# A Cluster-based and On-demand routing algorithm for Large-Scale Multi-hop Wireless Sensor Networks

Natale Guzzo, Nathalie Mitton, Pascal Daragon, Arulnambi Nandagoban

# A Cluster-based and On-demand routing algorithm for Large-Scale Multi-hop Wireless Sensor Networks

Natale Guzzo[1,2*], Nathalie Mitton[2], Pascal Daragon[1], and Arulnambi Nandagoban[1]

[1] TRAXENS SAS, Marseille, France
[2] Inria, Villeneuve d'Ascq, France
natale.guzzo@inria.fr

**Abstract.** Reducing the energy consumption and improving the robustness of a Wireless Sensor Network (WSN) are the main requirements for many industrial and research applications. The sensors usually use a routing protocol in order to deliver the sensing data to a Base Station (BS) which may be far away from the monitoring area. Many algorithms proposed in the literature compute the routing process by clustering the network and by designing new election mechanisms in which the cluster-heads are chosen taking account of the remaining energy, the communication cost and the density of nodes. However, they do not consider the connectivity to the BS, and assume that all the nodes or only few prefixed nodes are able to directly communicate with it. We believe that this assumption is not suitable for many applications of WSN and to tackle this problem we propose CESAR, a multi-hop and energy-efficient routing protocol for large-scale WSN which includes a new cluster-head selection mechanism aware of the battery level and the connectivity to the BS. Furthermore, our solution employs an innovative hybrid approach to combine both clustering and on-demand techniques in order to provide an adaptive behavior for different dynamic topologies. Simulation results show that our solution outperforms in terms of energy consumption and data delivery other known routing algorithms in the literature.

**Key words:** Wireless Sensor Networks (WSN), distributed clustering, multi-hop routing protocol, on-demand scheme, Base Station (BS) connectivity, energy-efficiency

* corresponding author

## 1 Introduction

The Wireless Sensor Networks (WSN) are often composed of a large number of sensors that collaborate in order to transmit the sensing data to the Base Station (BS) by satisfying some requirements such as coverage, robustness, scalability and lifetime. Several solutions concerning the physical, MAC and network layers have been proposed in the literature. Regarding the routing protocols, the clustering techniques are employed to reduce the message overhead, the overhearing and the interferences between the nodes in the network. The benefits introduced by this approach lead to a high scalability, simple routing decisions, and low energy dissipation by reducing the data traffic and the overhead caused by the flow of routing information.

In this paper we present the Cluster-based Energy Saving Affiliation Routing protocol (CESAR) which is a new multi-hop and energy-efficient algorithm that aims to reduce the energy consumption in WSN by introducing a scheme with innovative features. The main aspects considered in the design of the routing scheme are the energy consumed by the nodes and the data delivery, since our objectives are to create a robust and scalable network and extend its lifetime as long as possible, while fulfilling application requirements. We will show in the next sections that our clustering scheme outperforms the other simulated algorithms.

We survey the related work in Section II. The main features of CESAR are presented in Section III. In Section IV CESAR is employed in different simulated scenarios inspired by industrial use case applications and we analyze the results in comparison with other routing algorithms. Finally, Section V gives the concluding remarks and reports the direction for future works.

## 2 Background and related work

Several cluster-based protocols have been proposed in the literature in the last few years to reduce the energy consumption and prolong the network lifetime in WSNs. They can be classified according to the goals and the approaches employed for cluster formation and cluster-head selection. The main distinction is related to the cluster formation mechanism. In this sense, we can distinguish centralized algorithms such as PEGASIS [1] and CDC [2], and distributed algorithms like LEACH [3], HEED [4], and DSBCV [5]. Other schemes are based on Geographical clustering as RCHR [6] and TTDD [7], on the concentric clustering such as CCS [8], or on the use of specific cluster-head election techniques like BLAC [9]. The algorithm named SECC selects the nodes to add to the clusters according to their energy or distance from the cluster-head [10]. In this section we describe in brief some distributed clustering schemes by listing their features and limitations.

Nevertheless, the most of the algorithms studied in the literature suppose that all the nodes can always directly communicate with the BS. We believe that this assumption is quite restrictive and not suitable for many applications

of WSNs in which the sensors may not be able to connect to the BS because of the excessive distance or the bad radio environment. Moreover, some schemes such as LEACH and HEED need the synchronization between the nodes in order to start the clustering process at the same time. If the nodes are not synchronized the performances of the two algorithms in terms of energy consumption and data delivery are significantly degraded, since the cluster-head selection cannot be well performed. The solution presented in this paper does not adopt such an assumption, since it considers the connectivity to the BS as one parameter to use in the cluster-head selection scheme.

