A Localized Algorithm for Reducing the Size of Dominating Set in Mobile Ad Hoc Networks

Yamin Li and Shietung Peng
Department of Computer Science
Hosei University
Tokyo 184-8584 Japan
{yamin, speng}@k.hosei.ac.jp

Wanming Chu
Department of Computer Hardware
University of Aizu
Aizu-Wakamatsu 965-8580 Japan
w-chu@u-aizu.ac.jp

Abstract

Connected dominating set based routing is a promising approach for enhancing the routing efficiency in the mobile ad hoc networks. However, finding the minimum dominating set in an arbitrary graph is an NP-complete problem. Restricted Rule k is a localized algorithm for finding a small dominating set in mobile ad hoc networks. It starts with a large initial dominating set and uses the local information of all neighbors of a node to attempt to remove the node from the set. In this paper, instead of using a large initial set, we use a rather small number of nodes as the initial set and then reduce the size of the set with Rule k algorithm. The small initial set is generated by a distributed NLogN algorithm with lower computational complexity than Rule k. The simulation results show that using the small set generated with NLogN algorithm as the initial set can make Rule k algorithm to achieve better performance.

1. Introduction

Mobile ad hoc network (MANET) is an autonomous system of mobile nodes connected by wireless links, where the link between two neighboring nodes is established via radio propagation. Neighboring nodes can communicate directly when they are within transmission range of each other. Each node in MANET operates not only as a host, but also as a router to forward packets.

Design of efficient broadcasting and routing protocols is one of the challenging tasks in ad hoc networks. The simplest method is blind flooding: Every node forwards the message received for the first time to its neighbors. This redundant broadcast not only wastes the resources of mobile nodes but also causes serious contention and collision problems. One way to avoid flooding is to use connected dominating set (CDS) based routing.

A subset of vertices in a graph is a *dominating set* if every vertex not in the subset is adjacent to at least one vertex in the subset. The main advantage of dominating set based approach is that it simplifies the broadcasting or routing process to the one in a smaller subnetwork generated from the CDS. Only the dominating vertices, called *forwarding nodes*, need to be active.

The efficiency of dominating set based approach depends largely on the time complexity for finding and maintaining a CDS and the size of the corresponding subnetwork. It is desirable to find a small CDS without compromising the functionality, reliability, and efficiency of an ad hoc network. Moreover, the algorithm for constructing the CDS should be efficient, distributed, and based on local information only. Since finding a minimum CDS for most graphs is NP-complete, efficient approximation algorithms are used to find a CDS of small size.

There are many existing algorithms in the literature for broadcasting/routing in ad hoc networks using dominating set based approach or its extensions [2, 3, 4, 5, 8, 9, 10, 11, 12]. These algorithms can be evaluated by the efficiency in terms of the number of forwarding nodes, reliability in terms of delivery ratio, and running time for selecting the set of forwarding nodes. Basagni et al. compared the performance of the CDS based localized protocols [1].

The algorithm ensures *full coverage* if the found dominating set is connected and the nodes that are not in the set connect to at least one node in the set. Some algorithms do not ensure the full coverage, the *span* algorithm [2] for instance. On the other hand, even an algorithm ensures the full coverage, it cannot ensure 100% delivery rate practically due to the contention and collision [5]. In general, if the number of forwarding nodes is large, there will be a rather high probability to cause contention and collision. In order to increase the delivery rate, the algorithm should try to reduce the size of the set of forwarding nodes.

In this paper, we first introduce an algorithm, *NLogN*, an extended localized algorithm of [7], for finding an almost CDS on ad hoc wireless networks and then show the performance of using it as the initial set to Rule k algorithm [3, 4].

The rest of the paper is organized as follows. Section 2 reviews the previous algorithms for selecting the set of forwarding nodes. Section 3 presents the NLogN algorithm. Section 4 gives simulation results on the performance of applying NLogN to Rule k algorithm and compares these results to that of Rule k algorithm. Section 5 concludes this paper.

2. Previous Work

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 $N(F) = \cup_{v \in F} N(v)$.

