ON THE EFFICIENCY OF ROUND-2: A SIMPLE ALGORITHM TO FURTHER MINIMIZE CONNECTED DOMINATING SET OF STATIC PROTOCOLS IN AD HOC NETWORKS

Yamin Li
Department of Computer Science, Hosei University, Tokyo, Japan
yamin@hosei.ac.jp

Keywords: Ad hoc networks, connected dominating set, node degree, distributed algorithms, and static protocols.

Abstract

This paper examines the efficiency of a simple distributed algorithm, called Round-2, for further decreasing the size of connected dominating set (CDS) in ad hoc networks and evaluates its performance based on neighbors-covered and neighbor-pair-connected algorithms under two categories of local static priority information: node-identifier-based and node-degree-based. The Round-2 algorithm decreases the size of CDS by fixing half forward nodes determined in the first round and re-evaluating the other half nodes. The neighbors-covered algorithms decrease the size of CDS by checking whether neighbors of a node are covered by intermediate nodes with higher priority than that node. The neighbor-pair-connected algorithms do it by checking whether two neighbors of a node are connected via intermediate nodes with higher priority. The performance of the Round-2 algorithm with different numbers of non-restricted intermediate nodes is simulated.

1. Introduction

An ad hoc network is a collection of wireless mobile nodes which dynamically form a temporary network without the use of any existing network infrastructure or centralized administration. The network link between two neighboring nodes is established via radio propagation. The neighboring nodes can communicate each other directly but the communication between non-neighboring nodes needs the help of intermediate other nodes. In such a case, the intermediate nodes operate not only as hosts but also as routers, forwarding the received packets.

Design of efficient routing and broadcasting protocols is one of the challenging tasks in ad hoc networks. Flooding is a simple broadcasting approach, in which each node that receives the packet first time forwards the packet. Blind flooding results in redundant transmissions: A node with d neighbors will receive the same packet d times. The redundant transmissions may cause the contention and collision. In a collision, multiple nodes send packets at the same time and the sent packets are lost duo to the radio wave interference. In a contention, a node backs off when the channel is occupied.

Sending packets consumes more energy than receiving packets. In order to reduce the probability of causing the contention and collision, as well as to reduce the energy consumption, it is desirable to reduce the number of nodes which will act as routers. The connected dominating set (CDS) based protocols are promising approaches to relieving the contention and collision problems. A subset of nodes in a network is a dominating set if a node in the network is either in the set or a neighbor to a node in the set. A node in CDS is called forward node, acting as both a host and a router. A node which is not in CDS is called non-forward node, acting as a host only.

An algorithm which minimizes the CDS is called a distributed algorithm if it uses only the local information [2]. A non-distributed algorithm uses global information such as the total number of nodes in network and the neighboring nodes of each node [5, 6].

Local information can be divided into static information and dynamic information. The static information includes neighborhood topology and certain node attribute that servers as priority value. The static information can be collected by periodically exchanging “Hello” messages among neighbors. The dynamic information includes a small set of nodes that have forwarded the broadcast packet that depends on every broadcast operation.

The localized protocols based on static information are called static protocols. Distributed algorithms of static protocols for determining CDS can be divided into two categories: neighbors-covered and neighbor-pair-connected. Neighbors-covered algorithms check whether all neighbors of a node v are covered by other intermediate nodes with higher priority than v. Neighbor-pair-connected algorithms check whether two nodes of all neighbor pairs of a node are connected via other intermediate nodes with higher priority. If the intermediate nodes are restricted to be neighbors of node v, the algorithm is called restricted algorithm; otherwise, it is a non-restricted algorithm.

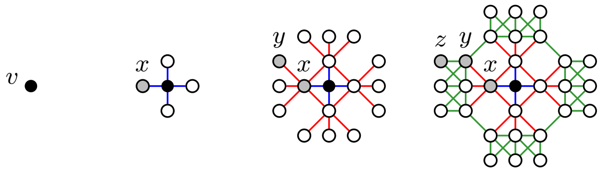
This paper focuses on localized static protocols. We first describe a general neighbor-pair-connected algorithm and a Round-2 algorithm for further reducing the size of CDS which is determined by the general algorithm, and then examine the efficiency of the algorithms. We use node identifier and node degree as the priority information to evaluate the efficiency of the neighbors-covered, neighbor-pair-connected, and Round-2 algorithms at different numbers of the intermediate nodes.

