An Adjustable Forwarding Policy Exploiting Path Vulnerability in Wireless Sensor Networks

Apostolos Demertzis, Konstantinos Oikonomou
Department of Informatics
Ionian University
Corfu, Greece
Email: {apoldem, okon}@ionio.gr

Ioannis Stavrakakis
National and Kapodistrian University of Athens,
Greece - Universidad Carlos III de Madrid and
IMDEA Networks Institute, Spain
Email: ioannis@di.uoa.gr

***Abstract*—Forwarding data packets in a wireless sensor network is a challenging task due to the energy hole problem that affects the network's operation. Taking into account *path vulnerability*, i.e., a metric based on the transmission distance and the energy left at the nodes' batteries, an *adjustable* policy is proposed here that allows nodes to choose different *parent* nodes for forwarding their data packets towards the sink node. Path vulnerabilities are propagated in the network with reduced overall overhead (e.g., no need to continuously reconstruct routing trees). As it is demonstrated here using simulation results, the proposed forwarding policy increases the time period until the first node runs out of energy when compared to other five similar policies that appear in the literature. Throughput is also increased and the proposed policy's performance is close to the other policies with respect to network lifetime, overhead and latency.**

***Index Terms*—Convergecast; Forwarding; Energy Consumption; Wireless Sensor Networks; Vulnerability.**

I. Introduction

Large scale wireless sensor networks [1] have become a feasible solution for monitoring applications due to recent advances in micro-electronics [2], allowing for hundreds or thousands of small and inexpensive sensor nodes to be randomly spread over large geographical areas. After their deployment, sensor nodes form a wireless network with the purpose of delivering the sensed data to a base station, also known as the *sink* node. The network's size is significantly larger than the communication range, therefore, data have to be relayed to the sink in a multi-hop fashion [3]. The network pattern of collecting data from the entire network to the sink is known as *convergecast* [4], in analogy to broadcast.

Due to the limited capacity of nodes' batteries, energy management is a critical issue in wireless sensor networks [5], [6]. Data transmission from node to node is the dominant factor in energy consumption and therefore, energy-efficiency of convergecast is an extensively researched topic. Usually, convergecast is accomplished by constructing a spanning tree whose root is the sink node [7]. Each node forwards the received data, along with the data generated by itself, to its parent node, thus, the underlying tree is called *forwarding* or *routing* tree [8]. Traditionally, tree construction is based on a shortest-cost algorithm combined with a power-aware metric for the edge cost [9]. The resulting tree consists of the minimum cost paths, where cost of a path is the total sum of the edges' cost along the path. This approach has the advantage of constructing only one tree during a network's lifespan. However, the fixed tree paths result in a much higher energy consumption for nodes that happen to have many descendants, known also as the *energy hole* problem [5].

In order to reduce the effects of this problem, a novel forwarding policy is proposed here that does not necessarily use the same paths during a network's lifespan. Initially, a shortest-cost tree is constructed and all nodes determine their *parent* nodes accordingly. At regular time intervals (*rounds*), nodes re-evaluate the *vulnerability* of possible paths and update their parent nodes. The *vulnerability of a node* depends on its residual energy and the required energy for transmission over the distance between the node itself and its parent node. *Vulnerability of a path* is the *maximum* node vulnerability along the path. Path vulnerabilities are propagated in a distributed manner by traversing each edge of the short-cost tree only once. During each round nodes select their own minimum vulnerability path. In that way data packets follow a path of minimum maximum node vulnerability along the path, thus the proposed policy is actually a *min-max* forwarding policy [10].

The main characteristic of this policy is that there is no need to construct an extensive number of trees for forwarding purposes. Furthermore, the values of path vulnerability from the sink node towards all nodes are propagated in an efficient manner. Simulation results are considered for evaluation purposes and the proposed policy is compared against five other forwarding policies that appear in the literature and are described later in Section II. It is shown that under the proposed policy the time until the first node expires due to depleted battery is increased. On the other hand, the proposed policy is the second best when network lifetime is considered (i.e., a significant number of nodes fail such that the sensor network is not operational anymore). It is also observed that it is close to the particular policy which maximizes network lifetime and significantly larger than the other four policies. The comparison with respect to overhead and latency demonstrates the fact that the proposed policy performs close to the other policies.

