transmitter broadcasts another RTS with the index of region 1. The regions are polled subsequently until some relays respond in a certain region (starting a contention), or until all regions are scanned without finding any available node. In the last case the node backs off.

GeRaF only requires nodes to know their own position and the position of the sink in order to complete a handshake. Any other information (such as the sender position, that allows relays to know the region they belong to) is exchanged through RTS and CTS packets.

The cross-layer design of GeRaF effectively performs relay selection and channel access jointly. However, due to its inherently greedy approach to routing, GeRaF has no mechanism for dealing with connectivity holes, which adversely affects its packet delivery ratio, especially in sparse networks. We also notice that basic greedy forwarding à la GeRaF does not consider current traffic at potential next hop neighbors when selecting a relay. Some nodes may become congested when some other neighbors could better handle packets (i.e., no load balancing is performed). Finally, in GeRaF nodes do not volunteer as relays when handling a packet. This means that when the traffic increases the number of available relays decreases, increasing the number of attempts and the overhead needed to advance the packets.

3.1.2 An Integrated Routing and Interest dissemination System (IRIS)

IRIS [5] is a framework for convergecasting in WSNs based on Hop Count (HC) routing. The scheme pairs up with an interest dissemination algorithm based on optimized probabilistic forwarding [6]. Convergecasting is performed by exploiting the knowledge of the HC of a node, stating its distance, in hops, from the sink. A node with HC n that has a packet to transmit starts a relay search procedure with a `REQ` packet triggering neighbors with HC $n-1$. Every awake node with this HC computes its own probability of replying with a `REP` packet based on a cost function, which captures how suitable the node is to serve as a relay. Depending on the computed probability value, the node uniformly chooses a random backoff time in the interval $[0, B_{max}]$, after which the reply is sent. The lower the probability, the closer the backoff to the maximum value B_{max}. In case of collisions among multiple `REP` packets, or if no `REP` is received, the sender triggers further rounds until one winner can be elected among neighbors with HC $n-1$. The same procedure is performed for relays with HC n, and a relay (with HC $n-1$ or HC n) is chosen based only on local measurements (similarly to [7]). This decision is influenced by the cost of the nodes involved. This cost can be defined to take into account any relevant metric, such as node queue length, battery energy level, number of previously forwarded packets, and so on.

While in general IRIS routes go through nodes with equal-HC or lower-HC nodes, in our performance comparison we impose that packets are routed only toward nodes with lower HC. This is to make the resulting multi-hop route as short as possible. We have also tested different versions of IRIS (with the different cost functions described in [5], and selecting relays among equal and lower-HC nodes). We observed that the relative performance of IRIS, ALBA-R and GeRaF was not affected by these changes.

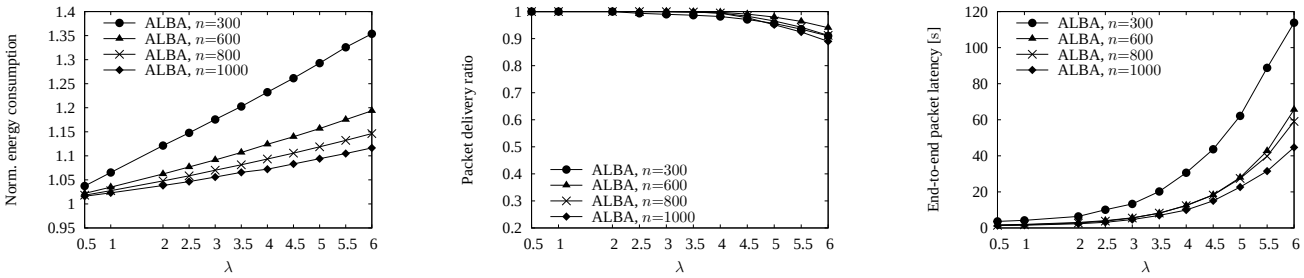
3.2 Performance comparison

Simulation parameters and scenarios are described in the main paper [1].

3.2.1 Performance comparison of ALBA and GeRaF

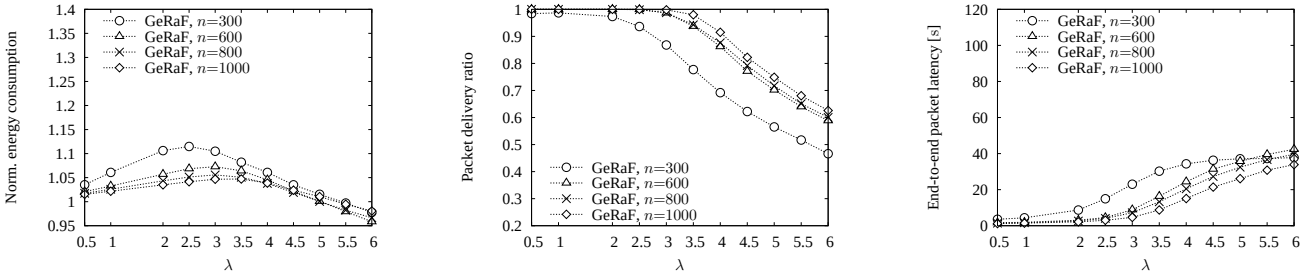
In this section we compare the performance of ALBA and GeRaF. The results shown here refer to scenarios where $d = 0.1$ and the number of nodes n varies between 300 and 1000.

