文章编号:1671 - 4598(2023)12 - 0265 - 06 DOI:10. 16526/j. cnki. 11-4762/tp. 2023. 12. 039 中图分类号:TP3 文献标识码:A

基于移动性预测分簇算法的仿真设计

刘 云[1], 黄润根[2]
(1. 南通大学杏林学院, 江苏 南通 226236; 2. 南通大学 信息科学技术学院, 江苏 南通 226019)

摘要: 针对车联网的特点, 结合节点间的节点相对移动性和节点的剩余生存时间, 提出了簇头选择权值参数 M 来表示节点作为簇头的能力强弱, 并基于这个参数设计出一种簇生成算法; 该簇生成算法的主要思想是比较每个节点的 M 值, 然后选出 M 值最小的节点成为簇头并生成簇, 这样可以保证簇的稳定性; 但是, 当一个簇内成员个数过多过少时都会使簇的存在变得低效, 针对这一现象, 提出了簇合并和簇分裂机制; 该机制可以在降低网络通信开销的同时, 保证簇的稳定运行; 仿真结果表明, 此算法在车联网场景中性能良好, 簇结构稳定, 孤立节点数量较少, 节点能够快速地进行信息交互并生成簇。

关键词: 车联网; 簇头选择权值参数; 簇生成与维护; 簇稳定性

Simulation and Design of Clustering Algorithm Based on Mobility Prediction

LIU Yun[1], HUANG Rungen[2]
(1. Xinglin College of Nantong University, Nantong 226236, China;
2. School of Information Science and Technology, Nantong University, Nantong 226019, China)

Abstract: According to the characteristics of VANET, and considering the relative mobility of nodes and the lifetime of nodes in VANET. The metric M of The Cluster Head Selection Metric is proposed. It represents the ability of nodes as the cluster head. Based on this metric, the cluster formation is designed. The main idea of the cluster formation algorithm is to compare the M values of each node. Then, selecting a node with the smallest M value to become cluster heads and form the cluster. This method can ensure the stability of the cluster structure. However, when there are too many or few nodes in the cluster. It is inefficient to the cluster. According to this phenomenon, the cluster merging mechanism and the cluster splitting mechanism are proposed. The mechanism can reduce the overhead of communication and ensure the stability of the cluster structure. The simulation results show that the algorithm has a good performance in VANET. It can reduce the number of isolated nodes and the cluster structure will be more stable. Nodes can exchange information quickly. Then, they will form the cluster rapidly.

Keywords: VANET; cluster head selection metric; cluster formation and maintenance; cluster stability

0 引言

车联网[1]是一种特殊的移动自组织网络 (MANET, mobile adhoc network), 是在其基础上发展而来的, 以车辆作为网络节点的自组织网络。车联网 (VANETs, vehicle Ad Hoc networks) 作为智能交通系统[2]的重要组成部分, 是一种快速移动的宽带多跳无线移动自组织网络。它可以通过车辆上的无线通信终端—车载单元 (OBU, on-board units) 实现车辆和车辆之间的通信 (V2V, vehicle to vehicle) 以及车辆与路边基础设施 (RSU, roadside units) 之间的通信 (V2I, vehicle to infrastructure), 完成智能交通系统中的信息交互, 是一个有着巨大发展潜力的新兴领域[3-6]。

在车联网中, 车辆节点的数量很多, 并具有速度快、移动性强等特点, 这可能会造成较大的通信开销并使网络的处理能力变弱, 传统自组织网络的分簇算法已不再适用[7-8]。为了解决这一问题, 提高通信的效率和可靠性, 国内外科研所研究验证了各种类型的分簇算法[9-11]并进行了进一步的优化[12-14]。Basu 等设计了一种 MOBIC[12]算法, 通过收到的两个连续消息的功率来确定节点的移动性, 并选择相对于邻居节点移动变化最小的节点作为簇头, 提高簇的稳定性, 但是 MOBIC 算法没有考虑到车辆节点的运动还与道路状态有关系。ALM 算法[13]在一定程度上对 MOBIC 算法进行了修改, 它使用 GPS 设备来获取节点位置信息, 通过两个节点的距离比来判断节点的移动性。MPECS[14]是基于区域的集群分簇算法, 它通过预测车辆在某个区域内的剩余寿命来选择簇头, 保证了簇的区域生存时间。除此之外, 还存在着基于方向[15]的分簇算法以及一些根据不同场景而选择不同分簇方式的算法, 如直道高速公路环境[16], 城市路口环境[17]、荒漠环境[18]等。

