A Scalable Clustering Algorithm in Dense Mobile Sensor Networks

Jianbo Li¹, Shan Jiang²
Information engineering college, Qingdao University
Qingdao 266071, Shandong Province, P.R.China
¹lijianboqdu@yahoo.com.cn
²jiangshanqdu2008@gmail.com

Abstract—Clustering offers a kind of hierarchical organization to provide scalability and basic performance guarantee by partitioning the network into disjoint groups of nodes. In this paper a scalable and energy efficient clustering algorithm is proposed under dense mobile sensor networks scenario. In the initial cluster formation phase, our proposed scheme features a simple execution process with polynomial time complexity, and eliminates the “frozen time” requirement by introducing some GPS-capable mobile nodes to act as cluster heads. In the following cluster maintenance stage, the maintenance of clusters is asynchronously and event driven so as to thoroughly eliminate the “ripple effect” brought by node mobility. As a result local changes in a cluster need not be seen and updated by the entire network, thus bringing greatly reduced communication overheads and being well suitable for the high mobility environment. Extensive simulations have been conducted and the simulation results reveal that our proposed algorithm successfully achieves its target at incurring much less clustering overheads as well as maintaining much more stable cluster structure, as compared to HCC (High Connectivity Clustering) algorithm.

Index Terms—mobile sensor networks, topology control, clustering scheme, high connectivity clustering, scalability

I. INTRODUCTION

With the recent technological advancements in wireless communication and the increasing proliferation in portable computing devices, wireless and mobile sensor networks have drawn impressive attention and are expected to play an increasingly important role in some situations [1], including disaster and emergency relief, sea monitoring, animal migration and so on, where wireless access to wired backbone is either ineffective or impossible.

Topology Control is one of the most important techniques used in ad hoc and sensor networks to reduce energy consumption which is essential to extend the network life time and reduce MAC (Medium Access Control) layer signal interference with a positive effect on the network capacity.

For stationary networks, extensive works have been studied in the literatures on a variety of topology control problems, such as location based algorithm LMST [2], direction based CBTC [3] and link quality based XTC [4]. Though performing well in stationary networks, these localized topology control algorithms still encounter many problems in mobile networks and it is hard to adapt them to mobile sensor networks directly. A mobile multi-hop sensor network needs to cope with the frequent underlying topology change, variable link quality and short-lived link duration due to the intrinsic dynamic nature caused by node mobility. Consequently, a topology control protocol should provide the mechanisms for a mobile sensor network to possess the ability of self-adapting, self-organizing and self-repairing in the face of node mobility [5], which makes topology control in mobile sensor networks a challenging and difficult problem.

Topology control algorithms for ad hoc and sensor networks can be broadly classified into two categories—hierarchical clustering organization and flat power control organization [6, 7]. However the power control schemes for mobile sensor networks have typical disadvantages as follows: (1) Most of the power control algorithms yield a minimally connected topology, which is prone to frequent link breakages, thus severely degrading the network performance in a highly mobile network (2) In a highly mobile sensor network, dynamic property of moving nodes requires periodically re-execution of the corresponding power control protocol, thus producing large amount of communication and computation overheads and also incurring considerable end-to-end delay. All these indicate that a flat power control scheme cannot support scalability well in a highly mobile sensor network. Therefore, a hierarchical clustering method is often adopted to achieve the basic performance guarantee and solve the scalability issues in a large dynamic sensor networks.

In a typical clustering algorithm, there are usually three types of nodes, named cluster heads, cluster members and cluster gateways respectively. In this context, Cluster Heads (CHs) are responsible for coordination among the nodes within their clusters (i.e. intra-cluster communication) as well as communicating with other cluster heads (i.e. inter-cluster communication). The Cluster Members (CMs) are ordinary nodes and only need to transmit their information to their respective cluster heads, which aggregate the received information and then forward it to the sink. While cluster gateways
are non-cluster heads with inter-cluster links, they can bridge neighboring clusters and forward information among them. Furthermore, the cluster heads and cluster gateways form a virtual backbone to facilitate intra-cluster and inter-cluster transmission, hence increasing spatial channel reuse and maximizing network capacity [8][9].