A broadcasting or routing algorithm is full coverage if the set of selected forwarding nodes is a CDS. The key issue on designing a distributed algorithm for broadcasting or routing on wireless ad hoc networks is to determine a set of forwarding nodes with its size as small as possible, based on affordable local information.

In previously known algorithms that select a set of forwarding 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 forwarding 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 forwarding nodes.

Chen et al. proposed an algorithm, called *Span* [2], to construct a set of forwarding nodes, called *coordinators*. 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 uses 3-hop information and cannot ensure a CDS.

Rieck et al. proposed an algorithm [10] that can be viewed as the enhanced Span. In Rieck's algorithm, a node v is a forwarding 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. Checking every pair requires $O(d^2)$ running time, where d is the maximum node degree of a network. Rieck's algorithm also checks an intermediate node that needs $O(d)$ running time. Therefore, the time complexity of Rieck's algorithm is $O(d^3)$,

Wu and Li [13] proposed a very simple distributed algorithm consisting of two local rules, the execution of which creates the desired CDS. The algorithm generates a large initial CDS: If a node v has two neighbors that are not neighbors themselves, then v enters the initial set. The algorithm is extremely simple but tends to create very large CDSs. The authors proposed two local rules, Rule 1 and Rule 2, to prune away unnecessary nodes.

- Rule 1: For every pair of nodes u and v in initial CDS, the one with the smaller ID, say v, can be removed from initial CDS if v and all its neighbors are covered by u, that is, they are u's neighbors.

- Rule 2: Assume nodes u, v, and w are in initial CDS, and assume that v's ID is the smallest of the three. Assume also that u and w are neighbors of v and are in each other's transmission range. If each neighbor of v is covered by u or w, then v can be removed from initial CDS.

Dai and Wu [3, 4] extended the Wu and Li's algorithm by using a more general rule called Rule k in which a forwarding node becomes non-forwarding if its neighbor set is covered by k other nodes that are connected and have higher priority values.

- Rule k: Assume that node u is in initial CDS and that a subset S of $\leq k$ neighbors of node u is such that

  1. S is contained in initial CDS,
  2. the subgraph spanned by S is connected,
  3. each node in S has an ID larger than u, and
  4. each neighbor of u is covered by the nodes in S.

  Then, u can be removed from initial CDS. Being the most general, Rule k prunes away the largest number of nodes. The Rule k is called *restricted Rule* k if every node in S is a neighbor of node u. In the following, we will use only the restricted Rule k.

Figure 1 shows an example marked by Rule k algorithm where the final CDS includes nodes 2, 4, 5, and 6. The neighbors of node 0 are covered by $S = \{2, 4, 5, 6\}$. So is node 1. The neighbors of node 3 are covered by $S = \{2, 5, 6\}$. Node 2 cannot be removed from CDS because there is no such S that can cover node 2's neighbors, nodes 4 and 5. Nodes 4, 5, and 6 are in the similar case as node 2.

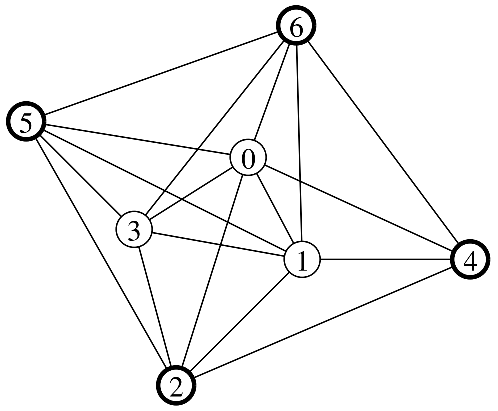


Figure 1. Forwarding nodes: 2, 4, 5, and 6

In our approach, instead of using a large initial set, we use a small initial set generated by NLogN algorithm and apply these rules to prune away unnecessary nodes. We will show that in the next section, nodes 2 and 4 can be marked as non-forwarding nodes by NLogN algorithm.

3. The NLogN Algorithm

This section describes the NLogN algorithm, an extended version of the localized algorithm for finding an almost CDS [7].

