Resilient Approach for Energy Management on Hot Spots in WSNs

Fernando Gielow* and Michele Nogueira and Aldri Santos
NR2 – Informatics Department – UFPR – Brazil
emails: {fhg07, michele, aldri}@inf.ufpr.br

***Abstract*—Mitigating Hot Spot energy consumption in Wireless Sensor Networks (WSNs) is a demanding task. Hot Spots have their nature bounded on routing – they are areas overloaded with high traffic rate, resulting in expanding energy holes. Hot Spots mitigation approaches have employed several techniques supported by routing protocols, such as multiple or mobile sinks, the deployment of more sensors in the Hot Spot area and unequal clustering. Albeit their advantages, cluster-based routing protocols to mitigate Hot Spots manage route maintenance inefficiently, leading to poor network performance and high energy consumption. This work presents an energy management approach to mitigate Hot Spots in WSN, supported by unequal clustering and low-costly dynamic route maintenance. We also generalize our solution for an energy management architecture that considers Hot Spot issues. Results show resilient routing and an efficient energy management, improving both network lifetime and performance.**

I. Introduction

Wireless sensor networks (WSN) have been envisaged to support different applications, such as environmental monitoring, security systems and others [1]. To obtain an effective communication among sensor nodes, the establishment and maintenance of routes to a base-station (or sink) is necessary, particularly, on networks with continuous data flows. Energy consumption is different on each sensor node of WSNs. Nodes distributed in a homogeneous way suffer a funnelling effect due to the many-to-one traffic pattern, present in data gathering applications, for instance. As sensors get close to the sink, the number of routes decreases, overloading some areas with data traffic and triggering a gradual process that creates and expands an energy hole around the sink [2]. Hot Spots comprehend those areas overloaded with data traffic.

Energy efficiency is one of the primary challenges of WSNs. The tiny sensors are powered with limited battery which cannot be recharged afterward. Approaches on different layers of the protocol stack attempt to manage energy and prolong the network lifetime [3]. However, the Hot Spot issue has its nature strictly bounded on routing, and it cannot be mitigated at other layers [4]. Hence, existing approaches employ different techniques in association with routing to reduce Hot Spot effects, such as transmission power control, the use of multiple data sink and cluster formation.

Different cluster-based routing protocols for WSNs aim to balance the unequal energy consumption that Hot Spots create, as DAR [2], UCR [5] and EECRP [6]. Organizing the network in clusters has as advantages data aggregation, low overhead on topology management and route stability. Those characteristics assist in the efficient energy management of the network. However, among existing cluster-based routing protocols, either they are not practical, or spend more energy to manage energy consumption [2], or have negative impact on the network performance - measured by packet delivery ratio, latency, overhead and others [2], [5].

In general, cluster-based routing protocols employ unequal cluster sizes to mitigate Hot Spot issues, such as in [5]. However, even though more routes can be used to reach the sink, clusters close to it have less members and, hence, less cluster-head rotations can be done. Further, cluster-head rotations can break a route when a more distant cluster-head is chosen. Thus, nodes that were able to reach the old cluster-head cannot transmit their data to the new one. Those aspects highlight the need of an efficient and low-costly route maintenance algorithm to repair those broken routes, without demanding too much resources from the network and deteriorating its lifetime or performance.

This work presents an energy management approach to mitigate Hot Spots in WSN. It is supported by a routing protocol based on unequal clusters, that also repairs the routes in a dynamic and low-costly way. Then, we generalize this protocol for an energy management architecture to WSNs that considers Hot Spot issues. Different from other approaches, we employ a transmission power control technique for cluster formation, instead of RSSI measurements and the exact distance between nodes. RSSI measurements [7] are unreliable, and knowing the exact distance between nodes is unrealistic (GPS could solve this, but at an expensive price). Further, as broken links may appear due to the cluster-head rotations, we develop a maintenance algorithm needing no control packets. All control information is piggybacked on messages responsible for data gathering, saving energy.

Simulation results show that our approach mitigates Hot Spots with an efficient energy management scheme, improving network lifetime and resulting in less node deaths close to the sink. Also, the dynamic maintenance of routes has increased routing resilience, as shown by the high data delivery rate on results. Our approach was compared with the UCR [5] protocol, since both mitigate Hot Spots by employing unequal sized clusters, even though with different clustering algorithms.

*This work was supported by Fundação Araucária Agency, grant number 2007021683.

The paper proceeds as follows. Section II presents the related work. Section III details the new protocol, named RRUCR. Section IV defines an energy management architecture, called CEA. Section V shows evaluations of RRUCR. Finally, Section VI presents conclusions and future work.

II. Related Work

Approaches to reduce or balance energy consumption in WSNs were proposed for different layers of the protocol stack [3]. At the routing layer, proposals have taken into account topology control, mobility and data reduction by in-network processing, compression and prediction. Further, routing also employs techniques common to other layers, such as transmission power control and sleep/wakeup turns.

Clustering has been applied in association with those techniques to further reduce energy consumption in WSN. The cluster-based routing protocol LEACH [8] uses cluster-head rotations to balance clusters internal energy. However, it results in a large energy consumption, since cluster-heads aggregate and send data directly to the sink. The HEED protocol [9] controls the transmission power used by nodes to communicate, and the initial probability of a node to become a cluster-head depends on its energy and on the intra-cluster communication energy cost. Albeit dealing with energy management, these proposals do not consider Hot Spot issues.

