LECOMP: Low Energy COnsumption Mesh Protocol in WSN

Juan A. Ortega, Alejandro Fernandez-Montes, Daniel Fuentes,
and Luis Gonzalez-Abril

Abstract. LECOMP is a mesh network protocol that is focused on minimizing battery consumption of nodes in a Wireless Sensor Network (WSN). This paper is a improvement of a first approach [1] which proposes a method to extend the lifetime and the reliability in a WSN balancing transmissions saving energy throughout different nodes in order to consume homogeneously nodes' batteries, and taking into account distances and hops of routes. First, a central node analyzes messages from nodes during a training period to determine new routing rules for each sensor. Second, the server configures the network adding the computed rules to the nodes. Third, central server can reconfigure the network routes when new balance of load is needed.

1 Introduction

Since sensors are battery powered in Wireless Sensor Networks applications, it is essential to minimize energy consumption so lifetime of the global network can be extended to reasonable times. The establishment of WSNs have grown in both research and real-world applications so improvements and solutions are required. Nowadays, energy-efficient sensor networks is a common topic in research works and the main directions to energy conservation are discussed. Some studies deal with the topology control which refer to the techniques that are aimed to super-imposing a hierarchy on the network organization to reduce the consumption [2]. In this sense, the location and the distance between the nodes are critical aspects. Some studies determines that multi-hop communication in mesh networks consumes less power than the traditional single-hop long distance radio communication in star networks

Juan A. Ortega · Alejandro Fernandez-Montes · Daniel Fuentes
Languages and Computer Systems Department, University of Seville, Seville, Spain
e-mail: {jortega,afdez,dfuentes}@us.es

Luis Gonzalez-Abril
Applied Economics I Dept, University of Seville, Seville, Spain
e-mail: luisgon@us.es

[3]. Other studies propose an efficiency power management in the network using sleep/wakeup protocols at the application or network layers [4, 5], or strictly integrated MAC [6] protocols. There are multiple application environments of energy-efficient wireless sensor networks and different routing techniques for them [7, 8]. Others methods save energy by reducing the number of sensor transmissions. In [9], sensor transmissions are sorted according to the magnitude of their measurements, and the sensors with small magnitude measurements, less than a threshold, do not transmit. However, in [10] the data compression allows the reduction of the data length. On the other hand, battery analysis is another approach to balance the transmissions and maximize lifetime of the network[11]. In this paper some of the described techniques are combined to achieve a energy-efficient communication in a mesh WSN. Specifically, the batteries levels, sensor distances and message routes are analyzed to balance load and to optimize the routes from nodes to the central station. In Section 2 we provide an overview of the objectives then, in Section 3, we describe our energy-efficient routing proposal LECOMP. Implementation and test results are related in Section 4 and finally, concluding remarks are made in Section 5.

2 Routing Proposal

The primary goal of LECOMP protocol for WSN is to increase the lifetime of the network saving batteries. Moreover, secondary goals are: decreasing the load of the netwrok by minimizing configuration messages needed to setup sensors routing constraints in an automated and live way, and on the other hand, to free the administrator from manually setup the network. The architecture of our sensor network is illustrated in figure . We consider next elements:

- A set of n motes or sensors, $S_1, S_2, ..., S_n$. Devices that sense on aspects of the physical world like the humidity or temperature levels. These sensors are typically battery powered and this information can be spread throughout the network.
- The central server (aka end node), H that stores aspects of the mote network state and acts as a proxy to communicate between mote network and client applications.
- A set of client applications, A that interfaces with the host server to communicate with nodes, to manage existing applications or deploy new ones.

We assume that sensors initially interact each other through a standard mesh network that tries to maximize the reliability of the network broadcasting packets to all nodes in order to reach the end node.

A message from a sensor can be captured by all sensors that are into its scope. The server can communicate with sensors within its scope, but client applications only communicate with the server. The size of a message is limited but it can be configured and modified to include or exclude information.

