Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An effective connected dominating set based mobility management algorithm in MANETs

Xin-yu WANG[†], Xiao-hu YANG, Jian-ling SUN, Wei LI, Wei SHI, Shan-ping LI[†‡]

(*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: wangxinyu@zju.edu.cn; shan@zju.edu.cn

Received Mar. 13, 2008; revision accepted July 2, 2008

Abstract: This paper proposes a connected dominating set (CDS) based mobility management algorithm, CMMA, to solve the problems of node entering, exiting and movement in mobile ad hoc networks (MANETs), which ensures the connectivity and efficiency of the CDS. Compared with Wu's algorithm, the proposed algorithm can make full use of present network conditions and involves fewer nodes. Also it has better performance with regard to the approximation factor, message complexity, and time complexity.

Key words: Mobile ad hoc network (MANET), Connected dominating set (CDS), Mobility, Dominator, No-key dominator, Approximation factor

doi:10.1631/jzus.A0820189 **Document code:** A **CLC number:** TP393

INTRODUCTION

Lack of infrastructure is an important reason why implementing mobile ad hoc networks (MANETs) is difficult. An effective solution for overcoming present problems is to construct backbone networks (Das *et al*., 1997), i.e., to assign MANET nodes with different roles, corresponding to core networks of special connected devices in the infrastructure of cell phone systems, such as gateways, bases, etc. The dynamic mobility management of backbone networking is needed after its construction. But as each node in this system can move freely, the network topology is unpredictable. Also, the resources of the node are limited and all nodes in this system are completely decentralized, so dynamic mobility management poses a difficult problem. Connected dominating set (CDS) can naturally form a backbone subnet with network characteristics. It has the following attributes: nodes in the network either belong to the dominating set, or are connected to at least one node belonging to the dominating set (Wu and Li, 1999). All nodes in the dominating set are connected to the network. Multipoint relaying (MPR) is often used to construct the CDS which forms the backbone (Wu *et al*., 2006), but MPR cannot resolve the problem in a mobile environment. There are normally two ways to overcome this problem: reconstruction of the whole CDS network or local adjustment. The degree of change to the network, the rate of convergence of network information, the cost of reconstructing or maintaining the network should be considered in order to achieve an optimal solution. When the degree of change to a CDS network is low, a simple local adjustment may be more efficient, i.e., a mobility management algorithm is preferred.

This paper proposes a CDS based mobility management algorithm (CMMA), which is MPR compliant, to overcome this problem in MANETs. This algorithm takes full advantage of the network topology to address different approaches of node entering, exiting and movement in MANETs. This is a distributed heuristic algorithm which has no dependency on the state of the node, and is also self-adaptive with respect to the intensity of node movement.

[‡] Corresponding author

RELATED WORKS

Das *et al.*(1997) proposed the idea of maintaining CDS in MANETs for the first time; they summarized network topological change into single-node and multiple-node movement. In the case of single-node movement, different approaches are taken to deal with three situations: ordinary node, leaf node and interior node. The movement of multiple nodes without overlapping neighbors can be deemed as an iterative movement of a single node. The idea of (Das *et al.*, 1997) is to distinguish attributes of nodes, and different routines for managing node movement are adopted, depending on which situation is under consideration. Realization is complex, and when multiple nodes with overlapping neighbors are moving simultaneously, the idea is not helpful and the CDS network needs to be reconstructed. Therefore, the idea lacks feasibility and practicality, and is not a suitable approach to the maintenance of virtual backbones in MANETs.

Wu and Li (1999) proposed a general algorithm for maintaining CDS networks. They analyzed movement of nodes with an on/off model from the perspective of management domain. The movement of nodes can be viewed as exiting from one small management domain and entering another management domain. This mobile model captures the essence of relative location changes among nodes and investigates general adjacency between nodes, ignoring sub-factors with less influence on CDS. Also, this model is more systematic and applicable than that of (Das *et al.*, 1997). However, Wu's algorithm cannot guarantee the CDS size after adjustment either, and has too strong a dependency on the absolute location of nodes. Compared with the idea of (Das *et al.*, 1997), Wu's CDS maintenance algorithm has its own defects, such as involving too many nodes and not fully using present network conditions of dominating sets.

