Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges

Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian, University of Arizona

Abstract

The large-scale deployment of wireless sensor networks (WSNs) and the need for data aggregation necessitate efficient organization of the network topology for the purpose of balancing the load and prolonging the network lifetime. Clustering has proven to be an effective approach for organizing the network into a connected hierarchy. In this article, we highlight the challenges in clustering a WSN, discuss the design rationale of the different clustering approaches, and classify the proposed approaches based on their objectives and design principles. We further discuss several key issues that affect the practical deployment of clustering techniques in sensor network applications.

Recent years have witnessed an increasing interest in using wireless sensor networks (WSNs) in many applications, including environmental monitoring and military field surveillance. In these applications, tiny sensors are deployed and left unattended to continuously report parameters such as temperature, pressure, humidity, light, and chemical activity. Reports transmitted by these sensors are collected by observers (e.g., base stations). The dense deployment and unattended nature of WSNs make it quite difficult to recharge node batteries. Therefore, energy efficiency is a major design goal in these networks.

Several WSN applications require only an aggregate value to be reported to the observer. In this case, sensors in different regions of the field can collaborate to aggregate their data and provide more *accurate* reports about their local regions. For example, in a habitat monitoring application [1], the *average* reported humidity values may be sufficient for the observer. In military fields where chemical activity or radiation is measured, the maximum value may be required to alert the troops. In addition to improving the fidelity of the reported measurements, data aggregation reduces the communication overhead in the network, leading to significant energy savings.

In order to support data aggregation through efficient network organization, nodes can be partitioned into a number of small groups called *clusters*. Each cluster has a coordinator, referred to as a *cluster head*, and a number of *member* nodes. Clustering results in a two-tier hierarchy in which cluster heads (CHs) form the higher tier while member nodes form the lower tier. Figure 1 illustrates data flow in a clustered network. The member nodes report their data to the respective CHs. The CHs aggregate the data and send them to the central base through other CHs. Because CHs often transmit data over longer distances, they lose more energy compared to member nodes. The network may be reclustered periodically in order to select energy-abundant nodes to serve as CHs, thus distributing the load uniformly on all the nodes. Besides achieving energy efficiency, clustering reduces channel contention and packet collisions, resulting in better network throughput under high load.

Clustering has been shown to improve *network lifetime*, a primary metric for evaluating the performance of a sensor network. Although there is no unified definition of "network lifetime," as this concept depends on the objective of an application, common definitions include the time until the first/last node in the network depletes its energy and the time until a node is disconnected from the base station. In studies where clustering techniques were primarily proposed for energy efficiency purposes (e.g., [2, 3]), the network lifetime was significantly prolonged. An example is shown in Fig. 2 (extracted from [2]), where the number of nodes that remain alive using LEACH is significantly larger (four to eight times larger) than that using static clustering or minimum transmission energy (MTE) routing. Note that even though the objective of several protocols was not to maximize network lifetime, lifetime improvements can still be achieved if data aggregation is exploited and the network is reclustered periodically.

Clustering has been extensively studied in the data processing and wired network literatures. The clustering approaches developed in these areas cannot be applied directly to WSNs due to the unique deployment and operational characteristics of these networks. Specifically, WSNs are deployed in an ad hoc manner and have a large number of nodes. The nodes are typically unaware of their locations. Hence, distributed clustering protocols that rely only on neighborhood information are preferred for WSNs (however, most studies in this area still assume that the network topology is known to a centralized controller). Furthermore, nodes in WSNs operate on bat-

---

This work was supported by NSF under grants ANT-0095626, ANT03 13234, ANT-0325979, and ANT-0435490, by EEC-0333046, and by the Center for Low Power Electronics (CLPE) at the University of Arizona. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not reflect the views of NSF.

