CGR: Centrality-based Green Routing for Low-Power and Lossy Networks

Bruno P. Santos, Luiz F. M. Vieira, Marcos A. M. Vieira

[a] *Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil*

Abstract

High throughput and energy are two important constraints of the Low-power and Lossy Network (L2N). We propose Centrality-based Green Routing for L2Ns (CGR) as a routing protocol that considers both centrality and energy to improve network performance and decrease power consumption. CGR is a collection routing protocol that combines the best features of both protocols Collection Tree Protocol (CTP) and Centrality Tree (CT). CGR uses centrality betweenness-based to choose intermediate nodes that can perform data fusion and employ Link Quality Estimation (LQE) to find routes of high throughput and delivery rate. The suitable combination of these techniques leads the protocol to improve the literature results in delivery rate, energy consumption, and time to deliver data. There is a trade-off by routing through central nodes and their power consumption. Thus we also propose a Policy-Aware algorithm to balance energy consumption and to increase the network lifetime.

Keywords: Sensor Network, Routing, Wireless Communication, Centrality

1. Introduction

A Low-power and Lossy Network (L2N) is a network type inspired by the idea that even the smallest low-power devices should be able to run in a network. Energy is a resource extremely restricted in L2N due to small devices and their energy constraints [1]. Usually, the devices employed in these networks have no rechargeable batteries, and they are not reachable for replacement. In this context, green communications can enable longer network lifetime. By exploiting the devices features, and network topology, green-designed protocols can be an efficient way to save resources and keep alive the L2N. The design of green protocols for L2N has importance in several fields, ranging from urban and industrial low power networks to underwater networks [2, 3, 4].

A key challenge is to keep these networks alive as long as possible [2, 5]. But, some factors reduce L2N lifetime. For example, intermediate nodes usually forward data traffic of other nodes, thus eventually these relay nodes will have their energy resource quickly drained. When different source nodes send individual data flows to the sink collector, the number of transmissions will increase, and it may also promote transmission collisions or channel occupancy. Another factor, but not least, is the link conditions (quality) which can cause packet retransmissions due to bad reception and eventually packet drop by exhaustive transmission failures. Mitigating these problems can extend the L2N lifetime.

In this scenario, the routing protocols play a significant role in the resource usage. The L2N are composed of a large number of nodes with limited capabilities of computation (memory and processing), wireless communication, and energy. The protocol stack makes use of the computational resource to store and process routes, it needs to estimate the wireless link communication quality, and for all that to be done, the protocol's execution needs energy. However, many applications need to transport a large amount of data (image, audio, video monitoring, so on). Thus, protocols have the responsibility to deliver data in an energy efficient way. Routing protocols are a fundamental part of the protocol stack. Therefore, efficient ways to deliver data are critical. These applications require high data delivery and long network lifetime. Thus, routing protocols for L2N that save resources and provide green efficient routes should be addressed.

In this work, we present the Centrality-based Green Routing for L2Ns (CGR) a green data collection routing protocol. CGR combines cost-efficient L2N features in order to be, simultaneously, energy aware and provide high throughput. First, CGR uses a centrality betweenness-based approach to choose intermediate nodes, which favors in-network data fusion techniques that can potentially save energy. Centrality metrics capture the topology importance of the nodes and therefore can be used to improve routing [6]. Second, CGR makes use of Expected Transmission Count (ETX) [7] or Four Bit [8] as the Link Quality Estimation (LQE) estimator. It assists CGR to find high throughput routes, reduce packets drop, and improve energy usage.

Data collection protocols suffer from the energy hole problem, where nodes next to the sink node tend to quickly drain their energy while sending and forwarding messages. To address that issue in CGR, we implement the Policy-Aware algorithm, specially designed to mitigate the energy hole problem. Policy-Aware keeps track of spots where energy is being drained (typically at central nodes) and then changes the route to balance the power consumption, by the cost of few controls packets.

Email addresses: bruno.ps@dcc.ufmg.br (Bruno P. Santos), lfvieira@dcc.ufmg.br (Luiz F. M. Vieira), mmvieira@dcc.ufmg.br (Marcos A. M. Vieira)

We also present a survey of other approaches in the literature that considered specifically each piece of our object of study: LQE, centrality importance criterion, and related protocols.

In summary, the contributions of this paper include:

- The proposal of Centrality-based Green Routing for L2Ns (CGR), which is a distributed green algorithm to find the best routing intermediate nodes based on the Sink Betweenness centrality;
- The proposal of the Policy-Aware algorithm, which is a load balance algorithm that mitigates energy waste of the routing nodes in the network.
- Results show that CGR is a suitable algorithm in terms of delivery rate, energy consumption, and time to delivery for L2N.

CGR uses a LQE to optimize routes choice and improve throughput and energy usage, according to state-of-the-art protocols like RPL [9] and XCTP [10]. Traditional centrality routing protocols, such as CT [11] and CNS [12], use hop count to choose their routes, which can decrease the network throughput and increase energy spent. Also, the CGR architecture enables the use of the Policy-Aware algorithm to deal with the energy hole problem, while traditional centrality routing protocols do not manage this issue; indeed they increase the energy hole problem by routing only through central nodes.

Our work is organized as follows. In the next section, we present the related work divided into three parts (link quality estimators, centrality metrics, and routing protocols). Then, in Section 3, we introduce the underlines of the green routing problem, its hardness, a complexity analyses, and how we address the issues. Our proposed solution is detailed in Section 4. Section 5 brings the Policy-Aware Algorithm. Next, in Section 6, we present CGR results and compare it against state-of-the-art and traditional protocols. Finally, we conclude and present insights for future work in Section 7.

2. Related Work

This section is organized into three parts. In the first one, we discuss the main techniques for LQE highlighting characteristics of each estimator and compare them. Next, we survey centrality metrics to rank nodes according to their topological position. Finally, in the third part, we classify CGR and the related state-of-the-art routing protocols and emphasize the differences among them.

2.1. Link Quality Estimation (LQE)

Table 1 shows several estimator features. We classified the LQEs into two main categories: 1. Hardware based; 2. and Software based. The *technique* criteria highlight the methods to qualify the wireless channel used for each estimator. The LQE may have link *asymmetry support*, e.g., the link quality from node A to target B and B to A may be different. The estimator may be aware the sender or receiver *locations* to perform link quality computation - to accomplish this estimation the LQE needs to track the links by sending probe packets (*active monitoring*), explore existing traffic flowing through it (*passive monitoring*) or using hybrid monitoring combining both techniques.

Hardware LQEs methods make direct reading on radio interface. Received Signal Strength Indication (RSSI) [13], Link Quality Indication (LQI) and Signal-to-Noise Ratio (SNR) [14] are representative LQEs in this category. There are no extra computation requirements to links measurements, making LQE hardware solution fast to provide a link estimation. However, these methods are not flexible for different techniques and advanced features (e.g. Asymmetry support).

LQEs software-based methods are more adaptable for allowing advanced techniques, asymmetry support, and different location. The Required Number of Packet transmissions (RNP) [15] performs the average of the number of packets that required transmissions and retransmissions to the successful delivery packet. Link inefficiency (Li) is defined as the inverse of the packet success probability, therefore, it is also an approximation for RNP [16]. Simple Unsupervised Neuron Estimator (SUNE) [17, 18] is a bio-inspired link estimator based on the neural network paradigm that tries to predict the next probe packet reception for a link by using early probe receptions and takes into account some bias. ETX [7] and Expected Transmission Time (ETT) [19] consider asymmetric links by estimating the probability of successful reception packets/probes and its acknowledgment (ACK). When applied, these estimators help to find high throughput paths. Finally, Four Bit [8] is one of the most advanced software bases link estimator that uses information from different layers of the protocol stack to make precise link inference.

There are also analytical methods to compute link probability, node degree, and coverage in networks, such as using geometry properties [20], however, they do not capture precisely real-world link qualities nor consider asymmetric links.