The rest of the paper is organized as follows. Section 2 describes general distributed algorithms of the static protocols and the Round-2 algorithm. Section 3 gives complete simulation results for evaluating the efficiency of the algorithms. And Section 4 concludes the paper.

2. Algorithms of Localized Static Protocols

We consider an ad hoc network as a graph $G = (V, E)$, where V is a set of nodes and E is a set of bidirectional links. For each node v, $N(v) = \{u|(u, v) \in E\}$ denotes its neighbor set. Let $F \subset V$. We say F is a CDS if F is connected and $V - F \subset N(F)$, where $N(F) = \cup_{v \in F} N(v)$. The key issue on designing a distributed algorithm for broadcasting or routing on wireless ad hoc networks is to determine a set of forward nodes with its size as being small as possible, based on affordable local information.

We use "k-hop" to denote the topology information that can be collected after k-round "Hello" message exchanges. Figure 1 shows neighborhood topologies with different values of k. In general, if the id of node u is a k-hop information of node v, the node degree of node u is a $(k+1)$-hop information of node v, one more than node id.



(a) 0-hop: v's id

(b) 1-hop: x's id or v's degree

(c) 2-hop: y's id or x's degree

(d) 3-hop: z's id or y's degree

Figure 1. Hop information viewed by node v (black node)

In previously known algorithms that select a set of forward nodes, for each node v in the network, all pairs of neighbors of v are checked in order to determine its forwarding status. Node v is marked as forward node if it has two neighbors that are not connected directly. They differ in the ways of pruning techniques that are used to reduce the number of forward nodes.

In Wu and Li's algorithm, two pruning rules are used to reduce the size of the resultant CDS [2]. In Rule-1, a forward node becomes non-forward if all of its neighbors are also neighbors of another node that has higher priority value. In Rule-2, a forward node can be non-forward if its neighbor set is covered by two other nodes that are directly connected and have higher priority values. Dai and Wu extended the Wu and Li's algorithm by using a more general rule called Rule-k in which a forward node becomes non-forward if its neighbor set is covered by k other nodes that are connected and have higher priority values [1, 2].

Chen et al. proposed an algorithm, called Span, to construct a set of forward nodes, called coordinators [3]. Dai and Wu revised Span algorithm to an enhanced version [2]: A node v becomes a coordinator if it has two neighbors that cannot reach each other by either directly connected, indirectly connected via one intermediate coordinator, or indirectly connected via two intermediate coordinators. Span algorithm requires 3-hop information if node id is used as the priority information or 4-hop information if node degree is used as the priority information.

Rieck et al. [4] proposed an algorithm that can be viewed as a special case of Span. In Rieck's algorithm, a node v is a forward node if it has two neighbors that cannot reach each other by either directly connected or indirectly connected via one intermediate node with higher priority than v. Rieck's algorithm requires only 2-hop information if node id is used as the priority information. It requires 3-hop information if node degree is used as priority information.

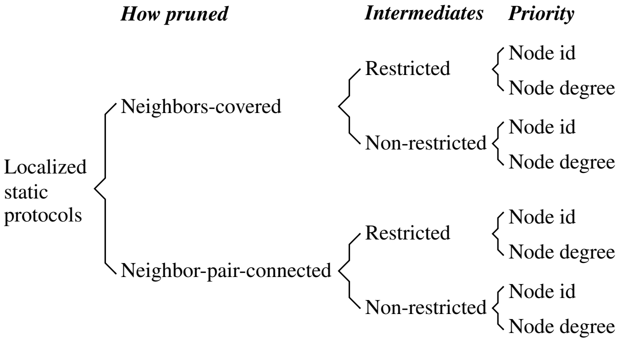


Figure 2. Categories of static protocols

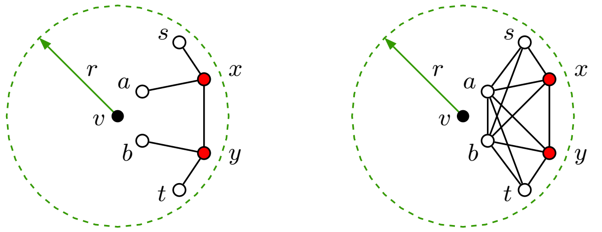
We divide the distributed algorithms of localized static protocols into two general categories: neighbors-covered and neighbor-pair-connected, as shown as in Figure 2, and describe the general algorithms as follows.