0890-8044/06/$20.00 © 2006 IEEE

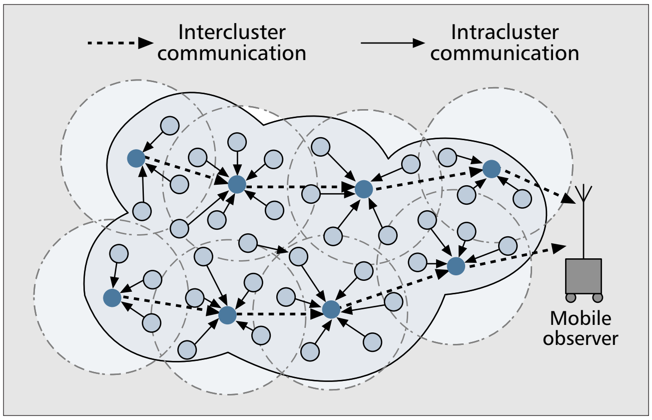


Figure 1. *Illustration of data flow in a clustered network.*

tery power with limited energy. Hence, the employed clustering approach must have low message overhead. Finally, harsh environmental conditions result in unexpected failures of nodes. Hence, periodic reclustering is necessary in order to heal disconnected regions and distribute energy consumption across all nodes. Periodic reclustering is also necessary, as the parameters used for clustering (e.g., the remaining energy, node degree, etc.) are *dynamic*. The clustering techniques proposed for data processing typically consider *static* parameters, such as the distance between the nodes, and assume that nodes are more reliable.

The goal of this article is to introduce the reader to different classes of clustering approaches that have been proposed for WSNs. We compare some representative protocols in terms of their design approach, clustering criteria, basic assumptions, and overhead. We consider clustering proposals that target the general class of ad hoc networks and are also suitable for sensor networks. This article focuses more on clustering "approaches" than on "protocols." A clustering "approach" in this context identifies the parameter(s) used for partitioning the network and whether clustering is performed in a centralized or a distributed fashion. In addition, we highlight the basic challenges in deploying clustering techniques in WSNs.

Classification of Clustering Techniques in Wireless Sensor Networks

Clustering in WSNs involves grouping nodes into clusters and electing a CH such that:
- The members of a cluster can communicate with their CH directly.
- A CH can forward the aggregated data to the central base station through other CHs.

Thus, the collection of CHs in the network form a *connected dominating set*. Research on clustering in WSNs has focused on developing centralized and distributed algorithms to compute connected dominating sets. We focus on distributed approaches in this article since they are more practical for large-scale deployment scenarios. Since obtaining an optimal dominating set is an NP-complete problem, the proposed algorithms are heuristic in nature.

We classify the clustering techniques based on two criteria:
- The parameter(s) used for electing CHs
- The execution nature of a clustering algorithm (probabilistic or iterative)

Election of Cluster Heads

One class of clustering techniques uses the node identifier to elect CHs. The success of this approach depends on two assumptions:
- Every node has a unique identifier.
- These identifiers are uniformly assigned throughout the field.

For example, the scheme in [4] favors nodes with lower identifiers to become CHs. This approach may not be suitable for energy-constrained sensor networks because it penalizes specific nodes in the network irrespective of their battery lifetimes.

Another class favors nodes with larger degrees (e.g., Kuhn *et al.* [5] and Amis *et al.* [6]) in order to create dense clusters and elect the minimal dominating set of CHs (the degree of a node is the number of its neighbors within a prespecified transmision called the *cluster range*). This, however, may result in quickly draining the battery of larger-degree nodes. From an application perspective, balancing cluster sizes reduces the load on CHs. However, this comes at the expense of having more clusters in the network and thus a larger routing overlay. A third class of techniques favors nodes with higher *weights* to become CHs. The weight of the node is used to define its significance. For example, it can be the residual battery energy (as in the HEED protocol [3]), its degree (as in the ACE protocol [7]), or a combination of parameters (e.g., remaining energy, degree, mobility, and average distance to neighbors).

Some protocols, such as GAF [8] and SPAN [9], were proposed for controlling the network topology by exploiting node redundancy. These protocols select certain nodes to be active (i.e., participate in sensing and data forwarding), while others are put to sleep to save their energy. In GAF [8], for example, a node belongs to a region that is determined by its location. A region in this context is defined as an area A in which any node u can communicate via a single hop with any node $v \in B$, where B is a neighboring region to A. Thus, only one representative node in any region needs to participate in the routing infrastructure at any given time to ensure network connectivity. In SPAN [9], a node decides whether to remain active or go to sleep according to its two-hop neighborhood connectivity. Although these protocols are not clustering techniques, their effect on the network topology is similar to that of clustering.

