On Calculating Stable Connected Dominating Sets Based on Link-Stability for Mobile Ad Hoc Networks*

Pi-Rong Sheu and Ying-Po Lee
Department of Electrical Engineering
National Yunlin University of Science and Technology
Touliu, Yunlin 640, Taiwan, R.O.C.
Email: sheupr@pine.yuntech.edu.tw

Abstract

In a mobile ad hoc network (MANET), routing based on a connected dominating set (CDS) has been recognized as a promising approach to adapt quickly to fast changing topologies. Due to the dynamic nature and the limited battery power of the mobile nodes, their association and dissociation to and from CDS perturb the stability of the network and thus reconfiguration of CDS is unavoidable. Re-computation of CDS and frequent information exchange among the participating nodes will result in high computation cost overhead. Therefore, it is obvious that a more stable CDS will directly lead to the performance improvement of the whole network. In this paper, we will propose an efficient distributed algorithm that can establish a stable CDS by keeping a node with many weak links from being elected as the members of the CDS. Computer simulations show that the CDSs generated by our distributed algorithm are more stable than those generated by other existing algorithms.

Keywords: Ad Hoc Network, Clustered Architecture, Connected Dominating Set, Stability, Wireless Network

I. Introduction

A mobile ad-hoc network is formed by a group of mobile hosts (or called mobile nodes) without an infrastructure consisting of a set of fixed base stations. A mobile host in a MANET can act as both a general host and a router; i.e., it can generate as well as forward packets. Two mobile hosts in such a network can communicate directly with each other through a single-hop route in the shared wireless media if their positions are close enough. Otherwise, they need a multi-hop route to finish their communications. In a multi-hop route, the packets sent by a source are relayed by multiple intermediate hosts before reaching their destination. MANETs are found in applications such as short-term activities, battlefield communications, disaster relief situations, and so on. Undoubtedly, MANETs play a critical role in situations where a wired infrastructure is neither available nor easy to install.

The research of MANETs has attracted a lot of attentions recently. In particular, since host mobility causes frequent unpredictable topological changes, the task of finding and maintaining routes in MANETs is nontrivial. Extensive research efforts have been devoted to the design of routing protocols for MANETs. At present, a number of efficient routing schemes have been developed [1, 2, 5, 6, 8, 11, 12]. Among those, routing based on a CDS has been recognized as a suitable and promising approach to adapt quickly to fast changing topologies of MANETs [2, 11, 12].

CDS-based routing is related to the concept of *dominating set* in graph theory [11, 12]. A subset of the nodes of a graph is a *dominating set* if every node not in the subset is adjacent to at least one node in the subset. A dominating set is *connected* if there exists a path between any two nodes in the set and the path only consists of the nodes in the set. When a MANET is modeled as a graph, mobile hosts in a CDS are generally called *gateways* while those outside the CDS are called *non-gateways*. Basically, as long as changes in network topology do not affect this CDS there is no need to re-calculate routing tables. As an example, Fig. 1 shows a MANET and its one possible CDS consisting of four gateways: gateways 2, 3, 8, and 9. Note that the CDS is a minimum one.

The main reason why the CDS-based routing is adopted is that it can reduce the routing and searching process to the sub-network induced from the CDS. One of

----

*This work was supported by the National Science Council of the Republic of China under Grant # NSC 91-2213-E-224-015

the main advantages of CDS-based routing is that only gateways need to keep routing information. Thus, to simplify the connectivity management, it is desirable to find a stable CDS of a given MANET. Furthermore, due to the dynamic nature and the limited battery power of the mobile nodes, their association and dissociation to and from the CDS perturb the stability of the network and thus the reconfiguration of CDS is unavoidable. This is an important issue since re-computation of CDS and frequent information exchange among the participating nodes will involve high overhead. Therefore, it is obvious that a more stable CDS will directly lead to the performance improvement of the whole MANET. While extensive research efforts have been devoted to the design of distributed algorithms for calculating CDS [2, 11, 12], little attention is paid to the issues related to the stability requirement of a CDS [11]. The objective of this paper is to build a stable CDS.

In this paper, we will propose an efficient algorithm that can establish a stable CDS by keeping a node with many weak links from being elected as a gateway. Computer simulations show that the CDSs generated by our proposed algorithm are more stable than those generated by other existing algorithms.

