An Energy-Aware Broadcast Scheme based on Connected Dominating Set for Mobile Ad Hoc Networks

I-Wei Ting, Chi-Chang Chen, Min-Yuan Tsai, Yeim-Kuan Chang

Abstract

In mobile ad hoc networks (MANETs), the straightforward broadcast scheme is the flooding algorithm which may result in a substantial amount of redundant messages contending the network bandwidth. The recently proposed connected dominating set (CDS) based broadcast schemes are promising because they are capable of reducing the number of the redundant broadcast messages. In these CDS-based broadcast schemes, only the mobile hosts (MHs) belonging to pre-computed CDS are responsible for relaying the broadcast messages. In this paper, an improved CDS scheme called energy-aware CDS (ECDS) broadcast scheme is proposed to balance the energy consumption of the network nodes. The CDS node selection algorithm in ECDS is based on a weighted utility function parameterized with the mobility and remaining power of MHs. Two types of periodical CDS update mechanisms, ECDS-I and ECDS-II, are proposed to deal with the dynamics of network topology and energy-consumption imbalance problem. ECDS-I computes only one CDS in an update cycle while ECDS-II computes multiple CDSs that are used in a simple round-robin fashion in a cycle. Simulation results show that both ECDS-I and ECDS-2 outperform the original CDS-based scheme in terms of successful broadcast rate and network lifetime.

Keywords: Network Broadcasting, Connected Dominating Set (CDS), Mobile Ad hoc Networks, Energy-Balanced Model

1. Introduction

Mobile ad hoc networks (MANETs) consist of various mobile devices called mobile hosts (MHs) such as notebooks, PDA’s, and cell phones that are capable of performing wireless communication. These MHs form a wireless network without the aid of any infrastructure such as access points or base stations. Each MH can communicate with its one-hop neighbors directly by sending broadcast messages, where the one-hop neighbors of an MH are the nodes within the transmission range of its broadcast channel. Each MH can also move to any location freely and communicate with another MH by using multi-hop wireless links at any time.

MANETs can be modeled as a graph $G = (V, E)$, where $V$ is the set of mobile hosts (MHs) and $E$ is the set of links. An edge $e = (u, v)$ exists if and only if $u$ is in the transmission range of $v$ and vice versa, where $e \in E$ and $u$ and $v \in V$. All links in $G$ are bi-directional, i.e., if $u$ is in the transmission range of $v$, $v$ is also in the transmission range of $u$. The network is assumed to be in a connected state. If it is partitioned, each component is treated as an independent network. A working group called MANET has been formed by the Internet Engineering Task Force (IETF) to stimulate research in this direction. Issues related to MANETs have been studied intensively [1][2][3].

Broadcast communication technology is frequently used to create one or more routing paths for a particular MH for sending messages to all other MHs in MANET. The well-known CDS-based broadcast scheme [4] is proposed to make use of some nodes called CDS nodes to form a virtual backbone for sending packets. The formal definition of CDS is that when a graph $G = (V, E)$ is given, CDS is a dominating set $V' \subseteq V$ such that each node in $V-V'$ is adjacent to some dominator node in $V'$ and the subgraph induced by dominating set $V'$ is connected. The
dominating set should be connected for the ease of the broadcast process that called CDS broadcasting. The advantage is to simplify the decision of re-transmission. The routing messages include unicast, multicast, and broadcast messages that can only be sent by the CDS nodes.

The main goal of most works related to CDS-based schemes is to find a smaller CDS. If the CDS size can be made smaller, the number of re-transmissions will be reduced. Thus the power consumption, redundant re-transmissions, and network contention can be reduced too. However, little attention has been given to the performance impacts caused by the mobility and energy consumption of MHs in the network. Therefore, in this paper, the proposed ECDS scheme uses a weighted utility function consisting of the information of the mobility and remaining power of the nodes for constructing CDSs. Another goal of our proposed scheme is to improve the stability of the CDS nodes, i.e., the network has a low probability of becoming disconnected. In addition, the different update methods for the ECDS-I and ECDS-II are also used to balance the energy consumption of the CDS nodes. Thus the life time of the network can be prolonged, which is verified by the simulation.

