MEPA: A New Protocol for Energy-Efficient, Distributed Clustering in Wireless Sensor Networks

Hung Quoc Ngo [1], Young-Koo Lee [2], Sungyoung Lee [3]

Department of Computer Engineering, Kyung Hee University
South Korea, 446-701
[1]nqhung@oslab.khu.ac.kr
[2]yklee@khu.ac.kr
[3]sylee@oslab.khu.ac.kr

***Abstract*— Clustering is an effective approach to hierarchically organizing network topology for efficient data aggregation in wireless sensor networks. Distributed protocols with simple local computations to accomplish a desired global goal, offer a good prospect for achieving energy efficiency. This paper presents MEPA – an energy-efficient distributed clustering protocol using simple and local message-passing rules. Our proposed clustering protocol combines both node residual energy and network topology features to recursively elect a near-optimal set of cluster heads. Simulation results show that MEPA can produce a set of cluster heads with compelling characteristics, and effectively prolong the network lifetime.**

I. Introduction

Wireless sensor networks (WSN) consist of thousands of tiny nodes deployed to collect environmental parameters and transmit the collected data to external observers. The dense deployment, resource constraints, and unattended nature of WSNs make the issue of energy efficiency a primary design goal in this field [1].

Clustering has been shown to be an effective approach to hierarchically organizing network topology for efficient data aggregation [2], [3], [4]. Sensor clustering essentially identifies a set of cluster heads (CHs) from the network population, and then forms small clusters of the remaining nodes with these heads. In each cluster, the cluster head acts as a coordinator to which the cluster-member nodes can communicate their measurements directly (intracluster communications). These cluster heads then forward the aggregated data to the external observers through other CHs on behalf of their clusters (intercluster communications).

There have been many clustering approaches proposed for WSNs, which can be differentiated depending on whether clustering is performed in a centralized or distributed manner [5]. Centralized clustering algorithms (e.g. [6], [7]) are often executed at a base station (BS) after all necessary information about the network topology is collected. Since huge communication overhead is involved in gathering such information, centralized protocols are very time and energy inefficient. Distributed (localized) clustering algorithms [8] rely only on local parameters and are executed on each node to achieve a desired global goal. These local parameters can be obtained from node's k-hop neighbors, such as residual energy, node degree, mobility, average distance to neighbors, etc. Distributed algorithms are thus very scalable and preferable in large-scale WSNs.

Energy-efficient clustering (e.g. [2], [3], surveys [5] and [9], and references therein) focuses on prolonging the network lifetime by selecting the CHs among nodes with higher residual energy, balancing energy consumption between CHs, or by ensuring rapid convergence with low message overhead during the construction of clusters. The hybrid energy-efficient distributed (HEED) clustering approach in [3], is one of the most recognized energy-efficient clustering protocols. In HEED, the clustering process is divided into a number of iterations, and in each iteration, nodes which are not covered by any CH double their probability of becoming a CH. Since these energy-efficient clustering protocols enable every node to independently and probabilistically decide on its role in the clustered network, they cannot guarantee optimal elected set of CHs in terms of residual energy. Furthermore, during the CH election process, the selecting criterion is based solely on node residual energy, while network topology features (e.g. node degree, distances to neighbors) are only used as secondary parameters to break tie between candidate CHs, thus the resulting set of CHs may not be optimal in terms of network connectivity.

In this paper, we present a new approach to energy-efficient, distributed clustering in WSNs. Our proposed clustering protocol takes into account both node residual energy and network topology features during cluster head election process. Furthermore, it does not assign any probability for node to become a CH; instead, the near-optimal set of cluster heads emerges after a bounded number of iterations using simple and localized message-passing rules (thus named MEPA). The MEPA clustering protocol is totally distributed, location-*unaware*., and very scalable to the network size. Simulation results show that our protocol can produce clusters with compelling characteristics e.g. CHs with high residual energy, and prolonged network lifetime.

The remainder of this paper is organized as follows. We present our network model, clustering parameter, and the clustering procedure along with the pseudocode in Section II. In Section III, we evaluate the proposed protocol through simulation, and compare its effectiveness to the HEED protocol. Finally, we give concluding remarks and future extensions in

Section IV.