In this paper, past related works are presented in Section II

and the required definitions in Section III. Section IV describes the proposed forwarding policy and Section V presents the simulation results. Finally, Section VI concludes the paper.

II. Related Work

Several routing and/or forwarding policies have been proposed over the last two decades for wireless sensor networks [6], [11]. Some are specifically designed for convergecast, e.g. [12], [13], while some are general routing policies that can be easily applied in convergecast, without any or with minor modifications [14], [15]. A straightforward manner of convergecast is the distributed construction of a shortest path tree, with a power aware metric instead of the euclidean distance [9]. Such a tree guarantees that each data packet follows the minimum cost path, where the cost is defined by some metric. This approach is adopted by the *Minimum Transmission Energy* (MTE) routing [16], which employs as edge cost the required energy to transmit over that edge. A similar metric is used by *Minimum Residual Energy* (MRE) [9], which employs as edge cost the inverse of the residual energy of the receiver (edge cost depends on the direction). A significant difference of MRE (apart from the metric) is that it requires a complete construction of the forwarding tree after each round, due to the fact that the nodal energies continually change.

Some methods of data gathering, e.g., *Sink Betweenness* (SBet) [17], do not depend on trees. SBet is a centrality metric, similar to normal betweenness, which measures the number of shortest paths that pass through a node as a percentage of the total number of shortest paths [17]. In SBet forwarding, a node transmits to neighbors closer to the sink than itself, with a probability related to the SBet of each neighbor (higher probability for the smaller SBet).

A forwarding policy similar to the one proposed here is *Maximum Capacity Path* (MCP) [18]. The characterization of an entire path is based on a property of a single node along the path, instead of the total sum of path's edges. A node selects as its next-hop neighbor the one that (i) is closer to the sink (in number of hops); and (ii) belongs to the path with the maximum capacity (i.e., the minimum residual energy of all nodes that belong to the particular path).

III. Network and Vulnerability Definitions

A. Network Model

The network consists of N nodes randomly and uniformly distributed over a geographical area. If for two nodes u and v, their euclidean distance $x(u, v)$ is less than or equal to the *maximum transmission range* d (i.e., $x(u, v) \leq d$), then edge (u, v) exists and it is assumed that this pair of nodes can exchange data with each other. The set $H(u)$ of nodes that an edge exists between them and node u, will be referred to as the *neighbor* nodes of node u.

The sink node, denoted as $\mathtt{s}$, is positioned at some point within the network area. Time is divided in gathering *rounds* (denoted by r). It is assumed that during round r, a node produces one *data packet*, which has to be delivered to the sink in a multi-hop manner. Nodes have no storing capabilities, thus, all generated data need to be delivered to the sink before the end of the round. Let $\mathcal{B}_r(u)$ denote the *energy* left at the battery of node u available at round r. It is assumed that initially all nodes have the same energy level.

Energy consumption follows a simple model [14], where the transmission of one data packet over distance $x(u, v)$ between nodes u and v, requires energy $\kappa + \lambda x^2(u, v)$ (κ and λ constants depending on the system). It is assumed here that data packet transmissions are the dominant factor with respect to energy consumption. All other causes of energy consumption (see for example [19]), such as energy for receiving, sensing, etc., are negligible compared to transmissions. Furthermore, κ is negligible compared to $\lambda x^2(u, v)$, thus, the energy cost of edge (u, v) is proportional to the square of distance $x^2(u, v)$.

B. Vulnerability

Let $p_r(u)$ denote the particular path that data packets follow between node u and the sink $\mathtt{s}$ at round r. For any node u, let its *parent* node $\hat{p}_r(u)$ be the particular neighbor that node u forwards its data packets, or else the first node of path $p_r(u)$, for the particular round r. Given the previously described model, the energy consumed for transmitting a data packet over distance $x(u, \hat{p}_r(u))$ is proportional to the square of this distance, $x^2(u, \hat{p}_r(u))$.

As the energy left at a node's u battery decreases, this node is more vulnerable to stop operating when compared to other nodes. Let fraction,

$$\mathcal{v}_r(u) = \frac{x^2(u, \hat{p}_r(u))}{\mathcal{B}_r(u)}, \tag{1}$$

denote the *node vulnerability* of node u at round r. By convention, node vulnerability of the sink node is zero.