Figures 2a and 3a show that for any protocol and any considered density and traffic load, packet transmissions and receptions have a moderate impact on energy consumption with respect to plain duty cycling. (We recall that the normalized energy consumption is defined as the network total energy consumption divided by the energy the network would have consumed if nodes would strictly follow the duty cycle d.) In all curves, the normalized energy consumption is at most 1.35. However, while the energy consumed by ALBA steadily increases with the traffic load, in GeRaF the normalized energy first increases and then starts decreasing, never exceeding 1.15. The main reason is shown in figures 2b and 3b. At moderate and high traffic ($\lambda > 2$) GeRaF experiences significant packet loss. Its delivery ratio is from 70 to 90% when $\lambda = 4$, and decreases to $50/60\%$ when $\lambda = 6$, depending on node density. Conversely, ALBA correctly delivers all packets for $\lambda \leq 4$ and more than 90% of the packets at very high traffic ($\lambda = 6$). This is due to the different behavior of ALBA and GeRaF when a node searches for a relay. While waiting for the end of a backoff interval, in ALBA nodes continue to follow the regular duty cycle and keep participating in contentions. In GeRaF, instead, a node currently handling a packet stops volunteering as a relay and therefore, as traffic grows it becomes harder to find relays. When $n = 300$ and $\lambda = 4$ ($\lambda = 6$), the average number of times each node backs off before finding a relay is 7.23 (8.68). With ALBA, this value decreases to 2.8 (4.15). The lower the number of potential relays of a node (i.e., the lower the density) the more pronounced this effect. Not only does ALBA rely on a higher average number of potential relays, but it is also able to more effectively forward traffic at high load. The load balancing mechanism provided by ALBA selects those relays with a better forwarding history; the transmission of packets back-to-back makes use of the channel (and of the relay search efforts) more effectively. As a consequence ALBA is able to successfully deliver to the sink a higher number of packets, which however imposes higher energy consumption.



(a) Norm. energy consumption. (b) Packet delivery ratio. (c) End-to-end packet latency.

Figure 2. ALBA: Energy consumption per node, packet delivery ratio and end-to-end packet latency.



(a) Norm. energy consumption. (b) Packet delivery ratio. (c) End-to-end packet latency.

Figure 3. GeRaF: Energy consumption per node, packet delivery ratio and end-to-end packet latency.

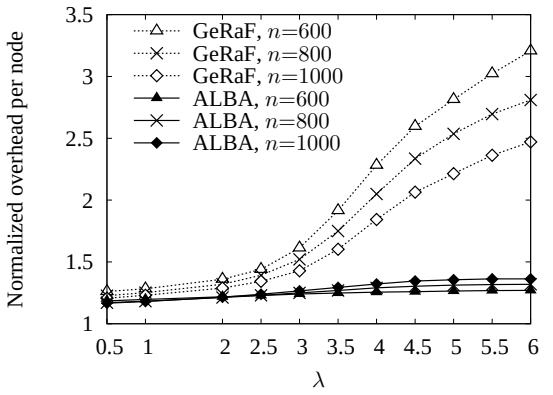
In order to explain why in GeRaF the normalized energy consumption is less than 1 at high loads, we recall that a large amount of traffic causes congestion and leaves nodes in backoff, unable to find a relay. As the average length of the backoff interval is longer than the time between two normal awake periods this results in decreased normalized energy consumption.

Figures 2c and 3c display the latency performance of ALBA and GeRaF, respectively. In scenarios with average node density ALBA shows a slightly lower latency than GeRaF, even when the traffic is low. For instance, when $n = 300$ and $\lambda = 1$ ALBA end-to-end latency is 4% lower than that of GeRaF. This reduction becomes more significant when the traffic increases, reaching 26% and 42% when $\lambda = 2$ and $= 3$, respectively. This might appear counter-intuitive in that ALBA contention may trigger two phases, one for the QPI and a second one for the GPI, which might seem to require more time. However, both QPI and GPI-based searches are fast, especially with low duty cycles, when the number of neighbors that are awake during a handshake is limited. Furthermore, we observed that GPI-based search is rarely required. GeRaF contention is simpler. However, the average number of eligible neighbors that are awake is lower (recall that nodes handling traffic do not answer the RTS), resulting in a longer time to find a relay. A count of the average number of RTS packets broadcast before finding a relay confirms this intuition: ALBA requires only an average of 1.5 rounds to find a relay, whereas GeRaF needs about 1.75 rounds (when $\lambda \leq 2.5$). When the network density increases, the number of eligible relays also increases, improving GeRaF relay selection. This explains why GeRaF end-to-end latency is slightly lower than that of ALBA in the highest density scenario ($n = 1000$), for $\lambda \leq 2.5$. This reduction is however quite limited (between 3 and 8%). As soon as the traffic increases and the number of relays available and not handling packets decreases, the performance of GeRaF degrades, suffering a significantly higher end-to-end latency than that experienced by ALBA. When $\lambda = 3$ the latency suffered by GeRaF is 25% higher than that of ALBA. The increase grows to 52% when $\lambda = 3.5$.

