

ALBA: AN ADAPTIVE LOAD-BALANCED ALGORITHM FOR GEOGRAPHIC FORWARDING IN WIRELESS SENSOR NETWORKS

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

In this paper we propose and analyze ALBA, an original packet forwarding protocol for ad hoc and sensor networks. ALBA follows an integrated approach that combines geographic routing and medium access control (MAC), exploiting the knowledge of node positions in order to achieve energy-efficient data forwarding. The scenario we consider is very critical for medium-high traffic, as contentions for channel access and the resulting collisions lead to performance degradation. To counter this effect, we leverage on network density, favoring the choice relay candidates that are not in overload. With our protocol, nodes strive to channelize traffic toward uncongested network regions, rather than just maximizing the advancement towards the final destination.

We carry out extensive simulations that compare ALBA to GeRaF and MACRO, two recently proposed cross-layer approaches with similar goals. The results show that our design achieves very good delivery and latency performance, and can greatly limit energy consumption.

I. INTRODUCTION

Wireless sensor networks are a very promising enabling technology for future information retrieval and ambient-interactive networking. Typical wireless sensor nodes are relatively cheap, and are expected to become even cheaper in the near future, making it feasible to rapidly deploy wireless networks formed by hundreds or thousands of nodes.

Indeed, sensor nodes present very strict limitations in terms of available resources. Foreseen scenarios, such as tactical as well as environmental surveillance, target tracking, ambient intelligence, etc., require that, after initial deployment, wireless sensor network work unattended, possibly for very long times. During this period, the network has to do its best to ensure information retrievability and to maximize its own lifetime, despite the scarce available resources. For this reason, it is very important that protocols are designed with energy efficiency as one of the main criteria.

Sensor nodes are specially suitable for use in very dense networks, where the unavailability or even the loss of a single node is compensated for by the inherent redundancy, for example by having some of the node’s neighbors perform the required tasks. Scenarios of interest for these systems include, e.g., tactical networks and battlefield

situations. Nodes could be parachuted on a wide scale from an airplane to perform movement detection over a large territory with fine sensing granularity. They would then set up an ad hoc network and begin reporting sensed data back to a base station. Direct intervention (e.g. to recharging nodes batteries) over such an unattended network would be highly impractical, hence it is very important to perform message exchanges in the most energy-efficient fashion.

An accurate cross-layer design, that tries to jointly optimize both MAC and routing behavior, seems to be the best paradigm for such networks [1]. To this end, we deploy an Adaptive Load-Balanced Algorithm for wireless sensor networks (ALBA), an integrated MAC and routing protocol, partially derived from an approach previously proposed in the literature [2], [3]. ALBA focuses on some protocol features that can substantially enhance latency and transmission success performance. In particular 1) it comprises mechanisms to re-route packets which reached a dead-end (so increasing the percentage of delivered packets in sparse and moderately sparse networks); 2) it balances the traffic load among the different nodes, thus reducing congestion; 3) it allows transmission of a burst of packets to the selected next hop relay thus reducing the control overhead per packet due to relay selection. We evaluate ALBA in a dense multihop sensor network, and compare it to MACRO [4], a recently proposed integrated MAC/routing approach that also strives to achieve substantial energy savings through the estimation of the position of the more convenient next hop. A comparison is also performed with GeRaF, from which ALBA is derived. ns2-based simulation results show that ALBA is able to significantly improve over these two protocols.

The remainder of the paper is organized as follows. Section II describes ALBA operations while Section III sketches the protocols considered in our comparison, namely MACRO and GeRaF. In Section IV, we show by extensive simulation results that our approach can compensate for different kinds of network impairments and compares favorably with MACRO and GeRaF. Section V concludes the paper.

A. Related Work

The use of geographic information allows nodes to avoid the need for network topology information. Provided that a means for location estimation exists, this is a considerable advantage. The literature on geographic routing for wire-

less sensor networks (a recent survey can be found in [5]) is mainly focused on designing the algorithms themselves. For example, from the early MFR, DIR and GEDIR [6], to the more recent FACE [7], GEAR [8], different solutions have been proposed for convergecasting transmissions or forwarding queries to geographically restricted regions.

In the pursuit of energy-efficient solutions, some works extend algorithm design to account for link costs, identifying them with a given metric. Among those, [9] optimizes the geographic advancement based on link reliability, preferring shorter hops when longer ones are infeasible because of losses. [10] accounts for weighed advancement, where the weight is appropriately defined to incorporate suitable performance metrics, such as link reliability (through packet error rate), delay, or energy consumption. In [11], the authors extend [6] by accounting for reliability in MFR or DIR routing. Link quality, along with other performance metrics, is also considered in [12].

In [2], [3] and further modifications [13], Geographic Random Forwarding (GeRaF) is presented. It is a contention-based integrated MAC and routing approach, which divides the region offering positive progress toward the final destination into various zones, and pings them subsequently [3], or according to some other criterion [13], in order to find a node available for relaying.

Another cross-layer approach recently proposed is MACRO [4]. MACRO protocol suite combines a scheme for node awake/asleep modes, a MAC and a geographic-based routing protocol. Relay selection is performed based on a function of both the advancement toward the sink and the energy consumption needed for packet transmission. A more thorough description of MACRO is provided in Section III.

In this paper, we will deal with the cross-layer integration of geographic forwarding with relevant link metrics in wireless sensor network. To this end, we develop a solution that encompasses routing and access control with congestion avoidance and dead-end detection, with the aim of being simple and energy-efficient, while also offering some degree of reliability. The protocol we consider is described in the next Section.