The rest of the paper is organized as follows: The related broadcast schemes are reviewed in Section 2. Proposed ECDS scheme is presented in Section 3. Simulation results are shown in Section 4. Finally, the conclusion is provided in the last section.

2. Related Work

Broadcast operation has been used to disseminate data and topology information in MANETs. The simplest flooding broadcast scheme results in broadcast storm problem [5]. Many broadcast schemes were proposed to reduce the communication overhead of broadcast in the literature. Williams et al. [6] classified the broadcast techniques into four types: flooding, probability-based, area-based, and neighbor knowledge-based schemes.

The neighbor-knowledge scheme maintains the state (forwarding or non-forwarding) on its neighborhood via periodic “Hello” messages sent by each MH. An MH finds the smallest set of forwarding nodes from its 1-hop or 2-hop neighbors. In self-pruning methods [7][8][9][10] each MH makes its local decision on forwarding status(forwarding or non-forwarding). However, in neighbor-designated methods [11][12][13] the forwarding status of each MH is determined by its neighbors. The source MH selects a subset of nodes from its 1-hop neighbors as forwarding nodes to cover all its 2-hop neighbors. In source-independent CDS-based broadcasting scheme [4][14], “Hello” message is used to collect localized topology information. Each MH executes heuristic of CDS and then determines whether it is a CDS host or non-CDS host. If an MH wants to send messages, it will broadcast the packet to all its neighbor hosts. The CDS hosts are responsible for delivering the messages and non-CDS hosts do not need to rebroadcast the messages after receiving the broadcast messages.

Finding a minimum CDS in MANETs is an NP complete problem [15]. Many approximation algorithms for solving the minimum CDS problems have been proposed in the literature [16]. In [4][14], authors proposed a distributed heuristic source-independent CDS broadcast scheme. The scheme was used in [17] for building the virtual backbone for wireless sensor networks, in [18] as the local solution selector for computing CDS iteratively, and in [19] for determining the moving paths of mobile sinks. In the scheme, the proposed election and elimination rules are used to determine if an MH should act as a CDS node (or called broadcast forwarding nodes) to relay the broadcast messages, instead of using all the nodes for broadcasting messages. These two rules ensure that the CDS nodes are connected and any non-CDS node is directly connected to at least one CDS-node and thus can receive the broadcast messages from one of the CDS nodes. All the nodes in the network are set as non-CDS nodes (i.e., their marks are set to 0) initially.

**Election Rule:** A node $x$ in the network acts as a CDS node ($x$.mark = 1) if it finds that any two of its one-hop neighbors can not reach each other directly (i.e., not connected directly).
Assume two nodes $y$ and $z$ cannot communicate directly with each other, and $y$ and $z$ are neighbors of another node $x$. Node $x$ should act as a CDS node to ensure the connectivity between $y$ and $z$. Although election rule guarantees that a CDS can be found, it may generate too many redundant broadcast messages when the node density of the network is high. For instance, two CDS nodes may have the same set of neighboring nodes calculated by election rule and thus one of them does not need to be chosen as CDS node. In general, an ID-based elimination rule that is executed after the election rule is proposed to remove the unnecessary CDS nodes as follows.

**Elimination Rule:** A CDS node $x$ is removed from CDS (i.e., set $x$.mark = 0) if there exists a subset of $k$ strongly connected CDS nodes, $S = \{v_1, ..., v_k\} \subseteq CDS$, such that (1) $N(x) \subseteq N(v_1) \cup ... \cup N(v_k)$ and (2) $x$.id < $v_i$.id for all $i = 1..k$.

To provide a distributed implementation [14] that only uses localized information to make election and elimination decisions, each host maintains its 1-hop neighbor list and collects its 2-hop neighbor lists by obtaining the 1-hop neighbor lists of all its 1-hop neighbors. Therefore, the subset of $k$ strongly connected CDS nodes, $S = \{v_1, ..., v_k\} \subseteq CDS$, in the above elimination rule should be changed to $S = \{v_1, ..., v_k\} \subseteq CDS \cap N(x)$, where $N(x)$ is the set of 1-hop neighbor of $x$, because it would be unrealistic for each host to keep global neighborhood information.

