Performance Comparison Study of Connected Dominating Set Algorithms for Mobile Ad hoc Networks under Different Mobility Models

Natarajan Meghanathan and Ilin Dasari
Jackson State University, Jackson, MS 39217, USA
Corresponding Author E-mail: natarajan.meghanathan@jsums.edu

Abstract

The high-level contribution of this paper is an exhaustive simulation-based comparison study of three categories (density, node id and stability-based) of algorithms to determine connected dominating sets (CDS) for mobile ad hoc networks and evaluate their performance under two categories (random node mobility and grid-based vehicular ad hoc network) of mobility models. The CDS algorithms studied are the maximum density-based (MaxD-CDS), node ID-based (ID-CDS) and the minimum velocity-based (MinV-CDS) algorithms representing the density, node id and stability categories respectively. The node mobility models used are the Random Waypoint model (representing random node mobility) and the City Section and Manhattan mobility models (representing the grid-based vehicular ad hoc networks). The three CDS algorithms under the three mobility models are evaluated with respect to two critical performance metrics: the effective CDS lifetime (calculated taking into consideration the CDS connectivity and absolute CDS lifetime) and the CDS node size. Simulations are conducted under a diverse set of conditions representing low, moderate and high network density, coupled with low, moderate and high node mobility scenarios. For each CDS, the paper identifies the mobility model that can be employed to simultaneously maximize the lifetime and minimize the node size with minimal tradeoff. For the two VANET mobility models, the impact of the grid block length on the CDS lifetime and node size is also evaluated.

Keywords: Connected Dominating Sets, Mobile Ad hoc Networks, Algorithms, Mobility Models, Stability, Simulations

1 Introduction

A Mobile ad hoc network (MANET) is a category of wireless networks in which the topology changes dynamically with time due to arbitrary node movement. The MANET nodes have limited battery charge and hence operate with a restricted transmission range to conserve energy as well as avoid interference and collisions that may result from transmitting over longer distances. As a result, MANET routes are often multi-hop in nature and also temporally change depending on node mobility and availability. MANET routing protocols (for unicasting, multicasting and broadcasting) have been predominantly designed to be on-demand in nature rather than using a proactive strategy of determining the routes irrespective of the need [1][2].

MANET on-demand routing protocols typically use a network-wide broadcast route-reply cycle (called flooding) to discover the appropriate communication topology (path, tree or mesh) [3]. A source node initiates the flooding of the Route-Request (RREQ) packets that propagate through several paths and reach the targeted destination (for unicasting) or receiver nodes (for multicasting). These nodes choose the path that best satisfies the principles of the routing protocol in use and respond with a Route-Reply (RREP) packet to the source on the selected route. With flooding, every node in the network is required to broadcast the RREQ packet exactly once in its neighborhood. Nevertheless, the redundancy of retransmissions and the resulting control overhead incurred with flooding is still too high, as every node spends energy and bandwidth to receive the RREQ packet from each of its neighbors.
Vehicular ad hoc networks (VANETs) are one of the most promising application areas of MANETs. VANET communication is normally accomplished through special electronic devices placed inside each vehicle so that an ad hoc network of the vehicles is formed on the road. A vehicle equipped with a VANET device should be able to receive and relay messages to other VANET-device equipped vehicles in its neighborhood. VANET applications can be broadly classified into two categories: safety applications and comfort applications. An example of a safety application is on-board active safety systems to assist drivers with information (like accidents, road surface conditions, intersections, highway entries and etc) about the road ahead. Comfort applications are those applications that can provide non-critical services like weather information, gas station or restaurant locations, mobile e-commerce, Internet access, music downloads and etc.

VANETs resemble MANETs with respect to the dynamically and rapidly changing network topologies due to fast moving vehicles. However, the mobility of the vehicles is normally constrained by predefined roads and speed limitations. Mobility of the vehicles is also affected due to traffic congestion in the roads and the traffic control mechanisms (like stop signs and traffic lights). Route stability is an important design criterion to be considered in the design of MANET and VANET routing protocols.

A Connected Dominating Set (CDS) of a network graph comprises of a subset of the nodes such that every node in the network is either in the CDS or is a neighbor of a node in the CDS. Recent studies have shown the CDS to be a viable backbone for network-wide broadcasting of a message (such as the RREQ message), initiated from one node. The message is broadcast only by the CDS nodes (nodes constituting the CDS) and the non-CDS nodes (who are neighbors of the CDS nodes) merely receive the message, once from each of their neighboring CDS nodes. The efficiency of broadcasting depends on the CDS Node Size that directly influences the number of redundant retransmissions. One category of CDS algorithms for MANETs aim to determine CDSs with reduced Node Size (i.e., constituent nodes) such that whole network could be covered as fewer nodes as possible. However, as observed in this paper and previous work, such minimum node size-driven CDS have been observed to be unstable (i.e., have a limited lifetime). The other two categories of CDS algorithms are those based on node ids and those based on stability. The node-id based algorithms prefers to include nodes with lower id (the neighbor nodes of a CDS node are covered by the CDS node) until all the nodes in the network are covered. The stability-based category of CDS algorithms focus on maximizing the CDS Lifetime such that the control overhead incurred in frequently reconfiguring a CDS is reduced. The paper identifies any potential CDS lifetime-tradeoff existing for the three categories of CDS algorithms.