II. ADAPTIVE LOAD-BALANCED ALGORITHM (ALBA)

ALBA stands for *Adaptive Load-Balanced Algorithm*, and is a holistic approach that integrates routing and Medium Access Control (MAC) in wireless ad hoc and sensor networks. It is designed to operate in a convergecasting scenario, i.e., where a central node (the sink) wants to gather data from a sensor network for a posteriori processing, storing or delivery. The protocol relies on some limited geographic information to be stored in any node, namely the node's own position and the location of the sink. A basic form of geographic routing is constructed

on top of that and extended through a finer adaptation to local traffic and congestion as perceived by the sensors. Furthermore, the protocol is designed to perform load-balancing. It tries to uniformly disseminate forwarding queries among the neighbors of any transmitter, discarding the most heavily congested nodes while preferring those with shorter backlogs. Finally, the whole approach is integrated and distributed, in the sense that forwarders dynamically change some key parameters according to their own network conditions, and do not require any coordinator to supervise this operation.

ALBA is based on the widely accepted idea that idling is a major source of power consumption in wireless sensor networks. Practical battery constraints call for energy efficient MAC and routing protocols, that enable communications without compromising network lifetime. Energy efficiency in packet radio sensor networks is obtained not only during radio sleep modes, but also through effective signaling whenever needed, by avoiding useless listening times, and by rapidly and effectively selecting the relaying nodes. In the definition of ALBA, we will follow these guidelines.

ALBA works by assigning two priority values to all eligible forwarders. First of all, a node in range of the transmitter is considered "eligible" if it is inside the forwarding area, i.e. it offers positive advancement toward the sink. The forwarding area is divided in N_r slices or regions, $0, 1, \dots, N_r - 1$, such that all nodes located in region i are closer to the sink than any node in regions $i + 1, i + 2, \dots$. Hence, region 0 offers the maximum advancement. The *geographic priority index* (GPI) of a node is defined as the region where it is located. For understanding its own location, each node has to rely on some localization method. A survey of such methods can be found in [14].

Furthermore, a *queue priority index* (QPI) is chosen dynamically by forwarders as follows. Each time a node sends data to the next hop, it adopts a back-to-back paradigm, and tries to transmit a whole sequence of up to M_B packets. The receiver reports back an ACK message for each packet in the group, according to whether or not the transmission was successful. Let the maximum number of packets sequentially sent without errors be $M_B^c \leq M_B$. Let the current queue occupancy of the node be indicated as Q . Whenever a node is asked to relay a packet, it is also communicated the number of packets N_B that will be sent back-to-back, i.e., the length of the data burst. The queue priority is then calculated as $\min \{ \lceil (Q + N_B) / M \rceil - 1, N_q \}$ and takes values between 0 and N_q , where N_q is chosen to be an upper bound to the number of different QPIs. M is the expected length of the burst which can be successfully transmitted back-to-back by the relay node. It is computed at the node based on the past history (it is given by a weighed average of the observed M_B^c). Whenever there

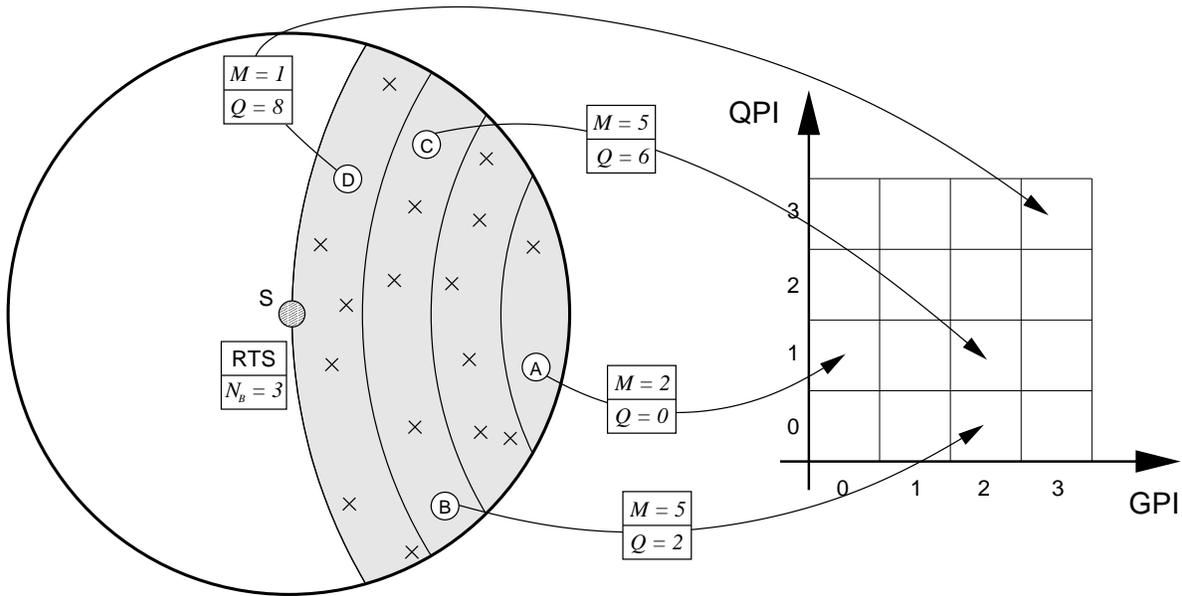


Fig. 1. Sample assignment of QPIs and GPIs to awake nodes upon RTS reception.

are fewer than M_B packets in the queue, if they are all successfully transmitted in a single burst we optimistically assume that a burst of M_B packets could be sent correctly.

