移动自组网中基于分簇的数据复制算法*

郑 静+, 卢锡城, 王意洁

(国防科学技术大学 计算机学院,湖南 长沙 410073)

A Clustering-Based Data Replication Algorithm in Mobile Ad Hoc Networks for Improving Data Availability

ZHENG Jing+, LU Xi-Cheng, WANG Yi-Jie

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4573666, E-mail: zhengjing621@hotmail.com

Received 2004-05-28; Accepted 2004-11-13

Zheng J, Lu XC, Wang YJ. A clustering-based data replication algorithm in mobile ad hoc networks for improving data availability. *Journal of Software*, 2005,16(8):1474–1483. DOI: 10.1360/jos161474

Abstract: In Mobile Ad Hoc Networks (MANET), network partitioning can cause sudden and severe disruptions to ongoing data accesses, and consequently data availability is decreased. A new distributed clustering algorithm is presented in this paper for dynamically organizing mobile nodes into clusters in which the probability of path availability can be bounded. Based on this clustering algorithm, a data replication algorithm is proposed to improve data availability. Theoretic analysis indicates that the algorithm has a proper complexity. Simulation results show that the clusters created by the clustering algorithm have desirable properties and the data availability is improved effectively by the clustering-based data replication algorithm.

Key words: MANET; replication; clustering; data availability; clique

摘 要: 在移动自组网络中,网络分割现象可能频繁发生,从而降低了数据的可用性.提出了一种新的分布式分簇算法来组织移动节点,算法保证簇内任意两点间路径的可用概率都大于某个确定的界.在此基础上提出了基于稳定路径分簇的数据复制策略,以提高在出现链路断接甚至网络分割时的数据可用性.对算法进行了理论证明和实验分析,实验结果表明,由分簇算法构造的簇能够满足我们所要求的特性,并且基于分簇的数据复制算法在移动自组网环境中有效地提高了数据的可用性.

关键词: MANET;复制;分簇;数据可用性;最大子图

中图法分类号: TP393 文献标识码: A

---

* Supported by the National Natural Science Foundation of China under Grant No.90412011 (国家自然科学基金); the Foundation for the Author of National Excellent Doctoral Dissertation of PR China under Grant No.200141 (全国优秀博士学位论文作者专项资金)

ZHENG Jing was born in1970. She is a Ph.D. candidate of National University of Defense Technology. Her current research areas are mobile computing, etc. **LU Xi-Cheng** was born in 1946. He is a professor of National University of Defense Technology and a CCF senior member. His current research areas are computer networks, etc. **WANG Yi-Jie** was born in 1971. She is an associate professor of National University of Defense Technology. Her current research areas are database, computer network, etc.

1 Introduction

In Mobile Ad Hoc Networks (MANET), mobile nodes move freely, and disconnections of links occur frequently. This may cause frequent network partitioning, which poses significant challenges to data management in MANET environment.

Data availability is referred to the ratio between the number of requests to a data object and that of responses. To improve data availability during network partitioning, a clustering-based data replication algorithm is proposed in this paper. The basic idea of the algorithm is that mobile nodes are dynamically organized into clusters and the data objects will be replicated in some clusters where these data objects are requested. The clustering mechanism is used to predict the network partitioning, and clusters should present the following desirable properties: (1) Each cluster is connected and there is at least one stable path between any pair of nodes in a cluster. The path stability can guarantee the data availability in a cluster. (2) The number of clusters should be limited. Because more clusters will cause more replicas which will result in high storage complexity. (3) Two clusters should overlap appropriately. On one hand, all common nodes of two clusters will have to maintain cluster state and carry intra-cluster traffic for both the clusters. Consequently overhead of these nodes will be increased. On the other hand, overlap between clusters can decrease the number of replicas. Hence, it is necessary to make a trade-off between them. (4) Cluster should be stable across node mobility. The clustering algorithm should adapt to the new nodes joining in the network, the existing nodes disappearing from the network and the nodes moving.

