Connected Dominating Set Construction Algorithm for Wireless Networks Based on Connected Subset

Qiang Tang[1,2], Yuan-Sheng Luo[1,2], Ming-Zhong Xie[1,2], and Ping Li[1,2]
[1]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha (410114), China
[2] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha (410114), China
Email: {tangqiangcsust, xiemingzhongcsust, lping9188}@163.com; luodyx@msn.com

Abstract —In this paper, a Connected Dominating Set (CDS) construction algorithm CSCDS (Connected Subset based CDS) is proposed, which is based on the connected subset concept. The CSCDS contains two main stages, which are dominating set construction stage and connected dominating set construction stage respectively. In the first stage, the dominators are proposed based on the one hop white neighbor information, and the redundant dominators are reduce to obtain the minimum number of dominators. In the second stage, the CDS is constructed based on the connected subset, which is used as the basis to select the connectors. The message complexity is $O(\Delta^2)$ ($O(\Delta^2)$ is a linear relationship function with Δ^2, and Δ is the maximum one hop neighbor degree), Simulation results show that CSCDS has smaller size compared with the classical algorithms.

Index Terms—Connected dominating set, connected subset, wireless networks, dominating set construction stage, connected dominating set construction stage

I. INTRODUCTION

Topology Control (TC) is a very important issue for wireless networks and wireless sensor networks. The main purposes of TC are reducing the size of the backbone, saving energy consumption, keeping networks connectivity and reliability as well as fault-torrent and coverage, etc [1]. The main methods for achieving the TC are mainly changing the communication range, scheduling the nodes' active and sleep mode, etc [2].

One of the topology construction approaches is building the hierarchical topologies, which contains an important step, which is building a backbone also named Minimum Connected Dominating Set (MCDS). The MCDS construction problem is an NP-hard problem [3]. In [4], the MCDS construction algorithms are classified into the MIS (Maximum Independent Set)-based algorithms and non-MIS-based or Tree-based algorithms.

For the MIS-based algorithms, there are many research works. In [5], P. J. Wan *et al.* at first deduce a lower bound, which is *O(nlogn)*, of the message complexity of distributed algorithms for leader election, spanning tree and nontrivial CDS. Then a two stage MCDS construction algorithm is proposed, in which the MIS nodes are selected based on the lower level neighbor's states and the connectors are selected based on the maximum dominator node degree. In [6], Y. Li *et al.* propose a greedy algorithm S-MIS as well as its variation rS-MIS. In the first stage, a MIS is constructed with the size upper-bound as *4opt+1.2*, and then in the second stage a greedy approximation algorithm selecting connectors to interconnect the nodes in the MIS is proposed, which has the performance ratio as *(2+ln4)opt*. The final S-MIS performance ratio is *(5.8+ln4)opt*. In [7], M. T. Thai *et al.* propose two algorithms named TFA (The First Algorithm) and TSA (The Second Algorithm) respectively. In TFA, the MIS is constructed by using the algorithm in [5], and the connectors are selected based on the black neighbor number in different black-blue components. In TSA, the node with the maximum radius is selected as dominator, and the connectors are selected based on highest number of black-blue components. The size relationship between a CDS and a maximum independent set is also derived. In [8], H. Du *et al.* consider both the CDS size and the routing path length for the MCDS. The algorithm contains two stages: constructing a MIS stage and connecting the MIS stage. The path length between any node *u* and any node *v* is at most *7sp(u,v)*, where *sp(u,v)* is the shortest path between *u* and *v*. In [9], Y. Xiang *et al.* propose two distributed algorithms for wireless networks. The first algorithm (XFA) is based on a growing tree and the second algorithm (XSA) is based on the MIS. The authors also derive the relationship of any MIS and MCDS in geometric k-disk graphs.

Besides the MIS-based MCDS construction algorithms, there also exist non-MIS-based MCDS construction algorithms. In [10], F. Dai *et al.* propose the redundant dominators reduction mechanism rule-k. In [11], K. Sakai *et al.* bring forward two timer-based CDS protocols, which are Single-Initiator (SI) version CDS and Multi-Initiator (MI) version CDS. In the SI version, single

Manuscript received July 16, 2015; revised January 6, 2016.
This work is supported in part by Scientific Research Foundation of Hunan Provincial Education Department (No. 13C1022, No. 13C1023 and No. 14A004), the Natural Science Foundation of Hunan Province, China (Grant No. 13JJ4052) and the National Natural Science Foundation of China (Grant No. 61303043).
Corresponding author email: tangqiangcsust@163.com.
doi:10.12720/jcm.11.1.50-57

initiator is generated, while in the MI version, multiple initiators are generated. Both of them can handle mobility. All the initiators are the roots of the generated trees, and in the SI version, the MCDS is a single tree, while in the MI version, multiple trees generated and should be connected together to form a MCDS. The growing tree in [11] is also a non-MIS-based algorithm.