The rationale behind accounting for the queue priority is for relays to measure how easy and fast it will be for them to further propagate data. The QPI captures the estimated number of burst transmissions needed before the new data can be relayed further. Since the choice of good relays is critical to set up a better route, nodes with low QPI are preferred. In fact, if a node finds it difficult to send out packet bursts (low M), then the channel is likely to be experiencing a temporary excess use. This makes the node a bad candidate relay. The same reasoning applies if the node has a very high queue level. In either case, the QPI is greater, and this will reduce the probability that the node can be chosen as a relay. On the contrary, if a node has a very low queue, or it is always able to successfully forward long data bursts (high M), then it should be addressed more frequently.

Fig. 1 illustrates the priority assignments. Node S is the sender and requests a total data burst of 3 packets. The forwarding area is identified in light gray, and the forwarding regions are delimited by circle arcs centered at the sink (supposed to be quite far away and not displayed). Asleep nodes are shown as crosses. The only awake sensors are thus nodes A, B, C and D. Node A has an empty queue but $M = 2$, so it is assigned a QPI of 1. Node D also has $M = 1$ but its backlog is longer ($Q = 8$), hence its queue priority index is 3. Similarly, nodes B and C both have $M = 5$, but B has a smaller queue. Therefore, B is assigned queue priority 0, whereas C is assigned 1.

In order to reduce energy consumption due to idle listening, ALBA assumes that all nodes alternate between awake and sleep states. Once a node has a packet to

send, it senses the channel to avoid collisions. The sensing time is chosen to be long enough so that incomplete ongoing handshakes are detected with high probability [3]. If the channel is sensed idle, an exchange of messages for collision avoidance occurs. First, the transmitter broadcasts a Request-To-Send (RTS) message, including enough significant information to allow the eligible forwarders to identify both the geographic and the queue priority regions. Namely, the RTS contains the location of the transmitter and the number of packets in the requested data burst, N_B . The position of the sink is assumed to be known. Based on gathered geographic data, each node is able to compute its own region and geographic priority index. Moreover, each node knows its own queue level, Q and the expected length of a successful burst, M . Along with N_B , these allow to compute the QPI.

Only eligible nodes which are awake and idle can respond to the RTS. In particular, all neighbors hearing the RTS and finding themselves outside the forwarding area return to sleep immediately. The others report back a Clear-To-Send (CTS) message based on their QPI.¹ Nodes with QPI 0 respond immediately, giving rise to three possible events. If only one node responds, it is the winner and it is sent the data packet. If none responds, the transmitter uses a further RTS to query the nodes with QPI 1 and waits for an answer, continuing similarly until a response is heard. If more than one CTS is received, multiple nodes share the same QPI, and the transmitter tries to locate the one offering the best advancement, i.e., having the best GPI. To this end, nodes with GPI 0 are called first, and the same answering procedure used for

¹Each CTS specifies how many packets the relay can receive, in order to avoid buffer overflows.

QPI is replicated. Finally, if the sender does not receive any kind of reply after having scanned all QPIs, it backs off and retries at a later time. A maximum number N_{MaxAtt} of failed attempts is set, after which the node discards the packet. In Fig. 1, for example, node S sends an RTS to query the nodes with QPI equal to 0. Only one CTS is received by node B. In this case node B will be selected as next-hop relay. Let us now discuss what would happen in case only nodes A, C, D were awake. After a first unsuccessful attempt (no node with QPI 0 exists in this case), A and C would become involved (they both have QPI 1). Their CTSs would collide, so the best GPI search begins. A would be the final winner, since it is the only node with GPI 0. Note that if some nodes have the same QPI and GPI, their CTSs would collide again. This event is solved through binary splitting [2].

Consecutive data packets are sequentially sent to the contention winner, as many as indicated in the control messages. Contention losers overhear the data packet, understand that it is not meant for them and go back to sleep. All data packets are individually acknowledged. More specifically, if an ACK is missing for any reason (collisions, channel errors, etc.) the sender stops transmitting and accordingly updates the value of M_B^c . Packets in the current burst that have not yet been sent are rescheduled for a later attempt after a backoff period. During this backoff, nodes follow their normal duty cycle, and may respond to relaying requests if needed.

As a final detail, we highlight that the awake/sleep schedules of the nodes take place with the same average duty cycle d , but are completely asynchronous. Therefore, some eligible relays may happen to wake up during a contention. In this case, a node can actively take part in the contention only if the best QPI is still being searched for. Whenever the QPI of the contenders has been chosen (i.e., the best GPI search phase has started), no node can enter the contention. Other than this, the participation of these nodes to the ongoing contention is always subject to the rules described before.

ALBA also includes a mechanism for dealing with dead-ends, i.e., with those nodes that cannot find relays in the direction of the sink. According to this much, each node y divides the circle centered in itself and with radius r (transmission range) into two parts. The first one, denoted F , is the forwarding region introduced before (colored gray in Fig. 1), the second one, namely F^C , is the remaining portion of the coverage area (white in Fig. 1). Originally, all nodes are labeled ‘yellow.’ They route information according to the ALBA operations described above (i.e., selecting relays among nodes in F). Whenever a node y has unsuccessfully attempted to forward packets for a number

of times high enough² to believe that there are no neighbors in F , it considers itself a dead-end. Being a ‘‘bad relay,’’ y exponentially decreases its own likelihood to participate as eligible forwarder in contentions initiated by other nodes. This is the way nodes on branches going only to dead-ends progressively realize that those branches are no good for routing packets to the sink, and hence they stop proposing themselves as relays for other nodes.