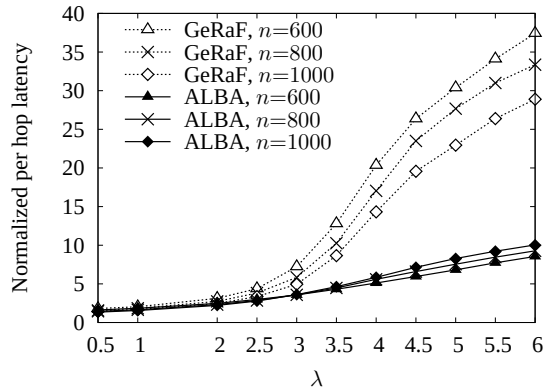
The higher packet delivery ratio justifies instead the larger end-to-end delay of ALBA at high traffic. While the relay search phases are still short (2.2 rounds vs. the 4 rounds of GeRaF when $\lambda = 6$), and routes become only 10 to 15% longer with respect to those traveled by packets routed by GeRaF, nodes have to manage a larger amount of traffic, which increases the delay suffered by each packet. However, it should be noted that as the traffic increases, ALBA experiences only a moderate increase in latency. The QPI-based relay selection enables better load balancing, and transmitting packets in bursts improves the protocol performance. We observe that the size of a correctly sent packet burst grows, on average, from 1.1 packets ($\lambda = 2$) to 1.6 ($\lambda = 4$), up to 2.6 ($\lambda = 6$).

We also compute the ratio between the average time ALBA nodes use the channel for relay selection, for transmitting data packets and receiving their ACKs, and the minimum time needed for relay selection and data transmission (i.e., the time to perform carrier sense, broadcast one RTS, receive the CTS, send the packet and receive its ACK). This metric, denoted as normalized overhead per packet in the following, is higher in

protocols whose relay selection scheme requires heavy traffic exchange. Figure 4a depicts the normalized overhead experienced by each node for relaying a packet to the next hop when n varies from 600 to 1000 and λ varies between 0.5 and 6. The per-node overhead required by ALBA is very limited. When $\lambda = 0.5$ it is only 20% greater than the minimum, whereas it grows for higher loads, as expected. The increase is however moderate, because ALBA relay selection scheme requires only the exchange of a few packets at each contention, and because the overhead for relay selection is shared among the packets in the transmitted burst. When $\lambda = 4$ ($\lambda = 6$) the normalized overhead is only 1.21 (1.37) for $n = 1000$. GeRaF performance is always inferior to that of ALBA because relay searches are more likely to be unsuccessful, as explained before. This results into a significant increase of the normalized overhead, which can be as high as 3.2 for $n = 600$ and $\lambda = 6$. Similar trends are observed for the normalized per-hop latency (Figure 4b). This metric is defined as the ratio between the average time required to advance a packet one hop and the best case latency, i.e., the time needed to perform carrier sense and send data with minimum overhead (one RTS, one CTS and one ACK). (Queuing delay is not considered here.) Both ALBA and GeRaF perform quite well considering that they have to deal with nodes following asynchronous awake/asleep schedules with a low duty cycle. When $\lambda \leq 2$, the normalized per-hop latency is very low, never exceeding 3.13. In GeRaF, as the load increases and congestion builds up, nodes back off more often, and the greater time required to complete handshakes degrades the forwarding performance. At high loads, ALBA makes a better use of the time needed to find a relay by sending an entire burst of packets back-to-back. Moreover, it is easier to find neighbors that are awake since nodes in backoff follow the duty cycle. This significantly decreases ALBA normalized per hop latency that is around one third of the per hop latency experienced by GeRaF when $\lambda = 6$.



(a) Overhead per node.



(b) Per-hop latency.

Figure 4. Performance comparison of ALBA and GeRaF: Overhead per node and per-hop latency.

3.2.2 Performance comparison of ALBA and IRIS

IRIS [5] is a recent representative of the class of protocols based on interest dissemination, pioneered ten years ago by Directed Diffusion [8]. It represents an important benchmark in our evaluation, as it allows to compare ALBA to protocols based on a different forwarding paradigm. In IRIS, convergecasting is performed based both on a hop count metric computed by the nodes during the interest dissemination phase, and on a local cost function. We carried out our comparison by considering a network of $n = 600$ nodes with a transmit range $r = 40$m. As before, the common protocol parameters have been set to the same value as in the main paper. The pros and cons of the different approaches taken by ALBA and IRIS are clearly visible in figures 5 and 6, reporting the average number of hops traveled by a packet on the path to the sink and the overall end-to-end latency, respectively. As expected, the hop-count forwarding strategy

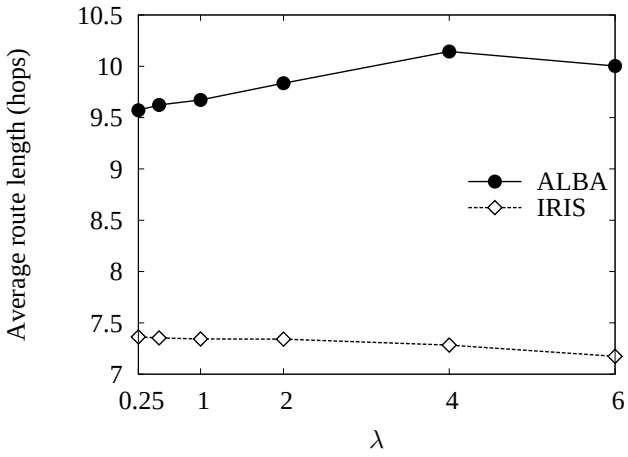


Figure 5. ALBA and IRIS: Per-packet traveled hops.