Kim *et al.*(2001) proposed the concepts of a stable zone and a caution zone, which are defined based on the location and speed information. Stable zone nodes have shorter distances and can establish stable connections with surrounding nodes, while caution zone nodes have longer distances and can only establish unstable connections with surrounding nodes. Su *et al.*(2001) proposed a node mobility prediction algorithm based on location/speed information, and thereby reduced maintenance cost. However, all of these algorithms heavily rely on the nodes' location/speed information provided by GPS (global positioning system), and thus lack generality.

Recently, the mobility model has become a hot research topic in mobility management. Santi (2005) divided topology control in ad hoc into homogeneous and nonhomogeneous approaches, and further classified the nonhomogeneous approach into three categories, i.e., location based, direction based and neighbor based. Lee and Hou (2006) built a mathematical model for characterizing both steady state and transient behaviors of user mobility. This model is based on real-world traces and is only a very specific scenario (Resta and Santi, 2008). Spyropoulos *et al.*(2006) proposed the idea that mobility can be used for transmission by making nodes carry data around the network, but this analysis does not take contention into account (San and Puerto, 2007).

CMMA DEFINITION

This paper proposes a simple CDS maintenance algorithm based on MPR which could process node entering and exiting. This algorithm maintains the CDS size well, while keeping network bandwidth needed to a relatively low level.

The notation used in the algorithm is as follows:

Suppose at time t, set V is the universal set; set D is a stable CDS, and any node in it is called a 'dominator'; set C is the set which contains all the nodes that are not included in D but have a connection with at least one node in D. Any node in C is called an 'ordinary node'. From the definition of a dominating set,

$$V=D\cup C,\quad D\cap C=\varnothing.$$

Meanwhile, there is a function f:

$$\{f: C\leftrightarrow D\},\ C \text{ is the domain of } f,$$

i.e., any node in C must have at least one corresponding dominator in D.

min(S) yields the node with the smallest ID in the node set S; *maxCov*(S) yields the node in S that could cover the maximum number of ordinary nodes; $N(p)$ yields the set of all nodes one hop away from node p; $N^2(p)$ yields the set of all nodes two hops away from node p.

'No-key dominator': there is only one dominator in all neighbors of dominator p, i.e., $count(N(p)\cap D)=1$, or all dominators around p can interconnect without going through p. 'Key dominator' is the dominator that does not satisfy the above conditions.

Node entering

Suppose node p enters this system (a CDS network is guaranteed to be formed, i.e., the system is still connected after p's entry). Then

$$V'=V\cup\{p\},\ C'=C\cup\{p\},\ T=N(p)\cap D. \tag{1}$$

And node p becomes an ordinary node in the network.

If $T\neq\varnothing$, i.e., there are dominators in $N(p)$, then select dominator d' with the smallest ID:

$$d'=\min(T),\ \ D'=D. \tag{2}$$

If $T=\varnothing$, i.e., there is no dominator in $N(p)$, then select and upgrade the node d' with the smallest ID from $N(p)$ to be a dominator, and add it to the dominator set. This selection process is accomplished by $N(p)$'s dominators in $N^2(p)$.

$$T=N(p),\ d'=\min(T),\ D'=D\cup\{d'\},\ C''=C'/\{d'\}. \tag{3}$$

Finally, renew the dominating relation between dominator d' and newly added node p:

$$f'=f\cup\{p\mapsto d'\}. \tag{4}$$

In particular, when many nodes enter the networks simultaneously, if $T=\varnothing$, then an ordinary node has to be upgraded to be a dominator, so the coverage of a newly upgraded node over newly added nodes should be considered, i.e., constructing a dominator set with the fewest nodes. We can replace $\min(T)$ with a function for the largest coverage rate that could cover the maximum number of new nodes that have just entered this system, i.e., $d'=maxCov(T)$.

