Message-Efficient CDS Construction in MANETs

Kazuya Sakai*, Min-Te Sun†, and Wei-Shinn Ku*
*Department of Computer Science and Software Engineering, Auburn University, Auburn, Alabama 36849–5347
†Department of Computer Science and Information Engineering, National Central University, Taoyuan 320, Taiwan

***Abstract*—The connected dominating set (CDS) has been extensively used for routing and broadcast in mobile ad hoc networks (MANETs). Due to the nature of MANETs, it is preferred that the CDS protocol not only creates a CDS with small size, but also incurs less communication and computational overhead, adapts to the nodal mobility, and generates CDS as quickly as possible. While the CDS protocols developed in the past create small size of CDS, they tend to incur too much communication overhead. In this paper, we propose the Message-Efficient Dominator Tree Connection algorithm (ME-DTC), which significantly reduces the number of control messages during the tree connection phase in Tree-based CDS protocols. By incorporating this algorithm with the Multi-Initiator CDS protocol, we have a CDS protocol which meets all the desirable features in MANETs. The simulation as well as analytical results validate that our proposed algorithm achieves its design goals.**

I. Introduction

Mobile ad hoc networks (MANETs) are well-suited for applications where a fixed infrastructure is not readily available, such as communications in the battlefield and rescue missions in case of a catastrophe [1]. In MANET research, the connected dominating set (CDS) is commonly used as a virtual backbone for protocols to provide different communication primitives, such as routing [2] and multicast/broadcast [3]. CDS is defined as a subset of nodes in a network such that each node in the network is either in the set or a neighbor of some node in the set, and the induced graph of the nodes in the set is connected.

To reduce the communication and storage overhead of the CDS applications in MANETs, the size of the CDS should be kept as small as possible. However, it is known that finding the minimum CDS for a general graph is NP-hard [4]. When it comes to MANETs, the issue is more challenging as practically only localized information is available at each node. A number of distributed protocols [5]–[11] have been developed that emphasize on constructing a small CDS with localized information. Considering several factors including the CDS size, the communication overhead, and the mobility handling capability, the family of the tree-based CDS protocols [10], [11] seem to be the best choice in MANETs. In these protocols, a number of initiators are elected and each initiator creates its dominator tree. Then, additional nodes are added to connect disjoint trees to form a CDS. However, to connect trees, each initiator needs to collect the neighboring tree information from all its leaf nodes, so a large number of control messages are introduced. In this paper, we propose a novel dominator tree connection scheme, namely Message-Efficient Dominator Tree Connection (ME-DTC) algorithm, for the tree-based CDS protocols. Unlike the original tree connection mechanism, a node in the boundary area where two trees meet makes the decision to connect its neighboring tree independently and autonomously. ME-DTC significantly reduces the number of control messages while keeping the CDS size competitive. Both simulation and analytical results have confirmed that the Multi-Initiator CDS protocol [10] incorporating ME-DTC achieves all the desirable properties for CDS construction in MANETs.

The rest of this paper is organized as follows. The existing distributed CDS protocols are surveyed in Section II. The ME-DTC algorithm is presented in Section III. The simulation results are demonstrated and analyzed in Section IV. In Section V, the analytical model is introduced to estimate the average performance of the proposed CDS protocol. The conclusion is provided in Section VI.

II. Literature Reviews

Calculating the minimum CDS is known to be NP-hard [4]. Although there exist several protocols to construct a CDS with the global topology information [12], such an assumption is not practical when it comes to MANETs, where only localized information is available at each node. As a result, a number of distributed CDS protocols [5]–[11] have been proposed to approximate a small size of CDS. These protocols can be classified into three categories: Pruning-based, MIS-based, and Tree-based.