Besides the regular methods for MCDS construction, there are also novel methods or issues about MCDS investigated. In [12] P. M. Wightman *et al.* propose an optimal solution for MCDS construction by using a mixed integer programming. The simulation results show that the mathematical programming method performs well in the low density and high node degree topologies. In [13] J. A. Torkestani put forward a degree-constrained minimum-weight version of CDS construction algorithm based on learning automata, and a heuristic algorithm is proposed for finding a near optimal solution. In [14] J. He *et al* consider load-balance factor, and investigate the MinMax Degree Maximal Independent problem, the Load-Balance Virtual Backbone problem as well as the MinMax Valid-degree non-Backbone node Allocation problem. In [15], Z. Lin *et al* study the MCDS problem in the cognitive radio networks, where the communication links are prone to failure. They consider both the maximum network lifetime and the minimum CDS size. A three stages centralized algorithm as well as two stages localized algorithm is proposed. In [16], J. Yu *et al.* investigate the extended dominating set (EDS), and propose three algorithms to find EWCDSs (Extended Weakly Connected Dominating Set). Each forward node covers itself and regular neighbor as well as partly quasi-neighbors. In [17], A. Buchanan *et al.* propose a MCDS construction algorithm with restricted diameter by using the compact integer programming formulation. A PSO-Optimized Topology Control Scheme is used in [18] by W. Guo *et al.* to construct a Minimum Spanning Tree, which can also inspire us to construct a MCDS. The problem is modeled as a model of multi-criteria degree constrained minimum spanning tree, which is solved by a non-dominated discrete particle swarm optimization. J. Qiao *et al.* in [19] present a local minimum spanning tree based topology control algorithm by applying the 0-1 robust discrete optimization theory, and a robust model for the MST problem was also put forward, which is solved by an algorithm. The methods in this paper can also lead us to the further research about MCDS construction. In [20], Y. Huang *et al.* survey the techniques about building the k-connected network as well as constructing a multi-path routing, which is also beneficial to constructing a k-connected MCDS or backbone.

The MCDS construction algorithms above are all MIS based and non-MIS based, and many kinds of methods have been used to construct MCDS. But the algorithm based on the connected subsets is not proposed, and thus we put forward a new method based on the connected subset concept. The algorithm needs to construct a dominator set which covers all the nodes in this network at first, and then construct the connected subsets which are composed by the dominators. And finally, the connectors are selected to connect the connected subsets until a CDS is constructed. The method in this paper does not require the MIS, and we use a simple method to construct a minimum dominator set. The core method is the connector election algorithm, which selects the connectors based on the connected subset degree, which is similar to the node degree. The algorithm performs well in term of CDS size compared with the classical algorithms.

II. CSCDS ALGORITHM

A. Preliminary Knowledge

Neighbor Definition: If and only if $d(u,v)<min(r_u,r_v)$, node u and v are one-hop neighbor of the other one respectively. Where $d(u,v)$ denotes the distance between node u and v. r_u and r_v denote the transmission ranges of node u and v respectively.

Color Definition: Dominator is marked black, and dominated node is marked grey. If a node is neither a dominator nor a dominated node, it is an independent node and marked white.

Neighbor Sets Definition: Every node maintains a one hop neighbor set *OHNS* and a one hop white neighbor (independent neighbor) set *OHWNS*. There are also one hop black neighbor (dominator neighbor) set *OHBNS* and one hop grey neighbor set *OHGNS* for every node.

Connected Subset Definition: In a dominator set, if any two dominators u and v are connected with each other through the dominators which all belong to this dominator set, the dominator set is a connected subset.

Other Set Variables Definition: For every dominated node (grey node), it receives the connected subset *ids* from its one hop dominators and stores them into set *CSIDS*. Every grey node exchanges *CSIDS* with its one hop grey neighbor and gets two hop connected subset *ids* and stores into set *TCSIDS*. Every grey nodes stores all the connected subset *ids* within two hops into the following set: $ALLCSIDS=CSIDS \cup TCSIDS$. The set *CONIDS* is the set of connected subsets *ids*.

B. Dominating Set Construction Stage

This stage contains two algorithms: dominator election algorithm and redundant dominator reduction algorithm.

