Distributed Formation of Overlapping Multi-hop Clusters in Wireless Sensor Networks

Adel Youssef
Dept. of Computer Science,
University of Maryland College Park,
(adel@cs.umd.edu)

Mohamed Younis
Dept. of Computer Science & Elec. Eng.,
University of Maryland, Baltimore County,
(younis@cs.umbc.edu)

Moustafa Youssef[§], Ashok Agrawala
Dept. of Computer Science,
University of Maryland College Park,
(moustafa, agrawala@cs.umd.edu)

Abstract – **Clustering is a standard approach for achieving efficient and scalable performance in wireless sensor networks. Most of the published clustering algorithms strive to generate the minimum number of disjoint clusters. However, we argue that guaranteeing some degree of overlap among clusters can facilitate many applications, like inter-cluster routing, topology discovery and node localization, recovery from cluster head failure, etc. We formulate the overlapping multi-hop clustering problem as an extension to the k-dominating set problem. Then we propose MOCA; a randomized distributed multi-hop clustering algorithm for organizing the sensors into overlapping clusters. We validate MOCA in a simulated environment and analyze the effect of different parameters, e.g. node density and network connectivity, on its performance. The simulation results demonstrate that MOCA is scalable, introduces low overhead and produces approximately equal-sized clusters.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have numerous applications in a variety of disciplines, both military and civilian [1][2]. The ability to remotely measure ambient conditions and track dynamic targets/events can be invaluable; especially in harsh environments where human intervention is risky or infeasible. In such applications nodes are usually dropped in an area and are expected to self-organize in an ad-hoc manner. Sensors are usually battery-operated and have a limited transmission range and onboard processing and storage capacity. Such constraints have motivated lots of research on effective management strategies of WSNs so that the network can stay functional for the longest duration.

A typical WSN architecture involves numerous sensors that report their measurements to data collection centers; often referred to as sink nodes. In many application setups large number of nodes is employed in order to boost coverage, increase the fidelity of collected data and mitigate potential sensor failure due to energy depletion or damage. Therefore, scalability is an important design attribute for WSNs. Grouping sensors into clusters has been widely pursued as a mean for achieving efficient and scalable performance in WSNs [3][4][5][6]. Each cluster would have a designated head that collects data from local nodes and forwards the aggregated report to the sink; either directly over long distance or through an inter-cluster-head multi-hop path. Clustering facilitates the distribution of control over the network and, hence, enables locality of communication [7]. In addition, clustering nodes into groups saves energy and reduces network contention because nodes send the data over shorter distances to their respective cluster heads [8].

Many clustering protocols have been investigated as either standalone protocols [8][9] or as a side effect of other protocol operations, e.g., in the context of routing protocols [5] and topology management protocols [10]. The majority of those protocols construct clusters where every node in the network is no more than 1 hop away from a cluster head. We call these *single hop* (1-hop) clusters. When the cluster heads are picked from the deployed large sensor nodes population, single hop clustering may generate a large number of cluster heads and eventually may lead to the same problem as if there is no clustering.

In addition, most of the clustering algorithms in the literature have a primary goal of producing approximately equal-sized non-overlapping clusters. Forming equal-sized clusters is a desirable property because it enables an even distribution of control (e.g., data processing, aggregation, storage load) over cluster heads; no cluster head is overburdened or under-utilized. However, having clusters with some degree of overlap is desirable and beneficial in numerous applications (e.g. node localization [11], routing [12], TDMA-based MAC [13]). The *boundary nodes* that belong to two or more clusters can serve as gateways for inter-cluster heads communication when the cluster heads do not have long range communication capabilities. Moreover, overlapped clusters can boost the network robustness against cluster head failure or compromise by facilitating and expediting the recovery of nodes which can join others alternate clusters.

In this paper, we propose a randomized, distributed Multi-hop Overlapping Clustering Algorithm (MOCA) for organizing the sensors into *overlapping* clusters. The goal of the clustering process is to ensure that each node is either a cluster head or within k hops from *at least* one cluster head, where k (*cluster radius*) is a given parameter. A cluster head is a sensor node that is assigned a leadership role. We formulate the overlapping k-hop clustering problem as an extension to the k-dominating set problem [14]. The nodes randomly elect themselves as cluster heads with some probability p. The *cluster head probability* (p) is used to control the number of clusters in the network. Through simulation, we show that MOCA is scalable for large networks. In addition, MOCA incurs low overhead in terms of exchanged messages. To the best of our knowledge, this is the first paper to discuss the problem of overlapping multi-hop clustering.