The maximum density-based CDS (MaxD-CDS) and ID-based CDS (ID-CDS) are picked to be representatives of the minimum node size and node id-based CDS algorithms respectively; the minimum velocity-based CDS (MinV-CDS) algorithm is chosen to be the representative for the stability-driven CDS. The MaxD-CDS algorithm prefers to include nodes with the largest number of uncovered neighbors into the CDS; the ID-CDS algorithm prefers to include nodes with a larger node ID and having at least one uncovered neighbor into the CDS; and the MinV-CDS algorithm opts for slow moving nodes to be part of the CDS. The paper evaluates the three CDS algorithms under diverse conditions of network density and node mobility with respect to four metrics: CDS Lifetime, CDS Node Size, CDS Edge Size and the Hop Count per Path, and under three different mobility models. The mobility models considered are the Random Waypoint model that is often used to simulate the mobility of nodes in classical mobile ad hoc networks; and the City Section and Manhattan mobility models that are used to simulate the mobility of nodes in vehicular ad hoc networks. This is the first such comprehensive comparison study of the CDS algorithms for ad hoc networks under different mobility models.

The following significant observation is made in this paper about the overall performance of the three CDS algorithms, across all the mobility models: The MinV-CDS incurs a significantly larger lifetime compared to the MaxD-CDS and ID-CDS. However, the MinV-CDS requires relatively more nodes to be part of the CDS to provide stability. Nevertheless, the Node Size-Lifetime tradeoff ratio is observed to be the lowest for the MinV-CDS for a majority of the scenarios.
The rest of the paper is organized as follows: Section 2 presents a generic CDS algorithm that can be used to study each of the three CDS algorithms (MaxD-CDS, ID-CDS and MinV-CDS) that will be analyzed in the simulations. Section 3 presents the algorithm used to validate the existence of a currently known CDS at a particular time instant and decide on the need to determine a new CDS. Section 4 presents examples to illustrate the construction of the ID-CDS and MaxD-CDS. Section 5 presents examples to illustrate the construction of the MinV-CDS. Section 6 describes the three mobility models that have been used in this research for comparative studies and presents related work. Section 7 outlines the simulation environment and presents the simulation results observed for the three CDS algorithms under the three mobility models. Section 8 presents the conclusions by outlining the significant observations and contributions of this paper. Throughout the report, the terms ‘vertex’ and ‘node’, ‘edge’ and ‘link’, ‘path’ and ‘route’, ‘packet’ and ‘message’ are used interchangeably. They mean the same.

2 Generic Algorithm to Construct the Different Connected Dominating Sets

In this section, we present a generic algorithm that can be used to construct the four different CDSs, with appropriate changes in the data structure (Priority-Queue) used to store the list of covered nodes that could be the candidate CDS nodes and the use of the appropriate criteria in each of the iterations of the algorithm to evolve the particular CDS of interest. The advantage with proposing a generic algorithm is that all the four CDSs can be constructed with virtually the same time complexity.

2.1 Network Model

The network model used in this research is described as follows:
- We assume a homogeneous network of wireless nodes, each operating at a fixed transmission range, $R$.
- We use the unit-disk graph model [16] according to which there exists a link between any two nodes $i$ and $j$ at $(X_i, Y_i)$ and $(X_j, Y_j)$ in the network as long as the Euclidean distance $\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$ is less than or equal to the transmission range per node.
- The set of neighbors of a node $i$, Neighbors$(i)$, comprises of nodes that are connected to vertex $i$ in the unit-disk graph model.
- A node learns about its own location through location service schemes such as the Global Positioning System (GPS) [17] or any other scheme (e.g. [18]).
- A node learns the location and mobility parameters (velocity and direction of movement – measured as the angle subscribed with respect to the positive X-axis) of its neighbor nodes through the beacon messages periodically broadcast by their nodes in the neighborhood.

2.2 Data Structures

The data structures used by the CDS algorithm are as follows:
- $\text{CDS-Nodes-List}$: This list includes all the nodes that are part of the CDS
- $\text{Covered-Nodes-List}$: This list includes all the nodes that are either part of the CDS or is at least a neighbor node of a node in the CDS.
- $\text{Priority-Queue}$: This list includes all the nodes that are in the $\text{Covered-Nodes-List}$ (but not in the $\text{CDS-Nodes-List}$) and are considered the candidate nodes for the next node to be selected for inclusion in the $\text{CDS-Nodes-List}$. The order in which the vertices are stored in the $\text{Priority-Queue}$ varies with the CDS algorithms as described below. Any tie between the nodes is broken based on the node IDs (the contending node with a relatively larger ID is placed ahead of the other contending nodes). For the MaxD-CDS algorithm, the $\text{Priority-Queue}$ stores the covered non-CDS nodes in the decreasing order of the node density (number of uncovered neighbors); the node with the largest number of uncovered neighbors is in the front of the queue.

For the ID-CDS algorithm, the **Priority-Queue** stores the covered non-CDS nodes in the decreasing order of their node IDs and have at least one uncovered neighbor node; the node with the largest ID and has at least one uncovered neighbor node is in the front of the queue.

For the MinV-CDS algorithm, the **Priority-Queue** stores the covered non-CDS nodes in the increasing order of the node velocities and have at least one uncovered neighbor node; the node with the lowest velocity and having at least one uncovered neighbor node is in the front of the queue.

### 2.3 Auxiliary Variables

- **CDS-Select-node**: For every iteration, the candidate node (node $s$ in the pseudo code of Figure 1) selected for inclusion into the **CDS-Node-List** depends on the criteria used by each of the CDS algorithms.
  - The **CDS-Select-node** for the MaxD-CDS algorithm is the covered non-CDS node with the largest node density (i.e., the largest number of uncovered neighbors).
  - The **CDS-Select-node** for the ID-CDS algorithm is the covered non-CDS node with the largest ID and has at least one uncovered neighbor node.
  - The **CDS-Select-node** for the MinV-CDS algorithm is the covered non-CDS node with the lowest velocity and has at least one uncovered neighbor node.