Execution of a Clustering Algorithm

The execution of a clustering algorithm can be carried out at a centralized authority (e.g., a base station) or in a distributed way at local nodes. Centralized approaches require global

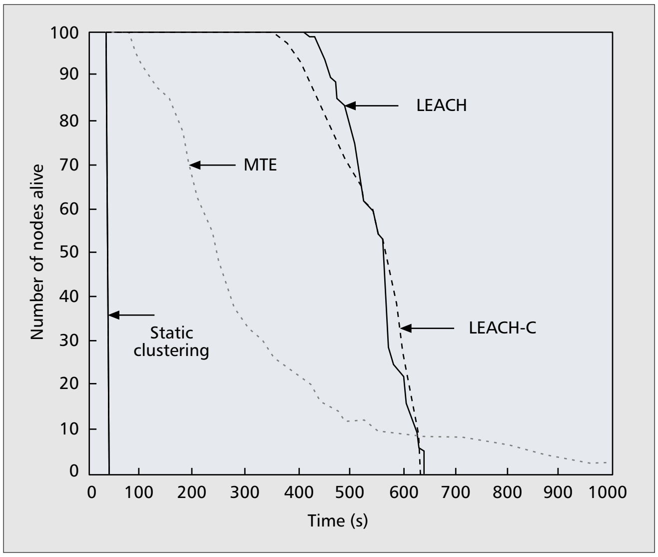


Figure 2. *Number of nodes that remain alive over time [2].*

information about the network topology. The classic K-Means clustering approach (proposed in the data processing literature) may be applied if the number of desired clusters can be determined a priori and node locations are available. In this case, an initial random set of clusters is selected and a node is moved from one cluster to another if this move reduces the objective cost function of the entire system. Banerjee *et al.* [10] proposed a centralized technique that does not require knowledge of node locations. Their technique is based on constructing a spanning tree of clusters that is rooted at an observer and forcing a bound on the maximum and minimum cluster size. A distributed protocol was also proposed in [10] for constructing the spanning tree.

The efficacy of centralized approaches is limited in large-scale networks where collecting all the necessary information at the central authority is both time and energy consuming. Distributed (localized) approaches are more suitable for large-scale networks. In such approaches, a node decides to join a cluster or become a CH based on the information obtained solely from its one-hop neighbors. Several distributed clustering techniques have been proposed in the literature. These techniques are either *iterative* or *probabilistic* in nature.

Iterative Clustering Techniques — In iterative clustering techniques, a node waits for a specific event to occur or certain nodes to decide their role (e.g., become CHs) before making a decision. For example, in the Distributed Clustering Algorithm (DCA) [11], before making a decision, a node waits for all its neighbors with higher weights to decide to be CHs or join existing clusters. Nodes possessing the highest weights in their one-hop neighborhoods are elected as CHs. If a node receives multiple CH announcements, it arbitrates among these CHs using a preference condition (e.g., higher weight wins). If none of the higher-weight neighbors of a node decides to become a CH, then this node decides to become a CH. The problem with most iterative approaches is that their convergence speed is dependent on the network diameter (path with the largest number of hops). In a two-dimensional field with n deployed nodes, the DCA algorithm requires

$$O\left(\sqrt{n}\right)$$

iterations to terminate. The worst-case convergence speed can be as slow as $n - 1$ iterations in a 1D setting. The performance of iterative techniques is also highly sensitive to packet losses. For example, if a node u discovers that one of its neighbors v has a higher weight, then u has to wait for v to decide before it makes a decision. If node v fails shortly after the neighborhood discovery phase, node u will wait indefinitely for v to make a decision.