2.2. Centrality metrics

Centrality metric	**Feature: Number of node participating in shortest path of other source nodes**
Degree	–
Closeness	–
Eccentricity	–
Page Rank	–
Stress	✓
Betweenness	✓
Sink Betweenness	✓

Table 2: Centrality metrics comparison

In graph theory and network analysis, centrality refers to an indicator, which ranks vertexes according to importance. Several criteria of importance can be found in the literature [21], e.g., a relevant vertex can be that associated with many route participations in the network. At the substantive level, impor-

	Software Based					Hardware Based
Estimator	RNP	Li	SUNE	ETX/ETT	Four Bit	RSSI, LQI, SNR
Technique	Avg.	Probability	Weighted sum	Avg.	Filtering	Radio reading Avg.
Asymmetry Support	–	–	✓	✓	✓	–
Location Monitoring	Sender	Receiver	Receiver	Receiver	Receiver	Receiver
Monitoring	P	P	A and P	A and P	A and P	P

A – Active
P – Passive

Table 1: Comparison among wireless link estimators

tance can be an involvement in/or contribution to the cohesiveness of the network.

In this work, node importance is defined as the number of participation of the node in the shortest paths of other source nodes. Centrality metrics can be applied to capture this importance criterion [22]. Next, we present a brief survey of centrality metrics and classifying them.

Table 2 compares different centrality metrics. Degree, closeness, eccentricity, and Page Rank™do not capture our required feature. Stress and Betweenness catch the demanded feature, but Sink Betweenness Centrality (SBC) stand out because it has been defined on Wireless Sensor Network (WSN) context by computing Betweenness for only targeted nodes [11].

2.3. Routing protocols

We present in Table 3 the main related centrality-based protocols. We classify them according to the strategies adopted to build the routing structure and the awareness of L2N constraints. Table 3 shows that CGR is, to the best of our knowledge, the only protocol designed with following features: 1. centrality based routing to rank nodes according to topological importance; 2. it employs LQE to provide high-throughput routes for multi-hop wireless. 3. it provides a distributed green algorithm with same complexity of the state-of-the-art.

Shortest Path Betweenness-Centrality (SPBC) and Traffic Load Centrality (TLC) assume Betweenness-Centrality and only shortest paths are used to transfer data, but TLC uses a different mechanism to choose the routes. Flow Betweenness-Centrality (FBC) equally considers routes of all lengths and applies the maximal flow as role to routing. In [22] is presented Routing Betweenness Centrality (RBC), this technique generalizes SPBC, TLC, and FBC. However, these approaches were not designed for L2N and do not consider loss rate on links.

In [23], the authors present LTRBSA an algorithm to find virtual data aggregation trees for WSN. CGR differs from LTRBSA by using SBC measure and by take into account the link quality. Classic protocols which do not use centrality are Shortest Paths Tree (SPT) and Center at Nearest Source (CNS) [12]. SPT picks the shortest path from the source node to the sink. In CNS, nodes near the sink act as an aggregator, thus source nodes send data to the aggregator that in turn sends an aggregated message to sink. Data-Aggregation Aware Routing Protocol (DAARP) and Information Fusion-based Role Assignment (InFRA) are presented in [24], these approaches are approximations to the Steiner tree [27]. The nodes are hierarchically organized into clusters that send data to sink, which favors event-radius model [12]. eXtend collection tree protocol (XCTP) [10] extends Collection Tree Protocol (CTP) [25] to create the reverse path between the root node and sensor nodes. Matrix [26] is a routing protocol that utilizes the existing tree structure of the network to enable reliable and efficient any-to-any data traffic and uses hierarchical IPv6 address assignment to optimize routing table size. IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [9] is a state-of-the-art protocol for L2N supporting traffic flows point-to-point, point-to-multipoint, and multipoint-to-point. Our work differs from these techniques because it uses centrality to route data and uses wireless link quality estimators.

An important issue in L2N is the quality of wireless links. A bulk of LQE were proposed as depicted in [14]. We designed CGR to be suitable for wide range of link estimators (like XCTP, Matrix, and RPL), for instance, we can use ETX or Four Bit as LQE. But state-of-the-art protocols does not regard the node topological importance in its routing schemes like CGR.

In summary, there are some approaches related to CGR. Among those that use centrality, we highlight the Centrality Tree (CT) which uses SBC as importance rank. Among those that do not use centrality, we highlight RPL as the state-of-the-art protocol for L2N. Thus, we propose CGR with the best features of literature protocols to provide routes of high throughput, being green energy efficient, and favoring data fusion techniques.

3. Problem definition and its hardness

In this section, we first describe the network model as a connected weighted directional graph in Section 3.1. Then, we present the green routing tree-based problem. Next, we highlight its hardness by given an intuition of its the NP-Completeness in Section 3.2. Also, in Section 3.2.1, we describe how our proposed work treats the problem.

3.1. Network model

In this work, we model the network as a connected weighted directional graph $G(V_G, E_G, w_G)$, where $v \in V_G$ represents the nodes, $(u, v) \in E_G$ is a directed link (connection) from node u

Protocol	Centrality support	Distributed algorithm	LQE
SBPC[22]	BC	–	–
TPC[22]	BC	–	–
FBC[22]	BC	–	–
RBC[22]	BC	–	–
CT[11]	SBC	✓	–
CGR	**SBC**	✓	✓
LTRA[23]	–	✓	–
SPT[12]	–	✓	✓
CNS[12]	–	✓	✓
InFRA[24]	–	✓	✓
DAARP[24]	–	✓	✓
CTP[25]	–	✓	✓
XCTP[10]	–	✓	✓
Matrix[26]	–	✓	✓
RPL[9]	–	✓	✓

*BC – Betwenness Centrality.
*SBC – Sink BC.

Table 3: Routing Protocols Comparison

to v , and $w_G(E_G)$ is a non-negative value in $[0, 1]$ representing a probability of link loss[1].

3.2. *Green Routing Tree-based Problem and its Hardness*

Given the network model presented in the previous section, our problem is to find a green routing tree that favors data fusion with the following requirements:

Req. 1) **Data-centric tree:** the protocol should provide the least number of intermediate nodes (routers), thus favoring data-centric routing and data fusion algorithms [12].

Req. 2) **Reliability:** the routing protocol should deliver as much as possible data packets when there is a feasible route between the participants of the communication.

Req. 3) **Robustness/Resilience:** the routing protocol should operate in different topologies, loads, amount of sensor nodes and in the presence of failures.

Req. 4) **Green efficiency:** the protocol must deliver packets with the least amount of transmissions, saves energy and keep the least amount of possible states.

All features mentioned above are critical to providing a green-aware routing. In the following, we highlight the requisite (1) given its hardness. Several efforts were made to find optimal data fusion tree for WSN and L2N [28, 22]. Those work reveal that traditional address-centric routing schemes are not suitable for energy saving against data-centric routing schemes. It is well-known that the optimal data-centric routing tree, which favors data fusion, is a NP-Complete problem, resulting from NP-completeness of the minimum Steiner Tree problem [12].

[1]Note that model allow asymmetric links.

Let $G(V, E)$ be a graph that represents a L2N, where V is a set of nodes in the network, $s \in V$ is a special node called sink, and E is a set of edges that represent connections between two nodes. If the number of transmissions between two directly connected nodes is one, then the optimal data fusion tree is the reverse of the *multicast* tree. In other words, all source nodes send data to the same receiver that will perform some data fusion technique. The *multicast tree* with the minimal number of edges is a NP-Complete problem, transformed into the problem of *minimum Steiner Tree* [12].

3.2.1. *Addressing the problem*

In the following, we describe for each requisite presented in the previous section how we address it.

The *minimum Steiner Tree* problem in graphs is well-known to be NP-Complete, thus is difficult to find an exact algorithm to solve it in polynomial time. Therefore, approximation algorithms for *minimum Steiner Tree* that have polynomial running time have been proposed in the literature. The first works address the problem by computing a minimum spanning tree instead of a minimum-cost Steiner tree or are simple greedy algorithm [29, 30]. More recently, genetic algorithms have been used as an approximation solution for the problem [31, 32].

In the context of L2N and networks, in the last years, there have been efforts in designing optimal solutions using centrality measure [22, 11, 33] or bio-inspired algorithms [34, 35]. In this work, we adopted the Sink Betweenness Centrality measure to provide an approximation to the minimum Steiner Tree problem, this is due to its low cost in terms of complexity and implementation overhead. These points are critical given the L2N devices constraints.