[§] Also affiliated with Alexandria University, Egypt.

1-4244-0357-X/06/$20.00 ©2006 IEEE

This paper is organized as follows. Section 2 reviews related work in the literature. In Section 3, we discuss the MOCA algorithm in details. Validation results are presented in Section 4. Finally, Section 5 concludes the paper.

II. RELATED WORK

In the last few years, many algorithms have been proposed for clustering in wireless ad-hoc networks. Clustering algorithms can be classified as either deterministic or randomized. Deterministic algorithms, e.g. [3][15][16], use weights associated with nodes to elect cluster heads. These weights can be calculated based on the number of neighbors (node degree) [3], node Id [15], and mobility rate [16]. Each node broadcasts the calculated weight. Then a node is elected as a cluster head if it has the highest weight among its neighboring nodes. In randomized clustering algorithms, the nodes elect themselves as cluster heads with some probability p and broadcast their decisions to neighbor nodes [4][5][6][9]. The remaining nodes join the cluster head that requires minimum communication energy. The probability p is an important parameter in a randomized algorithm. It can be a function of node residual energy [5] or hybrid of residual energy and a secondary parameter [4]. In [6], the authors analytically obtained the optimal value for p that minimizes the energy spent in communication. In MOCA, the probability p is tuned to control the number of clusters.

Recently, a number of clustering algorithms have been designed for sensor networks [4,5,6,7,8,17]. Most of those algorithms aim at generating the minimum number of disjoint clusters that maximize the network lifetime. Both HEED [4] and LEACH [5] form single-hop non-overlapping clusters with the objective of prolonging network lifetime. In [6], the authors proposed a LEACH-like randomized multi-hop clustering algorithm for organizing the sensors in a hierarchy of clusters with an objective of minimizing the energy spent in communicating the information to the processing center. In [7], the authors present a clustering algorithm (FLOC) that produces non-overlapping and approximately equal-sized clusters. FLOC partitions a multi-hop wireless network into clusters of bounded physical radius [R, mR] where m is a constant greater than or equal to 2. That is, each cluster has a header node such that all nodes within distance R of the header belong to the cluster but no node beyond distance mR from the header belongs to the cluster. In [8][17], the clustering algorithm assumes gateway (master) nodes are already known and the objective is to perform load balancing between different clusters by changing cluster radius. None of the above algorithms construct overlapping clusters.

III. PROBLEM FORMULATION

In this section, we formulate the overlapping *multi*-hop clustering problem as an extension to the *k*-dominating set problem. First, we describe the system model.

A. System Model

We consider a wireless sensor network where all nodes are alike and each node has a unique Id. The nodes are location-unaware, i.e. not equipped with GPS. There are neither base stations nor infrastructure support to coordinate the activities of subsets of nodes. Therefore, all the nodes have to collectively make decisions. We assume that the nodes are stationary, which is typical for sensor networks. All sensors transmit at the same power level and hence have the same transmission range (T_r). We also assume that nodes have timers, but we do not require time synchronization across the nodes. Timers are used for timing out when a node is waiting on a condition.

All communication is over a single shared wireless channel. A wireless link can be established between a pair of nodes only if they are within the radio range of each other. The MOCA algorithm only considers bidirectional links. It is assumed that the MAC layer will mask unidirectional links and pass bidirectional links to MOCA. Two nodes that have a wireless link are, henceforth, said to be 1-hop away from each other or immediate neighbors. Nodes can identify neighbors using beacons.