To ameliorate the above problems, some protocols enforce a bound on the number of iterations for each node. For example, in ACE [7] when a node finishes executing a number of iterations (e.g., 5), it makes a decision based on the available information. These iterations are enough to achieve a stable average cluster size. The protocol in [6] allows a cluster to include nodes that are D hops away from the CH. A node executes 2D iterations before making a decision. This results in a constant number of iterations for convergence.

Probabilistic Clustering Techniques — The probabilistic (or randomized) approach for node clustering ensures rapid convergence while achieving some favorable properties, such as balanced cluster sizes. It enables every node to independently decide on its role in the clustered network while keeping the message overhead low. We now discuss a few examples of this approach.

The LEACH protocol [2] is an application-specific clustering protocol, which has been shown to significantly improve the network lifetime. It assumes that every node is reachable in a single hop and that load distribution is uniform among all nodes. LEACH assigns a fixed probability to every node so as to elect itself as a CH. The clustering process involves only one iteration, after which a node decides whether to become a CH or not. Nodes take turns in carrying the role of a CH.

The HEED protocol [3] considers a multihop network and assumes that all nodes are equally important. A node uses its residual energy as the primary parameter to probabilistically elect itself to become a CH. In case of a tie between two CHs, say u and v, u concedes to v (i.e., gives up its CH candidacy) according to a secondary parameter, such as node degree or average distance to neighbors. This results in the uniformly distribution of the elected set of CHs across the network. In HEED, each node executes a constant number of iterations. An implementation of HEED in TinyOS (the operating system for Berkeley motes) showed that clustering and data aggregation at least double the network lifetime.

Kuhn *et al.* [5] proposed a probabilistic technique to elect CHs, in which the probability is dependent on the node degree. The convergence of their proposed technique, which depends on the number of nodes in the network and the node degree, is much faster than iterative techniques. In addition, this approach elects a dominating set of CHs that is asymptotically optimal (minimal).

Table 1 compares representative examples of distributed clustering techniques. (Note that several other techniques were proposed in the literature but are not discussed here due to space limitations.) The table shows that all distributed clustering protocols have a constant message overhead per node. This is an important advantage of the distributed techniques over the centralized ones, which involve significant message overhead. The processing overhead at each sensor, however, is negligible in a centralized approach compared to a distributed one (in which nodes participate in the computations). For example, in iterative approaches, a node has to check the received messages to decide how it should react. These messages can be $O(\delta)$, where δ is the node degree. As energy consumption for data processing is typically lower than that for communication, the message processing overhead may be ignored.

As shown in Table 1, most of the distributed protocols have low overhead. This overhead is dependent on the frequency of reclustering, which is small in typical applications. Throughput is typically not negatively affected by clustering, as shown in an implementation study of HEED [3]. In fact, throughput is improved under high load because of the reduction in channel contention. The performance of each clustering protocol depends on the network configuration, the system model, and the application scenario.

Open Issues

Clustering in WSNs faces several deployment challenges, such as ensuring connectivity, selecting the optimal frequency of CH rotation, computing the optimal cluster sizes, and clustering the network in the presence of a node duty cycle.[1] Addressing these problems will require cross-layer design, especially with regard to the MAC and network layers.

---

[1] *In a typical WSN, a sensor alternates between active and sleep states. The ratio of the active time to the total time is referred to as the node duly cycle.*

Protocol	I/P	Cluster criteria	Objective	Assumptions	Complexity (per node)	
					Time (iter.)	Message
Baker *et al.* [4]	I	Identifier	Minimum dominating set	Unique node identifiers and uniform node distribution	$O(Diam)$	$O(1)$
GAF [8]	I	Position	Redundancy elimination	Topology and field dimensions are known	$O(1)$	$O(1)$
SPAN [9]	I	Two-hop connectivity	Redundancy elimination	Every node has a routing table	$O(1)$	Routing protocol overhead
DCA [11]	I	Weight	Dominating set of higher-weight nodes	Unique, real value weight for every node	$O(Diam)$	$O(1)$
Max-Min D-Cluster [6]	I	Degree	D-hop cluster formation, D is a constant	MAC layer provides collision avoidance	$O(D)$	$O(D)$
ACE [7]	I	Degree	Rapid convergence, denser cluster sizes	Nonuniform load distrubution	$O(1)$	$O(1)$
LEACH [2]	P	CH frequency	Rapid convergence and maximized network lifetime	Single-hop network and uniform load distribution	$O(1)$	$O(1)$
HEED [3]	P	Residual energy	Rapid convergence and mazimized network lifetime	Nonuniform load distribution	$O(1)$	$O(1)$
Kuhn *et al.* [5]	P	Degree	Minimum dominating set	Three independent communication channels	*polylog(n)*	*polylog(n))*