Note that the values for the density (the number of uncovered neighbors) and NSI (sum of the predicted LETs of the links with uncovered neighbors) for the nodes could change after the end of a particular iteration, depending on the updates to the **CDS-Nodes-List** and **Covered-Nodes-List**; this would lead to rearrangement of the entire **Priority-Queue** at the end of each iteration of the MaxD-CDS algorithm. On the other hand, the values for the node velocity and node ID remain the same for each of the iterations of the MinV-CDS and ID-CDS construction algorithms on a particular instance of the network graph.

### 2.4 Description of the Generic CDS Algorithm

The algorithm forms and outputs a CDS based on a given input graph representing a snapshot of the MANET at a particular time instant. Specifically, the algorithm outputs a list (**CDS-Node-List**) of all nodes that are part of the CDS formed based on the given MANET. The generic description and pseudo code given here (Figure 1) can be adapted for the particular CDS algorithm by appropriately incorporating the criteria for choosing the **CDS-Select-node** for each of the iterations as well as the criteria for maintaining the **Priority-Queue** during the different iterations of the CDS algorithm on a particular instance of the network graph.

The first node to be included in the **CDS-Node-List** is the **CDS-Select-node** ($s$) chosen according to the criteria discussed in Section 2.3. A CDS node is considered to be “covered”, so a CDS node is additionally added to the **Covered-Nodes-List** as it is added to the **CDS-Node-List**. All nodes that are adjacent to a CDS node are also said to be covered, so the uncovered neighbors of a CDS node are also added to the **Covered-Nodes-List** and to the **Priority-Queue**, as the node is added to the **CDS-Node-List**. The **Priority-Queue** for a particular CDS algorithm is maintained according to the criteria discussed in Section 2.2. The node at the front of the **Priority-Queue** is the next candidate node to become the **CDS-Select-node**, as long as the node satisfies the criteria mentioned in Section 2.3.

If the **Covered-Nodes-List** does not contain all the vertices of the input graph and the **Priority-Queue** is not empty, we dequeue the **Priority-Queue** to extract a **CDS-Select-node** $s$ that is not yet in the **CDS-Node-List**. All the uncovered neighbor nodes of node $s$ are added to the **Covered-Nodes-List** and to the **Priority-Queue**; node $s$ is also added to the **CDS-Node-List**. The above procedure is repeated until the **Covered-Nodes-List** contains all the vertices of the graph or the **Priority-Queue** becomes empty. If the
Priority-Queue becomes empty and the Covered-Nodes-List does not have at least one node that is present in the network graph, then the underlying network is considered to be disconnected.

The time complexity of the generic CDS algorithm can be given as: $\Theta((|E| + |V|) \times \log|V|)$ on a network graph of $|V|$ nodes and $|E|$ edges – each of the $|V|$ nodes and their associated edges have to be explored for inclusion in the CDS. If the Priority-Queue is implemented as a binary heap, it takes $\Theta(\log|V|)$ time to reorganize the heap after each dequeue and enqueue operation. There could be a total of $|V|$ dequeue operations, one for each of the iterations and a total of $|E|$ enqueue operations, one for every edge that is traversed across all the iterations.

**Input:** Snapshot of the Network Graph $G = (V, E)$, where $V$ is the set of vertices and $E$ is the set of edges

**CDS-Select-node** $s$ – vertex chosen according to the selection criteria in Section 2.3

**Auxiliary Variables and Functions:**
- CDS-Node-List, Covered-Nodes-List, Priority-Queue, Neighbors($v$) for every $v$ in $V$

**Output:** CDS-Node-List // contains the list of nodes part of the CDS.

**Initialization:**
- Covered-Nodes-List = $\{s\}$; CDS-Node-List = $\Phi$; Priority-Queue = $\{s\}$

**Begin** Construction of CDS

```plaintext
while ( |Covered-Nodes-List| < |V| and Priority-Queue $\neq \Phi$) do
  CDS-Select-node $s = \text{Dequeue}(\text{Priority-Queue})$
  // where $s \in \text{Covered-Nodes-List}$ and $s \notin \text{CDS-Node-List}$
  // CDS-Select-node $s$ is selected according to the criteria mentioned in Section 2.3.
  CDS-Node-List = CDS-Node-List U $\{s\}$

  for all $u \in \text{Neighbors}(s)$ and $u \notin \text{Covered-Nodes-List}$
    Covered-Nodes-List = Covered-Nodes-List U $\{u\}$
    Priority-Queue = Priority-Queue U $\{u\}$
  end for

  Rearrange the Priority-Queue according to the criteria for ordering the nodes (refer Section 2.2)
end while

if ( |Covered-Nodes-List| < |V| and Priority-Queue = $\Phi$) then
  return NULL // the network is disconnected and there is no CDS covering all the nodes
end if

return CDS-Node-List
```

**End** Construction of CDS

---

**Figure 1:** Generic Pseudo Code for CDS Construction

### 3 Algorithm to Validate the Existence of a CDS

To reduce the control overhead, we intend to use a CDS as long as it exists and opt for a new CDS only if the currently known CDS ceases to exist. To determine whether a currently known CDS exists at a
particular time instant, we construct a \( CDS-Edge-List \) comprising of edges that exist between any two nodes (according to the unit-disk graph model) in the \( CDS-Node-List \) known at that time instant. The two conditions that we test for are: (i) The CDS-induced graph comprising of the \( CDS-Node-List \) and the \( CDS-Edge-List \) should be connected and (ii) For every non-CDS node, there should be at least one CDS node as neighbor node. The pseudo code for the CDS validation algorithm is presented in Figure 2. Condition (i) is validated by running the classical Breadth First Search (BFS) algorithm [19] on the CDS-induced graph. Condition (ii) is validated only if condition (i) holds true. The algorithm returns true only if both the conditions hold true; otherwise, it returns false.

