Adaptive Distributed Topology Control for Wireless Ad-Hoc Sensor Networks

Kai-Ting Chu*, Chih-Yu Wen*, Yen-Chieh Ouyang*, and William A. Sethares†

***Abstract*— This paper presents a decentralized clustering and gateway selection algorithm for wireless ad-hoc sensor networks. Each sensor uses a random waiting timer and local criteria to determine whether to form a new cluster or to join a current cluster and utilizes the messages transmitted during hierarchical clustering to choose distributed gateways such that communication for adjacent clusters and adaptive distributed topology control can be achieved. The algorithm operates without a centralized controller, it operates asynchronously, and does not require that the location of the sensors be known a priori. A performance analysis of the topology management and the energy requirements of the algorithm are used to study the behaviors of the proposed algorithm. The performance of the algorithm is described analytically and via simulation.**

I. Introduction

Without a robust infrastructure, sensors in an ad-hoc network may be required to self-organize. Such sensor networks are self-configuring distributed systems and, for reliability, should also operate without centralized control. In addition, because of the limited energy source, energy-efficiency is a critical consideration.

There has been extensive research on the design and development of energy efficient networking techniques. In [1], the Low-Energy Adaptive Clustering Hierarchy (LEACH) utilizes a randomized periodical rotation of clusterheads to balance the energy load among the sensors. LEACH-C (Centralized) [2] uses a centralized controller to select clusterheads. The main drawbacks of this algorithm are nonautomatic clusterhead selection and the requirement that the position of all sensors must be known. LEACH's stochastic algorithm is extended in [3] with a deterministic clusterhead selection. Simulation results demonstrate that an increase of network lifetime can be achieved compared with the original LEACH protocol. The Ad hoc Network Design Algorithm (ANDA) [4] maximizes the network lifetime by determining the optimal cluster size and the optimal assignment of sensors to clusterheads but requires a priori knowledge of the number of clusterheads, number of sensors in the network, and the location of all sensors. The Weighted Clustering Algorithm (WCA) [5] considers the number of neighbors, transmission power, mobility, and battery usage in choosing clusters. It limits the number of sensors in a cluster so that clusterheads can handle the load without degradation in performance. These clustering methods rely on synchronous clocking for the exchange of information among sensors which typically limits these algorithms to smaller networks [6]. In [7], the distributed topology control using the cooperative communication (DTCC) algorithm is proposed to provide a connected network topology with minimal total energy consumption.

In order to provide reliable communication in wireless ad-hoc networks, maintaining network connectivity is crucial [8]-[15]. An implementation of the linked cluster architecture may consider the following tasks: cluster formation, cluster connectivity, and cluster reorganization. In order not to rely on a central controller, clustering is carried out by adaptive distributed control techniques via random waiting timers. To this end, the *Adaptive Distributed Topology Control Algorithm* (ADTCA) forms clusters and links in three phases: (I) clusterhead selection; (II) gateway selection, and (III) cluster reformation. In Phase I, clusterheads are selected and cluster members are assigned. A decentralized algorithm [8] is used to organize the network into clusters. Each sensor operates independently, monitoring communication among its neighbors. Based on the number of neighbors and a randomized timer, each sensor either joins a nearby cluster, or else forms a new cluster with itself as clusterhead. In Phase II, based on bidirectional message exchanges and the cluster architecture, sensors are selected as gateways in a fully distributed way. Once the network topology is specified (as a hierarchical collection of clusters and distributed gateways), maintenance of the linked cluster architecture becomes an issue. In Phase III, localized criterions governing cluster reformation are described and illustrated via simulations.

This proposed self-configuration protocol is energy efficient, scalable, and may extend the lifetime of the network. Several aspects of this cluster-based topology control (such as the time synchronization problem and efficient network routing) are studied. A performance analysis and simplified models of the algorithm are derived, and the results are compared to the behavior of the algorithm in a number of settings.

II. The Adaptive Distributed Topology Control Algorithm (ADTCA)

This section describes a randomized distributed algorithm that forms clusters and reselects clusterheads efficiently. The network setup is performed in three phases: "clustering," "selecting gateways," and "restructuring the clusters." The main assumptions on the network are that (a) the sensors are in fixed but unknown locations, (b) all links between sensors are bidirectional, and (c) all sensors have the same transmitting

*The Department of Electrical Engineering, Graduate Institute of Communication Engineering, National Chung Hsing University, Tai-Chung, Taiwan. (email:cwen@dragon.nchu.edu.tw)

†The Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA (email:sethares@ece.wisc.edu)

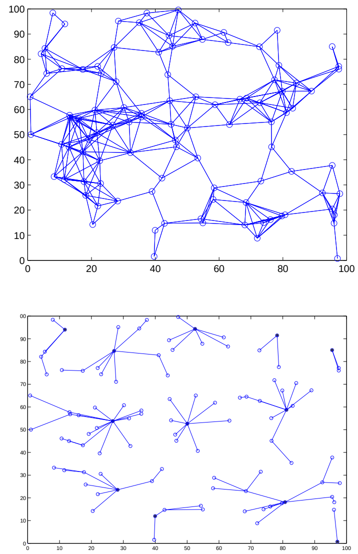

Fig. 1. The connectivity of the network (top); clusters are formed in a random network of 100 sensors with $R/l = 0.17$ (bottom).

range. Observe that there is no base station or centralized control to coordinate or supervise activities among sensors.