As data packets are forwarded from node u towards the sink along path $p_r(u)$, energy is consumed from the batteries of all nodes $v \in p_r(u)$. Let *path vulnerability* of path $p_r(u)$ be the maximum node vulnerability among all nodes of path $p_r(u)$, denoted as $\mathcal{V}_r(u)$ for node u at round r and given by,

$$\mathcal{V}_r(u) = \max_{\forall v \in p(u)} \mathcal{v}_r(v). \tag{2}$$

Note that node vulnerability is a local property, directly related to an edge, whereas path vulnerability is a global property which characterizes the entire path.

IV. The Proposed Forwarding Policy

The aim of the proposed policy is to let nodes decide on possibly different parents at each round depending on the minimum value of their path vulnerability. This decision requires for every round r all nodes u to be informed about the path vulnerabilities $\mathcal{V}_r(v)$ of their neighbor nodes. In order to avoid sending unnecessary messages (e.g., when flooding approaches are employed) regarding path vulnerability values, the aim here is to reduce the needed messages by suitably selecting the set of nodes the messages will be forwarded to.

Initially, a shortest-cost tree T is constructed, e.g., by a distributed Bellman-Ford algorithm, with edge cost given by

Equation (1). Assuming the same initial energy level of nodes' batteries, it is evident that according to Equation (1) the only factor that changes among nodes is the square of the edge's length. Any path of T, from a node u to the sink $\mathtt{s}$, has the minimum total sum of cost, which is also considered as the cost $c(u)$ of node u. After the construction of the initial tree T, each node becomes aware of its own cost along with its neighbors cost. For a node u *eligible parents*, denoted by $K(u)$, are those neighbors with cost less than its own cost $c(u)$, i.e., $K(u) = \{v : v \in H(u),\, c(u) > c(v)\}$. Neighbors with cost greater than $c(u)$ are the *non-eligible parents* $L(u)$ of u, i.e., $L(u) = \{v : v \in H(u),\, c(u) < c(v)\}$. The following lemma establishes that these two sets are mutually exclusive almost surely.

Lemma 1. *Any neighbor v of u is either an eligible node (i.e., $v \in K(u)$) or non-eligible one (i.e., $v \in L(u)$) almost surely, or equivalently, $K(u) \bigcap L(u) = \emptyset$ and $H(u) = K(u) \cup L(u)$ almost surely.*

Proof. The cost of a node is a sum of distances' squares between the particular node and the sink node. Given that nodes are randomly distributed and distance is a continuous variable, the probability for any two nodes to have the same cost is almost surely zero. For any $v \in H(u)$ the expression $c(v) \neq c(u)$ holds true almost surely, therefore, v is either a member of $K(u)$ or a member of $L(u)$ almost surely. □

Any node u is allowed to forward data packets through eligible parents only. This constraint prevents the formation of loops and also it guarantees that data packets are always transfered closer to the sink after each hop. Note that eligible parents do not change during a network's lifespan. For the sink node, all its neighbors are non-eligible parents.

During each round r, each node u is aware of its own node vulnerability $v_r(u)$ given by Equation (1). However, there is no knowledge about its own path vulnerability $\mathcal{V}_r(u)$. Furthermore, during a round, energy is consumed due to the forwarded data packets and therefore, both node and path vulnerabilities have to be re-evaluated after each round. At the beginning of a new round r, the sink sends to all of its neighbor nodes its own path vulnerability (by definition zero). Upon receiving such a message (which also indicates a new round), a node waits to receive the path vulnerabilities from all of its eligible parents. Then, it calculates its own path vulnerability and forwards the result to all of its non-eligible parents. This distributed algorithm is presented in Algorithm 1. Note that "hello" messages are regularly exchanged among neighbor nodes and therefore, path vulnerabilities could be easily piggybacked in these messages.