The topological position of the nodes is critical to approximate the Steiner tree, so we rank nodes according to its position. Several approaches can be used for this ranking as discussed in Section 2.2. We choose SBC to rank nodes according to the number of participation of node in the paths from all nodes to sink. The SBC of a node t is defined as follows. Consider a graph $G = (V, E)$ that represents the L2N, where V is a set of nodes and E is a set of links. A special node $s \in V$ is the sink, then $SBC_t = \sum_{i \in \psi_t} \frac{\sigma_{ts}}{\sigma_{is}}$, where σ_{ts} means the number of shortest paths from t to s, σ_{is} is the number of shortest paths from i to s, $\psi_t = \{i \in V | t \in SP_{i \to s}\}$ is the set of all shortest-paths from a node i to s ($SP_{i \to s}$) containing t, as an intermediate node in at least one of their shortest paths.

To be reliable (requisite 2), it is necessary to find high-quality routes in the network in order to maintain as high as possible the successfully delivery rate. This requisite is highly affected by the route metric employed, thus an efficient LQE should be used (Section 2.1 overviews LQEs metrics). To be robust and resilient (requisite 3), our protocol does not assume anything about network infrastructure, nodes capabilities, and scale of the network. To act green efficiently (requisite 4), we combine a LQE metric and a data-centric routing centrality-based to provide a routing tree that favors data fusion techniques in order to reduce the number of transmissions and saves energy.

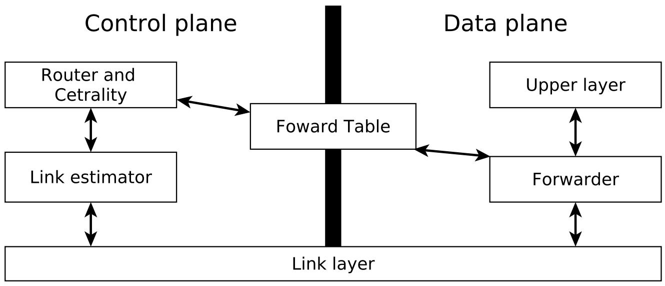


Figure 1: CGR architecture.

4. Centrality-based Green Routing

In this section, we describe the Centrality-based Green Routing for L2Ns, a routing protocol that considers both centrality and energy to improve the network performance and decrease energy consumption. First, we present the CGR architecture, its modules and relationship in Section 4.1. Next, in Section 4.2, we present the CGR Algorithm and implementation details. CGR complexity analysis regarding control packets to compute the centrality tree is presented in Section 4.3. Finally, we discuss some aspects of CGR algorithm in Section 4.4.

4.1. CGR Architecture

Here, we describe CGR architecture. To accomplish the task of forwarding packets using important nodes, we propose CGR architecture that has routing rules to consider SBC rank. The protocol rules were implemented at the control and data planes. The data plane queries the forwarding table. The control plane is responsible for creating, update and delete entries in this table.

In Figure 1, we show the relationships between modules. The Router and Centrality modules are responsible for filling the Forward table. That table indicates what is the next hop for the data packet to be transmitted. Link Estimator module estimates the quality of the links to the neighboring nodes, reporting for Router and Centrality module to construct better routes. The link quality is estimated using beacons and data packets by using, for example, Four Bit estimator. The Forward module queries the Forward table and determines any router inconsistencies to inform the Router module. It also keeps a packet queue for transmission and checks for duplicate packets. The Link Layer module contains the features used in radio communication. Finally, the Upper Layer module is the interface provided to implement components that utilize CGR.

4.2. CGR Algorithm

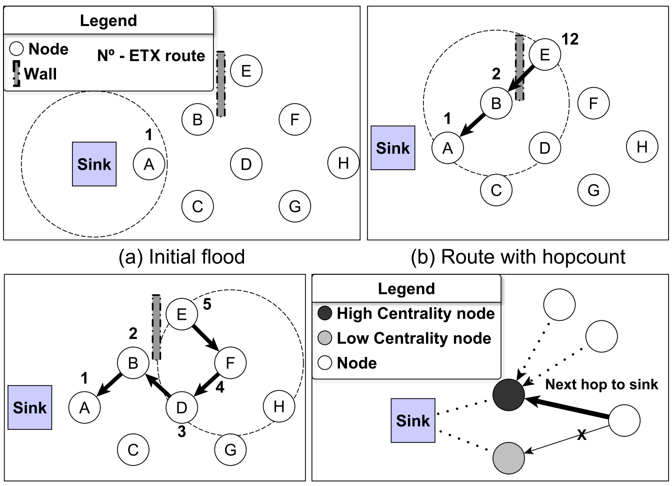
Algorithm 1: Find the best paths.

$Sink \leftarrow 1$ `// Base station.`
$Paths_t \leftarrow 1$
$SBC_t \leftarrow 0$
$Hops_t \leftarrow \infty$
$Nhop_t \leftarrow this$ `// Next Hop.`
$RQ_t \leftarrow \infty$ `// Current route quality.`
$Max_{SBC} \leftarrow -\infty$ `// Route Quality.`
if $\underline{Sink == this}$ **then**
    `// Initializes flood.`
    $HP_{id} \leftarrow this$
    $HP_{hops} \leftarrow Hops_t$
    $HP_{paths} \leftarrow Paths_t$
    $HP_{RQ\_Path} \leftarrow 0$
    Broadcast DP
else if Receives packet DP **then**
    if $\underline{HP_{RQ\_Paths} + RQ^{\dagger}_{t \to HP_{id}} < RQ_t}$ **then**
        `// Set new best route`
        $Hops_t \leftarrow HP_{Hops} + 1$
        $Paths_t \leftarrow HP_{Paths}$
        $Nhops_t \leftarrow HP_{id}$
        $RQ_t \leftarrow HP_{RQ\_Path} + RQ_{t \to HP_{id}}$
        $HP_{RQ\_Path} \leftarrow HP_{RQ\_Paths} + RQ_t$
        $HP_{Hops} \leftarrow Hops_t$
        $HP_{Paths} \leftarrow Paths_t$
    else if $\underline{HP_{RQ\_Path} = RQ_t}$ **then**
        `// Detect two or more best paths`
        $Paths_t \leftarrow Paths_t + 1$
        $HP_{Paths} \leftarrow Paths_t$
    end
    Schedule Broadcast DP.
    Schedule Broadcast RP.
end
†Quality for direct link between node t and RP_{ID} (e.g. ETX measurement)

We describe the operation of CGR in this section. We show how to bind SBC and LQE metric (ETX-like), thus CGR can extract the best features of both techniques in order to connect the source nodes to the sink. CGR has two main phases:

Algorithm 2: Choose central nodes based on SBC.

[1] **if** Receives RP **then**
[2] **if** $RP_{RQ\_Path} + RQ_{t \to RP_{id}} > RQ_t$ **then** from a descendant
[3] `// for each RP received (nodes in` ψ_t`)`
[4] `// Update SBC rank`
[5] $SBC_t \leftarrow SBC_t + \frac{Paths_t}{RP_{Paths}}$
[6] $RP_{SBC} \leftarrow SBC_t$
[7] `// update content packet and forward`
[8] Forward_RP(RP)
[9] **else if** $RP_{RQ\_Path} + RQ^{\dagger}_{t \to RP_{id}} \leq RQ_t$ **then**
[10] **if** $RP_{SBC} > MAX_{SBC}$ **then**
[11] $MAX_{SBC} \leftarrow RP_{SBC}$
[12] $Nhop_t \leftarrow RP_{id}$
[13] **end**
[14] **end**
[15] **end**
[16] **procedure** Foward_RP(Packet RP)
[17] $RP_{NH} \leftarrow Nhop_t$
[18] $RP_{RQ} \leftarrow RQ_t$
[19] *Broadcast RP*
[20] **end**
[21] †Quality for direct link between node t and RP_{ID} (e.g. ETX measurement)



(a) Initial flood (b) Route with hopcount

(c) High throughput route (d) Update Next Hop by Centrality Rank

Figure 2: Illustration of CGR algorithm in operation.

1. Sink floods the network to find the number of high-quality paths from nodes to sink using ETX metric. 2. Nodes respond their information (number of paths), i.e., the denominator of the equation, which allows computing SBC.

CGR uses two control packets to create the routing tree. The first one is Hello Packet (HP) that has the following fields: node ID; hops; paths; route quality path. The second the control packet is the Reply Packet (RP) which has the same fields of HP plus an SBC field to compute the centrality. In Algorithm 1 and Fig. 2(a)(b)(c) we illustrate the first flood. In Fig. 2, we assume that all links are perfect, except by the link (B, E), which present poor delivery rate (i.e., 10% of delivery rate) due to the wall between the nodes. In this scenario, traditional protocols that use hop count favor the route depicted in Fig. 2(b), this will result in a larger number of retransmissions, overhearing effect [36] and high energy consumption to deliver messages. Our proposal mitigates these issues by using ETX to find a high-quality route as shown in Fig. 2(c). The rules to perform our approach are shown in Algorithm 1 lines 15 to 29.