The rest of this paper is organized as follows. In Section II, the problem considered in the paper is further stated and related researches are reviewed. In Section III, the proposed algorithm is presented. In Section IV, the performance of our algorithm is evaluated by computer simulations. Finally, in Section V, we make some conclusions.

II. Problem definition and related researches

In this section, we will formally introduce our problem and present the related researches.

CDS-Based Routing

Assume that a CDS has been calculated for a given MANET. A CDS-Based-Routing can been described as follows [11]. If the source is not a gateway, it forwards the packets to a *source gateway*, which is one of its adjacent gateways. Then, this source gateway acts as a new source to route the packets in the subnetwork induced by the CDS. Eventually, the packets will be transmitted to a *destination gateway*, which is either the destination itself or a gateway adjacent to the destination. In the latter case, the destination gateway will forward the packets directly to the destination.

Since each node in a MANET is mobile, the topology of the MANET may change dynamically. In the viewpoint of routing, communication between two nodes far away may be broken because of the link failure between any two intermediate nodes. Thus, it becomes a significant work to provide a stable communication in a MANET. Some investigations on routing protocols have emphasized to find routes consisting of links with higher stability [1, 4, 7, 10]. A link with higher strength is considered to have a higher *link-stability*. Basically, most of them rely on the strength of received power to estimate the stability of a link.

Stability in CDS

As mentioned in Section I, in order to reduce the CDS maintenance overheads and to provide a more stable CDS for upper layer protocols, stability of CDS should be taken seriously. A CDS is more stable if it can be held for a longer period of time. That is, no gateways in the constructed CDS need to update the information of its routing table. Because each node in a MANET is mobile, the topology may change dynamically. Like routing protocols mentioned above, CDS must also deal with link-stability. In this paper, we discover that an efficient way to form a stable CDS is to avoid selecting a node with many links of low link-stability as a gateway as much as possible.

Before describing our proposed algorithm, let us first review some existing algorithms for calculating CDS. In [12], Wu and Li have proposed a simple distributed algorithm for calculating CDS in MANETs (we will call it the *WL algorithm*). The *WL* algorithm mainly consists of a marking process and a re-marking process. Since the problem of determining a minimum CDS of a given connected graph is NP-complete, the CDS derived from the marking process is normally non-minimum. Therefore, it is necessary to use the re-marking process to reduce the size of CDS generated from the marking process. Let the given MANET be model as a graph $G = (V, E)$. A distinct ID, $id(v)$, is assigned to each node v in G. Let $m(v)$ be a marker for each node $v \in V$, which is either T (marked as a gateway) or F (unmarked back as a non-gateway). Let $N(v) = \{u \mid \{v,u\} \in E\}$ represent the *open neighbor set* of node v (i.e., $v \notin N(v)$) and $N[v] = N(v) \bigcup \{v\}$ represent the *closed neighbor set* of node v (i.e., $v \in N[v]$). The marking process of *WL* algorithm consists of the following three steps: Step 1.

Initially assign marker F to every node v in V. Step 2. Every node v exchanges its open neighbor set $N(v)$ with all its neighbors. Step 3. Every node v assigns its marker $m(v)$ to T if it has two unconnected neighbors. Let G' be the subgraph induced by all the nodes with marker T. The re-marking process of *WL* algorithm consists of the following two rules:

Rule 1: *Consider two nodes v and u in* G' *. If* $N[v] \subseteq N[u]$ *in G and* $id(v) < id(u)$ *, the marker of v is changed to F if vertex v is marked; that is,* G' *is changed to* G' *-{v}.*

Rule 2: *Assume that nodes u and w are two marked neighbors of marked node v in* G' *. if* $N(v) \subseteq N(u) \bigcup N(w)$ *in G and id(v) = min{id(v), id(u), id(w)}, then the marker of v is changed to F.*

To establish a smaller and more stable CDS, Wu, Gao, and Stojmenovic proposed another distributed algorithm (we call it the *WGS algorithm*) to calculate power-aware CDS [11]. Basically, the *WGS* algorithm gives preference to nodes with higher battery powers and/or higher node degrees in the selection process of gateways. Specifically, the *WGS* algorithm first expands the original re-marking rules based on node degree to reduce the size of CDS generated from the marking process. Next, the two re-marking rules are further expanded based on energy level to prolong the average lifetime of a gateway.