Table 1. *Comparison of representative distributed clustering techniques. I/P denotes whether a protocol is iterative or probabilistic,* Diam *denotes the network diameter, and* n *denotes the number of nodes in the network.*

Connectivity

An important objective of any clustering technique is network connectivity. For intracluster communication, a cluster member communicates with its CH either directly (e.g., [2, 3, 7]) or via multiple hops (e.g., [6, 11]). Connectivity in this case is a result of the success of cluster formation. For intercluster communication, two approaches were adopted in order to maintain connectivity. In one approach (e.g., [6, 10, 11]), nodes on cluster boundaries are used as gateways to relay data among CHs. This approach (depicted in Fig. 3a) is suitable in networks that use a fixed transmission power. Network density has to be sufficiently high in order to ensure that enough gateways are present at the intersection areas between clusters. In another approach (e.g., [2, 3]), the CH overlay constitutes the routing infrastructure, and intercluster routing proceeds only through CHs. This approach (depicted in Fig. 3b) is appropriate if:

- A node can tune its transmission power[2] (e.g., Berkeley motes).
- The CH density and intercluster transmission range satisfy the connectivity conditions specified by Gupta and Kumar [12].

An advantage of this approach is that it enables all non-CHs to sleep while not sensing or transmitting. Selecting the optimal intracluster and intercluster transmission ranges to ensure connectivity and prolong the network lifetime is still an open issue.

MAC Layer Design

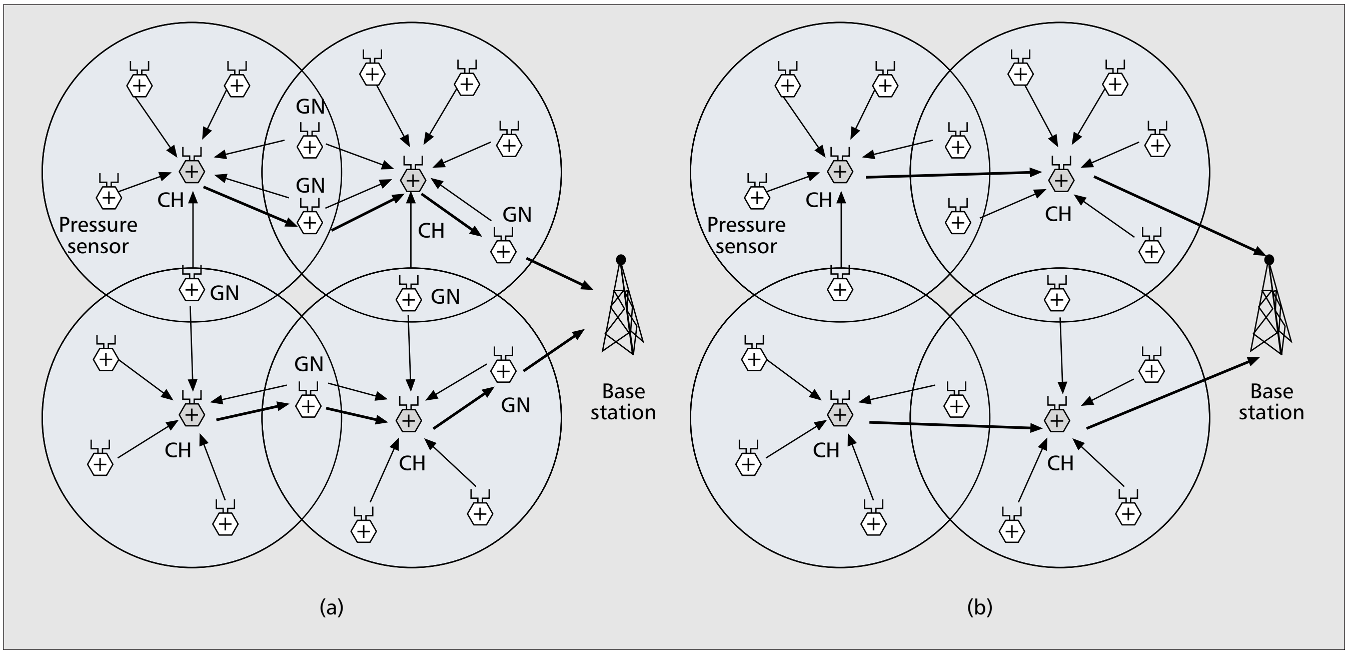
Scheduling intracluster and intercluster transmissions is another important design challenge. Since clustering is typically employed in applications where data aggregation is performed, time division multiple access (TDMA) is best for intracluster transmissions. This is because a CH can set the TDMA schedule and inform its cluster members about it. The problem is how to prevent the TDMA intracluster transmissions from colliding with transmissions in neighboring clusters or with intercluster frames, especially when CHs use longer ranges for communicating with each other. One possible solution is to use orthogonal frequency channels or different CDMA codes to parallelize intracluster and intercluster transmissions. It may not be cost effective, however, to equip cheap sensors with expensive radios in order to support orthogonal channels or codes. Enabling concurrent intracluster and intercluster communications using a single radio channel is an open research issue.

