Energy-Aware Connected Dominating Set Construction in Mobile Ad Hoc Networks

Bonam Kim*, Junmo Yang*, Dong Zhou†, and Min-Te Sun*
*Department of Computer Science and Software Engineering, Auburn University, Auburn, Alabama 36849–5347
Email: {kimbona, yangjun, sunmint}@eng.auburn.edu
†Department of Computer Science and Engineering, Ohio State University, Columbus, Ohio 43210–1277
Email: zhoudo@cse.ohio-state.edu

Abstract—Connected dominating set (CDS) has been used widely in mobile ad hoc networks by numerous routing, broadcast, and time synchronization protocols. Although computing minimum CDS is known to be NP-hard, many protocols have been proposed to construct a sub-optimal CDS. However, these protocols are either too complicated, need non-local information, are not adaptive to topology changes, or fail to consider the difference in energy consumption for nodes within and outside the CDS. In this paper, we present two Timer-based Energy-aware Connected Dominating Set Protocols. Our protocols extend the Mac-layer Timer-based Connected Dominating Set protocol (MTCDS) so that the energy level at each node is taken into consideration when constructing the CDS. As with MTCDS, our protocols are able to maintain and adjust the CDS when the network topology is changed. Simulation results have shown that our protocols effectively construct an energy-aware CDS with a very competitive size and prolong the network operation under different levels of nodal mobility.

I. INTRODUCTION

The connected dominating set (CDS) has been used extensively as core or virtual backbone [1] in mobile ad hoc networks (MANETs). A dominating set is a subset of nodes in a graph such that each node not in the subset has at least one direct neighbor that belongs to the subset. If the nodes in the dominating set form a connected graph, the set is called connected dominating set. It has been found extremely useful in routing [2] [3] [4] [5] [6], message broadcast [7] [8] [9], and collision avoidance [10].

Due to the nature of MANETs, it is impractical for any MANET protocol to assume a single node with the global view of the network acting as the coordinator. Hence, a good CDS protocol for MANETs should be fully distributed. In addition, it should possess the following properties.

- The resulting CDS should be as small as possible – The impact of the size of the CDS is two folds. First, when the network topology changes due to nodal movements, a smaller CDS is easier to maintain. Second, the size of the CDS is closely related to the size of the routing table at each node. The smaller the CDS is, the smaller routing table each node has and thus more efficient the message communications can be achieved.
- The CDS protocol should take into account the energy level at each nodes – In addition to nodal mobility, the energy level of nodes in a CDS is an important factor in determining the lifespan of the CDS. Since the process of constructing a CDS is in general costly and time-consuming, in the case of a static network it is desirable to prolong the lifespan of the CDS in order to avoid the need to reconstruct the CDS shortly. Moreover, in the above routing and collision avoidance protocols, nodes in the CDS are commonly used to forward more packets and participate in traffic management for the network, so they are likely to consume more energy than nodes not in the CDS. In cases where nodal mobility frequently renders the existing CDS obsolete, the CDS protocol should act as a key to evenly distribute the energy consumption to all the nodes in the network, thus allowing the network to remain operational for longer periods of time.
- The protocol should avoid introducing extra messages – Bandwidth is a precious resource in wireless networks, and introducing any extra messages may degrade the performance of the system significantly. If possible, the protocol should be carefully designed so that the necessary local information can be collected solely via beacons.
- The protocol should adapt to station mobility – The protocol should maintain and incrementally adjust the CDS under changes of network topology caused by nodes either leaving or joining the network after the construction of CDS.