A. Phase I: Clusterhead Selection

When sensors of a network are first deployed, they may apply the Clustering Algorithm via Waiting Timer (CAWT) from [8] to partition the sensors into clusters using the waiting timer

$$WT_i^{(k+1)} = \gamma \cdot WT_i^{(k)}, \quad (1)$$

where $WT_i^{(k)}$ is the waiting time of sensor *i* at time step k and $0 < \gamma < 1$ is inversely proportional to the number of neighbors. If the random waiting timer expires and none of the neighboring sensors are in a cluster, then sensor *i* declares itself a clusterhead. It then broadcasts a message notifying its neighbors that they are assigned to join the new cluster with ID *i*.

After applying the CAWT, there are three different kinds of sensors: (1) the clusterheads (2) sensors with an assigned cluster ID (3) sensors without an assigned cluster ID, which will join any nearby cluster after τ seconds and become 2-hop sensors, where τ is a constant chosen to be larger than all of the waiting times. In this phase, each sensor initiates 2 rounds of local flooding to its 1-hop neighboring sensors, one for broadcasting sensor ID and the other for broadcasting cluster ID, to select clusterheads and form 2-hop clusters. Hence, the time complexity is $O(2)$ rounds. Figure 1 shows the network connectivity and cluster formation of a random network of 100 sensors with $R/l = 0.17$, where R/l is the ratio of transmitting range *R* to the side length *l* of the square. Thus, the topology of the ad-hoc network is now represented by a hierarchical collection of clusters.

B. Phase II: Gateway Selection

Observe that Phase I induces nonoverlapping clusters. Accordingly, to interconnect two adjacent nonoverlapping clusters, one cluster member from each cluster must become a gateway. This subsection presents a method of choosing distributed gateways for adjacent nonoverlapping clusters. As in Phase I, random waiting times and local information are applied to select gateways and further achieve communication between clusters. The result of the phase II processing is that each cluster i assigns a single member to communicate with each nearby cluster j. The waiting timers help to ensure that the chosen member is one of the nearest members even though the topology of the system is unknown. If the clusters are too far apart (outside the range of communication R), no gateway sensors will be assigned.

At the beginning of the task, clusterheads broadcast messages to trigger the gateway selection process. After applying the procedure for determining gateways, the gateway nodes broadcast messages to update the connectivity information and activate the linked cluster architecture. The procedure for choosing gateways is summarized in Table I. Let n_i denote sensor *n* in cluster *i* and m_j denote sensor *m* in cluster *j*. G_{ij} will denote the gateway sensor that connects cluster *i* to cluster *j*. $d_{n_i m_j}$ is the distance between sensors n_i and m_j, which could be estimated by received signal strength. The parameter β controls the rate at which the timers increase or decrease in response to the reception of messages from nearby sensors. Note that in gateway selection (in step d)(1)), a clusterhead may be able to communicate with a nearby cluster directly. Therefore, a larger counter (10β) is assigned to the clusterhead in order to be selected as a gateway in this case. Otherwise, each cluster member follows a regular control rate β to increase the counter and decrease the waiting time.

C. Phase III: Cluster Reformation

This subsection presents two methods of choosing a new clusterhead for an existing cluster. If the energy E_i of clusterhead *i* is less than a threshold level η, then sensor *i* broadcasts a message to its cluster members to start the reselection process. Only those sensors with energy larger than η are eligible.

The first method is a centralized technique that the current clusterhead, sensor i, determines a new clusterhead by aggregating energy and neighbor information from its cluster members and solving the optimization problem:

$$\arg\max_{l} \quad (1 - \frac{E_l^{(k)}}{E_l^{max}})^{N_l} \quad (2)$$

$$\text{subject to}: \quad E_l > \eta;\ l \in C_i, \quad (3)$$

where $E_l^{(k)}$ is the energy at time step *k*, E_l^{max} is the initial energy of sensor *l*, C_i is the index set of the cluster members of sensor *i*, and N_l is the number of neighbors of sensor l. That is, the current clusterhead picks the new clusterhead, choosing a member with large energy and many neighbors.

The second method is a distributed technique, which operates much like the CAWT in utilizing a random timer. Once the energy in the current clusterhead is below the threshold,

TABLE I

DESCRIPTION OF GATEWAY SELECTION.

a) Based on the cluster formation in Phase I, each sensor broadcasts its cluster ID information.

b) Initialize a vector of random waiting times $WT_{ij}^{(n_i,k)}$, where $WT_{ij}^{(n_i,k)}$ is the waiting time of sensor n_i for cluster *j* at time step *k*.

c) Initialize a counter of sensor n_i, $C_{ij}^{(n_i)} = 0$, for gateway selection in cluster *i* to cluster *j*.

d) If sensor n_i receives a message from sensor m_j.

(1) increase the counter

if n_i is a clusterhead

$C_{ij}^{(n_i)} = C_{ij}^{(n_i)} + 10\beta$.

else

$C_{ij}^{(n_i)} = C_{ij}^{(n_i)} + \beta$.

end

where $C_{ij}^{(n_i)}$ is the counter of sensor n_i for cluster *j*,

$\beta = \alpha(1 - \frac{d_{n_i m_j}}{R})$ with a positive integer α,

$d_{n_i m_j}$ is the distance between sensors n_i and m_j, and

R is the transmission range.

(2) decrease the waiting time

$WT_{ij}^{(n_i,k+1)} = WT_{ij}^{(n_i,k)} - C_{ij}^{(n_i)}$.

e) Gateway check:

if $WT_{ij}^{(n_i,k)} = 0$

(1) assign $G_{ij} = n_i$, and then G_{ij} broadcasts the gateway information to its neighbors.