The EECS protocol [10] introduces a competitive approach among network nodes without packet exchange iterations, extending LEACH and HEED. Unequal clusters are created with size inversely proportional to their distance to the sink. Thus, distant clusters are smaller and their cluster-heads preserve energy to send long haul messages. However, as there is direct communication with the sink, energy consumption is high, and the region where Hot Spots occur is hard to be determined due to the absence of a backbone.

Several works have analysed the impact of Hot Spots in WSNs. In [11], authors evaluated strategies that mitigate Hot Spots. They concluded that Hot Spots cannot be mitigated only using power transmission control techniques, although network lifetime can be extended applying them. The use of the sink mobility approach to manage energy was examined in [12]. They observed that sink mobility can prevent Hot Spots, however, its use in real scenarios is constrained or even impractical, due to space or energy restrictions [13]. In [14], authors analysed the use of different parameters, such as buffer occupation, packet loss and link layer contention, to detect Hot Spots, but focusing on MANET multimedia applications.

In order to mitigate Hot Spots, different approaches have been developed. In [15], energy balance was modeled as a particle swarm optimization problem, based on redefining the particles fly rules for the routing optimization. Nonetheless, its suitability for real applications is not comproved, as authors do not measure overhead and delay added by the proposal. In [16], Wu and Chen proposed the uneven distribution of sensor nodes in the network, deploying more nodes close to the sink. Although this approach has balanced the energy and mitigated Hot Spots, its use in real applications is impractical, as it depends on a biased deployment of sensors.

DAR [2] uses a slightly different technique. In order to manage energy consumption, it may establish longer routes, enforcing a more balanced participation of nodes. However, the approach resulted in higher energy consumption, because there are cases in which a packet must travel backward before travelling in the direction of the sink, for instance.

In the last years, unequal sized clustering has been extensively employed to mitigate Hot Spots, such as in LUCA [17], UCR protocol [5], EBUCP [18] or EECRP [6]. In contrast to EECS [10], which increases the cluster size as nodes get close to the sink, UCR and EECRP decrease the cluster size close to the sink, creating more possible routes. However, these protocols are not concerned about backbone maintenance. Thus, as cluster-heads suffer rotations, the communication link between them can break when more distant nodes in relation to the other cluster are selected as new cluster-heads. Such behaviour damages data delivery rate. Further, these works are not particularly focused on the energy management problem.

III. The RRUCR protocol

This section details our protocol, called Rotation Reactive Unequal Cluster-based Routing protocol (RRUCR), developed to support our energy management approach. RRUCR mitigates Hot Spots by applying unequal sized clusters (as shown on Figure 1), cluster-head rotations [19], and integrating the backbone maintenance in data flows without the use of control messages. The RRUCR protocol consists of five operations: definition of each node scope (its competition range), clustering, initial backbone creation, cluster-head rotation, and data gathering, which supports also the backbone maintenance.

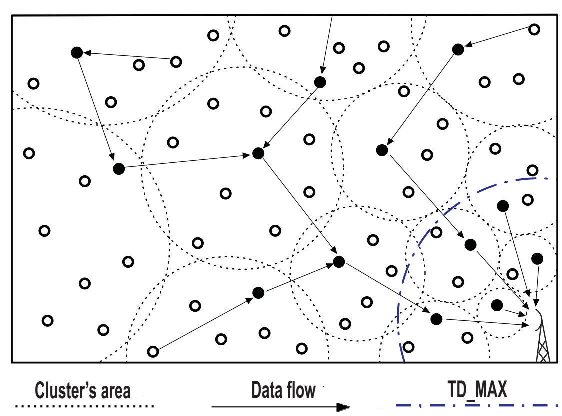

Figure 1. Network organization in unequal sized clusters

The first three operations occur at the deployment of the network. Thus, the number of clusters can only decrease throughout time as a result of nodes depleting their batteries. An initial backbone is established after the creation of clusters, and the remaining operations occur many times, providing energy balance and higher data delivery rates. The next subsections describe individually the operations of RRUCR.

The transmission powers employed in the definition of scopes are indexed in a ranked table, kept by all nodes, and only their indexes are sent on any kind of message. Thus, indexes point out the referenced or stored transmission power.

A. Definition of scopes

The sink initially broadcasts *INCR_POT* messages covering the transmission powers to be used, as shown on Figure 2-top (these potences are pre-defined and they may vary according to equipment used). When nodes receive this message for the first time, they store on a variable $RBase$ the transmission power used by the sink, and return an acknowledgement message. Hence, the sink knows the lowest ($RFMin$, used to reach node A) and highest ($RFMax$, used to reach node B) transmission powers used to reach a node. Next, the sink broadcasts a *SETUP_CONFIG* message in the maximum transmission power, containing the values of the variables $RFMin$ and $RFMax$.