### 3. The Proposed Energy-Balanced CDS (ECDS) Scheme

Although the elimination rule in the CDS-based schemes [14] may remove many unneeded CDS nodes selected by the election rule, it does not consider various characteristics of MANETs such as node’s mobility and remaining power.

In the proposed ECDS scheme, each node is associated with a priority defined by a utility function of node mobility and remaining power. The original CDS algorithm is enhanced by using the priority as the primary factor and ID as the secondary factor in the elimination rule. The node with more remaining power, fewer neighbors, and lower mobility is given to a higher priority to become the CDS node. The proposed ECDS scheme includes three algorithms, Hello(), Update(), and ECDS(). Algorithm Hello() exchanges the 1-hop and 2-hop neighborhood information between neighboring nodes and algorithm Update() updates the 1-hop and 2-hop neighbor lists periodically to capture the possibility that some nodes may move out of the transmission range of their previous neighbors. Algorithm ECSD() selects CDS nodes in the power and mobility aware fashion.

#### 3.1. Algorithm Hello

In algorithm Hello(), nodes use Hello messages to distribute the neighborhood information of 1-hop and 2-hop neighbors. Hello message is a light-weight communication packet sent periodically between two nodes covered within their transmission range. With neighborhood information, local connectivity in MANET can be maintained. Figure 1 shows the detailed pseudo code for algorithm Hello() and Figure 2 shows the associated algorithm for updating the 1-hop and 2-hop neighbor lists of a node. Notations and their description are first explained as follows.

- $N(x)$: One-hop neighbors of node $x$.
- $N_2(x)$: Two-hop neighbors of node $x$.
- Mark: The attribute mark (1 or 0) associated with a node indicates that this node is a CDS node or not after executing the election and elimination rules.
- $N(x)$.y: The status of node $y$ recorded in $x$, where $y \in N(x)$. $N(x)$.y consists of the attributes, $N(x)$.y.signal (the signal strength from $y$), $N(x)$.y.speed (speed of $y$ relative to $x$), $N(x)$.y.mark (indication of node $y$ being a CDS node (mark = 1) or not (mark=0)), $N(x)$.y.1-hop (1-hop neighbors of $y$), $N(x)$.y.priority (priority of $y$ estimated by $y$ itself), $N(x)$.y.time
An Energy-Aware Broadcast Scheme based on Connected Dominating Set for Mobile Ad Hoc Networks
I-Wei Ting, Chi-Chang Chen, Min-Yuan Tsai, Yeim-Kuan Chang

// Node \(x\) computes its priority, 1-hop neighbor list, and 2-hop neighbor list, \(N(x)\) and \(N_2(x)\), after receiving a “Hello” message denoted as \(H(y)\) from its 1-hop neighbor \(y\).

Algorithm Hello(\(H(y)\))

\[
\begin{align*}
01: & \quad N(x) = N(x) \cup \{y\} \\
02: & \quad N(x) = (N(x) \cup H(y).I-hop) - N(x) - \{x\}; \\
03: & \quad \text{if} \ (N(x)_y.signal \neq 0) \\
04: & \quad N(x)_y.speed = \left\lfloor \text{dist}(H(y)_y.signal) - \text{dist}(N(x)_y.signal)/(current\_time - N(x)_y.time) \right\rfloor; \\
05: & \quad \text{else} \ N(x)_y.speed = 0; \\
06: & \quad N(x)_y.signal = H(y)_y.signal; \\
07: & \quad N(x)_y.I-hop = H(y)_y.I-hop; \\
08: & \quad N(x)_y.priority = H(y)_y.priority; \\
09: & \quad N(x)_y.time = current\_time; \\
10: & \quad N(x)_y.mark = H(y)_y.mark; \\
11: & \quad \text{// Update the expiration time of } y \\
12: & \quad N(x)_y.expire\_time = current\_time + alive\_period; \ // \text{ex: alive\_period = 3 seconds} \\
13: & \quad \text{for each } v \in N(x) \{ x.speed += N(x)_v.speed; \} \\
14: & \quad x.speed = x.speed/|N(x)|; \\
15: & \quad \text{for each } v \in H(y).1-hop - N(x) - \{x\} \} \\
16: & \quad N_2(x)_v.expire\_time = current\_time + alive\_period; \\
17: & \quad N_2(x)_v.mark = H(y)_y.I-hop.v.mark; \\
18: & \quad x.priority = \frac{(x\_remaining\_power)^{WP} \cdot (x.speed)^{WS}}{W_P \cdot W_S}; // W_P \text{ and } W_S \text{ are pre-defined parameters}
\end{align*}
\]