A node which cannot advance packets toward the sink switches its color to ‘red.’ Red nodes handle a packet that they generate or that they receive according to a different rule: the packet is sent away from the sink selecting as relay yellow or red nodes in F^C . This process is repeated until a yellow node is reached. Starting from the yellow node, ‘yellow (i.e., regular) ALBA operations’ are resumed. The packet is forwarded to the sink along a route which goes only through yellow nodes: the yellow brick route. If a red node is unable to find relays in F^C it progressively stops proposing itself as relay for other red nodes, eventually switching to ‘blue.’ Blue nodes do not have yellow neighbors but they could have a red neighbor in the F area. They will not candidate themselves as relays for messages sent by red or yellow nodes. However they will search for a route for their own packets by asking other blue or red neighbors in F to play as relay (giving priority to red nodes). When blue nodes fail too much forwarding the packets, they again assume to be a dead end and try routing around. They do this by switching to ‘violet.’ Such nodes search for relays in F^C , looking for either blue or violet nodes, giving priority to blue ones.

The rationale behind this re-routing mechanism is simply explained with the need to both ensure that a ‘yellow brick’ path is eventually reached and that dead end nodes are able to send back the packet for some hops. Also note that a node starts from yellow, and when experiencing bad network conditions, it sequentially switches to red, blue and violet. Yellow and blue nodes look for relays in F , while red and violet nodes address F^C , thus favoring changes of the routing direction. The mechanism can be generalized to work with any sequence of colors C_1, \dots, C_n . Following the same guidelines, odd colors will search for relays in F and even colors in F^C . It is possible to prove that this technique finds loop-free routes [15].

Note that ALBA combines energy saving (through duty cycles, backoffs, and shutdown of useless nodes), MAC (through sender-initiated contentions) and routing in a seamless way. This cross-layer design results in better efficiency thanks to the integration of different network layer operations, while striving to be simple and easily implementable on nodes with limited computational capa-

²This number has been tuned via extensive simulation and selected not to lead to false positives when varying the scenarios parameters in realistic ranges.

bilities. The joint iterated optimization, first on the queue length, and then on the advancement toward the sink, is necessary for selecting the best relays, i.e., those with shorter queues and/or that transmitted successfully in the past. The QPI is maximized first because we found that there is no point in looking for the best advancement, when the addressed receiver cannot support the required amount of traffic [13]. It is instead more convenient to look for uncongested nodes offering a good probability of advancement, and to achieve the best progress only among those.

III. MACRO AND GERAF

In this section we describe MACRO [4] and GeRaF [3], [13], the two protocols used as benchmarks in our comparative performance evaluation.

In MACRO all sensors follow a sleep/awake schedule similar to ALBA. MACRO however uses a different MAC and a different relay selection rule. Relay selection is based on the relay “weighed progress,” i.e., the advancement toward the destination divided by the power needed to reach that relay. In the following we will use the terms “weighed progress” and “gain” interchangeably. In MACRO, each node divides the forwarding region in N_m zones. A prescribed power level P_i suffices to reach any node in zone i , with $P_i < P_{i+1} \forall i$. Let r be the maximum transmission range. The i -th region comprises all neighbors whose Euclidean distance from the source is smaller or equal to $r_i = i * r / N_m$.

The operations performed by node x and its neighbors when searching for a relay in a given relay region are the following. Node x wakes up all its neighbors within the relay region by transmitting WAKE_UP messages addressed to that region over a time T_{cycle} . T_{cycle} is selected long enough to be sure that each neighbor will wake up and will be able to receive the WAKE_UP message before T_{cycle} expires. Node x then transmits a GO message asking for possible relays. x 's neighbors compute their gain, and based on the gain value, they will compute a random time (the higher the gain the lower this time) to wait before they transmit a CONTROL_ACK message with which they candidate themselves as possible relays. Such times are upper bounded by a value WT (Wait Time), which is one of the protocol parameters. CONTROL_ACK messages include the neighbor's identity and associated gain. Whenever node x receives a CONTROL_ACK message it checks if it has a better gain than the ones received so far. It also computes the probability of receiving a CONTROL_ACK message from a neighbor with a better gain located in the same relay region. If such probability is greater than a threshold p_{th} , it will keep waiting for an additional message (or for the timeout WT expiration). Otherwise, node x computes the expected highest gain that will be found by searching the next relay region. Only if such value is greater than the gain

of the best relay already discovered, is the next relay region searched. At the end of this procedure, node x selects the discovered neighbor with the highest gain as next hop relay and forwards the data packet to it.

On one hand, MACRO is a completely distributed, on-line, cross-layer algorithm, which jointly optimizes transmit power consumption and geographic advancement. On the other hand, it needs to wake up all nodes before any handshake, which consumes both time and power.

GeRaF [2], [3] and GeRaF⁺ [13] are also of interest for our study and are briefly summarized here. GeRaF divides the positive advancement region in, say, N_r circular slices. Unlike MACRO, these slices are not centered in the transmitter but in the sink. They are used to discretize the advancement offered by different neighbors and to choose the best one. GeRaF assumes that every node knows its own position and the location of the sink. Every other relevant geographic information is piggy-backed in signaling messages. Each transmitter issues an RTS message whenever it has a packet to transmit. The message reaches all neighbors and initiates a contention among relays. The nodes in the farthest region from the sender respond first with a CTS message. If more than one reports back, the sender issues a COLLISION message to solicit the choice of a single node. This is distributely achieved through binary splitting.

If a region is empty (e.g., because all nodes located in it are sleeping) the transmitter issues a CONTINUE message to solicit the following region. When the first nonempty region is found, a contention among nodes is originated as specified before. After having identified a relay, GeRaF sends out data, waits for an ACK, and then lets the sender go back to sleep, in order to allow the relay to forward the packet. A backoff state is entered whenever the node cannot succeed in finding any relay. In the GeRaF⁺ variant, nodes entering the awake state during a contention can participate, provided that they belong to a region which has not already been queried.