Rotating the Role of Cluster Heads

It is essential to rotate the role of CHs among nodes so as not to burden a few nodes with more duties than others. There are several possibilities for CH rotation. One way is to use a timer expiration to trigger the clustering algorithm, as described in "Node Synchronization" below. Another way is to use a dynamic parameter (e.g., remaining battery) for triggering the clustering algorithm at local regions. For example, a CH might trigger a new CH election process in its local region if its remaining battery lifetime goes below a prespecified threshold. The CH rotation mechanism is typically independent of the clustering protocol.

---

[2] *Note that power control in WSNs is used for switching between CH and non-CH operation modes, and not on a per-packet basis, as proposed for more capable ad hoc networks.*



■ Figure 3. *Intercluster connectivity models: a) routing via gateway nodes; b) routing via CHs only. GN: gateway node.*

It is obvious that more frequent CH rotation results in more clustering overhead and network interruption, while less frequent rotation may cause some nodes to die faster than others. The study of this trade-off is essential for achieving optimal network lifetime. Currently, applications set this rotation frequency heuristically, based on some intuitive factors such as the expected battery lifetime and the node degree.

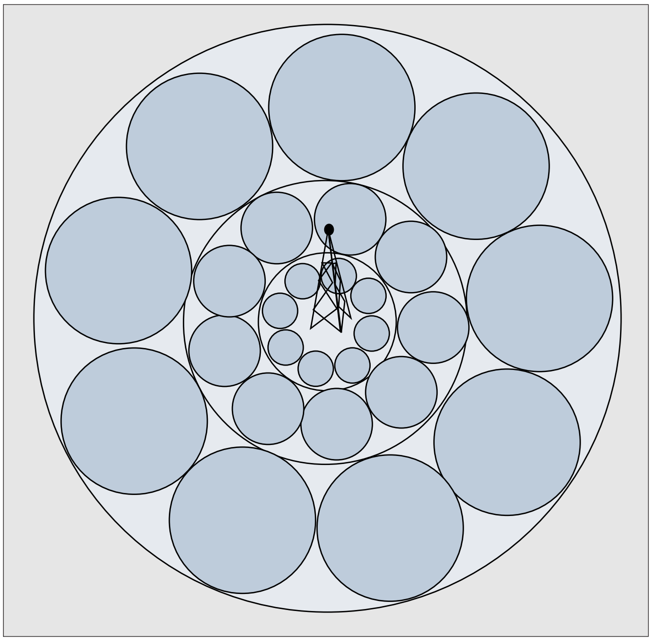
Node Duty Cycle

A primary factor that prolongs the battery lifetime is allowing sensors to sleep when not active. This is due to the following three reasons. First, idle listening consumes significant energy that is comparable to transmission or reception. For example, in a Berkeley Mica2 mote, idle listening consumes energy close to that of reception. In contrast, the energy drainage during sleep time is about three orders of magnitude less than the reception energy consumption. Second, battery discharge is nonlinear, and some of the unusable charges can be restored in the battery after the sleeping period. Third, sensors are typically deployed redundantly, which implies that not all the nodes need to be awake simultaneously.

Therefore, clustering techniques should incorporate the node's duty cycle in their design. This can be done in one of two ways, depending on the type of the application. First, non-CH nodes can be allowed to sleep when they are not sensing or communicating with their CHs. This approach is suitable for applications where sensors are periodically sending reports. Second, if the application requires the sensors to continuously monitor the field for unexpected events, then a CH can determine which of its cluster members are redundant and advise them to turn themselves off. Thus, a CH maintains a minimal active set of nodes in the cluster. It is also possible to elect an active set of nodes to cover the field prior to cluster formation. In this case, the CH and all its members are active during the network operation cycle.

Optimal Cluster Size