Figure 1. This algorithm is invoked every time a node receives a hello message.

// Update() is run periodically (say every second) to remove the disconnected MHs from \(N(x)\) // and \(N_2(x)\).

Algorithm Update()

\[
\begin{align*}
01: & \quad \text{for each } MH v \in N(x) \\
02: & \quad \text{if } (current\_time > N(x)_v.expire\_time) \text{ remove } v \text{ from } N(x). \\
03: & \quad \text{for each } MH y \in N_2(x) \\
04: & \quad \text{if } (current\_time > N_2(x)_v.expire\_time) \text{ remove } v \text{ from } N_2(x).
\end{align*}
\]

Figure 2. Update algorithm for \(N(x)\) and \(N_2(x)\).

(the time when the hello message from \(y\) is received), and \(N(x)_y.expire\_time\) (the expiration time when the connection between \(y\) and \(x\) is considered to be valid).

- \(N(x)_y.speed\): This speed of node \(y\) relative to node \(x\) is computed by using the information of two successive “Hello” messages received from \(y\) as follows.

\[
N(x)_y.speed = \frac{current\_time - N(x)_y.time}{\text{dist}(H(y)_y.signal) - \text{dist}(N(x)_y.signal)}
\]

where \(\text{dist}(H(y)_y.signal)\) and \(\text{dist}(N(x)_y.signal)\) are the current and last distances between nodes \(x\) and \(y\) computed based on two successive “Hello” messages. The wireless energy model that can be used to measure the distance between two nodes can be found in [20]. The timestamps when the two successive “Hello” messages are received are denoted by \(current\_time\) and \(N(x)_y.time\), respectively. We assume GPS is not equipped with any node in the network.

- \(H(y)\): The “Hello” message received from \(y\) consists of signal strength \((H(y)_y.signal)\), \(y\)’s speed \((H(y)_y.speed)\), \(y\)’s 1-hop neighbor list \((H(y)_y.I-hop)\), and \(y\)’s priority \((H(y)_y.priority)\).
The priority of a node is computed based on a weighted utility function defined as follows.

\[
x.\text{priority} = \frac{(x.\text{remaining}\_\text{power})^{WP}}{(x.\text{speed})^{WS}}
\]

(2)

where parameters \(WP\) and \(WS\) are adjustable weights according to the characteristics of the network. When the remaining power is not considered, equation (2) is changed by setting \(WP\) to zero. However, when the node moving speed is zero as in the static network, parameter \(WS\) in equation (2) is set to zero.

We use the priority to eliminate the redundant hosts that has been selected as the dominators in the election rule. Each node can calculate its own priority by the priority equation, and then exchange its priority value with its neighbors by the “Hello” messages. The process of CDS selection in ECDS scheme is to determine which nodes should be put in a CDS with a high priority. First, it is better to forward broadcast messages through the hosts that have sufficient remaining power. Thus, the priority of each node can be set to be directly proportional to its remaining power. If power consumption balance among all nodes is preferred, a larger \(WP\) can be used compared to \(WS\). Second, in MANET, nodes can move around freely. When a node moves in a high speed, it is highly probable that it will move out its neighbors’ transmission range and thus, the link between two nodes may break frequently. As a result, some CDS-hosts may be disconnected from other hosts. From this observation, a node with a low speed is stable in the network. Thus, the node speed is put in the denominator of the utility function. The goal of the utility function is to select CDS-hosts that have a low moving speed (i.e., mobility) and high remaining energy. As a result, based on the proposed ECDS, the network can be kept connected for a long time.