(2) set $C_{ij}^{(x_i)} = 0$ and stop the waiting timer for all neighboring sensors x_i in cluster *i*.

else

go to step d).

end

it transmits a message to start the reselection process. Each cluster member then checks the energy constraint. As long as the cluster member satisfies the constraint, it generates a random waiting time:

$$WT_i^{(k+1)} = (1 - \frac{E_i^{(k)}}{E_i^{max}})^{N_i^c} \cdot WT_i^{(k)}, \tag{4}$$

which depends on the number of neighboring cluster members N_i^c and the remaining energy level. The motivation for forming subclusters is to provide a way to do multi-hop communication within a cluster, which may be needed because sensors are no more than 2 hops away from the initial clusterhead and sensors may be up to 4 hops away from the new clusterhead. Hence, sensors in a cluster may be further classified as: (1) subcluster member, (2) subclusterhead, or (3) clusterhead. Subclusters and subclusterheads are generated by applying this distributed protocol to the cluster topology.

For real applications, it is possible that the clusterhead may malfunction before broadcasting the reselection message. One solution is that if a certain amount of time has passed with no messages from the clusterhead, then all sensors begin their timers and apply the algorithm. As a result, restructuring the cluster formation of the network may be required when the clusterhead malfunctions or when none of the cluster members satisfy the energy constraint. In this case, it may necessary to re-initialize the network into new clusters to help balance the energy burden. Such reformation may also be useful in the event that the network topology changes or the sensors move.

III. Performance Analysis

Because of the complexity of the ADTCA, it is difficult to evaluate the algorithm directly other than via simulation. Since the connectivity among sensors and the number of neighboring sensors play important roles in the ADTCA, it is reasonable to investigate the performance from the perspective of these parameters. The performance analysis of cluster formation (Phase I) is derived in [8]. In this section, we abstract the behavior of gateway selection (Phase II) protocol using two simplified models which approximate the desired global behavior and serve to analyze its performance.

A. The Density Model

The first simplified model is the Density Model which is detailed in Table II. The basic idea of this model is to suppose that the probability of sensor n_i in cluster *i* of being a distributed gateway to cluster *j*, $p_{ij}^{(n_i)}$, is proportional to the number of the neighboring sensors which belong to cluster *j*, $N_{ij}^{(n_i)}$. That is,

$$p_{ij}^{(n_i)} \propto \frac{N_{ij}^{(n_i)}}{M_j}, \tag{5}$$

where M_j is the number of sensors in cluster *j*.

If sensor n_i and its neighboring sensors are not already chosen as a gateway to cluster *j*, then the sensor with the largest $p_{ij}^{(n_i)}$ is chosen to be a gateway and it assigns probability 0 to its neighbors which have not yet become a gateway to cluster *j*. Thus, a sensor becomes a gateway to cluster *j* if it has the highest neighboring density among all sensors which have not yet become gateways. After updating the probability distribution of sensors, the procedure repeats until all gateways are chosen. The rationale for this choice is that, if the random waiting time of each sensor is long enough (in the sense that each sensor is able to collect sufficient neighboring information), then the model is likely to closely approximate the behavior of Phase II in the ADTCA on any given ad-hoc network. The close connection between the model and the algorithm is explored via simulation.

B. The Distributed Randomized Model

Since a cluster is a small network, the behavior of the algorithms may be analyzed (following our results in [8]) by the Averaged Model to investigate and describe the clustering behavior. Moreover, gateway selection is highly related to the cluster formation such that distributed gateways can be applied to connect adjacent clusters, which implies that the number of gateways in a network may be induced by a probabilistic model with the number of clusterheads and cluster-based network topology.

1) Overview of The Averaged Model: The CAWT can be modeled by a simplified averaging procedure. Assume that a single clusterhead and an average number of neighboring sensors $E^{(k)}[N_i]$ are removed during each iteration k. Assume that each sensor will be removed with probability $p_{rm}^{(k)} = r_k/m_k$, where r_k is the number of sensors to be removed and m_k is the number of sensors remaining at iteration *k*.

TABLE II

DESCRIPTION OF THE DENSITY MODEL.

a) Assign a probability to sensor n_i for being a gateway to cluster *j*, $p_{ij}^{(n_i)}$, proportional to the number of neighbors which belong to cluster *j*, $N_{ij}^{(n_i)}$. That is, $p_{ij}^{(n_i)} \propto \frac{N_{ij}^{(n_i)}}{M_j}$, where M_j is the number of sensors in cluster *j*.

b) Let S_{ij} denote the set of probability measures $\{p_{ij}^{(n_i)}\}$ in cluster i for selecting a gateway to cluster *j*.

c) Let $B_{ij}^{(n_i)}$ be the set of neighboring sensors in cluster j with respect to sensor n_i.

d) Let $\mathbf{P}^{(k)}$ be a set of probability measures $\{S_{ij}^{(k)}\}$ at time step k.

e) Assign $k = 0$ and $\mathbf{P}^{(0)}$=$\{S_{ij}^{(0)}\}$.

while (sum($\mathbf{P}^{(k)}) > 0$)

(1) Gateway selection

G_{ij} = arg $\max_{S_{ij}^{(k)}}$ $\{p_{ij}^{(n_i)}\}$

(2) Update the probability distribution

$p_{ij}^{(m_i)} = 0, \forall\ m_i \neq G_{ij}$

$p_{ji}^{(l_j)} = \min\{2p_{ji}^{(l_j)}, 1\}, \forall$ sensor $l_j \in B_{ij}^{(n_i)}$.

set $k = k + 1$.

end

TABLE III

DESCRIPTION OF THE DISTRIBUTED RANDOMIZED MODEL.

a) Let $N_b^{(k)}$ be the sum of neighboring sensors of sensors at iteration k.
$N_b^{(k)} = \sum_{i=1}^{m_k} N_i^{(k)}$.
m_k is the number of sensors remaining at iteration *k*.
$i \in I_k$; I_k is the index set of sensors at iteration *k*.