The main goal of this routing proposal is the optimization of the number of messages copies traveling through the network and the improvements of routes in order

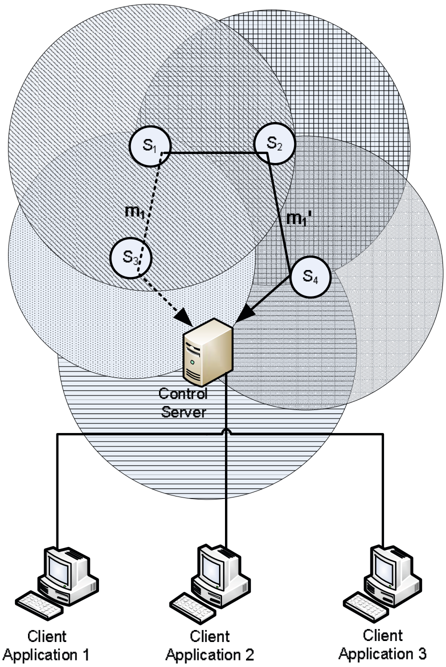

Fig. 1 Representation of typical WSN. A message is sent from S_1 node to H server.

to save batteries in the nodes by a previous training period. This way the H server computes rules for each sensor to determine when is necessary to forward a received message.

When the training-time begins, all the sensors start sending messages to the server. In every message, the description of the route (nodes list, where a message has been routed) is attached to the message. When a message arrives to the H server, the message contains the whole route from the origin sensor to the server H. With this information, the H server can establish some rules to improve the communication between the nodes. This rules are computed by an implementation of the A* algorithm for pathfinding. A* algorithm [12] uses a distance-plus-cost heuristic function $f(x)$ which is a sum of two functions:

- $g(x)$ The path cost function. We assume this function returns the standardized weight (power in watts needed) of the edge between two nodes. This weight is the quantity of power needed to make the transmission between two nodes.
- $h(x)$ The heuristic function. Our solution uses the minus geometric mean of battery levels (in [0-1]) of current path. This way we are taking into account battery levels of a path in order to benefit paths where battery levels of nodes are higher. Notice that geometric mean is used to detect paths where at least one of the nodes has a very low battery level.

Comparing with Dijkstra or others algorithms for pathfinding, A* includes both cost function and heuristic function so it mixes depth-first search and breadth-first search. A* is equals to Dijkstra algorithm if $h(x) = 0$. For example, let's consider two routes R_1 and R_2 of two copies m_1 and $m2$ of the same message respectively:

$$R_1 = [S_1, S_2, S_4, H]$$

$$R_2 = [S_1, S_3, S_4, H]$$

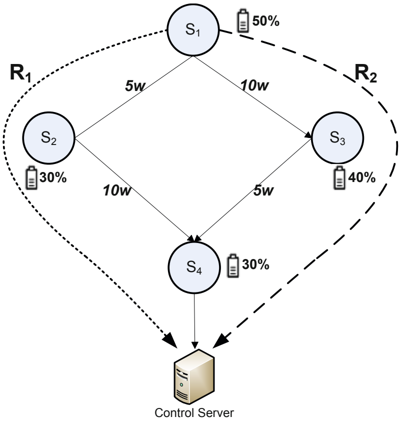

Fig. 2 Case study for 4 nodes and two possible paths

Nodes, routes, costs and battery levels are shown in figure 2. Once the server receives paths, costs and battery levels computes best paths. In the situation shown route R_1 final cost is computed bellow:

$$f(1) = g(1) + h(1) = 0 - 0.5 = -0.5$$
$$f(2) = g(2) + h(2) = 0.5 - \sqrt[2]{0.5 \cdot 0.3} = 0.11$$
$$f(4) = g(4) + h(4) = 1 - \sqrt[3]{0.5 \cdot 0.3 \cdot 0.3} = 0.64$$
$$Cost(R_1) = f(1) + f(2) + f(4) = -0.5 + 0.11 + 0.64 = \mathbf{0.25}$$

And for route R_2:

$$f(1) = g(1) + h(1) = 0 - 0.5 = -0.5$$
$$f(3) = g(3) + h(3) = 1 - \sqrt[2]{0.5 \cdot 0.4} = 0.56$$
$$f(4) = g(4) + h(4) = 0.5 - \sqrt[3]{0.5 \cdot 0.4 \cdot 0.3} = 0.1$$
$$Cost(R_2) = f(1) + f(3) + f(4) = -0.5 + 0.56 + 0.1 = \mathbf{0.16}$$

With these paths, the server can detect that it is not necessary that the S_3 sensor forwards the message m_2 (which is a copy of m_1). Hence, the rule *"Don't forward a message if the previous nodes sequence is* $[S_1]$*"* will be established for the S_3 sensor when the server reconfigure the network. In this manner, the server H continues receiving messages and computing new rules until a time threshold is exceeded. In this case, the nodes stop sending messages to the server and the reconfiguration of the nodes starts. In this stage, the server establishes the computed rules to the nodes and the network starts using this new configuration.