Currently, most clustering protocols assume a fixed cluster-transmission range, which results in uniform cluster sizes. However, this results in a skewed load distribution on CHs (e.g., CHs that are closer to the observer will carry more intercluster traffic and hence will deplete their batteries faster than faraway CHs). In order to balance energy consumption among CHs, both intracluster aggregation and the data forwarding responsibilities of a CH have to be considered. Shu *et al.* [13] studied how to achieve this balance by assigning larger cluster sizes to CHs that are responsible for less data forwarding. An example is illustrated in Fig. 4, in which sensors are deployed in a circular region around a base station. This approach is constrained by the maximum possible sensor-transmission range, and the supporting MAC layer. The reader is referred to [13] and the references therein for more information on load- and power-balanced clustering techniques in WSNs. Several issues are still open for research, including how to optimally select cluster sizes without knowl-



■ Figure 4. *Selecting cluster sizes to evenly distribute energy consumption among CHs.*

edge of the node locations and how to exploit knowledge of the locations of data sinks for efficient cluster formation.

Node Synchronization

Distributed clustering protocols achieve their best performance when sensor nodes are synchronized. Node synchronization ensures that the clustering process starts simultaneously throughout the network. Lack of synchronization may result in a suboptimal choice of CHs, especially for probabilistic approaches. However, the clustering process can be triggered by nodes with faster clocks. This happens when such nodes start querying their neighbors for updated information in order to start the clustering process. Received queries trigger the clustering process at these neighbors, and these neighbors in turn trigger their neighbors, and so on. Note that it is not essential that "all" the nodes in the network start the clustering process simultaneously. In fact, it is sufficient that the process starts in different regions (nodes within one or two hops) simultaneously. Probabilistic techniques that use a number of iterations (e.g., [3, 5]) are therefore less impacted by the lack of synchronization than single-iteration techniques (e.g., [2]).

Conclusions

Node clustering is a useful topology-management approach to reduce the communication overhead and exploit data aggregation in sensor networks. We have classified the different clustering approaches according to the clustering criteria and the entity responsible for carrying out the clustering process. We have focused on distributed clustering approaches, which are more suitable for large-scale sensor networks. We highlighted some of the basic challenges that have hindered the use of clustering in current applications. We surmise that the most compelling challenges are how to schedule concurrent intracluster and intercluster transmissions, how to compute the optimal cluster size, and how to determine the optimal frequency for CH rotation in order to maximize the network lifetime.

References

[1] "Habitat Monitoring on Great Duck Island," http://www.greatduckisland.net/, 2006.

[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, Oct. 2002, pp. 660–70.

