

# 负载约束的 C-V2X 车辆缓存节点选择算法

徐哲鑫, 高楷蒙, 贾文康, 吴怡

(福建师范大学光电与信息工程学院, 福建 福州 350007)

**摘要:** 为了解决城市环境下的 C-V2X 车辆拓扑高度动态化且车辆节点负载能力有限的问题, 提高车辆缓存的利用率, 减轻基站负荷, 提出了负载约束下的车辆缓存节点选择算法。首先, 通过定义链路稳定性度量, 构建预测权重邻接矩阵, 微观地描述车辆拓扑关系; 其次, 在负载约束和无重叠覆盖约束下构建目标函数, 以最少的缓存节点实现全覆盖且最大化簇平均链路权重; 最后, 引入贪婪思想并合理定义节点状态, 求解负载约束下车辆拓扑的最小支配集, 并择优选择服务邻居节点。仿真结果表明, 所提算法在缓存节点个数和簇平均链路权重均值方面接近全局最优, 其重复应答率恒为零, 请求应答率可达理论上界并可有效提高缓存源应答次数。

**关键词:** C-V2X; 缓存节点选择; 最小支配集; 负载约束

**中图分类号:** TN92

**文献标识码:** A

**DOI:** 10.11959/j.issn.1000-436x.2021065

## Vehicular cache nodes selection algorithm under load constraint in C-V2X

XU Zhixin, GAO Kaimeng, JIA Wenkang, WU Yi

School of Optoelectronics and Information Engineering, Fujian Normal University, Fuzhou 350007, China

**Abstract:** In order to solve the problem that the C-V2X vehicle topology in urban environment was highly dynamic and the load capacity of vehicle nodes was limited, and improve the utilization of vehicular cache resources and reduce the load of base station, a vehicle cache nodes selection algorithm under load constraints was proposed. Firstly, by defining the link stability metric, the predicted weight adjacency matrix was constructed to describe the vehicular micro-topology in essence. Next, the objective function was further constructed under the load constraints and non-overlapping coverage constraint, which maximized the average link weight of the clusters by using the least cache nodes. Finally, the greedy concept was then introduced and the node states were reasonably defined. As a result, the minimum dominating set of the vehicle topology was figured out under the load constraints. Besides, the serviced neighbor nodes were then determined preferentially. The simulation results show that the proposed algorithm is close to the global optimal results in terms of the number of cache nodes and the average weight of cluster links. Moreover, the repeated response ratio of the proposed algorithm is always zero while the request response ratio can achieve the theoretical upper bound. Furthermore, the response times of cache resources can be also effectively improved.

**Keywords:** C-V2X, cache node selection, minimum dominating set, load constraint

### 1 引言

随着现代交通和科技的发展, 我国汽车行业的

需求量日益增长, V2X (vehicle to everything) 受到工业界和学术界的广泛关注。高效的 V2X 系统开发基于大量可靠功能的传感器, 通过在车辆、

收稿日期: 2020-09-23; 修回日期: 2020-12-20

通信作者: 吴怡, wuyi@fjnu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.U1805262, No.61871131); 福建省自然科学基金资助项目 (No.2020J01132418)

**Foundation Items:** The National Natural Science Foundation of China (No.U1805262, No.61871131), The Natural Science Foundation of Fujian Province (No. 2020J01157)

行人和道路基础设施之间交换关键信息, 提供增强的环境感知<sup>[1]</sup>。为充分利用蜂窝移动通信网络的技术优势, C-V2X (cellular vehicle to everything) 应运而生, 将车辆与一切事物连接, 并向 NR-V2X (new radio vehicle to everything) 演进, 在高度动态的车辆拓扑环境下, 实现低时延、高可靠的数据传输。