**Input:** \( CDS-Node-List \) // Set of vertices part of the CDS

**Auxiliary Variables and Functions:**

- \( CDS-Edge-List \) – Set of edges, \( \subseteq E \), between the vertices that are part of \( CDS-Node-List \)
- \( connectedCDS \) – Boolean variable that stores information whether the CDS-induced sub graph of \( G \) (comprising of the \( CDS-Node-List \) and \( CDS-Edge-List \) are connected)

**Output:** true or false

- // true, if the nodes in \( CDS-Node-List \) form a connected sub graph of \( G \) and every vertex \( v \notin CDS-Node-List \) is at least a neighbor of a vertex \( u \in CDS-Node-List \)
- // false, if the nodes in \( CDS-Node-List \) do not form a connected sub graph of \( G \) and/or there exists at least one vertex \( v \notin CDS-Node-List \) that has no neighbor in \( CDS-Node-List \)

**Initialization:**

- \( CDS-Edge-List = \Phi \)

**Begin CDS-Validation**

\[
\text{for every pair of vertices } u, v \in CDS-Node-List \text{ do}
\]

\[
\text{if there exists an edge } (u, v) \in E \text{ at time instant } t \text{ then}
\]

\[
CDS-Edge-List = CDS-Edge-List \cup \{(u, v)\}
\]

\[
\text{end if}
\]

\[
\text{end for}
\]

\[
connectedCDS = \text{Breadth-First-Search}(CDS-Node-List, CDS-Edge-List)
\]

\[
\text{if } connectedCDS = \text{true then}
\]

\[
\text{for every vertex } v \notin CDS-Node-List \text{ do}
\]

\[
\text{if there exists no edge } (u, v) \in E \text{ where } u \in CDS-Node-List \text{ at time instant } t \text{ then}
\]

\[
\text{return false}
\]

\[
\text{end if}
\]

\[
\text{end for}
\]

\[
\text{return true}
\]

\[
\text{end if}
\]

\[
\text{return false} // \text{if } connectedCDS = \text{false}
\]

**End CDS-Validation**

**Figure 2:** Pseudo Code for the CDS Validation Algorithm

4 Examples to Illustrate the Working of the ID-CDS and MaxD-CDS Algorithms

We demonstrate the construction of the ID-CDS and MaxD-CDS through examples illustrated in Figures 4 and 5 respectively. As indicated in the legend (Figure 3) for these two figures wherein each circle
indicates a node in the network graph, we represent: (i) a CDS node with a thick black-bordered circle that is gray-shaded inside; (ii) a covered node (a non-CDS node that is a neighbor of at least one CDS node) with a thick gray-bordered circle that has black-dots inside; (iii) a non-covered node (a node that is neither a CDS node nor a covered node) with a black thin circle with no shades or dots inside. In Figure 4 (illustrating the construction of the ID-CDS), the integer values inside each circle represents the node ID; in Figure 5 (illustrating the construction of the MaxD-CDS), the integer values outside of each circle represents the number of uncovered neighbors for the corresponding node. Both Figures 4 and 5 comprise of 16 nodes. The ID-CDS is constructed in 7 iterations and consists of 7 nodes and 8 edges; the MaxD-CDS is constructed in 5 iterations and also consists of 5 nodes and 5 edges.

Figure 3: Legend for Figures 4 and 5

Figure 4: Example to Illustrate the Construction of an ID-CDS

Figure 5: Example to Illustrate the Construction of a MaxD-CDS
5 Example to Illustrate the Working of the Minimum Velocity-based CDS (MinV-CDS) Algorithm

We demonstrate the construction of the MinV-CDS through an example of a 24-node network illustrated in Figure 6 wherein each circle indicates a node. We represent: (i) a CDS node with a bold circle; (ii) a covered node (a non-CDS node that is a neighbor of at least one CDS node) with a shaded circle; (iii) a non-covered node (a node that is neither a CDS node nor a covered node) with a black thin circle that is neither shaded nor made bold. The integer values outside each circle represent the node ID; the integer values inside of each circle represent the number of uncovered neighbors for the corresponding node; the real-number inside the circle represents the node velocity (in m/s). As observed in Figure 6, the MinV-CDS is constructed in 13 iterations and consists of 14 nodes and 16 edges.
6 Description of the Mobility Models

In this section, we provide a brief overview of the Random Waypoint mobility model commonly used in MANET simulation studies and the widely used City Section and Manhattan mobility models for VANET simulation studies. All the three mobility models [14] assume the network is confined within fixed boundary conditions. The Random Waypoint mobility model assumes that the nodes can move anywhere within a network region. The City Section and the Manhattan mobility models assume the network to be divided into grids: square blocks of identical block length. The network is thus basically composed of a number of horizontal and vertical streets. Each street has two lanes, one for each direction (north and south direction for vertical streets, east and west direction for horizontal streets). A node is allowed to move only along the grids of horizontal and vertical streets.