III. Our proposed stable CDS-formed algorithm

In this section, we will propose a new stable CDS-formed algorithm for calculating stable CDS.

Before describing the idea behind our proposed algorithm, let us first introduce some notations and definitions.

Power strength awareness and the Friis free space model

When electromagnetic wave propagates through air, it will experience power reduction due to reflection, refraction and diffraction. The simplest mode of wave propagation is the Friis free space model [9]. For simplification, let us rewrite the Friis free space equation as follows, which is for short or middle distance transmission: $P_r(d) = \frac{P_t A}{d^2}$, where A is a constant, P_t is the transmission power strength, P_r is the received power strength (which is a function of the T-R separation), and d is the T-R separation distance in meters. By the Friis free space equation, each node can compute the separation from its neighbor by detecting the strength of the signals transmitted from the neighbor if the transmission power strength is known. In this paper, the transmission power strength of each mobile host is assumed to be the same. Furthermore, we assume that each node can often receive beacon packets from its each neighbor and, thus it can detect the received power strength between it and its each neighbor.

The definition of danger link

According to the Friis free space equation, we can obtain the curve shown in Fig. 2, which represents the relationship between the received power strength of node *i* from node *j* and the separation between node *i* and node *j*. In general, if the received power strength of node *i* is under a certain value: $P_{threshold}$, the received signal will be treated as a noise. Now let us define another important parameter: the danger received power strength P_{danger} , which is larger than $P_{threshold}$ (please refer to Fig. 2). If the received power strength of node *i* from node *j* is between P_{danger} and $P_{threshold}$, the corresponding link (*i, j*) is called a *danger link*. When the velocity of each mobile node is no more than ℓ *m/s* and the received power strength of node *i* from node *j* is P_{r_ij} , from Fig. 2, we know that the lifetime of the link (i, j) can be guaranteed for at least $\frac{D_{danger} - D_{threshold}}{2\ell}$ seconds if $P_{r\_ij} \geq P_{danger}$. On the other hand, a danger link is easily broken because of node mobility.

If the velocity of each node in a MANET is restricted under a given value, the communication of two closer neighboring nodes can be guaranteed for a longer period. That is, a link between two closer neighboring nodes is considered to be more stable. According to this observation, a CDS may hold for a longer period if the distance between any two gateways is as close as possible. As an example, Fig. 3 shows a MANET, where there is a bi-directional link between any two nodes if the two nodes can communicate with each other, i.e., if their received power strength is above a predefined value [6] (it is equal to $P_{threshold}$ in this paper. In this examples, we assume $P_{threshold}$ is 3 and P_{danger} is 3.5.). The value on each link represents the received power strength of the two neighboring nodes. A link with a higher value reveals that its two ending nodes are closer to each other. It is easy to observe that in the example of Fig. 3, between nodes 9 and 10, node 9 is more suitable to become a gateway than node 10 because node 9 has no danger links and, which in turn implies that the

CDS including node 9 may hold for a longer time than that including node 10. According to such an observation, to form a stable CDS, we should avoid as much as possible electing a node with many danger links as a gateway. In fact, we will elect nodes with fewer danger links as gateways in our stable CDS algorithm. Obviously, the determination of the P_{danger} value is critical to the performance of our proposed CDS algorithm. The value of P_{danger} will be determined by computer simulations and will be discussed in Section IV.

It is possible that the numbers of danger links of several nodes are the same. Thus a tie may happen frequently. To overcome the case, we need to introduce the so-called average received power strength as the secondary criterion in our method of electing gateways.