It is interesting to observe that the sink node $\mathtt{s}$ will not execute the while loop (lines 4-13) since $K(\mathtt{s}) = \emptyset$ and $\mathcal{V}_r(\mathtt{s}) = 0$, $\forall r$. However, all non-eligible parents of the sink node, will receive the path vulnerability (lines 14-16). This message actually triggers the update of path vulnerabilities and parent node selection throughout the network. Each node u that executes Algorithm 1 receives the path vulnerabilities

Algorithm 1 The Proposed Forwarding Policy.

```
   u : the node that executes this algorithm
   r: the current round
   S: a set of nodes
1: S ← K(u);
2: V_r(u) ← ∞;
3: p̂_r(u) ← nill;
4: while S ≠ ∅ do
5:     wait until a new V_r(v) is received;
6:     S ← S − {v};                          ▷ Remove v from S
       ▷ Check whether v is a better parent node
       ▷ Primed variables hold tentative values
7:     v'_r(u) ← x²(u,v) / B_r(u);
8:     V'_r(u) ← max(v'_r(u), V_r(v));
9:     if V'_r(u) < V_r(u) then
           ▷ A smaller vulnerability has been found
10:        V_r(u) ← V'_r(u);
11:        p̂_r(u) ← v;
12:    end if
13: end while
   ▷ Send the path vulnerability to all non-eligible parents
14: for ∀v ∈ L(u) do
15:    Send V_r(u) to node v;
16: end for
```

of its eligible parents (line 5), selects the proper parent node that corresponds to the minimum path vulnerability (lines 7-12) and sends its own path vulnerability to its own non-eligible parents (lines 14-16). According to this approach, path vulnerabilities are propagated throughout the network in an asynchronous distributed manner, and the number of messages required for each round is equal to the number of edges. The following Lemma 2 shows that there is no deadlock with respect to Algorithm 1.

Lemma 2. *All nodes terminate the execution of Algorithm 1 within a finite time interval.*

Proof. Consider an array of all nodes sorted according to their cost, in ascending order. The first node of the array is the sink (which has zero cost) and the last node is the one that has the highest cost. According to Lemma 1 there are no equal costs among nodes, consequently, the sorted array is unique. By definition, the eligible parents of a node u are always lying on the left of u's position in the considered array.

Base case: The sink always terminates its while loop (lines 4-13), since it has no eligible parents. The second node of the array (the first after the sink) only has one eligible parent, the sink. Therefore, it always receives the path vulnerability of the sink and executes its own while loop within finite time.

Inductive case: Assume that all the nodes from the second to the i^{th} position of the array have finished the execution of the while loop (lines 4-13). The node at the $(i+1)^{\text{th}}$ position receives the path vulnerabilities of all eligible parents and terminates its own while loop. □

V. Simulation Results

Simulations have been contacted in a custom Java program. The proposed forwarding policy is compared with the five similar policies already presented in Section II: (i) the Shortest Path (SP) tree (edge cost equals the euclidean distance); (ii) the SBet tree; (iii) the Maximum Residual Energy (MRE) tree; (iv) the Maximum Capacity Path (MCP); and (v) the Minimum Transmission Energy (MTE) tree. The network consists of $N = 2000$ identical nodes, randomly and uniformly distributed over a unitary square area. All nodes have the ability to adjust their transmission range according to the distance between each node and its parent node. All nodes have the same maximum transmission range d. Given the average number of neighbor nodes g, then $d = \sqrt{g/(\pi N)}$ [20]. Energy consumption follows the model described in Section III-A. No energy is consumed here due to possible collisions or retransmissions, assuming an ideal Data Link Layer.

The initial energy of nodes is critical for a fair comparison of forwarding policies. For example, if nodes are provisioned with an excessive amount of energy, then differences among the various forwarding policies are sometimes hardly noticeable. On the other hand, if the initial energy is significantly small, then the network stops operating after a few rounds resulting in not easily evaluated results. A fair choice is to provision nodes with an amount of energy related to the network size, i.e., each sensor has enough energy to transmit N data packets as far as its maximum transmission range d.

In the sequel, pairs of figures corresponding to two different locations of the sink, one at the center and one at the upper right corner of the network area, are presented for various metrics of interest for all six considered forwarding policies (the proposed one plus the five previously mentioned).

First Death Time is defined as the time until the first node runs out of energy [21]. Fig. 1 shows the effective rounds until the first node failure as a function of the average number of neighbors. It can be seen that the proposed policy gives the best results in both cases.