6.1 Random Waypoint Mobility Model

Initially, the nodes are assumed to be placed at random locations in the network. The movement of each node is independent of the other nodes in the network. The mobility of a particular node is described as follows: The node chooses a random target location to move. The velocity with which the node moves to this chosen location is uniformly randomly selected from the interval \([v_{\text{min}}, v_{\text{max}}]\). The node moves in a straight line (in a particular direction) to the chosen location with the chosen velocity. After reaching the target location, the node may stop there for a certain time called the pause time. The node then continues to choose another target location and moves to that location with a new velocity chosen again from the interval \([v_{\text{min}}, v_{\text{max}}]\). The selection of each target location and a velocity to move to that location is independent of the current node location and the velocity with which the node reached that location. In Figure 7, we observe that nodes A and B move independent of each other, in random directions with randomly chosen velocities.

6.2 City Section Mobility Model

Initially, the nodes are assumed to be randomly placed in the street intersections. Each street (i.e., one side of a square block) is assumed to have a particular speed limit. Based on this speed limit and the block length, one can determine the time it would take move in the street. Each node placed at a particular street intersection chooses a random target street intersection to move. The node then moves to the chosen street intersection on a path that will incur the least amount of travel time. If two or more paths incur the least amount of travel time, the tie is broken arbitrarily. After reaching the targeted street intersection, the node may stay there for a pause time and then again choose a random target street intersection to move. The
node then moves towards the new chosen street intersection on the path that will incur the least amount of travel time. This procedure is repeated independently by each node. In Figure 8, the movement of two nodes A and B according to the City Section mobility model has been illustrated.

6.3 Manhattan Mobility Model

Initially, the nodes are assumed to be randomly placed in the street intersections. The movement of a node is decided one street at a time. To start with, each node has equal chance (i.e., probability) of choosing any of the streets leading from its initial location. In Figure 9, to start with, node A has 25% chance to move in each of the four possible directions (east, west, north or south), where as node B can move only either to the west, east or south with a 1/3 chance for each direction. After a node begins to move in the chosen direction and reaches the next street intersection, the subsequent street in which the node will move is chosen probabilistically. If a node can continue to move in the same direction or can also change directions, then the node has 50% chance of continuing in the same direction, 25% chance of turning to the east/north and 25% chance of turning to the west/south, depending on the direction of the previous movement. If a node has only two options, then the node has an equal (50%) chance of exploring either of the two options. For example, in Figure 9, once node A reaches the rightmost boundary of the network, the node can either move to the north or to the south, each with a probability of 0.5 and the node chooses the north direction. After moving to the street intersection in the north, node A can either continue to move northwards or turn left and move eastwards, each with a probability of 0.5. If a node has only one option to move (this occurs when the node reaches any of the four corners of the network), then the node has no other choice except to explore that option. For example, in Figure 9, we observe node B that was traveling westward, reaches the street intersection, which is the corner of the network. The only option for node B is then to turn to the left and proceed southwards.

7 Simulations

The network dimensions are 1000m x 1000m. The Random Waypoint (RWP) model works by assuming an open network field without any grid constraints. For the City Section and Manhattan mobility models, the network is divided into grids: square blocks of length (side) 100m. The network for the two VANET mobility models is thus basically composed of a number of horizontal and vertical streets. Each street has two lanes, one for each direction (north and south direction for vertical streets, east and west direction for horizontal streets). A node is allowed to move only along the grids of horizontal and vertical streets. Simulations are conducted for grid block lengths (BL) of 50m, 100m and 200m. The wireless transmission range of a node is 250m. The network density is varied by performing the simulations with 50 (low-moderate) and 100 (moderate-high) nodes. The maximum node velocity values used for each of the three mobility models are 5 m/s (about 10 miles per hour), 25 m/s (about 50 miles per hour) and 50 m/s (about 100 miles per hour), representing levels of low, moderate and high node mobility respectively. The pause time is 0 seconds; so, all the nodes are constantly in motion.

A centralized view of the network topology is obtained by generating mobility trace files for 1000 seconds under each of the three mobility models. The network topology is sampled for every 0.25 seconds to generate the graphs. Two nodes a and b are assumed to have a bi-directional link at time t, if the Euclidean distance between them at time t (derived using the locations of the nodes from the mobility trace file) is less than or equal to the wireless transmission range of the nodes. Each data point in the performance figures shown in this section is an average computed over 5 mobility trace files and 20 randomly selected s-d pairs from each of the mobility trace files. The starting time of each s-d session is uniformly distributed between 1 to 20 seconds.

Overall, the simulations are conducted for six different combinations of network density and node mobility:
(1) **Low-moderate density and low node mobility**: represented by network of 50 nodes and $v_{\text{max}} = 5$ m/s per node, corresponding to an average neighborhood size of 10 nodes and average node velocity of about 2.5 m/s

(2) **Moderate-high density and low node mobility**: represented by network of 100 nodes and $v_{\text{max}} = 5$ m/s per node, corresponding to an average neighborhood size of 20 nodes and average node velocity of about 2.5 m/s

(3) **Low-moderate density and moderate node mobility**: represented by network of 50 nodes and $v_{\text{max}} = 25$ m/s per node, corresponding to an average neighborhood size of 10 nodes and average node velocity of about 12.5 m/s

(4) **Moderate-high density and moderate node mobility**: represented by network of 100 nodes and $v_{\text{max}} = 25$ m/s per node, corresponding to an average neighborhood size of 20 nodes and average node velocity of about 12.5 m/s

(5) **Low-moderate density and high node mobility**: represented by network of 50 nodes and $v_{\text{max}} = 50$ m/s per node, corresponding to an average neighborhood size of 10 nodes and average node velocity of about 25 m/s