CGR second phase starts when source nodes send their information (number of paths to sink) in the RP (see algorithm 1 line 31). The delay to fire HP and RP are proportional and inversely proportional to sink distance, then nodes close to sink node have a short delay to fire HP and long delay to RP, leaf nodes presents the opposite. When a node receives a RP, each router node has the information needed to compute SBC (σ_{ts} and σ_{is} from equation in Section 3.2.1). The rules that receive and compute the SBC are shown in Algorithm 2 in lines 1 to 4. After the computation and/or update the SBC rank, the nodes update the RP with their information and broadcast the packet (line 8) aiming to reach immediate descendants neighbors. Based on the centrality rank, the descendant nodes can decide which parent has more topological importance to forward/fuse their packets (see Fig. 2(d)). These decision rules are shown from line 9 to 14 of the algorithm 2.

4.3. CGR complexity analysis

In this subsection, we provide a brief CGR complexity analysis in terms of the number of control packets necessary to compute the centrality measure.

In order to compute the SBC, the CGR protocol needs to establish the routing tree and de facto centrality computation, which are the two main phases of CGR algorithm (details in Section 4.2). The cost to build the tree is n HP packets sent by each node to form the routing tree. In the reply phase, all nodes, except the sink, will send a RP, adding more $n - 1$ packets of control overhead. The total cost of the CGR is $2n - 1$ control packets. Therefore, CGR complexity is $O(N)$, where N is the number of nodes in the network. In the literature, protocols optimized for data fusion also present similar complexity [37].

4.4. CGR Discussion

CTP uses Trickle algorithm [38] and 4-Bit to calculate link quality efficiently with no network overhead [25]. Like CTP, CGR uses the same methods to provide reliability, robustness, and efficiency. Energy consumption of central nodes is the other issue that CGR should deal suitably, energy hole mitigating techniques [39, 40, 41] can be applied to balance central nodes energy expenditure. In the next section, we propose an energy balance algorithm that can be easily attached to CGR protocol.

CGR was designed for L2N with low topology dynamic. Thus, it is a challenge to accomplish the rank computation with highly dynamic moving nodes. Therefore, we note that CGR needs almost two floods to compute SBC, which may be costly for frequently abrupt changes in the topology.

5. Policy-Aware

Algorithm 3: Policy-Aware algorithm.

```
[1]  Procedure Init()
[2]      Best_RQ ← +∞
[3]      HighCentraltiy_t ← −∞
[4]      if Policy = Threshold then
[5]          // Threshold limit is reached
[6]          MP_command ← Alert
[7]          Broadcast MP
[8]          UpdateThreshold()
[9]          return
[10]     else
[11]         Increase(Policy)
[12]         return
[13]     end
[14] Procedure InterceptAlert(MP)
[15]     if MP_hops ≥ Hops_t then
[16]         Broadcast NRP
[17]         return
[18]     else if MP_ID = NextHop_t then
[19]         Best_RQ ← +∞
[20]         HighCentraltiy_t ← −∞
[21]         return
[22]     end
[23] Procedure InterceptNRP(NRP)
[24]     if NRP_alertID = NextHop_t ∧ NRP_RQ ≤ Best_RQ
         then
[25]         if
             NRP_RQ = Best_RQ ∧ NRP_SBC > HighCentrality_t
             then
[26]             HighCentrality_t ← NRP_SBC
[27]             NextHop_t ← NRP_senderID
[28]             Hops_t ← NRP_H
[29]         else
[30]             Best_RQ ← NRP_RQ
[31]             NextHop_t ← NRP_senderID
[32]             Hops_t ← NRP_H
[33]             RQ_t ← NRP_RQ + RQ_{t→NRP_senderID}
[34]         end
[35]         // Update all field of New Route
                Packet (NRP)
[36]         Broadcast MNR
[37]         return
[38]     end
```

In this section, we introduce the Policy-Aware algorithm, an approach to deal with energy holes in the L2N. Energy hole problem is defined as excessive energy expenditure in parts of a network [40]. In L2N, energy hole can cause disconnected components. The Policy-Aware approach comes as an alternative to mitigate this problem by driving out existing flows of the central nodes to non-central nodes.

The Policy-Aware algorithm combines three features in its scheme. First, it uses a metric (a LQE) to select high-throughput routes, the second one is a node constraint (residual energy), and finally the node centrality. The Policy-Aware approach differs from literature ones (e.g., RPL Objective Functions [42]) by taking into account the centrality feature.

The Policy-Aware approach needs two new packet types. The first one is Management Packet (MP) that has the following fields: node ID, hops, and command (management type). The second is NRP that has the fields: sender ID (last node to forward the packet), alert ID (the node that sends alert), hops, RQ (route quality), Centrality (SBC rank).

The Algorithm 3 shows the main rules of the Policy-Aware. Initially, we define two variables to keep track the best route quality ($Best_{RQ}$) and more central node seen since the Policy-Aware starts (Lines 2 and 3). The algorithm checks if a policy counter reaches a given threshold. This policy can be battery level, forwarding traffic or whatever metric the user wants. For example, we can define the policy as the number of transmissions and the threshold N packets. If the threshold is reached then the node broadcasts a MP with command *alert* (lines 4 – 9). If the threshold is not reached, then the algorithm increases the policy counter.

One node by receiving a MP with command alert execute the procedure in line 14. The node checks if the alert comes from a descendant or other node in the same level in the tree (line 15). If the rule is true then the node sends a NRP packet promoting themselves as a router. Otherwise, if the rule is false, the node checks if the sender of MP is his parent node (line 18), then the node should get ready to keep tracking alternative routes (lines 18 to 21). By intercepting a NRP the node should suitably update its next hop in the path to the sink. If the NRP alert ID is the parent of the node and the route is the best seen (line 23), then the node based on tracked variables chooses the best alternative route through the more central node (see lines 25 to 33). Finally, the node tries to help their parent node by sending NRP (line 36) for other nodes that received no options for new routes.

5.1. Policy-Aware Discussion

In the evaluation section, we show that policy-aware can lead to significant energy balance consumption, increasing the central node lifetime, and consequently the network life. On the other side, the Policy-Aware algorithm introduces trade-offs since it adds new control packets to the network and it changes routes through the central nodes. We performed several experiments and noted that Policy-Aware is not suitable for a network with high asymmetric links, which can cause low throughput in link changes.

6. Evaluation

We analyze CGR and compared it with two protocols: CT, SPT, and RPL. From the protocols presented in Table 3, RBC and the ones derived from it were not designed for L2N, therefore they can not be compared with CGR. InFRA and DAARP belong to a different class of protocols that approximate Steiner

Parameter	Value
Sink	1 corner
Number of sensor nodes	1024
Radio range	100 m
Link Loss (value/edge)	$0 < p \leq 1$
Nodes sending data	45 %
Data packets event (packets/node)	30 packets/node
Sending packets duration	300 s
Default retries dispatch packet	10

Table 4: Simulation parameters

Tree that do not classify the nodes by centrality. CNS is a classical protocol and its results are lower than CT. Therefore, we compared CGR against CT and SPT. We also compare CGR with RPL, the state-of-the-art protocol for L2N, to easily identify CGR contribution.We also show the CGR evaluation operating with the Policy-Aware Algorithm.

The protocol evaluation goals aim to demonstrate the CGR performance in terms of green routing requisites (See Section 3): *data-centric tree*, *reliability*, *robustness/resilience*, and *green efficiency.* We use well-known metrics to evaluate L2N protocol just as it was done in the RPL performance evaluation [43]. We use the following metrics:

(i) **Steiner Number** used to analyze *minimum Steiner Tree* approximation, where small Steiner number means better data-centric routing tree;

(ii) **Delivery Rate** aims to evaluate the protocol reliability. High delivery rate means high reliability;

(iii) **Energy Consumption** also concerned to the green efficiency, lower energy consumption is better;

(iv) **Transmissions Per data Reported (TPR)** aims to evaluate the number of transmissions which is required to deliver data packets, small TPR means better green efficiency;

(v) **Latency** is the time interval between the source nodes fires a data packet and the sink receives it, low latency is better.