Neighbors-covered algorithm: A node v is in the CDS if it has a neighbor that is not covered by k connected intermediate nodes with higher priorities than v. In other words, node v is a non-forward node if its all neighbors are covered by k connected intermediate nodes with higher priorities than v.

Neighbor-pair-connected algorithm: A node v is in the CDS if it has two neighbors that are not directly connected or indirectly connected via k connected intermediate nodes with higher priorities than v.

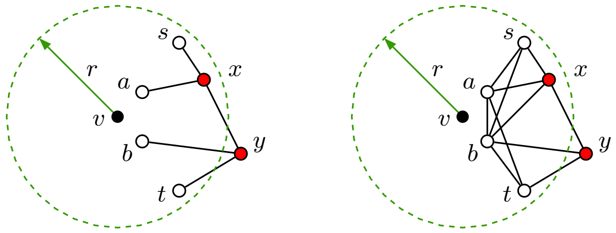
Restricted or non-restricted algorithm: If all the higher priority intermediate nodes mentioned above must be the neighbors of node v, the algorithm is said to be a restricted algorithm; otherwise, it is a non-restricted algorithm.

Figure 3 shows some algorithm examples. The dotted circle is the range the radio wave of node v can reach. r denotes the radius of the radio wave, thus all the other nodes inside the circle are the neighbors of node v. Figure 3(a) shows the restricted neighbors-covered algorithm. Node v can be removed from CDS because its neighbors are covered by nodes x and y which are neighbors of v and are connected, and have higher priority than node v. Figure 3(b) shows the restricted neighbor-pair-connected algorithm. The two nodes in each pair of v's neighbors are directly connected, or indirectly connected via one node x or y or two nodes x and y which are inside the circle and have higher priority than node v. Figures 3(c) and 3(e) show the non-restricted neighbors-covered algorithm. Compared to Figure 3(a), one or two intermediate nodes are not the neighbors of node v. Even in such cases, node v can be removed from CDS. Figures 3(d) and 3(f) show the non-restricted neighbor-pair-connected algorithm. Compared to Figure 3(b), one or two intermediate nodes are not the neighbors of node v. In both cases, node v can be removed from CDS.



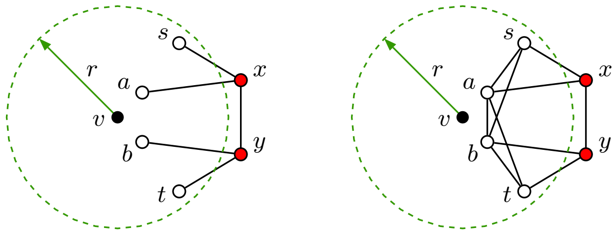
(a) Neighbors are covered restrictedly

(b) Two neighbors are connected restrictedly



(c) Neighbors are covered non-restrictedly

(d) Two neighbors are connected non-restrictedly



(e) Neighbors are covered non-restrictedly

(f) Two neighbors are connected non-restrictedly

Figure 3. Algorithm examples (nodes x and y have higher priority than v and v will be a non-forward node)

Priority: The node identifier (id) or the node degree (the number of its neighbors) can be used as the priority value in the algorithms described above. Note that once the priority is determined (either id or degree), it is not allowed to change the rule during the pruning process.

Rule-1, Rule-2, and Rule-k are the neighbors-covered algorithms. Rieck's and Span algorithms are special cases of the neighbor-pair-connected algorithm with $k = 1$ and 2, respectively, and the intermediate nodes are non-restricted.

We can revise the algorithms by considering the following matter. If a node x has a leaf neighbor, it can be marked as a forward node directly. This information can be sent to its neighbors. When a neighbor does self-pruning, the node x can be used as an intermediate node, in spite of its id or degree. That is, the priority of x is $+\infty$.

Round-2 algorithm: After getting a CDS by any algorithm, Round-2 algorithm lets the forward nodes with odd id be the final forward nodes and re-evaluates the forward nodes with even id by the following rule: The forward nodes with odd id determined by neighbors-covered or neighbor-pair-connected algorithm can be used as the intermediate nodes in the second round pruning process, performing neighbors-covered or neighbor-pair-connected algorithm on the nodes with even id.