## 3 CESAR algorithm

Our solution is addressed to specific applications where the sensors are equipped with a long-range radio module to communicate on the link towards the BS (BS-link) and a short-range radio module for the communication peer-to-peer with the other sensor nodes (P2P-link). The nodes do not know the position of the BS and they are supposed to communicate directly with it only whether the received signal strength on the BS-link is high enough. Moreover, depending on the radio environment and the distance from the BS, the nodes do not detect the same quality on the BS-link. As a result, only some of them may be able to connect to the BS and they may need to use different transmission powers in order to deliver their data. Such nodes may also change over time depending on the variations of the BS-link quality. In terms of energy consumption, the transmission of data on the BS-link is much more expensive than the transmission of the same amount of data on the P2P-link. In fact, we suppose that the communication on the BS-link requires a transmission power 10 to 20 times higher than the communication on the P2P-link and introduces a connection delay lasting approximately one minute.

We focus on sensor networks where the nodes are positioned at the edges of a 3-dimensional grid. We believe that this model well describes different kinds of scenarios such as the monitoring of goods in storage areas. We believe that CESAR is also suitable in applications where the nodes move within the sensing area. Vehicle tracking in a town, surveillance in an airport, and fauna monitoring are some examples in which CESAR can be employed.
The most significant features of CESAR are the following:

- CESAR is a hybrid algorithm that combines clustering and on-demand approaches in order to dynamically adapt the behavior of the nodes to the topology changes. For this purpose, it does not involve every node in a cluster without preventing it to send data to the BS.
- The proposed cluster-head selection mechanism is aware of the connectivity to the BS and the battery level: only the nodes that have the signal strength on the BS-link and the battery level greater than prefixed thresholds can become cluster-heads.

- A recovery scheme is employed to keep alive the routes between the nodes and the cluster-heads with low cost operations.
- A data aggregation scheme can be employed by the members of the clusters before to send data packet to the cluster-head

In the next paragraphs we explain in more details the cluster-head selection mechanism and the cluster formation scheme. Before that, we need to introduce the types of nodes and routing messages used in different processes. We defined four types of nodes:

- HEAD: nodes which are able to communicate with the BS and are in charge of creating and managing a cluster by sending announcement messages.
- MEMBER: nodes which have joined a cluster after receiving an announcement transmitted by a HEAD node.
- LOOSE: nodes which have not joined any cluster and are not able to transmit data to the BS.

The routing messages were defined as follows:

- RANN: hop-bounded broadcast messages used by HEAD nodes to build their clusters.
- REQR: hop-bounded broadcast messages used by LOOSE nodes to discover a nearby cluster.
- REP: unicast messages used to reply to REQR messages.
- ROFF: broadcast messages used by HEAD nodes to destroy their clusters.
- DATA: unicast messages used by MEMBER nodes to deliver their data.
- ACK: unicast messages used by HEAD nodes to acknowledge DATA packets.

### 3.1 The cluster-head selection mechanism

Every node in the network periodically executes the cluster-head selection process to check whether they have the requirements to become HEAD nodes. As already explained, a node can become a cluster-head only if both the battery level and the signal strength on the BS-link are greater than a prefixed threshold. However, these are not the only conditions to become HEAD. There may be other cluster-heads nearby that have better conditions than the considered node. In this case, the latter should become member of one of such clusters, rather than becoming HEAD itself.
This decision process is made by calculating the HEAD metric defined as cost and it takes account of the battery level and the quality of the BS-link. Therefore, the higher the metric, the less likely a node to be elected as cluster-head. As we can see from Algorithm 1, if node $u$ is MEMBER and has the requirements to become HEAD, it checks whether the metric of its cluster-head is 1.5 times greater than its own metric, as shown at line 5. In that case, $u$ becomes HEAD, otherwise it remains a MEMBER. Such an approach is introduced in order to avoid the election of several HEAD nodes close to each other, and reduce the overall number of connections to the BS and, thus the global energy consumption.