Node exiting

Suppose the exiting of this node has no impact on the connectivity of the whole network. When node p has exited from the system, the definition can be

$$V'=V/\{p\}. \tag{5}$$

If $p\in C$, then

$$C'=C/\{p\},\ D'=D,\ f'=f/\{p\mapsto f(p)\}, \tag{6}$$

that is, if p is an ordinary node, then eliminate p from the set of ordinary nodes, keep the set of dominators unchanged and remove the dominating relation with p.

Note that if $u=f(p)$ is a no-key dominator and satisfies $\forall v,\ N(u)\cap C|N(v)\cap D\neq\varnothing$, i.e., if every ordinary neighbor around u has other dominators, u can be degenerated to an ordinary node, then

$$\begin{cases} D''=D'/\{u\},\ T=N(u)\cap C,\ C''=C'/T, \\ f''=\{x:C;y:D\,|\,x\mapsto y\in f'\wedge y\neq u\}. \end{cases} \tag{7}$$

For all nodes in T, i.e., nodes originally dominated by u, the operation of entering is performed.

If $p\in D$, i.e., p is a dominator, let $u=p$, $f'=f$, $C'=C$, $D'=D$, the node is removed after Eq.(7) is performed.

But if p is a key dominator, all nodes in T shall be marked as dominators (after that, leaf dominators shall degenerate to ordinary nodes), i.e.,

$$\begin{cases} C'''=C''/T,\ D'''=D''\cup T, \\ f'''=\{x:C;y:D\,|\,x\mapsto y\in f''\wedge\ x\notin T\}. \end{cases} \tag{8}$$

After being processed by the maintenance algorithm above, CDS shall keep its original attributes. This is because according to supposition, exiting of nodes has no impact on the connectivity of the whole network. When dominator p is removed, network connectivity is kept even if it causes segmentation of the sub-graph in the dominating set.

Therefore there exists a node b in $N(p)$, $N(b)-N(p)\neq\varnothing$, i.e., b has neighbors that do not belong to the dominating domain of node p. And like the attributes of the dominating set, there must exist dominators other than p in $N^2(p)$. Nodes similar to b can be called 'bridge nodes', the set of which is denoted as B. Suppose a CDS network is segmented into connected dominating subsets D_1 and D_2 after node p has been removed. Then $N^2(B)\cap D_1\neq\varnothing$ and $N^2(B)\cap D_2\neq\varnothing$ can be deduced from network connectivity. Namely, B can be the middle nodes connecting D_1 and D_2. Without loss of generality, assume subnet G_1 dominated by D_1 interconnects with subnet G_2 dominated by D_2 through node set $N(p)\cup\{p\}$. Otherwise, if there are other paths connecting G_1 and G_2, connectivity of the dominating set on this path can be guaranteed with the dominating set construction algorithm and in such situations the exiting of *p* has no impact on the connectivity of the dominating set. If nodes in *B* belong to either G_1 or G_2, then $N(p)$ is the connected set from the interconnection of G_1 and G_2; otherwise, G_1 and G_2 are disconnected. If one node in *B* belongs to G_1 and G_2, then adding this single node into the dominating set can guarantee its connectivity. In conclusion, a network dominating set still covers the whole network and is still CDS after using the above maintenance algorithm.

Node movement

There are several choices of node movement models, such as Random Walk Model (Davies, 2000), Random Waypoint Model (Johnson and Maltz, 1996), Random Direction Model (Royer *et al*., 2001), Boundless Simulation Area Model (Haas, 1997), Gauss-Markov Mobility Model (Tolety, 1999), etc. For simplicity and without loss of generality, Random Waypoint Model is chosen here, i.e., after a node has moved to a place, it pauses for a while before moving with a new speed and in a new direction. Then the process of node movement can be divided into three parts: node start, node movement, and node stop (with a corresponding reaction for each part). The description is as follows:

(1) Node start: When a node starts to move, its neighbors detect its departure, and the CDS maintenance algorithm corresponding to node exiting is executed.