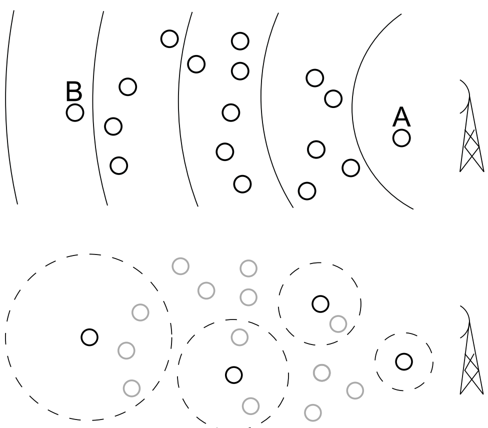

Figure 2. Scope definition

Given that nodes may not be reached by this message, farther nodes (which $RBase = RFMax$) will retransmit the *SETUP_CONFIG* message, and the remaining nodes, which had not been reached, will also retransmit the message until all possible nodes are covered. This message also contains a counter $cont$ that informs how many hops it has been forwarded until reaching the current node.

The clustering operation pre-stipulates two limits for the transmission powers: pot_limit, the index of the maximum power that can be used by the nodes reached on the first wave of messages *SETUP_CONFIG*, and pot_max_global, the index of the maximum transmission power that can be used by the other nodes. These limits exist because with the cluster-head (CH) rotations it is possible that nodes which were reachable get too far from each other, not being covered by the inter-clusters transmission power. With these limits, clusters will have a diameter inferior to the range of the transmission power used on inter-clusters communication, avoiding to break links.

Nodes reached by the first wave of the message *SETUP_CONFIG* will use the transmission power of the following index on the clustering operation (coverage scope of some nodes is illustrated in Figure 2-bottom):

$$Scope = \lfloor ((1 - \frac{(RFMax - RBase)}{(RFMax - RFMin)}) * pot\_limit) \rfloor \quad (1)$$

For other nodes, $Scope = (pot\_limit + cont)$, with the maximum value of $Scope$ determined by pot_max_global.

B. Clustering

Having the competition scope of each node defined, the clustering operation starts. Its flowchart is illustrated on Figure 3 (due to lack of space, *FINAL_HEAD* messages will have their names shortened to *FINAL* on the flowchart). Based on a pre-stipulated probability $pBeTHead$, nodes are randomly selected as candidates for CHs. Then, they send a competition message informing their energy to everyone on scope (defined on Equation 1). Nodes that do not receive a competition message will also candidate to CH, this measure avoids areas without any CH.

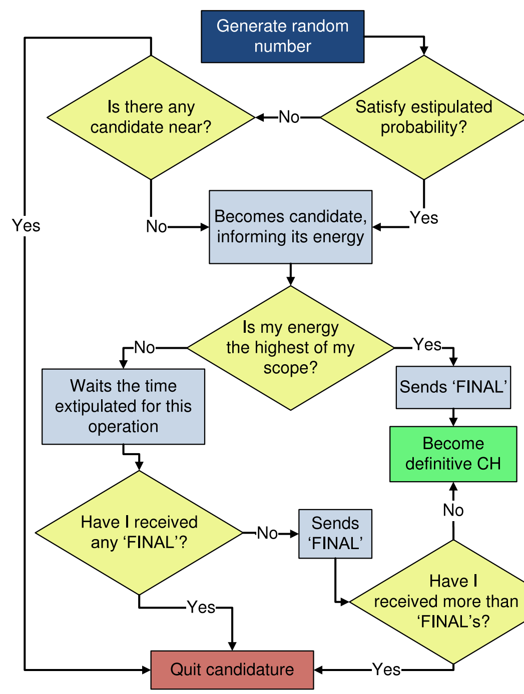

Figure 3. Flowchart of the clustering operation

After receiving those competition messages, nodes verify if their energy is higher than the energy of their neighboors in order to become definitive CHs, then they broadcast a *FINAL_HEAD* message. All nodes count the number of received *FINAL_HEAD* messages and store it in the $finals$ variable. After the time dedicated to this phase, nodes that did not receive any of these messages also become a candidate, sending a *FINAL_HEAD* message. At the end, if $finals > 2$, the candidate node gives up the election. Hence, it increases the network coverage, without creating too many clusters.

Nodes definitively established as CHs broadcast an announcement message. Hence, common nodes will be able to select a CH based on its RSSI (Received Signal Strength Indicator). Each node keeps the identifier (ID) of the selected CH, and sends a *JOIN_CLUSTER* message, informing its energy. On receiving those messages, the CH keeps the highest energy value for future rotation operations. Note that no re-clustering is scheduled, hence, this phase does not happen again.

C. Initial backbone creation

This operation is represented by the Algorithm 1, and consists on the establishment of a valid initial backbone that allows nodes to reach the sink in few hops. The sink broadcasts a *BEACON_ROUTE* message (*l.* 3) to all nodes in the *TD_MAX* area (Figure 1), defined by a transmission power index. CHs in this area will forward received data directly to the sink, and this is usually where the Hot Spot takes place.