V2X 融合了车辆和基站之间的蜂窝通信以及车辆之间的直通通信, 2 种模式相互补充, 实现基站和车辆之间的负载均衡<sup>[2]</sup>。若将车辆节点作为缓存节点, 可通过车辆之间的连接为其他车辆提供数据服务, 以降低数据响应时延, 减轻基站的负荷, 减少因基站切换引起的开销, 优化基站通信资源分配。在 C-V2X 的演进中, 3GPP R14 定义了针对车载通信的 mode-3 和 mode-4, 这 2 种模式都支持车辆间直通通信, 区别在于 mode-3 通过蜂窝网络集中式分配无线资源, 而 mode-4 由车辆分布式自主选择无线资源。相比之下, mode-3 中基站能获取更完整的网络状态和不同车辆对资源的需求信息, 更有效地利用无线资源<sup>[3]</sup>, 因此本文主要研究 C-V2X mode-3 下的车辆缓存节点选择问题。城市环境 C-V2X 中高度动态的车辆拓扑和车辆负载约束使车辆缓存节点的选择面临挑战。1) 增加缓存节点的覆盖面, 则更多的车辆可通过 V2V (vehicle to vehicle) 从缓存节点获取文件; 2) 减少缓存节点数量, 以减少指派缓存节点的控制开销和预置缓存文件的带宽开销, 并且过多的缓存节点将加剧信道竞争; 3) 降低应答冗余, 即减少缓存节点对于同一请求的重复应答, 以节省带宽开销。因此, 合适的缓存节点选择成了亟须解决的问题。

由于车联网本身具有快速的拓扑变化, 且车辆节点服务能力有限, 现有研究较少利用车辆可获取的移动轨迹信息对链路变化进行预判, 同时忽视了节点的服务能力上限以及缓存节点的覆盖性等方面, 故车联网中节点筛选的研究尚不完善。本文提出基于负载约束的 C-V2X 车辆缓存节点选择算法, 主要贡献有以下几个方面。

1) 合理定义节点状态, 即状态未定节点、服务邻居节点以及缓存节点, 为缓存节点选择奠定重要基础。其中, 状态未定节点为初始状态, 通过所提算法将所有的状态未定节点转化为服务邻居节点或缓存节点, 实现了节点间的无重叠全覆盖; 服务邻居节点有别于传统邻居节点的定义, 是缓存节点

根据每个周期可服务的节点个数上限择优筛选出的邻居节点的子集。

2) 定义链路稳定性度量, 并构建预测权重邻接矩阵, 从微观层面精确描述未来周期的拓扑关系, 并以此筛选缓存节点, 使缓存节点及其服务邻居节点的选择更加合理。

3) 根据节点的服务能力的局限性, 将节点一个周期内可服务的节点个数上限作为负载约束, 提出了在给定拓扑下, 无重叠全覆盖的最少缓存节点及其服务邻居节点的选择算法。该算法改进传统的最小支配集筛选算法, 结合 C-V2X 场景中节点负载约束, 以匹配实际节点响应能力, 提升了算法的实用性, 同时实现了不重叠全覆盖, 优化了系统带宽利用率。

4) 将所提算法与穷举算法计算得到的全局最优结果对比, 经大量仿真, 统计结果表明, 所提算法在缓存节点个数和簇平均链路权重均值方面都接近全局最优结果。

5) 将所提算法与 k-means 无监督聚类算法、最大连接度算法和 CCMP 算法进行对比。所提算法在缓存节点及其服务邻居节点的选择方面更加合理, 实现了无重叠全覆盖, 请求应答率、重复应答率、缓存源应答次数均值等性能指标均优于对比算法, 并且重复应答率恒为零, 请求应答率可达理论上界。

## 2 相关工作

在 C-V2X 场景下, V2X 的通信效率及稳定性至关重要, 通过 V2V 通信可以使整个网络中的系统负荷得到合理的分配, 自适应地快速实现车联网业务的高可靠低时延通信<sup>[2]</sup>。基于 D2D (device to device) 的缓存策略已经被证明能有效提升网络性能<sup>[4-7]</sup>, 借助于 V2V 的 C-V2X 车辆缓存也将有利于车辆节点间数据共享, 其中车辆缓存节点的选择至关重要。

