

基于邻域跟随的车联网多跳分簇稳定性研究

林志坚¹, 吴文浩¹, 陈小培², 曾泽雄¹, 林永航¹, 陈平平^{1*}

(1. 福州大学物理与信息工程学院, 福建福州 350108; 2. 华南理工大学未来技术学院, 广东广州 511442)

摘要: 针对车联网计算卸载需求量大及车辆网络拓扑变化随机性强的问题, 本文提出一种基于邻域跟随的车辆多跳分簇方法, 并采取多指标综合加权的方式研究其稳定性特征. 现有针对车辆分簇的研究方案普遍存在指标单一化、分簇机制局限于单跳或固定跳数、分簇效率低及缺少簇头替换机制等问题. 鉴于邻域跟随策略稳定性较优且分簇效率较高等特性, 本文提出车辆基于邻居节点内综合指标最优准则选择稳定性最好的节点跟随方案, 通过去环和剪枝算法使得车辆的跟随结构规范且扁平化, 进而形成一个稳定性更高的车辆簇, 并利用主从簇头和簇维护机制增强车辆簇的鲁棒性. 仿真结果表明, 与现有的车辆分簇方法相比, 所提算法在簇稳定性和分簇效率上均具有较好的性能.

关键词: 车联网; 邻域跟随; 多跳分簇; 组网稳定性; 状态转移; 分簇效率

基金项目: 国家自然科学基金(No.62171135, No.62001117); 福建省杰出青年科学基金(No.2022J06010)

中图分类号: TP393.0

文献标识码: A

文章编号: 0372-2112(2024)04-1260-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20231099

Research on the Stability of Multi-Hop Clustering Based on Neighbor Following Method in Vehicular Networks

LIN Zhi-jian¹, WU Wen-hao¹, CHEN Xiao-pei², ZENG Ze-xiong¹, LIN Yong-hang¹, CHEN Ping-ping^{1*}

(1. College of Physics and Information Engineering, Fuzhou University, Fuzhou, Fujian 350108, China;

2. School of Future Technology, South China University of Technology, Guangzhou, Guangdong 511442, China)

Abstract: In view of the problems of high demand of computation offloading in vehicular networks and the randomness of topology variation of vehicular networks, this paper proposes a vehicular multi-hop clustering scheme based on neighbor following and investigates its stability by multiple indicators. Existing research offers various vehicular clustering schemes, but they suffer from issues such as single indicator, being limited to a single hop or fixed number of hops, low clustering efficiency, and lacking a mechanism for cluster head replacement. As the neighbor following strategy has better stability and higher clustering efficiency, in the scheme designed in this paper, vehicles should follow the most stable vehicle based on the comprehensive indicators within neighboring nodes. The de-ringing and pruning algorithms are used to standardize and flatten the following structure of vehicles, forming a more stable vehicular cluster. The master-slave cluster header and the cluster maintenance mechanism are employed to enhance the robustness of vehicular cluster. Simulation results show that the proposed algorithm outperforms existing methods in terms of cluster stability and clustering efficiency.

Key words: vehicular networks; neighbor following; multi-hop clustering; networking stability; state transition; clustering efficiency

Foundation Item(s): National Natural Science Foundation of China (No.62171135, No.62001117); Fujian Province Outstanding Youth Science Fund (No.2022J06010)

1 引言

随着车联网中时延敏感型和计算密集型任务的激增, 设计高效的车对车 (Vehicle-to-Vehicle, V2V) 和车对路边单元 (Vehicle-to-Road side unit, V2R) 信息传输方

式的难度也在不断增加^[1-4]. 车辆分簇^[3,4]方案可以增加车辆通信时长和提高车辆通信成功率, 适用于通信距离有限、节点移动性高和网络拓扑变化大的车联网应用场景, 因此受到了研究者广泛关注. 根据分簇网络

中车辆链路长度的不同,可以将其分为单跳分簇网络与多跳分簇网络两大类.

针对单跳分簇网络,研究者们通过高斯-马尔可夫迁移模型^[5]和智能簇头机制^[6]等方法探索了车辆分簇网络在特殊条件下的移动性感知问题.此外,还有研究者通过设计形成分簇时簇头的选择标准^[7,8]以提升车辆网络的稳定性.这种局限于单跳的方法在静态网络或弱移动性网络中较为稳定且具有时延低、管理开销小等优点.然而,在拓扑高动态变化的车辆自组织网络中,这种覆盖范围小的分簇方法在维持分簇稳定上具有较大的缺陷,车辆容易脱离簇的覆盖范围,从而降低簇集网络的稳定性.在实际场景中,车辆数通常较多,因此可以使用多跳分簇的方式以应对车辆在一定范围内的快速变动.针对多跳分簇网络,研究者提出了基于粒子群优化^[9]、相对移动性^[10]、分层聚类协议^[11]和优先级^[12]的聚类算法,通过设计适应网络变化的动态簇头选择方法,在通信效率和稳定性等方面提升网络的性能.然而,文献^[9~11]在形成分簇网络时仅考虑了单一稳定性指标,文献^[12]虽使用综合指标但缺少簇头替换机制,因此形成的车辆簇在分簇稳定性及分簇效率等方面仍性能不佳.

针对以上问题,为提高整体车辆分簇网络的稳定性和分簇效率,本文提出一种面向稳定性的邻域跟随多跳分簇(Neighbor Following Multi-hop Clustering, NFMC)方案,考虑车辆的社交属性、移动性及通信质量的综合影响,采用邻域跟随方法形成拓扑并初始化分簇,所形成的簇集网络由于具有多指标综合稳定性,可以进行多种任务类型的传输.同时,设计车辆节点的状态转移机制,提出基于主从簇头^[13]的簇集网络维护策略,从而有效提高簇的平均生存时间.仿真结果表明,所提的多跳分簇策略在簇稳定性和节点分簇效率两个方面均具有较好的性能优势.