The average received power strength

The average received power strength of each node may reflect the status of separations from its neighbors. We will prefer nodes with higher average receive power strengths to become gateways. The reason why we elect gateways according to their average received power strengths instead of their total received power strengths is as follows. The total received power strength of a node may have a close relation to its node degree. That is, a node may have a higher total received power strength just because it has a higher node degree, even if many of its links are unstable. As an example, in Fig. 3, neither node 3 nor node 5 has any danger link. Thus, there is a tie between them and a secondary criterion is needed. If we adopt the total received power strength instead of the average received power strength as the secondary criterion, node 5 will be elected as the gateway. However, it can be observed that each link of node 3 is more stable than the links of node 5. Thus, node 3 may be more suitable to become a gateway than node 5. In other words, the average received power strength can more really reflect the status of the links of a node than the total received power strength. In fact, a higher average received power strength is also helpful in maintaining a longer lifetime of the formed CDS. Thus, in our CDS algorithm, we will use the average received power strength as the secondary criterion in electing gateways. Finally, if a tie still happens, the gateway with smaller ID is preferred to be remarked.

Our proposed algorithm

We consider that if a CDS has fewer danger links, it may have a longer lifetime. Thus, the basic idea behind our algorithm is that a node will have a higher probability to become a gateway if it has fewer danger links. The outline of our proposed algorithm is as follows. First, each node in a MANET broadcasts its beacon packets periodically to declare its existence. The beacon packet of a node carries its ID. Thus, each node can detect the received power strength between it and its each neighbor. Next, the marking process of the *WL* algorithm is executed. Finally, our remarking process, which is similar to the one of the *WGS* algorithm, is applied. The main difference between these two remarking processes is that in ours, a node with more danger links will has a higher probability to be remarked as a non-gateway. However, different gateways may have the same number of danger links. Thus, ties may occur frequently. Therefore, a secondary remarking criterion, the average received power strength of gateway, needs to be introduced to solve such ties. The second remarking criterion is also helpful in maintaining a longer lifetime of the formed CDS. Finally, if a tie still happens, the gateway with smaller ID is preferred to be remarked.

Let us first assume that each node has a unique ID and knows its degree. Next, let us define an important packet used in our algorithm. It is the *CRITERION*(*#-of-danger*, *avg-rev-pwr*, *id*) packet, which is broadcasted by each node. Parameter *#-of-danger* is the number of danger links between the node and its neighboring nodes, *avg-rev-pwr* is the average received strength power of the node, and *id* is the node's identifier.

The marking process of our distributed algorithm consists of the following three steps (which are similar to the *WL algorithm's*):

1. Initially assign marker *F* to every node *v* in *V*.
2. Every node *v* exchanges its *CRITERION*(*#-of-danger*, *avg-rev-pwr*, *id*) packet with all its neighbors. Thus, its open neighbor set *N(v)* and the closed neighbor set *N[v]* can be computed easily.
3. Every node *v* assigns its marker *m(v)* to *T* if it has two unconnected neighbors.

Our re-marking process consists of the following two rules (which are similar to the *WGS algorithm's)*:

Rule 1: Consider two nodes *v* and *u* in G'. If $N[v] \subseteq N[u]$ in *G and* if one of the following conditions holds, the marker of *v* is changed to F:

1. $N[u] \not\subseteq N[v]$ and *#-of-danger*(v) > *#-of-danger*(u)
2. $N[u] \subseteq N[v]$ and *#-of-danger*(v) > *#-of-danger*(u)
3. $N[u] \subseteq N[v]$ and *#-of-danger*(v) =

#-of-danger(u) and *avg-rev-pwr*(v) < *avg-rev-pwr*(u)

4. $N[u] \subseteq N[v]$ and *#-of-danger*(v) = *#-of-danger*(u) and *avg-rev-pwr*(v) = *avg-rev-pwr*(u) and id(v) < id(u)

Rule 2: Assume that *u* and *w* are two marked neighbors of marked vertex *v* in G'. The marker of v is changed to F if one of the following conditions holds:

1. $N(v) \subseteq N(u) \cup N(w)$, but $N(u) \not\subseteq N(v) \cup N(w)$ and $N(w) \not\subseteq N(v) \cup N(u)$。
2. $N(v) \subseteq N(u) \cup N(w)$ and $N(u) \subseteq N(v) \cup N(w)$, but $N(w) \not\subseteq N(v) \cup N(u)$; and one of the following conditions holds:
   - (a.) *#-of-danger*(v) > *#-of-danger*(u)
   - (b.) *#-of-danger*(v) = *#-of-danger*(u), but *avg-rev-pwr*(v) < *avg-rev-pwr*(u)
   - (c.) *#-of-danger*(v) = *#-of-danger*(u) and *avg-rev-pwr*(v) = *avg-rev-pwr*(u), but id(v) < id(u)
3. $N(v) \subseteq N(u) \cup N(w)$, $N(u) \subseteq N(v) \cup N(w)$ and $N(w) \subseteq N(v) \cup N(u)$ if one of the following conditions holds:
   - (a.) *#-of-danger*(v) > *#-of-danger*(u) and *#-of-danger*(v) > *#-of-danger*(w)
   - (b.) *#-of-danger*(v) = *#-of-danger*(u) and *#-of-danger*(v) > *#-of-danger*(w) and *avg-rev-pwr*(v) < *avg-rev-pwr*(u)
   - (c.) *#-of-danger*(v) = *#-of-danger*(u) and *#-of-danger*(v) > *#-of-danger*(w) and *avg-rev-pwr*(v) = *avg-rev-pwr*(u), id(v) < id(u)
   - (d.) *#-of-danger*(v) = *#-of-danger*(u) = *#-of-danger*(w) and *avg-rev-pwr*(v) = $\min\{\text{avg-rev-pwr(v)}, \text{avg-rev-pwr(u)}, \text{avg-rev-pwr(w)}\}$
   - (e.) *#-of-danger*(v) = *#-of-danger*(u) = *#-of-danger*(w) and *avg-rev-pwr*(v) = *avg-rev-pwr*(u) < *avg-rev-pwr*(w) and id(v) < id(u)
   - (f.) *#-of-danger*(v) = *#-of-danger*(u) = *#-of-danger*(w) and *avg-rev-pwr*(v) = *avg-rev-pwr*(u) = *avg-rev-pwr*(w) and $id(v) = \min\{id(v), id(u), id(w)\}$

Now, let us use the MANET shown in Fig. 3 as an example to illustrate the operation of our proposed algorithm. The number within a node represents its identifier. The value on each link represents the received power strength of the two neighboring nodes.

First, let us apply our marking process to Fig. 3. Among the neighboring nodes of node 2, node 5 is not connected to node 1. Therefore, node 2 will be marked as gateway. For the same reason, nodes 3, 5, 8, 9, and 10 will all be marked as gateways. Next, let us apply our remarking process to the CDS generated from the above marking process. Consider gateways 9 and 10. Observe that because $N[9] = \{8,9,10,11,12\}$ and $N[10] = \{8,9,10,11,12\}$, $N[9] \subseteq N[10]$ and $N[10] \subseteq N[9]$. Furthermore, #-of-danger(10) > #-of-danger(9). Thus gateway 10 will be remarked to become a non-gateway (according to Rule 1-2).

Consider gateways 2, 3, and 5. Because $N(2) = \{1,3,5,6,7\}$, $N(3) = \{1,2,4,5\}$, and $N(5) = \{2,3,4,6,7\}$, $N(5) \subseteq N(2) \cup N(3)$ and $N(2) \subseteq N(5) \cup N(3)$. $N(3) \not\subseteq N(2) \cup N(5)$. Furthermore, #-of-danger(5) = #-of-danger(3). But, *avg-rev-pwr*(5) < *avg-rev-pwr*(3). Thus gateway 5 will be remarked to become a non-gateway (according to Rule 2-2(b)). The final CDS obtained by our algorithm is shown in Figure 4.

Fig. 5 shows the resulted CDS architecture when the *WL* algorithm [12] is applied to Fig. 3. Gateway 9 is remarked to become a non-gateway according to their Rule 1. Thus, there are five gateways in Fig. 5. Fig. 6 shows the resulted CDS architecture when the *WGS* algorithm [11] is applied to Fig. 3. Between gateways 9 and 10, gateway 9 is remarked to become a non-gateway according to their Rule 1a. Among gateways 2, 3, and 5, gateway 2 is remarked to become a non-gateway according to their Rule 2a.

By observing Fig. 4 and 5, Fig. 4 has fewer gateways than Fig. 5. Furthermore, none of the gateways in Fig. 4 has a danger link while gateway 10 in Fig. 5 has a danger link. Compared Fig. 4 with Fig. 6, it can be seen that Fig. 6 has a gateway with danger link, and the average received power strength of gateway 5 in Fig. 6 is smaller than that of gateway 3 in Fig. 4. In summary, the CDS formed by our algorithm is more stable than those formed by the other two algorithm. Computer simulations in the next section will further verify this.