in IRIS optimizes the number of traversed hops, which is in fact stable to an average of 7.25 at any value of λ. By way of contrast, the routes found by ALBA are slightly longer (8 to 10 hops), because ALBA focuses on the forwarding capabilities of the relays, and uses advancement only as a secondary metric. This choice actually pays off in terms of latency. At low traffic, ALBA delivers packets three times as fast as IRIS. At medium

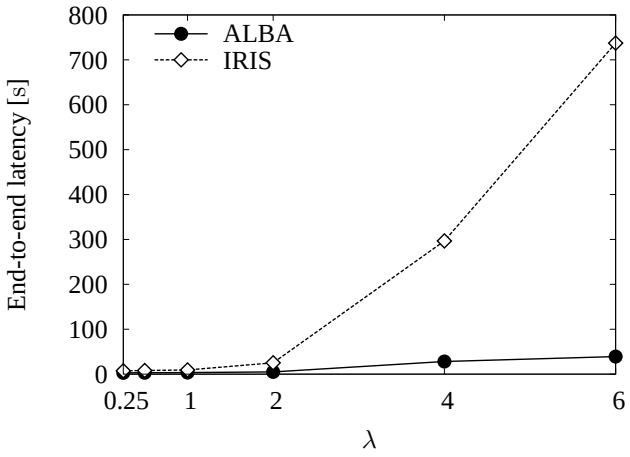


Figure 6. ALBA and IRIS: End-to-end packet latency.

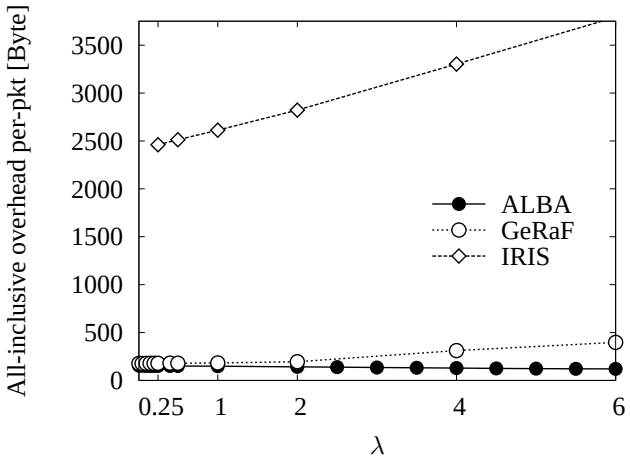


Figure 7. Per packet overhead bytes.

and high traffic the gap widens significantly. At $\lambda = 6$, ALBA end-to-end packet latency is 39s, almost 20 times smaller than IRIS, which requires longer contentions in order to find hop count and cost-optimal relays. ALBA relay selection trades off optimality with performance, instead. Being based on the QPI metric, it finds sub-optimal relays in terms of hop count or advancement (which causes the route length to increase as observed in Figure 5); however, selected relays are faster at further forwarding packets after reception.

The advantages of ALBA and a further proof of its lightness can be observed from Figure 7, which depicts the average number of bytes of overhead incurred for any successful data packet transmission. This figure compares all three protocols considered so far, and concludes our evaluation of ALBA against its benchmarks. Figure 7 shows that ALBA scales very well with traffic, and proves once again that back-to-back transmissions help maintain the overhead contained (ALBA is the only protocol for which the overhead decreases at high traffic). The heavier relay search phases required by the other protocols (especially IRIS) tend to cause a much larger overhead. The lighter ALBA procedures, instead, have also a beneficial impact on all relevant metrics including latency, delivery ratio, and energy consumption.

4 Testbed-based Performance Results

ALBA-R has been implemented and tested in a vineyard in Frascati, in the surroundings of Rome, Italy. The aim of these outdoor experiments is to demonstrate the suitability of ALBA-R as a routing solution in realistic scenarios and to validate the accuracy of our simulation model.

4.1 Experimental scenario and parameters

We have implemented the ALBA and Rainbow protocols described in the paper in TinyOS and run them on TelosB nodes which operate in the 2.4GHz ISM band with a channel rate of 250Kbps. ALBA-R cross layer relay selection exploits the IEEE 802.15.4 CSMA/CA primitives made available by TinyOS. The carrier sensing has been implemented invoking subsequent CCA primitives over the required carrier sensing interval.

Experiments have been performed on a 40 node deployment for vineyard monitoring, located in Frascati, right outside Roma, Italy. The considered topology is depicted in Figure 8. It spans an area 80×200 square meters. Each wireless node was deployed at the site of a moisture soil sensor used in the vineyard for monitoring purposes. Such sensors were placed on a grid in adjacent positions that are 20 meters apart, resulting in the topology displayed in the figure.

Following the specifications of the CC2420 radio chip, a power level of 27 has been used for each node, resulting in a fairly omnidirectional coverage area with a radius of about 60m. Each node has therefore 3 to 5 neighbors in the direction of the sink.