6.1. Simulation

We implemented CGR, CT, SPT, and RPL-Like in the Sinalgo simulator [44]. CT implementation follows its default specification presented in [11]. The SPT protocol is an implementation of the shortest path (Dijkstra's algorithm). CT and SPT are our baselines. The RPL-Like implementation follows the RFC 6550 [9] with the following features: non-store mode, DAO-ACKs packets control, ETX link metric as Objective Function.

We considered 33 different topologies, in each one we ran 33 simulations, totaling 1089 runs. In the results, the curves represent average and error bar with the confidence interval of 95%. Table 4 shows the defaults parameters utilized if any parameter varied we notify accordingly. Note, packets are illegible if errors (Link Loss) occur when decoding a packet. Also, nodes can not receive more than one packet in the time unit. Data packets can be lost by link failure during a transmission.

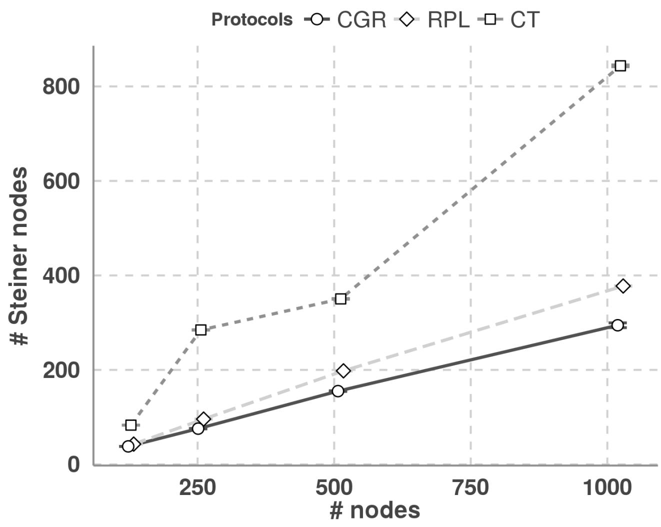


Figure 3: Number of Steiner nodes for different number of nodes.

6.2. Simulation results

The first result is about the favoring data fusion and data-centric routing. We analyze the number of Steiner Nodes in the network. Low Steiner number promote common points in the routing tree where data fusion techniques can be performed. Figure 3 shows the number of Steiner. CGR presents a lower number of Steiner Nodes even for a different number of nodes. CGR is followed by RPL. CT uses the same SBC measure as CGR, but the CT routing metric promotes a different routing tree than CGR ones, which impact directly the Stainer Number.

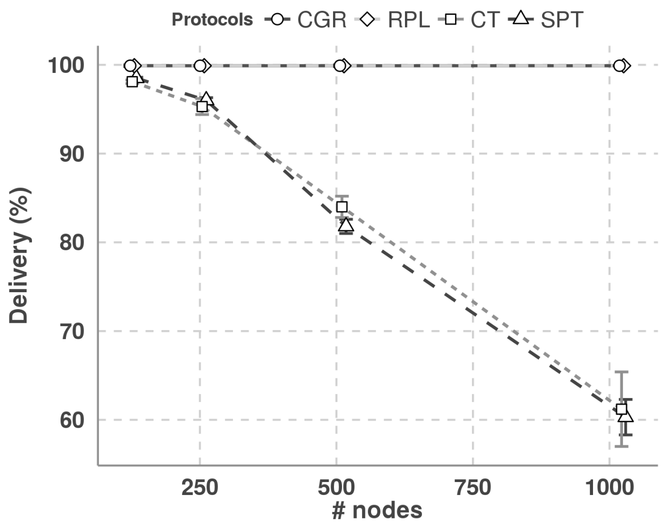


Figure 4: The delivery rate for different number of nodes.

The second result is concern about the protocol reliability. For this, it is allowed data packets be dropped if a maximum of sending/forwarding attempt is reached (see Table 4). The delivery rate for different amounts of nodes is shown in Figure 4. CGR an RPL always deliver more than 99 % of the data packets, while CT and SPT present low delivery rate when the

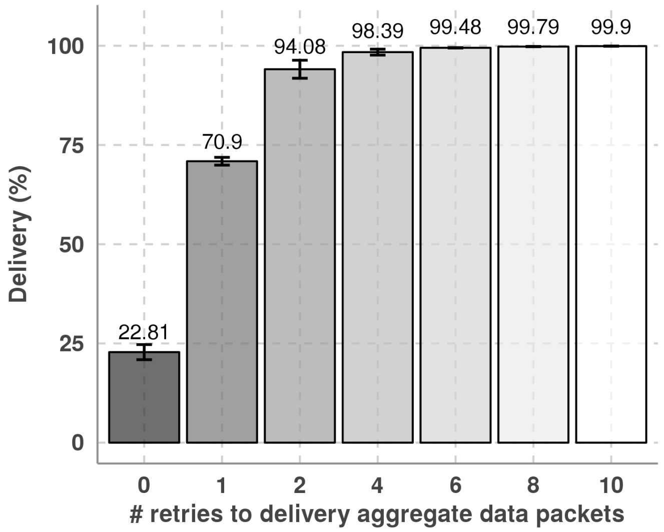


Figure 5: Data delivery rate of CGR + simple periodic fusion function.

number of nodes grows. Figure 5 shows the results regarding the percentage of data delivery of CGR and a simple periodic data fusion function. Such data fusion function receives data packets during a preset *t* period and then sends only one aggregated packet. This function is performed by the nodes every time they receive a packet. Observe that if the nodes do not do any effort to deliver an aggregated packet (0 retries), then the data delivery rate is dramatically low around 22 %. However, if only 1 retry is allowed, then the delivery rate grows significantly up to 70 %. With 10 retries the delivery is close to 100 %. In the previous Figure 4, The SPT protocol even with 10 retries, presents only 62 % of delivery is achieved.

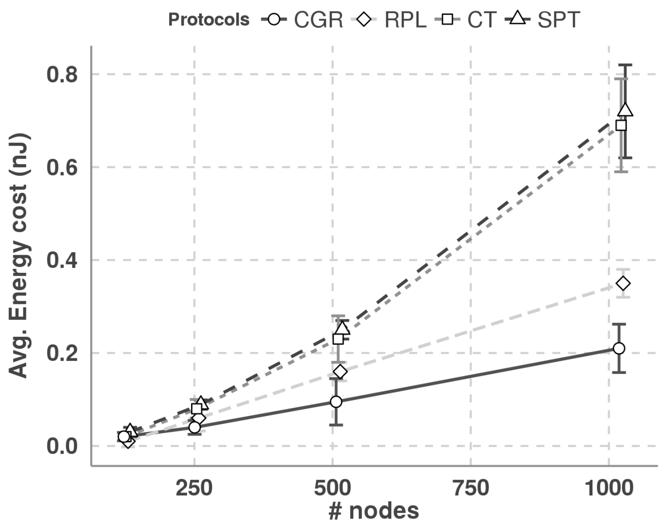


Figure 6: Global energy consumption for different number of nodes.

We evaluate the network energy consumption over a simulation where data packets are guaranteed to be delivered to the sink node[2]. The Figure 6 shows the energy consumption varying the number of nodes. It is possible to note that CT and SPT are energy-greedy, while CGR and RPL are more green energy efficient, but CGR presents the lower level of power consumption for 256, 512, and 1024 nodes. RPL provides a fast optimal tree set up at the cost of relatively high overhead as shown below and reported in [43].

The last two results (delivery rate and energy consumption) can be better explained by route quality and the data control overhead. The Figure 7 highlight insights about the protocols energy consumption. In Figure 7(a) and Figure 7(b) present the route quality measured by the ETX Path[3] for a different number of nodes and radio communication range respectively. On the one hand, When ETX path is high then the delivery rate drops, retransmissions happen, and energy consumption increases. On the other hand, when we fix 1024 nodes and the radio range increases, the route quality drops because the distance in hops of the nodes to sink decreases. In both cases CGR and RPL present lowest ETX Path, which justify they high delivery rate (Figure 4), SPT and CT present similar higher ETX Path. The Figure 7(b) shows an expected protocol behavior, SPT was suppressed because its curve is very similar to CT and for better visualization.

The Figure 7(c) shows the control packet overhead for each protocol. The theoretical SPT overhead cost depends on the network size, this is, SPT needs only a flooding to build its routing tree, this cost was reflected on our experiments. CGR and CT need at least two floods to compute the centrality measure (See Section 4.3). RPL presents the highest control packets overhead to build its DODAG structure.