Full coverage of a broadcasting algorithm in ad hoc network can be achieved theoretically by selecting a CDS as the set of forwarding nodes. However, practically, the delivery ratio (the percentage of nodes that correctly receive a data packet) in most of cases is lower than 100% due to collision, contention, and mobility. Therefore, it is desirable to design a distributed broadcasting algorithm that is efficient in selecting a small set of forwarding nodes and the running time for the selection is fast although the set of selected forwarding nodes might not be a CDS with a very small probability. This is especially important for real-time applications.

The existing algorithms for deciding forwarding or non-forwarding status for a node v need to check every pair of neighboring nodes of v. If there is any pair of neighboring nodes of v that are not directly connected then v will be included in the initial set of the forwarding nodes. Therefore, the initially selected CDS might contain too many redundant nodes for forwarding the message in broadcasting or routing. Although some pruning techniques are used to reduce the size of the selected CDS in many algorithms, the overhead is high, especially when the size of the initially selected set is large.

For deciding forwarding or non-forwarding status for a node v, our algorithm does not check all pairs of v's neighbors. The number of pairs checked by the algorithm is $O(d \log d)$, where d is the maximum degree of nodes in the network.

In [6], the following strategy is used: If there is a cycle that connects all the neighbors of a node then the node is marked as non-forwarding since other nodes are still connected after removal of the node. This strategy leads to a very simple $O(d)$ time heuristic algorithm that checks whether the cycle exits or not for the set of its neighbors in a random order. However, the coverage rates of the algorithm from the simulations were not completely satisfied. For ad hoc networks with 40 - 200 nodes in 2000m $\times$ 2000m area, the coverage rates are between 97% and 99% in average.

To increase the coverage rate of the algorithm, we should increase the connectivity among the neighbors of a non-forwarding node. This leads to the algorithm in which for a node v, every neighbor of v checks log r other neighbors, where $r = deg(v)$ is the degree of node v. Node v is marked as non-forwarding if for every neighbor of v, all the log r neighbors checked have direct links or are connected via an intermediate node with higher priority than v. Intuitively, the log r connectivity for the neighbor set of a node should provide very high coverage rate of the algorithm after removing that node.

Our NLogN algorithm works as follow. The algorithm first provides a circular array of the set $N(v)$. Let v_i be the $(i+1)$th node in the array where $i = 0, \ldots, r-1$. The pair of nodes in $N(v)$, (v_i, v_j) is selected if $s = |i-j|$ is a power of 2. There are $r$log r pairs in total. If all selected pairs, (v_i, v_j), have a direct link or 2-hop link, where the 2-hop link means that v_i and v_j are connected via a node u that has a higher priority than v, then v is marked as forwarding node.

In order to reduce the size of forwarding node set, for each node, the marking process is repeated up to r times. Each time we shuffle the order of a node's neighbors. The distributed and localized NLogN algorithm is shown in Algorithm 1. We use my_id and my_degree to denote node v and $deg(v)$, respectively. In the algorithm, my_neighbor_id, an array of length $deg(v)$, stores the IDs of v's neighbors. The output of the algorithm is my_status that will be "forwarding" or "nonforwarding".

We show that nodes 2 and 4 in Figure 1 can become non-forwarding with our algorithm. Consider node $v = 2$: $r = deg(2) = 5$, $s = 1, 2, 4$, and node[2]_neighbor_id $= (v_0, v_1, v_2, v_3, v_4) = (0, 1, 3, 4, 5)$. For $s = 1$ as shown in Figure 2, node pair $(v_0, v_1) = (0, 1)$ are connected directly; node pair $(v_1, v_2) = (1, 3)$ are connected directly; node pair $(v_2, v_3) = (3, 4)$ are connected via node 6; node pair $(v_3, v_4) = (4, 5)$ are connected via node 6; and node pair $(v_4, v_0) = (5, 0)$ are connected directly;

For $s = 2$ as shown in Figure 3, all node pairs $(v_0, v_2) = (0, 3)$, $(v_1, v_3) = (1, 4)$, $(v_2, v_4) = (3, 5)$,

