A Reliable Virtual Backbone Scheme in Mobile Ad-Hoc Networks *

Manki Min   Feng Wang   Ding-Zhu Du
Dept. of Computer Science and Engineering
Universiry of Minnesota
Minneapolis, MN 55455, USA.
{min, fwang, dzd}@cs.umn.edu

Panos M. Pardalos
Dept. of Industrial and Systems Engineering
University of Florida
Gainesville, FL 32611, USA.
pardalos@ufl.edu

Abstract

In wireless ad-hoc networks, hosts communicate each other without help of any physical infrastructure. Inevitably the communication tends to be inefficient in terms of computational and network resources. Study on virtual infrastructures or backbones in wireless ad-hoc networks gets more attention in the hope of reducing the communication overhead. But the backbone structure is very vulnerable due to various factors like node mobility and unstable links, and so on. So a new scheme which is reliable and efficient both to construct and maintain the backbone structure is needed. In this paper, we present our noble virtual backbone scheme which is reliable and efficient by considering stability and coverage of nodes.

1. Introduction

Communications in wireless ad-hoc networks assume that there is no physical infrastructure. This assumption not only increases the communication cost, but also leads to a severe problem, known as *broadcast storm problem* [13], induced by flooding inherent in on-demand routing schemes. Recently many researchers proposed virtual backbone schemes which are inspired by physical backbone to maximize resource utilization and to minimize damage caused by flooding. Virtual backbones can be used to:

1. collect topology information for routing

2. provide a backup route

3. multicast or broadcast messages

Among them, most research on virtual backbones are focused on their applications to routing schemes. One of such works can be found in [14] and their simulation results show that DSR/AODV over virtual backbones (*core*) performs better than plain DSR/AODV.

Constructing and maintaining virtual backbones impose another control overhead onto the overall communication, so the constructed backbone size should be as small as possible. The roles of virtual backbones require connectivity of nodes and hence a minimum connected dominating set can make a good candidate. In this paper, we assume that every node has the same transmission range so that we can model the network topology using unit-disk graphs, UDG in short. Unfortunately finding a minimum connected dominating set, MCDS in short, in UDG is known to be NP-hard [7] and its approximation is studied for backbones.

In a mobile ad-hoc network (MANET), topology of nodes is dynamic due to node mobility as well as instability of links. As a result, the reliability and maintenance cost of the backbone heavily depends on node mobility. To our best knowledge, previously published backbone schemes didn't consider node mobility in the backbone construction and hence the structure is very fragile.

In this paper, we propose a noble backbone scheme, *CMIS*, in MANET which is more reliable and more efficient than previously proposed schemes. We approach the problem based on four factors: small backbone size, efficiency of construction and maintenance, and reliability. *CMIS* generates a backbone guaranteed to have small size (at most eight times of $|MCDS|$) using low message (time) complexity of $O(n\Delta)$ ($O(n)$). For reliability and efficient maintenance, we have to consider node mobility since it affects the topology of network. We tackle the node mobility by considering two factors: stability and coverage of nodes. Selection of nodes with bigger stability and coverage resulted in at least 88% (32%) increases of average connectivity (DS) lifetime of the backbone against equivalent size backbone schemes. And the increases of average backbone lifetime becomes more dramatic against other schemes. We define the connectivity and DS lifetime of a backbone by the first time that the backbone gets disconnected and an

uncovered node appears, respectively.

The rest of this paper is organized as follows: We survey some of existing backbone schemes in section 2. Section 3 explains our backbone scheme *CMIS* and proves its performance ratio and message and time complexity. Finally, simulation results indicating the reliability and the maintenance efficiency of *CMIS* are provided in section 4 and section 5 concludes this paper.

2. Backbone Schemes

In this section, we briefly survey several schemes for virtual backbones in wireless ad-hoc networks. Again, the size of constructed backbone affects the overall overhead when the backbone is used with other protocols, like routing, so we consider the size of the backbone as one of performance. Hereafter, we use *performance ratio* to denote the ratio of the worst case size of the constructed backbone to the size of MCDS of the given graph topology ($\frac{|CDS|}{|MCDS|}$).

2.1. Non-constant ratio schemes

Recent works [9, 17, 15, 5] have non-constant performance ratios and message complexities of at least $O(n \log n)$ and time complexities of at least $O(n)$.

Das *et al*'s scheme first finds a dominating set D and then connects the nodes in D. Nodes with the most 2-hop neighbors which are not in D are added to D. And they grow one fragment to the CDS C by selecting one or two-hop paths between the fragment node and nodes outside the fragment which covers the most nodes not in C. Their scheme is based on [11]. This scheme has performance ratio of $O(H(\Delta))$ which is $\Theta(\log n)$.

Wu and Li proposed a scheme that first directly finds a CDS C and then removes redundant nodes. They proposed a simple and efficient *marking process* for the formation of C in which every node simply exchanges its neighbor list and adds itself to C if it has at least two unconnected neighbors. The nodes in C are called *gateway* nodes and the rest are called *non-gateway* nodes. Any gateway node determines itself to be redundant when its one or two adjacent gateway neighbors with larger IDs dominate all of its neighbors. But this approach may produce a poor result where the output CDS consists of all nodes while the MCDS consists of only 2 nodes under certain extreme cases. This scheme has performance ratio of $O(n)$.

Stojmenovic *et al*'s scheme is very similar to Wu and Li's scheme except that they use a triple $(degree, x, y)$ instead of $degree$. And this scheme also has performance ratio of $O(n)$. They also proposed its application to broadcasting, combined with *neighbor elimination*. By eliminating redundant rebroadcast, the communication overhead

was reduced significantly, but this approach requires neighborhood topology which may be achieved by GPS or other location technique.