Algorithm 1 - Initial backbone creation

```
1:  procedure INITIALIZE PROCCESS
2:    sj.wave ← ∞, ∀node sj
3:    base station si broadcast BEACON_ROUTE(si.ID,0);
4:  end procedure
5:  procedure HANDLE RECEIVED MESSAGES
6:    on node si receiving BEACON_ROUTE(sj.ID,counter,RSSI) from sj do
7:      if si.wave > msg_wave or
8:            (si.wave = msg_wave and RSSI > si.lastRssi) then
9:        si.wave ← msg_wave;
10:       si.lastRssi ← RSSI;
11:       if si is CH then
12:         si.next_hop ← sj.ID;
13:       end if
14:     end if
15:     si broadcast BEACON_ROUTE(si.ID,si.wave+1);
16:   end on
17: end procedure
```

The *BEACON_ROUTE* message carries a $counter$ field that informs how many hops the message has travelled, enabling nodes to know how far they are from the sink. Nodes keep such data in a variable $wave$. Thus, when a node receives a *BEACON_ROUTE* message, its $wave$ is updated if $counter$ is lower than its current $wave$ (*l.* 7), and $next\_hop$ is set with the ID of the node that sent the message (*l.* 9-13). $next\_hop$ is also updated if $counter = wave$ and the message's RSSI is higher than the RSSI of the message that caused the last $next\_hop$ update (*l.* 8). The *BEACON_ROUTE* message is then retransmitted, increasing its $counter$ field (*l.* 15).

D. Cluster-heads rotations

An energy percentage threshold, called $pRotate$, is prestipulated for the CHs rotation. When the CH's energy gets lower than the $pRotate$ percentage of the highest node's energy in its cluster (originally obtained from the *JOIN_CLUSTER* messages), this CH broadcasts a rotation request, informing its energy. Nodes will answer informing their energy if they belong to the cluster and have a higher energy.

The CH that requested a rotation will select as new CH the node that has informed the highest energy, also adopting its ID as $next\_hop$ (Figure 4-I and 4-II)[1]. Then the requester

[1]Please note that the figure represents only a part of the network

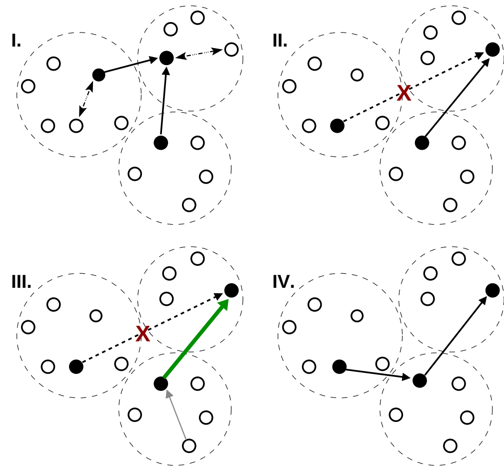

Figure 4. Repairing a broken link generated in a rotation of cluster-heads

CH will broadcast a *DENOMINATE_CH* message, containing its previous $next\_hop$ and the new CH's ID. On receiving this message, the new CH will update its $next\_hop$ to the informed on the received message and will consider the highest energy on the cluster as being its own. The remaining nodes in the cluster will now communicate with the new CH.

As broken links may appear due to rotations (dotted link in Figure 4-II, marked with a 'X'), on receiving a *DENOMINATE_CH* message, both the new CH and the CHs that used to communicate with the old CH will be forced to update their routes as soon as a valid route is detected.

E. Data gathering & Routes maintenance

This operation is represented on Algorithm 2. In order to send collected data to the sink (*l.* 1-3), nodes broadcast a *DATA_GATHERED* message with both the data and the value of $next\_hop$ (thin grey line of Figure 4-III)[2]. The CH whose ID equals the received message's $next\_hop$ (*l.* 15-17) will update the $next\_hop$'s field to its own $next\_hop$ and then forward the message in broadcast (thick line of Figure 4-III). In such message, the CH will also send a value $wr = 100 * wave + RBase$ that identifies its distance to the base, and will be used for the backbone maintenance.

Whenever a CH receives a *DATA_GATHERED* message, it may also update its route. When the maintenance is obligatory due to rotation (*l.* 6-10), the CH will only verify if the CH which sent the message is between it and the sink (its wr is lower), and if it is, it will be adopted as being the next hop (Figure 4-IV). If the maintenance is not obligatory (*l.* 11-14), the route will be updated if the new possibility of route has a lower cost (wr) than the current. The cost of the selected route is always stored for future comparisons (*l.* 8 and 13).

The cost wr mentioned is calculated through the variables $RBase$, $Scope$ and $wave$ – all of which are good estimations of distance to the sink. Nodes have these values proportional to their distance to the sink, being $wave$ itself the number of hops necessary to reach it on the initial backbone.

Algorithm 2 - Data collection and routes maintenance

```
procedure SEND COLLECTED DATA
    node si broadcast DATA_GATHERED(si.{data,ID,wr,next_hop});
end procedure
procedure HANDLE AND ROUTE RECEIVED MESSAGES
    on node si receiving DATA_GATHERED(sj.{data,ID,wr,next_hop}) from sj do
        if si.forceRouteUpdate and si.wr > sj.wr then
            si.next_hop ← sj.ID;
            si.routeCost ← sj.wr;
            si.forceRouteUpdate ← false;
        end if
        if si.routeCost > sj.wr then
            si.next_hop ← sj.ID;
            si.routeCost ← sj.wr;
        end if
        if si is not base and si.ID = sj.next_hop then
            si broadcast DATA_GATHERED(si.{data,ID,wr,next_hop});
        end if
    end on
end procedure
```

Although the backbone maintenance does not guarantee the shortest path, it decreases overhead and saves energy, since it is a dynamic process that does not use control packets.