2 场景描述

2.1 系统模型

如图 1 所示,车辆节点的集合可定义为 $V_{\text{car}} = \{v_1, v_2, \dots, v_n, \dots, v_N\}$, $v_n \in V_{\text{car}}$, 车辆统一装载 GPS 系统,可实时获取自身速度、位置和方向等信息,具有 V2V 和 V2R 通信功能.所有成为簇头的车辆节点均可以进行 V2V 和 V2R 通信,所有未成为簇头的车辆节点均仅进行 V2V 通信.在形成车辆网络时,本文主要考虑 V2V 通信对成簇的影响.车辆根据其是否在簇内以及是否结束行程设置不同的车辆状态(这一设定将在第 3 节中详细阐述).簇头定时向路侧单元(Road Side Unit, RSU)传输簇内节点信息,同时负责将簇成员的数据汇集后传送到 RSU. 车辆 v_n 定时广播节点属性: $I_{v_n} =$

$\{\delta_{v_n}, s_{v_n}, p_{v_n}, u_{v_n}, j_{v_n}, o_{v_n}, f_{v_n}, k_{v_n}, q_{v_n}\}$, 其中包含车辆 ID δ_{v_n} 、车辆状态 s_{v_n} 、位置 p_{v_n} 、速度 u_{v_n} 、跟随对象的车辆 ID j_{v_n} 、可用的一对一 V2V 通信接收天线数目 o_{v_n} 、与簇头或者出度为 0 节点(即从该节点指出去的箭头个数为 0)的跟随关系跳数 f_{v_n} 、簇头 ID k_{v_n} 以及下级跟随节点 q_{v_n} . 本文所用的其余重要符号及含义列于表 1 中.

表 1 重要符号意义对照表

符号	描述
$N(v_n)$	v_n 的邻居节点集合
D_{v_n}	v_n 的节点跟随度
G_{v_n}	v_n 在同一车道线上的邻居节点集合
Q_{v_n}	通过邻域跟随直接或间接依附于 v_n 的节点集合
LLT_{v_n, v_m}	v_n 与单跳邻居车辆节点 v_m 的链路预测生存时间
H_{v_n, v_m}	v_n 和 v_m 之间的信道容量积
r_{v_n, v_m}	v_n 向 v_m 发送数据的速率
λ_{v_n, v_m}	v_n 和 v_m 之间的综合邻域跟随指标

车辆 v_n 接收到通信半径 R 内的邻居节点定时发送的广播消息后,保存并更新邻居节点集合 $N(v_n)$. 车辆将检查邻居节点的定时广播消息,如果 v_n 未接收到邻居广播消息的次数连续累计超过连接灵敏度阈值 η ,则视为与该邻居节点断联,需要从 $N(v_n)$ 中删除该节点信息. 车辆节点根据 $N(v_n)$ 定时计算自身和邻居节点的相关指标,通过归一化处理构成综合的分簇权重 λ_{v_n, v_m} ,并将其作为邻域跟随选择的指标参考.此外,本场景考虑单向行驶车道,且车辆可能在道路上由于行程结束而在道路侧方停车.

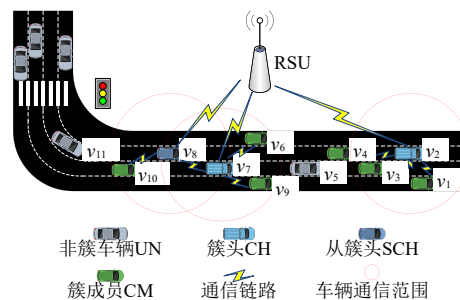


图 1 多跳分簇场景图

2.2 邻域跟随指标建模

2.2.1 节点跟随度

将通信实体视为图的某个节点后,该节点的节点跟随度^[14]可以表示该实体通信范围内的可通信节点数目,是分簇算法中一种用于建立节点关系的重要指标.因此,本文将基于节点度扩展的跟随度 D_{v_n} ^[12] 作为节点稳定性的评估指标之一:

$$D_{v_n} = |G_{v_n} \cup Q_{v_n}| \quad (1)$$

其中, G_{v_n} 是节点 v_n 在同一车道线上的邻居节点集合; $|G_{v_n}|$ 表示该集合中的元素个数, 其值越大, 节点 v_n 越稳定. 假设车辆通信半径为 R , 车身长度为 l , 在同一车道上, 车辆 v_n 通信范围内的平均车辆间距为 d_{avg} , 则 $d_{\text{avg}} = \frac{2R}{|G_{v_n}|}$. 设驾驶员的反应时间为 ε_0 , 则车辆跟驰的安全速度不超过 $u_{v_n}^{\text{ss}} = \frac{d_{\text{avg}} - l}{\varepsilon_0}$. 由此可知, G_{v_n} 增大时, d_{avg} 的值减小, 从而 u_{v_n} 的值减小, 则 v_n 节点更加稳定. Q_{v_n} 代表通过邻域跟随直接或间接依附于 v_n 的节点集合, $|Q_{v_n}|$ 表示该集合中的元素个数. 直接跟随关系定义为 $\text{rel}_d = \{v_m \rightarrow v_n | v_m \in N(v_n)\}$. 间接跟随关系由直接跟随关系构成, 如 $v_k \rightarrow v_m \rightarrow v_n$ 是一个长度为 2 的多跳跟随关系链. 在图 1 表示的情况下, v_{10} 直接跟随节点 v_8 , 而节点 v_8 直接跟随节点 v_7 , 因此形成一个间接的多跳跟随关系 $v_{10} \rightarrow v_8 \rightarrow v_7$. 本文将具有一般形式的间接跟随关系定义为: $\text{rel}_i = \{v_k \rightarrow v_{k+1} \rightarrow \dots \rightarrow v_{n-1} \rightarrow v_n | v_{k+1} \in N(v_k), \dots, v_n \in N(v_{n-1})\}$, 为方便描述, 记为 $v_k \Rightarrow v_n$. 综上所述可知, $Q_{v_n} = \{v_i | v_i \rightarrow v_n, v_i \Rightarrow v_n, i \in V_{\text{car}}, i \neq n\}$. 如图 1 所示, 与簇头 v_7 具有直接跟随关系的下级节点有 v_6, v_8 和 v_9 , 节点 v_{10} 直接跟随节点 v_8 , 故 $|Q_{v_7}| = 3 + 1 = 4$. 在后文中, 若没有特别说明, 将上述直接和间接跟随关系中的依附节点分别称为被依附者的直接下级节点和间接下级节点, 合称为下级节点. 车辆节点形成直接跟随关系是通过在其邻居范围内挑选一个具有最佳跟随指标的节点.