One of the most important issue in virtual backbone schemes is the size of the constructed backbone since the control overhead on the backbone will be proportional to the backbone size. And in that context, these schemes may not fit into some environment. We summarize three nonconstant performance ratio schemes and move onto constant performance ratio schemes.

measure	[9]	[17]	[15]
Perf. ratio	$\Theta(\log n)$	$O(n)$	$O(n)$
time complexity	$O(n^2)$	$O(\Delta^3)^*$	$\Omega(n)$
MSG complexity	$O(n^2)$	$\Theta(m)^{**}$	$O(n^2)$

* Δ is the maximum degree in unit-disk graphs

** m is the number of edges in unit-disk graphs

2.2. Constant ratio schemes

Wan *et al*'s scheme has performance ratio of 8 and message complexity of $O(n \log n)$ [16]. Meanwhile, Cardei *et al* also proposed a scheme with performance ratio of 8 and message complexity of $O(n\Delta)$ [4]. Both schemes have time complexity of $O(n)$. And both schemes finds a *maximal independent set* (MIS in short) as a dominating set. Wan *et al* showed that an MIS has size of at most $4|C^*| + 1$, where C^* is an MCDS. This property plays a fundamental role in proving the constant performance ratio of 8 in both schemes. Both consist of two phases: (1) find an MIS, (2) transform the set into a CDS. Initially every node has white color and during the two phases, each node changes its color. MIS nodes change to black and the rest change to gray. And both require a leader election [6] phase before the first phase. The leader election phase will elect a leader node and generates a spanning tree rooted on the leader node.

The first phase of Wan *et al*'s scheme [16] determines the level of each node (the number of hops between itself and the root of the spanning tree which is constructed by the leader election procedure) and finds MIS nodes. The ranking they use is an ordering of level and ID pair. MIS nodes consist of independent nodes with higher ranking than their neighbors. And every pair of two complementary sets of MIS nodes are separated by exactly two hops. After an MIS is formed, *dominating tree* is constructed and this tree is rooted at the gray neighbor of the leader node with maximum black degree. They proved the scheme has performance ratio of 8 based on the property of MIS, stated above. Also they established the lower bound $\Omega(n \log n)$ on message complexity of distributed algorithms for CDS construction which supports the tightness of the message complexity of their scheme.

The first phase of Cardei *et al*'s scheme [4] is performed

in a similar pattern as in [16], but they use *effective degree* (number of white neighbors) instead of level. It requires that the effective degree of a node be broadcasted as many times as its degree. To connect the MIS found in the first phase, they use an approximation of Steiner tree. The second phase is based on the distributed depth-first search spanning tree algorithm and they find gray node with maximum black degree and mark them as interconnecting nodes. This scheme also has performance ratio of 8 based on the above property of MIS.

Recently a virtual backbone scheme with performance ratio of 12 and the backbone is constructed locally using low message complexity of $O(n)$ [1]. Their method reduces the message complexity to $O(n)$ by using only local information and thus significantly saves the construction and maintenance efforts.

3. Our scheme *CMIS*

An MIS is a dominating set by definition. So the simplest way to construct a CDS will be first to determine an MIS and then to interconnect the set. In this section we first present the simple version and describe the improved version which interlace the two phases.

Any pair of complementary subsets of an MIS are separated by either two or three hops. And we can easily find an MIS without such three-hop separations. This is an interesting property because with this property, interconnection of the MIS can be constructed easily. Now consider a tree T spanning the MIS B. Since B is an independent set, any pair of nodes in B can not be interconnected directly. On the other hand, due to the above property, we can always interconnect B using at most $|B|-1$ nodes. Wan *et al* showed that the size of an MIS is bounded by $4|MCDS|+1$ [16]. Also note that any node can have at most five independent neighbors in unit-disk graphs.

The MIS B by other constant ratio schemes in [16, 4] also have the above interesting properties. We need an initiator for the construction of MIS and we can determine an initiator by running leader-election algorithm [6]. This algorithm has time complexity of $O(n)$ and message complexity of $O(n \log n)$.

We define the ranking to be an ordering of (stability, coverage, id) of nodes. And we say that a node v with rank (s_v, c_v, id_v) has a higher order than a node u with rank (s_u, c_u, id_u) if:

1. $s_v > s_u$, or

2. $s_v = s_u$ and $c_v > c_u$, or

3. $s_v = s_u$ and $c_v = c_u$ and $id_v > id_u$

The stability and the coverage of each node can be estimated using location information and hence our scheme assumes

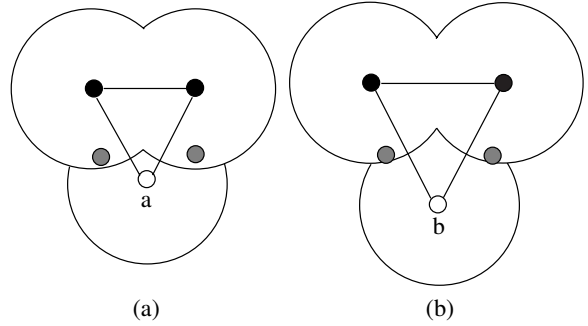that every node is equipped with a physical location device such as GPS.