- **Pruning-based -** The pruning-based CDS protocols, such as Dai's [5] and Stojmenovic's [6], obtain CDS by eliminating unnecessary nodes from the network. Although these protocols are successful in constructing CDS in an arbitrary graph, they tend to incur many control messages as each node needs to constantly exchange messages with its neighbors to obtain two-hop neighboring information.
- **MIS-based -** The MIS-based CDS protocols, such as Wan's [7], [8], obtain CDS by expanding the maximal independent set (hence the name MIS) under the assumption of the unit-disk link model. These protocols consist of two phases. In the first phase, a MIS is distributively computed. In the second phase, additional nodes are added to connect nodes in MIS. Since a spanning tree is adopted in these protocols to construct the CDS, a number of messages are transmitted across nodes in the tree several times.
- **Tree-based -** The tree-based protocols, such as those in [9]–[11], consist of three phases. In the first phase, a set of nodes, called initiators, are elected. In the second

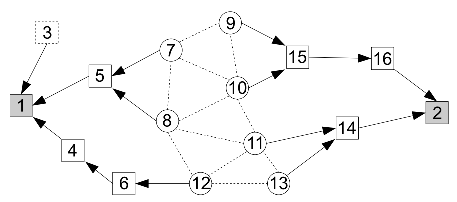

Fig. 1. Topology I.

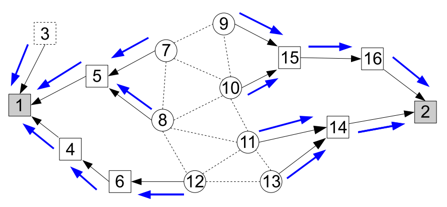

Fig. 2. Tree connection by MI protocol.

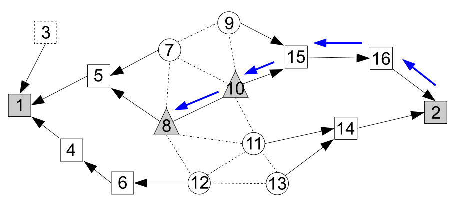

Fig. 3. Tree connection by MI (cont.).

phase, a dominator tree is generated independently from each initiator. In the third phase, additional nodes are added to connect disjoint dominator trees. Note that if only one initiator is elected in the first phase, there will be only one dominator tree and therefore no tree connection phase is required [9]. While the Single-Initiator CDS protocol [9] tends to create the smallest CDS and introduce less communication overhead, it incurs the single point of failure at the sole initiator. To tackle this issue, the Multiple-Initiator (MI) CDS protocol [10] was proposed. Fast Convergence Dominator Tree Construction (FC-DTC) [11] scheme further improves the performance of tree-based CDS protocols in terms of the convergence time.

While all the aforementioned protocols create CDS with localized information, it has been shown in [10], [11] that the MIS-based and tree-based protocols generally produce smaller CDS than the pruning-based ones and the tree-based protocols incur less communication overhead. Furthermore, the only CDS protocols capable of maintaining CDS under changes of network topology are Dai's [5], Stojmenovic's [6], and the tree-based protocols [9]–[11]. The other protocols [7], [8] have to construct the CDS from scratch when the CDS is corrupted due to nodal mobility. Based on these considerations, MI [10] combined with FC-DTC [11] seems to be a better choice for MANETs.

III. Message-Efficient CDS Construction

A. Motivation

As stated in Section II, tree-based protocols, such as MI [10] with FC-DTC [11], are more suitable for MANETs since they are able to keep a competitive size of CDS and deal with node mobility without the need of reconstructing the whole CDS from scratch. There are three phases in MI CDS construction: i) Initiator Election: Nodes with local minimum id (i.e., MAC address) among nodes within α hops are selected as initiators. ii) Tree Construction: A dominator tree is grown from each initiator by means of timers and localized information. iii) Tree Connection: The initiator determines what nodes to use to connect the neighboring trees. FC-DTC modifies the original timer-based tree construction in MI so that a node determines its own state after exactly 2 beacon periods, which greatly reduces the convergence time of dominator tree construction. However, the tree connection phase in MI [10] still introduces too many control messages, as each initiator needs to collect the information of neighboring trees from all its leaf nodes. According to [10], the number of control messages is estimated to be $0.41\Delta\cdot\frac{n_{init}-1}{n_{init}}\cdot n$, where Δ is the average number of neighbors, n is the number of nodes in the network, and n_{init} is the number of initiators. Thus, the amount of control overhead increases in proportion to the network size.