结合车联网的网络特性和高速公路的特点, 提出了基于移动性预测的分簇算法, 该算法能够适用于高速公路场景, 可以较好地减少网络中孤立节点的数量、提高簇结构的稳定性。该分簇算法由簇生成和簇维护两部分组成, 在簇生成算法中, 设计了一个以节点间的相对移动性和节点剩余在网时间为根据的簇头选择权值参数 M, 并通过此参

收稿日期:2023 - 10 - 11; **修回日期**:2023 - 11 - 08。
作者简介:刘 云(1984 -),女,硕士研究生,讲师。
引用格式:刘 云,黄润根. 基于移动性预测分簇算法的仿真设计[J]. 计算机测量与控制,2023,31(12):265 - 270,283.

数进行簇头的选择，簇维护包括周期性的对簇进行维护更新以及簇的合并和分裂。此算法可以在提高簇结构稳定性的情况下降低网络的通信开销，最后在仿真平台上对这一算法的性能进行数据分析验证。

1 系统模型

本论文设计的分簇算法的系统模型如图 1 所示，是以高速道路的场景为基础创建的。假设所有的车辆节点都装备着全球定位系统（GPS，global positioning system），并通过 GPS 获取到自己的位置、速度、方向等信息。此外，假设接入网络中的所有车辆节点发送的数据包的功率是相同的，并可以一直收发消息。

如图 1 所示，在车联网中可以存在着很多不同的簇，图中同一矩形框中的所有车辆节点属于同一个簇。每个簇分别由一个簇头节点和若干个簇成员组成，簇头负责簇内及簇间成员的信息交互及转发，孤立节点不属于任何一个簇，独立的存在于网络之中。另外，算法规定：只有在同一方向上行驶的车辆节点才能加入到同一个簇中，并且当孤立节点申请并加入到某个簇头所在的簇后，它便脱离孤立节点的状态并成为该簇的簇成员。

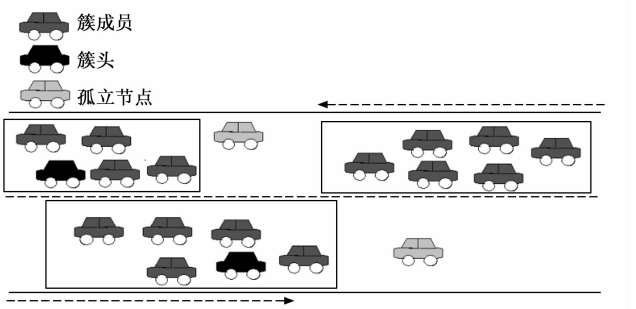

图 1 系统模型

在车联网中，由于其车辆节点具有高速移动性，车辆的位置时刻发生着变化，因此，车辆间的相对距离也在不停的变化，当簇成员与簇头的距离超过一定的界限时，就会因为其超过了簇头的通信范围而从簇中脱离退出。另外，孤立节点将会周期性的广播 Beacon 消息来进行消息交互，如果发现网络中存在簇就试图加入到簇中。同时，如果簇头在网络中发现了孤立节点，就向其发送 Invite 消息并等待它的回复。

如果网络中的存在的孤立节点数量过多，将会导致车联网的网络通信开销变大，降低网络的性能。因此，设计一个能够减少孤立节点数量、提高簇的生存时间的分簇算法是非常必要的。

2 相关参数

2.1 相对移动参数

将实验中的场景模型定义为一个二维坐标轴，车辆通过 GPS 获取到自身的位置、速度矢量（包含速度和方向信息），设节点 i 的移动方向矢量为 $\vec{\boldsymbol{D}} = D_{ix}\vec{x} + D_{iy}\vec{y}$，则可知节点 i 和节点 j 的方向夹角 $\theta_{i,j}$ 为：

$$\theta_{i,j} = \arccos\left[\frac{(D_{ix}D_{jx} + D_{iy}D_{jy})}{(D_{ix}^{\ 2} + D_{iy}^{\ 2})^{1/2}(D_{jx}^{\ 2} + D_{jy}^{\ 2})^{1/2}}\right] \tag{1}$$