In the dominator election algorithm, every node gets the one hop independent neighbor information, and then saves the information into set *OHWNS*. By exchanging the size of *OHWNS* with one hop neighbor, the node selects the node with the maximum *OHWNS* size as a dominator. The algorithm is shown as follows:

Dominator Election Algorithm (DEA)

for every node i

01: if *i* is white

02: gets one hop independent neighbor information and saves into $OHWNS_i$

```
    exchanges the size of OHWNS_i with one hop neighbors
    sorts the neighbors according to the size of OHWNS the in descending order
  selects the nodes with the maximum size of OHWNS
  stores them into a temporary set TMP
  selects the node j with the minimum id in TMP
  sends an elected message to node j
end
if receives an elected message
   marks itself black and broadcasts a black message
end
if receives a black message
    if it is white
       marks itself grey and broadcasts a grey message
   end
   updates neighbor set OHBNS_i
end
if receives a grey message
   updates neighbor set OHGNS_i
end
```

The sketch map of DEA is shown in Fig. 1:

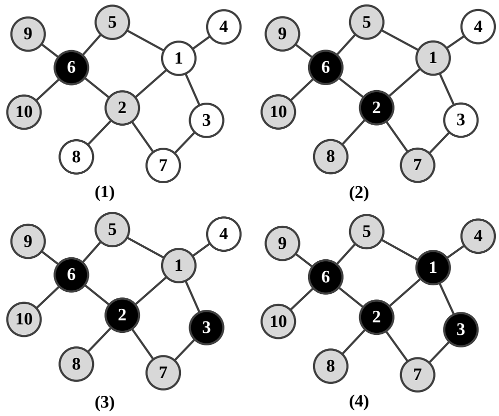


Fig. 1. The sketch map of DEA

In Fig. 1, we assume the node execute following the order {10, 9, 8, 7, 6, 5, 4, 3, 2, 1}. Because node 6 has the maximum size of *OHWNS* compared with node 10 and 9, node 10 and 9 select node 6 as the dominator (shown in Fig 1(1)). Then node 8 selects node 2 as the dominator, shown in Fig. 1(2). Node 7 selects node 3 as the dominator, shown in Fig 1(3). Node 4 finally selects node 1 as dominator, shown in Fig. 1(4). Node 1, 2, 3 and 6 compose the dominator set.

By executing the DEA, the dominators cover all the nodes, but there are a few redundant dominators, such as node 3, which should be eliminated by an algorithm shown in the following pseudo code.

Redundant Dominator Reduction Algorithm (RDRA)

for dominator *i*

```
if OHBNS_i is not empty
    get the OHBNS_j from the node j∈OHGNS_i
    if  OHBNS_j ≥2 for every j
         dominator i marks itself grey
         broadcasts a grey message
   end
end
```

In the Fig. 2(1), the node 1, 2, 3 and 6 compose the dominator set. According to RDRA, nodes in *OHGNS* of node 3 are also dominated by node 2. Besides the *OHBNS* of node 3 is not empty, thus node 3 is a redundant dominator, and node 3 marks itself grey shown in Fig.2 (2).

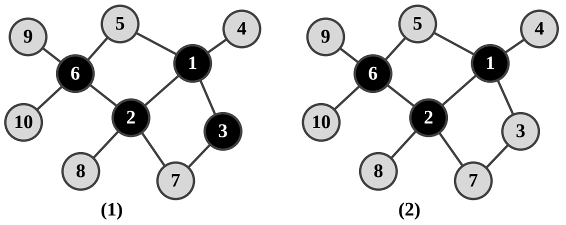


Fig. 2. Execution of RDRA

Because all the grey neighbors of node 6 are not covered by other dominator besides node 6, node 6 is not a redundant dominator. The same situation is also suitable for node 1 and 2. Finally, the node 1, 2 and 6 compose the minimum dominator set.

C. Connected Dominating Set Construction Stage

This stage contains two algorithms: connected subset construction algorithm and CDS construction algorithm.

In the connected subset construction algorithm, every dominator for example dominator *i* exchanges the $OHBNS_i$ with the dominators in its $OHBNS_i$ and updates its $OHBNS_i$ by including the *OHBNS* of nodes belonging to $OHBNS_i$.

The progress is repeated twice, and finally dominator *i* get the dominators information within three hops, which is the connected subset in the view of *i*. The algorithm is shown in following:

Connected Subset Construction Algorithm (CSCA)

for dominator *i*

1: exchanges $OHBNS_i$ with nodes belonging to $OHBNS_i$

2: updates $OHBNS_i = OHBNS_i \cup OHBNS_j$, $j \in OHBNS_i$

3: repeats 1 and 2 once more and gets the connected subset as $OHBNS_i$ in the view of dominator *i*.