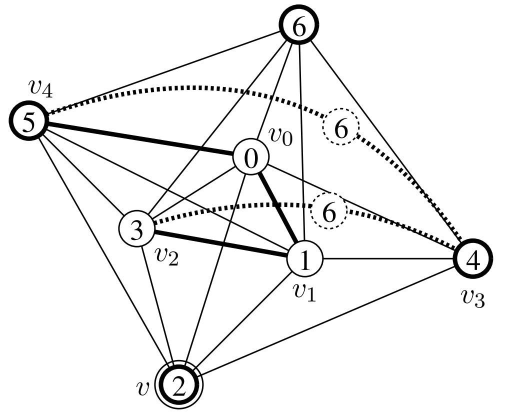


Figure 2. Removing node 2 from CDS (*s* = 1)

$(v_3, v_0) = (4, 0)$, and $(v_4, v_1) = (5, 1)$ are connected directly.

For $s = 4$, node pair $(v_0, v_4) = (0, 5)$ are connected directly; node pair $(v_1, v_0) = (1, 0)$ are connected directly; node pair $(v_2, v_1) = (3, 1)$ are connected directly; node pair $(v_3, v_2) = (4, 3)$ are connected via node 6; and node pair $(v_4, v_3) = (5, 4)$ are connected via node 6. Actually, this is the same as the case of $s = 1$. Since those links exist, we mark node 2 as a nonforwarding node.

Referring to Figure 4 and Table 1, node 4 can be also marked as a nonforwarding node. Note that Rule k algorithm marks these two nodes as forwarding nodes as we discussed in the previous section.

Table 1. Links for node 4

$s = 1$		$s = 2$	
Link	Connected	Link	Connected
(v_0, v_1)=(0,1)	Directly	(v_0, v_2)=(0,2)	Directly
(v_1, v_2)=(1,2)	Directly	(v_1, v_3)=(1,6)	Directly
(v_2, v_3)=(2,6)	Via node 5	(v_2, v_0)=(2,0)	Directly
(v_3, v_0)=(6,0)	Directly	(v_3, v_1)=(6,1)	Directly

The algorithm does not guarantee the fully coverage. However, our simulations show that the coverage rate is 0.998 or higher, and the size of the forwarding node set is reduced about 3% compared to Rule k algorithm. Figure 5 shows a sample ad hoc network of 100 nodes located in a 2000m $\times$ 2000m area (the transmission range is set to 250m). The yellow nodes and the blue nodes are forwarding and nonforward-

```
Algorithm 1: NLogN
begin
  my_status = nonforward;
  $r$ = my_degree;
  if $r > 1$
    found = false;
    $k = 0$;                                  /* r iterations */
    while ($k < r$) and (found = false)
      success = true;
      $s = 1$;                          /* are there log r links */
      while ($s \leq r$) and (success = true)
        $i = 0$;                      /* for each of my neighbors */
        while ($i < r$) and (success = true)
          $j = (i + s) \bmod r$;
          $x$ = my_neighbor_id[$i$];
          $y$ = my_neighbor_id[$j$];
          if (($(x, y) \notin E$) and ($\nexists z$ such that
            $z$.id > my_id and $(x, z) \in E$ and $(z, y) \in E$)
             success = false;     /* not success this ite. */
          endif
          $i$++;                              /* next neighbor */
        endwhile
        $s = 2s$;                                 /* next link */
      endwhile
      if (success = true)
        found = true;                                 /* exist */
      endif
      $k$++;                                 /* next iteration*/
      shuffling(my_neighbor_id);
    endwhile
    if (found = false)
      my_status = forwarding;              /* does not exist */
    endif
  endif
end
```

ing nodes, respectively, marked by our algorithm and original Rule k algorithm. The red nodes are nonforwarding nodes in our algorithm but forwarding nodes in original Rule k algorithm. The sizes of the set of forwarding nodes in the two algorithms are 59 and 65, respectively.