车辆缓存节点为其通信范围内的节点提供文件共享服务, 也可广义地视为形成车辆簇, 以车辆缓存节点为簇头, 以被服务节点为簇成员。因此, 除缓存节点筛选算法外, 成簇以及簇头选择算法可为车辆缓存节点的选择提供重要参考。

利用节点属性构建节点度量值, 再通过门限、最值等筛选条件求得簇头并成簇是常见的算法思路。这在无线传感网等节点固定的网络<sup>[8-10]</sup>以及车联网等节点移动的网络<sup>[11-18]</sup>中均得到广泛应用。在

无线传感网中, 由于节点能量有限, 节点的剩余能量是着重考虑的因素。Ray 等<sup>[8]</sup>以节点剩余能量、节点到基站的距离、连续轮数为参数构建节点度量, 度量值小于所设定门限的节点为簇头。Qiao 等<sup>[9]</sup>将节点剩余能量和节点到数据采集中心的距离整合为竞争半径, 将该值最大的节点作为簇头, 其他节点就近成簇。Saloni 等<sup>[10]</sup>将区域划分为网格, 网格中选择剩余能量最大的节点作为每轮的簇头。多因素加权获得度量值并与门限比较得到簇头并成簇的方法值得借鉴, 但是节点移动的网络中节点拓扑结构变化导致所需要考虑的因素有别于固定网络。Gao 等<sup>[11]</sup>将中断容忍网中每个节点到其他节点可达率的平均概率作为度量值, 然后选择度量值排名靠前的若干节点作为簇头。车联网中移动性信息是关键因素之一, 尤其是速度。Rawashdeh 等<sup>[12]</sup>考虑了车联网中节点的速度、位置、节点度和方向等移动信息, 将它们整合为适应性度量值, 从而选择该值最大的节点成为簇头, 并将其通信范围内速度差小于某个门限的节点成簇。Daknou 等<sup>[13]</sup>以速度最慢的节点作为成簇发起点, 将高速公路分区, 每个簇中节点以与邻居的距离和速度差作为参数, 加权得到适应度量值, 度量值最小的节点作为簇头。Farooq 等<sup>[14]</sup>令不同车道的车辆各自成簇, 同一车道速度相近的车辆成簇, 速度最接近设定门限的节点作为簇头。速度关系本质上体现的是节点间的连接关系, 此外, 还可以根据节点密度划分出热区, 并选择在热区的平均逗留时间更长的节点作为缓存节点<sup>[15]</sup>, 也可以直接研究节点连接关系, 并综合其他因素完成簇头筛选和成簇。Alsuhi 等<sup>[16]</sup>综合考虑节点与邻居的平均距离、速度差、连接度、信道质量、链路持续时间并归一化后加权得到度量值, 该度量值越大, 越有资格成为簇头, 同时还选择了备用簇头, 簇头收到入簇请求后, 若其簇成员数目未达上限则纳入簇。Qi 等<sup>[17]</sup>通过分割道路完成初始分簇, 只有当两节点链路的持续时间超过给定门限时, 才记为有效链路; 根据有效链路统计节点的连接度, 在连接度大于门限的节点中选择距离 RSU (road side unit) 最近的节点作为簇头。Cheng 等<sup>[18]</sup>通过连接预测评估节点间拓扑关系, 若节点的邻居节点密度大于门限, 则将其设置为核心车辆节点。

尽管车联网拓扑高度动态, 但车辆节点之间有明显的跟随特性, 在道路约束下的车辆轨迹也存在规律性, 这些都体现在更本质的微观邻接关系

中。因此, 本文也从节点的连接关系入手, 与上述文献不同的是, 将节点的属性转化为更本质的链路属性, 构建链路稳定性度量, 并且采用基于贪婪思想的迭代算法筛选簇头并成簇, 而非使用门限或最大值选择簇头。这是因为车联网具有高度动态的环境, 节点间相对关系复杂多变, 难以用绝对门限清晰划分度量值的优劣。