GeRaF is designed to integrate MAC message exchanges and the designation of the most convenient relay (from a geographic point of view). Thanks to awake/sleep cycles and to this cross-layer design, it is very energy-efficient. Moreover, it is simple and easy to implement on real nodes. However, it has some drawbacks, e.g., it cannot route around connectivity holes, and thus may not be able to deliver all messages in sparse networks (because of the physical absence of nodes). Also, it is not able to operate in dense traffic scenario (when congestion builds up).

In the next Section we provide simulation results for ALBA and carry out a comparison among our approach, GeRaF and MACRO.

IV. PERFORMANCE EVALUATION

In this section we report the results of a comparative performance evaluation aimed at assessing ALBA's effectiveness 1) in reliably delivering the generated packets to the sink; 2) in performing convergecasting according to a low-energy, low-overhead paradigm; and 3) in providing good trade-offs between energy efficiency and end-to-end packet latency. ALBA, GeRaF [2], [3] and MACRO [4] have been implemented using the VINT project network simulator ns-2 [16]. GeRaF is a natural benchmark for ALBA as ALBA has been designed to address its performance limits outlined in [13]. MACRO has been selected for comparison as it is a recently proposed cross-layer solution, designed to perform better than geographic greedy forwarding schemes. Instead of selecting as relay the active neighbor closest to the sink, in MACRO a node x which has a packet to transmit wakes up and inquires all the neighbors in its forwarding area. Each neighbor y has associated a gain given by the ratio of the advancement toward the sink which would be obtained by relaying the packet to y and the energy consumption needed to transmit the packet from x to y . Out of the neighbors in its forwarding area, x selects that with the highest gain as relay, in this way trading-off energy consumption and latency.

The three protocols have been compared in a scenario in which $n = 600$ nodes are randomly and uniformly deployed in a square area with side $L = 160$ m. The sensor nodes' transmission range r varies between 15 m, 20 m and 30 m (all realistic transmission ranges according to currently available sensor nodes prototypes). These ranges correspond to scenarios where nodes have an average nodal degree equal to 16, 30 and 66, respectively.

In our simulations, nodes alternate between awake and asleep states according to a predefined schedule with a duty cycle $d = 0.1$. The energy consumption when transmitting, receiving and when asleep follow the *first order energy model* outlined in [17]. The energy consumed per bit when receiving E_{Rx} is constant, while the energy consumed per bit when transmitting $E_{Tx}(r)$ is expressed by the following equation:

$$\begin{aligned} E_{Tx}(r) &= E_{Tx-elec} + E_{Tx-amp}(r) & (1) \\ E_{Tx-amp}(r) &= \epsilon_{amp} * r^2 & (2) \end{aligned}$$

The first term $E_{Tx-elec}$ accounts for the energy needed to run the transmitter circuitry (and is set equal to E_{Rx}), while the second term $E_{Tx-amp}(r)$ accounts for the emission power. The latter term depends on the transmission range, as shown in eq. (2). According to this energy model, $E_{Tx-amp}(r)$ and $E_{Tx-elec}$ have comparable values when $r = 22.5$ m, after which the transmission power $E_{Tx-amp}(r)$ becomes the dominant factor in eq. (1). The

energy cost when asleep is assumed equal to 1/1000 of the cost when in the receive state.

Data traffic is generated according to a Poisson process with parameter $\lambda \in \{0.01, 0.05, 0.1, 0.25, 0.5, 1.0, 2.0, 4.0\}$. A packet arrives to the network and one of the sensor nodes is randomly and uniformly assigned to the packet as its source. Nodes have packet buffers with size equal to 20 packets. A newly generated packet is *accepted* by the source node if its buffer size is not full. All packets are addressed to the sink which is randomly placed in the deployment area. Convergecasting of the sensed data to the sink is performed according to the specific protocol considered (ALBA, MACRO or GeRaF). Data packets have a length equal to 250 bytes, while the size of control packets is set to 25 bytes. The channel data rate is 38.4 Kbps.

All the topologies we have considered for our experiments are connected, i.e., there is always at least one route between any pair of nodes in the network. Since we consider WSNs with random and uniform deployment of the sensor nodes, it might be the case that a node does not have any available relaying neighbors, in the direction of the sink. This would result in unsuccessful packet delivery, unless additional mechanisms are introduced for re-routing the packet (which have been included in ALBA but are not accounted for in MACRO and GeRaF). We have verified however that such cases rarely occur for sufficient node densities, as considered in this paper for ALBA, MACRO and GeRaF. The specific parameters settings are summarized in the following Tables I–III.

All the protocols divide the area where possible relays are located into four different relay regions. They all adopt the same carrier sense length T_{Sense} , the same backoff interval length $T_{Backoff}$, and they all attempt to forward a data packet to the next hop for a maximum of N_{MaxAtt} times. As for the specific ALBA's parameters, the maximum number of packets which can be transmitted back-to-back in a burst has been tuned and is equal to 5, while the possible queue lengths have been divided into $N_q = 4$ groups.

The MACRO parameters p_{th} , T_{cycle} and WT have been tuned. Their values are reported in Table III.

The performance of the three protocols has been evaluated with respect to the following metrics of interest.

- 1) *Packet delivery ratio*: defined as the percentage of generated packets which are successfully delivered to the sink.
- 2) *End-to-end packet latency*: defined as the time from when a packet is generated to when it is delivered to the sink.
- 3) *Node energy consumption*: this metric refers to the average energy consumed by nodes over a given timeframe normalized to the energy nodes would consume by strictly following the duty cycle.

TABLE I
COMMON PARAMETERS.

Name	Value	Description
N_r	4	Number of relay regions
T_{Sense}	0.0521 s	Carrier sense length
$T_{Backoff}$	1.095 s	Backoff interval length
N_{MaxAtt}	50	Max. number of attempts for searching a relay

Figs. 2 to 6 display the results of our comparative performance evaluation. Results have been obtained by averaging over 100 simulation runs.