b) Let $E^{(k)}[N_i]$ be the average number of neighbors at iteration k.
$E^{(0)}[N_i] = \frac{N_b^{(0)}}{m_0}$.

c) Assign the probability $p_i^{(k)}$ to sensor *i*, proportional to the number of neighboring sensors, N_i^k. That is, $p_i^{(k)} \propto \frac{N_i^{(k)}}{N_b^{(k)}}$.

d) Assign $k = 0$, $m_0 = n$, $r_0 = 0$.

while $(m_k - r_k) > 0$

$m_{k+1} = m_k - r_k$,

$E^{(k+1)}[N_i] = \frac{N_b^{(k)} - r_k \cdot E^{(k)}[N_i]}{m_{k+1}}$,

$r_{k+1} = \lceil E^{(k+1)}[N_i] \rceil^* + 1$,

$k = k + 1$.

end

$^*\lceil\cdot\rceil$ is the ceiling function.

e) Given the estimated number of clusterheads, N_{ch}, generate a random network of N_{ch} sensors with transmission range $R^2 \approx l^2(\ln(l)/N_{ch})$

f) Approximate the number of distributed gateways in the network by the sum of the neighboring sensors of the N_{ch} sensors
$N_g = \sum_{i=1}^{N_{ch}} N_i^{(g)}$,
where $N_i^{(g)}$ is the number of the neighboring sensors for sensor i during the procedure of gateway selection.

Denote the collection of sensors at iteration *k* by V_k. Since a clusterhead and its neighboring sensors are removed at each iteration, the collection of sensors at the next iteration, V_{k+1}, is simply a new and smaller network. The Lindeberg Theorem [16] can be applied to approximate the distribution of the number of clusterheads at iteration *k* by $\mathcal{N}(\mu_k, \sigma_k^2)$, where $\mu_k = \sum_{i=1}^{m_k} p_i^{(k)}$, $\sigma_k^2 = \sum_{i=1}^{m_k} p_i^{(k)}(1-p_i^{(k)})$, m_k is the number of sensors in V_k, $p_i^{(k)}$ is the updated probability distribution of sensor *i* at iteration k, which is proportional to the number of neighboring sensors, $i \in I_k$, and I_k is the index set of sensors at iteration *k*. Once the procedure terminates, the number of iterations is an estimate of the number of clusterheads formed in the network.

2) The Prediction Formula: The operation of the ADTCA with the distributed model is partitioned into rounds, where each round initializes, clusters are formed, and gateways are selected. The Distributed Randomized Model is described in Table III.

To obtain the mean and variance of the number of clusterheads of each iteration, the probability distribution of these random variables must be updated. However, it is not simple to calculate $p_i^{(k)}$ at iteration *k* since the process of selecting a clusterhead at each iteration is complex. The following simplified analysis restructures the connectivity of the network so that each sensor has the same average neighboring density at each iteration. Therefore, we have

$$E^{(k+1)}[N_i] = \frac{N_b^{(k)} - r_k \cdot E^{(k)}[N_i]}{m_{k+1}}. \tag{6}$$

Thus, the distribution of the number of clusterheads can be approximated by $\mathcal{N}(\mu_{ch}, \sigma_{ch}^2)$, where

$$\mu_{ch} = \sum_{k=1}^{N_{it}} \mu_k = \sum_{k=1}^{N_{it}} \sum_{i=1}^{m_k} p_i^{(k)}, \tag{7}$$

$$\sigma_{ch}^2 = \sum_{k=1}^{N_{it}} \sigma_k^2 = \sum_{k=1}^{N_{it}} \sum_{i=1}^{m_k} p_i^{(k)}(1 - p_i^{(k)}), \tag{8}$$

where N_{it} is the number of iterations.

Moreover, suppose that the expectation of the number of neighboring sensors of each sensor in the network is used to approximate the number of neighboring sensors that will be removed at each iteration (i.e. the sensors which will eventually join the new cluster). Thus,

$$E^{(k)}[N_i] = E[N_i] = \frac{1}{n} \sum_{i=1}^{n} N_i, \text{ for all } k.$$

Then

$$r_k = \lceil E[N_i] \rceil + 1,$$

and a simple formula for predicting the number of clusterheads is

$$N_{ch} = \frac{n}{\lceil E[N_i] \rceil + 1}. \tag{9}$$

Based on the cluster formation and given the estimated number of clusterheads, N_{ch}, a random network of N_{ch} sensors with transmission range $R^2 \approx l^2(\ln(l)/N_{ch})$ [17] is generated to abstract the behavior of gateway selection and approximate the number of distributed gateways in the network. This is attributed to the close relationship between the cluster formation and gateway selection. Therefore, following the analysis of the Averaged Model, the total number of gateways N_g is given by

$$N_g = \sum_{i=1}^{N_{ch}} N_i^{(g)}, \tag{10}$$

where $N_i^{(g)}$ is the number of the neighboring sensors for sensor i during the procedure of gateway selection. Hence, the average number of gateways $N_{g(\text{avg})}$ in a cluster is

$$N_{g(\text{avg})} = \frac{N_g}{N_{ch}} \quad (11)$$
$$= \frac{N_g}{n}(\lceil E[N_i] \rceil + 1). \quad (12)$$

The relationship between the behavior of gateway selection (Phase II) of the ADTCA and that of the Distributed Randomized Model is shown experimentally in Section V.

IV. Energy Consumption Analysis

This section analyzes the energy consumption of the ADTCA when executing the three phases: clusterhead selection, gateway selection, and cluster reorganization. The total power requirements include both the power required to transmit messages and the power required to receive (or process) messages.