Clustering-based routing algorithm has been an extensive research topic. However, only a few researches have focused on the clustering algorithm for data availability. A lot of clustering algorithms are used to support routing in MANET, forming *cluster-based hierarchical routing* or *backbone-based routing*[1−3]. A cluster selection method is proposed in Ref.[1], which uses the link structure, combining local and relative densities. In Ref.[2], clusters are created using spanning tree. In Ref.[3], another clustering method based on *domination set* in a graph is proposed. The other protocols attempt to reduce energy consumption by clustering algorithms[4,5]. However, because the goals of aforementioned clustering algorithms are different from ours, the requirements for clustering are different too. Reference [6] proposes a method to ensure that the centralized service (such as web servers) is available to all nodes during network partitioning. But service availability could not be guaranteed any more in the widely employed mobility model, Random Waypoint Mobility Model[7]. Hare’ work[8] focuses on data accessibility in MANET, but does not consider connection stability. Reference [9] presents a stable path based clustering routing algorithm, but it is very complicated to compute path availability, and the formed clusters could not meet some properties described above.

The rest of the paper is organized as follows. In Section 2 the problem statement is posed in a graph theoretic framework. In Section 3 a distributed algorithm to construct *α-Stable Graph* is presented in detail. In Section 4, the clustering-based data replication algorithm is described. In Section 5, the algorithms are evaluated through simulations. And finally in Section 6, the summary and some future work are presented.

2 Problem Statement

Mobile ad hoc networks are comprised of mobile nodes communicating via radio. If two mobile nodes are too far apart to communicate directly, intermediate nodes can relay their messages. A widely employed model is used, so-called *Unit Disk Graph* (UDG) $G(V,E)$, to study ad hoc networks: vertices in G are nodes that are located in the Euclidean plane and assumed to have identical transmission radio; an edge between two vertices, representing that the corresponding two nodes are in mutual transmission range, exists iff their Euclidean distance is not greater than the maximal transmission distance.

Definition 1. *α-stable neighbor*: During the time slice Δt, the connectivity probability between node i and its neighbor j is denoted as $Pr(link_{i,j})$. Node j is a *α-stable neighbor* to i, if $Pr(link_{i,j})$ is larger than α (α is the threshold). Let $SN(i)=\{j|\ Pr(link_{i,j})\}>\alpha\}$ be the *α-stable neighbors set* of i.

Definition 2. *α-stable path*: The path $Path_{m,n}$ between nodes m and n is comprised of links $link_{i,j}$, $link_{i,j}\in Path_{m,n}$, and the *path availability* is denoted as $Pr(Path_{m,n})$. According to the assumption of independent link failures, the path availability is given by $\Pr(path_{m,n})=\prod_{(i,j)\in path_{m,n}}\Pr(link_{i,j})$. $Path_{m,n}$ is a *α-stable path*, if $Pr(Path_{m,n})>\alpha$.

Definition 3. *α-stable path nodes set*: *α-stable path nodes set* for i, $SP(i)$, for every node j in nodes set $SP(i)$, if there is at least one *α-stable path* between nodes i and j. $SN(i)\subseteq SP(i)$.

Definition 4. *α-stable graph* G_α: The *α-stable graph* of graph G, $G_\alpha(V,E')$, vertices in G_α are the same as those in G, edge $(m, n)\in E'$ iff there is at least one *α-stable path* between nodes m and n, i.e. $Pr(Path_{m,n})>\alpha$.

Figure 1 shows the UDG of a mobile ad hoc network, and Fig.2 is the G_α for Fig.1. In Fig.2, the dashed line between two nodes indicates that there is a α-stable path between them, and the solid line between two nodes indicates that they are stable neighbors each other.