For example, Figure 1 shows a snapshot of an ad hoc network after dominator trees are generated. A gray square represents an initiator, a square represents a dominator, a dashed square represents a dominatee, a circle represents a node at the boundary area where two trees meet. An arrow shows the relationship between a parent and its child in a tree (e.g., node 1 is the parent of node 5). In Figure 2, each initiator collects the information of neighboring trees from leaf nodes. Then, in Figure 3 the initiator with larger id (node 2) selects node 10, which is represented by a gray triangle, to connect two trees. In the end, node 8 belonging to initiator 1 and node 10 belonging to initiator 2 are added to the dominating set. As can be seen in Figure 2 and 3, 18 control messages are required. To avoid introducing these many control messages, additional nodes should be elected for tree connection locally and independently from their initiators.

In the following sections, we elaborate the Message-Efficient Dominator Tree Connection (ME-DTC) algorithm for tree-based CDS protocols. By using ME-DTC instead of the original tree connection mechanism, the performance of MI [10] protocol can be improved significantly in terms of control overhead.

B. Definitions and Notations

A node is said to be a *border node* if it has at least one neighbor which belongs to a different tree. For example, in Figure 1, node 7-13 are border nodes. The node added in the tree connection phase is said to be the *bridge node*. For instance, in Figure 3 node 8 and 10 are bridge nodes. A node maintains its border state (denoted as $bstate$), which is one of $\{nonborder, unconnected, bridge, connected\}$. Nodes which are not border nodes are always in $nonborder\ bstate$. A border node is initially in $unconnected\ bstate$. At the end of the proposed ME-DTC algorithm, a border node ends up in either $bridge$ or $connected\ bstate$. In addition, each node maintains local variables S_{nt}, the set of initiators of its neighboring trees, and S_{ct}, the set of initiators of neighboring trees which it connects directly or through several dominators.

C. Message-Efficient Dominator Tree Connection

While the proposed Message-Efficient Dominator Tree Connection algorithm works on the tree connection phase of any tree-based CDS protocol, for ease of presentation we confine

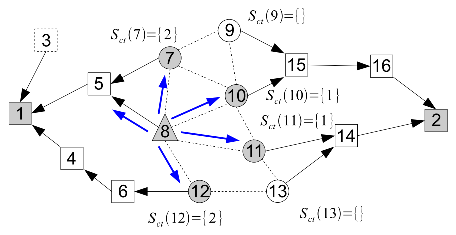

Fig. 4. Example of ME-DTC on topology I.

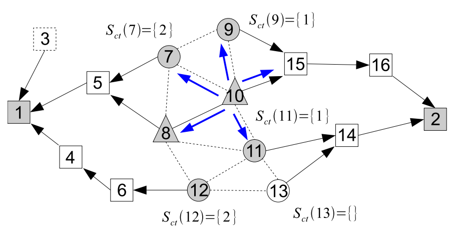

Fig. 5. Example of ME-DTC (cont.).

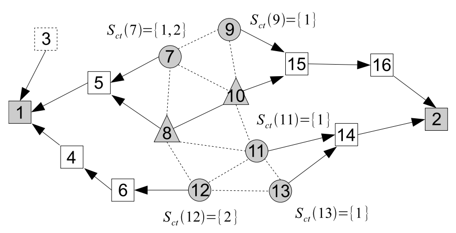

Fig. 6. Example of ME-DTC (cont.2).

TABLE I
DEFINITION OF NOTATIONS

Symbol	Definition
$N(i)$	The open neighbor set of node i
$id(i)$	The identifer of node i
$initiator(i)$	The initiator of node i
$bstate(i)$	The border state of node i
$dominator(i)$	The dominator id of node i
$n_{uc}(i)$	The number of unconnected border neighbors of node i
$N_b(i)$	The border neighbor set of node i
$S_{nt}(i)$	The set of id of neighboring tree of node i
$S_{ct}(i)$	The set of id of neighboring tree which connects node i
$BTimer(i)$	A clock down timer of node i
BT_{max}	The initial value of $BTimer$
α	The number of hop to elect the local minimum

the discussion of our algorithm on MI [10] and FC-DTC [11]. In other words, we borrow the initiator election phase from MI and and the dominator tree construction phase from FC-DTC.