A. Phase I

The energy consumption of clusterhead selection assuming homogenous sensors is examined. In the initialization phase, each sensor broadcasts a $Hello$ message to its neighboring sensors. Therefore, the number of transmissions N_{T_x} is equal to the number of sensors in the network, n, and the number of receptions N_{R_x} is the sum of the neighboring sensors of each sensor. That is,

$$N_{T_x} = n \text{ and } N_{R_x} = \sum_{j=1}^{n} N_j. \quad (13)$$

As a sensor, say sensor i, meets the conditions of being a clusterhead, it broadcasts this and assigns cluster ID i to its neighboring sensors. Its neighboring sensors then transmit a signal to their neighbors to update cluster ID information. During this clustering phase, $(1+N_i)$ transmissions and $(N_i + \sum_{j\in C_i} N_j)$ receptions are executed, where C_i is the index set of neighboring sensors of sensor i. This procedure is applied to all clusterheads and their cluster members. Now let $N^c_{T_x}$ and $N^c_{R_x}$ denote the number of transmissions and receptions for all clusters, respectively. Hence,

$$N^c_{T_x} = \sum_{i\in I}(1+N_i), \quad (14)$$

$$N^c_{R_x} = \sum_{i\in I}(\sum_{j\in C_i} N_j + N_i), \quad (15)$$

where I is a index set of clusterheads. Therefore, the total number of transmissions N_T and the number of receptions N_R are

$$N_T = N_{T_x} + N^c_{T_x} = n + \sum_{i\in I}(1+N_i), \quad (16)$$

$$N_R = N_{R_x} + N^c_{R_x} = \sum_{j=1}^{n} N_j + \sum_{i\in I}(\sum_{j\in C_i} N_j + N_i). \quad (17)$$

Suppose that the energy needed to transmit is E_T, which depends on the transmitting range R, and the energy needed to receive is E_R. From (16) and (17), the total energy consumption, E_{total}, for cluster formation in the wireless sensor network is

$$E_{total} = N_T \cdot E_T + N_R \cdot E_R. \quad (18)$$

Observe that the above analysis is suitable for any transmitting range. However, overly small transmission ranges may result in isolated clusters whereas overly large transmission ranges may result in a single cluster. Therefore, in order to optimize energy consumption and encourage linking between clusters, it is sensible to consider the minimum transmission power (or range R) which will result in a fully connected network. The performance of the total energy consumption of Phase I with different selections of R is examined via simulation.

B. Phase II

The energy consumption for determining gateways is evaluated based on the description of Table I. Figure 2 shows The possible determination of a gateway in a cluster. In order to simplify the presentation, the main notations are introduced as follows: let *I* denote the index set of clusterheads; let *H* denote the index set of 1-hop cluster members in the network; let H_i denote the index set of 1-hop cluster members of cluster *i* (a subset of *H*); let *M* denote the index set of 2-hop cluster members in the network; let M_i denote the index set of 2-hop cluster members of cluster *i* (a subset of *M*); similarly, let *S* be the index set of sensors neighboring with 2-hop cluster members; let S_i be the index set of sensors neighboring with 2-hop cluster members of cluster *i* (a subset of *S*); let *G* be the index set of gateway nodes.

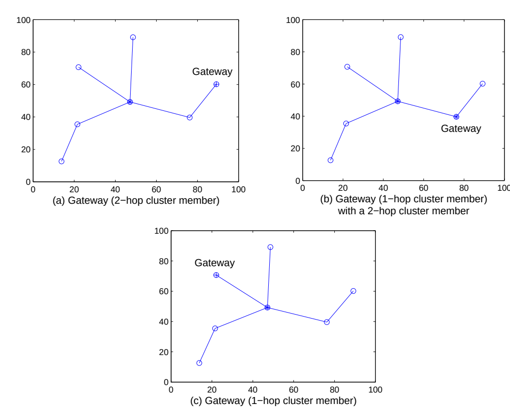

Fig. 2. The possible determination of a gateway in a cluster: (a) a 2-hop cluster member, (b) a 1-hop cluster member with a 2-hop member, and (3) a 1-hop cluster member.

When clusterheads broadcast messages to trigger the gateway selection procedure, the number of transmission N_{T_1} and reception N_{R_1} can be expressed by

$$N_{T_1} = \sum_{i\in I} N_i + \sum_{i\in I}\sum_{j\in S_i} N_j \quad (19)$$

$$N_{R_1} = \sum_{i\in I}\sum_{j\in H_i} N_j + \sum_{i\in I}\sum_{j\in M_i} N_j. \qquad (20)$$

After applying the procedure for choosing gateways, the gateway nodes broadcast messages to update the connectivity information and activate the linked cluster architecture. For this task, the number of transmission N_{T_2} and reception N_{R_2} is given by

$$N_{T_2} = \sum_{i\in I} N_i + \sum_{i\in S, i\notin G} N_i + \sum_{i\in G} N_i \qquad (21)$$

$$N_{R_2} = \sum_{i\in H} N_i + \sum_{i\in M} N_i + \sum_{i\in G}(N_i - \widetilde{N}_i), \qquad (22)$$

where $\widetilde{N}_i$ is the number of neighboring cluster members of the gateway.

Thus, based on the energy needed to transmit and receive, the total energy consumption for gateway selection can be assessed.

C. Phase III

This subsection considers energy consumption of cluster reformation using both the centralized and distributed methods. The 1-hop and 2-hop cluster members depend on the initial hierarchy of clusters. A *n*-hop cluster member is a sensor which is *n* hops away from its initial clusterhead. Let N_i be the number of neighboring sensors of sensor *i*, N_i^{n-hop} be the number of *n*-hop cluster members of clusterhead sensor *i*, and I_s be the index set of the subclusterheads.