B. The Overlapped K-hop Clustering Problem

An ad-hoc network can be modeled as a graph $G = (V, E)$, where two nodes are connected by an edge if they can communicate with each other. Since all nodes are located in the plane and have the same T_r, G is *unit disk graph*. We define $N_k[u]$ as the set of nodes that are reachable to a node u in at most k hops including u itself. A *k-Dominating Set* (KDS) S is a subset of V such that every node in $(V - S)$ is reachable to at least one node in S within or less than k hops. Finding a KDS is an NP-Hard problem [14]. Given an ad-hoc network that is modeled as a unit disk graph, the overlapped *k*-hop clustering problem can be formulated as finding the set of nodes S that satisfy the following two conditions:

1. *Coverage Condition.* S is a KDS; means that each node is either a cluster head or within k hops from a cluster head.
2. *Overlapping Condition.* For each node $u \in S$ $\exists$ at least one node $v \in S$ such that $N_k[u] \cap N_k[v] \geq 1$. In other words, for each cluster, there exists at least one other cluster that overlaps with it with overlapping degree ≥ 1.

The problem of overlapping clusters is totally new. There is no formulation of the problem in the literature and no known algorithm that satisfies these two conditions. The proposed MOCA (Multi-hop Overlapping Clustering Algorithm) is a distributed simple randomized algorithm that meets the above two conditions with high probability. MOCA pursues heuristics with the objective of decreasing processing and message complexity in order to meet the requirements of wireless sensor networks. We will show that by tuning some of the algorithm parameters (k, p, T_r), we can generate overlapping clusters with some average overlapping degree with high probability. The cluster head probability (p) will be tuned to control the *coverage condition* and the cluster radius (k) and node transmission range (T_r) will be used to control the overlapping degree between adjacent clusters.

IV. DETAILED MOCA ALGORITHM

In this section, we describe the detailed clustering process. First we clarify the notation and define some data structures to be maintained at each node.

A. *Notation and Data Structure*

We use the following notation in describing MOCA:

- *NID*: A unique node Id assigned prior to deployment.
- *Status*: A node status can be either a cluster head (CH) or a non-CH (NCH). Initially all nodes are set to NCH.
- Adjacent Clusters Table (*AC_table*): A table maintained by CH nodes to store information about adjacent clusters. The table consists of tuples of the form (*CHID*, *BN*), where *CHID* is the CH node Id, and *BN* is a list of *boundary node* Ids.
- Cluster Heads Table (*CH_table*): A table maintained by each node to store information about the clusters known to this node. If the table contains more than one entry, this means that the node is a *boundary node*. The table consists of tuples of the form (*CHID*, *HC*, *prev*), where *CHID* is the CH node Id, *HC* is the number of hops leading to this cluster head, and *prev* is the node ID of a 1-hop neighbor node that can lead to the CH node of this cluster using minimum number of hops. In essence, *CH_table* acts as a routing table where the *CHID* field uniquely identifies shortest path to a CH node.

B. *Cluster Head Selection*

The essential operation in any clustering protocol is to select a set of cluster heads among the nodes in the network, and group the remaining nodes around these heads. MOCA does this in a distributed fashion, where nodes make autonomous decisions without any centralized control. The algorithm initially assumes that each sensor in the network becomes a cluster head (CH) with probability p. The probability p is determined a priori based on the network size. Each cluster head then advertises itself as a cluster head to the sensors within its radio range. This advertisement is forwarded to all sensors that are no more than k hops away from the CH. The advertisement (CH_AD) message's header include *SID*, *CHID*, and *HC*; where *SID* is the sender node ID, *CHID* is cluster head ID, and *HC* is the number of hops leading to the CH node. The *SID* field is used to update the *CH_table.prev* field such that each node knows the path to the cluster head. The *HC* field is used to limit the flooding of the CH_AD message to k hops.

As we explain later, a sensor that receives such advertisements joins the cluster even if it already belongs to another cluster. Since the advertisement forwarding is limited to k hops, if a sensor does not receive a CH advertisement within time duration, it can infer that it is not within k hops of any cluster head and hence become a CH. In MOCA, the maximum time that a node should wait for CH advertisement messages is set to $t(k) + \delta$, where $t(k)$ is the time needed for a message to travel k hops and δ is the maximum time needed for any node to finish bootstrapping and start the clustering process.