As expected in a rural area, no external interfering sources (e.g., WiFi networks) were operating on the same frequency band. Real life signal propagation and link dynamics however affect the protocol performance in the testbed. We have for instance verified seasonal differences in link quality, e.g., depending on foliage. The topology used in the simulations is obtained from the testbed experiments using only links with low packet error rate (PER).

We considered three different traffic scenarios in both simulations and testbed. Nodes more than one hop away from the sink generate packets according to a Poisson process with average value λ ranging in the set $\{0.25, 0.5, 1\}$, which corresponds to a packet being created and injected into the network every 4, 2 and 1 seconds, respectively. The value of the node duty cycle d ranges in $\{0.05, 0.1, 0.3\}$; the sink is always awake. Other parameter values used for both simulation and testbed experiments are listed in Table 1.

In all testbed scenarios the average number of hops traveled by packets successfully delivered to the sink ranges from 2.13 to 2.95 hops, for increasing traffic.

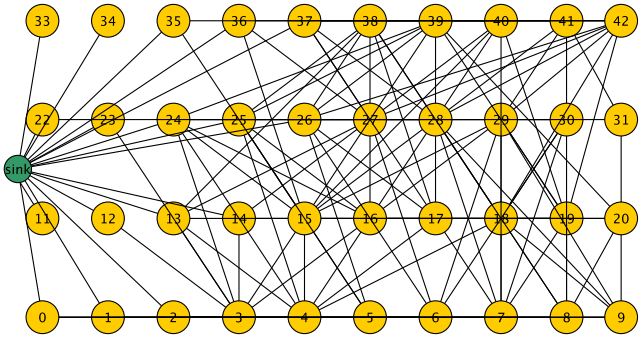


Figure 8. Network topology used in the simulations, derived from that of the actual deployment.

Table 1
Simulation and testbed parameters.

Parameter	Value
On time	400 ms
Average backoff length	1200 ms
Queue length	40 pkts
Max burst length	10 pkts

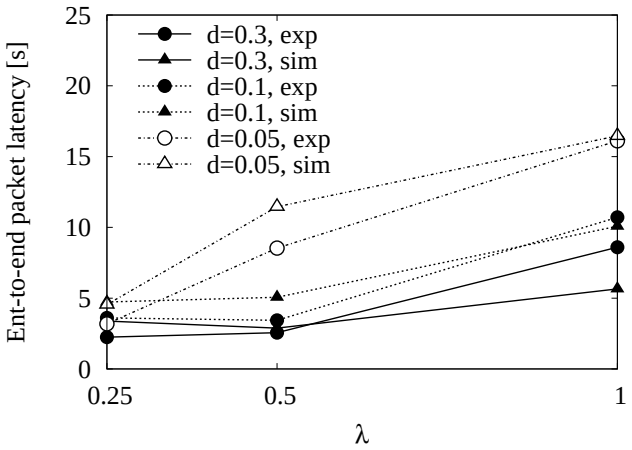


Figure 9. End-to-end latency.

Similar values have been observed for the simulation scenarios, where the average number of hops varies between 2.25 and 3.05 hops. This shows the consistency of the implementation of ALBA-R on the testbed and in the simulations, as well as the goodness of the topology used for simulations. Each experiment lasts 60 minutes. Results are collected after the network reached its steady state.

We investigate the performance of ALBA-R in terms of the ratio of packets successfully delivered to the sink versus those generated by the nodes, and in terms of end-to-end packet latency.

Packet delivery ratio. In both simulation and testbed results we observed a 100% packet delivery ratio, except for when $\lambda = 1.0$ and $d = 0.05$. In this case simulations predict a 99% average packet delivery ratio while the value collected from the testbed is 98%. This slight discrepancy depends on the maximum number of retransmission attempts per packet, after which a node discards a packet. We observed that this number of attempts is sometimes reached by nodes with very few neighbors, more frequently in the testbed than in the simulations. This effect is due to actual physical channel impairments (not considered in simulations).

End-to-end latency. Figure 9 shows the comparison between simulation and testbed experiments in terms of the time it takes to successfully deliver a packet from its source to the sink.

The observed trends are similar. The maximum difference between the values of simulation and testbed results is never more than 3 seconds. In order to understand the reasons of this performance difference, we investigate what contributes to end-to-end latency at the node and link level, breaking down local delays and their causes into the following components: Average number of re-transmission attempts for each successfully relayed packet; average time between when a node starts serving a packet and when it successfully delivers it to the next hop relay (contention time, in seconds); average number of backoffs experienced by each packet due to a busy channel (backoff busy); average number of backoffs experienced by each packet due to lack of awake eligible relays (backoff empty); percentage of lost ACKs, and average length of a burst of packets. Results concerning these metrics are displayed in Table 2 ($\lambda = 0.25$), Table 3 ($\lambda = 0.5$) and Table 4 ($\lambda = 1.0$).