在实际的车辆联网场景中, 邻域跟随还应满足以下约束关系. 首先, 由于假设车辆的 V2V 接收天线数目为 L_{max} , 其中有一根天线用于接收跟随对象的消息, 因而车辆 v_n 的直接下级节点数量不能超过 $L_{\text{max}} - 1$, 则有 $o_{v_n} \in [0, L_{\text{max}} - 1]$. 其次, 邻域跟随关系也不能被无限制地延长, 如果某一节点在跟随关系上到达簇头的跳数过大, 则其通过多跳与簇头的通信时延和开销都有可能难以满足用户的使用需求, 因此簇集网络中成员节点通过间接跟随关系与簇头节点通信的跳数不应超过设定的最大跳数限制 f_{max} . 后文中所提的跟随过程中的跟随合理性即为需要满足上述两个约束条件.

2.2.2 链路生存时间

链路生存时间 (Link Life Time, LLT)^[15] 是评估簇集网络生存能力的重要指标. 节点和选择的依附对象之间维持稳定链路的时间越长, 簇集网络的生存能力越强, 而链路的稳定持续时间和车辆的运动特性存在紧密联系, 以下分析 LLT 与车辆的运动参数之间的关系.

设车辆 v_n 在时刻 t 的速度为 $u_{v_n}(t)$, 位置为 $p_{v_n}(t)$, 车辆 v_n 定期广播的消息中含有上述信息, 当 v_n 的邻居节

点 v_m 接收到该消息时将计算该节点的 LLT 参数并存入 LLT 集合. 设车辆 v_m 速度为 $u_{v_m}(t)$, 位置为 $p_{v_m}(t)$, 假设在时间间隔 τ 内车辆速度不变, 则存在:

$$\begin{aligned} p_{v_n}(t+\tau) &= p_{v_n}(t) + \tau \cdot u_{v_n}(t) \\ p_{v_m}(t+\tau) &= p_{v_m}(t) + \tau \cdot u_{v_m}(t) \end{aligned} \quad (2)$$

当车辆之间的距离等于 R 时, 两车断开连接, 其临界状态可以表示为

$$\begin{aligned} & \left| p_{v_n}(t + \text{LLT}_{v_n, v_m}(t)) - p_{v_m}(t + \text{LLT}_{v_n, v_m}(t)) \right| \\ &= \left| p_{v_n}(t) - p_{v_m}(t) + \text{LLT}_{v_n, v_m}(t) [u_{v_n}(t) - u_{v_m}(t)] \right| \\ &= R \end{aligned} \quad (3)$$

车辆 v_n 与单跳邻居车辆节点 v_m 的链路预测生存时间 LLT_{v_n, v_m} 表示为

$$\text{LLT}_{v_n, v_m}(t) = \frac{|\Delta u_{v_n, v_m}(t)| \cdot R - \Delta u_{v_n, v_m}(t) \cdot \Delta p_{v_n, v_m}(t)}{(\Delta u_{v_n, v_m}(t))^2} \quad (4)$$

其中, $\Delta p_{v_n, v_m}(t) = |p_{v_n}(x, t) - p_{v_m}(x, t)| + |p_{v_n}(y, t) - p_{v_m}(y, t)|$ 表示车辆节点 v_n 和 v_m 之间的街区距离, 即曼哈顿距离; $\Delta u_{v_n, v_m} = u_{v_n}(t) - u_{v_m}(t)$ 表示车辆 v_n 和 v_m 的速度差. 每当车辆收到邻居节点的消息时, 将根据邻居集合信息计算并更新此参数.

2.2.3 信道容量积

鉴于邻域跟随场景中发送方与接收方两处的信道容量均对链路稳定性有作用, 为更准确地研究链路稳定性对分簇结果的影响, 本文提出采用车辆节点之间两个传输方向信道容量的乘积表示节点 v_n 和 v_m 之间通信的链路质量, 即信道容量积:

$$H_{v_n, v_m} = r_{v_n, v_m} \times r_{v_m, v_n} \quad (5)$$

其中, $r_{v_n, v_m}, r_{v_m, v_n}$ 分别表示车辆 v_n 向 v_m 的发送数据率和反向发送数据的速率, 表示为

$$r_{i,j} = B \log \left(1 + \frac{P_t h d_{i,j}^{-\gamma}}{\sigma^2 + \sum_{k \neq i}^{N(j)} P_t h d_{k,j}^{-\gamma}} \right) \quad (6)$$

其中, B 为信道带宽, P_t 为车辆的发射功率, $d_{i,j}$ 为发送节点 i 与接收节点 j 的距离, σ^2 为信道噪声功率, h 为信道增益, γ 为阴影衰落系数.