This way we can achieve energy saving by three actions:

- reducing message hops and avoiding copies.
- prioritizing the route where costs are lower.
- balancing load to maximize general battery levels.

Obviously, during the training time, the decrease of the energy consumption is null. However, after this period, the key limitations of WSNs, the storage, power and processing, are treated. The server reconfigures the nodes with the calculated rules and the communication in the network. With this optimization, in the last example the message m_2 will not arrive to server. Hence, duplicated messages in the network are avoided and consequently, the information load in buffers, the time processing and the battery consumption in sensors are reduced and, consequently, the lifetime of the network is increased. Furthermore, by deleting data redundancy and minimizing the number of data transmissions, there is less data load in the network, the possibilities of an overload are lower and the message management is simplified.

```
// Called when the Dispatcher receives a message for this
   protocol.
begin public void stackReceive [Receiver rcvr]
    // Gets the message
    RoutedMsg msg=(RoutedMsg)rcvr.getData();
    // Check hasn't been received already and route is
       allowed
    if (!msg.equals(prevMessage) &&
    !msg.currentRoute.equals(restrictedRoute)) then
        // Adds current sensorId to the route
        msg.currentRoute.addElement(id);
        // Not-relevant operations
        TestReceiver meta=new TestReceiver();
        meta.addMetadata(rcvr);
        meta.data = msg.data;
        // This will forward the received data
        ds.dispatch(meta);
        prevMessage = msg;
    end
end
```

Algorithm 1. Routine running on sensor nodes that checks if message forwarding is needed

As batteries drain the *H* server computes new rules, and periodically, the *H* server reconfigures the routing rules to balance batteries of the whole network. In addition to this reconfiguration if a node stops working due to an attack, the lack of battery or another external factor it will not send any reply. In this case, the *H* server will check affected paths and compute new paths to replace invalid ones.

3 Implementation, Tests and Results

For test purposes we have implemented the routing proposal in Sentilla Work, the IDE supplied with Sentilla Development Kit. The real implementation is an extension of one of the protocols supplied by Sentilla, therefore the algorithm shown it was written in Java.

Results are shown in Table 1 below where the base mesh protocol, the first version of the method called LECOMP-1 [1] and present method LECOMP-2 are compared. Tests have been done at the WSN deployed at Computer and Systems Languages department of the University of Sevilla. It consists on 9 Zigbee sensors deployed in 9 offices. The number of avoided messages in LECOMP-1 and LECOMP-2 is calculated through an approximation between the number the time computed and the duplicated messsages received in the base case. Table 1 shows that in case of LECOMP-1 the sensor network lifetime is 421h and for LECOMP-2 439h. Compared to 360h when no LECOMP is applied this is an improvement of approximately 17% in case LECOMP-1 and 22% of LECOMP-2 in terms of sensor network lifetime. That implies that when LECOMP-1 is used the WSN runs 2,5 days longer and in case of LECOMP-2 over 3 days longer than when no intelligent routing applied. Moreover, we would like to emphasize the improvement of about 7% in avoided messages and 10% in avoided steps in this proposal comparing with the first version of the method. In figure 3, the battery life and the (no-duplicated) received messages

Table 1 Base-Protocol, LECOMP-1 and LECOMP-2 comparison

	Base	LECOMP-1	LECOMP-2
#sensors	9	9	9
timeComputed	360h.	421h.	439h.
#msgsPerHour	30	29.5	29.7
#msgs	10800	12425	13044
trainingTime	0h.	1h.	1h.
#duplicatedmsgs	1944	5	5
avgStepsPerMsg	2,3	2,02	2,12
#msgsReceived	12744	12430	13049
#msgsAvoided	0	2249	2945
%avoidedMsgs	0	**18,1**	**22,5**
#Steps	29311	25524	27933
#StepsAvoided	0	7368	9020
%avoidedSteps	0	**25,1**	**35,9**

in the three cases are compared. Using our method the number of received messages and the lifetime of the WSN increments.

Figure 3 the battery life and the (no-duplicated) received messages in the three cases is compared. Using our method the number of received messages and the lifetime of the WSN increments.