Clustering can be normally separated into two phases, cluster formation and cluster maintenance [10]. Cluster formation means how to select the CHs and CMs to form the initial cluster structure at the very beginning. Cluster maintenance refers to how to subsequently maintain and update the cluster structure so as to achieve some goal or accommodate the underlying topology change caused by node mobility. In a static sensor network, cluster maintenance is usually about to re-elect CHs in the case of some cluster head(s) failure or for the purpose of achieving load balancing. In the mobile sensor networks, whereas cluster maintenance mainly focuses on timely updating the cluster structure according to the continuously dynamic underlying topology incurred by nodes movement. So how to reduce the cluster maintenance costs to the fullest extent is vital to validate the effectiveness and efficiency for a clustering algorithm in mobile sensor networks.

There are two specific problems associated with cluster formation and cluster maintenance phase, namely “frozen time” requirement and “ripple effect”, respectively. In the cluster formation phase, some existing clustering schemes normally require mobile nodes to exchange corresponding information to elect the CHs with specific attributes, such as LIC (Lowest Identifier Clustering) [11] and HCC (Highest Connectivity Clustering) [12] demand the elected CHs have the lowest identifier and highest node degree respectively, in their one-hop neighborhood. In a mobile network environment, all the nodes participating the clustering process must be assumed frozen [13] (i.e., static) during the whole cluster formation period to guarantee that the exchanged information are accurate and the elected CHs satisfy the predefined properties. However, this may be impractical in an actual scenario, such as in random walk model [14], where all the mobile nodes move all the time.

In the following cluster maintenance phase, as stated earlier, the cluster structure needs to be updated to respond to the underlying network topology change. Membership update and cluster head change are inevitable because of energy depletion or not conforming the corresponding property. For example, when a cluster head finds a one-hop neighbor with a higher node degree in HCC, it then resigns the cluster head role to this node. However, this will make the clustering scheme re-execute the clustering process again (i.e., a single node’s failure may cause all the other nodes participate the new cluster formation process), which will arise the so-called “ripple effect” and then produce large amounts of computation and communication overheads, thus adversely affecting the performance of clustering protocol. Consequently how to maintain a stable cluster structure in the presence of node mobility and minimize the impact of cluster head or cluster member change on the cluster structure should be a major target for a clustering scheme in a mobile sensor network environment.

This paper proposes a scalable and efficient clustering algorithm to solve the two problems mentioned above. The algorithm first divides the deployment region into small grids and dispatches the corresponding number of GPS-capable mobile nodes to these grids to act as CHs. The remaining ordinary nodes then join the corresponding clusters as CMs by only listening the “Hello” message sent by CHs. Since a cluster gateway is covered by more than one cluster head, its movement may likely affect more clusters, which is not favorable for maintaining network structure. Consequently, cluster gateways are excluded in our algorithm for the purpose of simplifying cluster maintenance in face of node mobility. Since the cluster heads are system dispatched rather than elected by some rule, the proposed clustering scheme has a simple clustering formation process and eliminates the assumption on the “frozen time”.

In our clustering scheme, each node allocates a memory space called Information Table for the following cluster structure maintenance. The details of Information Table organization is referred to section III. In this paper, the cluster structure update according to the underlying topology change is event-driven and the impact of cluster head or cluster member change on the cluster structure is only limited to a local area, where the event incurring cluster maintenance occurs, instead of the whole deployment region as in HCC, thus fundamentally removing the “ripple effect” brought by re-clustering. Furthermore, the participating nodes only need to exchange information and update their Information Table to complete the cluster maintenance process. In other words, the proposed clustering protocol maintains a comparatively stable cluster structure and achieves good energy efficiency in the face of node mobility.

The remainder of this paper is organized as follows. An overview of related works in the field of topology control for mobile networks is presented in Section II. Section III explains our clustering scheme in detail. Section IV shows simulation results and comparisons with HCC. Finally, a brief conclusion is drawn in Section V.

II. RELATED WORKS

Many researchers are underway to find energy efficient and scalable topology control schemes for large scale and highly mobile wireless networks. In earlier works, node movement, if incorporated at all, was handled in one of four ways [15]: (1) specific topology control protocols based on specific but simple and impractical mobility model [16], (2) energy efficient clustering algorithms to maintain a comparatively stable cluster structure in face of node mobility [10], (3) mobility-aware topology control schemes to incorporate node movement patterns into consideration to facilitate topology management [17] and (4) let each node augment its transmission range by a redundant transmission range (RTR) factor to involve all the possible movements of that node so as to tolerate
inaccurate node position information, such as [18] combined RTR with relay region method. The buffer zone adopted in [19] is also essentially equivalent to the idea of RTR. For each above category, a typical topology control scheme is illustrated respectively as follows.