Figure 1. Coverage gains of two nodes

We define the stability of a node to be the reciprocal of the average distance between the initial location and the current location during a time period. So the more a node moves, the less it is. And we define the coverage of a node to be the average distance between itself and its 1 or 2 hop backbone neighbors plus the average distance among its 1 or 2 hop backbone neighbors. This is an estimate for the coverage gain of a node for the backbone in case the node is selected as a backbone node. Here the coverage gain consists of two values: the existing coverage by nodes that are already in the backbone and the new coverage caused by the node. The reason for considering the existing coverage is to distribute the backbone nodes more uniformly so that they can cover the most no matter how high mobility the network has. It is computationally expensive to calculate the area of coverage (expressed as union of overlapped circles in UDG), so we estimate the coverage using average distance between a node and its 1 or 2 hop backbone neighbors. It is reasonable because the bigger the distance is, the bigger the coverage is. Figure 1 depicts an example comparing two nodes' coverage gains. The node a in (a) is closer to its backbone neighbors and also backbone nodes in (a) are closer to each other than the case of (b). Hence we can say the node b has a larger coverage gain than a.

In our schemes to be described in next two subsections, we assume that every node records its own location at every second during the period. And whenever a node selects a node from its neighbors, it chooses the one with the highest rank. Every message contains color, rank and locations of both the sender and the 1 hop backbone neighbors of the sender. Initially every node has white color and changes its color during the procedure. Note that in UDG, any node can have at most five independent neighbors, so the amount of the above additional information in the message is bounded. To simplify the description of our schemes, we use the following notations:

- $u \to v, [msg]$: a node u unicasts a message $[msg]$ to

its neighbor v with the highest rank.

- $u \leftarrow [msg]$: a node u receives a message $[msg]$.
- $u \Uparrow [msg]$: a node u broadcasts a message $[msg]$.

3.1. *MIS*

MIS algorithm:
1: *Initiator* $\rightarrow$ its white neighbor, [BLACKAC-CEPTED].
2: If a node $u \leftarrow$ [BLACKACCEPTED], u changes its color to black and $u \Uparrow$ [BLACKDONE].
3: If a white node $u \leftarrow$ [BLACKDONE], u changes its color to gray and $u \Uparrow$ [GREYDONE].
4: If a white node $u \leftarrow$ [GREYDONE], $u \Uparrow$ [BLACK].
5: If a white node $u \leftarrow$ [BLACK] and u has the highest rank among the senders of [BLACK] messages, u changes its color to black and $u \Uparrow$ [BLACKDONE].
6: If a white node u broadcasted [BLACK] and did not hear any [BLACK], u changes its color to black and $u \Uparrow$ [BLACKDONE].
7: Any node whose neighbors are all colored black or gray terminates the procedure.
8: The algorithm ends when every node terminates.

Lemma 1 *The resulting set B at the end of* MIS *forms a maximal independent set. Moreover for any $B' \subset B$, B' and $B \setminus B'$ are separated by exactly two hops.*

Proof. At every round, independent white neighbors are selected by gray nodes and are marked black. This implies that B forms an independent set. And since *MIS* is an incremental algorithm, it will end up with black or gray nodes only. Hence B is a maximal independent set. And *MIS* starts from a single black node and incrementally enlarges the black nodes set by adding black nodes 2 hops away from the previous black nodes set, so there cannot be a chance that any two complementary subsets of B are separated by three hops. $\square$

The next step is to interconnect the black nodes and we can use a distributed algorithm for MST [6] with a slight modification. *MIS* with MST algorithm has performance ratio of 8 and it has equivalent message and time overhead as other 8-performance MCDS algorithms in [16, 4].

3.2. *CMIS*

In this subsection, we improve the previous version *MIS* by interlacing the selection of interconnecting nodes into the construction of B.

In the previous scheme *MIS*, after B is constructed we used an algorithm based on MST to find interconnecting nodes. And in the interconnection algorithm, we chose nodes which are equivalent to 2-hop paths to interconnect two or more black nodes. But this approach requires significant message and time complexity. We can achieve the interconnection of the black nodes by letting a newly marked black node select a neighbor which interconnects itself to the nodes already marked black. That is, we can use a greedy set-cover algorithm [8] to select interconnecting nodes. Consider the i-th round of *MIS*. Let B_i and G_i be the set of nodes marked black and gray at the i-th round, respectively. Then each gray node in G_i must have at least one neighbor in B_i and zero or more neighbors in B_{i+1}. In other words, it is enough to find gray nodes which interconnects many B_i nodes at i-th round. Figure 2 explains this procedure. In the figure, we already know that every G_1 node is connected to the B_1 node and it is connected to some B_2 nodes if there are any B_2 node. So if every B_2 node selects its own interconnecting node in G_1 nodes, we can guarantee to form a tree (thick lines and nodes in the figure). R_1 in the figure denotes interconnecting nodes.
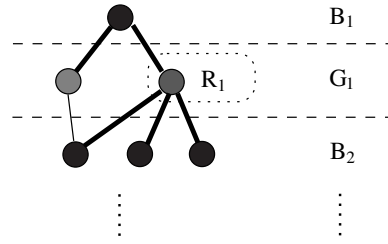

Figure 2. Interconnecting backbone nodes

Note that finding such nodes is equivalent to finding set-covers. Each gray node g in G_i can be represented as a subset X_g of B_{i+1} such that the node g is connected to the nodes in X_g. And we know that $max\{|X_g|\} \leq 4$ for any gray node g, since any node can have at most five independent neighbors in UDG. The greedy algorithm presented in [8] is an incremental approach such that at every iteration it selects the cover which covers the most uncovered elements. We can reduce the number of iterations to 2 as follows: Every node in B_i selects the highest ranked cover (gray neighbor with the highest rank) which covers itself. This will give a set-cover but there can be some redundant covers. Redundancy can be removed if every selected gray node knows the black neighbors of all the other selected gray nodes which covers the same node in B_i.

We call this scheme as *CMIS*. In *CMIS*, interconnections of black nodes are established in an interlaced fashion during the construction of an MIS. *CMIS* is based on *MIS* and needs a little modification.