(6) **Moderate-high density and high node mobility**: represented by network of 100 nodes and $v_{\text{max}} = 50$ m/s per node, corresponding to an average neighborhood size of 20 nodes and average node velocity of about 25 m/s

### 7.1 Randomization of the Node IDs for the ID-CDS Algorithm

In the simulations, to be fair to all nodes in the network, the node IDs are randomized before every reconfiguration of the ID-CDS. In other words, before the ID-CDS algorithm is reinitiated after the failure of the currently known ID-CDS, the node IDs are randomly redistributed and the ID-CDS algorithm is run based on the network graph formed based on the new node IDs. In each such reconfiguration, the appropriate mapping between the new randomized node IDs and the original node IDs (that will never change throughout the simulation) is maintained. While the original node IDs are used to determine the location and mobility of the nodes (from the mobility profile) at any time instant; the randomized node IDs are used to decide the inclusion of nodes from the Covered-Nodes-List to the CDS-Node-List as part of the ID-CDS algorithm. Note that the neighbors of a node that is recently added to the CDS-Node-List are considered covered based on their original node IDs; however they are listed in the Covered-Nodes-List based on their randomized node IDs.

### 7.2 Overall Simulation Methodology

The overall simulation methodology is as follows: For each mobility model, snapshots (static graphs) of the network topology are constructed for every 0.25 seconds, starting from time 0 to the simulation time of 1000 seconds. If a CDS is not known at a particular time instant, the appropriate CDS construction algorithm is run on the network snapshot. The CDS determined at a particular time instant is used in the subsequent time instants until the CDS ceases to exist. For a CDS to be considered to exist at a particular time instant, two conditions have to hold good: (i) All the CDS nodes have to stay connected – i.e. reachable from one another directly or through multi-hop paths; and (ii) Every non-CDS node should have at least one CDS node as its neighbor. If a CDS ceases to exist at a particular time instant, the appropriate CDS construction algorithm is again run and the new CDS is continued to be used as explained above. This procedure is continued for the duration of the simulation time.

### 7.3 Performance Metrics

The following two performance metrics are measured under the three mobility models:

- **Effective CDS Lifetime**: The duration of time each instance of a CDS actually exists is measured and the average of the CDS lifetimes observed across the entire simulation time period (for all the
mobility profiles representing a particular combination of network density and node mobility) is computed. These average values of the actually observed CDS lifetimes are referred to as the Absolute CDS Lifetime. However, since the connectivity of the different CDSs are less than 1.0 for most of the scenarios evaluated, a better representative metric called the Effective CDS Lifetime is introduced – defined as the product of the Absolute CDS Lifetime and the CDS Connectivity – to effectively capture the stability of a CDS taking into consideration the chances of determining the CDS. For example, if a CDS has an average absolute lifetime of 10 seconds; but its connectivity is only 0.8 – it is more prudent to use the effective lifetime of 10*0.8 = 8 seconds as a more realistic measure of the stability (lifetime) of the CDS. The effective CDS lifetime incurred under the three mobility models under diverse conditions of network density and node mobility is illustrated in Figures 10, 12, 14, 16, 18 and 20.

- **CDS Node Size**: This is a time-averaged value for the number of nodes that are part of the CDS used for every time instant over the entire simulation. For example, if there exists a sequence of three CDS of size 30 nodes, 40 nodes and 20 nodes in a network for 6, 10 and 4 seconds respectively, then the average CDS Node Size for a total of 20 seconds is \((30*6 + 40*10 + 20*4)/(6 + 10 + 4) = 33.0\) and not simply the average of 30, 40 and 20 nodes = 30. The CDS Node Size incurred under the three mobility models under diverse conditions of network density and node mobility is illustrated in Figures 11, 13, 15, 17, 19 and 21.

### 7.4 Effective MaxD-CDS Lifetime

The MaxD-CDS incurs the largest lifetime under the Random Waypoint model for 5 of the 6 simulation conditions; for the condition of low-moderate density and moderate node mobility, the MaxD-CDS incurs the largest lifetime under the City Section model operated with a block length of 50m. The ratio of the difference between the maximum and minimum values of the effective MaxD-CDS lifetime incurred under the different simulation conditions ranges from 1.4 to 2.2. Between the two VANET mobility models, the MaxD-CDS is observed to incur a larger lifetime under the Manhattan mobility model for all the 6 simulation conditions. The effective MaxD-CDS lifetime incurred under the Manhattan model could be as large as 60% more than that incurred under the City Section model. For a given simulation condition, it is observed that the difference in the effective MaxD-CDS lifetime incurred with the two VANET mobility models decreases with increase in the block length (from 50m to 200m). It is also observed that for a given block length and node mobility, for most of the simulation conditions, the absolute values for the effective MaxD-CDS lifetime decreases with increase in network density; on the other hand, the relative difference in the effective MaxD-CDS lifetime incurred between the Manhattan model and the City Section model increases with increase in network density. For both the VANET mobility models, it is observed that the effective MaxD-CDS lifetime decreases with increase in the block length from 50m to 200m. This could be attributed to the nature of the MaxD-CDS algorithm to include only as minimal nodes as possible in the CDS; with a larger block length, even though the number of nodes constituting the MaxD-CDS reduces, the physical distance between the edges that are part of the CDS edges increases. As a result, a MaxD-CDS is more vulnerable to failure when operated in a VANET grid of larger block length.