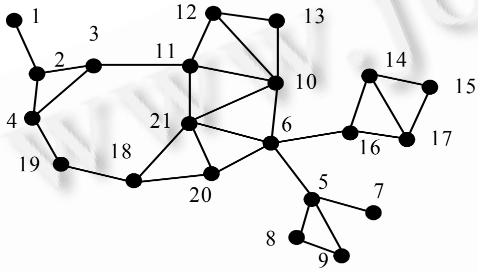


Fig.1 UDG

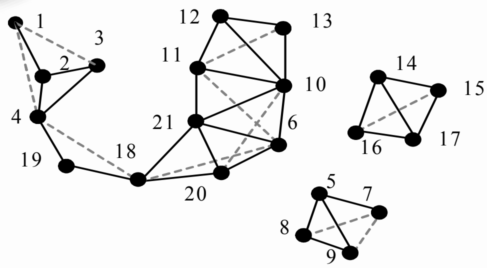


Fig.2 *α-stable graph G*

In G_α, mobile nodes are separated into several disconnected sub-graphs, i.e. there is no stable paths between two nodes that belong to two different sub-graphs; therefore, the probability of network partitioning between the two sub-graphs is high. Our goal is to construct clusters where the path availability between any two nodes is bigger than α, and the size of each cluster is as large as possible. Therefore the clustering problem can be mapped to the *maximum clique problem* (MCP) in graph G_α. However, the cluster in our algorithm must satisfy property (1). The graph satisfying the above two requirements simultaneously may not exist. In Fig.3, the maximum clique in G_α is composed of nodes A, C, E, and F, but is not connected in graph G. Therefore, we create clusters with cliques in G_α which is connected in G.

3 Distributed Algorithm to Construct *α*-Stable Graph (*α*-SGA)

3.1 The connectivity probability between neighbors

Assuming that link disconnection is caused by node mobility, the distance between two neighbors is used as the metric of connectivity probability between neighbors. We assume that the effective wireless communication range of node i is the disk D_i for which i is its center and R is its radius. The distance between node i and its neighbor j is d (which can be obtained from GPS or any other position-info service), and all nodes move in the Random Waypoint Mobility Model, i.e. node speed is uniformly distributed between $(0,Vel_{\max})$ and the direction is uniformly distributed over $(0,2\pi)$. Therefore after time slice Δt, node j is uniformly distributed in disk D_j whose center is j and radius equals to $2Vel_{\max}\Delta t$. Finally, the connectivity probability between i and j in time slice Δt can be described as:

$$\Pr(link_{i,j}) = \frac{S(D_i, R) \cap S(D_j, 2Vel_{\max}\Delta t)}{S(D_j, 2Vel_{\max}\Delta t)} \tag{1}$$

As shown in Fig.4, $S(D_i)$ means the area size of disk D_i. If $Pr(link_{i,j})>\alpha$, j is a stable neighbor of i.

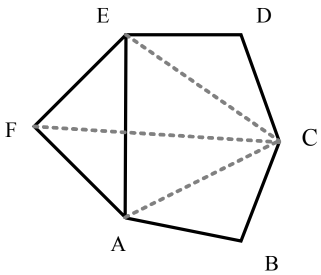


Fig.3 A clique in G_α

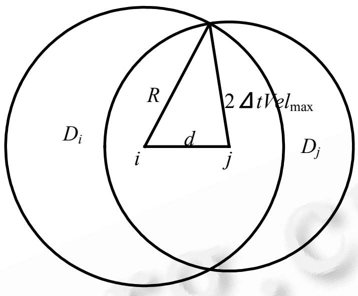


Fig.4 The connectivity probability

Δt is a systemic parameter, which is the time interval to estimate the validity of connectivity probability. According to different distances to center i, the disk is partitioned into n cirque regions, $H_1,H_2,\ldots,H_n$, from nearby to far away. Connectivity probability between every cirque region and center i equals to the average value of the connectivity probability between every point in this region and center i. If node j is located in one region H_k of D_i, $k\in\{1,2,\ldots,n\}$, $Pr(link_{i,j})$ is the connectivity probability between H_k and i. In this way the distance change within a cirque region will not cause the change of connectivity probability, and the connectivity probability can be obtained by simply looking up a pre-setting region connectivity probability table with the *ID* of the region.

3.2 Distributed α-stable graph algorithm

A fully distributed algorithm α-SGA (α-Stable Graph Algorithm) is proposed to construct α-stable graph by only exchanging information between neighbors. On each mobile node i, a global *ID* $id(i)$ and the following local states are maintained:

1) Profile of measurement of distance to all neighbors, $P(i)$. Every mobile node i measures the distance to each neighbor j and calculates $Pr(link_{i,j})$ respectively, which is recorded in 2-tuple $\langle id(j),Pr(link_{i,j})\rangle$.
2) The set of stable neighbors, $SN(i)$. This set is constructed by nodes which satisfy Definition 1. In $SN(i)$, the *ID* of i's every stable neighbor j and its connectivity probability to i are recorded in 2-tuple $\langle id(j),Pr(link_{i,j})\rangle$.
3) The set of α-stable path nodes, $SP(i)$. This set records all such nodes that satisfy Definition 3. For each $j\in SP(i)$, a 3-tuple $\langle id(j),id(N(Path_{i,j})),Pr(Path_{i,j})\rangle$ is maintained. If multiple paths exist between nodes i and j, $Pr(Path_{i,j})$ is the maximum path availability among that of these paths. $N(Path_{i,j})$ is the next hop node from i to j in this selected path with $Pr(Path_{i,j})$.
4) The $SN(j)$ and $SP(j)$ of stable neighbor j, $PN(j)$. For each α-stable neighbor j, i maintains $SN(j)$ and $SP(j)$ of j. Each $PN(j)$ has a 3-tuple $\langle id(m),Pr(Path_{j,m}),Pr(Path_{i,j,m})\rangle$, $m\in SP(j)\cup SN(j)$. $Pr(Path_{i,j,m})$ is the availability probability of path from node i to m, passing j.

The distributed algorithm α-SGA allows mobile nodes to find their stable neighbors and α-stable path nodes by exchanging information with their stable neighbors and construct their α-stable graph G_α. Algorithm is described as follows:

1) Measurement. Every mobile node i calculates the connectivity probability to each neighbor. For each $j\in P(i)$, if $Pr(link_{i,j})>\alpha$ then $SN(i)=SN(i)\cup\{j\}$. During initiation, $SP(i)=SN(i)$.
2) Exchange. Node i exchanges its $SN(i)$ and $SP(i)$ with all its stable neighbors. It sends 2-tuples $\langle id(m), Pr(Path_{i,m})\rangle$ to all its stable neighbors, where $m\in SP(j)\cup SN(j)$. Upon receiving information from the stable

neighbors, i constructs its $SP(i)$ and $SN(i)$.

3) Update. When the connectivity probability between i and its neighbor j has been changed, or i receives update information from other stable neighbors, state information $P(i)$, $SN(i)$, $SP(i)$ and $PN(j)$ should be updated. Algorithm's detail is described as follows.

```
if connectivity probability between i and its neighbor j has been changed
      then update P(i);
      if j∈SN(i) and Pr(link_{i,j}) ≤α then SN(i):= SN(i)/{j},delete all info in PN(j); endif;
      if j∈SN(i) then update the corresponding probability value in SN(i) and PN(j);
            add j to update message queue;
      endif;
      if j∉SN(i) and Pr(link_{i,j}) >α
            then SN(i):= SN(i) ∪ {j}, SP(i):= SP(i) ∪ {j};
                  update SN(i) and PN(j); add j to update message queue;
      endif;
      update SP(i): for all m∈SP(i), recalculate id(N(Path_{i,m})) and Pr(Path_{i,m});
      propagate the update message to all stable neighbors except j;
endif;

if i receives update the message from stable neighbor j
      then do similar update to SP(i), SN(i) and PN(j) as above;
            propagate update the message to all stable neighbors except j;
endif;
```

By exchanging information with stable neighbors, α-SGA makes every node i construct a set $SP(i)$ containing all the nodes that have α-stable path to i. Thus, there is an edge between node i and each node in $SP(i)$ in G_α. Because messages are exchanged between pairs of stable neighbors, and the change of connectivity probability is propagated only among its α-stable path nodes, the communication overhead to maintain G_α is dependent on the size of $SP(i)$. Because usually the size of $SP(i)$ is much smaller than the total size of the network and only the incremental update is needed, the communication overhead of this algorithm is acceptable in MANET environment.

3.3 Heuristically connected clique of α-stable graph algorithm

The CCGA (Connected Clique Graph Algorithm) algorithm is proposed to construct clique in G_α, based on sequential greedy heuristic method[10]. The clique constructed as follows contains node i, and is connected in graph G.

```
    procedure local-search-add-move ( output: Clique (i); input: SP(i), SN(i) )
    begin
1       PA:=SP(i), Clique(i):= i;
2       repeat
3               PA′:= ∅;
4               for all j∈Clique(i)
                      PA′: =PA′∪ (PA∩SN(j));
                endfor;
5               search node v∈PA ′& SP(v)∩PA′=max_{k∈PA′}{SP(k)∩ PA′}
6               if for all u∈Clique (i), v∈SP(u) is true then Clique(i):=Clique(i) ∪{v}; endif;
                   PA:=PA/{v};
                if k∈SN(v) ∩PA and SN(k) ∩PA= ={v} then PA: =PA/{k}; endif;
7       until PA= =∅;
8       return Clique(i);
    end;
```