C. *Clusters Membership*

Each node maintains a table, *CH_table*, that stores information about the clusters it knows. Upon receiving a new CH_AD message, a node will add an entry in its *CH_table*. In case a similar message was received, the node will check the hop count, i.e. the *HC* field in the recent message, and will then update *HC* and *prev* fields in the corresponding entry in the *CH_table* if the recent message came over a shorter path. Often a message traveling the shortest path in terms of the number of hops would arrive first. However, delay may be suffered at the MAC or link layer. If *CH_table* contains more than one entry, this means that the node is a *boundary node*.

For every entry in its *CH_table*, a node sends a join request (*JREQ*) message to the CH in order become a member of the corresponding cluster. To limit the flooding, the message is unicasted using the field *CH_table.prev*. The *JREQ* message has the form [*JREQ*, *RID*, *SID*, *CHID*, *nc*, $(CHID)_{0..nc}$] where *RID* is the receiver node Id (i.e. *CH_table.prev)*, *SID* is the Id of the node that will join the cluster, *CHID* is the Id of the CH node responsible for this cluster, *nc* is the number of clusters that this node can hear from them (=|*CH_table*|), and $(CHID)_{0..nc}$ are 0 or more clusters that this node can hear from.

Each cluster head maintains a list of all cluster members, a list of adjacent clusters, and a list of boundary nodes to reach those clusters along with the maximum hop count to reach the adjacent cluster. There can be multiple boundary nodes between overlapping clusters. Moreover, a node can be a boundary node for more than two overlapping clusters. The CH node also will enforce a time-out for *JREQ* which is set in MOCA to $c * t(k) + \delta$; where c is a constant that depends on the MAC protocol, node density and the value p. A finite state machine for the MOCA protocol is given in Fig. 1. Analytical formulation for the convergence rate and complexity of MOCA can be found in [18].

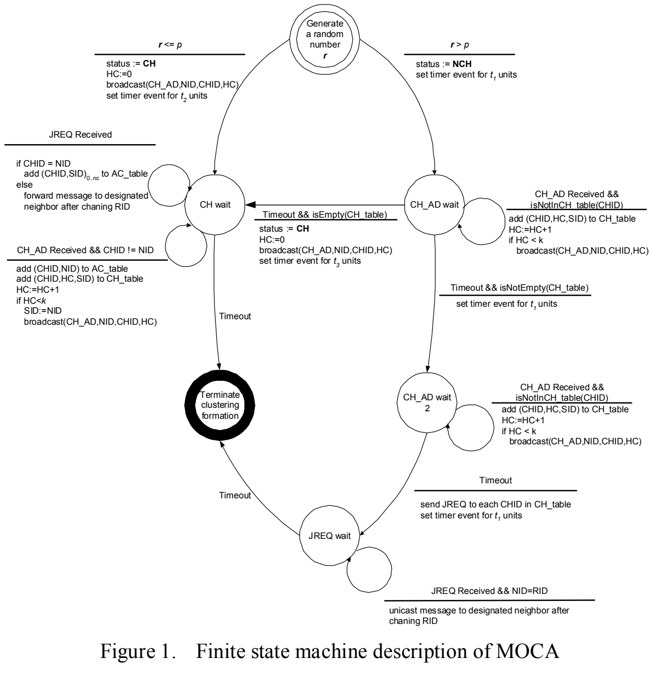

Figure 1. Finite state machine description of MOCA

V. SIMULATION EXPERIMENTS

We have implemented the MOCA clustering algorithm using MATALB 6.1 release 12.1 and validated it using simulation. In this section, we discuss the experiment setup and results.

A. Parameters, Objective and Metrics

There are four parameters used in our simulation:

- *Network size* (*n*): the number of sensor nodes in the network. Since all the simulation experiments assume a square area of side length *l*, changing the network size will implicitly change the node density in the network ($\mu = n/l^2$).
- *Cluster radius* (*k*): the maximum path length, i.e. hop count, between any node in a cluster *i* and the cluster head CH_i.
- *Average node degree* (*ND*): The degree of a node *u* is the number of nodes that are neighbors to *u*. Node degree is a function of the node transmission range (T_r). Assuming that *n* sensor nodes are uniformly distributed over a square field of side *l*, the probability *P(ND)* of a node *u* having degree *ND* obeys binomial distribution with the probability *Pr* of being within the transmission range T_r of a node *u* equals $\pi(T_r / l)^2$ [19]. For very large *n*, the binominal distribution converges to a Poisson distribution with $\lambda = n \times Pr$. Hence, the relation between *ND* and T_r is given by:

$$ND = n\,\pi\,(T_r / l)^2 = \mu\,\pi\,T_r^{\,2}$$

- *The cluster head probability* (*p*): Since each node decides randomly to be a cluster head with probability p, raising the value of p will increase the number of clusters.