II. The MEPA Protocol

A. Assumptions on WSN Model

Consider a network of N sensors. In the sequel we use the terms "sensor" and "node" interchangeably. Let $\mathcal{G}$ be a undirected graph defined by a set of vertices (or nodes) $\mathcal{V}$ = {1,...,N} and a set of edges (or links) $\mathcal{E}$. Nodes i and j are neighbors if they are connected by an edge, i.e. $(i, j) \in E$. Let $\mathcal{N}(i) := j|(i, j) \in \mathcal{E}$ denote the set of neighbors of node i and $\mathcal{N}(i)\backslash j$ denote the set obtained by excluding j from $\mathcal{N}(i)$. The WSN model we are focusing has some basic assumptions. First, we assume the sensor nodes are quasi-stationary, location-unaware, and left unattended after deployment. Second, every node is assumed to use the same, fixed power level for intracluster communication (e.g. broadcasting, and communicating with CH). For intercluster communications, CHs are capable of increasing its transmission power level to reach other CHs or the base stations (Berkeley Motes [10] are typical examples). Third, the communications are assumed to be symmetric, i.e. if node i can communicate with node j, then node j can also communicate with node i using the same transmission power level. Finally, we assume all sensors are synchronized by employing some mechanism, such as the one described in [11].

B. Clustering Parameters

To prolong network lifetime, CH selection should be in favor of nodes with higher residual energy. We assume that each node is readily equipped with some mechanism for estimating its residual energy up to some accepted level of accuracy [12]. Residual energy is the primary parameter in our energy-efficient clustering algorithm, which is proportional to the *preference* of one node to select another node as its CH in a localized point of view. On the other hand, from the network topology point of view, high-degree nodes are also preferred to be selected as CHs, since they play an important role in connecting other nodes and act as data fusion/aggregation centers.

These observations motivate us to use the *normalized preference* as our clustering parameter, which is essentially node residual energy divided by the total residual energy of neighboring nodes. Let us consider a sensor node i in Fig. 1. The *normalized preference* of sensor i for one of its neighbors, sensor j, is defined as:

$$p_i\left(j|j \in \mathcal{N}(i)\right) = \frac{re_j}{\sum\limits_{j' \in \mathcal{N}(i)} re_{j'}} \tag{1}$$

We can observe that the normalizing factor $\sum\limits_{j' \in \mathcal{N}(i)} re_{j'}$ implicitly captures network topology feature by taking into account the neighboring nodes of i.

There are several important implications from the *normalized preference* in Equation 1. First, the *self-normalized preference*, $p_i(i) = \frac{re_i}{\sum\limits_{j' \in \mathcal{N}(i)} re_{j'}}$, indicates the willingness of a node to be a CH. With the same level of residual energy, a node is more willing to become a CH when its neighboring nodes have less residual energy. Second, the higher *normalized preferences* a node receives from all of its neighbors, the higher chances are that it will be elected as a CH.

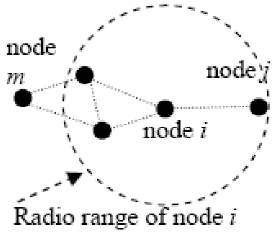

Fig. 1. A snapshot of a Wireless Sensor Network

C. Near-Optimal Clustering

From the above discussions, CH selection favors the nodes receiving higher preferences from its neighbors. Thus the sensor clustering issue now becomes finding a subset of nodes in the whole network which maximizes the total preferences they receive. It is known that exactly maximizing the net preference is computationally intractable, since a special case of this maximizing problem is the NP-hard k-mean problem in data clustering [13]; we can only find approximate solutions which are heuristic in nature. We propose a new approach for recursively finding a near-optimal clustering that maximizes the net preference, using the max-sum algorithm, a message-passing procedure that operates in a factor graph [14]. Message-passing algorithms were first invented in information theory to derive the best error correction algorithms to date [15], and recently used in *belief-propagation* [16] to obtain impressive results in probabilistic inference problems [17], computer vision [18], and many other disciplines [19]. Due to space limitation, we just briefly introduce the concepts here, and present the derived message-passing rules for the near-optimal clustering issue.

D. Message-Passing Rules for Near-Optimal Clustering

Factor graphs [14] can be used to represent a complicated global function that is a product of simpler "local" functions, each of which depends on a subset of the variables. In a factor graph, the sum-product algorithm can compute, either exactly or approximately, various marginal functions using a single, simple computational message-passing rule. The technique can be modified to find the most probable state, giving rise to the max-sum algorithm [20]. For our near-optimal clustering problem, we first represent the net preference function using a factor graph, and then apply max-sum algorithm to recursively search for the near-optimal cluster configuration that maximizes the net preference. The derived message passing rules [19] are quite simple:

- Request message $req_i(j)$ sent *from* sensor i to its neighbor j, reflects the accumulated *suitability* for sensor i to select neighbor j as its CH, taking into account other neighboring CH candidates j' of sensor i.

$$req_i(j) = p_i(j) - \max_{j' \in \mathcal{N}(i)/j} [p_i(j') + res_i(j')] \quad (2)$$