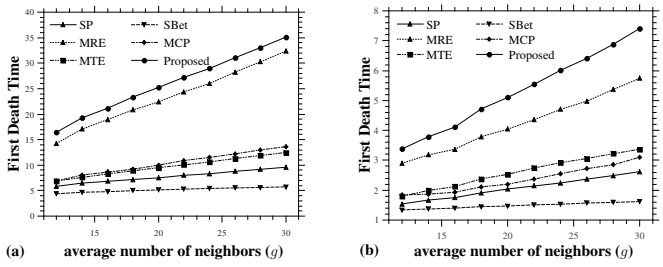

Fig. 1. First Death Time versus g, (a) sink at center, (b) sink at corner.

Lifetime is defined as the time until a severe network partitioning is occurred [22]. Network's demise is recognized by the sink from the sharp reduction in the amount of the delivered data. Fig. 2 depicts simulation results demonstrating that the proposed policy is not that good as the best one (on average 13% lower than MTE forwarding policy). However, it is observed that it performs better than the other four forwarding policies. Despite the fact that a policy may allow a network to live longer, the number of delivered data packets may remain small as shown next.

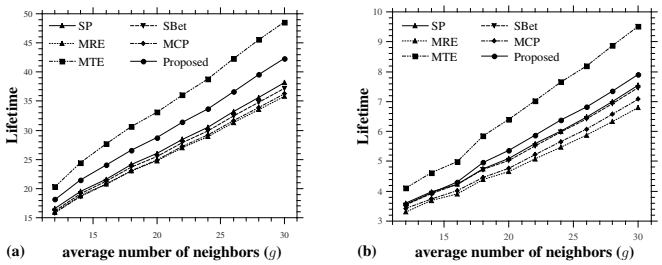

Fig. 2. Lifetime versus g, (a) sink at center (b) sink at corner.

Throughput is defined as the total number of data packets received by the sink during the network's lifetime. As observed in Fig. 3 throughput under the proposed policy is slightly larger than that of MTE and significantly larger than the rest forwarding policies.

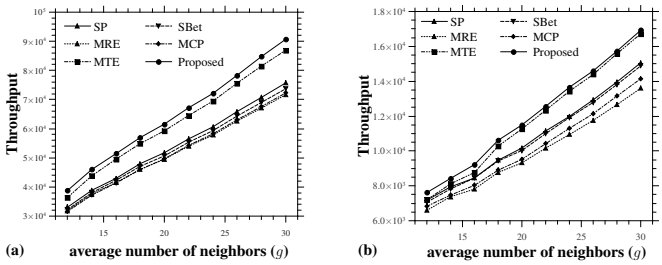

Fig. 3. Throughput versus g (a) sink at center (b) sink at corner.

Overhead is defined as the average number of control packets required for delivering a data packet to the sink. An exact calculation of overhead depends on the actual implementation of each forwarding policy. The aim here is to give an estimation of the control overhead for all six forwarding policies in order to allow for a fair comparison. For example, the number of basic operations required by the Bellmand-Ford algorithm is about N times the number of links. Similarly, a broadcast over a tree is one basic operation. It is assumed here that a basic operation corresponds to one control packet. These rules are applied to all policies. Note that the exact number of the exchanged messages between nodes is much larger than the control packets as estimated here. As observed from Fig. 4, the proposed policy lies in-between the other policies with respect to the overhead.

Latency is defined as the average number of hops for a data packet to cover the distance from the initial node to the sink. The proposed policy favors small hops over long ones and for that reason its latency is significantly larger than the other four policies, even though, lower than MTE, as depicted in Fig. 5.

VI. Conclusions and Future Work

The forwarding policy proposed in this paper relies on changing the paths that data packets follow towards the sink in

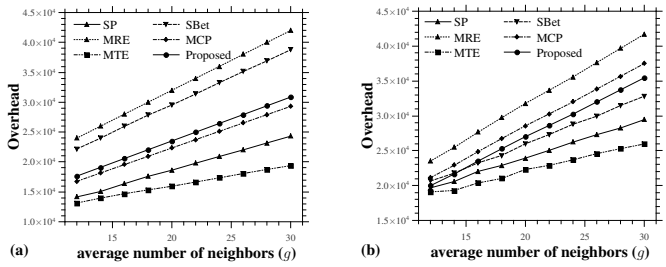

Fig. 4. Overhead versus g (a) sink at center (b) sink at corner.