2.2.4 综合加权指标

综上, 定义邻域跟随的综合指标 λ_{v_n, v_m} 为

$$\begin{aligned} \lambda_{v_n, v_m} &= \alpha \cdot \varphi(D_{v_n}) + \beta \cdot \varphi(\text{LLT}_{v_n, v_m}) + \theta \cdot \varphi(H_{v_n, v_m}) \\ \alpha + \beta + \theta &= 1, \quad \alpha, \beta, \theta \geq 0 \end{aligned} \quad (7)$$

其中, φ 表示离差归一化法则^[16] 处理数据方法: 假设 A_{v_n} 表示一组数据, 即车辆节点 v_n 与其各邻居节点 $v_m \in N(v_n)$ 间的上述任一独立指标 A_{v_n, v_m} 的集合.

算法 1 修正邻域跟随关系的去环算法

输入: 车辆集合 V_{car} , 初始邻域跟随关系图 rel_o , v_n 的跟随 ID j_{v_n}

输出: 无环邻域跟随关系图 rel_n

1. 初始化跟随链集合 $F = \emptyset$, $V'_{\text{car}} = V_{\text{car}}$
2. WHILE 存在车辆节点 $v_n \in V'_{\text{car}}$
3. $F = \emptyset$, 将 v_n 纳入集合 F
4. IF v_n 的跟随对象 $v_i \in V'_{\text{car}}$ 且 $v_i \notin F$
5. 将 v_i 纳入集合 F , $v_n = v_i$, 转到步骤 4
6. ELSE 转到步骤 8
7. END IF
8. IF v_n 的跟随 ID $j_{v_n} = -1$ 或者 $v_i \notin V'_{\text{car}}$
9. 将集合 F 中的节点从 V'_{car} 中删除
10. ELSE IF v_n 的跟随对象 $v_i \in F$
11. 将集合 F 中的节点从 V'_{car} 中删除
12. 将 F 中, 在 v_i 之前加入的节点从 F 删除得到环上节点集合, 取 F 中具有最多环外下级节点的节点 $v_b = \arg \max_{v_i \in F} (q_{v_i}, V)$, 令 $j_{v_b} = -1$, 并更新跟随关系
13. END IF
14. END WHILE

构, 对应一个有向有环图 G_r ^[14], 通过去环算法判断出环中具有最多下级节点的车辆为 4, 因为车辆 4 在 G_r 中除了环上的下级节点外还有一个环外的下级节点 5. 将 4→3 的跟随关系断开, 车辆 4 的跟随对象 ID 改为 -1, 继而得到如图 3(b) 所示的去环后簇结构, 是一个有向无环图 G_d .

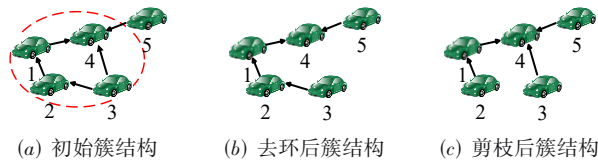


图 3 去环和剪枝示意图

4.1.3 剪枝

对于去环之后的有向无环图 G_d , 可能存在跟随关系链过长的问题. 由于多跳转发的时延和跳数呈正相关, 故应尽量减少转发跳数以获得更好的用户体验. 针对跟随关系链的长度问题, 本文设计了一种剪枝算法, 通过对已经形成簇的连通分量不断执行剪枝操作, 以扁平化簇结构, 从而提供更好的服务质量. 剪枝的流程如算法 2 所示.

假设最大跟随关系长度 f_{max} 为 2, 则在图 3(b) 所示的连通分量 G_d 中, 车辆 3 通过车辆 2 和车辆 1 到达 G_d 中出度为 0 的节点 4, 跟随关系的长度为 3, 已经超过 f_{max} , 因此需要执行剪枝: 节点 3 将根据当前跟随指标列表选择除了节点 2 以外的最优跟随合理邻居节点. 计算排序后发现邻居节点 4 具有除 2 节点以外最优的跟随指

标, 而节点 4 到出度为 0 的节点 (节点 4 本身) 的跟随链长度为 0, 当节点 3 跟随 4 时, f_3 值为 1, 满足跟随长度条件. 此外, 节点 4 的直接下级节点仅有 1、5 两个, 未达到其可直接通信节点上限, 因此可以接收节点 3 作为新的直接下级节点. 综上, 节点 3 将断开与节点 2 的跟随关系并跟随节点 4, 形成如图 3(c) 所示的剪枝后簇结构, 是一个有向无环图 G_s .

算法 2 限制邻域跟随跳数的剪枝算法

输入: V_{car} , 无环邻域跟随关系图 rel_n , 邻域跟随指标 λ , 簇 N , 跟随关系长度 f , 直接下级节点数量 o , 当前车辆状态 s