4: get the connected subset id: $csid = \min\{j | j \in OHBNS_i\}$

5: broadcasts *csid* to the grey nodes in $OHGNS_i$

for dominated node *k*

1: if receives a *csid* from its one hop dominator neighbor

2: puts the *csid* into set $CSIDS_k$

3: end

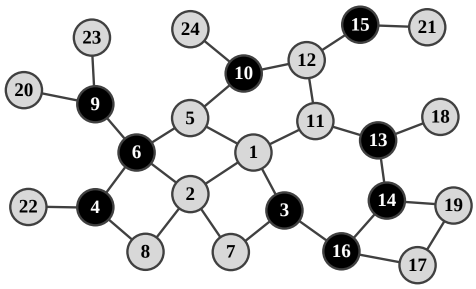


Fig. 3. Execution of CSCA

The execution of CSCA is shown in Fig. 3. According to CSCA and in Fig. 3, node 10 is a connected subset with only one element 10. Node 15 is same as node 10. Nodes 4, 6, 9 have the same view of connected subset as {4, 6, 9}, and the *csid* is 4. Nodes 3, 13, 14 and 16 have the same view of connected subset as {3, 13, 14, 16}, and the *csid* is 3.

If the dominator gets the only one *csid*, it broadcasts it to the one hop grey neighbor. For example, node 12 receives *csid* from node 15 and 10. Node 11 receives *csid* from node 13 *et al.*

CDS Construction Algorithm (CDSCA)

for dominated node *i*

```
exchanges CSIDS_i with nodes belonging to OHGNS_i , flag=1
gets two hop csids (tcsids) and stores them into TCSIDS_i:
      TCSIDS_i={ids|ids∈CSIDS_j\ CSIDS_i, for all j∈OHGNS_i }
exchanges TCSIDS_i with nodes belonging to OHGNS_i
if size(CSIDS_i)=1 & size(TCSIDS_i)=0
    flag=0;
end
computes all csids and store into set ALLCSIDS_i:
      ALLCSIDS_i= CSIDS_i∪TCSIDS_i
for each node k∈OHGNS_i∪THGNS_i
    calculates ALLCSIDS_k= CSIDS_k∪TCSIDS_k
    if ALLCSIDS_i⊂ALLCSIDS_k
        flag=0; break;
    elseif ALLCSIDS_i=ALLCSIDS_k
        if k<i
            flag=0; break;
        end
    end
end
if flag=1, i becomes to be a connector
if i becomes a connector
    if connectorflag_i=1
        ALLTIDS_i= TCSIDS_i\ CONIDS_i
    else
        ALLTIDS_i= TCSIDS_i
    end
    while(1)
      selects a grey neighbor v as a connector, according to:
        size(CSIDS_v)>size(CSIDS_k), for all k∈OHGNS_i;
    connectorflag_v=1;
    CONIDS_v= CONIDS_v∪TCSIDS_i∪CSIDS_i
    updates ALLTIDS_i = ALLTIDS_i \ CSIDS_v
    updates OHGNS_i= OHGNS_i\{v}
    if ALLTIDS_i is empty
        break;
    end
  end while
end
```

After executing the CSCA, there are some connected subsets in the network. In order to connect all the connected subsets, we select the grey nodes as the connectors according to CDS construction algorithm.

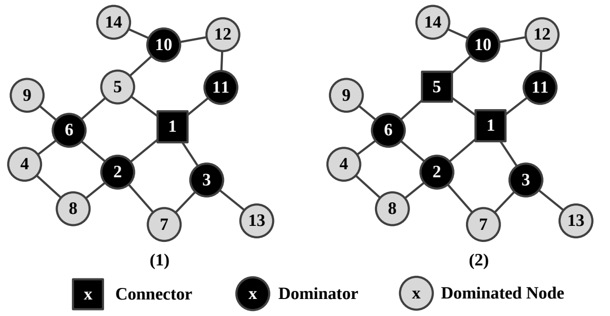


Fig. 4. Execution of CDSCA

The execution of CDSCA is shown in Fig. 4. In Fig. 4, network has connected subsets' ids as 2, 3, 10, and 11. We can get the following information:

Node 5's $ALLCSIDS_5$ is {2, 10, 11, 3}

Node 1's $ALLCSIDS_1$ is {2, 3, 11, 10}

$ALLCSIDS_9=ALLCSIDS_4=ALLCSIDS_8=\{2\}$

$ALLCSIDS_{14}=\{10\}$

$ALLCSIDS_{12}=\{10, 11\}$

$ALLCSIDS_{13}=\{3\}$

$ALLCSIDS_7=\{2, 3\}$

Because $ALLCSIDS_5= ALLCSIDS_1$, and node 1 has the maximum number compared with its one and two hop neighbor. And compared with node 5, node 1 has the minimum node id. Thus node 1 is a connector, shown in Fig. 4 (1). In order to cover node 5's two-hop connected subsets 11 and 3, node 5 select node 1 as the next level connector. Because *ALLCSIDSs* of node 4, 8, 9, 7, 13, 14, 12 are all included in $ALLCSIDS_5$ and $ALLCSIDS_1$, then they all drop to be a connector. The final CDS is shown in Fig. 4 (2).