CMIS requires each node to maintain a little bit more in-

formation than *MIS* does. In *CMIS* each gray node should maintain two lists b_1 and b_2 of black neighbors which are marked before and after the node is marked gray, respectively. As in *MIS*, each message contains local information of a node and [REDACCEPTED] contains b_1, b_2 of itself and those received.

CMIS algorithm:
1: Same as *MIS* algorithm for exchanges of messages other than [RED], [REDACCEPTED], [RED-DONE].
2: If a gray node u does not have any white neighbor, $u \Uparrow$ [RED].
3: If a black node $u \leftarrow$ [RED] and u does not have a red neighbor, $u \to$ its gray neighbor, [REDACCEPTED].
4: If a gray node $u \leftarrow$ [REDACCEPTED] and u is not redundant, u changes its color to red and $u \Uparrow$ [REDDONE].
5: Any black node which has at least one red neighbor and no white neighbors terminates the procedure.
6: Any node other than black terminates the procedure when all its neighbors are colored black, gray or red.
7: The algorithm ends when every node terminates.

CMIS will also find the same MIS B with a little bit more message and time complexities than *MIS*. Besides B, it will find the interconnections of B nodes, marked as red, right after each new black node is selected with only small number of messages. Since every new black node selects an interconnecting node, the resulting nodes will form a connected graph. Moreover since a new black node either already has a red neighbor or selects only one red neighbor (only nonredundant ones will be changed to red), cycle cannot be formed. Hence the resulting nodes form a tree. And at any i-th round, the nodes marked red at the i-th round are connected to nodes marked black at the $i - 1$-th round and the i-th round.

It is proven that the set-cover problem is NP-hard and its greedy algorithm with a pretty good performance ratio can be found in [8].

Lemma 2 *Let R be the set of nodes selected to interconnect B nodes and marked red by CMIS. And let C^* be an MCDS for the given set of nodes. Then $R \cup B$ is connected and $|R| \leq 4|C^*| - 1$.*

Proof. At each round, B nodes select a gray neighbor which is connected to another B nodes which was marked black in the previous round. Hence it is clear that $R \cup B$ is connected.
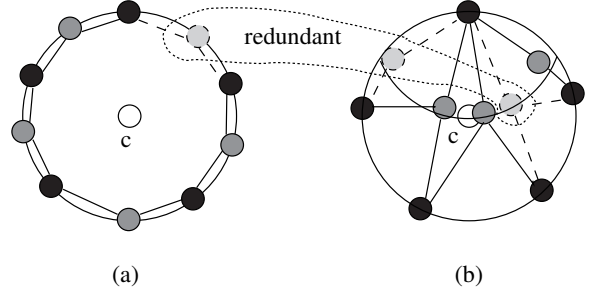Consider any C^* node c and its neighborhood. c has up to

(a)   (b)

Figure 3. Number of R_i neighbors of a C^* node

five black neighbors and we can divide it into two cases as described in figure 3. (a) is the case when every five black neighbors of c are marked black at the same round. In this case, c can have at most four red neighbors if we remove the redundancy. Next consider the other case when one of c's black neighbor is marked prior to other black neighbors, as in (b). Assume the topmost black neighbor is marked prior to other black neighbors. Choose a black neighbor which has the largest distance from the topmost black neighbor. And any red neighbor of this chosen black node makes at least one of the other red neighbors of c redundant. In other words, if c has black neighbors marked in different rounds, c can have at most three red neighbors. Now take an arbitrary traversal of C^*, $c_1, c_2, ..., c_k$, where c_1 contains the black node marked for the first time in its neighborhood. Then the number of red nodes is at most $3 + 4(k - 1) = 4k - 1$. Therefore, $|R| \leq 4|C^*| - 1$. □

Now we are ready to prove performance ratio and message (time) complexity of *CMIS*.

Theorem 1 *CMIS has performance ratio of 8. And it has message complexity of $O(\Delta n)$ and time complexity of $O(n)$, where Δ is the maximum degree.*

Proof. Let T be a tree spanning black and red nodes found by *CMIS*. B and R denotes the set of black and red nodes in T, respectively. And let C^* be the optimal solution of MCDS for the set of given terminals. Then,

$$\begin{aligned} |V(T)| &= |B| + |R| \\ &\leq 4|C^*| + 1 + |R| \\ &\leq 4|C^*| + 1 + 4|C^*| - 1 \ldots \text{Lemma 2} \\ &\leq 8|C^*| \end{aligned}$$

Therefore $V(T)$ is an 8-approximation for MCDS in UDG. Now consider time and message complexity.
Let l be the number of rounds. Since $CMIS$ is incremental, $l \leq |B| \leq \frac{n}{2}$. And each round needs at most three round-trip (δ) message exchanges between any pair of nodes. Hence time complexity is at most $\frac{3}{2}n\delta$, i.e. $O(n)$.
Now define three sets B_i, G_i and R_i for each round as

the set of nodes marked black, gray and red at i–th round, respectively.

Note that $\sum_{i=1}^{l} |B_i| = |B|$ and $\sum_{i=1}^{l}(|G_i| + |B_i|) = n$. For each round, the number of each type of messages is as follows:

#(BLACKACCEPTED) = 1 (only by initiator),
#(BLACKDONE) = $|B_i|$, #(GREYDONE) = $|G_i|$,
#(RED) = $|G_{i-1}|$, #(REDACCEPTED) $\leq |B_i|$,
#(REDDONE) = $|R_i|$ and #(BLACK) $\leq (\Delta - 1)|G_i|$.
So #(MSG) $\leq \sum_{i=1}^{l}(2|B_i| + \Delta|G_i| + |R_i| + |G_{i-1}|) + 1$
$\leq (\Delta + 2)n + 1$, i.e., $O(\Delta n)$. □

We proved that *CMIS* is guaranteed to construct a small backbone efficiently. Note that the leader-election algorithm before *CMIS* has time complexity of $O(n)$ and message complexity of $O(n \log n)$. Together with leader-election algorithm, the total time complexity is still $O(n)$ and the total message complexity is $O(n \log n)$ or $O(n\Delta)$. Now we show the reliability and maintenance efficiency of *CMIS* using simulation results in next section.