Most of the distributed CDS protocols, such as [3] [4] [5] [6] [11], failed to put energy level at each node into consideration when constructing the CDS. In [12], Wu et al. proposed an energy-aware CDS protocol, but their protocol requires the introduction of extra messages. In this paper, we present two versions of the Timer-based Energy-Aware Connected Dominating Set Protocols (TECDS). TECDS protocols are extensions of the MAC-layer Timer-based CDS protocol [3] (MTCDS). Our TECDS protocols enjoy every benefit of the MTCDS protocol, such as no introduction of additional messages and capable of generating CDS with very competitive size. In addition, just like MTCDS, in our TECDS protocols the required information for the protocols is strictly obtained via beacon exchanges. The major difference between the TECDS and MTCDS protocols is that the energy level at each node is used in both the initiator election and the CDS construction phases, which allows the TECDS protocols to produce better energy-aware CDS. First, in the initiator election phase the node with the most neighbors and highest energy level is elected as the initiator. In the CDS construction phase, each candidate node sets up a timer based on the number of uncovered neighbors and its own energy level, and determines whether or not to join the CDS when the timer
expires. Our TECDS protocols are simple, distributed, inexpensive (i.e., introduce no extra messages and computation), and adaptive to nodal mobility. The simulation studies have shown that the proposed TECDS protocols yield better CDS than most of the existing CDS protocols in terms of the CDS size and also prolong the lifespan of the network.

The remainder of this paper is structured as follows. In Section 2 we review the existing distributed and energy aware CDS protocols. In Section 3, we present the pseudo code used for the TECDS protocols and explain how the TECDS protocols work. The simulation results and analysis are provided in Section 4. Finally, we conclude the paper in Section 5.

II. RELATED WORK TO THE CONNECTED DOMINATING SET

Constructing minimum CDS for an arbitrary graph is known to be NP-hard [13] [14]. The problem becomes more challenging when the knowledge of complete network topology is not available prior to computation, which is a practical assumption in MANETs. Hence, the distributed CDS protocols proposed in the past have settled with constructing a “smaller” CDS for MANETs based on local information available at each node. Among them, the most noticeable protocols are [3] [5] [12].

The CDS protocol proposed in [5] obtains the CDS by eliminating unnecessary nodes from the network. It is basically a two-phase approach. In the first phase, a node exchanges the neighbor list with its neighbors. If a node finds that all its neighbors are neighbors to each other, it removes itself from the consideration of the CDS. In the second phase, some heuristic rules are applied to further reduce the size of the CDS. The protocol is simple, distributed, and most of time computes a CDS with a small size. However, the protocol requires immediate neighbors to exchange neighbor list among one another. In addition, when the network topology changes, the protocol does not have a mechanism to maintain the CDS.

In the Mac-Layer Time-based Connected Dominating Set Protocol [3] (MTCDS), the node with minimum MAC address in the network is first elected as the initiator. Propagated from the initiator, each node then sets up a timer based on the number of uncovered neighbors and determines whether or not to join the CDS when its timer expires. Using the timer, nodes with more neighbors are more likely to be included in the CDS in the MTCDS protocol. It has been shown in [3] that the MTCDS constantly produces the smallest CDS among different protocols and is suitable for MANETs because it does not require extra messages and has been shown to be able to incrementally adapt to the changes of network topology.

Since energy (i.e., battery power) is a limited resource for nodes in MANETs, the energy-aware protocols for MANETs [15] [16] have always received special attention. In general, nodes in the CDS forward more packets and participate network management, so they tend to consume more energy than those outside of the CDS. However, none of the above CDS protocols takes nodal energy into consideration when constructing the CDS. In [12], Wu et al proposed an extended marking process that constructs an energy-aware CDS for MANETs. This extended marking process aimed at both reducing the size of CDS and evenly distributing the energy consumption to all nodes in the network. In [12], the simulation results show that Wu’s newer protocol allows the network to go through more number of CDS reconstruction, an indication that the protocol prolongs the operation of the network. However, similar to [5], this protocol also requires exchanges of extra messages between immediate neighbors.

If we are able to improve the MTCDS protocol to be energy-aware, the resulting protocol would enjoy every benefit of the original MTCDS protocol and also be able to prolong the network operation. Based on this idea, we propose two versions of Time-based Energy-aware Connected Dominating Set Protocols (TECDS) in this paper.

III. TIMER-BASED ENERGY AWARE CONNECTED DOMINATING SET PROTOCOL

Like MTCDS, Our Timer-based Energy aware Connected Dominating Set protocols (TECDS) have two phases: initiator election and CDS construction. In the first phase, an unique initiator is elected; in the second phase, the CDS is constructed rooted from the initiator. The major difference between MTCDS and our TECDS protocols is that the energy level at each node is taken into consideration in both phases. In Subsections III-B and III-C, we will elaborate on the operation of each phase in detail.