输出: 限制跳数的跟随关系图 rel_l

1. 初始化跟随指标列表 LT 为空列表, V_{car} 中所有节点计算出邻域跟随指标 λ
2. WHILE 存在无下级节点的车辆 $v_i \in V_{\text{car}}$ 且 $f_{v_i} > f_{\text{max}}$
3. IF $s_{v_i} \neq \text{UN}$
4. FOR $v_m \in N(v_i)$, $\{v_m | k_{v_m} = k_{v_i}\}$
5. v_i 通过式 (7) 计算在同一簇内的所有邻居车辆节点 v_m 的跟随指标 λ_{v_i, v_m} , 并存储为列表 $LT_{v_i}(v_m, \lambda_{v_i, v_m})$
6. END FOR
7. ELSE v_i 计算所有邻居车辆节点 v_m 的跟随指标 λ_{v_i, v_m} , 并存储为列表 $LT_{v_i}(v_m, \lambda_{v_i, v_m})$
8. END IF
9. 按照 λ_{v_i, v_m} 降序的方式排列 LT_{v_i}
10. FOR $v_m \in LT_{v_i}$
11. IF v_m 满足跟随 v_i 合理性
12. 改变跟随关系 $v_i \rightarrow v_m$, 转至步骤 16
13. END IF
14. END FOR
15. v_i 没有找到可以跟随的新节点, 解除与当前跟随对象的关系并修改状态为 UN
16. 如果 v_m 是非 UN 节点, v_i 改变自身状态为 CM, 更新跟随关系
17. END WHILE

4.1.4 簇形成

对于任一非孤立节点的非簇连通分量 G_s , 所有其他节点都通过直接或间接跟随链依附于其中唯一出度为 0 的节点, 将该节点作为簇的初始化簇头. 将其状态标志由 UN 改为 CH, 所有 G_s 中的其他节点状态由 UN 变更为 CM. 车辆簇的集合表示为: $C = \{c_1, c_2, \dots, c_i, \dots, c_j, \dots, c_M | i, j \in [1, M], i \neq j, c_i \cap c_j = \emptyset\}$, 其中 M 是车辆簇的数目, $M \geq 1$.

此外, 对任一簇集网络 $c_i \in C$, 其 CH 节点 v_c 从其直接下级节点中寻找一个具有最多从属节点的成员 $\arg \max (q_{v_s})$, 其中 v_s 为 v_c 的直接下级节点

$\{v_s | v_s \in c_i, v_s \rightarrow v_c\}$, 将其状态改为 SCH.

综上, NFMC 分簇算法流程如图 4 所示.

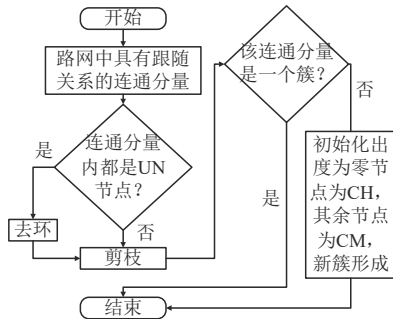


图 4 NFMC 算法流程图

4.2 簇维护机制

4.2.1 簇维持策略

某一簇内节点 v_c 与跟随对象断联时, 首先根据跟随指标 λ_{v_n, v_m} 在簇内重新选择最佳节点作为新的依附对象. 如果 v_c 的下级节点中有不满足跳数约束的节点, 将在剪枝过程中被处理. 如果没有找到合适的跟随对象, 该 v_c 节点将脱离当前簇, 并断开和直接下级节点的跟随关系. 这一策略使得簇内成员即使在断开和跟随对象的关系时仍然有机会留在簇内, 降低了簇级网络解散的概率并提高了成员的生存时间.

4.2.2 簇合并

当两个同向行驶的簇集网络的簇头彼此靠近时, 设置一簇合并计数器. 如果各自接收到互相发送的广播消息(假设每隔 500 ms 定时发送), 则计数器的值+1. 考虑到计数器值太小时不能保证两个簇集网络合并后的状态保持相同, 计数器值太大时则会浪费资源且簇合并机制的优势不明显, 因此选定临界条件为 3, 当计数器值为 3 时, 两个簇集网络进行合并, 流程如下: 先比较两个簇的成员数量, 较大者保持簇头和从簇头的身份成为合并簇的簇头和从簇头; 较小簇的簇头和从簇头成为合并簇头的下属成员, 即直接为该簇头添加依附关系, 依附对象为较大簇的簇头, 并将较小簇的簇头和从簇头状态变更为 CM. 此外, 被合并的簇集网络将不再增长其生存时间, 但其独立存在的历史信息将被保留, 用于计算平均簇生存时间. 为便于区分, 将持续生存的合并新簇称为优合并簇 c_e , 将被合并的历史簇称为劣合并簇 c_b .

4.3 NFMC 算法复杂度分析

本文所提的 NFMC 算法主要包括车辆跟随关系建立, 去环及剪枝三个步骤. 三个步骤的计算复杂度分别如下:

(1) 在跟随关系建立步骤中, 每次执行 NFMC 需要先对每个车辆执行邻域跟随找到依附对象, 设共有 n 个

车辆节点, 这一步骤的计算复杂度为 $O(n^2)$.

(2) 在去环步骤中, 对于每个连通分量(节点数目 ≥ 3), 分为从任一节点出发找到环的过程和在环上找到最佳断开边的过程. 设某连通分量 $\delta_k, k=1, 2, \dots$ 中有 n_δ 个节点, 最坏情况下这个连通分量形成一个环, 需要遍历所有节点后才能判定形成环, 去环所需要断开的一条边需要在环上节点进行比较, 故连通分量 δ_k 最坏情况下的计算复杂度为 $O(n_{\delta_k}^2)$, 整个路网执行去环算法的计算复杂度为 $O(\max_{k=1, 2, \dots} n_{\delta_k}^2)$.