In particular, Figs. 2 to 6 compare the performance of ALBA, MACRO and GeRaF when $r = 20$ m. MACRO results are displayed only for $\lambda \leq 0.25$. For larger traffic loads the protocol performance significantly degrades (a significant percentage of packets are lost, and latencies are very large). GeRaF and ALBA scale to higher traffic loads (no packets are lost and end-to-end latencies are limited to a few seconds for $\lambda \leq 4.0$). This is due to the much higher overhead of MACRO's relay selection scheme over ALBA's and GeRaF's.

As GeRaF and ALBA favor low latency (maximum advancement toward the sink) while MACRO also accounts for the energy needed to forward a data packet toward a given relay, the energy consumed for end-to-end data packet transmission is lower in MACRO than in GeRaF and ALBA. This is clearly shown in Fig. 2 which compares the three protocols for $r = 20$ m.

This metrics only accounts for the energy consumed to transmit and receive the data packet over the selected route: the energy spent for relay selection is not included. We observe that ALBA is the one with the highest energy consumption to transmit a data packet along the route to the sink. Such metric is given by the average number of hops traversed by a packet times the average energy consumption needed to transmit the packet to the next hop. ALBA is the protocol leading to longer routes: 8.4 hops on average vs. the 7.4 hops experienced when GeRaF is adopted. This is due to ALBA's relay selection rule: a node with a better QPI is chosen even if it does provide a large advancement toward the sink. The energy consumption needed to transmit the data packet to the next hop relay is instead exactly the same in GeRaF and in ALBA. In both protocols the emission power is the same (no power control is used, nodes transmit always at the maximum emission power). Being the number of traversed hops lower in GeRaF than in ALBA, and being the energy consumed to relay a packet to the next hop the same, the energy consumed to transmit a data packet end-to-end, from the source to the sink, is lower in GeRaF than in ALBA.

MACRO leads both to shorter routes (6.7 hops long, on average) and to lower energy consumption to transmit a data packet to the next hop. The latter is due to the fact

TABLE II
ALBA PARAMETERS.

Name	Value	Description
M_B	5	Max burst length
N_q	4	Number of queue size regions

TABLE III
MACRO PARAMETERS.

Name	Value	Description
T_{Cycle}	0.16 s	Duration of the wakeup phase
WT	0.3 s	Maximum Wait Time for CONTROL_ACKs
p_{th}	0.3	Threshold prob. for achieving a better gain

that MACRO exploits power control (transmitting at the minimum emission power able to reach the relay region where the selected relay is located). MACRO reduced route length is somehow less intuitive. In short range scenarios as the one being considered, the energy consumption reduction that can be achieved by selecting relays closer to the source is quite limited (being the factor $E_{Tx-elec}$ non negligible). Indeed, MACRO selects as relays the nodes with the highest gain which are located in this case either in the third or in the fourth relay region (which contains the neighbors closest to the sink). GeRaF is not always able to find relays so close to the sink, since it operates only on the subset of *active* neighbors. On the contrary, MACRO wakes all the neighbors and then selects the best one out of them. As a result, on average MACRO leads to higher advancements (thus, shorter routes).

A main drawback of MACRO is the high cost to be paid for relay selection. The average number of bytes transmitted by nodes involved in a relay selection procedure to identify the best next hop relay is reported in Fig. 3. While GeRaF and ALBA have very similar performance, and prove themselves to be lightweight protocols, MACRO suffers from a much higher overhead (more than ten times as large) due to the high message exchange associated to neighbors wake up. Around 750 bytes are transmitted by a node to wake-up all its neighbors in a given relay region *à la* STEM [18]. Given that on average the selected relay is found in the third or fourth relay region searched, the overhead due to WAKE_UP messages sums up to between 2250 and 3000 bytes. WAKE_UP messages thus are the major component of the overall relay selection overhead. The overhead associated to GO and CONTROL_ACK messages is instead negligible, since they are short, and at most four GO s and one CONTROL_ACK is needed for each potential relay. The number of eligible relays for a node is limited to around 15 in this scenario.

A corresponding performance degradation can be observed in the time needed to identify the next hop relay. In MACRO, each relay region inquiry requires 0.16 s to wake up all the nodes in the region, plus up to 0.3 s to gather all the gains of the nodes in the region. Given that

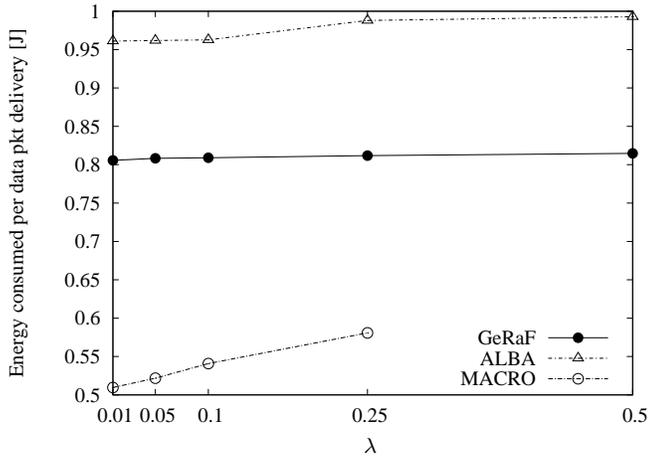


Fig. 2. Avg. energy consumed to deliver a data packet to the sink [J].

two or three regions are usually scanned unsuccessfully before the next hop relay is identified, more than 1 s passes before scanning the relay region where the next hop relay is located. This justifies the relay selection duration which is way longer than in ALBA and GeRaF. As the traffic load increases, collisions during the transmission of the data packet may occur. This results in the need to perform again the relay selection phase, which in turn causes a longer contention duration. This effect is not evident in ALBA and GeRaF. Being much more lightweight the probability of collisions is much lower in these protocols.