### 7.5 MaxD-CDS Node Size

The MaxD-CDS incurs the lowest value for the node size under the Random Waypoint model for 5 of the 6 simulation conditions. For the condition of low-moderate density and low node mobility, the MaxD-CDS incurs a lower node size under the City Section model (block length of 50m). Thus, it is possible to simultaneously maximize the MaxD-CDS lifetime as well as minimize the node size when operated under the Random Waypoint model; there is no significant tradeoff.

City Section vs. Random Waypoint
Manhattan vs. Random Waypoint

Figure 10: Effective CDS Lifetime [50 nodes, \(v_{\text{max}} = 5 \text{ m/s}\)]

City Section vs. Random Waypoint
Manhattan vs. Random Waypoint

Figure 11: CDS Node Size [50 nodes, \(v_{\text{max}} = 5 \text{ m/s}\)]

City Section vs. Random Waypoint
Manhattan vs. Random Waypoint

Figure 12: Effective CDS Lifetime [100 nodes, \(v_{\text{max}} = 5 \text{ m/s}\)]

City Section vs. Random Waypoint
Manhattan vs. Random Waypoint

Figure 13: CDS Node Size [100 nodes, \(v_{\text{max}} = 5 \text{ m/s}\)]

**Figure 14:** Effective CDS Lifetime [50 nodes, $v_{max} = 25$ m/s]

**Figure 15:** CDS Node Size [50 nodes, $v_{max} = 25$ m/s]

**Figure 16:** Effective CDS Lifetime [100 nodes, $v_{max} = 25$ m/s]

**Figure 17:** CDS Node Size [100 nodes, $v_{max} = 25$ m/s]

**Figure 18:** Effective CDS Lifetime [50 nodes, $v_{\text{max}} = 50$ m/s]

**Figure 19:** CDS Node Size [50 nodes, $v_{\text{max}} = 50$ m/s]

**Figure 20:** Effective CDS Lifetime [100 nodes, $v_{\text{max}} = 50$ m/s]

**Figure 21:** CDS Node Size [100 nodes, $v_{\text{max}} = 50$ m/s]
The ratio of the maximum MaxD-CDS node size (incurred mostly with the Manhattan model) and the minimum MaxD-CDS node size (incurred mostly with the Random Waypoint model) typically ranges from 1.2 to 1.5. Between the two VANET mobility models, the MaxD-CDS is observed to sustain a lower node size for the City Section model. This indicates a CDS lifetime-node size tradeoff between the two VANET mobility models. In Section 4.4, it is observed that under a given simulation condition, the MaxD-CDS incurs a larger effective CDS lifetime under the Manhattan model; but, this is accompanied by a larger CDS node size. The relative difference in the MaxD-CDS node size between the Manhattan and City Section models could be as large as 25%, and the relative difference decreases with increase in the grid block length from 50m to 200m.

For the Random Waypoint model, as the network density is doubled from 50 to 100 nodes for a given mobility level, the MaxD-CDS node size decreases by about 10-25%. This decrease, rather than an increase that is typically expected, could be attributed to the nature of the MaxD-CDS algorithm to include only those nodes as part of the CDS so that as many non-CDS nodes could be covered as possible. The negative impact of this greedy approach is observed in the effective CDS lifetime that decreases with increase in network density, due to the need to cover an increased number of non-CDS nodes with a reduced number of CDS nodes at high network density. Under the Manhattan mobility model, there is no appreciable change in the MaxD-CDS node size as the network density is doubled. For the City Section mobility model, for a given block length and node mobility, as the network density is doubled from 50 to 100 nodes, the MaxD-CDS node size increases minimally, only at most by 10%.

7.6 Effective ID-CDS Lifetime

The ID-CDS incurs a larger effective lifetime under the two VANET mobility models compared to the Random Waypoint model under all the 6 simulation conditions. This is a trend different from what was observed in Section 4.4 for the MaxD-CDS which incurred the largest effective lifetime under the Random Waypoint model. The effective ID-CDS lifetime was the largest under the Manhattan model (with a block length of 50m) for 4 of the 6 mobility conditions. For conditions corresponding to low-moderate network density coupled with moderate and high node mobility conditions, the ID-CDS incurred the largest effective lifetime under the City Section model (for block lengths of 50m and 200m in moderate and high node mobility conditions respectively). For any given block length, the difference in the effective ID-CDS lifetime under the two VANET mobility models ranges from 15 to 90%, with the typical difference ranging from 20 to 40%. For any level of node mobility and block length, the ratio of the difference between the maximum effective CDS lifetime ranges from 2 to 3 in networks of low-moderate density and ranges from 1.4 to 1.8 in networks of moderate-high density. Under networks of low-moderate density, the effective ID-CDS lifetime incurred with the Random Waypoint model could be significantly lower (by about 50-60%) compared to the maximum effective ID-CDS lifetime incurred (with either of the two VANET mobility models). However under networks of moderate-high density, the effective ID-CDS lifetime incurred with the Random Waypoint model is only marginally lower (by about 5-20%) compared to the maximum effective ID-CDS lifetime incurred with the VANET mobility models. Overall, it is also observed that for the two VANET models, the effective ID-CDS lifetime decreases with increase in the grid block length from 50m to 200m.