IV. Cluster-based Energy Management Architecture

In order to complement our energy management approach, we generalized our routing protocol to a Cluster-based Energy Architecture (CEA) that considers Hot Spot issues, as illustrated in Figure 5. Gray boxes represent main modules, arrows represent interactions and white boxes represent components of modules. CEA consists of four basic modules: an intra-cluster energy manager, inter-cluster energy manager, data gathering and route management. Since CEA is based on clusters, energy management occurs essentially by their control, formation and maintenance. The two modules inside the dotted box have functionalities related to energy management.

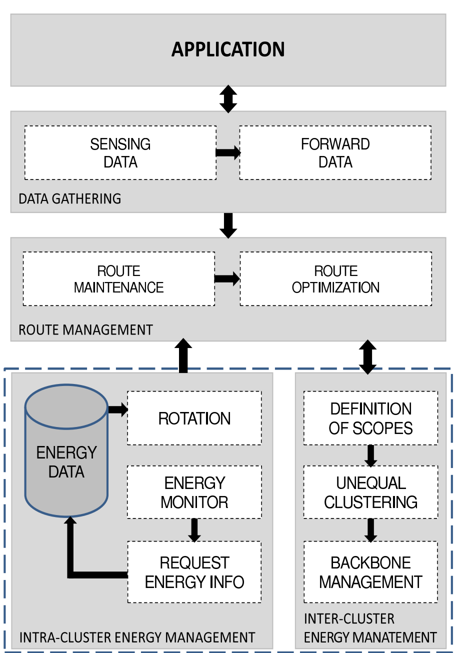

Figure 5. The CEA Architecture

The **inter-cluster energy management module** consists of procedures responsible for managing energy control among clusters of the network. Procedures in this module intend to reduce energy consumption of nodes by efficiently defining the scope of nodes, determining unequal clusters and creating a backbone for the network. Those procedures are executed in a distributed way, in which each node runs those procedures cooperatively. This module represents a passive way to manage energy consumption across the network. Since each node executes its procedures without global information, it reduces complexity.

In the inter-cluster energy module, competition range of nodes (scope) is the first step towards energy management. Criteria, such as transmission power, physical distance, RSSI measurements, can be employed to determine the competition range of nodes. A previous anaylsis of those criteria must be done, in order to choose one. As example, in [7], we can find analyses about the use of RSSI measurements. Considering the scopes and energy information, clusters are then formed, aiming to create more clusters close to the sink. At the end of cluster formation, the network backbone should be determined.

The **intra-cluster energy management module** is composed of procedures to monitor energy of the node, request energy information to cluster co-members and participate in the rotation process of cluster-heads. The energy of a node is monitored periodically to verify its level and compared to a threshold, defined considering the highest energy level among all members of a cluster. Depending on the node energy level in relation to this threshold, actions are triggered, such as cluster-head rotation. Such action intends to balance energy consumption avoiding premature node deaths. To evaluate the highest energy level of a cluster, a node requests this information to its co-members following an efficient procedure. All replied energy data of co-members will be stored and used by the cluster-head rotation procedure. When the energy procedure indicates the necessity to rotate the cluster-head, that is, when the energy level of the cluster-head is below a stipulated percentage of the highest energy on the cluster, the rotation procedure starts. The rotation triggers an on-demand route maintenance operation for both the new cluster-head and the cluster-heads that used to communicate with the previous one - this takes place together with the data gathering module.

The **data gathering** and **route management** modules consist of procedures that support energy management modules. It integrates routing management with energy management, defining how to maintain and optimize routes and aiming to improve both network performance and the efficient use of energy. It innovates in relation to existing energy management architectures [20], [21], by considering together energy and performance. This module owns also a component to sense data, being data humidity, temperature, light or others, which is determined strictly by the application needs. Data forward procedures should follow the network backbone created until reaching the sink. The backbone aims to optimize the path to the sink in terms of energy consumption.

V. Evaluation of Resilience

Although there are several cluster-based routing protocols, we only compared ours with the UCR [5] protocol, since both mitigate Hot Spots by employing unequal sized clusters, even though with different clustering algorithms. Node mobility was disconsidered because the mobility itself is an alternative way to mitigate Hot Spots, due to the dynamic routes [11].

We implemented both protocols on the NS-2.30 environment and simulated operating with IEEE 802.11b at the MAC layer. A homogeneous WSN was established, and the parameters were chosen in order to simulate a periodic data gathering application with Mica2 sensor motes. The radio parameters were set according to the CC1000 radio used by the Mica2 architecture. Each node has a 0.1% probability per second to generate data, being roughly transmitted to the sink, i.e., without any aggregation technique.