(2) Node movement: In the process of moving a node, nodes within its path of movement determine the node is moving because they detect its entering and exiting in a short time, so no maintenance is needed for these nodes.

(3) Node stop: When a node stops moving in a certain place, the surrounding nodes detect that the node keeps connecting to them for a long time, and determine that this node is the newly added node, and the corresponding CDS maintenance algorithm is executed.

Therefore, the process of node movement consists of node entering and node exiting. Please refer to the description of node entering and node exiting for all relevant algorithms.

CASE STUDY

Figs.1a and 1b represent the topological structure and corresponding CDS before and after nodes are added to the network in a simulation experiment, respectively. Fig.1a consists of 11 nodes, and a CDS has been constructed with a dominating set construction algorithm. After nine nodes have been added to Fig.1a, the topological result is shown in Fig.1b, with the nine nodes labeled from 12 to 20 according to their entering sequence. It can be seen that with the CDS maintenance algorithm, the CDS network generated still stays relatively succinct. Before the addition of new nodes, nodes 2, 4, 6 and 10 are the dominators which dominate the other nodes (Fig.1a). After the nine nodes entered the system, nodes 1, 9 and 10 turned out to be the dominators that dominate all the other nodes (Fig.1b). The addition of nodes 12 and 13 upgrades node 9 to be a dominator and the addition of node 14 upgrades node 13 to be a dominator. Nodes 15 and 16 are dominated by nodes 6 and 4, respectively. The addition of node 17 upgrades node 1 to be a dominator which dominates nodes 18 and 19 too. Node 20 is dominated by node 2.

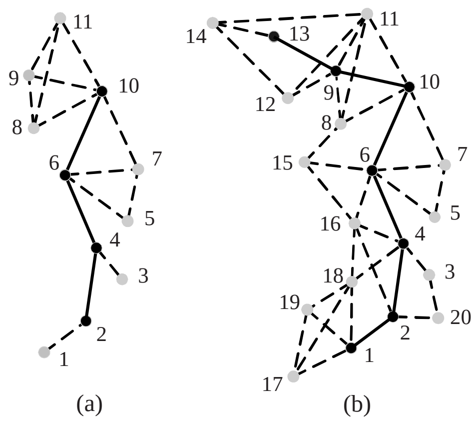

Fig.1 Topological structure and corresponding CDS before (a) and after (b) nodes are added to the network. Black dots are dominators and gray dots are ordinary nodes

Note that node 10 is not the node with the smallest ID around nodes 8 and 11 but, for the purpose of limiting the number of groups, node 10 is used to dominate nodes 8 and 11 instead of the new dominator 9 with a smaller ID.

Figs.2a and 2b represent the topological structure and corresponding CDS before and after four nodes are removed from the 12-node network sequentially in a simulation experiment, respectively.

Before the exiting of the four nodes, nodes 2, 4, 5, 8 and 11 are the dominators (Fig.2a). As shown in Fig.2b, after the exiting of nodes 1, 5, 10 and 12, node 2 does not need to be a dominator as node 3 is dominated by node 4, so node 2 degrades to be an ordinary node dominated by node 4. The exiting of node 5 upgrades node 7 to be the dominator connecting dominators 4 and 8. The exiting of nodes 10 and 12 makes dominator 11 useless, and thus it degrades to be an ordinary node dominated by node 8.