Zhao [16] presented a topology control algorithm to solve a specific model called Constant Rate Mobile Network (CRMN) optimally from a theoretical viewpoint, where each node moves along its moving route (a line segment) at its own constant rate and direction in the unit time interval. Instead of maintaining the network connectivity during the whole network lifetime, the algorithms firstly partition the network lifetime into some unit time slots and then try to preserve the network connectivity in the unit time interval. Zhao provided two polynomial algorithms for solving this problem: one for the decision version, the other for the optimization version to minimize the maximum power used by any network node in producing a connected network. Since the adopted CRMN mobility model is far from realistic, the results mainly embodies on its theoretic value.

Yu [10] proposed an efficient clustering scheme, named ECS, for large and dense mobile ad hoc networks. By using a “random cluster head claim” mechanism, the frozen period requirement for cluster formation phase is eliminated. A new clustering node status, named cluster guest, is introduced to avoid the formation of small or unnecessary clusters and effectively improves the cluster stability by prolonging the cluster head life with moderate clustering overheads. In addition, the implementation of cluster guest helps reduce cluster overlapping in ECS. Simulations reveal that ECS successfully achieves its goal in maintaining cluster stability and reducing cluster overheads, so it represents the development trend of clustering schemes for large and mobile networks.

Sripongwutikorn [17] presented two topology control algorithms—Absolute Distance-based (ABD) and Predictive Distance-based (PRD) in a mobile ad hoc network. Both algorithms attempt to maintain a logic number of neighbors between predefined values \( K_{\text{min}} \) and \( K_{\text{max}} \) by adjusting the transmission range of individual nodes to achieve good network throughput. The ABD algorithm uses the absolute distance as the neighbor selection criteria while the PRD algorithm incorporates mobility information to extend the neighbor lifetime, which makes the PRD particularly suitable for a mobile network having correlated or similar node mobility patterns, such as a Vehicular Ad hoc NETwork (VANET).

Wu [19] proposed a mobility-sensitive topology control method that extended many mobility-insensitive protocols. This method is based on two mechanisms: local view consistency based on synchronous and asynchronous “Hello” messages, and buffer zone created to tolerate the inaccurate location information caused by node mobility by slightly increasing the actual transmission range. These two mechanisms ensure the connectivity of both logical topology and effective topology, two notions proposed in this paper for topology control in dynamic networks, so as to maintain the global network connectivity. The buffer zone is similar to the notion of redundant transmission range (RTR), which helps to reduce the link breakage even in the case of inaccurate node position information, thus enhancing the system robustness.

To the best of our knowledge, there are some open issues in the field of topology control for mobile sensor networks that need to be addressed as follows:

- Decentralized topology management: if a central topology control protocol is adopted, the network management will react to the overall dynamic topology, thus resulting in high overheads and being lack of scalability. Consequently, a distributed topology control scheme is favored for the purpose of reliability and scalability.
- Impact of mobility model on topology control: the mobility models employed by the current topology control algorithms in mobile sensor networks are too ideal and impractical to simulate the actual nodes movement. Hence, more general and practical mobility models should be proposed and the characteristics of these models, such as the distribution of moving nodes, should be thoroughly taken into consideration to facilitate topology control in the network.
- View inconsistency: some recent studies show that mobility causes incorrect information in terms of link availability and view consistency [20] to adversely influence the correctness and performance of topology control protocols. Therefore new methods should be presented to enforce the view consistency to give the correct input for the corresponding topology control protocols and preserve the global connectivity.

### III. OUR CLUSTERING SCHEME

In this scheme, the basic assumptions are defined first followed by the detailed description of our clustering mechanisms.

#### A. Network Model Assumption

- The mobile sensors are randomly distributed into a square field of side length \( L \)
- There are two types of mobile nodes: one is equipped with GPS hardware, while the other is ordinary without any positioning or localization capability.
- All the nodes have the same initial energy supply, denoted as \( e \), and the maximum transmission radius, \( TX_{\text{range}} \).

#### B. Cluster Formation Phase

The proposed clustering scheme employs the same method as in GAF [21] to divide the deployment region. That is to say the scheme partitions the deployment region into small square grids with side length \( r \) to satisfy the condition that each node in a grid can directly communicate with any other node located in its neighboring grid, where \( r \leq \frac{TX_{\text{range}}}{\sqrt{5}} \) can be derived.
for the condition to be true and $r$ is chosen as an integer approaching this value in our paper.