The WSN operates for 5000 seconds and consists of 700 sensors distributed in a square area, measuring 1000m at each side. The location of all nodes and sink is random in each simulation. The initial energy of each node comprehends values between 0.9 and 1.1 joules. For both protocols the probability of a node to participate in the cluster-head election was 35%, collected data has 32 bytes, and the inter-cluster transmission power is 3.16227mW (the highest power supported by the Mica2 motes). The TD_MAX area, where all nodes communicate directly with the sink, has 149m. In UCR, the maximum cluster radius limit was 140m. In RRUCR, were used $pot\_limit = 0.25118mW$, $pot\_max\_global = 0.63095mW$ and $pRotate = 65\%$. The power values used for scope definition were obtained from the CC1000 datasheet, they were all crescently indexed, as previously detailed.

The protocol had its performance evaluated under three types of simulation scenarios: operation without failures, with failures close to the sink and with failures far from the sink (the distance is quantified in hops). In the scenario with failures close to the sink, 8 nodes that take from 0 to 2 hops to reach the sink, and 8 nodes that take from 1 to 5 hops are randomly turned off. In the scenario with failures far from the sink, 25 nodes are turned off, being them 12 that take from 2 to 5 hops, and 13 that take from 3 to 6 hops. In both situations failures occur at 400s of simulation.

The metrics used for the evaluation of resilience, that is, the capability of saving energy and keeping network performance, are *number of hops from each cluster-head to the sink*, *total amount of energy*, *lifetime*, *number of dead nodes in relation to their distance to the sink*, *Data delivery rate* (which considers the percentage of arrival of the last 30 data packets sent) and *number of rotations*. Those metrics assess the Hot Spot mitigation, the efficiency of the created routes, their maintenance and the energy balance of the network. We ran 35 simulations for each protocol and each kind of described network, obtaining in a 95% confidence interval.

The NS-2.30 RRUCR code is available under the terms of the GLPL license and can be found at the website www.nr2.ufpr.br/~fernando/rrucr/rrucr_codes.php.

A. Cluster distribution & Energy consumption

The number of clusters formed in the network is an important factor for WSNs. With too many clusters, there is more energy consumption due to the increased number of messages exchanged. However, a small quantity of clusters causes more overhead and higher energy consumption due to the necessity of higher transmission powers. Due to its characteristics, UCR's clusters must be smaller, in order to decrease the probability of choosing a cluster-head much far.

In Figure 6, we observe that although RRUCR owns clusters with more hops to reach the sink, it creates less clusters. On average, RRUCR created 43 clusters, while UCR created 67. This difference of 35.82% more clusters in UCR resulted in a higher energy consumption on it, as shown in Figure 7. This happens because with an increased number of clusters, more messages will be sent due to the higher number of rotations. Also, with more but too small clusters, the efficacy of rotations is compromised. The increase of the rotation amount is proved in the next subsection.

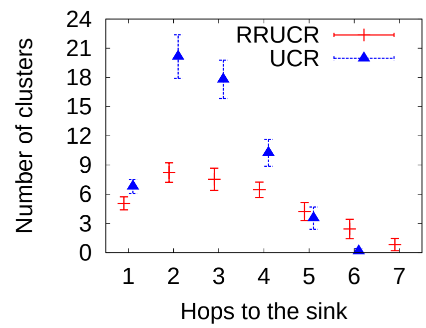

Figure 6. Clusters distribution

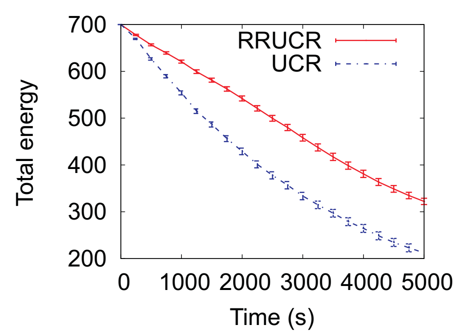

Figure 7. Energy decrease across time

B. Network lifetime

Lifetime is the time elapsed until the first node death. A way to extend this time on cluster-organized WSN is using cluster-head rotations to distribute cluster energy consumption. Compared to UCR, RRUCR managed better energy consumption and increased the network lifetime in 21.36% on scenarios of network without failures. On scenarios with failures far from the sink there is an increase of 17.16%, whereas under failures close to the sink, it is 13.55%, as shown in Figure 8.

Thus, the creation of more routes and their maintenance balance the energy consumption of the network as a whole. On both protocols the best lifetime is reached on scenarios of networks with failures far from the sink, because there are less packets needing to travel longer distances.

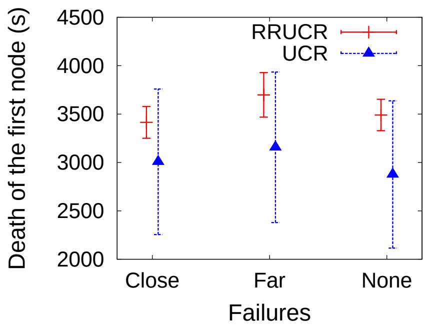

Figure 8. Network lifetime

To evaluate the performance of both protocols on the mitigation of Hot Spots, the number of dead nodes across the distance in meters to the sink (DS) was measured, as shown in Figure 9. In this figure, we consider distances inferior to 40m, between 40m and 80m, between 80m and 160m and superior to 160m. The number of dead nodes was measured in the simulation times of 3000, 4000 and 5000 seconds. Both protocols minimized Hot Spot effects by decreasing the number of deaths near the sink. Hence, more nodes close to the sink can be used for last hops in communication. But more nodes die on UCR, due to the increased number of clusters and the consequent cluster-head rotations (Figure 10).