We start by first considering the cases of low and medium traffic ($\lambda \leq 0.5$). For all considered values of the duty cycle, the end-to-end latency obtained through simulations is always higher than that from the testbed experiments. In the simulated scenarios, nodes experience a higher number of transmission attempts per packet, a higher average duration of each contention and a much higher probability that a node backs off for lack of an available relay. For instance, the simulated contention time is from 10% to 77% higher than that measured in the testbed. This is because it is typically easier to find a relay in practice (testbed) than in the simulations, where links with high PER are not included in the simulated topology. These links are instead sometimes used successfully to advance packets in the testbed. As expected, the difference is less remarkable for $d = 0.3$, as there is a higher chance to find an awake relay.

In the case of higher traffic and longer duty cycles ($\lambda = 1.0$ and $d \geq 0.1$), all behaviors contributing to latency in the low to medium traffic cases are still observed. However, the impact of re-transmissions because the channel is sensed busy ("backoff busy") becomes dominant. The sporadic transmissions over longer links observed in the testbed results in higher chances that transmission is detected through carrier sensing, causing a higher number of "backoff busy" and making latency

Table 2
Node metrics, $\lambda = 0.25$.

d	Scenario	Re-transmission attempts	Contention time (s)	Backoff busy	ACK lost (%)	Burst length
0.05	Testbed	0.62	1.37	0.4	6	1.18
0.05	Simulations	1.54	1.88	0.01	0.0	1.16
0.1	Testbed	0.39	1.17	0.27	2	1.1
0.1	Simulations	1.24	1.57	0.009	0.0	1.05
0.3	Testbed	0.26	0.85	0.22	3	1.06
0.3	Simulations	0.62	1.28	0.005	0.0	1.08

Table 3
Node metrics, $\lambda = 0.5$.

d	Scenario	Re-transmission attempts	Contention time (s)	Backoff busy	ACK lost (%)	Burst length
0.05	Testbed	1.49	2.35	1.06	12	1.69
0.05	Simulations	3.33	4.16	0.34	0.0	1.59
0.1	Testbed	0.61	1.37	0.49	5	1.15
0.1	Simulations	2.35	1.67	0.01	0.0	1.1
0.3	Testbed	0.44	1.08	0.39	4	1.09
0.3	Simulations	0.6	1.2	0.005	0.0	0.11

Table 4
Node metrics, $\lambda = 1.0$.

d	Scenario	Re-transmission attempts	Contention time (s)	Backoff busy	ACK lost (%)	Burst length
0.05	Testbed	3.61	4.05	3.11	13	2.39
0.05	Simulations	4.49	5.65	0.09	0.0	1.92
0.1	Testbed	3.11	3.47	2.81	14	1.94
0.1	Simulations	2.45	3.15	0.05	0.0	2.06
0.3	Testbed	2.67	3.20	2.48	11	1.62
0.3	Simulations	0.84	1.67	0.02	0.0	1.72

on the testbed higher than that measured through simulations. Moreover, real-life communications is impaired by higher interference and PER. Packet loss may be incurred in the testbed even when the channel has been successfully acquired. Also, the probability that an ACK is lost is negligible in the simulations, while it can be as high as 14% in testbed experiments. This motivates what shown in Table 4, which lists a higher number of backoff busy and a higher percentage of lost ACKs for testbed experiments than for simulations when $d \geq 0.1$, resulting in longer contention times and therefore in longer end-to-end latency.

As a final observation, we note that because of the grid-like topology used for the testbed, greedy paths can always be found from any source node to the sink. However, when we performed experiments in the Summer, because of the heavy foliage of the vineyard some of the links were no longer available, and connectivity holes actually arose in more than one experiment. For instance, we observed that node 2 was very seldom able to find a greedy path to the sink. In this case, we ran ALBA with and without the Rainbow mechanism presented here. In the first case, ALBA-R was able to allow node 2 to deliver its packets to the sink through nodes 3 and 15 after a short convergence time and a change of colors, as required by Rainbow, therefore improving the delivery ratio with respect to the case when Rainbow was not used (from 96% to 100%). In this case we also observed a slight increase in the average end-to-end latency, from 6.79 s to 7.74 s, as expected.

5 Correctness of Rainbow

We prove the correctness of the Rainbow mechanism described in Section 4 of the paper, i.e., we show that if there exists a route from a node to the sink, packets from that node can find it without getting stuck (and discarded) at dead ends. We start by showing that within a finite time from the start of the network operations, each node assumes a color C_h allowing it to decide where to find a viable next hop relay on a route to the sink S. We then show that routes that are so found are loop-free.

In the following we refer to network nodes through their unique identifiers x, y, etc. With $d(x)$ we indicate the distance of node x from the sink (e.g., the Euclidean distance). It clearly is $d(S) = 0$. The transmission area of each node x is partitioned into two regions F_x and F_x^C. The first region includes all nodes y such that $d(y) < d(x)$, i.e., all the neighbors of x that provide positive packet advancement towards the sink. Region F_x^C contains all neighbors of x such that $d(y) > d(x)$. In case $d(y) = d(x)$, y belongs to F_x if $x < y$, otherwise it belongs to F_x^C. A change in the region where a node x queries for relays (i.e., a change between F_x and F_x^C) is called an *alternation*. An alternation corresponds to x choosing a relay with a different color from its own. Alternations are characteristics of routes. For instance, a route where all nodes have color C_0 has no alternations.

A route whose nodes have only the colors C_0 and C_1 has one alternation, and so on. In the proofs below we assume that there always exists at least a route from each node x to the sink.