- Response message $res_i(j)$ sent *to* sensor i from its neighbor j, reflects the accumulated *appropriateness* for sensor i to choose neighbor j as its CH, taking into account the requests from other neighbors j' of sensor j.

$$res_i(j) = \min \begin{bmatrix} 0, req_j(j) \\ + \sum\limits_{j' \in \mathcal{N}(j)/i} \max(0, req_j(j')) \end{bmatrix} \quad (3)$$

$$\underset{(self-response)}{res_i(i)} = \sum_{j \in \mathcal{N}(i)} \max(0, req_i(j)) \quad (4)$$

These are localized, simple computational rules that are easy to implement, and well-suited to a WSN setup; since messages are only passed between pairs of neighboring nodes. The optimal set of CHs emerges from this message-passing procedure. At any time, the (intermediate) CH candidate of node i can be decided by the value that maximizes the sum:

$$CH_i = \underset{j \in \mathcal{N}(i) \cup \{i\}}{\arg\max} \, [res_i(j) + p_i(j)] \quad (5)$$

The procedure on each node may terminate if the message changes are smaller than some threshold, or the intermediate set of CHs is unchanged after several iterations.

E. Protocol Execution

From the local rules of message passing and update, derived above, we now describe the localized clustering algorithm executed at each sensor node which can achieve the global goal: Electing the near-optimal set of CHs. We divide the lifetime of WSN into a number of rounds; each round begins with a clustering phase, followed by a network operation phase (T_{OP}) when data is sent from the cluster-member nodes to the CHs and onto the observers [2]. The clustering phase in MEPA consists of three procedures, as described in Fig. 2. In the initialization phase, each node calculates the normalized preferences (for all of its neighbors and for itself) using Equation 1.

The CH election procedure – the main procedure – is essentially comprised of receiving, updating, and broadcasting operations on the request/response message pairs. During each iteration, every sensor has to collect all incoming messages broadcasted by its neighbors before updating its requests/responses using Equations 2, 3, and 4 (lines 4 and 7 of phase II in the pseudo code). These procedures take some time to finish, thus timeout periods have to be added in real implementation. Only one outgoing request/response message is broadcasted by each sensor, by marshalling all <neighborID, update_value> pairs into one "compact" packet. The procedure terminates if the temporary cluster head ID (CH_{temp} estimated in Equation 5) is unchanged after a number of *conv_iter* iterations, or when the maximum number of iterations *max_iter* is reached. These are two key parameters that need to be carefully selected in real implementation, since the more number of recursions, the better approximation of the optimal clustering, at the cost of more messages to be broadcasted. Through our results of 100 runs, under different simulation setups, good upper bounds for *conv_iter* and *max_iter* were found to be 5 and 15 respectively.

I. INITIALIZATION

1. $S_{NBR} \leftarrow$ {j| one-hop neighborhood}
2. broadcast(nodeID, re_{nodeID});
3. **for** j∈S_{NBR}∪ {nodeID}
4. computePreference(nodeID,j);
5. $res_{nodeID}(j) \leftarrow 0$;
6. **end**
7. $S_{CH} \leftarrow 0$ //Set of candidate CHs

II. CLUSTERHEAD ELECTION

1. **repeat**
2. updateAllRequest();
3. broadcastCompactRequest();
4. collectAllRequest();
5. updateAllResponse();
6. broadcastCompactResponse();
7. collectAllResponse();
8. updateAllResponse();
9. $CH_{temp} \leftarrow \arg\max_{j \in S_{NBR}} [res_{nodeID}(j) + p_{nodeID}(j)]$
10. **until** TERMINATE

III. CLUSTER FORMATION

1. **if** CH_{temp} = nodeID
2. CH ← nodeID;
3. announceCH(nodeID, cost);
4. collectJoinCluster();
5. **else**
6. collectAnnounceCH();
7. $S_{CH} \leftarrow$ {j| incoming_announceCH(j)};
8. CH ← j| (j∈S_{CH} AND j has least cost); //tie-breaking
9. joinCluster(nodeID,CH);
10. **end**

Fig. 2. MEPA Clustering Protocol Pseudocode

In the subsequent cluster forming procedure, if one sensor identifies itself as a CH, it will broadcast an announcement message carrying a cost value (line 3 of phase III). This secondary parameter reflects the intracluster communication cost when a node joins the cluster under this CH [3]. In case there are several candidate CHs are within the radio range of a non-CH node, using this cost the node can decide to join a more energy-efficient cluster. Minimum node degree proved to be a rough yet effective tie-breaking condition, as it tends to balance the load between CHs and thus extending network lifetime [3].