The following cluster formation process is very simple. First, the GPS-capable mobile nodes are dispatched into the center position of each small grid to act as CHs and the reason why choosing the center position is that it will incurs a less intra-cluster transmission distance between CHs and their CMs. Then each cluster head periodically sends “Hello” messages, which contain its own ID and a simple cluster head status claim. After that, the remaining ordinary mobile nodes move randomly or by some mobility model within the deployment region and each mobile node can hear more than one “hello” messages from neighboring CHs due to the way of dividing deployment region. In this situation, each ordinary mobile node simply compares the signal strength of the received “Hello” messages and chooses the node sending the strongest signal strength as its own cluster head, and then it sends a “Hello” message including its own ID and CID (i.e., the identifier of its cluster head) to acknowledge its joining the corresponding cluster head as a cluster member. Meanwhile, each cluster head stores the information of all its CMs into its Information Table according to the collected “Hello” messages sent from ordinary mobile nodes. After a short period of time, the initial cluster structure is shown as in Fig.1.

It should be noted that this clustering scheme does not rely on geographical position of each node, but rather on the received signal strength, thus being more realistic with existing hardware equipment. As shown in Fig.1, node 5 residing the lower right cell chooses node 1, instead of node 7, as its cluster head, since the signal strength is not necessarily proportional to the distance between nodes.

### C. Cluster Maintenance Phase

In a mobile sensor network, the cluster structure should be timely updated to accommodate the continuously changing underlying topology caused by node mobility, which makes the cluster maintenance phase be more important than the cluster formation stage in the clustering scheme. In the cluster maintenance phase, each node keeps a storage space called Information Table to record the corresponding information. A ordinary cluster member owns a simple Information Table, which contains its own ID, CID, and its status (i.e., cluster member), while a cluster head has a more complicated Information Table owing to its functioning as the coordinator among its CMs. Table I shows a typical Information Table for cluster head 1 under the cluster structure illustrated in Fig.1 as an example. For the cluster head, Entry Update Time refers to the time when it scanned its own Information Table last time, but for the cluster member it denotes the time when it received the “Hello” message most recently from its cluster head.

To timely update the cluster structure, each mobile node periodically sends a “Hello” message every $T_s$ to promptly report the possible changes brought by node movement and $T_s$ is set to be 1s in this paper. The “Hello” message sent by a cluster member consists of its ID and CID, whereas the message sent by a cluster head includes its own ID and cluster head status claim. In a mobile sensor network, when a cluster member moves to a new position but not changes its cluster head, its cluster head only sets the Entry Update Time for the corresponding entry to current time, which will not lead to the cluster structure update. In our clustering scheme, maintaining the cluster structure is event-triggered and there are altogether four events that may cause cluster maintenance, namely cluster head failure, cluster member failure, a new mobile node joining and cluster member changing its cluster head. The detailed maintenance processes are described as following:

1) Cluster head failure: When a cluster member node $A$ finds that the time elapsed is longer than the predefined threshold $T_c$ since it received the “Hello” message from its cluster head $B$ last time, it

<table>
<thead>
<tr>
<th>Node Id</th>
<th>Status</th>
<th>Entry Update Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cluster head</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>cluster member</td>
<td>99.5</td>
</tr>
<tr>
<td>3</td>
<td>cluster member</td>
<td>100.2</td>
</tr>
<tr>
<td>4</td>
<td>cluster member</td>
<td>100.6</td>
</tr>
<tr>
<td>5</td>
<td>cluster member</td>
<td>99.8</td>
</tr>
<tr>
<td>6</td>
<td>cluster member</td>
<td>99.2</td>
</tr>
</tbody>
</table>

© 2011 ACADEMY PUBLISHER
still has not received any message from B yet. In this case, node A considers B as obsolete and then deletes B from its Information Table. Meanwhile node A must receive the “Hello” messages from other CHs as a result of the manner like GAF to partition the deployment region and A chooses the cluster head node C with the strongest signal strength as its new cluster head. Such cluster head change will be indicated in the next “Hello” message of node A and hence all its neighbors can update their Information Table accordingly. Specially when cluster head C hears such “Hello” message from A, it needs to do two things: it should firstly transfer the information about cluster head B failure to sink node through backbone network composed of CHs and then the sink node supplements and dispatches another mobile node equipped with GPS hardware to the corresponding grid, where B located, to replace B as the new cluster head; Finally node C checks that there is not any information about A in its Information Table and finds that the CID segment included in the “Hello” message of A is exactly the same as its own ID (i.e., A chooses C as its cluster head), cluster head C consequently inserts a new item about A into its Information Table