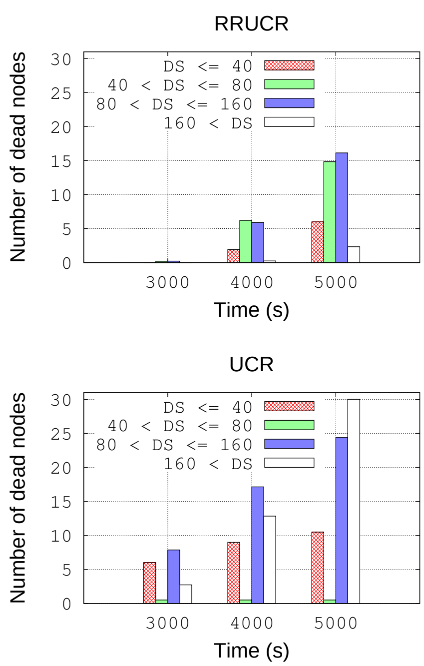

Figure 9. Dead nodes vs. distance

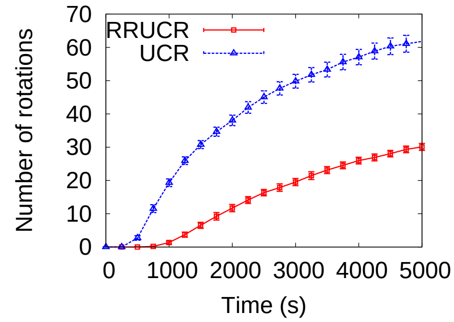

Figure 10. Cluster-head rotations

C. Data delivery rate

Only energy distribution and the existence of a valid initial backbone do not guarantee a satisfactory delivery rate. Figure 11 shows that RRUCR presented higher data delivery rates, and UCR matched them only in the beginning of the simulations, before any cluster-head rotation or failure takes place. When rotations start, at approximately 700s, data delivery start to drop due to the broken links that appear.

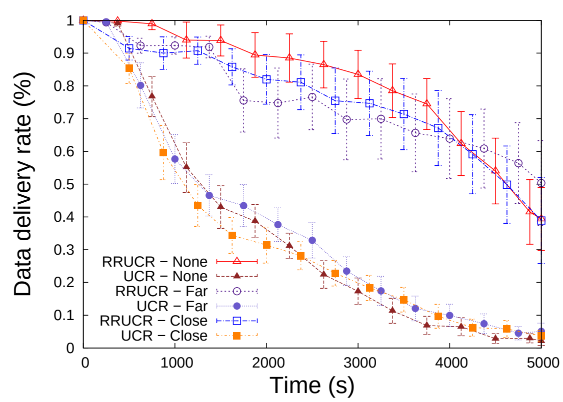

Figure 11. Delivery rates with and without failures

Repairing the routes is essential when there is possibility of having broken links, and our dynamic maintenance algorithm enabled RRUCR to have higher data delivery rates also when the number of cluster-head rotations is regarded, as shown in Figure 12. On overall, UCR is harmed in a more severe way. It needs to perform more cluster-head rotations (Figure 10) due to the higher quantity of clusters (Figure 6), thus generating more broken links between clusters.

It is proved that a route maintenance operation is required to keep data delivery rate unharmed. By these results, we observe that RRUCR keeps better network performance when compared to UCR in terms of packet delivery rate, which can lead to better energy efficiency. If retransmissions were considered in the case of data loss, the absence of an efficient backbone would burden nodes with even more intense traffic, demanding much more energy with retransmissions.

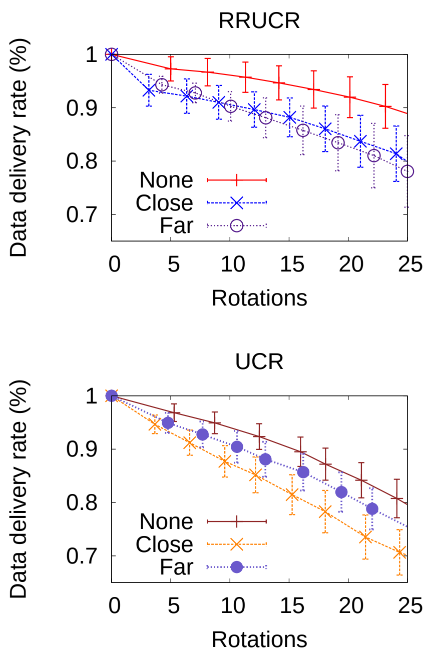

Figure 12. Data delivery rate vs. Cluster-head rotations

VI. Conclusion

This work presented an approach composed of a routing protocol and an architecture to mitigate WSN Hot Spots, balancing energy consumption and increasing network performance. The cluster-based routing protocol, called RRUCR, makes dynamic route maintenance without the use of control packets, saving energy. The protocol has five operations, from which, rotation of cluster-heads and data gathering offer better energy balance and do not compromise network performance, and the clustering scheme, that employs different transmission powers and creates unequal sized clusters efficiently and in balanced quantity.