**Algorithm 1** CESAR (Cluster-head selection algorithm run on node $u$)

**Parameters:**

- connectivity: signal strength measured on the BS-link
- battery: remaining energy
- conn_thr1, conn_thr2: thresholds for the connectivity parameter
- batt_thr: threshold for the battery parameter

```
1: if ( connectivity(u) > conn_thr1 AND battery(u) > batt_thr ) then
2:    if (u is HEAD) then
3:       Send RANN;
4:    else if (u is MEMBER) then
5:       if (metric(head of u) > 1.5 * metric(u) then
6:          Become HEAD;
7:          Send RANN;
8:       end if
9:    else if (u is LOOSE) then
10:      Become HEAD;
11:      Send RANN;
12:   end if
13: else if ( connectivity(u) > conn_thr2 AND battery(u) > batt_thr ) then
14:   if (u is HEAD) then
15:      Send RANN;
16:   end if
17: else if ( connectivity(u) < conn_thr2 OR battery(u) < batt_thr ) then
18:   if (u is HEAD) then
19:      Become LOOSE;
20:      Send ROFF to destroy the cluster;
21:   end if
22: end if
```

We defined a second threshold for the signal strength on the BS-link to avoid frequent cluster-head elections and resignations when such a parameter continuously varies around the main threshold. Regarding the resignation of a HEAD node, it may occur in two cases. In the first case, the node has no longer the requirements to be a cluster-head, since it has not enough energy or it is no longer able to directly communicate with the BS. The second case occurs when the considered node receives an announcement from a new HEAD in the neighborhood that has a metric lower than its, as we explain in the next paragraph. When a node resigns as HEAD, it destroys its cluster and, in the second case, it becomes MEMBER of another cluster.

### 3.2 Cluster formation and destruction

The cluster-heads form the clusters in the network and periodically connect to the BS to deliver the sensing data received from other nodes. The clusters are built by broadcasting a bounded-hop RANN in which the header field named Time To Live (TTL) is set to the number of hops that can be traversed by the

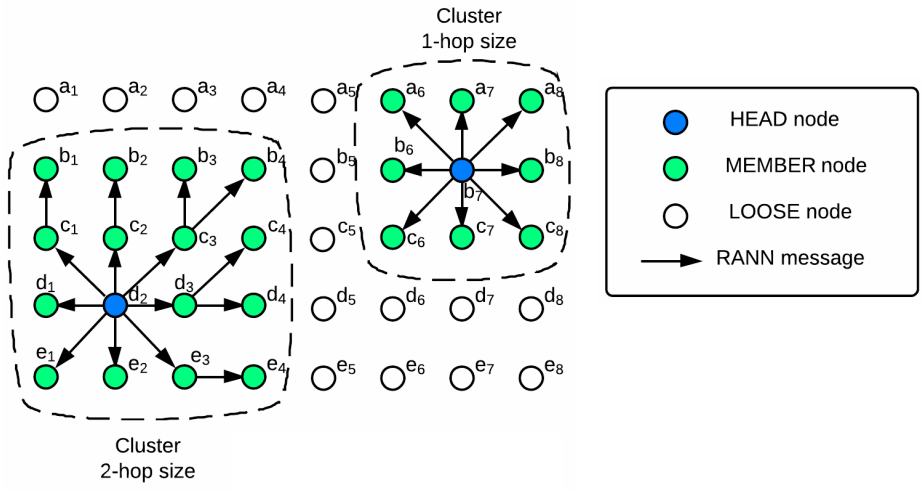


**Fig. 1.** Cluster formation. The RANN are broadcast by the HEAD nodes to form clusters of different sizes.

message before to be discarded. In this way, only the nodes that are at a limited hop-distance from the considered HEAD are recruited into the cluster.

The RANN messages are periodically broadcast by the HEAD nodes in their cluster in order to keep the MEMBER nodes up-to-date about their metric.

If a non-clustered node receives a RANN message from a HEAD node, it immediately becomes MEMBER of the announced cluster. However, if the considered node receives multiple RANN messages from different cluster-heads, it then compares their metrics and joins the cluster with the lowest one. In the case a HEAD node receives a RANN message, it checks whether its metric is 1.5 times lower than the metric of the announced HEAD. If the latter condition holds, the considered node resigns as HEAD, destroys its cluster and joins the cluster of the HEAD announced in the RANN message. Such a condition has been defined in order to avoid frequent cluster destruction when the HEAD nodes are close to each other. After joining a cluster, the MEMBER nodes store in the routing table the last hop traversed by the received RANN in order to have a route to the HEAD. As a result, a routing tree is formed into the clusters between the MEMBER nodes and the cluster-head.