(3) 在剪枝步骤中, 对于经过去环操作的图, 设无下级的超过最大跟随长度节点 v_k 共有 n_{cut} 个, 它们的邻居节点分别有 m_k 个, $k=1, 2, \dots, n_{cut}$. 算法首先分别计算 v_k 的邻居节点的跟随指标, 计算复杂度为 $O(m_k)$, 得到与 v_k 在同一簇内的邻居 $m'_k \leq m_k$. 冒泡降序排列的计算复杂度为 $O(m'_k)^2$, 遍历列表搜索跟随目标的计算复杂度为 $O(m'_k)$. 因此, 剪枝算法的计算复杂度为 $O(\sum_{k=1}^{n_{cut}} (m_k + m'_k + (m'_k)^2))$, 化简后得 $O(\max_{k=1, 2, \dots, n_{cut}} (m_k + (m'_k)^2))$.

综上, NFMC 算法的计算复杂度为 $O(n^2 + \max_{k=1, 2, \dots} (n_{\delta_k}^2) + \max_{k=1, 2, \dots, n_{cut}} (m_k + (m'_k)^2))$, 其中 $n \geq n_{\delta_k}$ 且 $n > m_k > m'_k$, 经化简后的整体计算复杂度为 $O(n^2)$.

5 仿真结果及分析

本文以 OMNeT++ 作为仿真平台, 使用 Veins (5.6.2 版本) 框架, 并在 SUMO 软件中实现交通模型的仿真. 设定车辆节点的最大加速度 A_{max} 和最大减速度 Z_{max} 分别为 2.6 m/s^2 和 4.5 m/s^2 ; 车辆最大速度 u_{max} 和通信半径 R 分别为 20 m/s 和 200 m ; 最大间接跟随长度 f_{max} 为 5. V2V 通信天线数 L_{max} 设置为 4, 车辆连接灵敏度 η 设为 2. 车辆场景为一个 $1000 \text{ m} \times 600 \text{ m}$ 的矩形轨道, 具有四条车道, 路网中的总车辆数量为 120~180 辆, 产生方式由 SUMO 中的车流描述文件描述. 所有车辆均位于 RSU 覆盖范围内, 当成为簇头时均可与 RSU 通信. V2V 通信带宽 B 取 10 MHz , 车辆发送功率 P_t 为 200 mW , 信道增益 h 为 0.001, 取信道阴影衰落因子 γ 为 4, 噪声功率 σ^2 为 $-90 \text{ dBm}^{[12]}$. 取跟随指标 λ_{v_n, v_m} 的加权因子 $\alpha = \beta = \theta = \frac{1}{3}$. 车辆广播数据包的发送间隔为 500 ms, 蒙特卡洛的实验次数为 50 次.

为验证所提算法的性能, 本文将其与单跳分簇 (single-hop clustering, 1-hop) 算法^[8]以及被动多跳分簇 (Passive Multi-hop Clustering, PMC) 算法^[12]进行比较. 1-hop 是一种典型的单跳分簇方式, 通过将车辆节点划分为簇头与簇成员两种状态以提供车辆间的高效通信. PMC 算法是一种基于优先级的多跳分簇算法, 保

证了聚类的稳定性和覆盖率,但簇生存能力较差. 由于 OMNeT++ 软件仿真时车辆节点从 0 开始逐渐增加, 考虑到多跳分簇算法对车辆节点基数的要求以及真实的交通场景, 需要通过一定的仿真时间, 让一部分车辆随机运动到道路上再进行性能测试, 因此设定路网形成时间 T_f 为 80 s, 在此时间之前, 车辆节点保持 UN 状态, 且路网中不产生车辆簇. 经过形成时间 T_f 后, 路网将具有一定规模的车辆数量并开始分簇. 记录在 $T_f=80$ s 后变化的平均簇生存时间、平均簇成员生存时间以及分簇效率这三个指标用以评估簇级网络稳定性.

5.1 平均簇持续时间

本文将 c_i 形成到解散这段时间表示为簇 c_i 在路网中的簇生存时间 $t_{c_i}^d$. 在合并过程中, 原有簇并非结束生命周期, 而是并入新簇继续生存, 故平均簇持续时间 t_{ad} 由非劣合并簇的平均持续时间表示:

$$t_{ad} = \frac{\sum_{c_i \in CC_b} t_{c_i}^d}{|CC_b|} \quad (8)$$

其中, CC_b 代表所有劣合并簇 c_b 的集合.

图 5 描述了不同算法随仿真时间变化的平均车辆簇网络生存时间. 由图可知, 随着仿真时间的增加, NFMC 算法在平均簇持续时间指标上呈现的性能优于另外两种算法, 比较簇形成后至仿真结束这段时间内

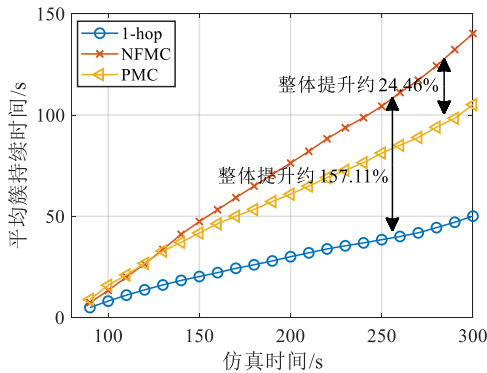


图 5 平均簇持续时间

各算法的结果, 可知 NFMC 算法相比 PMC 算法提高了约 24.46% 的平均簇持续时间, 相比 1-hop 算法提高了约 157.11%. 1-hop 方案在仿真过程中形成许多松散的单跳分簇, 易受某些成员的运动状态突变而导致簇解散, 因此具有最差平均簇生存时间. NFMC 算法优于 PMC 算法的原因则是随着仿真时间变大, 路网中的车辆数量增加, 由于加速行驶或者完成行程而脱离簇集网络的 CH 节点数量开始增大, 本研究所提出的主从簇头策略有效地应对了由于 CH 离开导致的簇集网络解散问题.