Similar to IEEE 802.11 [13], we assume that each node broadcasts the beacon periodically with its unique id (i.e., MAC address). The original MI extends the beacon frame to include initiator id, $state$, $color$, its own id, its dominator id, and the number of uncovered neighbors. The $state$ of a node is either $dominator$ or $dominatee$. $color$ is used in the mobility handling process to track the heartbeat of an initiator. The proposed ME-DTC further extends the beacon frame by adding the border state, $bstate$, and the number of unconnected border nodes, n_{uc}.

After the dominator tree construction completes, a border node knows its neighboring trees $\{\forall j \in N_b(i), initiator(j)\}$. Once a border node detects that it has the most unconnected nodes among its border neighbors, it starts the border timer, denoted as $BTimer$, with the initial value of BT_{max}. If two border neighbors have the same timer value, the tie can be broken by id. When $BTimer$ expires, if a border node still has the most unconnected border neighbors, it switches its $bstate$ to $bridge$ and selects a border node with the highest n_{uc} as the bridge to connect trees for each neighboring tree. To notify them, it broadcasts a $BridgeRequest$ message, which includes its id, its initiator id, the selected node id, and initiator id of the neighboring tree. On receiving $BridgeRequest$, the elected border node switches to $bridge$ and broadcasts $BridgeReply$, which includes its id, its initiator id, and the initiator id of the $BridgeRequest$ sender. When the neighbors of the new bridge nodes receive $BridgeRequest$ or $BridgeReply$, they add the initiator id of the senders to S_{ct}. If $S_{nt} \subseteq S_{ct}$, the border node switches its state to $connected$. Note that only border nodes are involved in the tree connection phase. If nodes in $nonborder$ state receive $BridgeRequest$ or $BridgeReply$, they simply drop the message. Based on this process, we can derive Lemma 1.

Lemma 1 *At least one bridge node will be elected between any two neighboring trees, as long as the network is stable for a period of time.*

Proof: For all border nodes between two neighboring dominator trees, there exists at least one border node which has the highest n_{uc} among its neighboring border nodes. If multiple neighboring border nodes have the same n_{uc}, the one with the smallest id has the priority to be the bridge node. Thus, at least one border node switches its $bstate$ to $bridge$ state within BT_{max} periods. The node which switches to $bridge$ state then broadcasts $BridgeRequest$ to select one additional bridge node to connect to the neighboring tree. Then, the selected bridge node also changes its $bstate$ to $bridge$. Thus, any pair of neighboring trees will be connected by bridge nodes. ■

In the above algorithm, there obviously can be multiple connections between two disjoint neighboring trees. However, since all the bridge nodes eventually will become part of CDS, more connections means more bridge nodes, which will lead to a larger CDS. To further reduce the number of bridge nodes, we introduce Lemma 2.

Lemma 2 *A border node i in unconnected state switches its bstate to connected if $\exists j \in N(i)$, $initiator(j) \neq initiator(i)$ and $bstate(j) = connected$ and $S_{nt}(i) \subset S_{ct}(i) \cup initiator(j)$.*

Proof: Since node j is a neighbor of node i, $initiator(i) \in S_{nt}(j)$. If $bstate(j) = connected$, node j is connected to all neighboring trees directly or indirectly. In other words, node i's and j's trees are already connected. Therefore, if $S_{nt}(i) \subset S_{ct}(i) \cup initiator(j)$, node i can switch to $connected$. ■

Lemma 2 shows that if a node is indirectly connected (i.e., through several nodes) to neighboring trees, it does not have to be a bridge node. This helps to reduce the number of bridge nodes. At the end of this process, all border nodes end up either in $bridge$ or $connected$ state, and bridge nodes will be included as part of CDS. In the proposed tree connection phase, the number of control messages is exactly two times of the number of links between bridge nodes.