3. Performance Analysis and Simulations

This section gives the simulation results that show the efficiency of the distributed algorithms of localized static protocols presented in the previous section. Table 1 lists the simulation parameters. The nodes are placed in a random position inside the network area.

Table 1. Simulation parameters

Parameters	Values
Network area	$2000 \times 2000\text{m}^2$
Number of nodes (n)	100, 150, 200, 250, 300, 350, and 400
Number of intermediate nodes (k)	1, 2, 3, and 4
Radio transmission ranges (radius r)	250m, 300m, 350m, 400m, and 450m
Number of trials	1000

The average numbers of leaves are listed in Table 2. From the table, we can see that the number of leaves becomes smaller when the number of nodes in network or the transmission range becomes larger.

Table 2. The number of leaves in network

Nodes	250m	300m	350m	400m	450m
100	3.780	1.490	0.526	0.181	0.073
150	1.521	0.376	0.108	0.029	0.004
200	0.604	0.111	0.027	0.005	0.000
250	0.242	0.036	0.003	0.002	0.001
300	0.112	0.018	0.004	0.000	0.000
350	0.044	0.004	0.000	0.000	0.000
400	0.023	0.004	0.000	0.000	0.000

Table 3 shows the average node degrees. As the number of nodes or the transmission range increases, the node degree also increases.

Table 3. The average node degrees in network

Nodes	250m	300m	350m	400m	450m
100	4.652	6.418	8.527	10.81	13.34
150	6.953	9.713	12.87	16.31	20.09
200	9.324	13.00	17.14	21.71	26.75
250	11.70	16.26	21.46	27.22	33.54
300	14.07	19.55	25.81	32.67	40.25
350	16.40	22.84	30.16	38.20	47.07
400	18.79	26.10	34.45	43.64	53.77

Figure 4 shows the efficiency of non-restricted neighbors-covered algorithms at transmission range of 250m. Label "Cov$k$id" means that the node id is used as the priority information where k is the number of intermediate nodes with higher priority. Label "Cov$k$de" means that the node degree is used as the priority information. We can see that increasing k from 1 to 2 improves performance greatly with the overhead of increase in computational complex, but by further increasing k from 2 to 3 or 4, the performance improvement is not obvious.

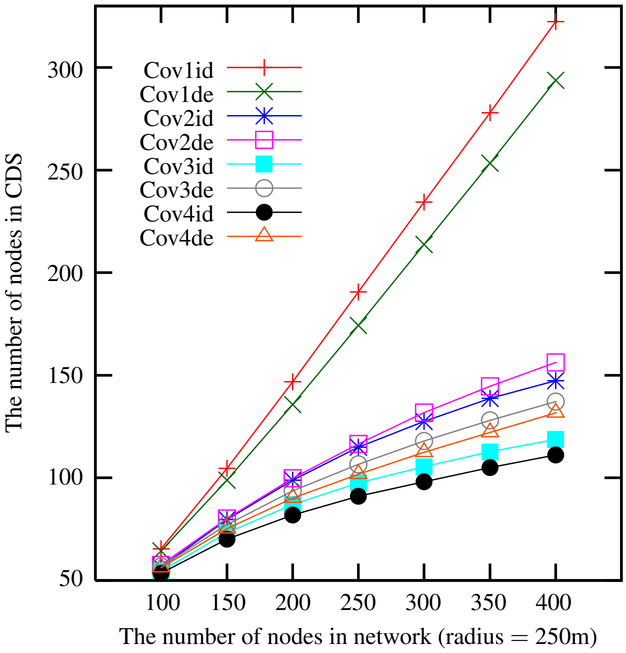


Figure 4. CDS size versus network size with neighbors-covered algorithms

Figure 5 shows another plot on the performance of neighbors-covered algorithms from where we can see that using node degree as the priority information gets better performance only when $k = 1$.