4. Simulation

We implemented five schemes: NC1 (Das *et al*), NC2 (Wu and Li), C1 (Wan *et al*), C2 (Cardei *et al*) and *CMIS* (ours). Note that NC1 uses algorithms in [11]. Wu and Li's scheme is very similar with Stojmenovic *et al*'s scheme, so we implemented only Wu and Li's one. Note that NC1, NC2 are non-constant performance ratio schemes and C1, C2 are constant performance ratio schemes.

4.1. Mobility model

Random waypoint model (RWP) is widely used to model the movement of individual node in most recently published papers and also implemented in network simulators like ns-2 or GloMoSim. It is first proposed by Johnson and Maltz in their simulation for DSR [12]. In RWP, each node chooses the destination and speed uniformly and moves toward the destination. When it reaches its destination, it stops there for uniformly chosen pause time.

But recently, several papers have been published pointing out the drawbacks of RWP [3, 18]. They pointed out that the original RWP model fails to reach a steady state in terms of i) node distribution and ii) average node speed. Authors of [3] refer to the failure of reaching a steady state in node distribution as *border effects* and this comes from that a node chooses its next destination only from the given network area, or region. As a result, destination selection is not uniform as expected and a node near the border of the region tends to choose a destination toward the center of the region. They analyzed this effect and showed that the node distribution along the various cuts is not uniform

rather it depends on the distance from the center. One of the authors proposed that several behaviors for border nodes in [2], namely bounce, delete and replace, and wrap around. Among them, especially bounce or wrap around showed better results than delete and replace.

Also they introduced a stability parameter so that some portion of nodes may stay where they are during the entire simulation. And it is shown that the adoption of stability parameter helped the mobile network to enter a steady state more quickly. Another common claim about drawbacks of the original RWP is that the average speed tends to reach zero instead of around the middle point of minimum and maximum speeds. This happens more frequently when the minimum speed is set to zero and maximum speed is relatively low.

We found that the adoption of static parameter is more realistic, so we deployed it in our mobility model. And we modified the destination selection so that a node selects its next destination from a bigger region beyond the border and when a node reaches a boundary, we bounce it to move it back toward the center. Through simulations, we also found that uniform selection of destinations resolves the problem of convergence-to-zero even for speed range of $[0, 1]$.

In our simulation which will be discussed later, we used the modified RWP model as our individual node movement model and our model quickly enters the steady state, i.e., the node distribution lasts nearly uniform and the average speed converges quickly.

4.1.1 Simulation results

In our simulation, 200 hosts are randomly generated in a 600^2 square units space so that the resulting graph is connected. The transmission range is fixed at 100. We generated a movement history for 150 seconds and measured the reliability and potential maintenance cost for each algorithm. Every node maintains its own location information for 10 second window. For mobility parameters, we used minimum speed v_1, maximum speed v_2 and stability parameter p_s, where $(v_1, v_2) = (0, 1), (1, 5), (1, 10)$ and $p_s = 0.5, 0.75, 1$ for each pair of speed range. Totally we set up 9 different mobile environments based on the combinations of the mobility parameters. We ran simulations 50 times for each of 9 different mobility parameters. Figure 4 depicts the summary of the simulation results. The last two are the results of our scheme using two different calculations of stability. *CMIS*-0 and *CMIS*-10 use the past and the future 10 seconds' movement history to compute each node's stability. Here 10 second period is an arbitrarily chosen value and may vary depending on the node mobility of the network. We distinguished them to learn about temporal locality of movements and as we expected, the *CMIS*-10 shows a better result than *CMIS*-0 but *CMIS*-0 still shows a
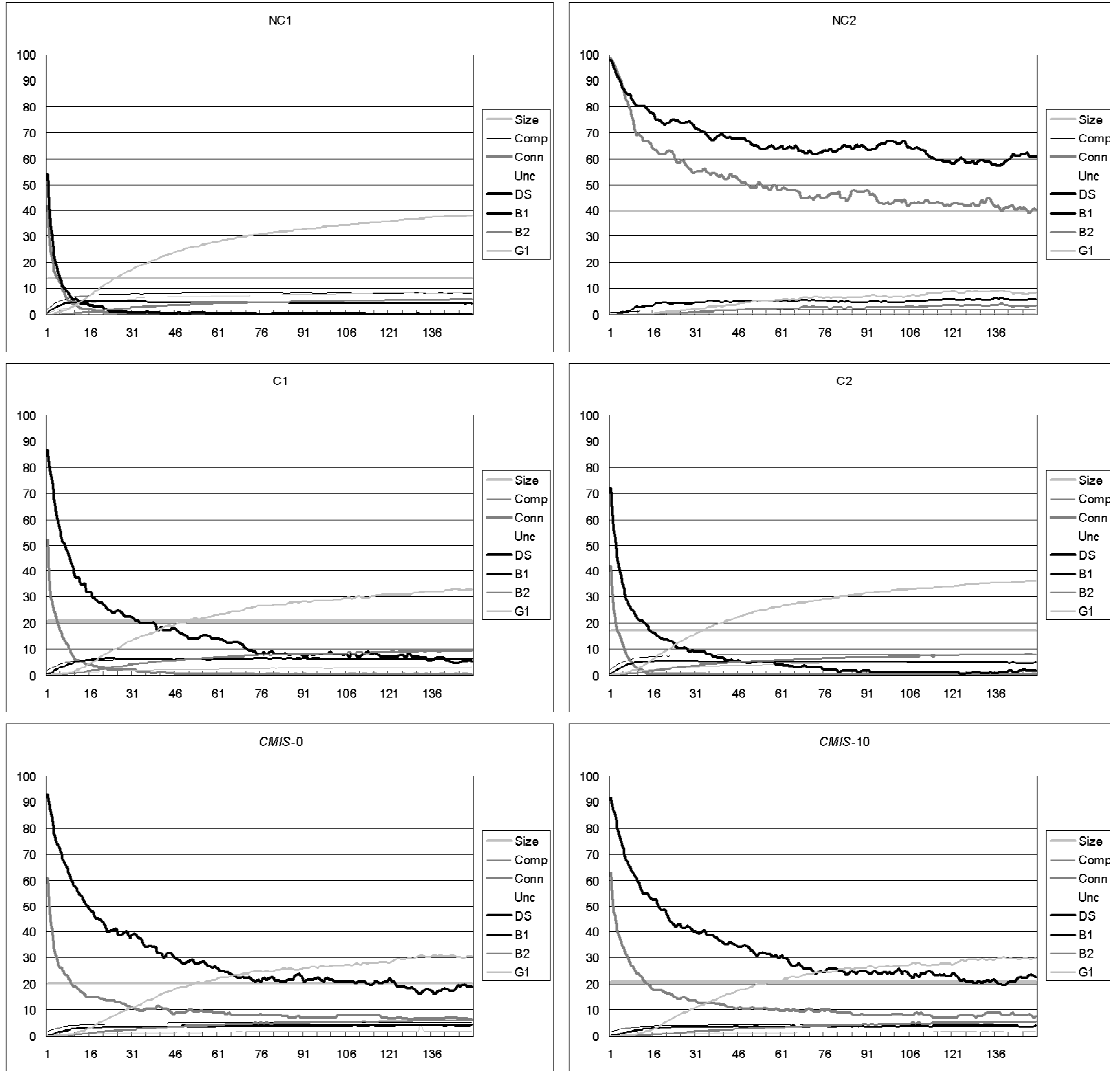
Figure 4. Summary of simulation results