### 3.2. Algorithm ECDS

Figure 3 illustrates the pseudo code of algorithm ECDS which is executed by each node to determine if it should act as a CDS node or not. In algorithm ECDS, all nodes have to compute their priorities based on equation (2). The first four lines execute the original CDS’s election rule to ensure the full coverage of broadcast operations. However, the original elimination rule is replaced by the proposed extended elimination rule as follows:

**Extended Elimination Rule:** A CDS node \(x\) is removed from CDS (i.e., \(x.\text{mark} = 0\)) if there exists a subset of \(k\) strongly connected CDS nodes, \(S = \{v_1, ..., v_k\}\), satisfying the following three conditions: (1) \(S \subseteq \text{CDS}\cap\text{N}(x)\cap\text{N}^2(x)\), (2) \(\text{N}(x) \subseteq \text{N}(v_1)\cup...\cup\text{N}(v_k)\), and (3) \((x.\text{priority} < v_i.\text{priority})\) or \((x.\text{priority} = v_i.\text{priority}\) and \(x.\text{id} < v_i.\text{id}\)) for all \(i = 1..k\).

The proposed extended elimination rule is similar to the original elimination rule with the following two differences. First, the node priority is included as the primary factor to eliminate CDS nodes. The node ID is used as the secondary elimination factor because it is possible that the priorities of two nodes may be the same although it is rare. Second, the set of \(k\) strongly connected CDS nodes, \(S = \{v_1, ..., v_k\}\), is the subset of \(\text{CDS}\cap\text{N}(x)\cap\text{N}^2(x)\) instead of \(\text{CDS}\cap\text{N}(x)\) in the original elimination rule provided with only 2-hop neighborhood information. In other words, these CDS nodes that covers all the 1-hop neighbors of node \(x\) must belong to the 1-hop and 2-hop neighbors of \(x\). Figure 4 shows an example network in which it is assumed that the priority order is \(y < u < v_1 < v_2\). The marked nodes are shown in gray color after the election rule is applied. After the extended elimination rule is applied, nodes \(y\) and \(u\) are unmarked because \(y\)’s neighbors are covered by \(v_2\) and \(u\)’s neighbors are covered by \(v_1\) and \(v_2\). We can see that \(v_2\) is \(y\)’s 2-hop neighbor and \(v_1\) is \(u\)’s 2-hop neighbor. However, if the original elimination rule is applied, neither \(y\) nor \(u\) can be unmarked. Also, when the priorities of nodes \(y\) and \(u\) decrease after a portion of their power is consumed, nodes \(v_1\) and \(v_2\) can then be selected as the CDS nodes.
3.3. Multiple CDSs

Algorithm ECDS() computes only one CDS whose size is minimized by the elimination rule. As a result, the CDS nodes will exhaust their power faster than non-CDS nodes. Therefore, it is better to compute multiple CDSs, and use them alternatively to balance the power consumption of all the nodes in the network. The simplest way is to compute multiple CDSs as follows. The same algorithm is run repeatedly in a number of rounds. In each round, the nodes that are already selected as CDS nodes in the previous rounds are set to the lowest priorities (i.e., zero). Note that it may be impossible to find disjoint CDSs because one node called gateway node may be the only gateway between two portions of the network. Our simulation results show that when the number of MHs is large, the number of gateway nodes among the CDSs is low.

In MANETs, network topology changes frequently due to the movements of MHs. Thus, the CDS selected by the algorithm ECDS() may not cover all the non-CDS hosts for a very long time. The reasons that cause the topology changes are (1) MHs’ switch on/off at will and (2) MHs’ move freely. It is better to have a mechanism to update/recompute the CDS periodically to prevent MHs from being disconnected from the other. We propose two CDS update scheme (ECDS-I and ECDS-II) to reflect the dynamics of network topology. ECDS-I is designed to

---

**Figure 3.** The pseudo code of algorithm ECDS.

```plaintext
// Assume node x executes algorithm ECDS to decide if it is a CDS node or not.
Algorithm ECDS()
{
    // Rule 1: Election Rule
    x.mark = 0; // node x is marked as a non-CDS node initially;
    for any subset \{u_i, u_j\} of N(x)
        if (u_i is not the 1-hop neighbor of u_j) // x.mark = 1; // node x is marked as the CDS node;
        break; // break the for loop
}
```