Theorem 1: All and only the nodes whose routes to the sink have at least h alternations assume the color C_h in finite time.

Proof: Let $N_F(x)$ and $N_{F^C}(x)$ be the sets of neighbors of a node x in the regions F_x and F_x^C, respectively. In the following we assume that upon starting network operations each node is colored C_0. We also assume that if at least a relay exists in F_x or in F_x^C, node x is eventually able to find it. In the proof we use the function $\Xi(x)$ to list the nodes in order of increasing distance from the sink. For node x that is the closest to the sink is $\Xi(x) = 1$; for node y that is second closest to the sink $\Xi(y) = 2$, etc. If two nodes x and y are equally distant from the sink, and $x < y$ we stipulate that $\Xi(x) < \Xi(y)$. We also use the function $\Xi'(x)$ to order the nodes by increasing number of alternations in their routes to the sink. If two nodes have routes to the sink with the same number of alternations, then the node closest to the sink has lower Ξ' (ties are broken like for Ξ).

The theorem is proven by induction on the number h of alternations of the routes from a node to the sink.

Case $h = 0$. We prove that a node x remains colored C_0 if there exists a route $x = x_k x_{k-1} \dots x_1 S$ from x to the sink S where each node x_i chooses its relay in F_{x_i} (no alternations), $1 \leq i \leq k$. We proceed by induction on $\Xi(x)$. If $\Xi(x) = 1$ the sink is in $N_F(x)$. In this case x keeps C_0 as its color because it finds the sink in F. This happens in finite time (the sink is always available). Let us now assume that for each y such that $\Xi(y) \leq k$, $k \geq 1$, the claim of the theorem holds true. Consider the node x such that $\Xi(x) = k + 1$. Since $h = 0$, there is at least one route $x = x_{k+1} x_k x_{k-1} \dots x_1 S$ from x to the sink S formed of relays chosen in the region F of each node. Then $d(x_1) \leq \dots \leq d(x_i)$, where $x_{i-1} \in N_F(x_i)$, $i > 1$. If for some i is $d(x_{i+1}) = d(x_i)$, then $x_i < x_{i+1}$. Therefore, $\Xi(x_1) < \Xi(x_i)$. To each of the x_i, $k \geq i \geq 1$, the inductive hypothesis applies, which means that every node x_i stays colored C_0, and this decision is taken in finite time. Now, $x = x_{k+1}$ has at least one relay x_k in $F(x_{k+1})$, and since x_k is C_0, x itself, i.e., the node with $\Xi(x) = k + 1$, remains colored C_0.

Let us now consider the case where x does not have routes to the sink without alternations (the "only" part of the "all and only" claim), namely, with $h > 0$. Each route from $x = x_k x_{k-1} \dots x_d$, where $x_{i-1} \in F_{x_i}$, $i \leq k$, reaches a node x_d that is a dead end. This node, not having any neighbors in its F_{x_d} region changes its color from C_0 to C_1 (within finite time). To prove the claim we can again proceed by induction on Ξ on the set $N'_{h>0}$ of nodes that have $h > 0$ and are not dead ends. By definition $N_F(y)$, for each node y in $N'_{h>0}$, contains only nodes that either belong to the same set (having a lower Ξ) or are dead ends. Let x be the first node in $N'_{h>0}$ according to the Ξ ordering. Being x the node in $N'_{h>0}$ with the lowest $\Xi(x)$, $N_F(x)$ includes only dead ends. Such dead ends assume color C_1 in finite time. When this happens, node x is no longer able to find relays colored C_0 in F_x and also gets colored C_1. Let us now assume, by inductive hypothesis, that the claim holds for all nodes y in $N'_{h>0}$ with $\Xi(y) \leq k$. We want to prove that it also holds for a node x such that $\Xi(x) = k + 1$. Node x, as explained above, has in $N_F(x)$ either dead ends or nodes $y \in N'_{h>0}$ such that $\Xi(y) < \Xi(x)$. For inductive hypothesis all these nodes assume a color $> C_0$ in finite time. Not finding relays colored C_0 that provide a positive advancement toward the sink, node x increases its color.