The size of the clusters is decided by the HEAD nodes according to their operating parameters. In this way, CESAR aims to concentrate the collection of the sensing data in the HEAD nodes with better conditions in order to smartly balance the energy consumption between them.

When a HEAD node resigns, it destroys its cluster by sending a ROFF message to all the members that leave the cluster and become LOOSE nodes upon the message reception.

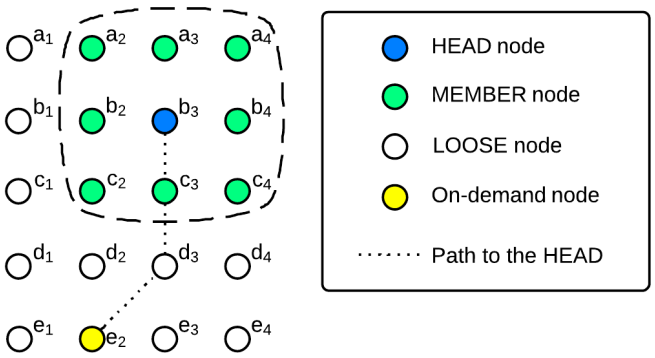


**Fig. 2.** On-demand scheme. The non-clustered nodes employ REQR messages to discover a cluster in the neighborhood and find a route a HEAD node.

### 3.3 The on-demand scheme

As already explained, the LOOSE nodes do not have any routes to any cluster-head, so they are not able to transmit data to the BS. Thus, when one of them has some data to deliver to the server, it starts a special procedure that aims to discover a cluster nearby.The discovering procedure consists of broadcasting a hop-bounded REQR message that will be received by the nodes in the neighborhood. When a MEMBER node receives such a message, it replies with a REP message containing information about the status of the HEAD of its cluster. The discovery process continues until a cluster is found or the max hop distance is reached. In the latter case, the process is reinitialized and restarted after a certain time interval.

For instance, let's assume that the node $e_2$ in Figure 2 needs to send the sensing data to the BS. It broadcasts a REQR message in which the TTL is set to TTL_MAX. The nodes $d_1$, $d_2$, $d_3$, $e_1$ and $e_3$ are not MEMBER nodes, so they do not reply to the request. Nodes $c_1$, $c_2$, $c_3$, members of the cluster created by the node $b_3$ receive the REQR and reply with a REP message that reports the metric of their cluster-head. Such a message passes through the nodes traversed by the latest REQR message. In the figure, we suppose that the REP messages are received by the node $d_3$ and forwards the message to the requesting node $e_2$. The latter stores in the routing table the next hop on the path towards to the reported cluster and records the metric of its HEAD. After that, it can deliver the collected data to the HEAD node $b_3$ by passing through the nodes $d_3$ and $c_3$.

If there are more than one cluster in the nearby, the considered node may receive more than one REP message, and then it should evaluate to which cluster to deliver its data by comparing the metrics of the respective cluster-heads.

### 3.4 Data aggregation

The aggregation of the sensing data allows the reduction of the time needed for transmissions and receptions decrease, and thus, the energy consumption and the risks of collisions decrease as well. The data aggregation in CESAR can be performed in three different ways. The least strong approach is the aggregation of data only at the cluster-heads before deliver it to the BS. A stronger approach is to aggregate data also at the member nodes at the borders of the clusters. In this case, such nodes collect the data coming from the LOOSE nodes and aggregate it before transmitting to the cluster-head. Finally, the strongest way is to perform the data aggregation at the cluster-heads and at every member of the clusters. However, in this paper we evaluate CESAR without performing any aggregation of data.

### 3.5 Routing maintenance and recovery

The CESAR algorithm uses ACK messages to acknowledge every data packet. Thus, all the nodes expect to receive an ACK message as confirmation that the transmission succeeded and the data was received by the cluster-head. If no ACK messages are received after a prefixed timeout, then the sender node considers the packet lost and transmits it again to the same HEAD. If the transmission fails 3 times, then the considered node becomes LOOSE and restarts the on-demand scheme to find another route to the same HEAD or to another cluster. Thus, we set a specific timeout in the MEMBER nodes to check if the RANN messages are periodically received. Such a timer is restarted at every reception of RANN messages from the cluster-head and if the timeout occurs, then the node leaves the cluster and becomes LOOSE.

## 4 Performance evaluation

In this section we simulate the CESAR algorithm in WSNET [11] and we analyze the obtained results in order to compare its performances with two of the most popular clustering protocols studied in the literature: LEACH [3] and HEED [4]. We consider a network in which the nodes are positioned in a 3d grid with dimensions 160x160x110 $m^3$ and for each experiment we vary the number of sensors in the same area. Thus, the density and the distance between the nodes change at each scenario. In such a way, we analyze the behavior of the algorithm in low-density and high-density networks.