5.2 平均簇成员生存时间

平均簇成员生存时间表示网络中所有节点作为簇内成员节点的累计时长的平均值, 由图 6 可知, 多跳分簇有效提高了平均簇成员生存时间, 这是由于在多跳跟随的场景中, 簇集网络的容纳能力显著提升, 克服了单跳分簇易受移动性影响和生灭频度高的缺点. 此外, 本文所提的 NFMC 算法考虑了簇维持策略, 当簇成员与跟随对象断开时, 其可以根据跟随指标在簇内选择其他节点跟随, 因此原本断开跟随关系的节点仍存在一定概率留在原簇中, 继而提高了总体的簇成员生存时间, 相比于 PMC 算法具有更优的性能.

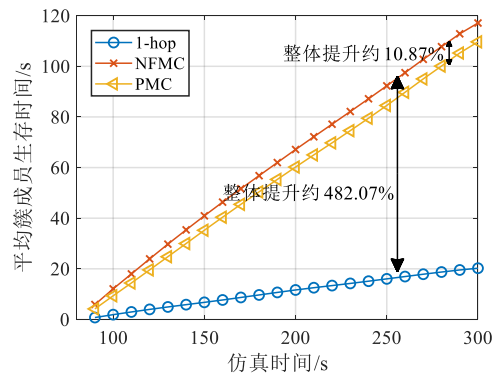


图 6 平均簇成员生存时间

5.3 分簇效率

图 7 比较了不同算法随仿真时间变化的车辆组网中非 UN 节点与总节点数量的比值, 即分簇效率. 分簇效率代表车辆节点形成簇集网络的能力. 本文所提的 NFMC 算法具有最高的分簇效率, 这是因为算法考虑了剪枝操作, 使得未成簇的车辆节点或与簇头之间超过最大跳数的车辆节点有概率和其他簇内节点形成跟随链路, 而 PMC 算法由于使用包含簇头和簇成员的持续时间和簇头变化次数作为综合指标, 当路网新加入车辆数量较大时, PMC 形成的网络簇无法同时处理所有节点, 从而导致部分节点仍处于未分簇状态. 随着仿真时间的增加, 路网中的车辆数量趋于稳定, 单位时间内路网新加入的车辆数量逐渐减少, PMC 可以通过簇合并不断扩大簇网络, 故其分簇效率逐渐靠近 NFMC, 但由于 PMC 缺少剪枝操作, 得到的簇结构复杂度较高, 导致部分节点与簇头间的跳数超过限定值, 无法成簇, 因此其分簇效率始终低于 NFMC 算法. 此外, 由于 1-hop 算法仅形成单跳网络, 只有车辆与其邻域车辆间的稳定性很高时才能成簇, 因此其分簇效率最低. 比较簇形成后至仿真结束这段时间内各算法的结果, 所提的 NFMC 算法相比于 PMC 算法的分簇效率提高了约 8.88%, 相比于 1-hop 算法提升了约 309.26%.

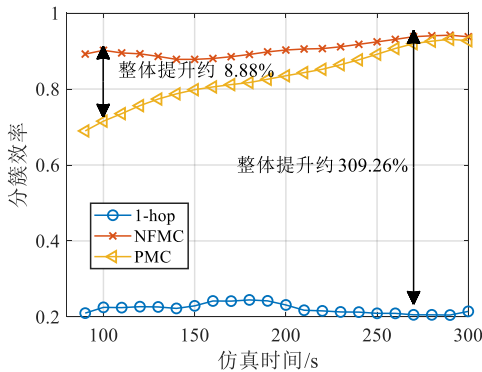


图7 分簇效率

5.4 分簇利用率

分簇利用率表示网络中车辆节点数与簇数的比值. 分簇利用率较高时, 车辆簇的规模较大, 可以减少RSU的管理压力, 并且使得簇内节点可以实现更大距离的多跳传输, 进而扩大了单个车辆节点的通信范围. 图8对比了PMC和NFM两种多跳策略是否采用本文提出的簇维护机制情况下的分簇利用率, 其中, NFM-Cun和PMCCun分别表示未采用簇维护机制情况下的NFM算法和PMC算法.

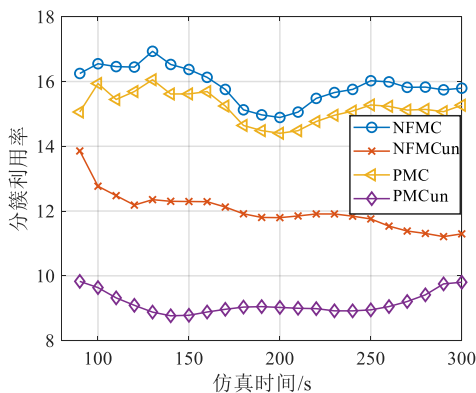


图8 分簇利用率

比较簇形成后至仿真结束这段时间内各算法的结果, 四种策略的平均分簇利用率分别为15.89、15.20、11.96、9.13. 图中也显示了仿真时间为250 s时四种策略的分簇利用率(分别为16.02、15.28、11.76、8.948). 由图8可知, 本文提出的簇维护机制有效提高了网络的分簇利用率, 且两种情况下NFM算法的分簇利用率均高于PMC算法. 此外, 不采用簇维护机制时, NFM-Cun算法显著优于PMCCun算法, 这是因为NFM-Cun采用了尽可能保持当前跟随状态的维持策略, 而PMCCun算法易受择优跟随机制的影响使得簇内节点脱离并依附到其他节点上.