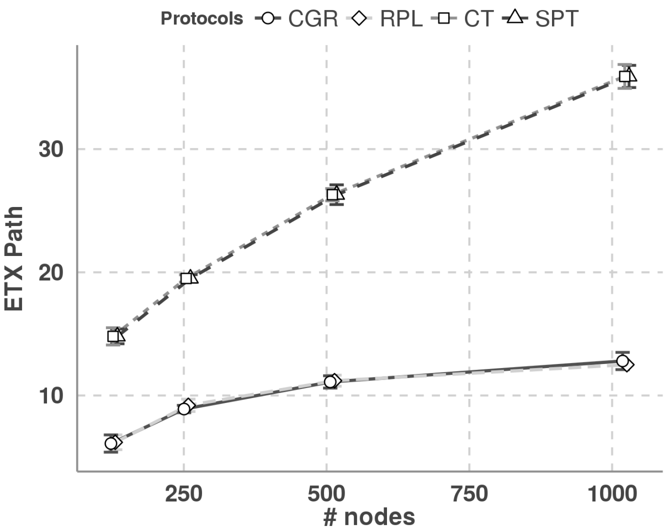
To evaluate the CGR green efficiency, we evaluate the average number of transmissions required to deliver data packets to the sink node (Transmissions Per data Reported (TPR)), where low TPR implies in better green efficiency. The Figure 8 shows, for each protocol, how the average TPR for a different number of nodes and the radio ranges. In Figure 8(a), CGR presents, in most of the scenarios, the lowest average TPR among the protocols, RPL also present low TPR when compared with SPT and CT. RPL and CGR present similar TPR when the number of nodes is 128, but if the number of nodes increases the control overhead boost the RPL's average TPR. The protocols SPT and CT show a high average of transmissions per data packets, this is explained by they routing metric, which selects routes with high ETX path. When the number of nodes in the network is 1024 CT decreases its TPR, because with more nodes CT has more options to centralize the routes, theses changes can exchange bad link to good ones.

In Figure 8(b), we keep fixed all simulation parameters except by the radio range, then we evaluate the TPR. Again CGR followed by RPL present the lowest values of TPR, while SPT and CT show high TPR. When the radio communication range increase the TPR decrease, this is a common behavior of all protocols.

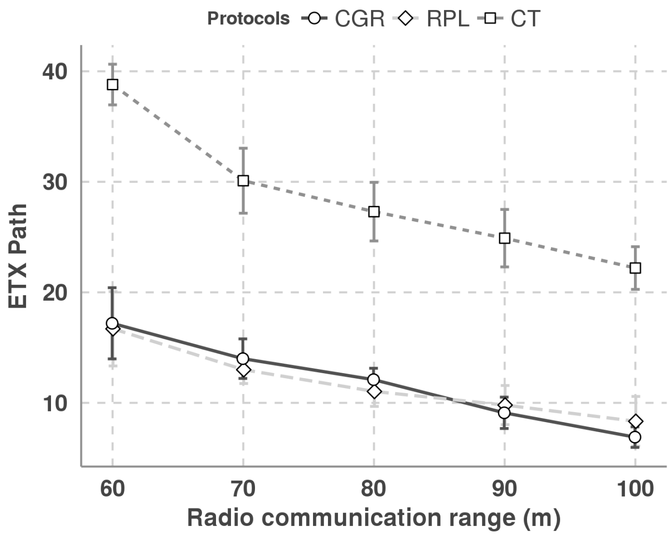
We also analyzed latency in our experiments. Figure 9 shows the global latency for the L2N. This result shows that CGR/RPL

[2]This is done by setting the default retries dispatch packet to infinity.

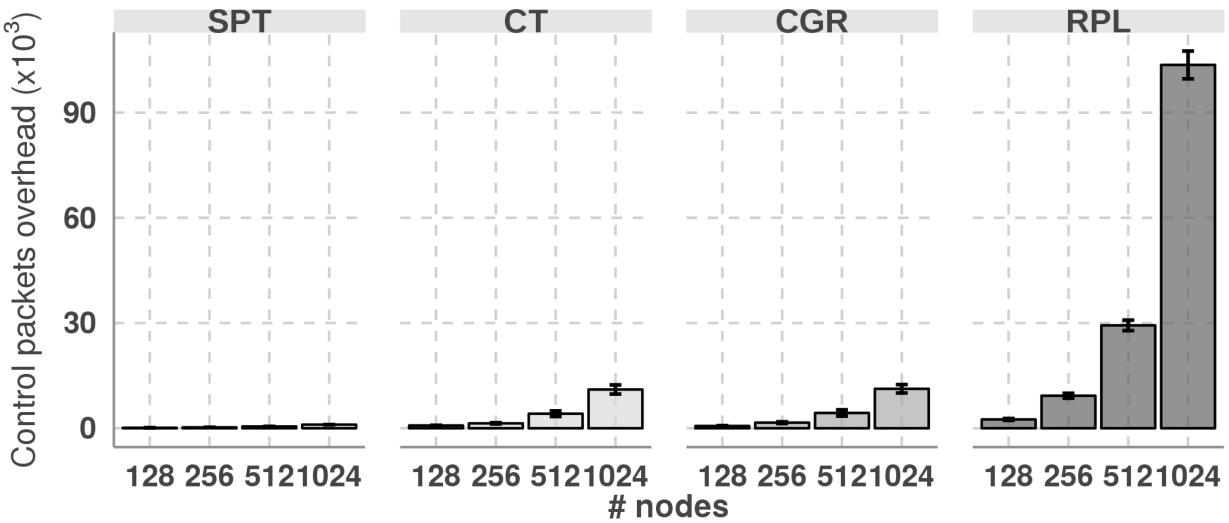
[3]ETX path is defined as the sum of the ETX of all nodes in a route.



(a)



(b)



(c)

Figure 7: In (a) ETX Path (route quality) for different number of nodes. (b) ETX Path varying the radio range. (c) Control packet overhead for each protocol.

and ETX make a good combination by finding high throughput routes to deliver data packets. In all scenarios, CGR and RPL present average global latency under 0.5 s, while the CT and SPT protocols showed, in some cases, a latency of 2 s. High latency of the protocols can be explained by the hop count metric that ignores link quality and causes latency, retransmissions overhead, energy consumption and packets drop.

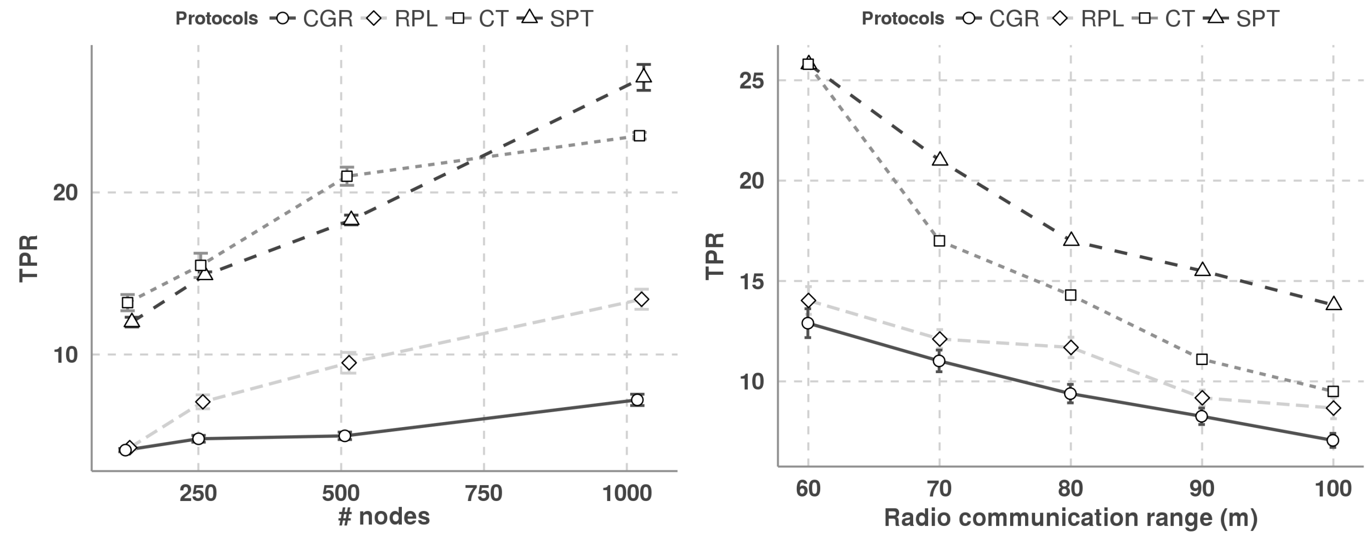
Low latency also is required when in-network data fusion algorithms are employed. We implement three different well-known in-network data fusion algorithms [37]:

1. *Simple:* the nodes fuse messages received during a preset time t. Then, it forwards an aggregated message.