IV. Computer simulations

In this section, we will evaluate the performance of our CDS-formed algorithm and compare it with the *WL* algorithm [12], and the *WGS* algorithm [11].

Our simulation environment is a physical area of $100 \times 100\ m^2$ in free space. Each mobile host is randomly distributed and the transmission range of each mobile host is 40 *m*. In our experiments, we adopt three different-sized of MANETs: 40-node, 60-node, and 80-node.

Mobility model

In [3], Chiang proposed a probability model which can provide more stable and more realistic node movement than a random mobility model can. Each mobile host tends to keep on going at the current direction and speed. The probability model is controlled by a three-state Markov chain and the state diagram is shown in Fig. 7. We name the mobility model *the Chiang's Markovian model*. The states represent motion directions. The probabilities of staying in the current state or changing to another state are specified in the transit matrix *P* and the transit matrix in [3] is $P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.3 & 0.7 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix}$.

In the Chiang's Markovian model, the velocities of each mobile host on the *x* and *y* coordinates are uniform. This makes each node in the model to have only eight motion directions. To support more realistic motion, we modify the uniform velocity into a limited random velocity. The state diagram of our modified Markovian model is shown in Fig. 8.

The determination of P_{danger}

The determination of the value of P_{danger} is critical to the performance of our algorithm. Recall that the main objective of our algorithm is to form a CDS with higher stability. Thus, in our computer simulations, we will evaluate our CDS-formed algorithm in terms of the CDS's lifetime. In our simulations, the definition of CDS's lifetime is the period during which no gateways in the constructed CDS need to update the information of its routing table. To be more specific, during the CDS' lifetime, there is no any link between gateways becoming broken and there is no any non-gateway being disconnected from the CDS. The longer the CDS lifetime is, the more stability the CDS has. After a CDS is formed by our algorithm, we will measure the CDS in terms of its lifetime influenced by node motion.

In our simulations, the movement of each node follows either the original Chiang' Markovian model or the modified Chiang' Markovian model as described previously. We will determine a proper P_{danger} values to obtain a longer CDS's lifetime. In Fig. 9, the velocity of the original Chiang's Markovian model is equal to 0.1 m/s. The transit matrix *P* is the same as that in [3]. Here, G is a constant (= 80000) and $G = P_t A$ for the Friis free space equation $P_r(d) = \frac{P_t A}{d^2}$. It can be observed that a CDS's lifetime is longer when P_{danger} is G/(35×35). Thus, in the following, we will set P_{danger} to G/(35×35).

Comparisons among different CDS-formed algorithms

Next, we will compare the CDS's lifetime of our algorithm with those of the *WL* algorithm[12], and the *WGS* algorithm[11].

Fig. 10 shows the comparison on the CDS's lifetimes among each different CDS-formed algorithms when the velocity of the original Chiang's Markovian model is equal to 0.1 m/s and the transit matrix is $P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.3 & 0.7 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix}$. It can be observed that the CDS's lifetime of our algorithm is 42.09% over than that of the *WL* algorithm, and 7.97% over than that of the *WGS* algorithm at this low speed and high mobility case.

Fig. 11 shows the comparison on the CDS's lifetimes when the velocity of the modified Chiang's Markovian model is less than or equal to 0.5 m/s and the transit matrix is $P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.3 & 0.7 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix}$. It can be observed that the CDS's lifetime of our algorithm is 49.79% over than that of the *WL* algorithm, and 15.97% over than that of the *WGS* algorithm at this higher speed and high mobility case.

Fig. 12 shows the comparison on CDS's lifetimes when the modified Chiang's Markovian model is less than or equal to 0.5 m/s and the transit matrix is $P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.7 & 0.3 & 0 \\ 0.7 & 0 & 0.3 \end{bmatrix}$. It can be observed that the CDS's lifetime of our algorithm is 48.11% over than that of the *WL* algorithm, and 8.88% over than that of the *WGS* algorithm at this higher speed and low mobility case.