Since we want to use a realistic model for the energy consumption, we consider the transmission and the reception powers reported in the datasheets of XBee and XBee-PRO DigiMesh 2.4 provided by Digi International [12]. We considered two different devices since CESAR uses a multi-hop approach and, therefore it needs a lower transmission power than LEACH and HEED which are single-hop routing protocols.

CESAR has some critical parameters that should be configured in order to evaluate its performances. Such parameters are set to optimized values obtained from long series of simulation as reported in Table 1.

**Table 1.** Critical parameters of CESAR

| | |
|---|---|
| Connectivity threshold 1 | 40% |
| Connectivity threshold 2 | 30% |
| Battery threshold | 20% |
| Status update frequency | every 2 min |
| RANN frequency | every 1 min |
| Maximum cluster size | 5 hops |
| Data delivery frequency | every 10 min |

Regarding the other simulation parameters we set to 6 hours the virtual duration for each experiment and we suppose to receive from the application layer 4096 Bytes of sensing data every 10 minutes. The energy available at each sensor at the beginning of the simulations is 100 Joules. We believe that the latter values are fair enough to simulate a generic application running on sensors with a generic hardware.

### 4.1 Data delivery

In many applications in which the sensor networks are employed, the robustness is a primary requirement to ensure the delivery of the sensing data to the BS. In this section, we measure the percentage of data generated by the sensors that is successfully delivered to the BS, with different experiments in which we change the number of nodes in the sensing area.

As we can observe in Figure 3, the LEACH and HEED algorithms are less robust than CESAR, which uses dedicated mechanisms to recover and maintain the routes between the nodes as explained in Section III. The losses experienced by the first two algorithms are mainly caused by the interferences between the cluster-heads in the set-up phase, in which some messages related to the cluster formation may be lost and ,thus, some nodes are not able to join any cluster. Moreover the LEACH algorithm experiences more data losses because of its cluster-head selection mechanism which, unlike HEED and CESAR, does not take account of the distribution of the cluster-heads into the network.

### 4.2 Energy consumption

The energy consumed by each node is measured by performing different tests in which we increase the number of the nodes in the sensing area.