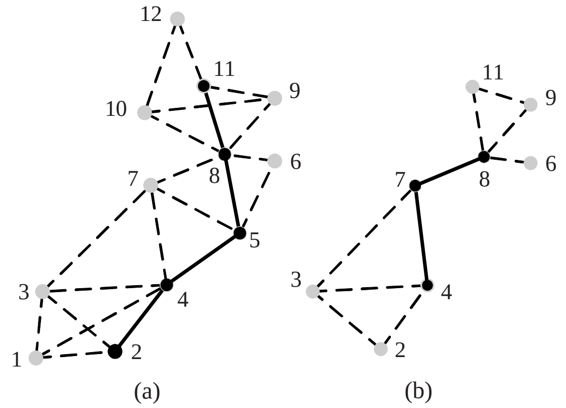

Fig.2 Topological structure and corresponding CDS before (a) and after (b) four nodes are removed from the 12-node network sequentially. Black dots are dominators and gray dots are ordinary nodes

PERFORMANCE ANALYSIS

Analysis of approximation factors

Since computing the minimum CDS is an NP-C problem, the approximation factor reflects the approximation degree between the algorithm result and existing optimum solution. In the best circumstances every new node luckily has a dominator nearby, and the CDS constructed by this algorithm might close in upon an optimum approximation factor. Even in the worst circumstances, as discussed above, every newly added node introduces one dominator (the probability of such a situation is low). If the number of nodes added is denoted by n_A, then the approximation factor of the maintenance algorithm is $O(n_A)$. This guarantees the efficiency of the algorithm result.

When an ordinary node dominated by dominator *p* is removed, if there are other ordinary nodes that must be covered by node *p*, then the effect of the maintenance algorithm is the same as that of its construction algorithm; if there is no ordinary node that must be covered by node *p*, then *p* degrades to an ordinary node. Therefore the maintenance algorithm keeps the approximation factor of its construction algorithm. The impact on the approximation factor of the exiting of no-key dominator *p* in the dominating set is equal to the node set *N*(*p*) entering the network, and therefore the approximation factor is $O(\Delta)$. All Δ in this paper denote the upper limit of the degree of vertices (i.e., neighbors of any node) in a graph. The exiting of key dominator *p* in the dominating set will also turn *N*(*p*) into dominators, so the approximation factor is also $O(\Delta)$.

Message complexity of the maintenance algorithm

Message complexity represents the number of messages sent when computing CDS in the worst circumstances. Burns (1980) proved that in asynchronous rings with point-to-point transmission, any distributed algorithm for leader election sends at least $O(n\log n)$ messages. Alzoubi *et al*.(2002) extended this famous conclusion by establishing the $O(n\log n)$ lower bound on the message complexity for distributed algorithms for leader election, spanning tree, and nontrivial CDS in WANETs.

For any newly added node, if there are dominators nearby, these dominators will detect newly added nodes and send messages to them, while new nodes will pick only one as its dominator. In the worst circumstances, all neighbors of the new node are dominators, and then the message complexity of the maintenance algorithm is $O(\Delta)$. If there is no dominator around the new node, dominators two hops away from the new node shall assign a new dominator to cover the new node. As illustrated in Fig.3, in the worst circumstances all neighbors of the new node may be promoted as dominators. Every dominator will send a message to its ordinary nodes, while all neighbors of the new node *p* will also send messages to *p*. In any case, *p* will choose one and only one neighbor as its dominator. In conclusion, the message complexity of the maintenance algorithm is $O(2\Delta)$. The message complexity of the whole maintenance algorithm is $O(2\Delta)$.

In the worst circumstances the exiting of an ordinary node may cause its dominator to be an ordinary node and proclaim its degradation. Ordinary nodes may send an entry request to other dominators, and the message complexity of this algorithm is $O(2\Delta)$. If a dominator is removed, whether the exiting node is a no-key dominator or a key dominator, the message complexity of this algorithm is the same as that when neighbors are added to the networks, i.e., $O(\Delta\cdot 2\Delta)=O(2\Delta^2)$.