To evaluate the performance of the MOCA clustering algorithm, we use the following performance metrics:

- *Percentage of Covered Nodes* (*CN*): this metric tests if the generated clusters satisfy the coverage condition as defined in subsection 3.2. CN is defined as the percentage of nodes that are either cluster heads or within k-hops from a cluster head after the first wave of CH advertisement is propagated though the network (i.e. after t(k) time units where t(k) is the time needed for a message to be forwarded for k hops).
- *Average Overlapping Degree* (*AOD*): This metric checks whether the generated clusters satisfy the overlapping condition in as subsection 3.2. AOD is defined as the average overlapping degree between any two overlapping clusters in the network. Assume that u, v are arbitrary cluster heads. Then the overlapping degree between the two corresponding clusters is $|Nk[u] \cap Nk[v]|$. We would like to note that the overlapping degree is defined only for overlapping clusters.
- *Average Cluster Size* (*Nc*): the average number of nodes per cluster is the average $|Nk[u]|$ for an arbitrary cluster head u. We use this metric to show that MOCA generates nearly equal-sized clusters, which is a desirable property to balance the load of control overhead between cluster head nodes.
- *Communication Overhead*: This metric assesses the total energy spent in communication. Without loss of generality, it is assumed that the cost of transmitting 1 unit (byte) of data is 1 unit of energy. This is a valid assumption since we assume that all the nodes have a fixed transmission range.

Our main goals behind the simulation experiments are: (1) to show that with the careful selection of input parameters (*p*, *k*, *ND*), the proposed clustering algorithm meets the conditions listed in subsection 3.2 with high probability; (2) to show that although we have overlapped clusters, MOCA still produces approximately equal-sized clusters; (3) to show that MOCA is scalable in terms of communication overhead. Since each of the above protocol parameters has a different effect on one of the performance metrics, we wanted to give a sensor network engineer a set of parameters to tune in order to achieve different design goals (e.g. minimize power consumption by varying the node transmission range, increase overlapping degree, reduce cluster size, increase inter-cluster connectivity, reduce number of clusters, etc.).

B. Experiment Setup and Results

All experiments were performed over 150 different topologies representing different network sizes (*n*) ranging from 50 to 800 sensor nodes. The nodes were randomly placed according to a uniform distribution on a square area. For each topology, the transmission range of each node (T_r) was varied in order to achieve different average *node degree* (ND) ranging from 7 to 21. In a wireless ad-hoc network with a uniform distribution of nodes, it has been shown that the average node degree should be at least 6 in order to guarantee global network connectivity [20]. Hence, we chose the minimum average node degree to be 7. We also assumed that all the CH nodes will finish bootstrapping and start transmitting CH_AD messages within 2 time units (i.e. we set δ to 2). In all experiments, the cluster radius (*k*) ranges from 1 to 5. The cluster head probability (*p*) was varied from 0.05 to 0.5. For each topology, since cluster heads are chosen randomly, we repeat the experiment 30 times, each time with a different random set of cluster head. We observed that with 95% confidence level, the simulation results stay within 2-6% of the sample mean. The error curves can be found in [18].

We start by studying the effect of cluster head probability (*p*) on the percentage of covered nodes (*CN*). From Fig. 2, we can see that raising *p* increases the coverage almost exponentially with a boost in coverage for larger *k*. It is also clear that for each *k*, there is a minimum value for *p* that guarantees 100% coverage with high probability. We have observed similar effect when the node degree was changed [18]. Fig. 3 confirms the effectiveness of MOCA showing that almost no cluster will be isolated (orphaned) without being overlapped by some other clusters.