其中：$\theta_{i,j}$ 为节点 i 和节点 j 的方向夹角，D_{ix}、D_{jx} 分别为节点 i 和节点 j 在 x 轴方向的移动距离，D_{iy}、D_{jy} 分别为节点 i 和节点 j 在 y 轴方向的移动距离，如果 θ 的值在 $-\pi/4$ 到 $\pi/4$ 之间时，则认为这两个节点属于相同类型的车道，是同一方向。否则判定车辆间的移动方向相反，不满足加入同一个簇的基本条件，不能加入同一个簇。

同样地，假设节点 i 和节点 j 所获取到的车辆的自身速度为 V_i 和 V_j，则节点 i 和节点 j 的相对速度 $V_{i,j}$ 可按下列公式计算：

$$V_{i,j} = |V_i - V_j| \tag{2}$$

其中：V_i、V_j 分别为节点 i 和节点 j 的速度，$V_{i,j}$ 为节点 i 和节点 j 的相对速度。根据得到的车辆节点间的相对速度，引进相对移动参数（RMM，relative movement metric）来表示节点 i 和节点 j 的相对移动性，具体公式如式（3）所示：

$$RMM(i,j) = \ln[V_{\max}/(V_{\max} - V_{i,j})] \tag{3}$$

其中：$RMM(i,j)$ 为节点 i 和节点 j 的相对移动性的参数，由于高速公路场景中一般会规定一个速度上限，$V_{\max}$ 为规定好的速度上限，如 90 km/h。如果节点 i 有 n 个邻居车辆（一跳的通信距离），则节点 i 的相对移动值计算公式如式（4）：

$$RMM(i) = (1/n)\sum_{j=1}^{n} RMM(i,j) * \sum_{j=1}^{n} RMM(i,j) \tag{4}$$

其中：$RMM(i)$ 为节点 i 相对 n 个邻居节点的相对移动参数，显而易见，$RMM(i)$ 的值不可能超过 1，$RMM(i)$ 越小，则表示它与它的邻居车辆的速度差异越小，那么，当这个拥有更小 RMM 值的车辆节点成为簇头时，由它所构成的簇的结构就会更加稳定。

2.2 车辆在网剩余时间参数

当节点 i 进入高速公路场景时，通过车联网可以知道它当前位置到它脱离时位置（即目的地）之间的距离，用 S_i 表示，则车辆在网剩余时间参数（VLM，vehicle lifetime metric）表示为：

$$VLM(i) = 1/S_i \tag{5}$$

当一个节点的 $VLM(i)$ 的值越小，就表示在相似的移动性下车辆节点可以行驶的时间更长，则构成的簇的结构就越稳定。

2.3 消息数据格式

簇生成和簇维护算法中涉及使用到的所有数据包的结构如下：

2.3.1 Beacon 消息

消息类型	节点 ID	位置	速度	方向

2.3.2 Sink 消息

消息类型	节点 ID	位置	速度	方向	M 值

2.3.3　Invite 消息

消息类型	簇头 ID	M 值

2.3.4　Join 消息

消息类型	节点 ID	簇头 ID

2.3.5　CHA 消息

消息类型	簇头 ID	位置	速度	方向	M 值

2.3.6　CMA 消息

消息类型	节点 ID	簇头 ID	位置	速度	方向	M 值

2.3.7　Turn 消息

消息类型	簇头 ID	候选者 ID

2.3.8　Cturn 消息

消息类型	簇头 ID	M 值

2.3.9　Split 消息

消息类型	簇头 ID	候选者 ID

2.3.10　Tinvite 消息

消息类型	簇头 ID	原簇头 ID

3　算法设计及具体流程

3.1　簇头选择权值参数

在 2.1 和 2.2 节中，给出了影响簇头选择权值的两个重要因素的定义：相对移动参数 RMM 和车辆在网剩余时间参数 VLM。接下来，根据这两个因素，我们定义了簇头选择权值参数 M 来体现节点 i 成为簇头的能力指数大小：

$$M(i) = 0.75RMM(i) + 0.25VLM(i) \qquad (6)$$

其中：$M(i)$ 为节点 i 的权值，节点 i 的 M 值越小，就表示节点 i 与周围的邻居节点间有着更相似的移动性，节点 i 有着更长的剩余时间，它就越有机会成为簇头节点。如果节点 i 的 M 值越小，那么当节点 i 成为簇头时，形成的簇结构的稳定性就会越强，簇生存时间更长。