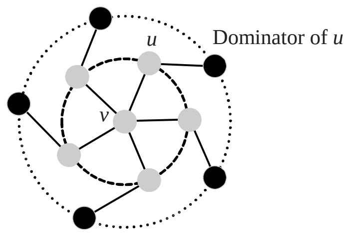

Fig.3 CMMA message complexity when nodes are added. Black dots are dominators and gray dots are ordinary nodes

Time complexity of the maintenance algorithm

For node entering, if there are dominators around the new node, no iterative process is needed for the maintenance algorithm, and the time complexity is $O(1)$. If there is no dominator around, dominators will pick the next hop dominator from neighbors to cover the new node, and the time consumed is within $O(\Delta)$. Then the time complexity of this maintenance algorithm is $O(\Delta)$.

For node exiting, if the exiting node is ordinary, $O(\Delta^2)$ iterations are needed for its dominator to determine whether there are other dominators near the neighbors of the remaining ordinary node. If a dominator is removed, ordinary nodes surrounding the exiting dominator perform an additional operation and, as shown above, the complexity is $O(\Delta)$. In conclusion, the time complexity of node exiting for the maintenance algorithm is $O(\Delta^2)$.

From the analysis above, this algorithm keeps a relatively small approximation factor and small message and time complexity, and thus has a satisfactory performance.

SIMULATION AND DISCUSSION

This paper validates the effectiveness of CMMA with ns-2 (http://nsnam.isi.edu/nsnam/index.php/Main_page). Through comparison with the CDS maintenance algorithm proposed by Wu and Li (1999), the advantages and disadvantages of the CMMA were analyzed for CDS maintenance in networks of different sizes. We chose networks of 10~50 nodes, and used the Random Waypoint Model (Johnson and Maltz, 1996) for node movement. The ratio between CDS and network size was investigated when the transmission diameter of nodes was 25 m.

As illustrated in Fig.4a, to validate the effectiveness of the maintenance algorithm, the average value of 10 results, obtained for each 30 min after simulations started, was taken (Unless otherwise stated, the following simulations would use the same environment parameters). In small networks of 10~30 nodes, the maintenance efficiency of the CMMA outperforms that of Wu's algorithm. This is because in small MANETs, the effect of optimization rules of Wu's algorithm is not obvious due to low node density. By comparison, CMMA has better performance, especially for networks of 20~30 nodes. Meanwhile, when a network enlarges, the effect of the optimization rules of Wu's algorithm becomes apparent and when the size reaches 60 nodes, the dominator ratio is lower than 0.5. In such two networks, the advantage of CMMA over Wu's algorithm becomes smaller. The CMMA dominator ratio is more stable, while the dominator ratio of Wu's algorithm fluctuates as the size of the network changes.

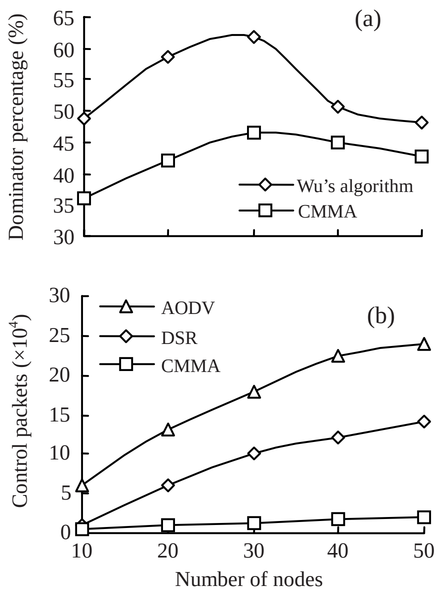

Fig.4 (a) CDS maintenance efficiency of CMMA and Wu's algorithms in different network sizes; (b) CDS maintenance overhead of three algorithms in different network sizes

CMMA has lower dominator percentage compared with Wu's algorithm. In Wu's algorithm the movement of a single mobile node may suppress many mobile nodes from transmitting or receiving their own packets. Also, if many mobile nodes in the network are in movement, the network topology may be greatly affected and thus the complete recalculation of a CDS with a large amount of message exchange is required (Yu and Chong, 2005). But in CMMA, the movement of a single mobile node only affects its neighbors. The upgrade of an ordinary node to a dominator only occurs when no dominators are found, and the ordinary node selected for upgrading should cover the maximum number of ordinary nodes. If a dominator is not required, due to the exiting of an ordinary node, it is degraded.