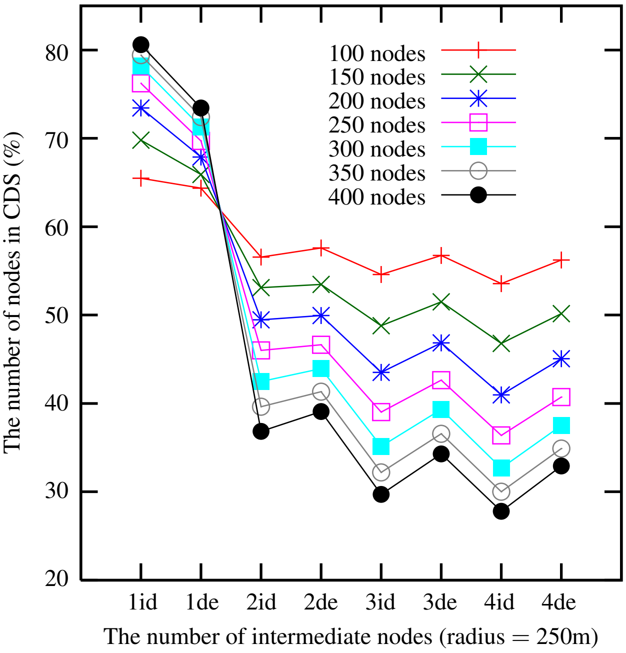


Figure 5. Relative CDS size versus number of intermediate nodes with neighbors-covered algorithms

Figure 6 shows the efficiency of non-restricted neighbor-pair-connected algorithms at transmission range of 250m. Increasing k from 1 to 2 improves performance greatly. Compared to Figure 4, we can also see that the neighbor-pair-connected algorithms have a little bit better performance than neighbors-covered algorithms.

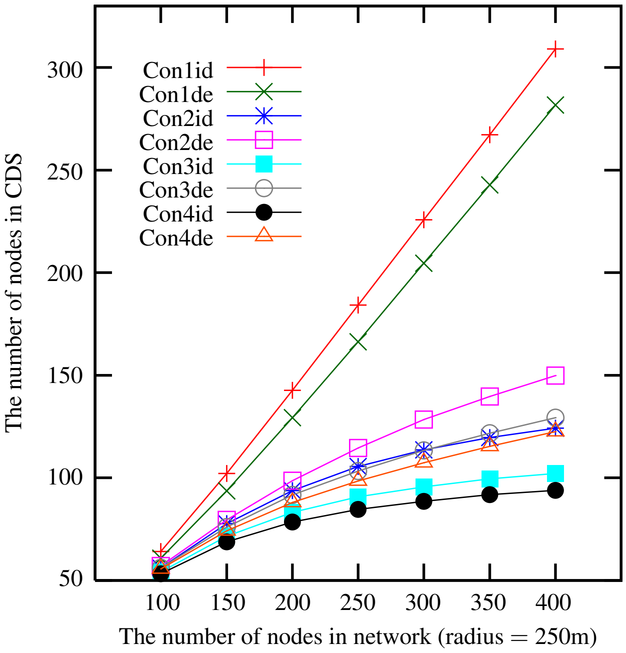


Figure 6. CDS size versus network size with neighbor-pair-connected algorithms

Figure 7 shows another plot on the performance of neighbor-pair-connected algorithms from where we can also see that using node degree as the priority information gets better performance only when $k = 1$.

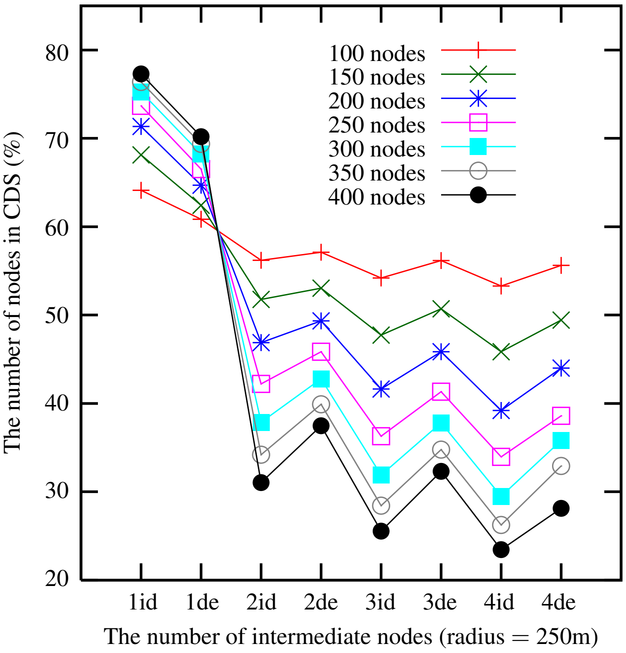


Figure 7. Relative CDS size versus number of intermediate nodes with neighbor-pair-connected algorithms

Figure 8 shows the performance improvement of the non-restricted algorithms compared to restricted algorithms when the node id is used as priority information. The improvements are not big.