利用优化算法逐步得到优化的节点作为簇头并成簇也是行之有效的手段, 还可避免设定门限的缺陷。Shin 等<sup>[19]</sup>将节点的剩余能量和连接度整合为度量 flooding value, 通过迭代交替地将节点的该度量值替换为最大和最小邻居度量值, 收敛之后根据迭代过程中度量值的规律筛选簇头。针对 VANET (vehicular ad-hoc network) 动态环境, Fathian 等<sup>[20]</sup>提出了多目标数据包络分析聚类算法的数学聚类模型和蚁群算法的启发式聚类模型, 聚类后将最靠近簇中心位置的节点作为簇头。Ahmad 等<sup>[21]</sup>引入博弈思想, 综合考虑车辆的速度、位置、方向和 LTE (long term evolution) 链路质量等参数, 提出合作利益感知聚类算法, 以节点是否接受成为簇头作为策略, 最大化整体利益并完成簇头筛选, 同时采用轮值簇头机制提升公平性。采用节点集合覆盖理论的算法可以获得精简优化的簇头集合, 例如, Liu 等<sup>[22]</sup>提出基于广义控制集和局部内容流行度的内容中心移动自组网协同缓存方案, 根据连接度筛选簇头节点, 并将给定多跳范围内的邻居节点纳入簇成员。Liu 等<sup>[23]</sup>提出基于最小顶点覆盖集理论的静态网络节点选择算法, 通过社会化关系的协同缓存来确定缓存点, 以解决车辆流动性带来的不连接问题。Li 等<sup>[24]</sup>考虑用户分布和流量负载的情况, 构建了基站和用户之间的二部图, 并通过求解最小支配集获得最优传输路径, 基站被默认为簇头, 与其连接的用户可视为簇成员。

在上述优化方法中, 迭代运算是重要的组成部分, 快速收敛并减少交互开销至关重要, 并且若带宽有限且需要实现全覆盖, 相较于其他优化方法, 基于最小支配集的方法可以快速计算出簇头。但其缓存节点服务能力有限, 每个周期只能服务有限数量的邻居节点, 目前研究较少直接考虑该因素。因此, 利用交通流规律以及拓扑邻接关系, 实现缓存节点和服务邻居节点的筛选有待进一步研究。

针对以上问题, 本文提出了基于负载约束的 C-V2X 车辆缓存节点选择算法。该算法定义了链路

稳定性度量以及 3 种节点状态，充分利用车辆轨迹的可预测性，在预测未来周期权重邻接矩阵的基础上，以服务邻居节点上限作为负载约束，完成最小支配集筛选，以最少缓存节点不重叠地覆盖所有节点。所提算法对于 C-V2X 的动态拓扑具有较强的自适应性和实效性，在请求应答、重复应答率以及缓存源平均响应次数方面具有显著优势，能有效提高车辆缓存节点的利用率，减轻基站负荷。

### 3 系统模型及问题描述

#### 3.1 系统模型

本文研究城市环境的 C-V2X 缓存文件传输场景，如图 1 所示。该场景中，主干道交汇处形成岔路口，每条主干道为双向多车道。设一个通信半径为  $R_B$  的基站覆盖该路口，其覆盖范围内车辆数为  $N_{veh}$ ，总车辆集合为  $\mathcal{N} = \{n_1, n_2, \dots, n_{N_{veh}}\}$ 。