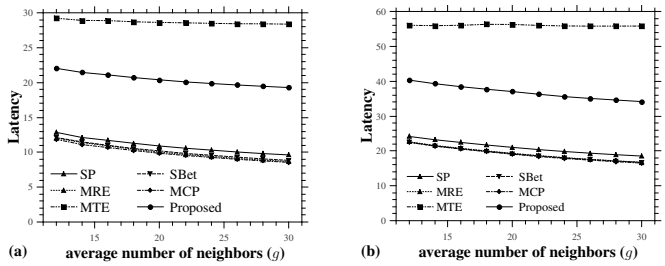

Fig. 5. Latency versus g (a) sink at center (b) sink at corner.

a wireless sensor network given the path vulnerability metric that is defined here. As evaluated against other forwarding policies, using simulations, the proposed policy was shown that increases the time period until the first node runs out of energy as well as throughput and behaves close to the other policies with respect to network lifetime, overhead and latency. Future work will attempt to further reduce the overhead and increase the network's lifetime in the particular convergecast environment.

ACKNOWLEDGMENTS

This work was supported in part by project "A Pilot Wireless Sensor Networks System for Synchronized Monitoring of Climate and Soil Parameters in Olive Groves," (MIS 5007309) which is partially funded by European andNational Greek Funds (ESPA) under the Regional Operational Programme "Ionian Islands 2014-2020." In addition, this work was supported in part by the European Commission as part of the ReCRED project (Horizon H2020 Framework Programme of the European Union under GA number 653417), by the Chair of Excellence UC3M - Santander Program and by the National and Kapodistrian University of Athens (S.A.R.G.).

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. Hanzo, "A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

[3] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, "Networking issues in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 799–814, 2004.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

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

[6] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 551–591, 2013.

[7] S. Upadhyayula and S. K. S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 5, pp. 626–648, 2007.

[8] J. Lian, L. Chen, K. Naik, T. Otzu, and G. Agnew, "Modelling and enhancing the data capacity of wireless sensor networks," *IEEE Monograph on Sensor Network Operations*, 2004.

[9] J. H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 2000, pp. 22–31.

[10] S. Singh, M. Woo, and C. S. Raghavendra, "Power aware routing in mobile ad hoc networks," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1998, pp. 181–190.

[11] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless ad-hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 940–965, 2012.

[12] W. Bechkit, M. Koudil, Y. Challal, A. Bouabdallah, B. Souici, and K. Benatchba, "A new weighted shortest path tree for convergecast traffic routing in WSN," *Proceedings - IEEE Symposium on Computers and Communications*, pp. 000 187–000 192.

[13] F. Ren, J. Zhang, T. He, C. Lin, and S. K. Ren, "Ebrp: Energy-balanced routing protocol for data gathering in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2108–2125, 2011.

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2000.

[15] C. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks by introduction to wireless ad hoc network," *IEEE Communications Magazine*, vol. June, no. 6, pp. 138–147, 2001.

[16] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[17] H. S. Ramos, A. C. Frery, A. Boukerche, E. M. R. Oliveira, and A. A. F. Loureiro, "Topology-related metrics and applications for the design and operation of wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 3, p. 53, 2014.

[18] S. C. Huang and R. H. Jan, "Energy-aware, load balanced routing schemes for sensor networks," in *Proceedings. Tenth International Conference on Parallel and Distributed Systems, 2004. ICPADS 2004.* IEEE, 2004, pp. 419–425.

[19] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1. IEEE, 2006, pp. 286–295.

[20] A. Demertzis and K. Oikonomou, "Avoiding energy holes in wireless sensor networks with non-uniform energy distribution," in *Information, Intelligence, Systems and Applications, IISA 2014, The 5th International Conference on*. IEEE, 2014, pp. 138–143.

[21] A. Liu, X. Jin, G. Cui, and Z. Chen, "Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network," *Information Sciences*, vol. 230, pp. 197–226, 2013.

[22] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 5, 2009.