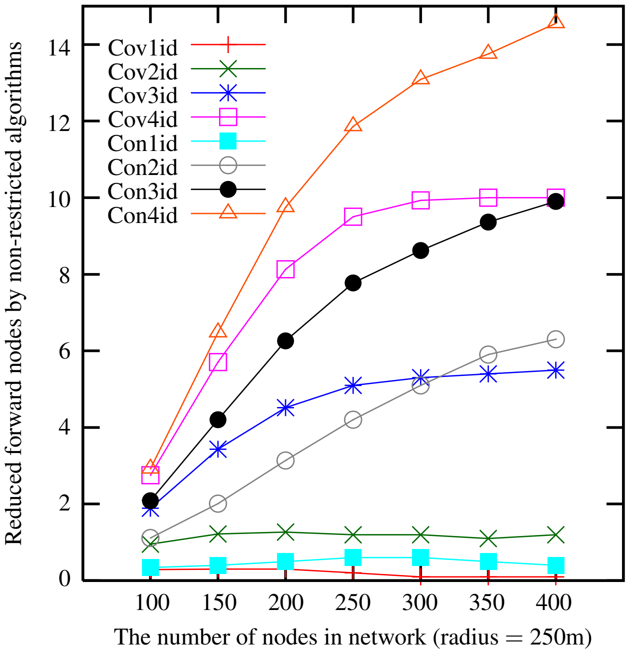


Figure 8. The reduced CDS size versus network size with node id priority information

Figure 9 shows the effect of the radio wave transmission range when the node id is used as priority information. The number of nodes in network is 400. As the transmission range increases, the size of CDS becomes smaller, because the number of neighbors increases. This is, a node in CDS can connect or cover more nodes.

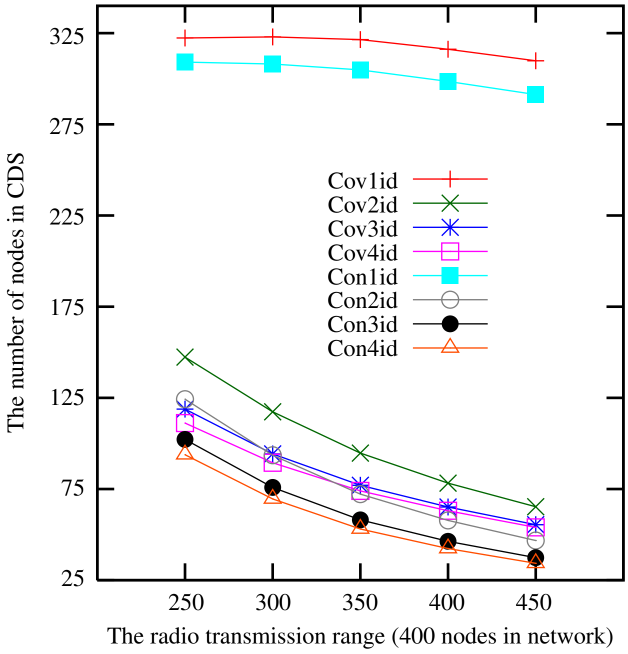


Figure 9. CDS size versus transmission range with node id priority information

Figure 10 shows the performance improvement of the non-restricted algorithms compared to restricted algorithms when the node degree is used as priority information. The improvements are also not big.