In ALBA (GeRaF) the relay regions are searched from the one with the highest QPI (GPI) down to the one with the lowest QPI (GPI). As soon as some nodes answer the RTS packet, a relay will be selected within that region. Collisions may occur due to multiple potential relays answering simultaneously but they are quickly solved using a splitting algorithm. All these operations can be performed quickly, and result in a significant advantage in terms of per hop latency, and overall end-to-end latency, as shown in Fig. 5.

MACRO's much higher overhead is also critical in terms of overall node energy consumption (Fig. 6).

Despite MACRO's power control techniques and shorter routes found, the many control packets needed for selecting a relay and the higher probability of collision lead to a greater number of bytes transmitted and received over the network, thus to an overall higher energy consumption. The increase over ALBA and GeRaF can be as high 12%.

No significant difference is noticeable between ALBA and GeRaF in dense, low traffic scenarios such as the one considered here. This is the only type of traffic scenario in which all the three protocols correctly operate (MACRO suffers severe degradation at medium-high traffic, GeRaF and MACRO lose packets due to dead-ends in sparse and moderately sparse scenarios).

We have carried out extensive simulations to assess the performance of ALBA in high density and/or high traffic

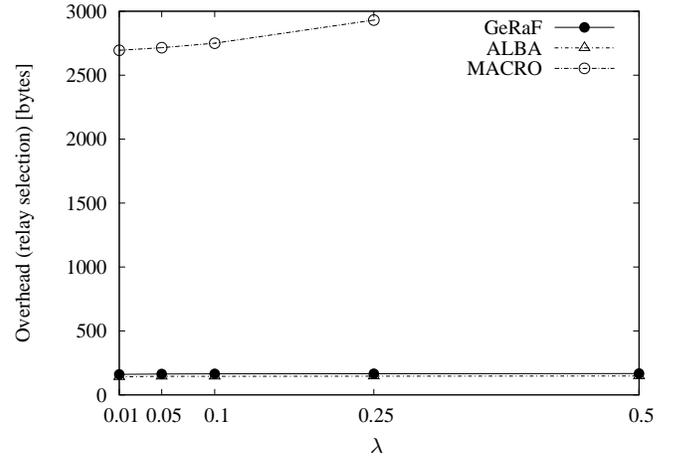


Fig. 3. Relay selection: overhead [bytes].

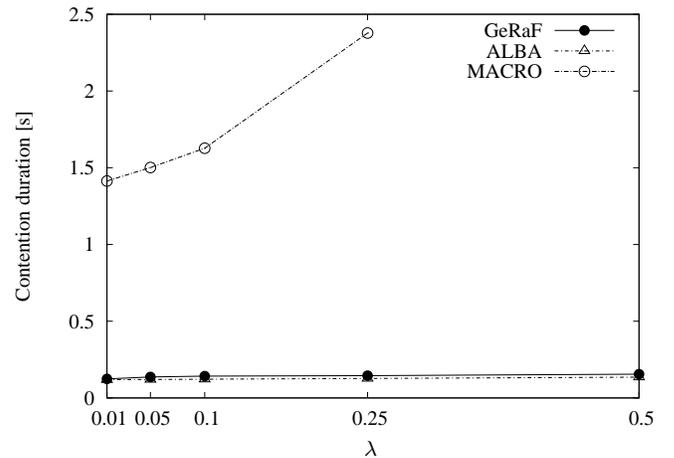


Fig. 4. Relay selection: duration [s].

scenario. We cannot show this additional material here due to lack of space, but our results indicate that, in such scenarios, ALBA improves over GeRaF in multiple ways.

First, it is able to re-route packets in case no relay toward the sink reducing (when four colors are adopted) the percentage of lost packets from 18%(57%) down to 0%(7%) in sparser scenarios, with nodal degree equal to 10(5), where this problem can severely degrade performance).

Second, it scales better than GeRaF. In particular:

- 1) As traffic load grows, ALBA leads to significantly better performance in terms of end-to-end packet latency than GeRaF, thanks to the queue load balancing strategy and to the more effective packet delivery. At $\lambda = 4.0$, GeRaF packet latency (23 s on average) is more than twice as much as ALBA's (11 s). This despite ALBA's relay selection scheme, which tends to lead to longer routes. The latter brings to a slightly higher energy consumption in ALBA over GeRaF. Such increase is however quite limited ($\leq 5\%$).
- 2) ALBA is able to sustain a higher load. At $\lambda = 6.0$, ALBA still operates correctly, while GeRaF

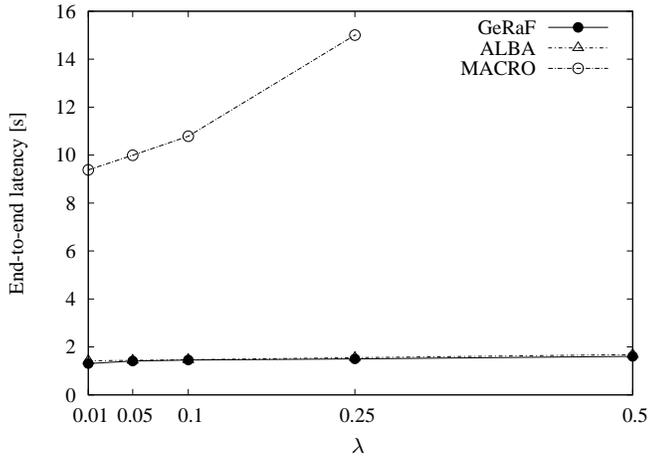


Fig. 5. End to end latency [s].