7.7 ID-CDS Node Size

The ID-CDS exhibits a significant lifetime-node size tradeoff. The ID-CDS node size is the lowest under the Random Waypoint model and is the largest under the Manhattan model that also yielded the largest effective lifetime. Thus, it is not possible to simultaneously maximize the effective ID-CDS lifetime by incurring a lower node size under any mobility model. The ratio of the maximum ID-CDS node size to that of the minimum node size ranges from 1.2 to 1.8; the difference increases with increase in the network density and/or node mobility. Between the two VANET mobility models, the ID-CDS node size is larger when run the Manhattan mobility model compared to the City Section model. For a given block
length and level of node mobility, the difference in the ID-CDS node size between the two mobility models is larger (by about 20-40%) when operated in networks of low-moderate density. In networks of moderate-high density, the difference in the ID-CDS node size incurred under the two VANET mobility models is only at most 15%. Another interesting observation is that the grid block length does not significantly impact the ID-CDS node size under both the VANET mobility models.

7.8 Effective MinV-CDS Lifetime

The minimum velocity-based CDS (MinV-CDS) incurs the largest effective lifetime under the Random Waypoint model for 5 of the 6 simulation conditions. For the simulation condition of low-moderate network density and low node mobility, the effective MinV-CDS lifetime is slightly larger (by about 9%) under the Manhattan model (block length of 50m) compared to the Random Waypoint model. The ratio of the maximum effective MinV-CDS lifetime (incurred typically under the Random Waypoint model) and the minimum effective MinV-CDS lifetime (incurred typically under the City Section model for low-moderate density scenarios and under the Manhattan model for moderate-high density scenarios) is about 1.6 to 2.8; the difference in the effective MinV-CDS lifetime between the Random Waypoint model and that of the two VANET mobility models decreases with increase in the network density (for a fixed node mobility) and increases with increase in node mobility (for a fixed network density). Between the two VANET mobility models, it is observed that the effective MinV-CDS lifetime incurred under the Manhattan model is greater (by about 20%) than that incurred under the City Section model in low-moderate network density and low-moderate node mobility scenarios. With increase in the network density and/or node mobility, the effective MinV-CDS lifetime incurred under the City Section model is larger (at most by about 25%) compared to that incurred under the Manhattan model. The grid block length significantly influences the effective MinV-CDS lifetime under both the VANET mobility models. The effective MinV-CDS lifetime incurred under a grid block length of 50m could be as large as twice the grid block lengths of 100m and 200m. This could be attributed to the presence of physically longer edges in the MinV-CDS determined under larger values of the grid block lengths.

7.9 MinV-CDS Node Size

The MinV-CDS node size is the lowest under the Random Waypoint model for all the simulation conditions. Thus, one could obtain a larger MinV-CDS lifetime and a lower MinV-CDS node size when operated under the Random Waypoint model. The difference in the MinV-CDS node size incurred with the Random Waypoint model and the two VANET mobility models increases predominantly with increase in the network density. The difference could range from 25% to 100%, with the difference increasing with increase in the grid block length. For a given network density, there is no significant impact of the node mobility levels on the MinV-CDS node size. Between the two VANET mobility models, it is observed that the MinV-CDS node size is larger when run under the Manhattan mobility model for all the 6 simulation conditions, with the difference increasing with increase in network density and/or node mobility. For a given condition of network density and node mobility, the difference in the MinV-CDS node size decreases with increase in the grid block length from 50m to 200m. Even though the effective MinV-CDS lifetime is not always the greater under the Manhattan model, it is interesting to observe the MinV-CDS node size to be always larger than that incurred with the City Section model. The MinV-CDS node size incurred under the Manhattan model could be relatively (compared to the City Section model) as large as 30% and 80% respectively under the low-moderate and moderate-high network density scenarios. Thus, between the two VANET mobility models, it is possible to simultaneously optimize both the MinV-CDS lifetime and the node size when operated under the City Section model vis-à-vis the Manhattan model.
8 Conclusions

The following significant conclusions can be drawn from this paper: (1) The MinV-CDS incurs the largest effective lifetime under all the three mobility models; and at the same time incurs the largest node size. Even though this indicates a CDS lifetime-node size tradeoff, it is interesting to observe that the effective MinV-CDS lifetime can be as large as 4 times the MaxD-CDS lifetime with just double the CDS node size. Similarly, the effective MinV-CDS lifetime could be double that of the ID-CDS lifetime with just at most a 33% increase in the CDS node size. (2) The MaxD-CDS and MinV-CDS incur the largest effective lifetime under the Random Waypoint model and also can sustain the lowest node size under this mobility model vis-à-vis the two VANET mobility models. On the other hand, the ID-CDS incurs the largest effective lifetime under the VANET mobility models (predominantly with the Manhattan model) and the lowest node size with the Random Waypoint model. Thus, there is a CDS lifetime-node size tradeoff with respect to using an appropriate mobility model for the ID-CDS. (3) With regards to the two VANET mobility models, the MaxD-CDS and ID-CDS yielded the largest effective lifetime under the Manhattan model for most of the simulation conditions; whereas the MinV-CDS yielded the largest effective lifetime under the City Section model for a majority of the simulation conditions. On the other hand, all the three connected dominating sets sustained a larger CDS node size under the Manhattan mobility model compared to the City Section model. This illustrates a CDS lifetime – node size tradeoff for the MaxD-CDS and ID-CDS with respect to the choice of usage between the two VANET mobility models. (4) With regards to the impact of the grid block length on the stability of the connected dominating sets, one can conclude that the lower grid block lengths yielded more stable connected dominating sets compared to larger grid block lengths. Also, there is no appreciable increase in the CDS node size when operated at lower grid block lengths for all the three CDS algorithms under both the VANET mobility models.

9 Acknowledgments

The work leading to this paper was partly funded through the U.S. National Science Foundation (NSF) grants DUE-0941959 and CNS-0851646. The views and conclusions contained in this paper are those of the authors and do not represent the official policies, either expressed or implied, of the funding agency.

10 References