good result compared to other schemes.

In terms of CDS size, the results show that NC1 outperforms all other schemes with any parameters and NC2 performs worst. All the constant ratio schemes perform a little bit worse than NC1 but much better than NC2. But note that NC1 has non-constant performance ratio and hence we cannot guarantee a small-sized backbone always. On the contrary, *CMIS* has the same constant performance ratio (8) as the other constant ratio schemes (C1 and C2). Moreover, our scheme's average case performance is almost equivalent to that of the other constant ratio schemes (C1 and C2) in terms of CDS size.

In terms of reliability of the backbone, we measured the percentages of connected backbones and DS backbones out of 50 runs at each second and also the average duration of the backbone staying connected and/or DS. Note that by a

connected / DS backbone, we meant a backbone which is connected / forms a dominating set (every node is covered by the backbone). Also we counted the number of components of the backbone and the number of uncovered nodes to measure how much the backbone is fragmented and how much the backbone can cover, respectively. According to expectation, the bigger the backbone is, the more reliable it is. Nonetheless we found that both *CMIS*-0 and *CMIS*-10 improve connectivity and DS percentages despite of equivalent size as C1 and C2. This improvement tells us that selection based on nodes' stability and coverage does not hurt the size but improves the reliability of the backbone. Also our scheme shows less fragmentation compared to any other schemes except NC2 and this is natural following the intuition that the more nodes a network has, the more probable that it is connected.
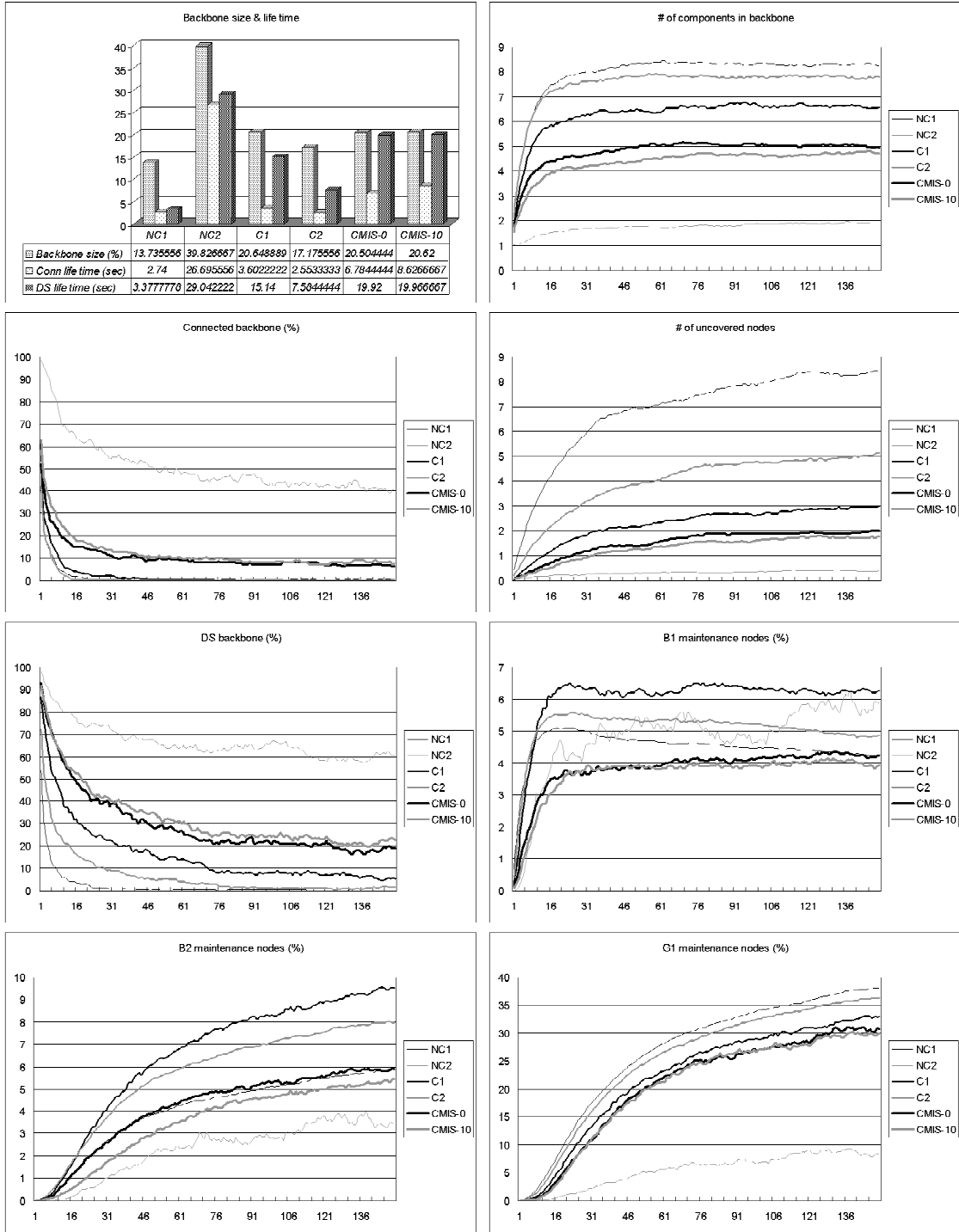
	NC1	NC2	C1	C2	CMIS-0	CMIS-10
Backbone size (%)	13.735556	39.826667	20.648889	17.175556	20.504444	20.62
Conn life time (sec)	2.74	26.695556	3.6022222	2.5533333	6.7844444	8.6266667
DS life time (sec)	3.3777778	29.042222	15.14	7.5044444	19.92	19.966667