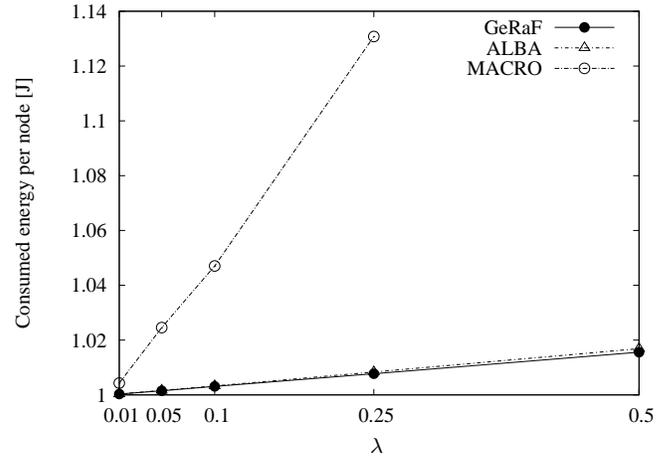


Fig. 6. Node energy consumption [J].

successfully delivers to the sink less than half of the generated packets.

As a final note, we have also performed experiments for $r = \{15, 30\}$ m. In these scenarios ALBA and MACRO have similar performance to the 20 m case for all metrics except the end-to-end latency. End-to-end latency decreases as expected for both the two protocols when the density increases. However the comparison between ALBA and MACRO end-to-end latencies at $r = \{15, 30\}$ m shows similar trends as in the 20 m case.

V. CONCLUSIONS

In this paper we propose and analyze ALBA, a novel packet forwarding protocol for ad hoc and sensor networks. ALBA follows an integrated approach that combines geographic routing and medium access control (MAC), and exploits the knowledge of node positions in order to achieve energy-efficient data forwarding. In order to reduce end-to-end latency and scale to high traffic, ALBA leverages on network density, choosing only relay candidates that are not in overload.

We have carried out extensive simulations that compare ALBA to GeRaF and to another recently proposed cross-layer approach with similar goals, MACRO. Simulation results have shown that our design achieves very good delivery and latency performance, and can greatly limit energy consumption, improving over both MACRO and GeRaF.

ACKNOWLEDGMENT

This paper describes work undertaken in the context of the e-SENSE project “Capturing Ambient Intelligence for Mobile Communications through Wireless Sensor Networks”. e-SENSE is an Integrated Project (IP) supported by the European 6th Framework Programme, contract number: 027227.

REFERENCES

- [1] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer design for wireless networks,” vol. 41, no. 10, pp. 74–80, Oct. 2003.
- [2] M. Zorzi and R. R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance,” vol. 2, no. 4, pp. 349–365, 2003.
- [3] 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, June 2004.
- [4] D. Ferrara, L. Galluccio, A. Leonardi, G. Morabito, and S. Palazzo, “MACRO: an integrated MAC/routing protocol for geographic forwarding in wireless sensor networks,” in *Proc. of IEEE INFOCOM 2005*.
- [5] S. Giordano and I. Stojmenovic, *Ad Hoc Wireless Networking*. Kluwer Academic Publisher, 2004, ch. Position based routing algorithms for ad hoc networks: A taxonomy, pp. 103–136, X. Cheng, X. Huang and D. Z. Du, editors.
- [6] I. Stojmenovic and X. Lin, “GEDIR: loop-free location based routing in wireless networks,” *Int’l. Conf. Paral. and Distr. Comp. and Net.*, 1999.
- [7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” *Proc. 3rd ACM DIAL-M*, pp. 48–55, Aug. 1999.
- [8] Y. Yu, D. Estrin, and R. Govindan, “Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks,” UCLA Comp. Sci. Dept., Tech. Rep. 010023, May 2001.
- [9] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, “Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks,” in *Proc. ACM ENSS 2004*, Baltimore, MD, 2005.
- [10] S. Lee, B. Bhattacharjee, and S. Banerjee, “Efficient geographic routing in multihop wireless networks,” in *Proc. ACM MobiHoc 2005*, Urbana-Champaign, IL, May 2005.
- [11] Z. Taha and X. Liu, “On the reliability-aware geographic routing,” in *Proc. of the 4th WTS*, 2005.
- [12] X. Jinghao, B. Peric, and B. Vojcic, “Energy-aware and link-adaptive routing metrics for ultra wideband sensor networks,” in *IEEE Workshop on Ultra Wide Band for Sensor Networks*, Rome, Italy, July 2006.
- [13] P. Casari, A. Marcucci, M. Nati, C. Petrioli, and M. Zorzi, “A detailed simulation study of geographic random forwarding (GeRaF) in wireless sensor networks,” in *Proc. IEEE MILCOM 2005*.
- [14] A. Savvides and M. B. Srivastava, *Mobile Ad-Hoc Networking*. IEEE Press and Wiley, Inc., 2004, ch. Location Discovery, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors.
- [15] P. Casari, M. Nati, C. Petrioli, and M. Zorzi, “ALBA: Geographic Forwarding and Adaptive Load Balancing in Wireless Sensor Networks,” University of Rome “La Sapienza”, Tech. Rep., June 2006.
- [16] “The network simulator – ns-2.” [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [17] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” in *Proc. Hawaii Int. Conf. on System Sciences*, Maui, Hawaii, Jan. 2000.
- [18] C. Schurgers, V. Tsatsis, S. Ganeriwal, and M. B. Srivastava, “Optimizing Sensor Networks in the Energy-Latency-Density Design Space,” vol. 1, no. 1, pp. 70–80, 2002.