D. Example of ME-DTC

In this section, an example is provided to demonstrate how ME-DTC works. In Figure 1, when the dominator tree construction completes, all border nodes are in $unconnected$ state. As border node 8 has the most unconnected neighbors, it starts $BTimer$. After a few beacon periods, if node 8 still has the most unconnected nodes, it switches its $bstate$ to $bridge$. To connect trees, node 8 selects node 10, the border neighbor with the highest n_{uc}, and broadcasts $BridgeRequest$ as shown in Figure 4. On receiving $BridgeRequest$, the neighbors of node 8 insert the initiator ids of nodes 8 and 10 (i.e., node 1 and 2) into their S_{ct}. Now, border node 7, 10, 11, and 12 switch their $bstate$ to $connected$ following Lemma 2. Here, node 5 is in $nonborder$ state, and so it simply drop $BridgeRequest$. After node 10 switches its $bstate$ to $bridge$ to connect the tree of initiator 1, it broadcasts $BridgeReply$ as shown in Figure 5. On receiving $BroadReply$, the neighbors of node 10 add $initiator(10)$ and $initiator(8)$ to S_{ct}. Then, border node 9 switches its $bstate$ to $connected$ following Lemma 2. After receiving the beacon from node 12, node 13 whose $S_{nt} = \{1\}$ knows node 12 is a direct neighbor of node 13 and node 12 is in $connected$ state. In other words, the trees 1 and 2 have already connected by other bridge nodes. As a result, node 13 adds $initiator(12)$ to $S_{ct}(13)$ and now $S_{nt}(13) \subset S_{ct}(13)$. Again, according to Lemma 2 it changes its $bstate$ to $connected$ as shown in Figure 6. In the end, all border nodes are in either $bridge$ or $connected$ state, and the collection of the initiators, dominators, and bridge nodes form a CDS.

E. Implementation Considerations

To maximize the performance of ME-DTC, the proper setting of parameters, such as α and BT_{max}, is critical. It is clear that the number of bridge nodes increases in proportion to the number of initiators. On the other hand, if only few initiators exist in the network, CDS is vulnerable to the initiator failure. Figure 7 shows the number of initiators and the number of bridge nodes with respect to the value of α. The network configuration is provided in Section IV. As can be seen in Figure 7, when the value of α is 1 or 2, too many bridge nodes are added to CDS. Figure 7 suggests that $\alpha \geq 3$ is the proper value to keep the CDS size competitive and avoid serious damage due to the initiator failure. Therefore, we set α to be 3 in the following simulations. For the value of BT_{max}, it should be set based on the probability of beacon collisions. Considering a realistic network configuration where the number of neighbors ranges from 5 to 30, the study in [9] indicates that $BT_{max} = 4$ is enough for nodes to successfully obtain beacons from their neighbors.

F. Mobility Handling

Since the proposed ME-DTC is primarily used to replace the tree connection protocol of MI [10], the resulting CDS protocol naturally inherits the nodal mobility handling of MI to deal with the following four cases:

- The initiator leaves its dominator tree.
- A redundant dominator switches to $dominatee$ state without disconnecting from the dominator tree.
- A dominator leaves its dominator tree.
- A new node joins a dominating tree after the tree is constructed.

Thus, the only additional case we need to address is that a bridge node leaves the network. When this happens, its corresponding bridge node selects another bridge node. For example, in Figure 4, if node 8 leaves the network, node 10 selects a new bridge node, such as node 7. If the bridge node can not find a new bridge node for connection, it switches its $bstate$ to $unconnected$ and their neighboring border nodes also switch to $unconnected$. Then, ME-DTC kicks in again.

IV. Simulation Results

To evaluate the performance of MI with ME-DTC, we implemented our protocol in C++ along with the other CDS protocols, including Dai's [5], Wan's [8], and the original MI [10] protocols with FC-DTC [11] (for simplicity we reference to MI with FC-DTC as MI hence after). In this section, the simulation results of different CDS protocols are reported and analyzed.

A. Simulation Configurations

The nodes are uniformly distributed in a square field with the simulation parameters provided in Table II. If the generated network is not connected, it is discarded and a new network topology is generated to ensure the connectivity of the whole network.

To evaluate the performance of different protocols, three metrics are used: the size of CDS, the convergence time, and the number of extra messages. The convergence time is defined as the number of beacon intervals a protocol takes to complete a CDS. The messages in the tree connection phase of MI and MI with ME-DTC are counted as extra messages. In Dai's protocol, beacons are considered as extra messages because the size of its beacons increases proportionally to the network density and it can be too large compared with the standard beacon frame. In Wan's protocol, the messages exchanged in both phases are counted as extra messages. Each protocol changes the beacon frame format to include additional information. For MI, node id, $color$, initiator id, dominator id, number of uncovered nodes are piggybacked in the beacon. MI with ME-DTC further includes border state, $bstate$ (2-bit), and number of unconnected border nodes, n_{uc} (8-bit), in its beacon frame. The beacon of Dai's protocol includes node id, $state$, $marker$, and the list of ids for one-hop neighbors. In Wan's protocol, dominator id and $color$ are added to the beacon.