We generalized the RRUCR protocol to a cluster-based energy management architecture that considers Hot Spot issues. The architecture consists of four basic modules: an intra-cluster energy manager, inter-cluster energy manager, data gathering and route management. Since it is based on clusters, energy management occurs essentially by their control, formation and maintenance. Our architecture innovates in relation to existing energy management architectures by considering together energy and performance.

Simulation results showed that RRUCR increased the network lifetime by around 21.36% in relation to UCR. Moreover, the number of created clusters was 35.82% lower than UCR, spending less energy on cluster-head rotations. The RRUCR resilience was also evaluated and, although the routes maintenance of RRUCR is simple, it showed efficacy when compared to the UCR, keeping an acceptable level of network performance. Future work includes operations that check the integrity on WSN links, carrying out more complex repairs. We are now working on a TinyOS implementation of RRUCR.

References

[1] N. Bulusu and S. Jha, *Wireless sensor networks - A system perspective*. Artech House, 2005.

[2] Y. Bi, N. Li, and L. Sun, "Dar: An energy-balanced data-gathering scheme for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14-15, pp. 2812–2825, 2007.

[3] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537 – 568, 2009.

[4] W. Jin, I. J. de Dieu, A. D. L. D. Jose, S. Lee, and Y.-K. Lee, "Prolonging the lifetime of wireless sensor networks via hotspot analysis," in *International Workshop on Computing Technologies and Business Strategies for u-Healthcare(CTBuH 2010)*, Seoul, Korea, July 2010.

[5] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wirel. Netw.*, vol. 15, no. 2, pp. 193–207, 2009.

[6] B. Xi-Rong, Q. Zhi-Tao, Z. Xue-Feng, and Z. Shi, "An efficient energy cluster-based routing protocol for wireless sensor networks," in *CCDC'09: Proceedings of the 21st annual international conference on Chinese control and decision conference*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4752–4757.

[7] A. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is rssi a reliable parameter in sensor localization algorithms: An experimental study," in *Field Failure Data Analysis Workshop (F2DA'09)*, 2009.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in *Proceedings of the Hawaii International Conference on Systems Sciences*, January 2000.

[9] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366–379, 2004.

[10] M. Ye, C. Li, G. Chen, J. Wu, and M. Y. E. Al, "Eecs: An energy efficient clustering scheme in wireless sensor networks," in *In: Proc. of the IEEE Int'l Performance Computing and Communications Conf.* IEEE Press, 2005, pp. 535–540.

[11] M. Perillo, Z. Cheng, and W. Heinzelman, "An analysis of strategies for mitigating the sensor network hot spot problem," in *in: Proceedings of the Second International Conference on Mobile and Ubiquitous Systems*, 2005, pp. 474–478.

[12] K. Thanigaivelu and K. Murugan, "Impact of sink mobility on network performance in wireless sensor networks," *International Conference on Networks and Communications*, vol. 0, pp. 7–11, 2009.

[13] N. Vlajic and D. Stevanovic, "Sink mobility in wireless sensor networks: a (mis)match between theory and practice," in *IWCMC*, 2009, pp. 386–393.

[14] S. Sesay, J. Xiang, J. He, Z. Yang, and W. Cheng, "Hotspot mitigation with measured node throughput in mobile ad hoc networks," in *the 6th International Conference on ITS Telecommunications (ITST 2006)*, L. Li, Ed. IEEE, 2006, pp. 749–752.

[15] A.-F. Liu, M. Ma, Z.-G. Chen, and W. hua Gui, "Energy-hole avoidance routing algorithm for wsn," *International Conference on Natural Computation*, vol. 1, pp. 76–80, 2008.

[16] X. Wu and G. Chen, "On the energy hole problem of nonuniform node distribution in wireless sensor networks," in *in Proc. The IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS*, 2006, pp. 180–187.

[17] S. Lee, H. Choe, B. Park, Y. Song, and C.-k. Kim, "Luca: An energy-efficient unequal clustering algorithm using location information for wireless sensor networks," *Wireless Personal Communications*, pp. 1–17, 2009, 10.1007/s11277-009-9842-9.

[18] J. Yang and D. Zhang, "An energy-balancing unequal clustering protocol for wireless sensor networks," 2009. [Online]. Available: http://www.scialert.net/pdfs/itj/2009/57-63.pdf

[19] S. Gamwarige and C. Kulasekere, "Optimization of cluster head rotation in energy constrained wireless sensor networks," *Wireless and Optical Communications Networks, 2007. WOCN'07. IFIP International Conference on*, pp. 1–5, 2007.

[20] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, and S. Shenker, "An architecture for energy management in wireless sensor networks," *SIGBED Rev.*, vol. 4, no. 3, pp. 31–36, 2007.

[21] L. Ruiz, J. Nogueira, and A. Loureiro, "Manna: a management architecture for wireless sensor networks," *Communications Magazine, IEEE*, vol. 41, no. 2, pp. 116 – 125, feb. 2003.