**Figure 4.** Unmarking process in the elimination rule with 2-hop neighbor information.

### 3.3. Multiple CDSs

Algorithm ECDS() computes only one CDS whose size is minimized by the elimination rule. As a result, the CDS nodes will exhaust their power faster than non-CDS nodes. Therefore, it is better to compute multiple CDSs, and use them alternatively to balance the power consumption of all the nodes in the network. The simplest way is to compute multiple CDSs as follows. The same algorithm is run repeatedly in a number of rounds. In each round, the nodes that are already selected as CDS nodes in the previous rounds are set to the lowest priorities (i.e., zero). Note that it may be impossible to find disjoint CDSs because one node called gateway node may be the only gateway between two portions of the network. Our simulation results show that when the number of MHs is large, the number of gateway nodes among the CDSs is low.

In MANETs, network topology changes frequently due to the movements of MHs. Thus, the CDS selected by the algorithm ECDS() may not cover all the non-CDS hosts for a very long time. The reasons that cause the topology changes are (1) MHs’ switch on/off at will and (2) MHs’ move freely. It is better to have a mechanism to update/recompute the CDS periodically to prevent MHs from being disconnected from the other. We propose two CDS update scheme (ECDS-I and ECDS-II) to reflect the dynamics of network topology. ECDS-I is designed to
focus on the nodes moving at high speed. Thus, ECDS-I computes only one CDS in a cycle and recomputes a new CDS in next cycle. ECDS-II is designed for balancing the energy consumed by multiple CDS’s to retransmit the broadcast messages. ECDS-II computes multiple CDSs in each cycle and a simple round-robin rotation scheme is employed in the cycle.

a. ECDS-I

The design principle of ECDS-I focuses on the dynamic mobility of the MHs in MANETs. The main concept of update is that when the algorithm ECDS() is performed by each MH, only one CDS will be computed. After a period of $T_{Period}$ seconds called update cycle period, a new CDS will be computed and used as the forwarding nodes. Setting $T_{Period}$ to a small value can ensure that the CDS nodes can always cover all nodes in the network.

b. ECDS-II

In ECDS-II, each MH calculates $N$ CDSs at a time. The cycle of period $T_{Period}$ is divided into $T_{Period}/\lambda$ time slots, where $\lambda$ is the time interval for each time slot. Then, $N$ CDSs are used in a round robin fashion to forward the broadcast messages in the network. Figure 6 shows an example that computes two CDSs and each cycle is divided into six time slots. These two CDSs are selected alternatively for these six time slots. After $T_{Period}$ seconds, these two CDSs are recomputed at time $t_1$ and used alternatively in the next cycle.

4. Performance Evaluation

4.1. The simulation model

We use network simulator ns-2 [21] to evaluate the performance of the proposed ECDS and the original CDS. The simulation parameters are listed in Table 1. The 802.11 MAC layer is used as the basis and the wireless bandwidth is set to 2MB/s. The broadcast model without RTS/CTS/ACK mechanisms is employed for all message transmissions, including “Hello”, Data, ACK messages in real wireless channels. Interface Queues (IFQ) have a length of 50 packets for buffering the received packets. The well-known Ad hoc On-demand Distance Vector (AODV) routing protocol is used as the underlying routing protocol.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Model</td>
<td>Random way point</td>
<td></td>
</tr>
<tr>
<td>Network area (meter × meter)</td>
<td>1,000×1,000</td>
<td></td>
</tr>
<tr>
<td>Bandwidth (MB/s)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td># of Mobile hosts</td>
<td>50</td>
<td>50, 100, 150, 200</td>
</tr>
<tr>
<td>Transmission range (meter)</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>MH speed (meter/s)</td>
<td>6</td>
<td>0, 6, 10, 20</td>
</tr>
<tr>
<td>Data size (Byte)</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>Pause time (second)</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Initial Energy (Joule)</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>Transmission power Consumption (Watt)</td>
<td>1.6</td>
<td></td>
</tr>
<tr>
<td>Reception power Consumption (Watt)</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>Idle power Consumption (Watt)</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>CDS’s update cycle period ($T_{Period}$ seconds)</td>
<td>10</td>
<td>$\infty$ or 10-60</td>
</tr>
<tr>
<td>ECDS-I’s update cycle period ($T_{Period}$ seconds)</td>
<td>10</td>
<td>$\infty$ or 10-60</td>
</tr>
<tr>
<td>ECDS-II’s update cycle period ($T_{Period}$ Seconds)</td>
<td>30</td>
<td>$\infty$ or 10-60</td>
</tr>
<tr>
<td>$\lambda$ (time slot interval) of ECDS-II (second)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>$W_p$ and $W_s$ of ECDS-I and ECDS-II</td>
<td>3, 2</td>
<td>0-5, 0-5</td>
</tr>
</tbody>
</table>