A. Notation Definitions and Assumptions

Before introducing the TECDS protocols, we would like to present the following notation and their definitions. These will be used in the following discussion, as well as the protocol pseudo code.

A mobile ad hoc network (MANET) is represented as an undirected graph $G = (V, E)$, where $V$ is the set of all stations in the MANET and $E$ is the edge set with $(u, v) \in E$ if and only if $u$ and $v$ are within each other’s transmission range.

If $G$ is connected, a set $DS \subseteq V$ is called a dominating set if for every vertex $v \in V - DS$, there exists a vertex $w \in DS$ such that $(v, w) \in E$. A dominating set is said to be covered if its induced graph in $G$ is connected.

A node $u \in V$ is said to be in the state of inDS, covered (by DS), or uncovered (by DS) according to the following:

- inDS: if $u \in DS$;
- covered: if $u \notin DS$ and there is an edge $(u, v) \in E$ for some $v \in DS$;
- uncovered: if $u \notin DS$ and there is no edge joining $u$ to any node in $DS$;

We assume that each node in the network has the same transmission range. Like every wireless network system, we assume that each node periodically broadcasts a beacon signal. Two types of beacon signals are used in the protocols: the regular beacon and the announce beacon. In the regular beacon, a node’s MAC address, status (i.e., uncovered, covered, or inDS), and color value (used to detect if the initiator is still active) are included in the header of the beacon. In the
"announce" beacon, a node encodes those included in the regular beacon as well as the energy level and number of neighbors for its initiator (for the possible election of a new initiator) in the header of the beacon. Notice that a broadcast message is actually a regular beacon encoded with inDS status. The reason for introducing two different beacon formats is to reduce the overhead required by the protocols. Since the announce beacon carries more information, it is larger than the regular beacon. The protocols are carefully designed so that the announce beacon is sent every initMax regular beacon period.

B. Initiator Election

The TECDS protocols are based on a similar greedy strategy to that used in MTCDS. In MTCDS, the node with the minimum MAC address is picked as the initiator. In TECDS, however, we are interested in creating a CDS with a smaller size that contains nodes with a higher energy level, so two new criteria are used when the protocol picks the initiator: the number of neighbors and the energy level. Depending on the order of consideration for these two criteria, two different versions of TECDS, namely TECDS1 and TECDS2, are introduced. In TECDS1, the node with the most energy is picked as the initiator. In cases where multiple nodes have the same energy level, the one with the most neighbors is picked as the initiator. In TECDS2, the node with the most neighbors is picked as the initiator. When multiple nodes are found to have the most neighbors, the one with the highest energy level is then elected as the initiator. In both protocols, when multiple nodes have the same number of neighbors and the same energy level, the node with the minimum MAC address is picked as the initiator to break the tie.

The following is the pseudo code for the initiator election phase of the TECDS1 protocol. This pseudo code is similar to that of MTCDS. The statements new to MTCDS are indicated in bold face fonts.

/* node i in MANET executes the following: */
Initialization:

\[
\begin{align*}
&\text{initiator}(i) \leftarrow \text{MAXINIT} \\
&\text{status}(i) \leftarrow \text{uncovered} \\
&\text{color}(i) \leftarrow 0 \\
&\text{DSTimer}(i) \leftarrow -1 \\
&\text{ODSTimer}(i) \leftarrow -1 \\
&\text{energy}(i) \leftarrow \text{energy level of node i} \\
&\text{nbrNum}(i) \leftarrow \text{number of neighbors for node i} \\
&\text{InitTimer}(i) \leftarrow \text{initMax, start InitTimer} \\
\end{align*}
\]

InitTimer expires

\[
\begin{align*}
&\text{if initiator}(i) = \text{MAXINIT then } /* initial announcement */ \\
&\text{initiator}(i) \leftarrow i \\
&\text{announce}(i, i, \text{color}(i), \text{energy}(i), \text{nbrNum}(i)) \\
&\text{InitTimer}(i) \leftarrow 2 \times \text{initMax, start InitTimer} \\
&\text{else if initiator}(i) = i \text{ then } /* initiator is selected */ \\
&\text{color}(i) \leftarrow \text{color}(i) + 1 \\
&\text{announce}(\text{initiator}(i), i, \text{color}(i), \text{energy}(i), \text{nbrNum}(i)) \\
\end{align*}
\]

Initiator Election Phase

The timers used in this protocol have an initial value of -1. A positive integer is assigned when a timer is started. The timer value will go down each beacon period. When the value reaches 0, the timer expires and the value stops at 0.