III. Theoretical Analysis

Lemma 1: The dominating set constructed by DEA covers all the nodes in this network.

Prove: Assume there is still a node *i* not dominated by the dominators selected according to DEA. Then node *i*'s neighbor for example node *j* has a non-empty *OHWNS*.

According to DEA, node *j* will receive an elected message from its one hop neighbor and mark itself black. Thus node *i* is dominated by dominator node *j*, and lemma 1 is proved.

Lemma 2: There exists at least a connected subset in three hops for every connected subset.

Prove: Assume there is a connected subset having no neighbor connected subset within three hops, shown in Fig.5.

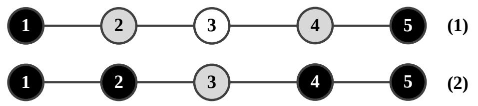


Fig. 5. Three hops circumstance

In Fig. 5, the distance between connected subset 1 and 5 is three hops. According to Lemma 1, node 3 is not dominated and is an independent node, marked as white, shown in Fig. 5 (1). Thus we know that the dominating set don't cover all the nodes in this network, which is contrary to Lemma 1.

If according to lemma 1 and DEA, node 3 should sent elected messages to node 2 and node 4, and the real dominating set is {1, 2, 4, 5}. The correct connected subsets ids are 1 and 4, shown in Fig. 5 (2). Thus, lemma 2 is proved.

Theorem 1: The set constructed by CDSCA is a connected dominating set.

Prove: Because the CDSCA is executed based on the dominating set constructed by DEA and RDRA and there is no dominator deleted in the execution of CSCA and

CDSCA. Thus the set constructed by CDSCA includes the dominating set constructed by DEA and RDRA, and is a dominating set.

After the CDSCA is executed and assume there is still a connected subset, which is not connected with its neighbor connected subset within three hops, shown in Fig. 6.

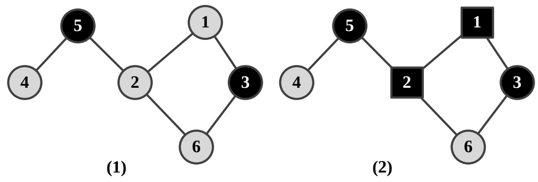


Fig. 6. Example of not connected

From Fig. 6 (1), we know that connected subset 5 and 3 are within three hops and not connected with each other. According to CDSCA, node 1, node 2 and node 6 have the same *ALLCSIDS* and node 1 has the minimum id. Thus node 1 becomes a connector and selects node 2 as a next level connector, shown in Fig. 6 (2).

Thus, we know that all the connected subset can be connected with its neighbor connected subsets within three hops. Then, according to lemma 1 and lemma 2, we know that the final set composed of dominators and connectors is a connected dominating set.

A. *Message Complexity*

We assume the maximum size of one hop neighbor set is Δ.

In the dominating set construction stage, if DEA is executed by node *i*, it should at most receive Δ neighbor status messages and Δ neighbor status changed messages. Thus, the message count of DEA received by node *i* is 2Δ. If RDRA is executed, dominator *i* receives Δ messages from its neighbor. Thus the dominating set construction stage has the maximum message count as 3Δ for any node.

In the connected dominating set construction stage, if the CSCA is executed by dominator *i*, it broadcasts a message two hops and gets the dominator information within three hops. Thus, the message count is at most Δ^2. If CDSCA is executed, the dominated node *i* needs the *csids* from the nodes within two hops, and the message count is at most Δ^2.

In summary, the CDS construction algorithm executed by one node at most requires $2\Delta^2+5\Delta$ messages. The message complexity is $O(\Delta^2)$.

IV. Performance Evaluation

A. *Parameters Settings*

In this section, we compare the size of CSCDS with other four algorithms, which are Y. Xiang's algorithms in [9] denoted by XFA and XSA respectively, and M.T. Thai's algorithms in [7], denoted by TFA and TSA respectively.

We assume all the nodes are deployed randomly in a square area, which has the length of a side as *M* meters.