V. Conclusions

In a MANET, a stable CDS can avoid frequent architecture update and reduce the overhead to re-establish CDS. Furthermore, it can provide a stable framework for upper-layer protocols. In order to establish stable CDS adaptable to mobile environments, in this paper, we have proposed a stable CDS-formed algorithm based on link stability. The results of computer simulations reveal that the CDS's lifetime of our algorithm is longer than those of the *WL* algorithm, and the *WGS* algorithm. Therefore, we conclude that the CDS constructed by our CDS-formed algorithm have higher stability for mobile environments than the present CDS-formed algorithms.

References

[1] S. Agarwal, A. Ahuja, J. P. Singh, and R. Shorey, "Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks", Proc. of IEEE ICC 2000, vol.3, pp. 1697-1701, Jun. 2000.

[2] K. M. Alzoubi, P.J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks", Proc. 35th Hawaii Int'l Conf. System Sciences, pp. 1-7, Jan.. 2002.

[3] C. C. Chiang, *Wireless network multicasting*, PhD dissertation, University of California, Los Angeles, Department of Computer Science, May 1998.

[4] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Personal Communications Magazine*, vol. 4, no. 1, pp. 36-45, Feb. 1997.

[5] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio network", ACM/Baltzer Journal of Wireless Networks, vol.1, pp. 225-238, 1995.

[6] A. Iwata, C. C. Chiang, G. Pei, M. Gerla, and T. W. Chen, "Scalable routing strategies for ad hoc wireless networks", IEEE J. Select. Areas Commun., vol.17, pp. 1369-1379, Aug. 1999.

[7] K. Paul, S. Bandyopadhyay, A. Mukherjee, and D. Saha, "Communication-aware mobile hosts in ad-hoc wireless network," *IEEE International Conference on Personal Wireless Communication*, pp. 83-87, 1999.

[8] T. S. Rappaport, *Wireless Communication: Principles and Practice*, Englewood Cliffs, NJ: Prentice-Hall, pp. 69-122, 139-169, 1996.

[9] R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile wireless networks for quality-of-service support", ACM/Baltzer Mobile Networks and Applications, vol.3, no.1, pp. 101-119, Jun. 1998.

[10] T. C. Tien, and S. J. Upadhyaya, "A local/global strategy based on signal strength for message routing in wireless mobile ad-hoc networks," *Proc. of IEEE Academia/Industry Working Conference on Research Challenges*, pp. 227-232, Apr. 2000.

[11] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks", Journal of Communications and Networks, vol.5, no.2, pp. 169-178, Mar. 2002.

[12] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks", Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7-14, 1999.