3.2　簇生成算法

在网络初始阶段，车辆节点均为孤立节点，每个车辆节点根据自身的 GPS 设备获取到自身的位置信息和速度矢量，通过周期性的广播 Beacon 消息将自身的运动状态广播给它们的邻居节点。同时车辆节点也会一直收集周围的邻居节点发送来的消息并建立邻居列表。接下来，节点根据式（1）计算它们之间的移动方向夹角 θ，然后根据这个夹角判断它们的方向是否是一致的[7]。如果它们的移动方向不同，就把这个节点从邻居表列表中剔除。所以，车辆节点的邻居列表中皆为行驶于相同移动方向且通信距离为一跳的邻居车辆节点的信息。之后根据车辆节点的邻居列表和式（2）～（6）来计算节点的簇头选择权值参数 M，并发送 Sink 消息给它的邻居列表中的其他节点。

在车辆节点进入网络后并广播 Beacon 消息后，假设孤立节点 i 的邻居列表中没有任何簇头存在，就将节点自身的 M 值和邻居列表中的其他节点的 M 值的大小进行比较，如果节点 i 的 M 值是最小的，那么节点 i 就被推选成为簇头，并向它的邻居列表中的车辆节点广播一个包含自身 ID 和权值 M 的 Invite 消息，当节点收到了来自簇头的 Invite 消息时，回复一个 Join 消息表示加入该簇，并更新它们的邻居列表。簇生成算法的流程如图 2 所示。

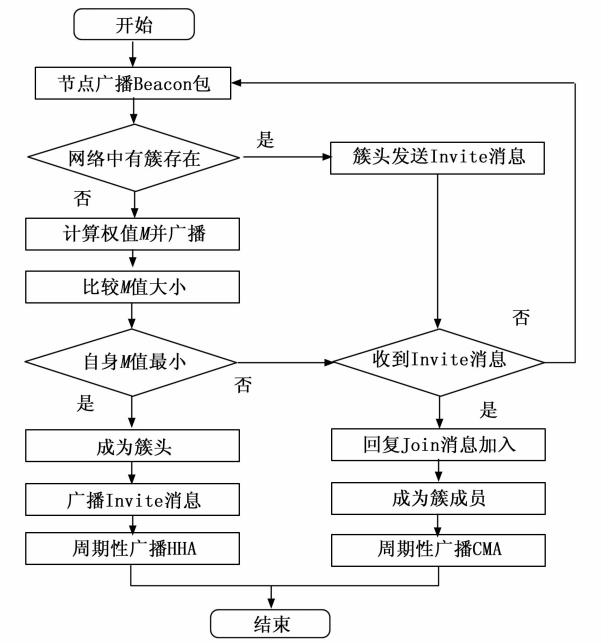

图 2　分簇算法流程图

在簇形成后，簇头节点和簇成员之间会周期性的广播 CHA 消息和 CMA 消息来更新它们的最新运动状态，这有便于维护簇的结构。由于孤立节点会周期性的广播 Beacon 消息，当簇头节点收到了来自孤立节点的 Beacon 消息后，会立即向该孤立节点发送一个 Invite 消息，收到 Invite 消息后，该孤立节点会选择回复一个 Join 消息后加入该簇。同时，簇头也会周期性的向外广播 CHA 消息，因此网络中的全部节点都会周期性的更新它们的邻居列表。

由于在簇头的选择中引入了权值参数 M，因而使用该算法的簇生成算法所形成的簇的结构更稳定，生存周期更长。

3.3　簇维护算法

当车联网中的簇形成以后，需要对网络中已经形成的簇执行簇维护算法来进行更新维护，簇的维护在一定程度上可以反映分簇算法的性能。本节将会介绍簇结构采取的维护措施以及执行簇合并和簇分裂的需要的条件和具体的

算法机制。

3.3.1 簇维护

当网络中簇的生成过程执行完成后，簇头节点、簇成员和孤立节点将会周期性的广播 CHA、CMA 和 Beacon 消息，当节点收到这些消息后，便会重新计算它的簇头选择权值参数 M，更新它的邻居列表。

在邻居列表更新完成后，如果簇头节点发现某个簇成员的簇头选择权值参数 M 的值比它小，该簇头节点就会广播包含自身节点 ID 和候选者 ID 信息的 Turn 消息，该簇成员就会替代原来的簇头成为新的簇头。