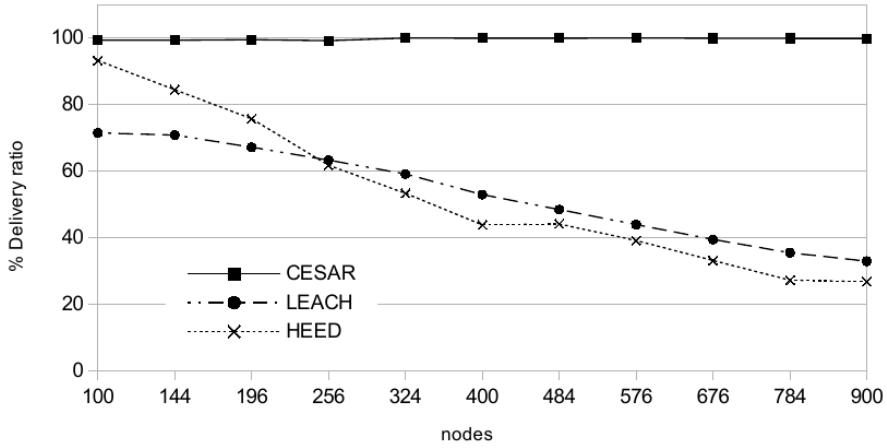


**Fig. 3.** Data delivery.

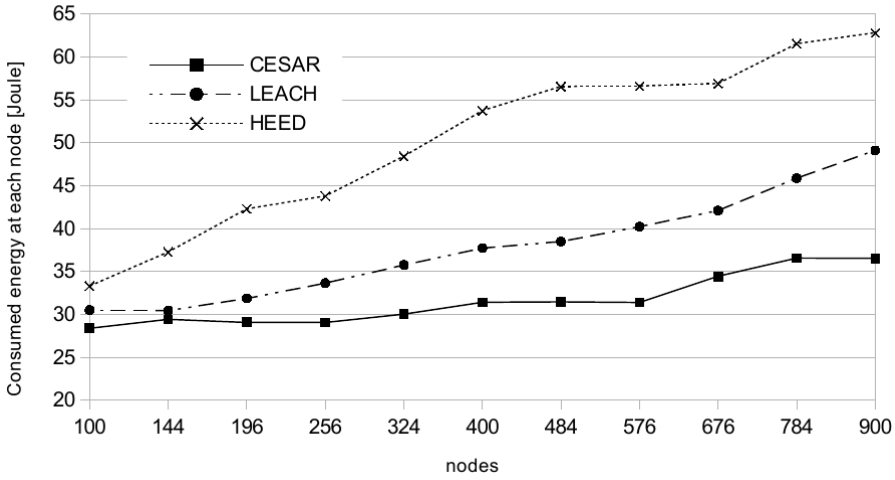


**Fig. 4.** Energy consumption at each node.

As we can see from the results shown in Figure 4, CESAR is less expensive in terms of energy and more scalable than LEACH and HEED, thanks to the mechanisms described in Section III.

The graphs show that CESAR is also more scalable than LEACH and HEED which suffer of interferences and high overhead for the assignation of the time-slots by the cluster-heads to the members of the cluster.

## 5 Conclusions

The algorithm described in this paper combines the clustering and the on-demand approaches in order to reduce the energy consumption in the whole network. A new cluster-head selection mechanism is proposed in order to consider the situations in which the nodes have limited battery capacity and not all of them are capable to communicate with the BS to which deliver the own sensing data. The further mechanisms employed for the reduction of the energy consumption and improvement of the robustness make CESAR suitable for many low data rate applications of WSNs. The simulations described in Section IV show that CESAR performs better than some popular algorithms such as LEACH and HEED, and the experiments performed on the FIT-IoT-lab [13] platform show that our solution well perform also in real sensor networks. Thanks to the on-demand mechanism and the adaptability of the cluster size, we believe that CESAR is also suitable for networks composed of mobile sensors. A new metric can be investigated to take account of the mobility of the nodes and adapt the above-mentioned features to the dynamic of the network topology. Regarding future developments, CESAR will be tested in combination with some Low Power Listening (LPL) protocols , such as X-MAC [14] and LA-MAC [15], in order to figure out which one is the best choice to minimize the energy consumption and ensure the robustness of the network.

## 6 Acknowledgements

This work was partially supported by a grant from CPER Nord-Pas-de-Calais /FEDER Campus Intelligence Ambiante.

## References

1. S. Lindsey and C. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, 2002, pp. 3–1125–3–1130 vol.3.
2. F. Bajaber and I. Awan, "Centralized dynamic clustering for wireless sensor network," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, 2009, pp. 193–198.
3. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, pp. 660–670, 2002.
4. O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, 2004, pp. –640.
5. Y. Liao, H. Qi, and W. Li, "Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks," *Sensors Journal, IEEE*, vol. 13, no. 5, pp. 1498–1506, 2013.

6. T. Gao and R. Jin, "A regional centralized-clustering routing algorithm for wireless sensor networks," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, 2008, pp. 1–4.
7. H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "Ttdd: two-tier data dissemination in large-scale wireless sensor networks," *Wirel. Netw.*, vol. 11, no. 1-2, pp. 161–175, Jan. 2005.
8. S.-M. Jung, Y.-J. Han, and T.-M. Chung, "The concentric clustering scheme for efficient energy consumption in the pegasis," in *Advanced Communication Technology, The 9th International Conference on*, vol. 1, 2007, pp. 260–265.
9. T. Ducrocq, N. Mitton, and M. Hauspie, "Energy-based Clustering for Wireless Sensor Network Lifetime Optimization," in *WCNC - Wireless Communications and Networking Conference - 2013*, Shanghai, Chine, Apr. 2013.
10. M. Bala Krishna and M. N. Doja, "Self-organized energy conscious clustering protocol for wireless sensor networks," in *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, 2012.
11. "http://wsnet.gforge.inria.fr/."
12. "http://www.digi.com/xbee/."
13. C. Burin des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. NoÃľl, and J. Vandaele, "Using senslab as a first class scientific tool for large scale wireless sensor network experiments," in *NETWORKING 2011*, ser. Lecture Notes in Computer Science, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Springer Berlin Heidelberg, 2011, vol. 6640, pp. 147–159.
14. M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, ser. SenSys '06, 2006, pp. 307–320.
15. G. Corbellini, E. Strinati, and A. Duda, "La-mac: Low-latency asynchronous mac for wireless sensor networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, 2012, pp. 380–386.