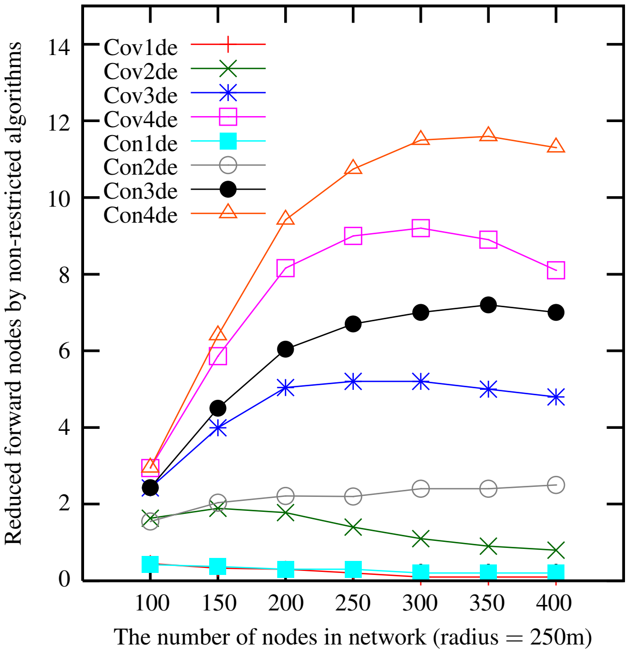


Figure 10. The reduced CDS size versus network size with node degree priority information

Figure 11 shows the effect of the radio wave transmission range when the node degree is used as priority information. Compared to Figure 9, we know that using node degree as the priority information gets better performance only when $k = 1$. For $k > 1$, node id should be used as the priority information.

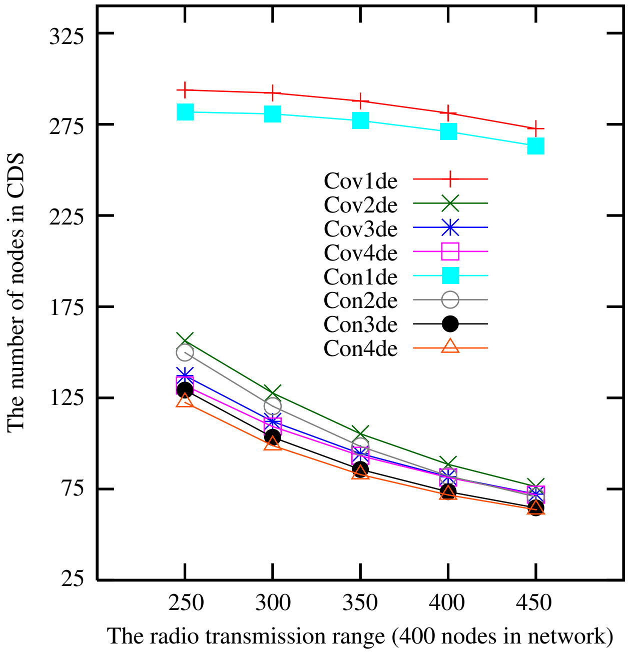


Figure 11. CDS size versus transmission range with node degree priority information

Figure 12 illustrates the performance improvement achieved by the Round-2 neighbors-covered algorithms with non-restricted intermediate nodes when node id is used as priority information. The labels containing “R2” indicate the Round-2 algorithms.

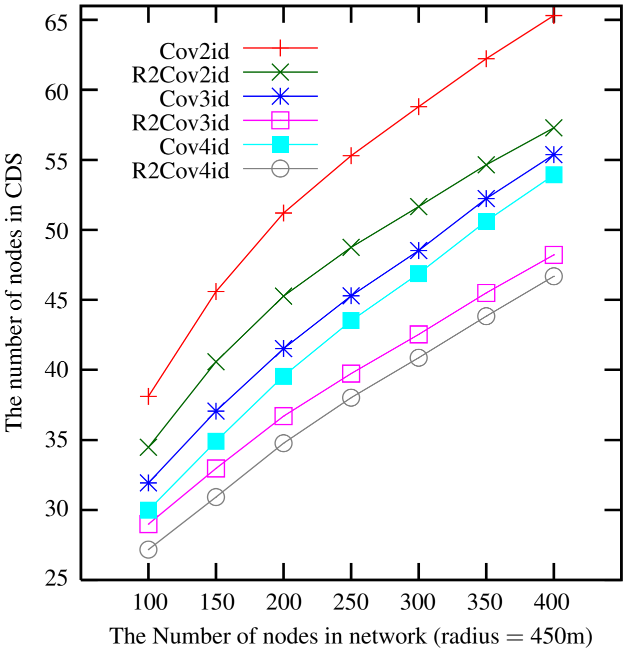


Figure 12. CDS size versus network size with neighbors-covered Round-1 and Round-2 algorithms

Figure 13 illustrates the performance improvement achieved by the Round-2 neighbor-pair-connected algorithms when node id is used as priority information. Compared to Figure 12, we can see that Round-2 neighbor-pair-connected algorithms get better performance than Round-2 neighbors-covered algorithms.