The initiator sends out an announce message every initMax number of beacon periods. It will refresh the InitTimer of other nodes, which expire after 2*initMax beacon periods. This is a soft state protocol in the sense that the expiration of InitTimer for nodes other than the initiator implies that the initiator leaves the MANET. The nodes will wait until 2*initMax to make sure that all the InitTimers expire, then the initiator election process starts again. Additionally, the color at each node indicates that the pseudo code of the initiator election phase for TECDS2 is basically the same as that above, except that the order of the criteria used for picking the initiator (i.e., the number of neighbors and energy level) is reversed. In other words, the first two if statements after the "Compare energy, neighbor numbers, and MAC ID" comment are changed to the following:
if \( \text{nbrNum}(i) < r \)

and

else if \( \text{nbrNum}(i) = r \) and \( \text{energy}(i) < e \) then

\[ \Delta T = T_{\text{max}} \cdot \frac{1}{N_{\text{uncovered}}} \cdot \frac{1}{E} \]  

(C. Time based Energy aware Connected Dominating Set Construction)

After the election of the initiator, the initiator enters DS first. It broadcasts to its neighbor information about its inDS status. A neighboring uncovered node becomes covered after receiving the message. Then the covered node calculates the \( \Delta T \) according to the following formula if it still has uncovered neighbors and starts its DSTimer.

\[ \Delta T = T_{\text{max}} \cdot \frac{1}{N_{\text{uncovered}}} \cdot \frac{1}{E} \]

In Equation 1, the term \( N_{\text{uncovered}} \) represents the number of uncovered neighbors and \( E \) is the energy level. This equation uses both the number of uncovered neighbors and the energy level at each node to compute \( \Delta T \). It is obvious that nodes with more uncovered neighbors or higher energy levels result in shorter defer times compared with nodes with fewer uncovered neighbors and lower energy level. This is the major difference between MTCDS and our TECDS protocols in the CDS construction phase.

When the DSTimer expires, the node enters DS and broadcasts to its neighbor about its inDS status. For an inDS node, the broadDS message is sent every beacon period. Again, we utilize a soft state technique to maintain the status of covered nodes. If a covered node does not receive a broadDS message for \( \delta t \) beacon periods, it implies that the dominator(s) have left. Notice that since the size of the necessary information for our protocols is fixed, there is no need for any new control messages for the TECDS protocols. All the message mentioned in the pseudo code are in fact various types of beacon signal.

Other than Equation 1, the pseudo code of the CDS construction phase for our TECDS protocols is exactly the same as that for MTCDS. For the sake of completeness, the pseudo code is presented in the following.

\[ \text{status}(i) \leftarrow \text{inDS} \]
\[ \text{broadDS}(i) \]

if ( i does not have any uncovered neighbor )

DSTimer(i) = -1

else if ODSTimer(i) = -1 then

DSTimer(i) = \( \Delta T \), start DSTimer

ODSTimer(i) = \( \Delta T \)

else if ODSTimer < \( \Delta T \) then

DSTimer(i) = \( \Delta T \), start DSTimer

ODSTimer(i) = \( \Delta T \)

Node \( i \) in inDS state :

if ( i does not have any covered neighbor ) and

(i has at least one inDS neighbor) then

status(i) = covered

DSTimer expires :

status(i) = inDS

broadDS(i)

CoveredTimer expires :

status(i) = uncovered

Connected Dominating Set Construction Phase

As long as the network topology remains stable for a period of time, it can be shown that both TECDS1 and TECDS2 always generate the CDS for any connected MANET. In addition, it can be proved that our TECDS1 and TECDS2 are capable of incrementally maintaining the CDS under changes of network topology by utilizing the color value, state of neighbors, and timers such as InitTimer. To be specific, the following four different topology changes are well supported by our protocols:

1) The initiator leaves the network.

2) A new node joins the network after the construction of the CDS.

3) A redundant inDS node can changes its status while still preserving the dominating set connection.