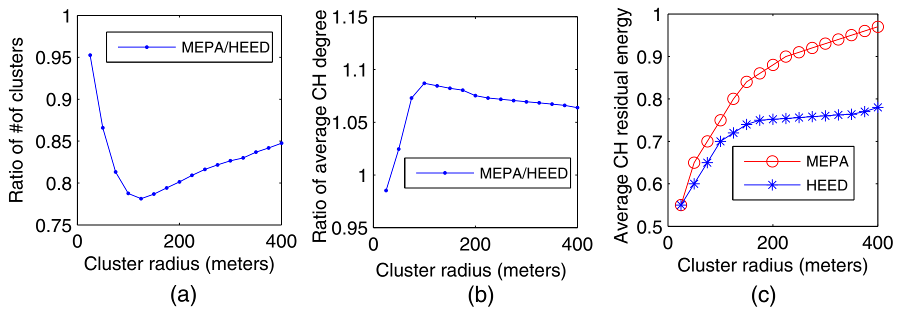

Fig. 3. Characteristics of selected CHs a) Ratio of average number of CHs, b) Ratio of average CH degree c) Average residual energy of selected CHs

III. Performance Evaluations

In this section, we evaluate the performance of our clustering protocol through two simulation setups. In the first simulation we analyze the clustering characteristics of MEPA protocol in clustering phase only, while in the second simulation we study the energy efficiency of the protocol during the network lifetime of a clustering application. We choose the HEED protocol [3] as the baseline to compare our results, and repeated the simulation setup of HEED using MATLAB.

A. Distributed Clustering Analysis

We assume that 1,000 nodes were randomly deployed in a field of size 2,000 meters $\times$ 2,000 meters. Residual energy of each sensor was first randomly generated between 0.1 and 1 Joule. We vary the radio range for intracluster communications from 25m to 400m to evaluate the protocol in different node density. For each cluster radius, 100 trials were conducted independently, and then the results are averaged for comparison.

Fig. 3(a) shows the ratio of the average numbers of clusters generated by MEPA and HEED, in which MEPA generates 15% to 25% less number of clusters than HEED. As a result, the average CH degrees is slightly higher in MEPA, up to 9% compared to HEED, as shown in Fig. 3(b). This is because node degree is just secondary parameter for CH election in HEED, while MEPA favors nodes with high residual energy as well as high degree, as presented in section II - clustering parameter. Thus, compared to HEED, MEPA produces less number of CHs with higher CH degree to cover the whole network.

In HEED, optimal CH selection is not guaranteed, since it randomly selects tentative cluster heads based on their residual energy. This is not the case of MEPA, since the message-passing algorithm identifies a near-optimal set of CHs having relatively high residual energy. Fig. 3 (c) compares the two protocols in terms of average cluster head residual energy. The results show that the CHs selected in MEPA, in average, have much higher residual energy, up to 25% compared to those selected in HEED. Especially, when the cluster range increases from 25m to 400m, the number of neighboring nodes having high residual energy for one node to select as CH also increases, thus the average CH residual energy approaches 1.

From the above characteristics of the elected cluster heads, we can see that compared to HEED, MEPA shows better performance by producing less number of clusters with higher residual energy CHs.

B. Hierarchical Data Aggregation Analysis

In this simulation setup, we analyze the effectiveness of our clustering protocol for sensor applications that require efficient data aggregation and prolonged network lifetime, e.g. environmental monitoring applications. We consider a network of size (150m x 150m), with one external sink located at (200m, 75m). The re-clustering process is triggered every T_{OP} TDM frames, which is set to 10 in our simulations. Designing an optimal re-clustering process to distribute energy consumption evenly among sensor nodes, and to overcome CH failures, is left for future work. In each TDM frame, every node sends its data to the CH according to the specified TDMA schedule. Each CH then performs data fusion and sends the fused data packets to the sink. Any ad hoc routing, such as Directed Diffusion [21] or Dynamic Source Routing (DSR) [22], can also be employed for intercluster routing. Since the issue of local data correlation is not our main focus [23], we assume perfect data correlation, thus only one data packet is enough to send all the aggregated data from each CH to the sink in each TDM frame [2]. The packet sizes are listed in Table I. We use the simple radio model used in LEACH and HEED, in which the power amplifier setting is free space (d^2 power loss) channel model when the distance between the CH and the sink is less than a threshold d_o; otherwise, the multipath fading (d^4 power loss) channel model is used [24]. The simulation parameters of the radio model are set to the same values with those used in [3].