We use ECDS-X($W_p$, $W_s$, $T_{period}$) to denote the parameter setup for the proposed schemes ECDS-X, where X = I or II. Since the original CDS scheme can also recompute a new CDS after a period of $T_{period}$, it is denoted by CDS($T_{period}$). In the experiments, CDS($\infty$) (i.e., no CDS recomputation in the
original CDS scheme), CDS(10), and CDS(30) are evaluated for comparison. In the experimental results shown below, the default settings of the proposed ECDS schemes are ECDS-I(3, 2, 10) and ECDS-II(3, 2, 30). We will evaluate the performance of the proposed schemes under the settings of various combinations of WP, WD, and WS. As a matter of fact, ECDS-I(3, 2, 10) and ECDS-II(3, 2, 30) perform better than most of the other settings as we will show later.

4.2. Results

We first determine how many CDSs (i.e., N) are appropriate for ECDS-II scheme. If N is too large and the node density in the network is low, there may exist a large number of gateway nodes. Thus, many nodes that are selected in different CDSs will be the same. If it is the case, it will not only contradict the goal of balancing traffic loads and power consumption among MHs but also waste the resource to re-compute CDSs. Based on the selection algorithm of ECDS, we measure the number of selected nodes and duplicated nodes when N is set to 1 to 5. According to our simulation experiment, N = 2 is a better choice for any network. Thus, in the experimental results presented below, only two CDSs are utilized for the proposed ECDS-II scheme.

4.2.1. Average CDS Size (ACS)

Average CDS size (ACS) is defined to be \( \sum b / N \), where b is the number of MHs in each of N CDSs. If ACS is low, the overall consumption of energy and wireless bandwidth will be reduced. Figures 5(a) to (d) show the results of ACS under node speed at 0, 6, 10, and 20 m/s for CDS(\( \infty \)), CDS(10), CDS(30), ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30) with time interval \( \lambda = 5 \). Notice that the power issue is ignored by setting \( W_p \) to 0 in ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30). As stated above, two CDSs are utilized in ECDS-II(0, 2, 30). We can see that all the schemes have a slight ACS growth as the network size increases. For the case of node speed = 0 and network size = 100/150/200, ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30) need more nodes to form the connected dominated sets than the original CDS schemes. In some cases of node speed = 20, the ACSs of ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30) are even smaller than the original CDS schemes. However, for most of cases, the differences in terms of ACS among CDS(\( \infty \)), CDS(10), CDS(30), ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30) are not big.

![Figure 5. Average CDS sizes.](image-url)
4.2.2. Successful Broadcast Rate (SBR)

Successful broadcast rate (SBR) is defined to be \( R/n \), where \( R \) is the number of MHs that receive the broadcast messages successfully and \( n \) is the total number of MHs in the network. Based on SBR, our focus is to investigate how much is the impact of network mobility on the rate of messages that can be transmitted to all MHs successfully. SBR can be considered as the network stability when a broadcast scheme is employed to deal with the node disconnection problem caused by node mobility. Good broadcast schemes for MANETs should have large SBR values.