4) The inDS node in the CDS leaves the network.

Due to space limitations, the proof of the protocol correctness and mobility support is omitted in this paper. However, since our TECDS1 and TECDS2 are extended from MTCDS, the proof for our protocols is essentially identical to that given for MTCDS in [3].

IV. Simulation Results and Analysis

Other than TECDS protocol (i.e., TECDS1 and TECDS2), we also implemented two other CDS protocols, namely MTCDs [3], Wu’s original connected dominating set protocol [5] (referred as Wu1 hence after), and Wu’s extended protocol [12] (referred as Wu2 hence after). Wu et al actually proposed two sets of extended rules based on the node’s energy levels in [12]. We have implemented both sets of extended rules and found that they showed almost the same results in terms of the metrics used for performance comparisons (e.g., the size
of CDS, the average energy level of nodes in the CDS, etc). Hence, we only show the results of Wu’s first set of extended rules in this paper.

A. Simulation environment and metric selection

We assume a link between two nodes only if their geometric distance is less than the wireless transmission range. In our simulation, the transmission range of a single station is normalized to 1 unit of distance. Random network topologies are generated by randomly placing nodes in $4 \times 4$ and $8 \times 8$ square grids of a two-dimensional simulation area. The value of $x$ and $y$ coordinate is uniformly distributed. The value of $T_{\text{max}}$, $\text{initMax}$, and $\delta t$ is chosen to be 100, 20, and 4 time units respectively. Two scenarios are considered and simulated differently as follows.

- static network or network with low mobility – 20 different network topologies are randomly generated. The energy level at each node is generated in normal distribution with the average 7.0 and the variance 2.0. The performance of protocols is assessed by the average size of CDS, the average energy level of the nodes in the CDS, the minimum energy level of the node in the CDS, and the variance of minimum energy level of the nodes in the CDS for both $4 \times 4$ and $8 \times 8$ with various nodal densities. Notice that when the network topology is static, the minimum and average energy level of the nodes in the CDS can be considered an indication of the lifespan of the CDS.

- when nodes are mobile – the simulation starts by initializing each node the same energy level of 100 units. The nodes in the CDS are subtracted by 2.0 and the nodes not in the CDS are subtracted by 0.1 each time the CDS is reconstructed caused by nodal movement. The reconstruction keeps going until a node reaches energy level 0 and the number of rounds is counted. This number can be used to indicate how long the network remains operational after a series of CDS reconstruction. The performance of protocols is assessed by comparing the average sizes of CDS and the number of rounds under random network topologies.

B. Performance evaluation

1) Static Networks: In Figures 1 and 2, the $x$–axis represents the size of the network and the $y$–axis shows the size of the resulting CDS from the five different protocols. It is clear that MTCDS consistently generates the smallest CDS and Wu1 consistently generates the largest CDS among all the protocols for both the $4 \times 4$ and $8 \times 8$ grids. The sizes of the CDS generated by TECDS1 and TECDS2 are very close (mostly within 10%) to those generated by MTCDS. When the scale of the network increases from the $4 \times 4$ to the $8 \times 8$ grid, the performance difference between MTCDS and our energy-aware TECDS1 and TECDS2 becomes less marked. Although Wu2 significantly reduces the size of the CDS compared to Wu1, its CDS size is still 40 to 50% larger than the CDS generated by TECDS1 and TECDS2.

Fig. 1. DS size in $4 \times 4$ square : static network

Fig. 2. DS size in $8 \times 8$ square : static network

Fig. 3. DS Average Energy in a $4 \times 4$ square : static network

Fig. 4. DS Average Energy in an $8 \times 8$ square : static network
In Figures 3 and 4, the $x$–axis represents the size of network and the $y$–axis shows the average energy level of the nodes in the resulting CDS from the five different protocols. Hence, TECDS1 and TECDS2 are able to achieve an approximately 20% higher average energy level of nodes in CDS than the others for both the $4 \times 4$ and $8 \times 8$ grids by slightly increasing the CDS size from MTCDS. On the other hand, Wu1 and MTCDS have the lowest average energy level among all the protocols for both the $4 \times 4$ and $8 \times 8$ grids. It is surprising to find that even though Wu2 considers the energy level at each node, it does not significantly improve the average energy level, as shown in Figures 3 and 4 (mere 5% better than Wu1).