除此之外，如果节点 i 在两个周期的时间里没有接收到节点 j 的消息，则认为该节点已经脱离了它的通信范围，将其从邻居列表中移除。如果簇成员超过两个周期没有收到来自簇头广播的 CHA 消息，就认为自身已经脱离了簇，该节点变为孤立节点并重新执行簇生成算法。而当簇头在超过两个周期的时间没有接收到来自簇成员广播的 CMA 消息，就认为该簇成员已经脱离了该簇，将其从邻居列表中剔除。当簇中不存在任何簇成员，只剩下一个簇头时，该簇自动解散，簇头节点变为孤立节点并执行簇生成算法。

3.3.2 簇合并算法

在车联网中，簇在运行时也会额外的产生一笔通信开销用来管理和维护簇内成员的通信，所以当簇内的成员数量过少时，其产生的开销可能会高于节点之间的直接通信开销，这时候生成簇反而事倍功半。另外由于簇头节点的负载能力等问题，每个簇头能够管理和维护的簇成员数量是有限的。假设在理想情况下，簇头节点能够管理维护的最大簇成员个数为 N_{max}，应确保簇合成后的簇内成员数不会大大超过 N_{max}。因此，簇合并机制的中心思想是将相互覆盖的、簇成员数小于等于 $0.5N_{max}$的簇进行合并。

由于车辆节点的高移动性，可能会存在两个及以上簇的通信半径相互覆盖的情况，为了降低簇的叠加度和冗余，当有两个及以上簇头的通信半径相互覆盖且簇成员数$<=0.5N_{max}$时，就启动簇合并算法。此时簇头 i 将会发送 Cturn 消息给其他的簇头，如果有簇头回复了它，则比较他们 M 的值，M 值小的簇头将会作为一个新簇头，并且权值大的簇会解散并加入到权值小的簇头的簇中。具体簇合并算法流程如图 3 所示。

簇合并机制是将两个规模过小的簇合并成一个正常规模大小的新簇，解决了当簇内成员数量过少时会增加额外的网络通信开销的问题。

3.3.3 簇分裂算法

随着网络中的车辆节点数量增加，车辆节点的密度会变大，簇内成员的数量也就会增加。如果簇内的成员数量超过了理想状态下簇成员的数量时，簇头便开始无法承受簇内及簇间通信产生的负载。基于这种情况，设计了一种簇分裂算法，中心思想为：当簇内的成员数量超过了理想状态下簇内成员的数量时，则从簇内成员中选出一个车辆节点作为新生成簇的簇头并邀请原簇内的其他簇成员加入。

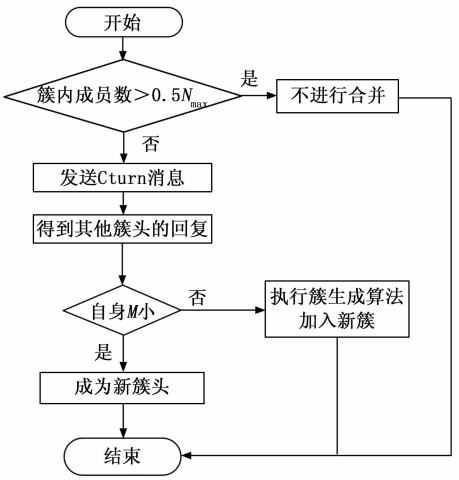

图 3 簇合并算法

这样，规模过大的簇就会被分裂成正常规模大小的簇，降低了网络中簇的簇平均大小。

在簇分裂算法中，当簇头节点 i 的簇内成员数的数量大于 $1.2N_{max}$时，簇头 i 便开始遍历它的邻居列表，从中选出一个除自身外簇头选择权值参数 M 的值最小的车辆节点，并向其发送 Split 消息。收到 Split 消息的车辆节点立即转换自身的状态为簇头，并发送 Tinvite 消息给原簇内的所有成员，理论上簇内成员选择加入每个簇的概率是相同的，因此算法定义收到 Tinvite 消息的车辆节点将会立即生成一个处于 0～1 的随机数 $Random$，当 $Random>0.5$ 时，该车辆节点就加入新生成的簇中，否则就继续留在原簇中。簇分裂具体流程如图 4 所示。