2. *Period per hop (PerHop):* the PerHop behavior is the same of Simple. However, if the node received one message from its children (concerning the tree rooted at the sink node), then an aggregate message is forwarded to next hop toward the sink node.

3. *Period per hop adjusted (PerHopAd):* in this approach the nodes also wait a time t, however, this time depends on the distance (hops or time) from the sink. The intuition is straightforward. t is inversely proportional to its distance from the sink; thus t will be shorter if the node is far from the sink and larger if the node is close to the sink.

Data fusion algorithms are very sensitive to several parameters. For instance, how much nodes are sending data, how long a node waits to receive data from children before forwarding a ”fused packet”, and how space is the period of a node report data. Thus, we set up for all nodes send data packets (except



(a) (b)

Figure 8: Transmissions Per data Reported varying number of nodes and radio range.

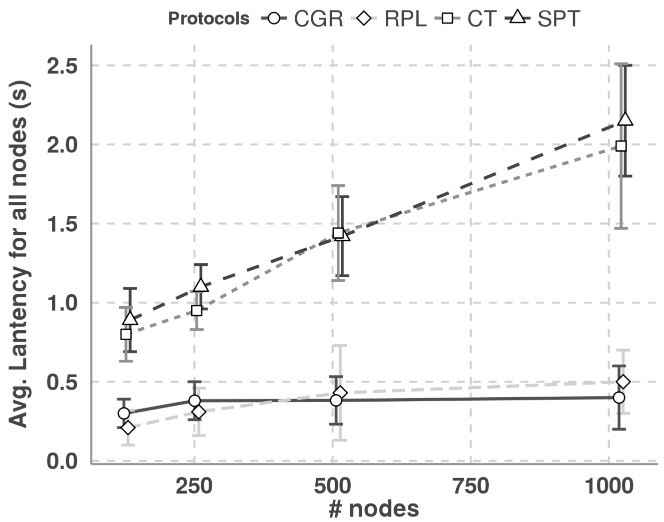


Figure 9: Average latency for different number of nodes.

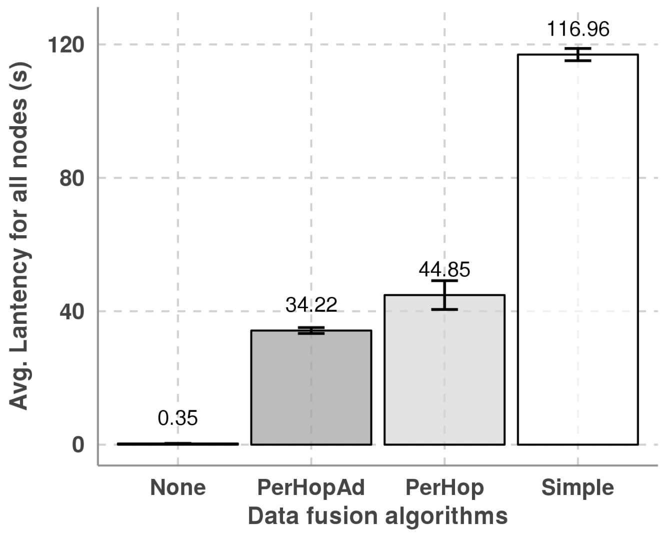


Figure 10: CGR average latency for different data fusion algorithms.

the sink node), the default t is 15 s. Also, the nodes start to send data exactly after the tree construction.

Figure 10 shows the CGR average latency for different data fusion techniques. *None* presents the lowest latency because no data fusion occurs and the data are routed individually. When there is in-network data fusion, *Simple* presents the highest latency because for every hop the nodes wait t seconds. *PerHop* decreases the latency of *Simple*, but the overhead is that every node knows its children. *PerHopAd* improves the result of *PerHop* by using a "aggregation wave" in its approach.

Finally, we show CGR protocol operating with Policy-Aware, our green algorithm that enables balancing energy consumption. Figure 11 shows CGR protocol working with and without the Policy-Aware algorithm. In the y-axis, it shows energy consumption and in the x-axis shows the node centrality. First, observe that without Policy-Aware the energy expenditure behaves as expected, that is, highly central nodes have heavy traffic demand to route and consequently consumed more energy. CGR with Policy-Aware balance the energy expenditure, that is, non-central nodes take parts of the traffic demand, which before was being routed by the more central node. Therefore we can see a decreasing the energy consumption of higher centrality nodes. Therefore, CGR with Policy-Aware achieves a more balanced energy consumption, increasing the network lifetime, and mitigating the energy hole problem in a distributed fashion.

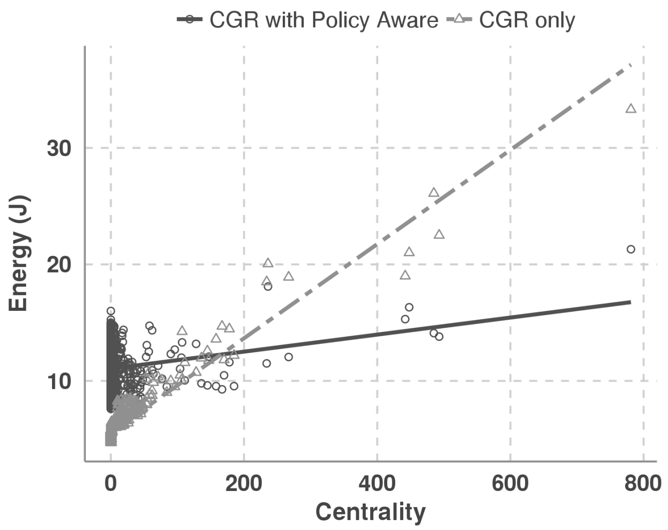


Figure 11: CGR protocol with Policy-Aware Algorithm.

7. Conclusion

We present CGR a reliable, robust, and energy-efficient routing protocol for L2N based on the nodes' centrality. CGR represents an alternative strategy for routing data in L2N that finds intermediate nodes based on the topological importance. CGR favors data aggregation by approximating the Minimum Steiner Tree. Our simulation results show that CGR presents smaller Steiner number and TPR than the protocols evaluated. Also, CGR and RPL show similar delivery rate close to 100 % when some packet retransmissions are allowed. In terms of energy consumption, CGR presented the lowest expenditure, followed by RPL (due its high overhead), and the traditional protocols. Unlike traditional protocols, CGR has great potential for L2N by dealing with the constraints of these networks and presenting low additional complexity costs.

To mitigate the energy hole problem, we propose the Policy-Aware algorithm. We showed that it could be used as a scheme to improve the network lifetime.

This work generates several insights for future work, such as: using multiple radios and/or multi-rate nodes; and select different aggregation techniques to work with CGR.

Acknowledgment

We would like to thank the research agencies, CAPES, CNPq and FAPEMIG for funding this research.

References

[1] M. A. M. Vieira, C. N. Coelho, D. C. da Silva, J. M. da Mata, Survey on wireless sensor network devices, in: EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings, Vol. 1, 2003, pp. 537–544. doi:10.1109/ETFA.2003.1247753.

[2] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, A. A. F. Loureiro, On the design of green protocols for underwater sensor networks, IEEE Communications Magazine 54 (10) (2016) 67–73. doi:10.1109/MCOM.2016.7588231.

[3] S. Dwars, T. Phinney, P. Thubert, K. Pister, Industrial Routing Requirements in Low Power and Lossy Networks, RFC 5673 (2009).

[4] M. Dohler, T. Watteyne, T. Winter, D. Barthel, Routing requirements for urban low-power and lossy networks, RFC 5548 (2009).

[5] B.-H. Liu, V.-T. Pham, N.-T. Nguyen, An Efficient Algorithm of Constructing Virtual Backbone Scheduling for Maximizing the Lifetime of Dual-radio Wireless Sensor Networks, Int. J. Distrib. Sen. Netw. 2015 (2015) 5:5–5:5. doi:10.1155/2015/475159.