In the algorithm, $Clique(i)$ is the node set of clique containing node i in G_α; PA is the set of all candidate nodes to be added to $Clique(i)$; PA' is composed of all the candidate nodes which at least have a stable neighbor in $Clique(i)$. In line 6 of the algorithm, if candidate node v at least has a stable path to every node in $Clique(i)$, then node v is added to $Clique(i)$, so $Clique(i)$ is a clique containing i in G_α. Line 5 searches the node which has maximum number of edges to other candidate nodes. The algorithm complexity is $O(|SP(i)|^3)$.

4 The Clustering-Based Data Replication Algorithm

4.1 α-Stable path based clustering algorithm

The distributed α-SPCA (α-Stable Path Based Clustering Algorithm) is based on the algorithms presented in Section 3. The α-SPCA algorithm has two phases: Cluster Creation and Cluster Maintenance. The cluster creation is invoked when network is in its initialization phase. The cluster maintenance is an inexpensive phase of the algorithm that handles node mobility leading to the local change of the existing cluster.

4.1.1 Cluster creation

1) Every node i obtains its $SP(i)$ by exchanging information with its stable neighbors.
2) If node i does not belong to any cluster and is the least ID node among its stable neighbors which are not clustered, node i becomes a cluster head.
3) Cluster head i executes the CCGA algorithm to find $Clique(i)$ in G_α. The nodes in $Clique(i)$ compose a cluster whose cluster head is node i and Cluster ID (CID) equates to the ID number of i.
4) Node i which is not clustered repeats steps 2) and 3) until every node belongs to at least a cluster.

4.1.2 Cluster maintenance

Once cluster creation phase generates a set of clusters, the cluster maintenance phase is invoked to perform some small changes to handle node mobility as the new nodes join and the existing nodes leave a cluster.

1) Node joins: When cluster head i finds any new stable path node j, i checks every cluster member in the cluster if node j is in its stable path nodes set and if node j is in some cluster member's stable neighbors set. If so, node j joins the cluster.
2) Node leaves: If there is no longer any stable path between a pair of nodes in the same cluster, the node that has the larger *ID* and is not the cluster head leaves the cluster. If the leaving node does not belong to any cluster and can not join other clusters, it executes CCGA algorithm to construct a new cluster. If the node has no stable neighbor, it is called orphan node. The orphan node is a very unstable node in the network.
3) Cluster removes: If all of the cluster members in a cluster C_i belong to multiple other clusters, the cluster head of C_i sends the apply for removing C_i to cluster heads whose cluster cover some members of C_i. If all these cluster heads agree on removing C_i, the cluster head of C_i declares that C_i is vanished. In case, some nodes do not belong to any cluster when multiple clusters are removed simultaneously. If two clusters have the identical members, the cluster that has the larger CID is removed.

The clusters created by the α-SPCA algorithm have the following properties:

Property 1. The path availability between any pair of nodes in the same cluster is larger than α.

Proof. In the α-SPCA algorithm, all nodes in one cluster compose a clique in the G_α, i.e. there is an edge between any pair of cluster members in the G_α. According to the definition of edge in the G_α, the path availability between any pair of cluster members is bigger than α.

Property 2. Each cluster is connected in graph G.

Proof. While executing CCGA algorithm to create clusters, the newly joined node is a stable neighbor node of some existing nodes in the cluster. Therefore, each cluster is connected in graph G.

Property 3. Each node i belongs to at most $|SP(i)|$ clusters.

Proof. Cluster heads are difference from each other, and the cluster head whose cluster includes node i is in the set $SP(i)$. Therefore, each node i belongs to at most $|SP(i)|$ clusters.

Example: In the topology showed in Fig.1, after executing the first iteration of the α-SPCA algorithm, the algorithm creates clusters C_1, C_5, C_6, C_{14} and C_{18} (showed in Fig.5). After executing the second iteration of the algorithm, the algorithm creates clusters C_{12} and C_{19} (showed in Fig.6).