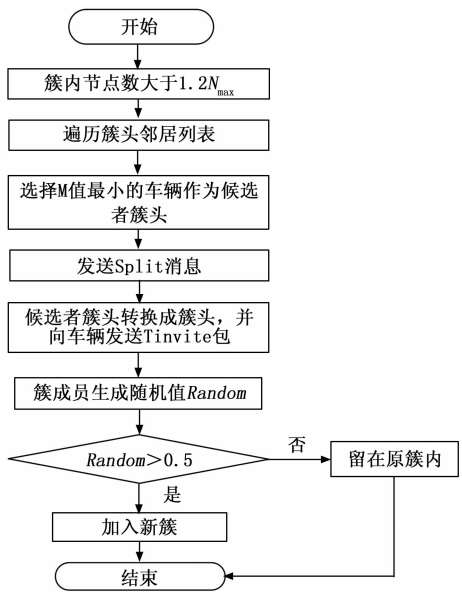

图 4 簇分裂算法

3.4 算法具体流程

基于移动性预测分簇算法的具体流程如下：

1）根据簇头选择权值参数 M 来进行簇头的选择，该参

数与车辆节点间的相对移动性以及节点的剩余寿命有关。

2）将车辆节点自身的参数 M 值和邻居列表中的其他节点的 M 值的大小进行比较，如果节点 i 的 M 值是最小的，那么节点 i 就被推选成为簇头。

3）簇头向车辆节点的邻居列表中的节点广播一个包含自身 ID 和权值 M 的 Invite 消息。当节点收到了来自簇头的 Invite 消息时，回复一个 Join 消息表示加入该簇，成为簇成员并更新它们的邻居列表。

4）簇头、簇成员和孤立节点会周期性的广播 CHA、CMA、Beacon 消息来更新它们的邻居列表并维护簇。在簇维护阶段，如果簇规模过小并且两个簇的通信半径相互覆盖时，转步骤 5)；如果簇的规模过大时，转步骤 6)。

5）在网络中簇规模过小并且两个簇的通信半径相互覆盖时对它们进行合并，避免网络资源的浪费，簇合并后转步骤 2)。

6）当网络中簇的规模过大时，就会执行簇分裂操作，将规模过大的簇分裂成若干个规模正常的小簇，簇内成员通过生成一个随机数来选择加入哪个簇，保证车辆节点在簇分裂时加入到每个簇的概率都是一样的，簇分裂后转步骤 3)。

车辆节点加入车联网络后始终执行步骤 1）～6)，直到离开网络为止。

4　实验仿真及结果分析

4.1　实验步骤和方法

本文使用城市交通仿真软件（SUMO，simulation of urban mobility）与 OMNeT＋＋[19] 的 Veins[20] 项目进行联合仿真，实现移动性预测分簇算法的仿真验证。车联网仿真中会涉及到信息流和交通流，其中交通流是由对真车行车环境的建模来体现，实验中用 SUMO 来实现；而信息流则反映网络中节点间数据的交换，使用 OMNeT＋＋来实现这一功能。最后的仿真数据通过 MATLAB 来绘制成更直观的图形。输入输出关系如图 5 所示。

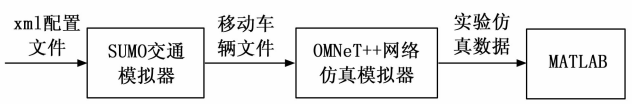

图 5　联合仿真平台设计

在仿真开始时刻，车辆将以随机的速度行驶在场景中，仿真主要对网络形成后车辆节点生成或加入到簇所需的时间进行了分析。表 1 为所需仿真参数。

表 1　节点首次加入簇需要时间仿真参数配置

参数	值
仿真时间	300 s
车辆数	10～100
车速限制	50 km/h
通信半径	200 m

4.2　仿真结果及分析

本论文设计的分簇算法是以高速道路的场景为基础创建的，实验仿真时间 300 s，车辆节点数 10～100 个，车速限制在 50 km/h，通信范围在 200 m 内进行仿真。

为了检验算法的性能，将本文算法和经典的 MOBIC 分簇算法[12] 从不同方面的网络性能进行研究，分别针对节点加入簇需要的时间、孤立节点数量、簇稳定性方面进行仿真对比分析。