Figure 5. Comparisons of simulation results

Next we consider the potential maintenance cost. In order to compute the cost, we define three types, B1, B2 and G1, of potential maintenance needs.

B1 occurs for a backbone node, when some but not all of its initial backbone neighbors generated at the initial construction are out of its transmission range.

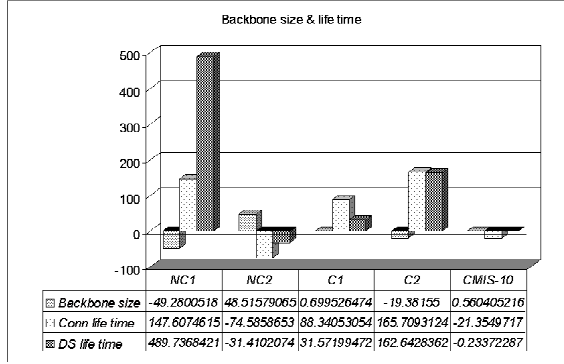**B2** occurs for a backbone node, when all of its initial backbone neighbors are out of its transmission range.

67

Backbone size & life time

	NC1	NC2	C1	C2	CMIS-10
⊞ Backbone size	-49.2800518	48.51579065	0.695526474	- 19.38155	0.560405216
⊟ Conn life time	147.6074615	-74.5858653	88.34053054	165.7093124	-21.3549717
⊠ DS life time	489.7368421	-31.4102074	31.57199472	162.6428362	-0.23372287

Figure 6. Improvements of $CMIS$-0 against each scheme

G1 occurs for a non-backbone node, when all of its initial backbone neighbors are out of its transmission range.

The reason that we measured the potential maintenance cost instead of real maintenance cost is because not every scheme provided the maintenance process. So we counted the percentages of B1, B2 and G1 at every second when the backbone is broken, i.e. either disconnected or not a DS, to see how much maintenance cost may be required. For B2 type, the backbone node is totally disconnected from other backbone fragments, so it may need to reconstruct the backbone structure at least locally. For B1 type, the backbone node is still connected to other backbone nodes, so the possible maintenance cost of this type may be less than that of B2 type. For G1 type, the non-backbone node may need to reconstruct the backbone structure at least locally. Our simulation shows that our scheme reduces the percentages of B1 significantly compared to any other schemes. Also the percentages of B2 and G1 was greatly reduced compared to other schemes except NC2 which generates at least 49% more backbone nodes. This observation supports the reliability of *CMIS* backbone even more.

Figure 5 compares each scheme in every measure and figure 6 shows relative improvements of *CMIS*-0 against each scheme. The results show that *CMIS* generates 49% bigger CDS but increased the average connected backbone lifetime and the average DS backbone lifetime by 149% and 490%, respectively, than NC1. Against NC2, *CMIS* generates 49% smaller CDS and maintained the average DS backbone lifetime up to 69% of NC2. This result is inspiring because it tells us that using coverage as a decision point for selecting backbone nodes turns out to be very effective. Comparing with the other constant ratio schemes, *CMIS* produces almost the same or a little bit more (up to 20% more) nodes as backbone nodes. But it greatly increases the average connected backbone lifetime by 88% and 166%, and also the average DS backbone life time by 32% and 163%. In addition to the average backbone lifetime, we tried to keep track of the life cycle of the backbone. The fragmentation rate, the percentage of uncovered nodes, the potential maintenance needs such as B1, B2 and G1 are all the better than other schemes except NC2.