Case $h > 0$. We assume that the theorem holds for nodes with routes to the sink with h alternations, and we prove it for nodes with routes with $h + 1$ alternations. We consider two cases. Let us first assume that $h + 1$ is odd. We proceed by induction on $\Xi'(x)$. *Base case*. Let x be the node closer to the sink with routes to the sink with $h + 1$ alternations. Node x cannot be colored with any of the colors $C_0, \dots, C_h$ because there is no route from x to the sink with at most h alternations. In any route $x = x_k x_{k-1} \dots x_1 S$ from x to the sink S with $h + 1$ alternations, $x_{k-1} \in N_F(x_k)$, and moreover, $\Xi'(x_{k-1}) < \Xi'(x_k)$. Therefore, x_{k-1} has routes to the sink with h alternation and, by induction, converges to C_h in finite time, allowing node x_k to stay colored C_h. *Inductive step.* Assuming that the theorem is true for any x such that $\Xi'(x) \leq m$, we prove that it is also true for $\Xi'(x) = m + 1$. Let y be the node such that $\Xi'(y) = m$. A route $x = x_k x_{k-1} \dots x_1 S$ with $h + 1$ alternations leads to the sink, with $x_{k-1} \in N_F(x_k)$. Since $\Xi'(x_{k-1}) < \Xi'(x_k)$, node x_{k-1} assumes color C_h or C_{h+1} in finite time by induction, allowing x_k to stay colored C_{h+1}. Let us now consider the "only" case of the theorem statement. Let x_k be a node whose routes to the sink have $h' > h + 1$ alternations. By induction on h, node x_k cannot be colored with any of the colors between C_0 and C_h. We want to show that x_k cannot keep the color C_{h+1}. Let us assume that x_k could maintain the color C_{h+1}. Then, there exists a route $x = x_k x_{k-1} \dots x_c \dots S$ such that x_c is the first node on a route to the sink S with $h+1$ alternations. In this case, however, since each node x_i, $c \leq i < k$ is in $N_F(x_{i+1})$, the route from x_k to S would have $h+1$ alternations, which contradicts the fact that x_k has routes with $h' > h+1$ alternations. The case of $h+1$ even is similar to the previous one. In this case, we need to use a different ordering function $\Xi''(x)$ listing nodes in increasing order according to the number of alternations in their routes to the sink. Differently from Ξ', nodes farther from the sink precede the closer ones in case their routes have the same number of alternations. □

We now show that packets are delivered to the sink via loop free routes.

Theorem 2 (Rainbow is loop-free): The Rainbow extension to ALBA always finds loop-free routes.

Proof: We proceed by induction on the number h of colors. Case $h = 0$. We start by showing that routes

only made up of nodes colored C_0 are loop free. Let $x_k x_{k-1} \dots x_1 S$ be a route of $k > 1$ nodes all colored C_0 from a node x_k to the sink S. If a k-cycle existed, e.g., $x_1 = x_k$, then $d(x_1) = d(x_k)$. However, C_0 nodes forward packets only if $d(x_k) \geq d(x_{k-1}) \cdots \geq d(x_1)$. If $d(x_1) = d(x_k)$, because of the definition of F it would have to be $x_k < x_{k-1} < x_1$, which is a contradiction.

Case $h > 0$. We assume that all routes with nodes colored with colors from C_0 to C_h are loop free, and we prove the theorem for routes whose nodes are colored with colors from C_0 to C_{h+1}. We consider the two cases depending on $h+1$ being either odd or even. If $h+1$ is odd, nodes colored with C_{h+1} search for relays in their region F. Let us consider a route $x_k \dots x_{j+1} x_j \dots x_1 S$ from a node x_k colored C_{h+1} to the sink S passing through nodes colored C_{h+1}, C_h and so on. Let x_j be the first node colored C_h. The route $x_j \dots S$ is loop-free by the inductive hypothesis. The route from x_k to x_{j+1} is made of all nodes colored C_{h+1}. Such route is also loop free. If a cycle existed, e.g., $x_k \dots x_{k'} \dots x_k$, $k' > j$, then given the forwarding rules it would be $d(k) = d(k')$ and $x_k < x_{k'} < x_k$, which is a contradiction. Finally, it cannot occur that $x_u = x_v$ for nodes $u \leq j$ and $v > j$, since each node assumes a unique color. The case when $h+1$ is even is similar to the previous one, considering F^C instead of F. □

Theorem 1 and Theorem 2 hold for any topology graph G, since the definition of $N_F(x)$ and $N_{F^C}(x)$ as given in Theorem 1, for each node x, is based on a neighborhood relation that does not depends on a node nominal transmission radius and distance. Therefore, ALBA-R works also in those more realistic settings that cannot be accurately modeled by unit disk graphs (UDGs [9]), as instead required by many previous solutions.

REFERENCES

[1] C. Petrioli, M. Nati, P. Casari, M. Zorzi, and S. Basagni, "ALBA-R: Load-balancing geographic routing around connectivity holes in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, 2013, in print.

[2] P. Casari, M. Nati, C. Petrioli, and M. Zorzi, "A detailed analytical and simulation study of geographic random forwarding," *Wireless Communications & Mobile Computing*, July 21 2011, dOI: 10.1002/wcm.1152.

[3] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, October-December 2003.

[4] M. Zorzi, "A new contention–based MAC protocol for geographic forwarding in ad hoc and sensor networks," *Proc. IEEE ICC*, vol. 6, pp. 3481–3485, Jun. 2004.

[5] A. Camillò, M. Nati, C. Petrioli, M. Rossi, and M. Zorzi, "IRIS: Integrated data gathering and interest dissemination system for wireless sensor networks," *Ad Hoc Networks, Special Issue on Cross-Layer Design in Ad Hoc and Sensor Networks, L. Galluccio, K. Nahrstedt, and V. R. Syrotiuk, eds.*, 2012.

[6] D. Dubhashi, O. Häggström, L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti, "Localized techniques for broadcasting in wireless sensor networks," *Algorithmica*, vol. 49, no. 4, pp. 412–446, 2007.

[7] M. Rossi, R. R. Rao, and M. Zorzi, "Statistically assisted routing algorithms (SARA) for hop count based forwarding in wireless sensor networks," *Springer Wireless Networks Journal*, vol. 14, no. 1, pp. 55–70, Feb. 2008.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[9] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–167, 1990.