4.2.1　网络性能仿真及分析

图 6 给出了在网络中放入 20 个车辆节点时它们首次加入到簇所需要的时间。直观地看，在网络初始阶段，所有车辆都为孤立节点，因此需要一定的时间来执行簇生成算法生成簇，当网络中有簇存在时，节点加入到簇需要的时间就会大大缩短。

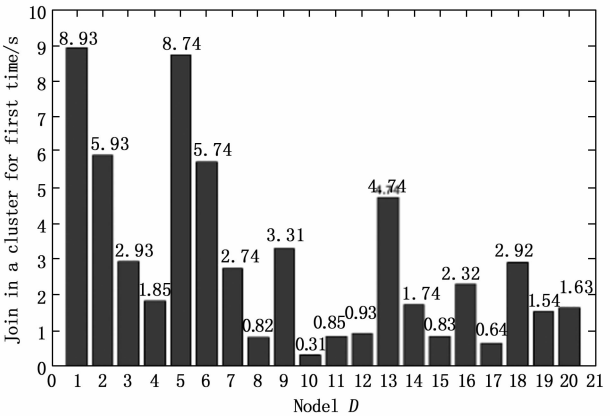

图 6　节点首次加入簇所需时间

当网络中放入不同数量的车辆节点时，通过仿真可以得到网络中的节点从进入网络开始平均生成或加入簇所需要的时间。

图 7 对网络中分别放入 10～100 个节点时，节点从进入网络开始平均生成或加入簇需要的时间作出了比较，在网络初始阶段，所有车辆都为孤立节点，需要一定的时间来执行簇生成算法生成簇，当网络中有簇存在时，节点加入到簇需要的时间显著缩短。当车辆数量增加时，网络内会形成更多的簇，网络中的孤立节点就会有更大的可能性加入到簇。

随着网络内节点数量的增加，节点平均加入到簇所需要的时间在减少，这在一定程度表现了分簇算法的网络性能良好，能够更快地生成簇，孤立节点也能更加迅速的加入簇，很好的降低了网络中孤立节点的数量，从图 7 可以看出本文算法中节点平均加入到簇需要的时间短于 MOBIC 算法，网络性能良好。

4.2.2　网络通信开销性能仿真及分析

当网络中放入不同数量的车辆节点时，通过仿真可以得到网络中存在的孤立节点数量的仿真数据来描述网络通信开销性能的好坏。

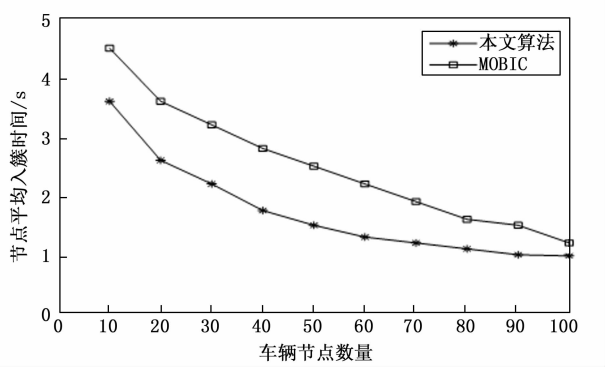

图 7 节点平均加入簇需要的时间

孤立节点数量等于节点总数减去所有簇头及簇成员的数量，分别向网络中放入 10～100 车辆节点，在运行一段时间后得到网络中孤立节点的数量。根据图 8 中的结果，从总体上看，本文算法中的孤立节点数量明显小于 MOBIC 算法中的孤立节点数量，网络中孤立节点的数量平均在注入车辆节点数量的 10%，随着网络内节点数量的增加，孤立节点所占比例保持稳定，算法的网络通信开销性能良好。

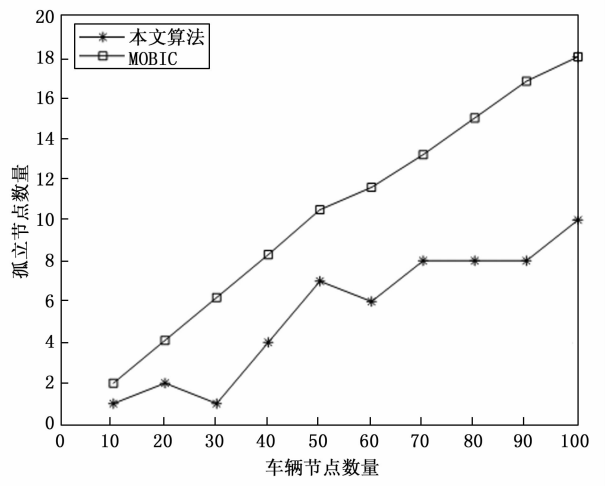

图 8 孤立节点数量

假设在车辆节点加入到簇成为簇成员时给节点增加一个簇停留时间预测 t，当这个值 t 变成 0 时，该簇成员重新执行簇生成算法。这样可以减少车辆节点作为孤立节点存在的时间，有效地减少孤立节点的数量，从而提高网络开销通信性能。