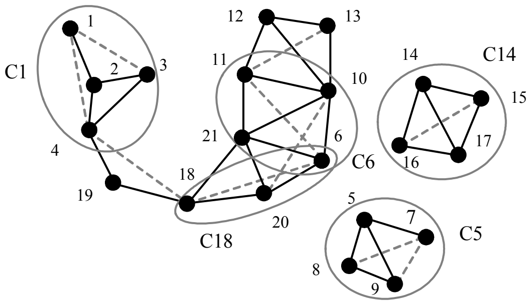


Fig.5 Result of the first iteration

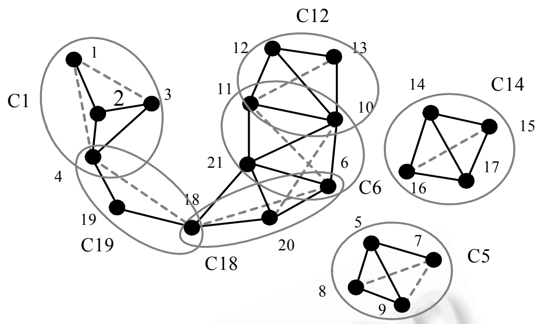


Fig.6 Result of the second iteration

4.2 Clustering-Based data replication algorithm

Based on the *α*-SPCA algorithm, the network is clustered into several clusters and network partitions often present between clusters, especially clusters without overlap. The basic idea of CDRA (Clustering-based Data Replication Algorithm) proposed in this paper is that the requested data object in the clusters is replicated to prevent deterioration of data accessibility at the point of network partitioning. The CDRA algorithm composed of the replica allocation and replica consistency is described as follows:

4.2.1 Replica allocation

1) Every cluster head maintains states of all other cluster heads in the networks. When a node requests to access a data object, the node broadcasts the access request in the whole cluster C_i that the node belongs to. If there are some replicas of the data object in C_i, the closest replica node serves the access request.
2) If there is no replica of the requested data in C_i, the request is propagated from the cluster head of C_i to all other cluster heads. If there is replica in some cluster, the cluster head sends the data to the cluster head of C_i .
3) The cluster head of C_i sends the data to request nodes. A node in C_i , which has request to the data object, is chosen to replicate the data object. Nodes in multiple clusters have the priority to be chosen as replica nodes (such as nodes 4, 6 and 18 in Fig.6). Because the path availability between any pair of nodes in a cluster is bigger than *α*, the data availability in a cluster is bigger than *α*.
4) The adaptive replica allocation algorithm (ARAM)[11] proposed by us is used to allocate the replica in the inter-cluster. The ARAM algorithm dynamically adjusts location and number of the replicas adapting to the nodes motion and the change of read-write pattern.

4.2.2 Replica consistency

The CDRA algorithm can be combined with some replica management protocols that provide weak or strong consistency guarantees. For example, updates can be disseminated to all cluster heads whose cluster has the replica by the probabilistic reliable group communication protocols, and then are forwarded to the replica nodes in the cluster. Because of the hierarchical control, the communication overhead of replica update will be decreased.

5 Simulation and Analysis

To evaluate the performance of our algorithm, extensive simulations have been performed. Because our main concern is to improve data availability in the presence of frequent network partitioning events, simulations are performed in a sparse MANET environment. 250 nodes are initially randomly activated within a bounded region of 5000m×5000m, and transmission range R=0.5 km. All nodes move in the Random Waypoint Mobility Model. A range of node mobility with mean speeds *MVel* between 5 to 10 m/s is simulated. During each epoch the speeds of each mobile node are uniformly distributed over (0,2*MVel*), and the direction is uniformly distributed over (0,2π). The pause-time is 4s. Two values of the path availability threshold α are used, 0.4 and 0.6. The systemic parameter Δt is 20s. In these simulations, the read requests issued by every node are uniformly distributed from 0 to 20, the ratio between read requests and write requests is 10:1, and there is only 1 data object to be replicated.

5.1 Comparison properties of the cluster with (*α*,*t*)-cluster algorithm

We compare the properties of the clusters in α-SPCA with those in (α,t)-*Cluster* (t=1 minute) algorithm[9]. The results show that the clusters created by α-SPCA can achieve the desirable properties described in Section 1.