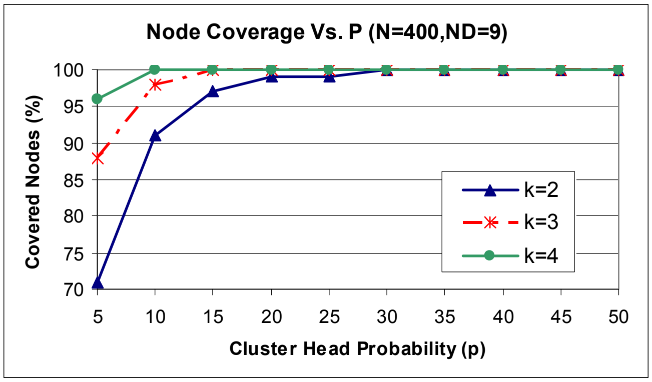

Figure 2. Effect of cluster radius (k) on percentage of covered nodes

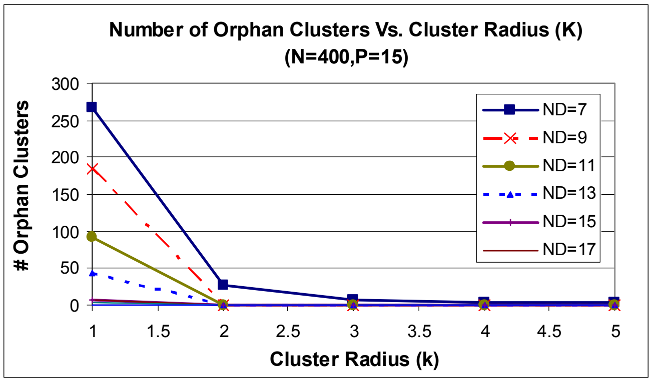

Figure 3. Number of clusters that do not overlap with others

Fig. 4 shows an interesting anomaly for the average overlapping degree between clusters. Although one may think that increasing *p* (i.e. increasing number of cluster heads) should increase the average overlapping degree (*AOD*), the results showed that *p* has no effect on *AOD* regardless of the values of other parameters (*ND*, *k*) and network size (*n*). The explanation is that higher values of *p* increases the number of clusters, and also the overlapping area between them, while at the same time increases the number of pair wise intersections between clusters. These two factors change with the same rate and hence the *AOD* stays constant. Such observation is further validated analytically in [18].

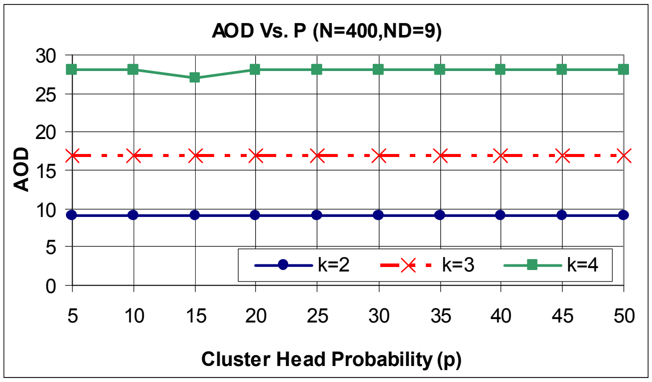

Figure 4. The cluster head probability (*p*) has no effect on the average overlapping degree (*AOD*)

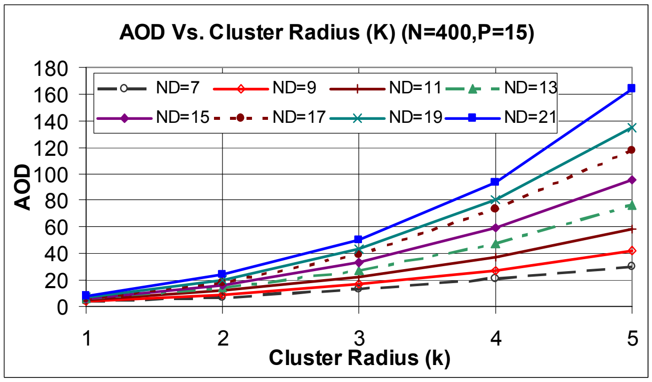

Figure 5. Effect of *k* on the average overlapping degree (*AOD*)

As shown in Fig. 5, increasing the cluster radius (*k*) will increase the *AOD* almost quadratically. The *AOD* also appears to be proportional to *ND* (we have analytically shown it to a linear relation [18]). It is worth noting that for many applications, an *AOD* below 10 would suffice. For example in localization, an *AOD* of 3 is enough and in routing protocols having 10 gateway nodes between clusters is more than enough. It is clear that we can guarantee an *AOD* of more than 10 with high probability using small *ND* (i.e. low transmission range) and small cluster radius (k = 2).