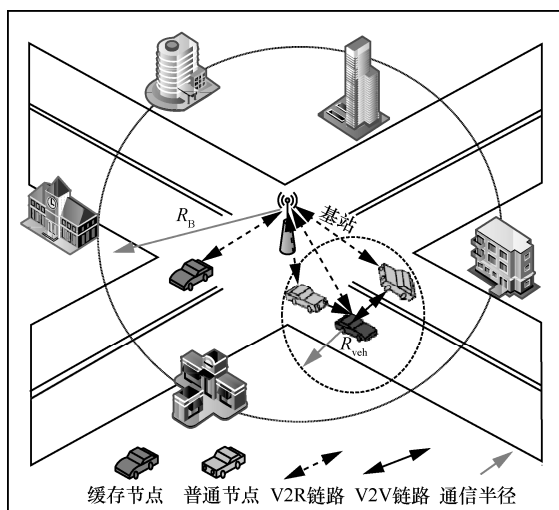


图 1 城市环境的 C-V2X 缓存文件传输场景

假设车辆节点都具有相同的缓存空间，可容纳  $C$  个文件。车辆节点既可以发出请求，也可以作为请求响应者。假设车辆内都装有 GPS 等定位设备，可以实时获取车辆的轨迹信息并上报给基站。在第  $t$  周期，第  $i \in \{1, 2, \dots, N_{veh}\}$  辆车的轨迹信息定义为  $p_i^t = (x_i^t, y_i^t, v_i^t)$ ，其中， $(x_i^t, y_i^t)$  为位置坐标， $v_i^t$  为速度。设车辆的通信半径为  $R_{veh}$ ，每辆车可以从通信范围内的缓存节点获取文件，也可以从所归属的基站获取。

基站作为区域服务者，为覆盖区内车辆节点提供文件服务和资源控制管理，基站通信半径为  $R_B$ 。通过给基站配置存储设备和计算设备，可让基站具备缓

存能力和边缘计算能力。基站采用 3GPP R14 版本中设定的 mode-3 模式管理车辆间通信的无线资源分配，以实现更高效的子载波利用率<sup>[3]</sup>。车辆节点在基站的集中控制下与缓存源节点通信，获取所需文件。同时，基站在缓存源节点的指派上借鉴底层的半持续调度 (SPS, semi-persistent scheduling) 机制，即在给定时间段内车辆节点由指定缓存源节点管辖，期间各子帧的数据获取均优先请求该指定缓存节点。这与底层的无线资源 SPS 机制匹配，以优化数据分组传输效率。

#### 3.2 问题描述

普通节点会先向自己所连接的缓存节点发起文件请求。当缓存节点已缓存相应文件时，则响应请求；否则用户经一跳或者多跳 V2V 链路向基站发出该文件请求，由基站经回程链路获取数据后进行响应。通过回程链路响应将严重增加响应时延，并且城市场景下高密度交通流发出的大量文件请求将导致基站过载，无法同时响应过多文件请求。为了减少通过回程链路的文件响应时延并减轻基站负载，应使车辆节点的请求尽可能由周边车辆缓存节点提供响应，即提高车辆缓存节点的利用率。具体地，应提高车辆节点的请求应答，即提高当前周期所有车辆发送的文件请求中，能被缓存节点响应的请求的比例，该比例越大说明越多的请求能被周边缓存节点响应，对基站负载的分流作用越明显。请求应答比传统的缓存响应率更直接地体现了请求用户成功获取所需文件的效率。

缓存文件流行度估计及分配策略不在本文讨论范围内，设文件流行度服从 Zipf 分布<sup>[6, 25-26]</sup>，节点发出的请求也服从该分布，缓存节点缓存最流行的前  $C$  个文件<sup>[27]</sup>。将请求应答最大化问题转化为节点全覆盖问题，即当选择的缓存节点能够覆盖所有普通节点时，每个普通节点总在至少一个缓存节点的通信范围内。那么，从概率意义上，当缓存节点在其缓存空间  $C$  的约束下缓存最流行的  $C$  个文件时，将实现请求应答的最大化，其理论上限为 Zipf 分布 CDF (cumulative distribution function) 中  $C$  对应的值。