Figure 7(a) shows the effects of mobility on mean cluster size. The results show that the α-SPCA clustering algorithm adapts cluster size to node mobility and threshold α. Another observation indicates that the mean cluster size in α-SPCA is bigger than that in (α,t)-*Cluster* algorithm. The reason is that in the α-SPCA algorithm cluster head adds nodes as many as possible into its cluster and allows proper overlaps among clusters. But in (α,t)-*Cluster* algorithm every node only belongs to one cluster, and cluster combination is not taken into account. Accordingly, the number of cluster relating to the size of cluster, mean number of cluster in α-SPCA is smaller than that in the (α, t)-*Cluster* algorithm. According to our replication strategy, the more clusters, the more replicas may be required. Therefore, fewer replicas are required in the α-SPCA than in (α,t)-*Cluster* algorithm.

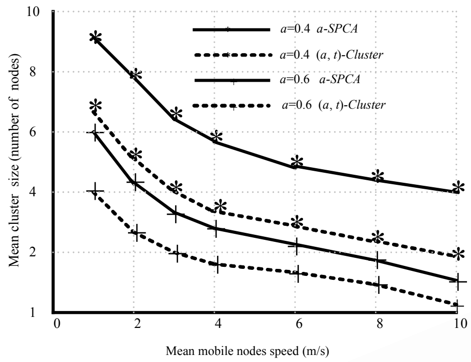
Figure 7(b) shows the effects of mobility on the probability that a node is clustered. Because the α-SPCA algorithm allows a node belong to multiple clusters, the probability that a node is clustered is very high even at high speed, and is higher than that in (α,t)-*Cluster* algorithm. This is a desirable property, because while a node does not belong to any cluster, the Cluster Creation must be invoked, namely, a high probability that nodes are clustered implies a low cost for cluster creation.

Figure 7(c) demonstrates the desirable stability property of the cluster. Cluster survival time is measured by taking the amount of elapsed time of each currently active cluster. Thus, it represents the lifetime of cluster. The chart implies that the speed affects the stability of cluster topology. A cluster is removed only when all its members belong to other clusters in the α-SPCA, so the cluster is stable and the mean cluster survival times can be accepted in terms of system performance. The great jump in the point at 10m/s in (α,t)-*Cluster* algorithm is due to the very low probability of a node actually being clustered (referring to Fig.7(b)).

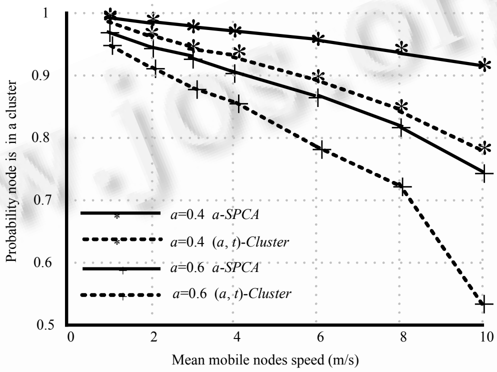
5.2 Comparison data availability with ARAM algorithm

We compare ARAM algorithm and Static Replica Allocation algorithm (i.e. SRA, replicas are distributed on fixed nodes, and the replica allocation scheme doesn't change during the whole process of simulation) with CDRA algorithm. The data availability of the two approaches is plotted in Fig.8. With nodes moving, the data availability changes whenever the network partitions or merges. Because the first two approaches do not consider the effects of network partitioning on data access, Fig.8(a) shows more periodic and frequent rises and drops in data availability than Fig.8(b). Predicting that the network partitions often present between clusters, CDRA replicates data object in every cluster that requests this data object. Thus, CDRA algorithm greatly improves the data availability to 96.5%.

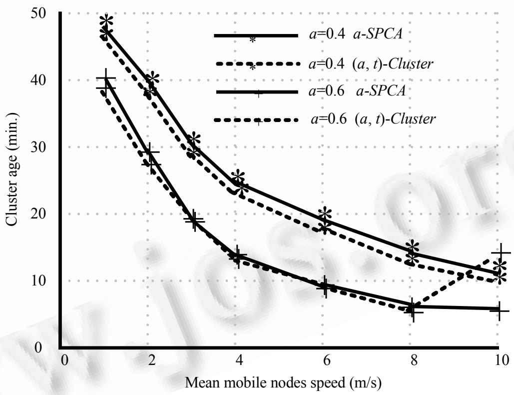
We can draw two conclusions from simulations. First, in CDRA algorithm, dynamic allocation of the replica effectively improves data availability, since the allocation is decided based on the changing network topology and the prediction of the network partitions. Second, our distributed algorithm only cannot ensure the data availability of those clusters in which all nodes have not requested the data object before network partitioning occurs. Therefore the data availability is interrupted by chance in Fig.8(b).