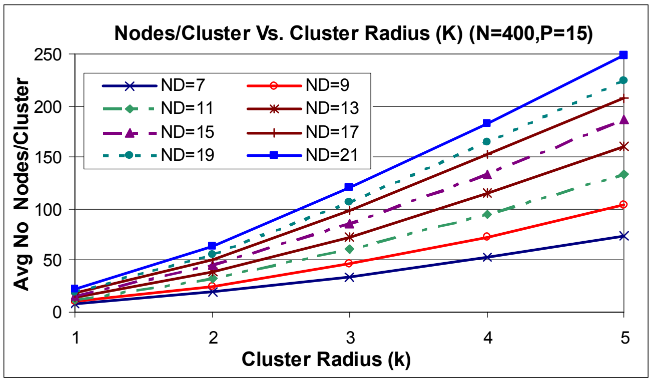

Figure 6. Average number of nodes per cluster (*Nc*) as *k* increases

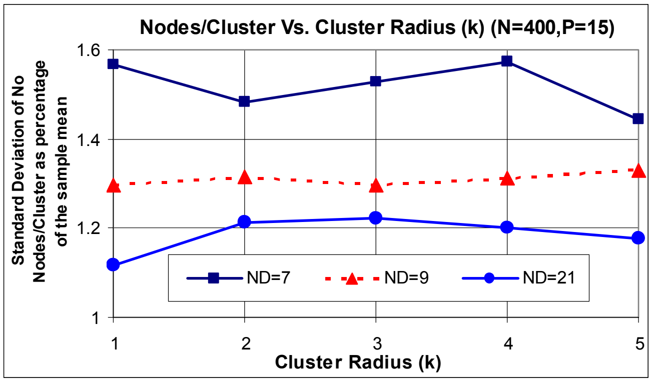

Figure 7. The standard deviation of the average cluster size (*Nc*)

Although the MOCA protocol generates overlapping cluster, the simulation results indicated that those clusters are nearly equal-sized that grows linearly with *k* (Fig. 6). Equal-sized clusters is a desirable property because it enables an even distribution of control (e.g., data processing, aggregation, storage load) over cluster heads; no cluster head is overburdened or underutilized. The standard deviation of average number of nodes per cluster is shown in Fig. 6. The figure shows very low standard deviation regardless of the values of *ND* and *k*. Moreover, the results show that the average cluster size can be controlled by tuning the average node degree (*ND*) or the cluster radius *k*.

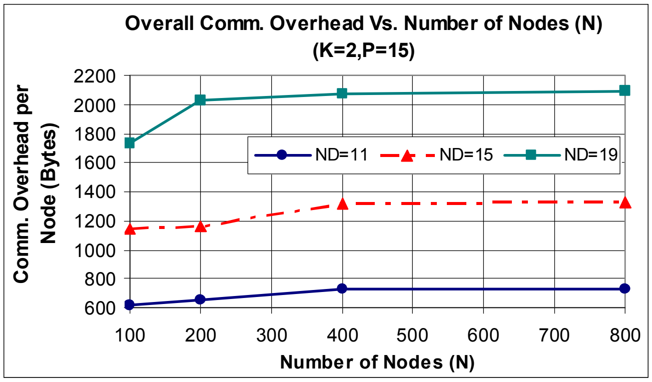

Figure 8. The impact of network size (*n*) on the communication overhead incurred per node

Finally, we will show that MOCA is scalable in terms of communication overhead. We tested the MOCA protocol for different network size ranging from 50 to 800 nodes. Fig. 8 shows the overall communication overhead per node as network size increases. We can clearly see that the number of bytes transmitted by a node slowly increase as the network size increases from 100 to 400. Then it remains almost constant afterwards.