4. Performance Analysis and Simulations

We conducted a simulation study to compare the performance of NLogN + Rule k and original Rule k algorithm for broadcasting on wireless ad hoc networks. Our interests here are on evaluating efficiency

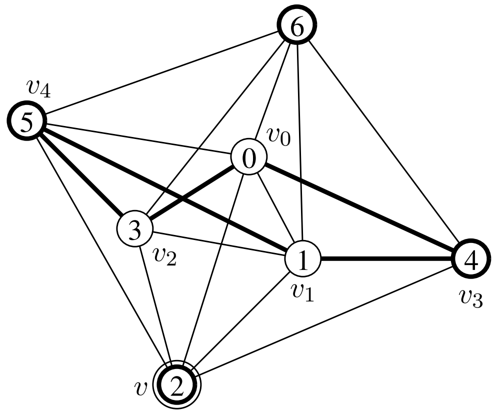


Figure 3. Removing node 2 from CDS (*s* = 2)

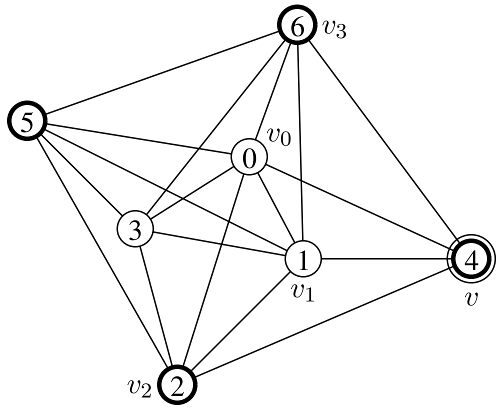
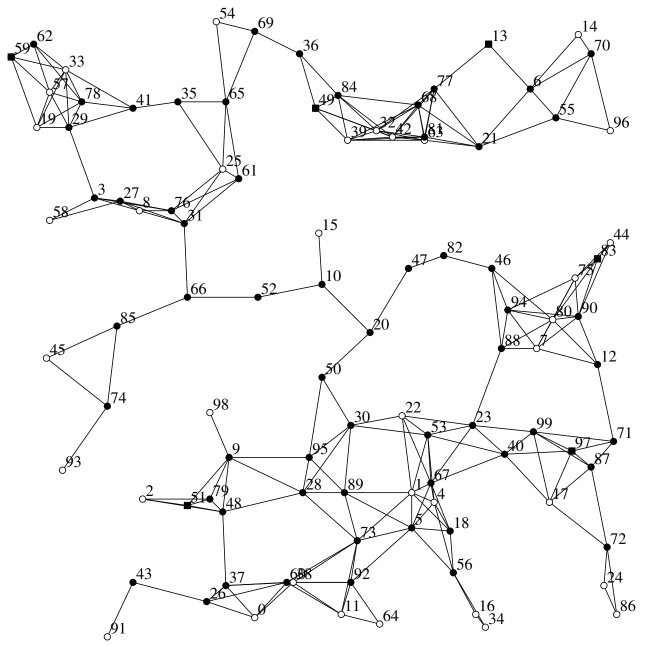


Figure 4. Removing node 4 from CDS

(the number of forwarding nodes), coverage rate (the percentage of the forwarding nodes forming a CDS), and redundancy (the number of packets received per node).

In addition to Rule k, we also tested Rule 1 + Rule 2 and Rule 1 + Rule 2 + Rule 3. Rule 3 is defined as following: Assume nodes u, v, x, and y are in initial CDS, and assume that v's ID is the smallest of the four. Assume also that u, x, and y are neighbors of v and are connected. If each neighbor of v is covered by u, x, or y, then v can be removed from initial CDS.

All simulations were conducted on static networks with a collision-free MAC layer. Each ad hoc network is generated by randomly placing n, $100 \leq n \leq 400$, nodes in a restricted 2000m $\times$ 2000m area. The transmission ranges are set to be 250m, 350m, and 450m.



- ● Forwarding node marked by both algorithms
- ○ Non-forwarding node marked by both algorithms
- ■ Non-forwarding node marked by our algorithm but forwarding node marked by original Rule *k* algorithm

Figure 5. The sets of forwarding nodes found by two algorithms on a sample ad hoc network

A wireless link is added between each pair of hosts that has a distance smaller than the given transmission range. For each configuration, we test 1,000 networks.