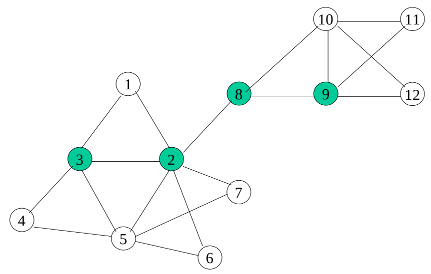

Fig. 1. A MANET with a minimum CDS

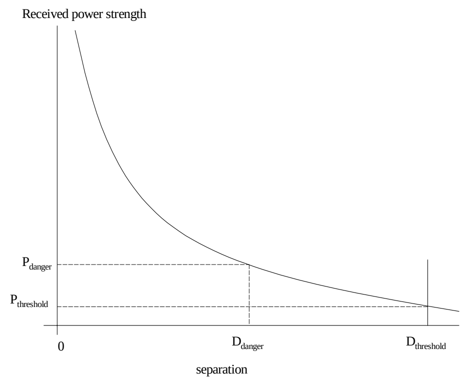

Fig. 2. The relationship between received power strength and separation

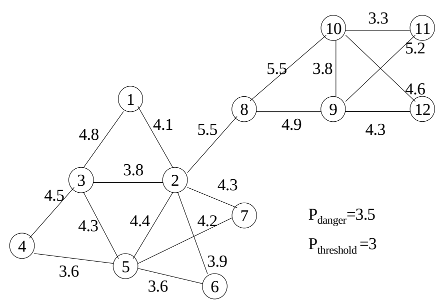

Fig. 3. A MANET with danger links

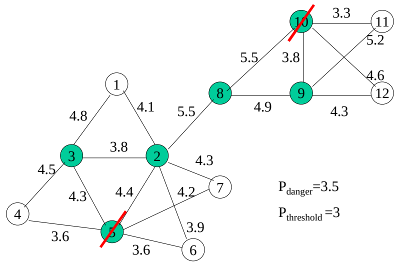

Fig. 4. An example to illustrate the operation of our CDS-formed algorithm

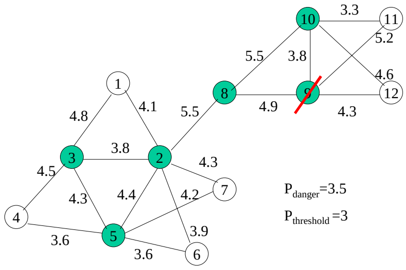

Fig. 5. The CDS obtained by the *WL* algorithm for the MANET in Fig. 3

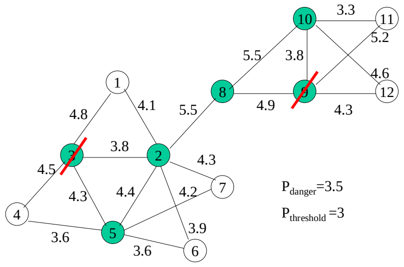

Fig. 6. The CDS obtained by the *WGS* algorithm for the MANET in Fig. 3

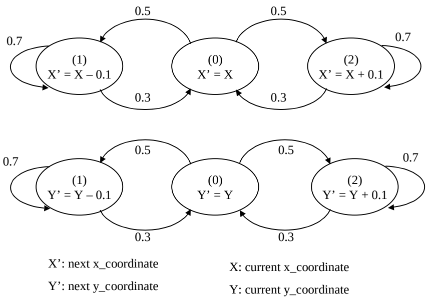

Fig. 7. The state diagram of a three-state Markov chain

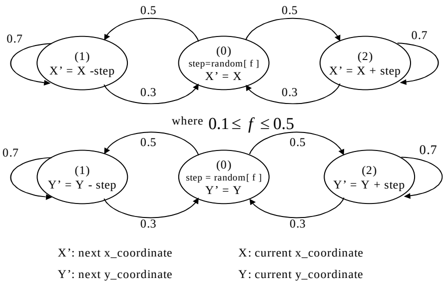

Fig. 8. The modified three-state Markov chain

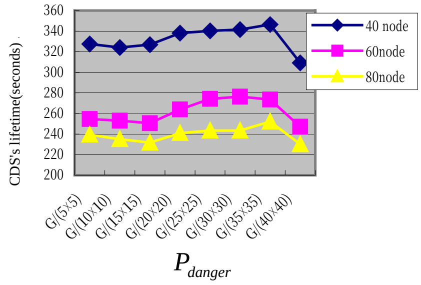

Fig. 9. The determination of P_{danger}

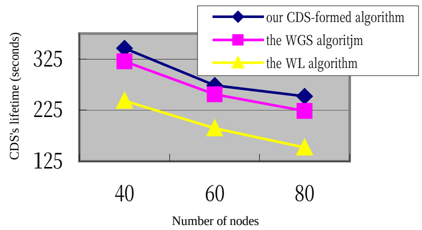

Fig. 10. Comparison on CDS's lifetime: Case 1

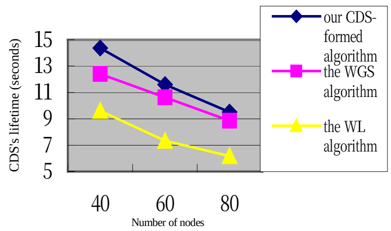

Fig. 11. Comparison on CDS's lifetime: Case 2

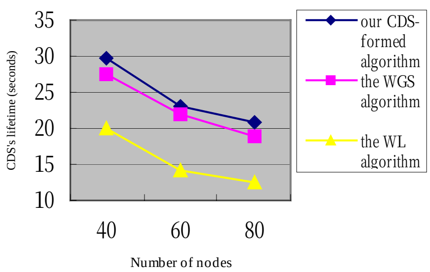

Fig. 12. Comparison on CDS's lifetime: Case 3