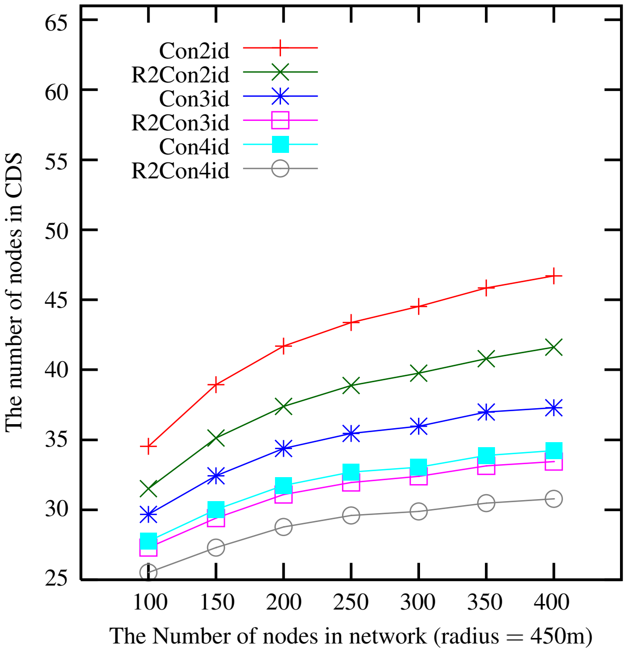


Figure 13. CDS size versus network size with neighbor-pair-connected Round-1 and Round-2 algorithms

The performance of Round-2 algorithms with $k = 3$ (R2Cov3id or R2Con3id in Figure 12 or 13) is even better than that of Round-1 algorithm with $k = 4$ (Cov4id or Con4id). Their time complexities are 2 times and d times, respectively, of the complexity of Round-1 algorithm with $k = 3$, where d is the node degree which is much bigger than 2, as shown as in Table 3.

4. Concluding Remarks

In this paper, we presented a general distributed neighbor-pair-connected algorithm and a Round-2 algorithm, and showed the efficiency of using node id and node degree as the priory information in both the neighbors-covered and neighbor-pair-connected algorithms for decreasing the size of CDS in ad hoc networks. Through the simulation, we knew that the size of the CDS was decreased obviously when the number of intermediate nodes k changes to more than one with the overhead of the increase in the computational complexity. When the simple $k = 1$ pruning algorithm is adopt, we suggest to use the node degree as the priority information; when $k > 1$, use the node id as the priority information. The non-restricted algorithm has better performance than that of the restricted one, but the improvement is not obvious. In Round-2 algorithm, only half of forward nodes determined by Round-1 algorithm, e.g., the nodes with even identifiers, perform the second round pruning process. It improves the performance at the overhead of the 2 times computational complexity compared to that of the Round-1 algorithms.

References

[1] Dai, F. and Wu, J., “Distributed Dominant Pruning in Ad Hoc Wireless Networks”, Proc. of IEEE Int’l Conf. on Communications, 2003, pp. 353−357.

[2] Dai, F. and Wu, J., “Performance Analysis of Broadcast Protocols in Ad Hoc Networks Based on Self-Pruning”, IEEE Trans. Parallel and Distributed Systems, 2004, 15, (11), pp. 1−13.

[3] Chen, B., Jamieson, K., Balakrishnan, H., and Morris, R., “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks”, Wireless Networks, 2002, 8, (5), pp. 481−494.

[4] Rieck, M. Q., Pai, S., and Dhar, S., “Distributed Routing Algorithms for Wireless Ad hoc Networks Using d-hop Connected Dominating Sets”, Proc. Int’l Conf. High Performance Computing in Asia Pacific Region, 2002, pp. 443−450.

[5] Velummylum, N. and Meghanathan, N., “On the Utilization of ID-based Connected Dominating Sets for Mobile Ad hoc Networks”, Int’l J. of Advanced Research in Computer Science, 2010, 1, (3), pp. 36−43.

[6] Meghanathan, N., and Dasari, I., “Performance Comparison Study of Connected Dominating Set Algorithms for Mobile Ad hoc Networks under Different Mobility Models”, Int’l J. of Combinatorial Optimization Problems and Informatics, 2013, 4, (2), pp. 12−30.