2) Cluster member failure: Each cluster head periodically scans its Information Table with an interval $T_i$ and when cluster head A finds out that the time elapsed since last update the Entry Update Time for cluster member B is longer than the predefined $T_{mn}$, A thinks of B as a failure node and then deletes the corresponding entry from its Information Table

3) A new mobile node joining: When a new mobile nodes A is added into the mobile sensor network, it chooses the cluster head B with the strongest signal strength as its own cluster head and declares such change in the next “Hello” message. As node B receives this message from A, it first assures that there is no entry recording about any information of A in its Information Table and the CID field in the “Hello” message of A is exactly the same as its own ID, which indicates that a new node has joined the cluster covered by B. Furthermore, cluster head B adds a new entry into its Information Table to record the correlated information about A

4) Cluster member changing its cluster head: This is the most frequent case occurred in the dense mobile sensor networks. When a cluster member A moves to a new position and discovers that the signal strength from cluster head C is stronger than that from its original cluster head B, A immediately updates its CID field to the ID of C and indicates such change in its next “Hello” message. After a short period, cluster head B receives this message and finds that the CID field contained in this message is not equal with its own ID. In this situation, B deletes the corresponding entry about A, while C takes the corresponding information of A as a new item to add its Information Table.

From the discussions mentioned above, our clustering scheme has the following advantages:

- Our work only requires that a small fraction of the mobile nodes have the positioning capability, while most of the mobile nodes only need to have the basic ability of sensing signal strength.
- Our work forms a non-overlapping cluster structure, i.e., a cluster member only belongs to a single cluster head, which simplifies the process of maintaining cluster structure in face of node mobility.
- CHs are the predefined GPS-capable mobile nodes without election process. Furthermore, the CHs do not bear some specific attributes in their one-hop neighbors, which thoroughly eliminates the rely on the “frozen time” requirement.
- Our work has a simple cluster formation process. Each mobile node only needs to monitor the “Hello” messages sent by surrounding CHs, then compares the relative signal strength, and sends it own “Hello” message to join the corresponding cluster to form the initial cluster structure. Since the number of messages sent and received by each node is constant, the time and message complexity of our scheme both achieve $O(n)$, where $n$ is the number of mobile nodes in the network, which greatly reduces the energy consumption on each node and achieves good energy efficiency.
- The cluster structure update according to the underlying topology change is event-driven and the impact of cluster head or cluster member change on the cluster structure is only limited to a local area, where the event incurring cluster maintenance occurs. That is to say the local changes need not to be seen and updated by the whole network, and the information processed and stored on each node is greatly reduced, thus fundamentally removing the “ripple effect” brought by re-clustering.
- Once a cluster head fails or uses up its energy, the neighboring CHs will transmit this information to the sink via backbone and then the system will dispatch a new GPS-capable node to the corresponding cluster to replace this failed cluster head. Besides our schemes are distributed for the reason each node makes its own decision only based on the local received information, hence our scheme features a good performance on adaptivity and scalability in the dense mobile networks.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our clustering scheme under dense and large-scale mobile sensor network via OMNET++[22] and present quantitative performance comparisons with HCC[12]. To measure the performance of our algorithm, three metrics are identified: (i) the number of messages sent by all the nodes during the whole simulation time (ii) the average
cluster member lifetime, which is defined as the average time period for a member node attaching to the same cluster head for judging the stability of our clustering scheme. The average cluster head lifetime, which is defined as the average surviving time for a cluster head, can be calculated as the time elapsed until the first node depletes its energy. We investigate the variations of these metrics with respect to number of nodes, \( N \), and the maximum moving velocity, \( V_{\text{max}} \). For each simulation scenario, ten runs were executed and the results were averaged. The parameter setting in the simulation environment is listed in Table II.

It is inevitable that any clustering scheme produces explicit clustering-related messages, so the number of messages sent from all the mobile nodes during the whole network running time is an important criterion to validate the performance and effectiveness of a clustering protocol. Fig.2 investigates the performance comparison between our proposed scheme and HCC in terms of number of messages with respect to the maximum speed, denoted as \( V_{\text{max}} \). In the simulation the number of nodes, \( N \), was taken to be 500 and random walk mobility model was adopted.