[6] R. W. Coutinho, A. F. Boukerche, L. Vieira, A. Loureiro, A Novel Centrality Metric for Topology Control in Underwater Sensor Networks, in: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '16, ACM, New York, NY, USA, 2016, pp. 205–212. doi:10.1145/2988287.2989162.

[7] D. S. J. De Couto, D. Aguayo, J. Bicket, R. Morris, A High-throughput Path Metric for Multi-hop Wireless Routing, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom '03, ACM, New York, NY, USA, 2003, pp. 134–146. doi:10.1145/938985.939000.

[8] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis, Four-bit wireless link estimation, in: Proceedings of the Sixth ACM Workshop on Hot Topics in Networks (HotNets-VI), 2007, pp. 419–420.

[9] P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 routing protocol for low power and lossy networks, RFC 6550 (2012).

[10] B. P. Santos, M. A. M. Vieira, L. F. M. Vieira, eXtend collection tree protocol, in: 2015 IEEE Wireless Communications and Networking Conference (WCNC), 2015, pp. 1512–1517. doi:10.1109/WCNC.2015.7127692.

[11] E. M. R. Oliveira, H. S. Ramos, A. A. F. Loureiro, Centrality-based routing for Wireless Sensor Networks, in: 2010 IFIP Wireless Days, 2010, pp. 1–5. doi:10.1109/WD.2010.5657731.

[12] B. Krishnamachari, D. Estrin, S. Wicker, Modelling data-centric routing in wireless sensor networks, in: IEEE INFOCOM, Vol. 2, 2002, pp. 39–44. doi:10.1.1.21.8739.

[13] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys '06, ACM, New York, NY, USA, 2006, pp. 419–420. doi:10.1145/1182807.1182885.

[14] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, M. Alves, Radio Link Quality Estimation in Wireless Sensor Networks: A Survey, ACM Trans. Sen. Netw. 8 (4) (2012) 34:1–34:33. doi:10.1145/2240116.2240123.

[15] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, D. Estrin, Statistical model of lossy links in wireless sensor networks, in: IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005., 2005, pp. 81–88. doi:10.1109/IPSN.2005.1440900.

[16] D. LaI, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, A. Keshavarzian, Measurement and characterization of link quality metrics in energy constrained wireless sensor networks, in: Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, Vol. 1, 2003, pp. 446–452 Vol.1. doi:10.1109/GLOCOM.2003.1258278.

[17] M. Caleffi, L. Paura, Bio-inspired link quality estimation for wireless mesh networks, in: 2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops, 2009, pp. 1–6. doi:10.1109/WOWMOM.2009.5282423.

[18] A. S. Cacciapuoti, M. Caleffi, L. Paura, M. A. Rahman, Link quality estimators for multi-hop mesh network, in: 2014 Euro Med Telco Conference (EMTC), 2014, pp. 1–6.

[19] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, MobiCom '04, ACM, New York, NY, USA, 2004, pp. 114–128. doi:10.1145/1023720.1023732.

[20] L. F. M. Vieira, M. G. Almiron, A. A. Loureiro, Link probability, node degree and coverage in three-dimensional networks, Ad Hoc Networks 37, Part 2 (2016) 153 – 159. doi:http://dx.doi.org/10.1016/j.adhoc.2015.08.011.

[21] S. P. Borgatti, M. G. Everett, A Graph-theoretic perspective on centrality, Social Networks 28 (4) (2006) 466 – 484. doi:http://dx.doi.org/10.1016/j.socnet.2005.11.005.

[22] S. Dolev, Y. Elovici, R. Puzis, Routing Betweenness Centrality, J. ACM 57 (4) (2010) 25:1–25:27. doi:10.1145/1734213.1734219.

[23] E. Mohsenifard, A. Ghaffari, Data Aggregation Tree Structure in Wireless Sensor Networks Using Cuckoo Optimization Algorithm, Information Systems & Telecommunication (JIST) (2016) 182.

[24] L. Villas, A. Boukerche, R. B. de Araujo, A. A. F. Loureiro, Highly Dynamic Routing Protocol for data aggregation in sensor networks, in: The IEEE symposium on Computers and Communications, 2010, pp. 496–502. doi:10.1109/ISCC.2010.5546580.

[25] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection Tree Protocol, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09, ACM, New York, NY, USA, 2009, pp. 1–14. doi:10.1145/1644038.1644040.

[26] B. S. Peres, O. A. d. O. Souza, B. P. Santos, E. R. A. Junior, O. Goussevskaia, M. A. M. Vieira, L. F. M. Vieira, A. A. F. Loureiro, Matrix: Multihop Address Allocation and Dynamic Any-to-Any Routing for 6LoWPAN, in: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '16, ACM, New York, NY, USA, 2016, pp. 302–309. doi:10.1145/2988287.2989139.

[27] F. K. Hwang, D. S. Richards, P. Winter, The Steiner tree problem, Elsevier, 1992.

[28] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: A Survey, Comput. Netw. 38 (4) (2002) 393–422. doi:10.1016/S1389-1286(01)00302-4.

[29] E. N. Gilbert, H. O. Pollak, Steiner Minimal Trees, SIAM Journal on Applied Mathematics 16 (1) (1968) 1–29. doi:10.1137/0116001.

[30] A. Z. Zelikovsky, An 11/6-approximation algorithm for the network steiner problem, Algorithmica 9 (5) (1993) 463–470. doi:10.1007/BF01187035.

[31] S. Vofi, K. Gutenschwager, A chunking based genetic algorithm for the steiner tree problem in graphs, Network Design: Connectivity and Facilities Location, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 40 (1998) 335–355.

[32] G. Robins, A. Zelikovsky, Improved Steiner Tree Approximation in Graphs, in: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '00, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000, pp. 770–779.

[33] H. S. Ramos, E. M. R. Oliveira, A. Boukerche, A. C. Frery, A. A. F. Loureiro, Characterization and mitigation of the energy hole problem of many-to-one communication in Wireless Sensor Networks, in: 2012 International Conference on Computing, Networking and Communications (ICNC), 2012, pp. 954–958. doi:10.1109/ICCNC.2012.6167567.

[34] L. Liu, Y. Song, H. Zhang, H. Ma, A. V. Vasilakos, Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks, IEEE Transactions on Computers 64 (3) (2015) 818–831. doi:10.1109/TC.2013.229.

[35] M. Caleffi, I. F. Akyildiz, L. Paura, On the Solution of the Steiner Tree NP-Hard Problem via Physarum BioNetwork, IEEE/ACM Transactions on Networking 23 (4) (2015) 1092–1106. doi:10.1109/TNET.2014.2317911.

[36] P. Basu, J. Redi, Effect of overhearing transmissions on energy efficiency in dense sensor networks, in: Proceedings of the 3rd international symposium on Information processing in sensor networks, ACM, 2004, pp. 196–204. doi:10.1145/984622.984652.

[37] I. Solis, K. Obraczka, In-Network Aggregation Trade-Offs for Data Collection in Wireless Sensor Networks, Int. J. Sen. Netw. 1 (3/4) (2006) 200–212. doi:10.1504/IJSNET.2006.012035.

[38] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The trickle algorithm, Internet Engineering Task Force, RFC 6206 (2011).

[39] J. Li, P. Mohapatra, Analytical Modeling and Mitigation Techniques for the Energy Hole Problem in Sensor Networks, Pervasive Mob. Comput. 3 (3) (2007) 233–254. doi:10.1016/j.pmcj.2006.11.001.

[40] X. Wu, G. Chen, S. Das, Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution, Parallel and Distributed Systems, IEEE Transactions on 19 (5) (2008) 710–720. doi:10.1109/TPDS.2007.70770.

[41] F. Ishmanov, A. S. Malik, S. W. Kim, Energy consumption balancing (ECB) issues and mechanisms in wireless sensor networks (WSNs): a comprehensive overview., European Transactions on Telecommunications 22 (4) (2011) 151–167.

[42] J.-P. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing metrics used for path calculation in low-power and lossy networks, RFC 6551 (2012).

[43] J. Tripathi, J. C. de Oliveira, J. P. Vasseur, A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks, in: 2010 44th Annual Conference on Information Sciences and Systems (CISS), 2010, pp. 1–6. doi:10.1109/CISS.2010.5464820.

[44] D. C. G. E. Zurich, Simulator for network algorithms (Sinalgo) (2014).