Since the power issue is inextricable from other aspects of the proposed broadcast schemes, we study the impact of network mobility in terms of SBR while the power issue is not involved. The remaining power factor is ignored in the utility function of priority by setting weight \( W_P \) in equation (2) to zero. Figure 6 shows the results of SBR for CDS(\( \infty \)), CDS(10), CDS(30), ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30) with \( \lambda = 5 \) with node speeds of 0, 6, 10, and 20 m/s. We can observe from Figure 8 that ECDS-I(0, 2, 10), and ECDS-II(0, 2, 30) perform better than the CDS-based schemes, especially when the maximum node speed is high. When the node speed is set to 20 meters per second, ECDS-I and ECDS-II have 23-42% of total number of nodes that can receive the broadcast messages successfully more than the original CDS scheme. In general, ECDS-I and ECDS-II have higher stabilities than the CDS-based schemes in a sparse and high speed environment. Notice that ECDS-I performs better than ECDS-II in the cases of high node speeds since ECDS-I recomputes CDS more frequently than ECDS-II, i.e., ECDS-I is more sensitive to the topology change than ECDS-II.

![Figure 6. Successful broadcast rates under different node speeds.](image)

4.2.3. Living Node Rate (LNR)

Living Node Rate (LNR) is defined to be \( a/n \), where \( a \) is the number of MHs that are alive and \( n \) is the total number of MHs in the network. By this metric, we can clearly know the network condition. A higher LNR means a larger number of living nodes and thus a smaller amount of power that has been consumed in the network. As a result, the network is not easy to be partitioned. Figure 7 show the LNR results for the networks of 50 to 200 nodes with the node speed of 6 meters/s. As we can see, the first
abrupt LNR drop happens almost at the same time, i.e., at 185, 270, and 290 seconds, for CDS-based schemes (CDS(∞), CDS(10), and CDS(30)), ECDS-I(3, 2, 10), and ECDS-II(3, 2, 30) with $\lambda = 5$, respectively. CDS-based schemes exhaust the power of a large number of MHs in about 185 seconds while almost all the MHs in ECDS-I(3, 2, 10), and ECDS-II(3, 2, 30) are alive in the same period. Not until about 270 and 290 seconds (the time when LNR drops abruptly) for ECDS-I and ECDS-II, respectively, there is no significant number of nodes that become dead. In other words, ECDS-I and ECDS-II improve the lifetime of fully functional network over CDS-based schemes by 46% and 57%, respectively. Obviously, the strength of ECDS-based schemes is that the power consumption can be balanced among MHs. Therefore, when the CDS-based broadcast scheme has a large number of MHs vanishing in the network, ECDS-based schemes still have a larger number of MHs to be responsible for forwarding messages to the entire network. Although the first abrupt LNR drops for ECDS-I or ECDS-II happen almost at the same time, the drop rates are different in terms of network sizes, which are 38%, 32%, 22%, and 19% for the network of 50, 100, 150, and 200 nodes, respectively. In general, the larger the network is, the less the LNR drop rate becomes.

5. Conclusions

Power consumption is always an important issue in MANETs since most MHs operate and rely on the battery. Broadcasting messages based on a connected dominating set (CDS) of nodes in a network is promising to be an efficient scheme. In this paper, we propose an energy-aware CDS scheme (ECDS) broadcast scheme to extend the lifetime of the network by balancing the energy consumption among MHs.

ECDS constructs CDS by using a utility function parameterized with only two factors: remaining power and node speed. The parameters $W_P$ associated with the power factor and $W_S$ associated with the node speed (i.e., mobility) in the utility function can be adjusted to adapt various network scenarios. The proposed ECDS-I and ECDS-II schemes are much better than the original CDS-based scheme in terms of number of nodes that successfully receive the broadcast messages and the lifetime of a fully functional network.

Our simulation experiments in NS-2 show the following results. When power issue is not considered and node speed is set to 20 meters per second, ECDS-I and ECDS-II have 23–42% of the total number of nodes that can receive the broadcast messages successfully more than the

![Figure 7. Living node rates in various network sizes with speed = 6 meters/s.](image)
original CDS scheme. Also, ECDS-I and ECDS-II improve the lifetime of fully functional networks over the original CDS-based schemes by 46% and 57%, respectively.

6. References