TABLE I

Packet Sizes in MEPA

Parameter	**Value**
Data packet size	200 bytes
Broadcast packet size (ADV, Announce-CH, Join-CH)	10 bytes
Compact REQ/RES packet size	40 bytes
Packet header size	10 bytes

We measure the network lifetime by the number of rounds until the first/last node dies. We conducted 100 independent simulations for each simulation setting, and then calculated the average network lifetime. Fig. 4(a) and (b) compare average

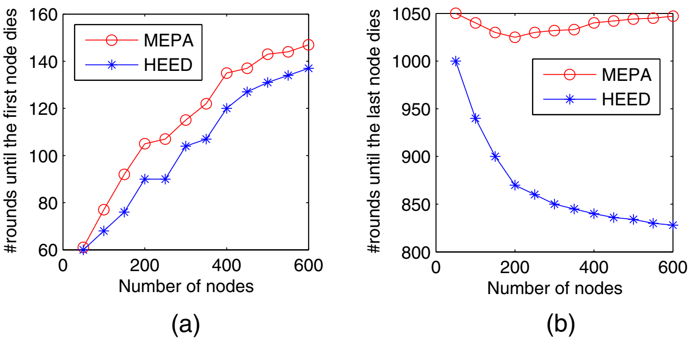

Fig. 4. Average network lifetime until a) the first and b) the last node dies

network lifetime when the first/last node dies between MEPA and HEED. MEPA constantly improves network lifetime over HEED for all node density settings, despite the fact that MEPA requires more messages to be sent and received during the clustering phase compared to HEED. This is mainly because in MEPA, the set of CHs is approximately optimally elected through the message-passing recursions, while in HEED, every node independently and probabilistically elects itself to be a cluster head.

IV. Conclusion and Future Work

We have introduced MEPA, a new energy-efficient distributed clustering protocol for WSNs. To prolong the network lifetime, the MEPA protocol takes into account both node residual energy and network topology features in its clustering parameter. By applying simple and localized message-passing rules, the near-optimal set of cluster heads emerges after a bounded number of iterations. Simulation results show that our clustering protocol elects CHs with high residual energy, and effectively prolongs network lifetime.

We are currently investigating the robustness of MEPA protocol in the presence of communication failures. We also plan to extend the MEPA protocol by considering node mobility, multi-hop clustering, and other practical issues in deployment. These issues include how to ensure intercluster connectivity, how and when to optimally initiate re-clustering process to rotate the role of CHs or to recover from CH failures, how to flexibly decide the optimal cluster size, and how to design an efficient MAC layer scheduling for concurrent intracluster and intercluster transmissions to minimize collision and interference.

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments and suggestions. This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency.

Corresponding author: Professor Young-Koo Lee.

References

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

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless. Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[3] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, Oct.-Dec. 2004.

[4] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48–63, 4th Quarter 2006.

[5] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challanges," *IEEE Network*, vol. 20, no. 3, May/June 2006.

[6] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multihop wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2001.

[7] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *IEEE Aerospace Conference*, vol. 3, Mar. 2002, pp. 1125–1130.

[8] S. Olarius, S.-R. D., and I. Stojmenovic, "Localized communication and topology protocols for ad hoc networks: A preface to the special section," *IEEE Trans. Parallel and Distrib. Syst.*, vol. 17, no. 4, 2006.

[9] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1, pp. 32–48, 1st Quarter 2005.

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors," in *ASPLOS-IX: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*, 2000, pp. 93–104.

[11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. Symp. Operating Systems Design and Implementation (OSDI)*, vol. 36, 2002, pp. 147–163.

[12] O. Younis and S. Fahmy, "An experimental study of routing and data aggregation in sensor networks," in *Proc. Int. Workshop on Localized Communication and Topology Protocols for Ad hoc Networks (LOCAN)*, Nov. 2005.

[13] M. Charikar, S. Guha, A. Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," *J. Comp. and Sys. Sci.*, vol. 65, no. 1, 2002.

[14] F. R. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Nov. 2001.

[15] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[17] B. J. Frey and N. Jojic, "A comparison of algorithms for inference and learning in probabilistic graphical models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1392–1416, 2005.

[18] T. Meltzer, C. Yanover, and Y. Weiss, "Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation," in *Proc. Tenth IEEE Int. Conf. on Computer Vision (ICCV'05)*, vol. 1, 2005, pp. 428–435.

[19] B. J. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, February 2007.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006, ch. 8, p. 740.

[21] R. G. Chalermek Intanagonwiwat and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MOBICOM)*, 2000.

[22] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, 1996, vol. 353.

[23] A. Jindal and K. Psounis, "Modeling spatially correlated data in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 4, pp. 466–499, 2006.

[24] T. Rappaport, *Wireless Communications: Principles and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.