In the following comparison, we first evaluate the impact of the node number on the CDS size. And then simulate the impact of the Transmission Range (TR) on the CDS size. Finally, the impact of the network area size on the CDS size is evaluated also.

B. *Number of Nodes Impact*

In this simulation, the length of a side of the square area is 200 meters. The node number varies from 50 to 200 with the step as 10. The transmission range intervals are set as [50, 70] for the circumstances, in which all the nodes have different transmission ranges.

We also set the transmission range as 60 for the circumstances, where all the nodes have the same transmission ranges.

The simulation results are shown in Fig. 7 and Fig. 8.

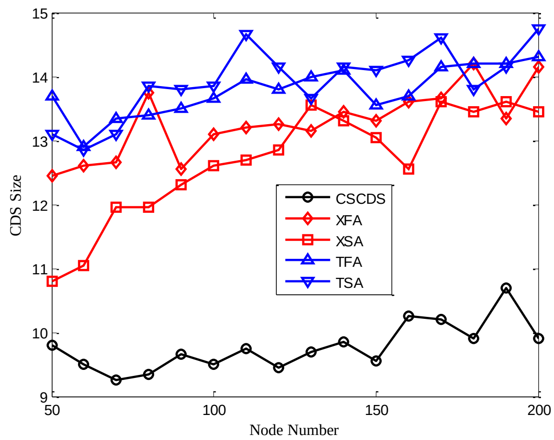


Fig. 7. Transmission range equals to 60 meters

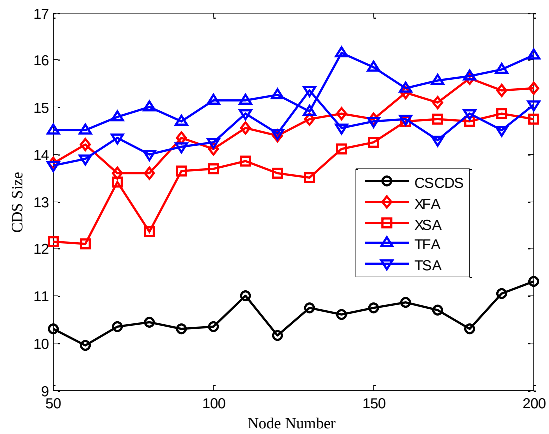


Fig. 8. Transmission range belongs to [50, 70]

According to the Fig. 7 and Fig. 8, we find that CSCDS has the minimum CDS Size compared with the other four algorithms.

As the node number increases, the CDS size increases slightly, which is because all the nodes are deployed randomly in a fixed network area, and the transmission range fixed or belong to a fixed interval. As the number of node increases, only approximately the fixed number of dominators is needed to cover all the network area, thus the CDS Size increases slightly.

C. Transmission Range Impact

In this simulation, the square area is 200×200. The node number is 100.

At first, the transmission range varies from 30 to 90 with the step as 5 for the situation, where all the nodes have the same transmission ranges.

Then we set the transmission range belong to the intervals which is from the interval [20, 40] to the interval [80, 100]. The minimum as well as the maximum value of these intervals varies with the step as 5.

The simulation results are shown in Fig. 9 and Fig. 10.

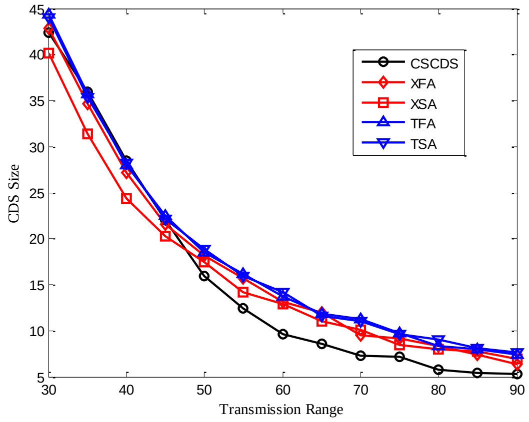


Fig. 9. Transmission range equals to a fix number

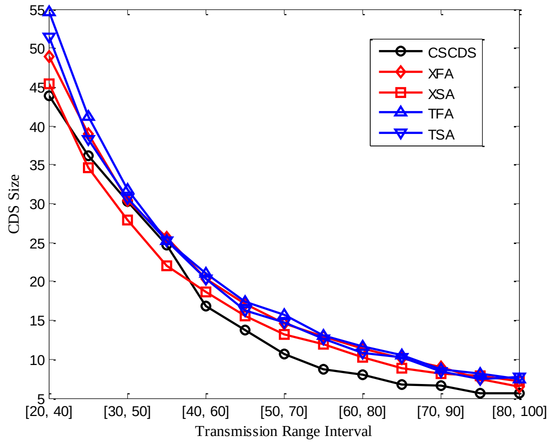


Fig. 10. Transmission range belongs to an interval

From the Fig. 9 and Fig. 10, we find that the CSCDS has the minimum CDS size when the transmission range is bigger than 40 meters, while if the transmission range is smaller than 40 meters, the CDS size of CSCDS is not the minimum, for if the transmission range is very small, one node can not cover a lot of neighbors and may create a lot of dominators, which is suitable for the other four algorithms. So, if the network is very sparse, our algorithm as well as other four algorithms may not performs well enough.