(a) Cluster size



(b) Cluster probability

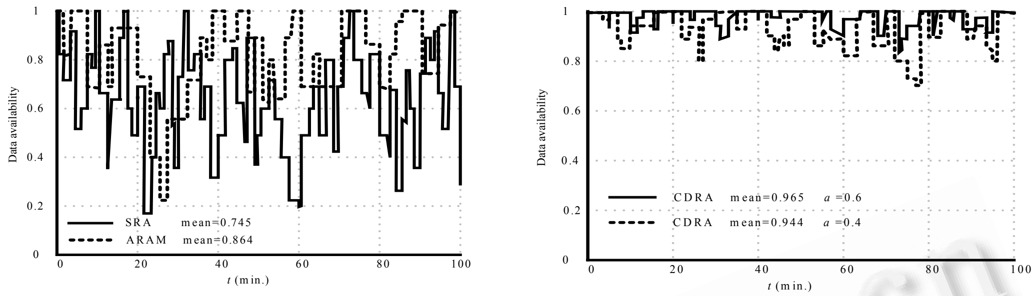


(c) Cluster survival

Fig.7 Comparison of cluster properties for different algorithms

6 Conclusions

In this paper, a distributed clustering-based data replication algorithm is proposed to address the data availability in partitionable ad hoc networks. A fully distributed clustering algorithm is presented based on the path stability between pairs of nodes, which is used to predict the network partitioning. The data objects are replicated in the clusters that request these data objects. Simulations show that the clusters created by our clustering algorithm have desirable properties and the replication algorithm can greatly improve the data availability under network partitioning. In the future work, the algorithm should be improved to deal with the replica conflict resolution and reconciliation problem in the MANET environment.



(a) SRA and ARAM algorithms (b) CDRA algorithm

Fig.8 Comparison of data availability for different algorithms

References:

[1] Virtanen SE, Nikander P. Local clustering for hierarchical ad hoc networks. In: Proc. of the WiOpt2004: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks. California: IEEE Computer Society Press, 2004. 404−405.

[2] Banerjee S, Khuller S. A clustering scheme for hierarchical control in muti-hop wireless networks. In: Proc. of the IEEE INFOCOM2001. California: IEEE Computer Society Press, 2001. 1028−1037.

[3] Chen YP, Liestman AL. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In: Proc. of the MOBIHOC2002. New York: ACM, 2002. 165−172.

[4] Kawadia V, Kumar PR. Power control and clustering in ad hoc networks. In: Proc. of the IEEE INFOCOM. California: IEEE Computer Society Press, 2003. 459−469.

[5] Younis O, Fahmy S. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In: Proc. of the IEEE INFOCOM2004. California: IEEE Computer Society Press, 2004.

[6] Wang K, Li B. Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks. In: Proc. of the INFOCOM2002. California: IEEE Computer Society Press, 2002. 1089−1098.

[7] Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research. Wireless Communication & Mobile Computing (WCMC): Special Issue On Mobile Ad Hoc Networking: Reach, Tends and Applications, 2002,2(5):483−502.

[8] Hare T. Effective replica allocation in Ad hoc networks for improving data accessibility. In: Proc. of the IEEE INFOCOM2001. California: IEEE Computer Society Press, 2001. 1568−1576.

[9] McDonal AB, Znati T. A mobility based framework for adaptive clustering in wireless Ad-hoc networks. IEEE Journal on Selected Areas in Communication, 1999,17(8):1466−1487.

[10] Bomze IM, Budinich M, Pardalos PM, Pelillo M. The maximum clique problem. Handbook of Combinatorial Optimization. Kluwer Academic Publishers, 1999. 27−39.

[11] Zheng J, Wang YJ, Lu XC, Yang K. A dynamic adaptive replica allocation algorithm in mobile ad hoc networks. In: Proc. of the IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom2004). California: IEEE Computer Society Press, 2004. 65−70.