1) The Centralized Method: For the present clusterhead to select a new clusterhead, it must gather information from the sensors in the cluster. Thus the clusterhead requests data by sending the interest message using 2 rounds of local flooding propagation to its 1-hop and 2-hop cluster members. The number of transmissions $N_{T_1}^c$ and receptions $N_{R_1}^c$ of this design choice are approximately given by

$$N_{T_1}^c \approx 1 + N_i^{1-hop}, \qquad (23)$$

$$N_{R_1}^c \approx N_i^{1-hop} + \sum_{j\in C_i} N_j, \qquad (24)$$

where C_i is the index set of the cluster members of sensor *i*.

Data from the cluster members is then sent towards the clusterhead. The number of transmissions $N_{T_2}^c$ and receptions $N_{R_2}^c$ are

$$N_{T_2}^c \approx N_i^{1-hop} + N_i^{2-hop}, \qquad (25)$$

$$N_{R_2}^c \approx \sum_{j\in C_i} N_j. \qquad (26)$$

When the clusterhead receives the desired information for solving the optimization problem of (2) and (3), it determines the new clusterhead and notifies all members. The number of transmissions $N_{T_3}^c$ and receptions $N_{R_3}^c$ are thus $N_{T_3}^c = N_{T_1}^c$ and $N_{R_3}^c = N_{R_1}^c$.

2) The Distributed Method: The energy consumption of the distributed method is examined in three steps. Step I of the method is to broadcast a message and group cluster members into subclusters. In this step, the cluster is considered as a small network where the energy consumption analysis of the CAWT [8] can be applied. Therefore, if the current clusterhead is sensor i, the number of transmissions $N_{T_1}^d$ and receptions $N_{R_1}^d$ in an error-free channel are approximately given by

$$N_{T_1}^d \approx 2\cdot(N_i^{1-hop} + N_i^{2-hop}), \qquad (27)$$

$$N_{R_1}^d \approx 2\cdot\sum_{j\in C_i} N_j. \qquad (28)$$

The mission of Step II is to collect sufficient information from subcluster members. The subclusterhead first broadcasts an interest message to inform its members about what kind of data it requires. Based on this message, the subcluster members propogate the desired data back to the subclusterhead. Thus, the number of transmissions $N_{T_2}^d$ and receptions $N_{R_2}^d$ are approximately

$$N_{T_2}^d \approx \sum_{j\in I_s}(1 + 2\cdot N_j^{1-hop} + N_j^{2-hop}), \qquad (29)$$

$$N_{R_2}^d \approx \sum_{j\in I_s}(N_j^{1-hop} + 2\cdot\sum_{k\in C_j} N_k). \qquad (30)$$

In the final Step, subclusterheads exchange ID information in order to determine the new clusterhead. The energy consumed in this phase may depend on the number of subclusterheads, the related positions among subclusterheads, and how they communicate with each other. Assume that there exists n_{sch} subclusterheads in a cluster. In this case, each subclusterhead broadcasts an interest message including its sensor ID to the whole cluster, which allows subclusterheads to figure out which subclusterhead is the new clusterhead immediately as they receive the ID information and thereby complete the reselection process. Therefore, we may approximate the number of transmissions $N_{T_3}^d$ and receptions $N_{R_3}^d$ by

$$N_{T_3}^d \approx n_{sch}\cdot(N_i^{1-hop} + N_i^{2-hop}), \qquad (31)$$

$$N_{R_3}^d \approx n_{sch}\cdot\sum_{j\in C_i} N_j. \qquad (32)$$

The analysis suggests that, compared with the overall energy consumption of the distributed method, the centralized method consumes less energy for reselecting a clusterhead while the reselection process may fail due to the malfunction of the current clusterhead and the corrupted information collection.

V. Experimental Results

The simulations of this section study the performance of the ADTCA and validate the simplified models for which analytical results have been derived. Assume that *n* sensors are uniformly distributed over a square region in two-dimensional space. Parameters for the random waiting timer, number of

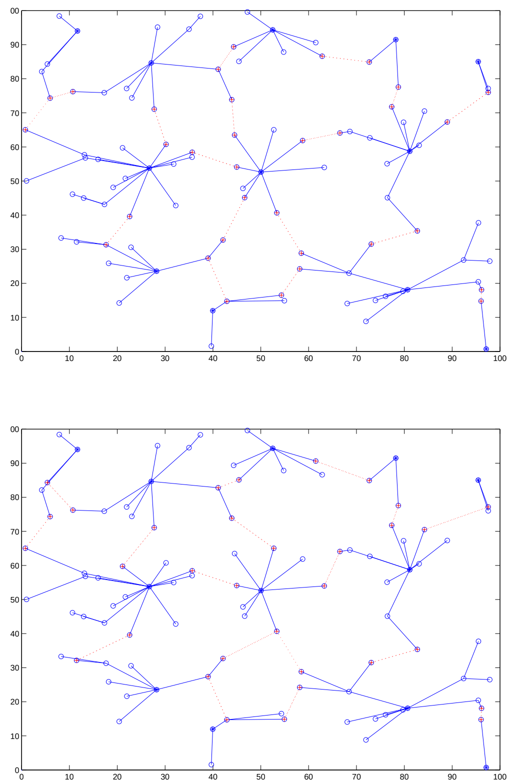

Fig. 3. Gateway selection in a random network with 100 sensors: Phase II of the ADTCA with $R/l = 0.17$ (top) and the Density Model with $R/l = 0.17$ (bottom).

sensors, and ratio of transmitting range *R* to the side length *l* of the square, *R/l*, are investigated to provide a simulation-based study of the ADTCA.