VI. CONCLUSION

Many applications of sensor networks involve large number of nodes. Grouping nodes into clusters and applying hierarchical management strategies are commonly pursued to achieve scalability. Most of the published clustering algorithms strive to form non-overlapping clusters that meet some additional objectives such as load balancing, minimizing communication energy, etc. In this paper, we have presented MOCA; a scalable randomized multi-hop clustering protocol for ad-hoc sensor networks. Unlike contemporary schemes, MOCA organizes the sensors into overlapping clusters. Having overlapping clusters is beneficial in node localization, ensuring inter-cluster connectivity, etc. Moreover, overlapped clusters can boost the network robustness against cluster head failure or compromise by facilitating and expediting the recovery of nodes which can join others alternate clusters.

We have formulated the overlapping multi-hop clustering problem as extension to the *k*-dominating set problem and proposed a distributed heuristics in which nodes randomly nominates themselves as cluster head based on a preset probability. We validated MOCA in a simulated environment. The simulation results have shown that MOCA is scalable in terms of communication overhead and achieves high node coverage. The experiments have demonstrated that MOCA's parameters, such as cluster radius, average node degree, and cluster head probability can be easily tuned to achieve the design goals with high probability.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, Vol. 38, pp. 393-422, 2002.

[2] C-Y. Chong and S.P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, 91(8), pp. 1247-1256, 2003.

[3] S. Basagni, "Distributed Clustering Algorithm for Ad-hoc Networks," in the *Proc. of the International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, Fremantle, Australia, June 1999.

[4] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. on Mobile Computing*, 3(4), pp. 366- 379, Oct.-Dec. 2004.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Application specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Networking*, 1(4), October 2002.

[6] S. Bandyopadhyay and E. Coyle, "An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," in the Proceedings of *IEEE INFOCOM*, San Francisco, CA, March 2003.

[7] M. Demirbas, A. Arora, and V. Mittal, "FLOC: A Fast Local Clustering Service for Wireless Sensor Networks," in the *Proceedings of the 1st Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, Florence, Italy, June 2004.

[8] S. Ghiasi, et al., "Optimal Energy Aware Clustering in Sensor Networks," *Sensors Magazine*, Vol. 1, pp. 258.269, January 2002.

[9] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks," in the *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, April 2001.

[10] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Con_guring Sensor Networks Topologies," in the *Proceedings of IEEE INFOCOM*, New York, NY, June 2002.

[11] X. Ji, "Sensor Positioning in Wireless Ad-hoc Sensor Networks with Multidimensional Scaling," in the *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[12] P. Krishna, et al., "A Distributed Routing Algorithm for Mobile Wireless Networks," *ACM SIGCOMM Computer Communications Review*, 1997.

[13] T. Wu and S. K. Biswas, "A self-reorganizing slot allocation protocol for multi-cluster sensor networks," in the Proceedings of the 4th *International Conference on Information Processing in Sensor Networks (IPSN '05)*, April 2005.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. Freeman, San Frncisco, 1978.

[15] A. D. Amis, et al., "Max-Min Dcluster Formation in Wireless Ad Hoc Networks," in the Proceedings of *IEEE INFOCOM*, March 2000.

[16] M. Chatterjee, S. K. Das, and D. Turgut. WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks. *Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking*, Vol. 5, pp.193.204, 2002.

[17] G. Gupta, M. Younis, "Load-Balanced Clustering in Wireless Sensor Networks," in the *Proceedings of the IEEE International Conference on Communication (ICC 2003)*, Anchorage, Alaska, May 2003.

[18] A. Youssef, et al., "The Overlapped K-hop (OK) Clustering Algorithm," Technical Report CS-TR-4735, Department of Computer Science, University of Maryland College Park, July 2005.

[19] Andreas Savvides, C. C. Han, and M. Srivastava, "Dynamic Fine-Grained Localization in Ad-hoc Networks of Sensors," in *the Proceedings of ACM Mobicom*, Rome, Italy, July 2001.

[20] B. Krishnamachari, S. Wicker, and R. Bejar, "Phase Transition Phenomena in Wireless Ad-hoc Networks," in the *Proceedings of IEEE GLOBECOM*, San Antonio, TX, 2001.