In Figures 5 and 6, the $x$–axis shows the size of the network and the $y$–axis is the minimum energy level of the nodes in the resulting CDS from five different protocols. From these figures, we can see that our energy-aware TECDS1 and TECDS2 protocols select the nodes with higher minimum energy levels than any of the others, while Wu1 selects the nodes having the lowest minimum energy level. These figures indicate that the CDS created by our energy-aware TECDS1 and TECDS2 live longer than any of the others under a static network. In the $4 \times 4$ grid, the minimum energy level of the nodes in the CDS generated by TECDS2 and Wu1 are about 5.1 and 2.6, respectively. This means the minimum energy level of the nodes in the CDS generated by TECDS2 is mostly 50% higher than generated that by the Wu1 protocol. For the $8 \times 8$ grid, the performance of the minimum energy level for different protocols shows a similar trend.

In Figures 7 and 8, the $x$–axis is once again the size of the network and the $y$–axis shows the variance of the minimum energy level of the nodes in the resulting CDS from the five different protocols. In these figures, TECDS1 shows a smaller variance than the other protocols when the network size is small. This implies that the performance of TECDS1 is more stable than that of other protocols when the network size is small. However, when the network size is bigger than 100 nodes, both TECDS1 and TECDS2 still demonstrate relatively smaller variances than other protocols. It is interesting to see that the variances of Wu2 fluctuate and, in general, are slightly bigger than those of TECDS1 and TECDS2 regardless of the size of the network.
2) Network with High Mobility: In Figures 9 and 10, the $x$–axis shows the size of the network and the $y$–axis is the size of the resulting CDS from five different protocols. For a $4 \times 4$ grid, the size of the CDS obtained by Wu1 is at least twice as large as the size of the CDS generated by MTCDS, regardless of the size of the network. While the size of the CDS generated by TECDS1 and TECDS2 is approximately 20 to 40% larger than that generated by MTCDS, it is always at least 20% smaller than that generated by Wu2. The same performance pattern of these protocols is shown in the $8 \times 8$ grid case.

In Figures 11 and 12, the $x$–axis represents the size of the network and the $y$–axis shows the number of rounds from the five CDS protocols. Here, the number of rounds can be thought of as the lifespan of the network when the network topology changes frequently. In both figures, it is obvious that the number of rounds obtained by our energy-aware TECDS1 and TECDS2 are 40 to 50% higher than that obtained by Wu in both the $4 \times 4$ grid and $8 \times 8$ grid scenarios. In general, there are two reasons why our TECDS1 and TECDS2 produce a higher number of rounds. First, since the size of the CDS generated by TECDS1 and TECDS2 is always smaller than that generated by Wu1 and Wu2 (as shown in Figures 9 and 10), the energy consumption of the network per unit time is considerably less if our protocols are used. Second, our protocols successfully distribute the CDS load to every node in the network, so that the lifespan of the whole network is improved.

The results show that the proposed energy-aware TECDS1 and TECDS2 protocols produce a better CDS in terms of the average/minimum energy levels of nodes in the CDS, and the number of rounds that represent the lifespan of the CDS and the network. In addition, the size of the CDS generated by TECDS1 and TECDS2 is consistently smaller than that generated by Wu1 and Wu2.

V. Conclusion

In this paper, we presented two versions of Timer-based Energy-aware Connected Dominating Set Protocols (TECDS). Our new TECDS protocols extend the MTCDS protocol so that the energy level at each node is taken into consideration when constructing the CDS. The TECDS protocols enjoy every benefit of MTCDS and effectively constructs an energy-aware CDS that prolongs the network’s operational life under different levels of nodal mobility. The simulation results have shown that our protocols consistently generate significantly smaller CDS than those proposed in [5] and [12]. Additionally, the CDS generated by our TECDS protocols consistently results in nodes with a higher energy level, which implies a longer lifespan for the CDS when the network is static. In cases where the network topology changes frequently due to high nodal mobility, the simulation results also show that our new TECDS protocols can go through a higher number of CDS reconstructions, which is a good indication that the network will remain operational for a longer period of time if either of our TECDS protocols is adopted.

REFERENCES