Figure 6 shows the number of forwarding nodes for randomly generated ad hoc networks of node ranges from 100 to 400, and the transmission range is set to be 300m. We compare the number of forwarding nodes of Rule 1 + Rule 2 and Rule 1 + Rule 2 + Rule 3 with the initial set generated with NLogN algorithm to that of original ones. From the figure, it is clear that using the initial set generated by NLogN algorithm results in a smaller number of forwarding nodes. For other transmission ranges (250m and 450m), the results are similar to that in Figure 6. Table 2 lists the details.

Table 3 gives the coverage rate, the percentage of the forwarding nodes forming a CDS. These are obtained by dividing the number of full coverages by the total number of trials. The worst case is that, in 10000 trials, there are only 3 times in which the forwarding nodes do not forward packets to all nodes in the network. We can see clearly from the simulation results that the coverage rates are higher than 99.96% in all cases in our simulations. We conclude that the set of forwarding nodes generated by our algorithm is *almost* a CDS practically.

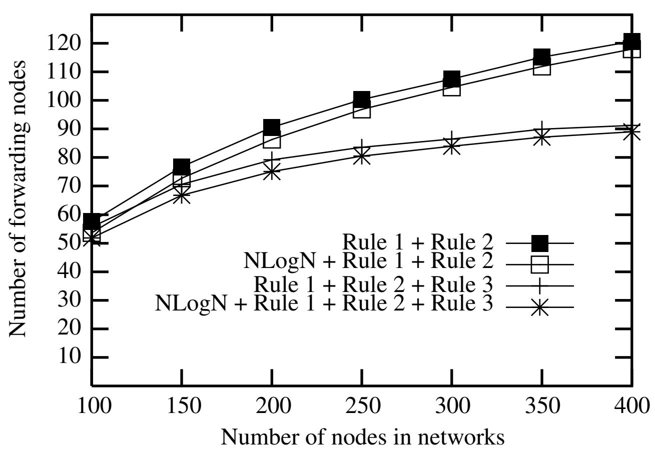


Figure 6. The number of forwarding nodes

Table 2. The number of forwarding nodes

r250m				
#nodes	R1+R2	NLogN+	R1+R2+R3	NLogN+
100	61.798	57.913	60.959	57.103
150	87.262	81.317	83.851	78.055
200	107.079	100.974	99.275	93.368
250	122.004	116.402	108.631	103.397
300	135.200	129.450	116.207	110.907
350	145.232	140.103	120.681	116.039
400	152.918	148.371	123.402	119.366
r350m				
#nodes	R1+R2	NLogN+	R1+R2+R3	NLogN+
100	51.712	48.700	48.426	45.546
150	67.064	63.471	59.044	55.666
200	75.217	72.419	62.529	59.953
250	82.443	79.725	65.540	63.108
300	89.299	86.032	68.942	65.963
350	93.474	90.781	69.659	67.265
400	98.463	95.575	71.562	69.004

Table 3. Rate of successful broadcasting

#nodes	r250m	r300m	r350m
100	99.9%	99.7%	99.8%
150	99.7%	99.7%	100.0%
200	100.0%	99.9%	99.9%
250	99.9%	100.0%	99.9%
300	100.0%	100.0%	100.0%
350	99.8%	99.9%	100.0%
400	100.0%	100.0%	100.0%

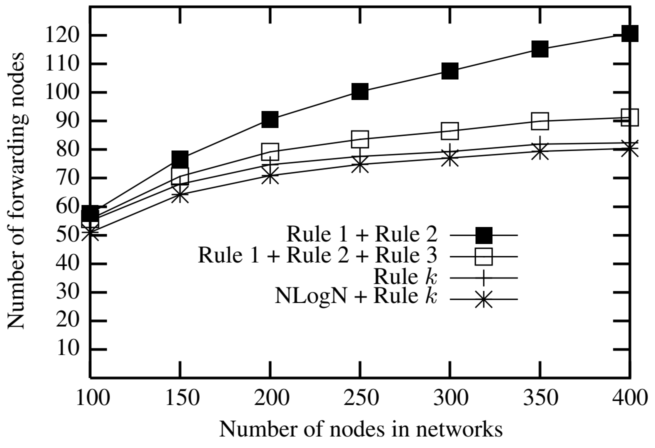


Figure 7. The number of forwarding nodes

Table 4. The number of forwarding nodes

#nodes	r250m		r350m	
	Rule k	NLogN+	Rule k	NLogN+
100	60.945	56.781	47.117	44.313
150	82.493	76.613	55.846	52.685
200	95.785	90.120	58.009	55.689
250	103.026	98.246	60.208	58.039
300	108.855	104.028	62.917	60.132
350	111.706	107.542	63.235	61.048
400	113.034	109.475	64.566	62.142