B. Simulation Results for Different CDS Protocols

In this section, the simulation results of different CDS protocols are presented.

Figure 8 shows the CDS size with respect to the network density. It is clear that MI consistently generates the smallest CDS and Dai's consistently generates the largest CDS among all the protocols. In addition, except Dai's protocol, the sizes of

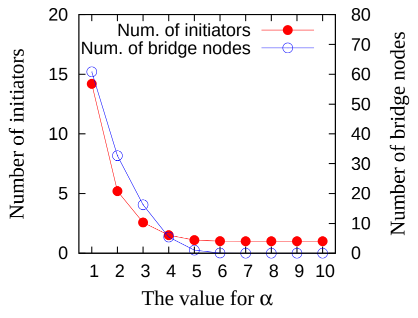

Fig. 7. The number of initiators and bridge nodes.

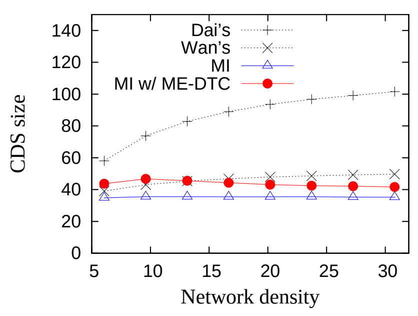

Fig. 8. CDS size.

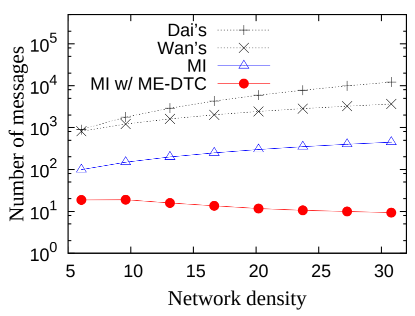

Fig. 9. Number of messages.

TABLE II
SIMULATION PARAMETERS

Symbol	Definition (default value)
The simulation area	$1000m$ by $1000m$
The number of nodes	100 to 450
The transmission range	$150m$
The value of α	1 to 10 (3)
The value of BT_{max}	4
One beacon period	1 second
The number of simulations	1000

CDS generated by other protocols are not sensitive to the density of the network. In case of high network density, MI with ME-DTC results in smaller size of CDS than Wan's protocol. Compared with the original MI protocol, MI with ME-DTC creates a slightly larger size of CDS, but the difference is not significant.

Figure 9 demonstrates the number of extra messages with respect to the network density. MI with ME-DTC outperforms the other CDS protocols significantly because it introduces very few control messages. To be exact, MI with ME-DTC introduces less than 10% of total messages of MI, and less than 0.1% of total messages than Wan's and Dai's.

Figure 10 shows the convergence time of different CDS protocols with respect to the network density. MI with ME-DTC requires almost the same convergence time as MI to form a CDS. The convergence time of Dai's protocol increases as the number of neighbors increases, since a node waits for 2 beacon periods to initiate the CDS protocol if its neighbors change their *state*. Unlike Dai's protocol, the convergence time of MI with ME-DTC is not affected by the network density. Compared with Wan's protocol, MI with ME-DTC completes CDS construction faster.

V. ANALYSIS

In this section, we build the analytical model to estimate the average CDS size and number of messages for MI with ME-DTC, and provide the comparisons between our analytical and simulation results.