6 总结与展望

针对车联网对车辆分簇稳定性的高要求, 本文提出了一种基于邻域跟随的多跳分簇算法. 该算法综合考虑了节点跟随度、LLT和信道容量积对车辆分簇稳定性的影响, 结合去环、剪枝、主从簇头及簇维护机制有效优化了簇网络结构, 提高了簇网络的鲁棒性. 经相关实验证明, 相比于1-hop算法和PMC算法, 本文提出的方法生成的车辆分簇网络在分簇稳定性和分簇效率等方面均得到了有效的提升. 后续研究可考虑交通流出现突发事件的场景以及V2R链路对算法结果的影响.

参考文献

- [1] 刘雷, 陈晨, 冯杰, 等. 车载边缘计算卸载技术研究综述[J]. 电子学报, 2021, 49(5): 861-871.
LIU L, CHEN C, FENG J, et al. A survey of computation offloading in vehicular edge computing networks[J]. Acta Electronica Sinica, 2021, 49(5): 861-871. (in Chinese)
- [2] 潘敏婷, 王韞博, 朱祥明, 等. 基于无标签视频数据的深度预测学习方法综述[J]. 电子学报, 2022, 50(4): 869-886.
PAN M T, WANG Y B, ZHU X M, et al. A survey on deep predictive learning based on unlabeled videos[J]. Acta Electronica Sinica, 2022, 50(4): 869-886. (in Chinese)
- [3] 曾胜, 黄小红, 李丹丹, 等. 基于多天线的移动边缘计算网络公平计算卸载策略研究[J]. 电子学报, 2023, 51(9): 2457-2468.
ZENG S, HUANG X H, LI D D, et al. Fair computation offloading for a multi-antenna NOMA aided mobile edge computing network[J]. Acta Electronica Sinica, 2023, 51(9): 2457-2468. (in Chinese)
- [4] 王巍, 彭力, 赵继军, 等. 移动物联网非完整约束中继的协同任务规划[J]. 电子学报, 2019, 47(6): 1251-1259.
WANG W, PENG L, ZHAO J J, et al. Cooperative task planning of ground relay with nonholonomic constraints in mobile Internet of Things based on UAVs[J]. Acta Electronica Sinica, 2019, 47(6): 1251-1259. (in Chinese)
- [5] ZIAGHAM A, NOORIMEHR M. MOSIC: Mobility-aware single-hop clustering scheme for vehicular ad hoc networks on highways[J]. International Journal of Advanced Computer Science and Applications, 2016, 7(9): 424-431.
- [6] ABBAS A H, MOHAMMED I, AUDAH L, et al. A novel intelligent cluster-head (ICH) to mitigate the handover problem of clustering in VANETs[J]. International Journal of Advanced Computer Science and Applications, 2019, 10(6): 194-203.
- [7] KHAYAT G, MAVROMOUSTAKIS C X, MASTORA-

- KIS G, et al. VANET clustering based on weighted trusted cluster head selection[C]//2020 International Wireless Communications and Mobile Computing (IWCMC). Piscataway: IEEE, 2020: 623-628.
- [8] KATIYAR A, GUPTA S K, SINGH D, et al. A dynamic single-hop clustering algorithm (DSCA) in VANET[C]//2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). Piscataway: IEEE, 2020: 1-6.
- [9] TEMURNIKAR A, VERMA P, DHIMAN G. A PSO enable multi-hop clustering algorithm for VANET[J]. International Journal of Swarm Intelligence Research, 2022, 13 (2): 1-14.
- [10] AZIZIAN M, CHERKAOUI S, HAFID A S. A distributed D-hop cluster formation for VANET[C]//2016 IEEE Wireless Communications and Networking Conference. Piscataway: IEEE, 2016: 1-6.
- [11] DUTTA A K, ELHOSENY M, DAHIYA V, et al. An efficient hierarchical clustering protocol for multihop Internet of vehicles communication[J]. Transactions on Emerging Telecommunications Technologies, 2020, 31(5): e3690-e3690.
- [12] ZHANG D G, GE H, ZHANG T, et al. New multi-hop clustering algorithm for vehicular ad hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(4): 1517-1530.
- [13] ALSUHLI G H, KHATTAB A, FAHMY Y A. A mobility-based double-head clustering algorithm for dynamic VANET[C]//2018 International Japan-Africa Conference on Electronics, Communications and Computations (JACECC). Piscataway: IEEE, 2018: 91-94.
- [14] 王义和. 离散数学引论[M]. 哈尔滨:哈尔滨工业大学出版社, 1986.
WANG Y H. Introduction to Discrete Mathematics[M]. Harbin: Harbin Institute of Technology Press, 1986. (in Chinese)
- [15] SRIVASTAVA A, PRAKASH A, TRIPATHI R. An adaptive intersection selection mechanism using ant Colony optimization for efficient data dissemination in urban VANET[J]. Peer-to-Peer Networking and Applications, 2020, 13(5): 1375-1393.
- [16] LI Z Q, USMAN M, TAO R T, et al. A systematic survey of regularization and normalization in GANs[J]. ACM Computing Surveys, 55(11): 232.

作者简介



林志坚 男,1984年出生于福建省漳州市. 现为福州大学电子信息工程系副教授,硕士生导师. 主要研究方向为车联网、边缘缓存与计算卸载. 中国电子学会会员编号:E190050609M.
E-mail: zlin@fzu.edu.cn



吴文浩 男,2000年3月出生于福建省宁德市. 福州大学物理与信息工程学院研究生. 主要研究方向为车联网、边缘缓存与计算卸载.
E-mail: 221127202@fzu.edu.cn



陈平平 男,1986年出生于福建省泉州市. 现为福州大学电子信息工程系教授,博士生导师. 主要研究方向为无线通信、信息传输与计算卸载. 中国电子学会会员编号:E190021215M.
E-mail: ppchen.xm@gmail.com