As can be seen in Fig.2, the two curves representing messages number trends both rise as the maximum speed, \( V_{\text{max}} \), increases. The behind reason is that the affiliation relationship among CHs and CMs becomes instable with the increased node moving speed, thus generating the increased number of interchanged message for the purpose of maintaining cluster structure. Fig.2 also shows that the number of generated messages of HCC is much larger than that of our proposed and the message number of our clustering scheme increases slowly in contrast to the rapid increase in the message number of HCC as the moving speed increases. As the nodes move faster in HCC, the variations in the node degree change much more rapidly, and this demands that the cluster structure should be updated much more quickly so as to maintain the property that the cluster head in HCC bears the largest node degree in its one hop neighbor, thus yielding excessive number of message. Whereas the produced messages in our scheme mainly come from the case that the moving CMs change their CHs, where the correlated CMs only need to send an extra “Hello” message.

As can be seen in Fig.3, the average cluster member lifetime of our scheme and HCC both decrease as the moving speed increases. This is due to the fact that the faster the nodes move, the more frequently the cluster member changes its cluster head, thus producing a shorter average cluster member lifetime. Moreover, the average cluster member lifetime of our clustering scheme is significantly longer than that of HCC under the same moving speed. This is because the faster node speed in HCC incurs the rapid change in node degree, thus causing more frequent CMs affiliation change and consequently a shorter average cluster member lifetime. However, when a cluster member moves in our clustering scheme, it only affects the very few neighboring clusters. That is to say our scheme achieves a comparatively stable cluster structure, hence obtaining a much longer average cluster member lifetime.

![Figure 2. The variation in the number of interchanged messages with respect to the maximum moving speed](image1)

![Figure 3. The average cluster member lifetime comparison between our scheme and HCC](image2)
Fig. 4 shows the average cluster head lifetime comparison between our scheme and HCC with different maximum speed, where \( N \) was set to 500 and random walk mobility model was adopted. In this simulation, both schemes’ cluster head lifetime drop with the increase of maximum speed. For our scheme, the cluster heads are pre-dispatched by system and fixed unless energy depletion occurs, resulting in a rare chance of cluster head change. As nodes move faster, CHs need to exchange much more messages with CMs to update its information table to reflect the CMs affiliation relation change, which will consume much more energy and in turn incur a less cluster head lifetime. However a high mobility environment will make the node degree of a mobile node change frequently, HCC suffers from far more frequent cluster heads changes, which will lead to a much shorter cluster head lifetime. As can be seen from Fig. 4, our clustering scheme achieves an improvement in average cluster head lifetime of factor about 300% that of HCC in a large mobile sensor network, which verifies that our clustering scheme maintains a comparatively stable cluster structure from another perspective.

Fig. 5 shows the performance of our clustering scheme on the system lifetime under two mobility models with respect to the number of nodes. Our scheme employs the same energy model as in [23] with \( V_{\text{max}} = 7 \text{m/s} \) and the pause time in random waypoint model [24] is 3s. As compared to the cluster member, a cluster head takes on more responsibility, thus being more apt to dissipate its energy, so the system lifetime actually refers to the time interval from the very beginning to the first cluster head failure.

We can observe two phenomena from Fig. 5: On the one hand, the system lifetime of our proposed scheme obviously drop under the two mobility models as the number of nodes increases. The main reason is that the increased node number will make a cluster head own more CMs on the average, thus consuming much more energy and in turn incurring a shorter system lifetime. On the other hand, the random walk model achieves a longer system lifetime than random waypoint model with the same node number. This is because the random waypoint model brings the so-called “central aggregation” effects (i.e., many nodes aggregate at the central position of deployment region), so the CHs within the proximity of central position of deployment region in random waypoint model likely cover more CMs than those in random walk model, hence making the CHs in random waypoint model more likely use up their energy and consequently incurring a shorter system lifetime.

V. CONCLUSION

In this paper, we have presented a scalable and efficient clustering scheme to solve the topology control problem for a dense mobile sensor network. Our clustering scheme has a simple cluster formation phase featuring polynomial time and message complexity. The cluster maintenance is asynchronously and event triggered so as to eliminate “ripple effect” and incur much less overheads compared with other traditional clustering methods. In summary, our proposed clustering scheme achieves a comparatively stable cluster structure in the presence of node mobility and consequently accommodates the changing underlying topology well under the mobile and dense sensor network environment, which makes it have a good performance on network management and scalability.

REFERENCES