A. The Analysis for The CDS Size

In our CDS protocol, the combination of initiators, dominators, and bridge nodes form a CDS. Therefore, the size of CDS, denoted by n_{cds}, can be obtained by Equation 1.

$$n_{cds} = n_{init} + n_{dom} + n_{bridge} \tag{1}$$

In Equation 1, n_{init} is the average number of initiators, n_{dom} is the average number of dominator nodes, and n_{bridge} is the average number of bridge nodes. As indicated in [14], the expected number of initiators can be obtained by Equation 2.

$$n_{init} = \max\{n \cdot (\frac{1}{\frac{n\pi(r+(\alpha-1)\bar{d})^2}{S}+1}), 1\} \tag{2}$$

Here, n is the number of nodes in the region S, and $\bar{d}$ is the average distance between two nodes, which based on [14], is $\frac{2}{3}r$, where r is the transmission range. From the property of geometry the average intersection area of two nodes is $\int_0^r \frac{2\pi x INTC(r,x)}{\pi r^2}dx = 0.59r^2$, where $INTC(r,d) = 4\int_{d/2}^{r}\sqrt{r^2-x^2}dx$.

According to ME-DTC, if a neighbor of a node in the intersection of transmission areas of the node and its dominator switches to *dominator* state, the node most likely will switch to *doinatee* state. Thus, the number of dominator nodes of each initiator can be estimated by $\frac{\sigma\{\frac{2}{3}r(h-1)\}^2-1}{0.59r^2\pi\sigma}$, where $\sigma = n/S$ and h is the average height of dominator trees (the number of hops from the initiator to the edge of a tree) in [14]. Hence, the expected number of dominator nodes in a network can be obtained by Equation 3.

$$n_{dom} = n_{init} \cdot \frac{\sigma\{\frac{2}{3}r(h-1)\}^2-1}{0.59r^2\pi\sigma} \tag{3}$$

The average boundary area where two trees meet can be approximated by $\pi(\bar{d}h)^2 - \pi\{\bar{d}(h-1)\}^2$. A node which has a bridge neighbor in the same tree do not become a bridge node for the same neighboring tree. Therefore, the number of bridge nodes in a dominator tree can be estimated by the number of nodes in the boundary area divided by three. Since there exist $\binom{n_{init}}{2}$ possible neighboring trees, the expected number of bridge nodes in a network can be calculated by Equation 4.

$$n_{bridge} = \binom{n_{init}}{2} \cdot \frac{[\pi(\bar{d}h)^2 - \pi\{\bar{d}(h-1)\}^2]}{3\pi S} \tag{4}$$

Since the expected value of n_{init}, n_{dom}, and n_{bridge} are obtained by Equation 2, 3, and 4, the average CDS size obtained by MI with ME-DTC can be estimated accordingly in Equation 1.

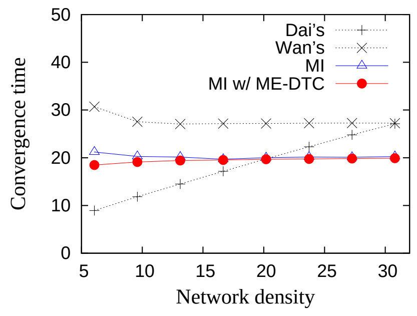

Fig. 10. Convergence time.

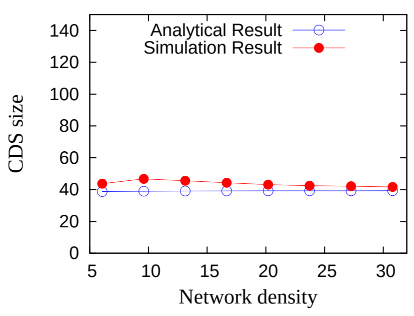

Fig. 11. CDS size.

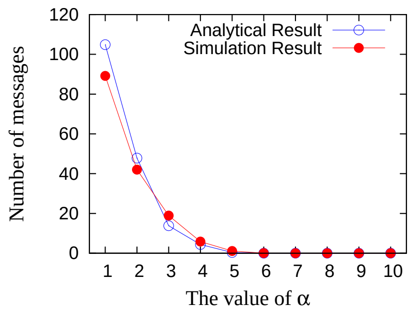

Fig. 12. The number of messages.