We can also find that as the transmission range increases, the CDS size decreases, which is because as the TR increases, smaller number of dominators are needed to cover all the network nodes compared with the bigger transmission range.

D. Network Area Size Impact

In this simulation, we set the node number as 100, and the transmission range is 50 meters for the circumstances where all nodes have the same transmission range. We also set the transmission range as [40, 60] for the circumstances in which all nodes have different transmission ranges. The network area is square area. The length of side varies from 100 meters to 200 meters with the step as 5. The simulation results are shown in Fig. 11 and Fig. 12.

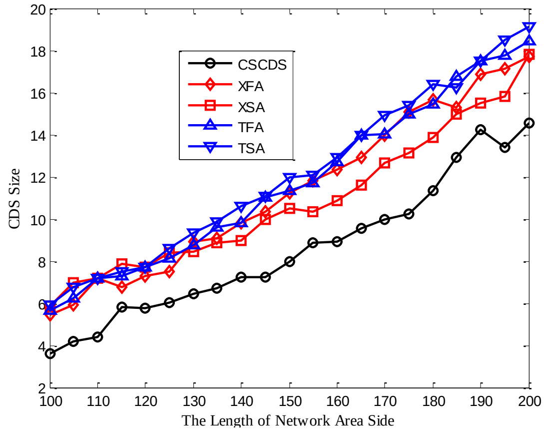


Fig. 11. Transmission range equals to 50 meters

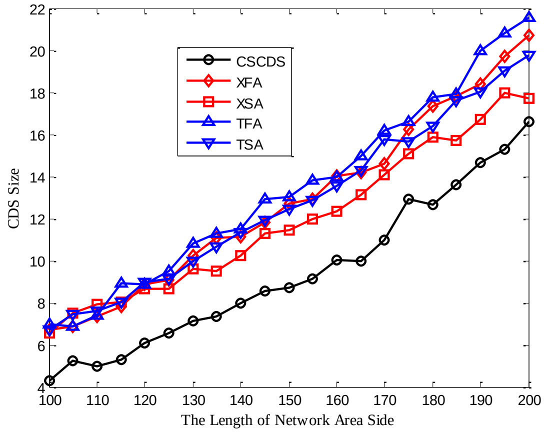


Fig. 12. Transmission range belongs to [40, 60]

According to Fig. 11 and Fig. 12, we find that the CDCDS has the minimum CDS size compared with the other four algorithms.

As the length of network area side increases, the CDS size increases gradually, for if the network area side length increases, the network becomes a sparse network gradually and the number of neighbors covered by a node is less and less, which causes more dominators are needed to cover all the network area and increases the CDS size.

V. Conclusions

In this paper, a connected dominating set construction algorithm CSCDS is proposed based on the connected subset. CSCDS has two stages and contains four sub-algorithms. The message complexity is O(Δ^2) and simulation results show that CSCDS has smaller size compared with Y. Xiang's algorithms and M.T. Thai's algorithms.

ACKNOWLEDGEMENTS

This work is supported in part by Scientific Research Foundation of Hunan Provincial Education Department (No. 13C1022, No. 13C1023 and No. 14A004), the Natural Science Foundation of Hunan Province, China (Grant No. 13JJ4052) and the National Natural Science Foundation of China (Grant No. 61303043).

REFERENCES

[1] M. Abbasi, M. S. B. A. Latiff, and H. Chizari, "An overview of distributed energy-efficient topology control for wireless ad hoc networks," *Mathematical Problems in Engineering*, vol. 4, pp. 708-715, Apr. 2013.

[2] Y. Huang, J. F. Martínez, V. H. Díaz, and J. Sendra, "A novel topology control approach to maintain the node degree in dynamic wireless sensor networks," *Sensors*, vol. 14, no. 3, pp. 4672-4688, Mar. 2014.

[3] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks–A comprehensive survey," *Computer Communications*, vol. 36, pp. 121–134, Jan. 2013.