Since NC2 generates the biggest (about 49% bigger than our scheme) backbone, we focus on comparisons of *CMIS* and other schemes except NC2. The connectivity and DS are maintained the best with the backbone of *CMIS*. The number of nodes uncovered by the backbone is also the smallest with *CMIS*. With *CMIS*, there are up to 2 uncovered nodes, while with other schemes there are 3 to 8.5 uncovered nodes. This tells us that *CMIS* increases not only the DS lifetime but also the coverage of the backbone. And this result comes from the effectiveness of finding a uniformly distributed backbone. *CMIS* backbone structure is less fragmented (up to 5 components) while other schemes' backbones are more fragmented (from 6.5 to 8.5 components). For the potential maintenance cost, the curves of B1, B2 and G1 have growths much smoother than the other schemes as well as lower values. Moreover, the B1 curve converges to the steady state quickly. This quick convergence of B1 along with the less fragmentation strongly supports that the backbone of *CMIS* is reliable and stable.

In summary, our scheme generates a backbone which i) is guaranteed to be small, and ii) has longer backbone lifetime (at least 88% and 32% longer connectivity and DS, respectively), and iii) reduces the potential maintenance cost so that no matter what the maintenance process is, the maintenance cost of *CMIS* is guaranteed to be small.

5. Conclusion

In this paper, we proposed a noble distributed scheme to generate a CDS, which can serve as a virtual backbone in mobile ad-hoc networks. Our backbone is more reliable and more efficient (for both construction and maintenance) than other backbone schemes as long as the backbone sizes are comparable.

We approached this problem based on four factors: backbone size (referred as performance), construction cost, reliability, and potential maintenance cost. For a small backbone size, we find a *maximal independent set*, which is a dominating set, and then interconnect the MIS nodes to form a CDS. MIS provides a good foundation to limit the upper bound for the backbone size, since it's proven that $|MIS| \leq 4|MCDS| + 1$ for any graph topology. Also we reduced the construction cost in terms of message and time by interpolating the two phases of 1) finding DS nodes and 2) interconnecting those nodes. We defined a new rank system as an ordering of (stability, coverage, id) of nodes. And using the rank we significantly increased the reliability of the backbone (both connectivity and coverage). Also the potential maintenance cost of our scheme is much smaller than any other scheme.

We proposed a reliable virtual backbone scheme and applied it to mobile environments with individual mobility model. But mobile environments may be modeled with other mobility models, such as group mobility model, or mixed model. Our next step includes the study of reliable backbones using other mobility models.

References

[1] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan and O. Frieder, Geometric Spanners for Wireless Ad Hoc Networks, *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 5, 2003, pp. 408-421.

[2] C. Bettstetter, Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks, *Proc. of 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, 2001, vol. 5, no. 3.

[3] C. Bettstetter, G. Resta and P. Santi, The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks, *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, 2003, pp. 257-269.

[4] M. Cardei, X. Cheng, X. Cheng and D.-Z. Du, Connected Domination in Multihop Ad Hoc Wireless Networks, *6th International Conference on Computer Science and Informatics*, March 2002, North Carolina, USA

[5] X. Cheng, X. Huang, D. Li and D.-Z.- Du, Polynomial-Time Approximation Scheme for Minimum Connected Dominating Set in Ad Hoc Wireless Networks, *Technical Report TR02-003, Computer Science and Engineering Department, Univ. of Minnesota*, 2002

[6] I. Cidon and O. Mokryn, Propagation and Leader Election in a Multihop Broadcast Environment, *Proc. of 12th International Symposium on Distributed Computing*, 1998, 104-119.

[7] B. N. Clark, C. J. Colbourn and D. S. Johnson, Unit Disk Graphs, *Discrete Mathematics*, 86 (1990), pp. 165-177.

[8] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, pp. 974-978, the MIT press, 1990.

[9] B. Das and V. Bharghavan, Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets, *Proc. of International Conference on Communication*, 1997, pp. 376-380.

[10] R. G. Gallager, P. A. Humblet and P. N. Spiram, A Distributed Algorithm for Minimum-Weight Spanning Trees, *ACM Transactions on Programming Languages and Systems*, 5 (1983), pp. 66-77.

[11] S. Guha and S. Khuller, Approximation Algorithms for Connected Dominating Sets, *Proc. of European Symposium on Algorithms*, 1996, pp. 179-193.

[12] D.B. Johnson and D.A. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, *Mobile Computing*, 1996, pp. 153-181.

[13] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, *Proc. of MOBICOM*, Seattle, Aug. 1999, pp. 151-162.

[14] P. Sinha, R. Sivakumar and V. Bharghavan, Enhancing Ad hoc Routing with Dynamic Virtual Infrastructures, *Proc. of Infocom*, 3 (2001), pp. 1763-1772.

[15] I. Stojmenovic, M. Seddigh and J. Zunic, Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks, *Proc. of IEEE Hawaii International Conference on System Sciences*, January 2001.

[16] P.-J. Wan, K. M. Alzoubi and O. Frieder, Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, *Proc. of Infocom*, 3 (2002), pp. 1597-1604.

[17] J. Wu and H. L. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, *Proc. of the 3rd ACM international workshop on Discrete algorithms and methods for mobile computing and communications*, 1999, pp. 7-14.

[18] J. Yoon, M. Liu and B. Noble, Random Waypoint Considered Harmful, *Proc. of Infocom*, 2003, pp. 1312-1321.