B. Analysis for The Number of Messages

Based on the expected value of n_{bridge}, we can estimate the average number of messages. The number of messages is twice of the number of links between bridge nodes, which is close to the number of bridge nodes. To be more precise, some bridge nodes have links to connect to more than one trees, and they broadcast control messages to connect to each neighboring tree. The probability that a bridge node has more than one link to different neighboring trees is $(1 - \frac{1}{n_{init}})^{0.59(\sigma-1)}$. The expected number of messages is the number of bridge nodes plus the number of bridge nodes which has more than one link to different neighboring trees, which can be approximated by Equation 5.

$$n_{bridge} + n_{bridge}(1 - \frac{1}{n_{init}})^{0.59(\sigma-1)} \quad (5)$$

C. The Comparisons between Analysis and Simulations

In this section, we compare the results obtained by our analytical model and simulations. The configurations used in analysis and simulations are the same as those shown in Table II. Figure 11 illustrates the CDS size with respect to the network density. As shown in the figure, the results from analytical model and simulations are close to each other. This indicates that our analysis to estimate the expected size of CDS is accurate.

Figure 12 illustrates the number of messages with respect to the value of α. This implies that our analysis to estimate the expected number of messages is accurate. From Figure 11 and 12, we are able to establish the validity of our analytical models for MI with ME-DTC in terms of the size of CDS and the number of messages.

VI. Conclusion

It is preferred that CDS protocols results in a small size of CDS, incurs less communication overhead, completes CDS quickly, and adapts to the changes of network topology. The MI protocol [10] is known to be effective in many performance metrics, but it tends to introduce too many control messages in its dominator tree connection phase. In this paper, we proposed the Message-Efficient Dominator Tree Connection (ME-DTC) protocol, which reduces control messages required in the tree connection phase of tree-based CDS protocols. The simulation results show that MI with ME-DTC outperforms significantly than other protocols in terms of message overhead and is comparable to the best results of the other CDS protocols in other performance metrics. In addition, analytical models are built to estimate the expected CDS size and number of messages. The results obtained by our analytical models and simulations are close enough to validate the accuracy of our analytical models and the protocol implementation in the simulations.

References

[1] J. Blum, M. Ding, A. Thaeler, and X. Cheng, "Connected Dominating Set in Sensor Networks and MANETs," 2005, pp. 329–369.

[2] J. Wu, "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links," *IEEE Transaction on Parallel Distributed Systems*, vol. 13, no. 9, pp. 866–881, 2002.

[3] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized Minimum-Energy Broadcasting in Ad-hoc Networks," in *Proceedings IEEE Conference on Computer Communications (INFOCOM)*, Mar. 2003, pp. 2210–2217.

[4] D. S. Hochbaum, Ed., *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., 1997.

[5] F. Dai and J. Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks," *IEEE Transaction on Parallel Distributed Systems*, vol. 15, no. 10, pp. 908–920, 2004.

[6] D. Simplot-Ryl, I. Stojmenovic, and J. Wu, "Energy-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks," *Handbook of Sensor Networks: Algorithms and Archtectures*, pp. 343–379, 2006.

[7] P. J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2002, pp. 141–149.

[8] P.-J. Wan, L. Wang, and F. Yao, "Two-Phased Approximation Algorithms for Minimum CDS in Wireless Ad Hoc Networks," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2008, pp. 337–344.

[9] D. Zhou, M.-T. Sun, and T. Lai, "A Timer-based Protocol for Connected Dominating Set Construction in IEEE 802.11 Wireless Networks," in *Proceedings of International Symposium on Applications and the Internet (SAINT)*, Jan. 2005, pp. 2–8.

[10] K. Sakai, F. Shen, K. M. Kim, M.-T. Sun, and H. Okada, "Multi-Initiator Connected Dominating Set for Mobile Ad Hoc Networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, May 2008.

[11] K. Sakai, M.-T. Sun, and W.-S. Ku, "Fast Connected Dominating Set Cnstruction in Mobile Ad Hoc Networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, June 2009.

[12] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[13] "IEEE 802.11b Standard," http://standards.ieee.org/getieee802/download/802.11b-1999.pdf.

[14] K. Sakai, M.-T. Sun, W.-S. Ku, and H. Okada, "Maintaining CDS in Mobile Ad Hoc Networks," *Lecture Notes in Computer Science - Wireless Algorithms, Systems and Applications (WASA)*, vol. 5258, pp. 141–153, Oct. 2008.