[4] Q. Tang, K. Yang, P. Li, J. Zhang, Y. Luo, and B. Xiong, "An energy efficient MCDS construction algorithm for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, pp. 519-533, Mar. 2012.

[5] P. J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proc. 21th IEEE Annual Joint Conference on Computer and Communications Societies*, New York, 2002, pp. 1597-1604.

[6] Y. Li, M. T. Thai, F. Wang, C. W. Yi, P. J. Wan, and D. Z. Du, "On greedy construction of connected dominating sets in wireless networks," *Wireless Communications and Mobile Computing*, vol. 5, pp. 927-932, Nov. 2005.

[7] M. T. Thai, F. Wang, D. Liu, S. Zhu, and D. Z. Du, "Connected dominating sets in wireless networks with different transmission ranges," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 721-730, July 2007.

[8] H. Du, W. Wu, Q. Ye, D. Li, W. Lee, and X. Xu, "CDS-Based virtual backbone construction with guaranteed routing cost in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 652-661, Apr. 2013.

[9] Y. Xiang, K. Xing, W. Cheng, E. K. Park, and S. Rotenstreich, "Distributed virtual backbone construction in sensor networks with asymmetric links," *Wireless Communications and Mobile Computing*, vol. 11, pp. 1051-1060, Aug. 2011.

[10] F. Dai and J. Wu, "An extended localized algorithms for connected dominating set formation in ad hoc wireless networks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 15, pp. 908-920, Oct. 2004.

[11] K. Sakai, S. C. H. Huang, W. S. Ku, M. T. Sun, and X. Cheng, "Timer-Based CDS construction in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1388-1402, Oct. 2011.

[12] P. M. Wightman, A. Fabregasy, and M. A. Labradorz, "An optimal solution to the MCDS problem for topology construction in wireless sensor networks," in *Proc. IEEE Latin-American Conference on Communications*, Bogota, 2010, pp. 1-6.

[13] J. A. Torkestani, "An energy-efficient topology construction algorithm for wireless sensor networks," *Computer Networks*, vol. 57, pp. 1714–1725, May 2013.

[14] J. He, S. Ji, Y. Pan, and Z. Cai, "Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks," *Theoretical Computer Science*, vol. 507, pp. 2–16, Oct. 2013.

[15] Z. Lin, H. Liu, X. Chu, Y. W. Leung, and I. Stojmenovic, "Constructing connected-dominating-set with maximum lifetime in cognitive radio networks," *IEEE Transactions on Computers*, 2013.

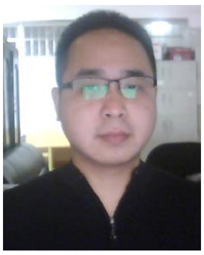
[16] J. Yu, N. Wang, and G. Wang, "Constructing minimum extended weakly-connected dominating sets for clustering in ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 72, pp. 35–47, Jan. 2012.

[17] A. Buchanan, J. S. Sung, V. Boginski, and S. Butenko. "On connected dominating sets of restricted diameter," *European Journal of Operational Research*, vol. 236, pp. 410-418, July 2014.

[18] W. Guo, B. Zhang, G. Chen, and X. Wang. "A pso-optimized minimum spanning tree-based topology control scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2, pp. 179-180, 2013.

[19] J. Qiao, S. Liu, and W. Duan, "A topology control algorithm for sensor networks based on robust optimization," *International Journal of Distributed Sensor Networks*, pp. 1-8, 2015.

[20] Y. Huang, J. F. Martínez, J. Sendra, and L. López. "Resilient wireless sensor networks using topology control: A review," *Sensors*, vol. 15, no. 10, pp. 24735-24770, 2015.



Qiang Tang is currently a lecturer at School of Computer and Communication Engineering in Changsha University of Science and Technology, Changsha, China. He received his Ph.D., M.S. and B.E. degrees from Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2010, 2007 and 2005, respectively. His main research interests include: wireless networks, smart grid.

Yuan-sheng Luo got his Ph.D degree in Computer Science and Technology from Xi'an Jiaotong University and M.S. and B.S. from Hunan University. He is now working at Changsha University of Science and Technology as a lecture. His main research interests now include: service computing, service composition, wireless networks, mobile computing and data science.

Ming-zhong Xie is currently a Master Degree Candidate at School of Computer and Communication Engineering in Changsha University of Science and Technology, Changsha, China. His main research interests include: wireless communications and smart grid.

Ping Li is currently a professor at School of Computer and Communication Engineering in Changsha University of Science and Technology, Changsha, China. He is an academic leader of Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation. He got his Ph.D. degree from the Hunan University. His main research interests include: network and information security, internet of things and sensor networks, data mining and big data processing.