[3] 0. Younis and S. Fahmy, "Distributed Clustering in Ad Hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004; an extended version appeared in *IEEE Trans. Mobile Comp.*, vol. 3, no. 4, Oct.–Dec. 2004, pp. 366–79.

[4] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Trans. Commun.*, vol. 29, no. 11, 1981, pp. 1694–701.

[5] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing Newly Deployed Ad Hoc and Sensor Networks," *Proc. ACM MOBICOM*, Sept. 2004, pp. 260–74.

[6] A. D. Amis *et al.*, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, Mar. 2000, pp. 32–41.

[7] H. Chan and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation," *Proc. 1st Euro. Wksp. Sensor Networks*, Jan. 2004, pp. 154–71.

[8] Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing," *Proc. ACM MOBICOM*, Rome, Italy, July 2001, pp. 70–84.

[9] B. Chen *et al.*, "SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *ACM Wireless Networks*, vol. 8, no. 5, Sept. 2002, pp. 481–94.

[10] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multihop Wireless Networks," *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1028–37.

[11] S. Basagni, "Distributed Clustering Algorithm for Ad Hoc Networks," *Proc. Int'l. Symp. Parallel Architectures, Algorithms, and Networks*, 1999, pp. 310–15.

[12] P. Gupta and P. R. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks," *Stochastic Analysis, Control, Optimizations, and Applications: A Volume in Honor of W. H. Fleming*, W. M. McEneaney, G. Yin, and Q. Zhang, Eds., Birkhauser, 1998.

[13] T. Shu, M. Krunz, and S. Vrudhula, "Power Balanced Coverage Time Optimization for Clustered Wireless Sensor Networks," *Proc. ACM Int'l. Symp. Mobile and Ad Hoc Networking and Comp.*, May 2005, pp. 111–20.

Biographies

OSSAMA YOUNIS [M] (younis@ece.arizona.edu) is an assistant research scientist in computer engineering at the University of Arizona. He received B.S. and M.S. degrees with honors in computer science from Alexandria University, Egypt, in 1995 and 1999, respectively, and his Ph.D. degree in computer science from Purdue University in August 2005. He has served on the Technical Program Committee for several international conferences, such as AINA 2006, WoWMoM 2006, WWIC '06, and ICWMC '06. His research interests include wireless sensor network protocols and applications, Internet tomography, and video communication. He is a member of the ACM.

MARWAN KRUNZ (krunz@ece.arizona.edu) is an associate professor of electrical and computer engineering at the University of Arizona. He received a Ph.D. degree in electrical engineering from Michigan State University in 1995. From 1995 to 1997 he was a postdoctoral research associate with the Department of Computer Science, University of Maryland, College Park. He also held visiting research positions at INRIA, Sophia Antipolis, France; HP Labs, Palo Alto, CA; and US West Advanced Technologies, Boulder, CO. His recent research interests include medium access and routing protocols for mobile ad hoc networks, quality of service provisioning over wireless links, constraint-based routing, Web traffic modeling, and media streaming. He has published more than 100 journal articles and refereed conference papers in these areas. He received the National Science Foundation CAREER Award (1998–2002). He currently serves on the editorial boards of *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Mobile Computing*, and *Computer Communications Journal*. He has been a guest editor for special issues of *IEEE Micro* and *IEEE Communications Magazine*. He served as technical program co-chair for IEEE INFOCOM 2004, IEEE SECON 2005, IEEE WoWMom 2006, and the 2001 Hot Interconnects Symposium. He consults for a number of corporations in the telecommunications industry.

SRINIVASAN RAMASUBRAMANIAN (srini@ece.arizona.edu) is an assistant professor in the Department of Electrical and Computer Engineering at the University of Arizona. He received a B.E. (honors) degree in electrical and electronics engineering from Birla Institute of Technology and Science, Pilani, India, in 1997, and a Ph.D. degree in computer engineering from Iowa State University, Ames, in 2002. His research interests include architectures and algorithms for optical networks, computer communication networks and protocols, fault tolerance, system modeling, and performance analysis. He served as the TPC Co-Chair for BROADNETS 2005 and is an editor of *Wireless Networks Journal*.