4.2.3 簇结构稳定性能仿真及分析

簇的稳定性是评价一个分簇算法优劣性的一项重要指标。

根据图 9，可以得到网络中平均簇大小与车辆节点数量的关系。从图 9 中可以看出，MOBIC 算法进行网络分簇时算法生成的平均簇大小增加得非常迅速。

当使用本文算法进行网络分簇时，在车辆节点数量增加的情况下，簇的成员数量逐渐稳定在 12～13 左右。这表现了在控制簇大小这项性能上，本文分簇算法的簇合并和簇分裂机制表现良好，能够有效地降低网络通信开销。因

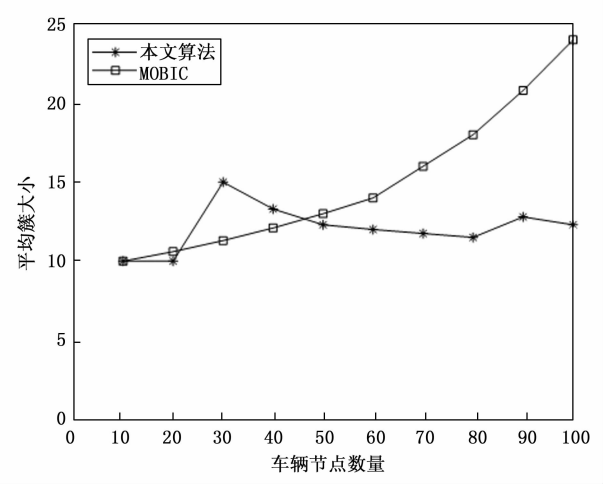

图 9 平均簇大小

此，在维护簇的结构的时候，簇合并和簇分裂的算法就显得尤为重要。

图 10 表示网络中平均簇头变化率与车辆节点数量的关系，其中，平均簇头变化率指在运行的仿真时间内，每秒簇头的变化次数（包括簇维护中簇头的更换、簇合并及簇分裂），网络中簇的结构稳定性可以用平均簇头变化率来体现。如果车联网中平均簇头变化率过大，则代表该网络由于簇头的频繁切换导致了网络拓扑结构变得不稳定。根据图 10 可以得知，随着车辆节点数量的增加，本文算法中簇的平均簇头变化率也在提高。但从总体上来看，簇头的平均变化率没有特别的显著提升，簇的稳定性有所保障，优于 MOBIC 算法。

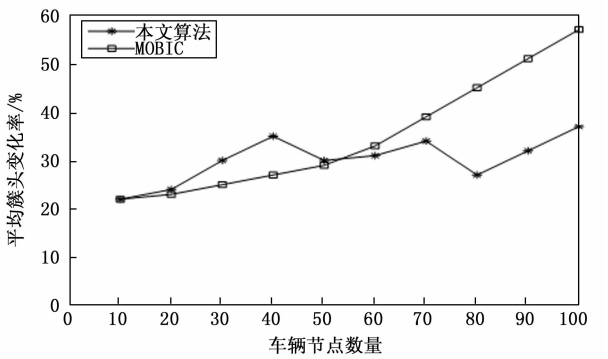

图 10 平均簇头变化率

5 结束语

本文提出一种基于移动性预测的分簇算法，该分簇算法自适应车辆移动，将具有相似移动性的车辆组成簇。该算法分为簇生成和簇维护两个阶段，其中，在簇生成阶段，根据簇头选择权值参数 M 来进行簇头的选择，该参数与车辆节点间的相对移动性以及节点的剩余寿命有关。在簇维护阶段，簇头、簇成员和孤立节点会周期性的广播 CHA、CMA、Beacon 消息来更新它们的邻居列表并维护簇。除此

（下转第 283 页）