As CMMA is a local adjustment algorithm and it does not require the reconstruction of the whole CDS, a node needs only to communicate with the related neighbors with the simple protocol in its movement. So the control overhead of this maintenance algorithm is low and the simulation results in 600 s, compared with DSR (dynamic source routing) and AODV (ad hoc on-demand distance vector), are shown in Fig.4b.

It is seen that the control overhead is related to the number of nodes. The more nodes involved in the system, the more control packets that are needed to maintain this topology. But the overhead in CMMA is much lower than that in DSR and AODV.

Although this algorithm works well only in low speed situations, i.e., the degree of topology changes is low, the impact of the mobility speed still needs to be considered. Fig.5a shows the dominator percentage at 30 nodes at different mobility speeds.

The dominator percentage changes at different mobility speeds, but it does not vary a lot. The average value is nearly 45%, almost the same as that in Fig.4a. Although the topology of the system is slowly changing, the node number in the system remains unchanged. One node's entering may cause an increase in the dominator number, but also another node's exiting may cause a decrease in the dominator number, so the dominator percentage remains almost unchanged.

But the mobility speed must have an impact on the control overhead, because as the topology of the system changes, more messages are needed to maintain the CDS. The control overhead results are shown in Fig.5b.

More control packets are needed when the mobility speed is higher. The situation of an even higher mobility speed does not need to be considered, as the assumption is that we use this local adjustment algorithm only at a low mobility speed.

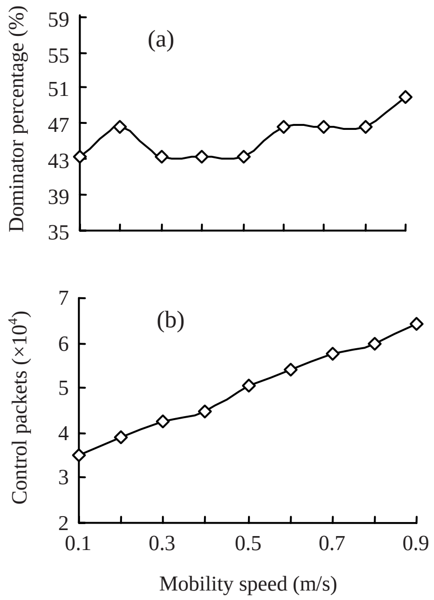

Fig.5 CMMA's CDS maintenance dominator percentage (a) and control overhead (b) at different mobility speeds at a fixed size of 30 nodes

CONCLUSION

In order to solve the problem of mobility management in MANETs, this paper introduces a CDS based mobility management algorithm, CMMA, and studies three cases of the algorithm: entering, exiting, and moving. Theoretical analysis and case discussion demonstrate that this algorithm has an ideal performance due to its influence on the approximation factor, message complexity, and time complexity. Thus the performance of this algorithm is much better than that of AODV, DSR and Wu's algorithm. In the future we would continue to focus on the application of CDS in MANETs and especially on the reliability of peer-to-peer transfer and flow control.

References

Alzoubi, K.M., Wan, P.J., Frieder, O., 2002. New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks. Proc. 35th Annual Hawaii Int. Conf. on System Sciences, Hilton Waikoloa Village, Island of Hawaii. IEEE Computer Society, Washington, DC, USA, p.3849-3855. [doi:10.1109/HICSS.2002.994519]

Burns, J., 1980. A Formal Model for Message Passing Systems. Technical Report TR-91. Computer Science Department, Indiana University, Bloomington, USA.