The first set of experiments in Figure 3 evaluates the performance of the Density Model, which compares gateway selection when using the Density Model and the operation of Phase II. The outputs of the two methods are not identical due to the randomness of the waiting timer. Nonetheless, both clustering structures are qualitatively similar given the same network settings, suggesting that the Density Model provides a good approximation to Phase II of the ADTCA.

The second set of experiments compares the estimates of the number of distributed gateways when applying the procedure of Phase II, the Density Model, and the Distributed Randomized Model. In each method, the results of 200 typical runs are merged. In order to compare the ADTCA and the simplified models, Figure 4 shows the standard deviation of the mean number of gateways. The plots vary the number of sensors n and the transmission power R/l. Also shown in Figure 4 are the confidence intervals for the mean number of gateways at a 90% confidence level. The graphs suggest that the Density Model approximates the ADTCA somewhat better than the Distributed Randomized Model. This is reasonable because the Density Model retains global connectivity information while the Distributed Randomized Model uses only the average density information. Though the Density Model outperforms the Distributed Randomized Model, these results provide evidence that the Distributed Randomized Model provides a way to roughly predict the performance of the ADTCA.

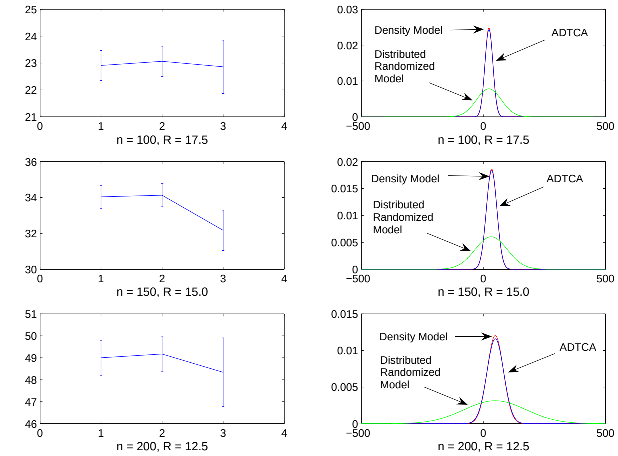

Fig. 4. The number of gateways formed in a random network using the (1) ADTCA, (2) Density Model, and (3) Distributed Randomized Model, respectively, with varying R/l ratio. The right hand side shows the standard deviation over 200 runs; the left hand side shows the confidence intervals at the 90% level.

The third set of experiments considers the total energy consumption of the ADTCA. Assume that the communication channel is error-free. Since each sensor does not need to retransmit any data, two transmissions are executed for clusterhead selection (Phase I), one for broadcasting the existence and the other for assigning a cluster ID to its cluster members or updating the cluster ID information of its neighbors. Hence, the total number of transmissions is $2n$. Under these circumstances, sensor *i* will receive $2N_i$ messages. Then, the total number of receptions is $2\sum_{i=1}^{n} N_i$. Figure 5 shows the average number of receptions of random networks after applying Phase I of the proposed algorithm. Figure 5 also shows that the number of receptions tends to increase as the ration *R/l* increases. This implies that energy consumption is higher for the network with larger transmission power. This can be attributed to the fact that larger transmission power allows sensors to detect more neighbors, which increases the number of receptions when assigning cluster ID or updating cluster ID information. Therefore, in order to minimize energy use and keep strong connectivity in the network, an appropriate selection of the transmission range *R* is essential.

Figure 6 illustrates the average number of transmission and reception in a cluster for executing gateway selection. Observe that the operation of Phase II in the ADTCA may lead to a minimal variation of the energy consumption with increased network density, which may help to achieve balance the load among the clusters.

For comparison, the same network topology and sensor energy level are used to study the performance of the two methods in Phase III during the first round. Let the threshold level η be $E_{max}/2$. Samples from the distributions, $E_{max} \cdot U(0,1)$ and $E_{max}/2 \cdot (1+U(0,1))$ are assigned to clusterheads

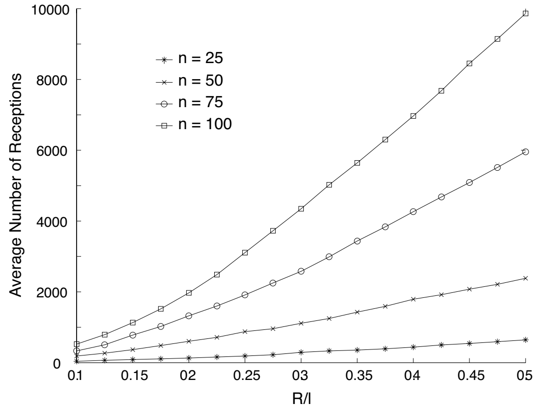

Fig. 5. The number of receptions in random networks as a function of the number of sensors and R/l ratio in Phase I (Cluster Formation).

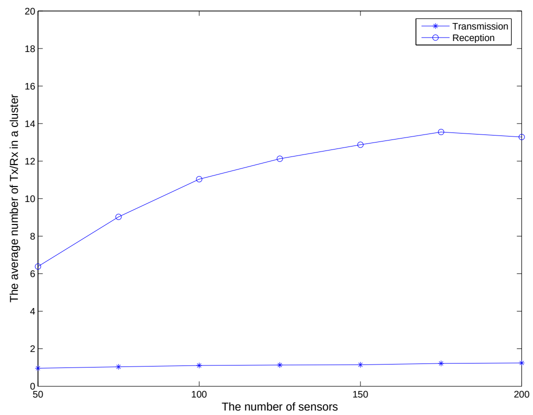

Fig. 6. The average number of transmission and reception in a cluster for executing gateway selection (Phase II) with increased network density.

and cluster members as the remaining energy, respectively. Figure 7 demonstrates typical runs of the operation of Phase III. It shows that this kind of local dynamic distribution of clusterheads allow each cluster to adjust its energy load among cluster members, which alleviates the problem that the battery of fixed clusterheads will drain quickly. Therefore, when the reselection operation is completed, the energy usage is spread among the network and thereby the lifetime of the network is extended.