Figure 7 shows the number of forwarding nodes for randomly generated ad hoc networks of node ranges from 100 to 400, and the transmission range is 300m. We compare the number of forwarding nodes of Rule k with the initial set generated with NLogN algorithm to that of Rule 1 + Rule 2, Rule 1 + Rule 2 + Rule 3, and Rule k. For other transmission ranges (250m and 450m), the results are similar to that in Figure 7. Table 4 lists the details.

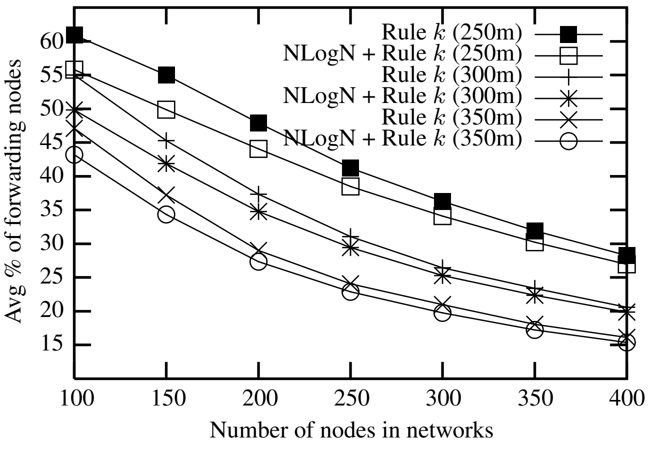


Figure 8. The number of forwarding nodes

Figure 8 compares the percentages of nodes in CDS for Rule k and Rule k with NLogN initial set under the transmission ranges of 250m, 300m, and 350m.

5. Concluding Remarks

In this paper, we introduced a distributed algorithm, NLogN, for finding an almost CDS on ad hoc wireless networks. The proposed algorithm improves the performance of the original Rule k algorithm. It is interesting to investigate the coverage rate of our algorithm using probability theory. For application consideration, to investigate the performance of the algorithm under dynamic environment with packet collision and node mobility is also worth further research.

References

[1] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. *IEEE Trans. Parallel and Distributed Systems*, 17(4):292–306, 2006.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, Sept. 2002.

[3] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. In *Proc. of IEEE International Conf. on Communications*, pages 353–357, 2003.

[4] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Trans. Parallel and Distributed Systems*, 15(10):908–920, 2004.

[5] F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Trans. Parallel and Distributed Systems*, 15(11):1–13, 2004.

[6] Y. Li, S. Peng, and W. Chu. An efficient distributed broadcasting algorithm for wireless ad hoc networks. In *Proc. of The Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 75–79, Dec. 2005.

[7] Y. Li, S. Peng, and W. Chu. An efficient algorithm for finding an almost connected dominating set of small size on wireless ad hoc networks. In *Proc. of The Third IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 199–205, Oct. 2006.

[8] W. Lou and J. Wu. On reducing broadcasting redundancy in ad hoc wireless networks. *IEEE Trans. Mobile Computing*, 1(2):111–123, 2002.

[9] E. Pagani and G.P. Rossi. Providing reliable and fault-tolerant broadcasting delivery in mobile ad hoc networks. *Mobile Networks and Applications*, 4:175–192, 1999.

[10] M. Q. Rieck, S. Pai, and S. Dhar. Distributed routing algorithms for wireless ad hoc networks using d-hop connected dominating sets. In *Proc. Int'l Conf. High Performance Computing in Asia Pacific Region*, Dec. 2002.

[11] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Trans. Parallel and Distributed Systems*, 13(1):14–25, 2002.

[12] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Trans. Computers*, 53(10):1343–1354, 2004.

[13] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 7–14, 1999.