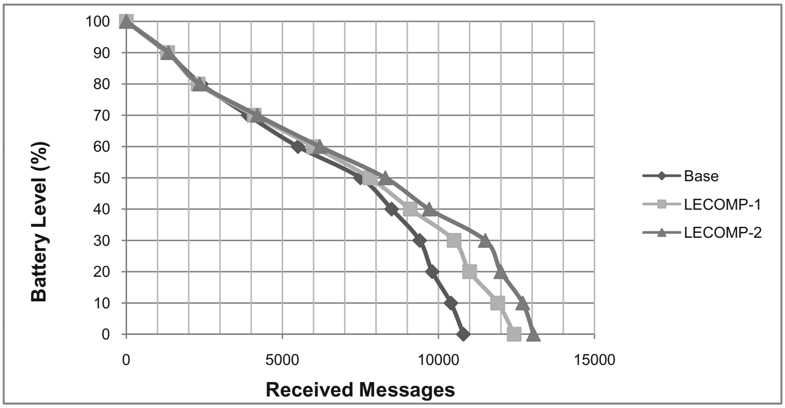

Fig. 3 Comparison Graphic between Base Mesh protocol, LECOMP V1 and LECOMP V2

4 Conclusion

In this paper, we have presented a energy-efficient method for mesh wireless sensor networks. We have got to save up energy by reducing the forwarding of messages. First, the central server determines the rules through the information of the messages routes, the battery levels and the distance between sensors. Second, the network is reconfigured avoiding duplicated messages and balancing the load. The described practical experiences with over Sentilla sensors and Zigbee technology shows an improvement in the WSN lifetime of 22% compared with the base protocol.

Acknowledgements. This research is supported by the MCI I+D project ARTEMISA (TIN2009-14378-C02-01).

References

1. Fuentes, D., Fernandez-Montes, A., Ortega, J.A., Gonzalez-Abril, L.: Energy-Efficient Routing Approach for Wireless Sensor Networks. In: Proceedings of the 4th Symposium of Ubiquitous Computing and Ambient Intelligence UCAmI 2010, pp. 9–12 (2010)
2. Shrestha, A., Xing, L.: A Performance Comparison of Different Topologies for Wireless Sensor Netwoks. In: IEEE Conference on Technologies for Homeland Security 2007, pp. 280–285 (2007)

3. Chandrakasan, A., Min, R., Bhardwaj, M., Cho, S.H., Wang, A.: Power Aware Wireless Microsensor Systems. In: Keynote Paper. 28th European Solid-State Circuits Conference (ESSCIRC), pp. 47–54 (2002)
4. Anastasi, G., Conti, M., Di Francesco, M., Passarella, A.: Energy conservation in wireless sensor networks: A survey. In: Proceedings of the 33rd Hawaii International Conference on System HICSS 2000, pp. 10–20 (2000)
5. Mathur, G., Desnoyers, P., Chukiu, P., Ganesan, D., Shenoy, P.: Ultra-low power data storage for sensor networks. In: Proceedings of the 5th International Conference on Information Processing in Sensor Networks, pp. 374–381 (2005)
6. Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., Chandrakasan, A.: Physical Layer Driven Protocol and Algorithm Design for EnergyEfficient Wireless Sensor Networks. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, pp. 272–287 (2001)
7. Pantazis, N.A., Nikolidakis, S.A., Vergados, D.D.: Energy-Efficient Routing Protocols in Wireless Sensor Networks for Health Communication Systems. In: Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments, vol. 34 (2009)
8. Zeng, K., Ren, K., Lou, W., Moran, P.J.: Energy Aware Efficient Geographic Routing in Lossy Wireless Sensor Networks with Environmental Energy Supply. In: Proceedings of QShine 2006 (2006)
9. Chen, X., Blum, R., Sadler, B.M.: A New Scheme for Energy-efficient Estimation in a Sensor Network. In: Proceedings of the Conference on Information Sciences and Systems, CISS (2009)
10. Marcelloni, F., Vecchio, M.: A simple algorithm for data compression in wireless sensor networks. IEEE Communications Letters 12(6) (2008)
11. Tang, Q., Yang, L., Giannakis, G.B., Qin, T.: Battery Power Efficiency of PPM and FSK in Wireless Sensor Networks. IEEE Trans. on Communications 6(4), 1308–1319 (2007)
12. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 4(2), 100–107 (1968)