VI. Applications of the ADTCA

A. Time Synchronization

The time synchronization issue is a typical problem in wireless sensor networks because of the observation and interaction with the physical world. Due to random phase shifts and clock skews of oscillators, the time reading of sensors might start to loose synchronization without calibration. The ADTCA technique may be a good way to keep the time readings of sensors as tightly as possible in the hierarchical cluster-based network structure since local pair-wise synchronization [18] is achievable within a cluster using two-way communications

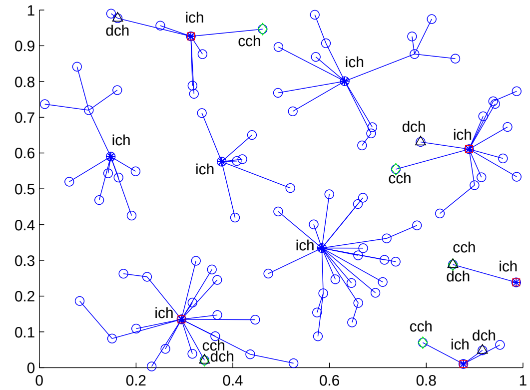

Fig. 7. Clusters are formed and clusterheads are reselected in a random network of 100 sensors with $R/l = 0.175$; "□" represents the initial clusterhead (ich); "◇" represents a new clusterhead using the centralized protocol (cch); " △ " represents a new clusterhead using the decentralized protocol (dch).

while global calibration can be achieved by (relatively sparse) communication between clusterheads.

B. Efficient Network Routing

Hierarchical cluster-based network routing is a well-known protocol with special advantages related to scalability and efficient communication for wireless sensor networks. In a hierarchical architecture, clusterheads can be used to process and deliver information efficiently while gateways are responsible for forwarding information between clusters. This implies that the creation of clusters and gateways greatly contributes to system scalability, network lifetime, and energy conservation. Therefore, the proposed ADTCA approach may be an efficient way to lower energy consumption since the number of transmitted messages to the destination is decreased by performing data aggregation and fusion in clusterheads and messages can be relayed with reliable broadcasting in distributed gateways.

VII. Conclusion

This paper describes a decentralized protocol for topology management in wireless sensor networks. The Adaptive Distributed Topology Control Algorithm (ADTCA) performs cluster formation and linkage using random waiting timers and local information. On the basis of the cluster-based network topology, this self-configuring technique may be applied to achieve local and global time synchronization and to provide efficient network routing.

This work assumes that all sensors operate with the same transmission range. Future plans involve generalizing the method to consider power control strategy for minimizing the total energy consumption, to consider certain failure scenarios, and to design efficient topology control protocols for mobile ad-hoc wireless networks.

References

[1] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of IEEE HICSS*, January 2000.

[2] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "An application specific protocol architecture for wireless microsensor network," in press: *IEEE Transaction on Wireless Networking*.

[3] M.J. Handy, M. Haase, D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," 4th *International Workshop on Mobile and Wireless Communications Network*, pp. 9-11, September 2002.

[4] C.F. Chiasserini, I. Chlamtac, P. Monti and A. Nucci, "Energy efficient design of wireless ad hoc networks," in *Proceedings of European Wireless*, February 2002.

[5] 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*, No. 5, pp. 193-204, 2002.

[6] J. Lundelius and N. Lynch, "An upper and lower bound for clock synchronization." *Information and Control*, Vol. 62 1984.

[7] M. Cardei, J. Wu, and S.-H. Yang, "Topology control in ad hoc wireless networks using cooperative communication," in *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 711-724, June 2006.

[8] C.-Y. Wen and W. A. Sethares, "Automatic decentralized clustering for wireless sensor networks," in *EURASIP Journal on Wireless Communications and Networking*, Volume 2005, Issue 5, pp. 686-697, December 2005.

[9] A.D. Amis, and R. Prakash, "Load-balancing clusters in wireless ad hoc networks," in *Proceedings of ASSET* 2000 , Richardson, Texas, March 2000.

[10] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 310-315, June 1999.

[11] M.N. Halgamuge, S. M. Guru, and A. Jennings, "Energy efficient cluster formation in wireless sensor networks," 10th *International Conference on Telecommunications*, vol.2, pp. 1571-1576, 2003.

[12] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communication*, Vol. 15 pp. 1265-1275, September 1997.

[13] A. B. McDonald, and T. Znati, "A mobility based framework for adaptive clustering in wireless ad-hoc networks," in *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1466-1487, Aug. 1999.

[14] L. Bao and J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *Proc. of MobiHoc'03*, pp. 129-140, Maryland, June 2003.

[15] D. J. Baker, A. Ephremides, and J. A. Flynn, "The design and simulation of a mobile radio network with distributed control," in *IEEE Journal on Selected Areas in Communications*, SAC-2(1): pp. 226-237, 1984.

[16] P. Billingsley, *Probability and Measure*, John-Wiley & Sons, Inc 1979.

[17] Paolo Santi, Douglas M. Blough, and Feodor Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, pps: 212-220, 2001, Long Beach, CA.

[18] C.-Y. Wen, R. D. Morris, and W. A. Sethares, "Distance Estimation Using Bidirectional Communications Without Synchronous Clocking," in *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1927-1939, May 2007.