Das, B., Sivakumar, R., Bharghavan, V., 1997. Routing in Ad Hoc Networks Using a Virtual Backbone. Proc. Int. Conf. on Computer Communications and Networks, Las Vegas, NV, USA. IEEE Computer Society Press, Los Alamtos, CA, USA, p.1-20. [doi:10.1109/ICC.1997.605303]

Davies, V., 2000. Evaluating Mobility Models within an Ad Hoc Network. MS Thesis, Colorado School of Mines, Illinois, USA.

Haas, Z., 1997. A New Routing Protocol for Reconfigurable Wireless Networks. Proc. IEEE Int. Conf. on Universal Personal Communications, San Diego, CA, USA. IEEE, New York, NY, USA, p.562-566. [doi:10.1109/ICUPC.1997.627227]

Johnson, D.B., Maltz, D.A., 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. *In:* Imelinsky, T., Korth, H. (Eds.), Mobile Computing. Kluwer Academic Publishers, Dordrecht, the Netherlands, p.153-181. [doi:10.1007/978-0-585-29603-6_5]

Kim, W.I., Kwon, D.H., Suh, Y.J., 2001. A Reliable Route Selection Algorithm Using Global Positioning Systems in Mobile Ad Hoc Networks. Proc. IEEE Int. Conf. on Communications, Helsinki, Finland. IEEE Piscataway, NJ, p.3191-3195. [doi:10.1109/ICC.2001.937260]

Lee, J.K., Hou, J.C., 2006. Modeling Steady-state and Transient Behaviors of User Mobility: Formulation, Analysis, and Application. Proc. 7th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing, Florence, Italy. ACM, New York, NY, USA, p.85-96. [doi:10.1145/1132905.1132915]

Resta, G., Santi, P., 2008. WiQoSM: an integrated QoS-aware mobility and user behavior model for wireless data networks. *IEEE Trans. on Mob. Comput.*, **7**(2):187-198. [doi:10.1109/TMC.2007.70728]

Royer, E.M., Melliar-Smith, P.M., Moser, L.E., 2001. An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks. Proc. IEEE Int. Conf. on Communications, Helsinki, Finland. IEEE Piscataway, NJ, p.857-861. [doi:10.1109/ICC.2001.937360]

San, J., Puerto, R., 2007. Contention-aware Analysis of Routing Schemes for Mobile Opportunistic Networks. Proc. 1st Int. Mobile System Workshop on Mobile Opportunistic Networking, p.1-8. [doi:10.1145/1247694.1247696]

Santi, P., 2005. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, **37**(2):164-194. [doi:10.1145/1089733.1089736]

Spyropoulos, T., Psounis, K., Raghavendra, C.S., 2006. Performance Analysis of Mobility-assisted Routing. Proc. 7th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing, Florence, Italy. ACM, New York, NY, USA, p.49-60. [doi:10.1145/1132905.1132912]

Su, W., Lee, S.J., Gerla, M., 2001. Mobility prediction and routing in ad hoc wireless networks. *Int. J. Network Manag.*, **11**(1):3-30. [doi:10.1002/nem.386]

Tolety, V., 1999. Load Reduction in Ad Hoc Networks Using Mobile Servers. MS Thesis, Colorado School of Mines, Illinois, USA.

Wu, J., Li, H.L., 1999. On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks. Proc. 3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, p.7-14. [doi:10.1145/313239.313261]

Wu, J., Lou, W., Dai, F., 2006. Extended multipoint relays to determine connected dominating sets in MANETs. *IEEE Trans. on Comput.*, **55**(3):334-347. [doi:10.1109/TC.2006.40]

Yu, J.Y., Chong, P.H.J., 2005. A survey of clustering schemes for mobile ad hoc networks. *IEEE Commun. Surv. Tutor.*, **7**(1):32-38. [doi:10.1109/COMST.2005.1423333]