针对节点全覆盖问题，在本文的系统中，假设基站作为区域服务者具有宏观调控的功能，可获取覆盖区域内所有车辆节点的轨迹信息。基站将根据第  $t$  周期车辆轨迹信息，预测第  $t+1$  周期的车辆位置，从而生成预测权重邻接矩阵

$$W^{t+1} = [\omega_{i,j}^{t+1}]_{N_{veh} \times N_{veh}}, \quad \forall i, j \in [1, N_{veh}] \quad (1)$$

其中, 链路的权重  $\omega_{i,j}^{t+1}$  表示第  $t+1$  周期车辆节点  $i$  和  $j$  之间链路稳定性度量的值。

基站利用预测的  $\mathbf{W}^{t+1}$ , 采用缓存节点选择算法计算缓存节点集合  $\mathcal{M}^{t+1} = \{m_1, m_2, \dots, m_{N_M}\}$ , 其节点个数为  $N_M$ 。为减少基站配置及管理缓存节点的开销, 并且减少缓存节点之间信道竞争, 应使满足最优性能时, 选择的缓存节点个数最少, 则目标函数为

$$\text{P1: } \min_N |\mathcal{M}^{t+1}| \quad (2)$$

对于任一缓存节点  $m_k, k \in \{1, 2, \dots, N_M\}$ , 在给定数据帧长和数据传输速率的条件下, 其每个周期的服务能力有限, 设最多只能响应  $N_{\max}$  个邻居节点的请求, 则将该约束称为负载约束。因此, 所研究的问题转化为负载约束下的节点全覆盖问题, 即缓存节点  $m_k$  将在其通信半径范围内所有节点中选择不多于  $N_{\max}$  个节点作为服务邻居节点。将缓存节点  $m_k$  的  $q$  个服务邻居节点集合记为

$$\mathcal{X}_{m_k}^{t+1} = \{\mathcal{X}_{m_k,1}^{t+1}, \mathcal{X}_{m_k,2}^{t+1}, \dots, \mathcal{X}_{m_k,q}^{t+1}\} \\ \text{C1: } q \leq N_{\max} \quad (3)$$

这些缓存节点可以覆盖基站管辖范围内其他所有普通节点, 即缓存节点集合  $\mathcal{M}^{t+1}$  与所有缓存节点的服务邻居节点集合的并集等于  $\mathcal{N}$ , 表示为

$$\text{C2: } \mathcal{M}^{t+1} \cup \mathcal{X}_{m_1}^{t+1} \cup \dots \cup \mathcal{X}_{m_{N_M}}^{t+1} = \mathcal{N} \quad (4)$$

为提高系统带宽利用率, 应减少重复应答率。重复应答率即多个缓存节点重复响应同一请求的比例。因此, 令每个普通节点只与一个缓存节点建立连接, 以实现无重复响应, 即一个普通节点在同一周期内对一份文件的请求不会同时被 2 个以上的缓存节点响应。因此, 所研究的问题进一步转化为在负载约束和无重叠覆盖约束下, 以最少的缓存节点实现节点全覆盖问题, 即任意 2 个缓存节点  $m_k$  和  $m_u$  的邻居集合的交集为空, 表示为

$$\text{C3: } \mathcal{X}_{m_k}^{t+1} \cap \mathcal{X}_{m_u}^{t+1} = \emptyset, \quad k \neq u \quad (5)$$

另外, 每个缓存节点及其服务邻居节点构成一个簇, 缓存节点为簇头, 服务邻居节点为簇成员。为提高簇稳定性和传输可靠性, 在确保每个普通节点且仅能被一个缓存节点覆盖的前提下, 每个缓存节点选择其覆盖范围内链路权重大的节点作为服务邻居节点。缓存节点选择的服务邻居节点应使簇平均链路权重最大化, 因此, 所研究的问题转化为在负载约束和无重叠覆盖约束下, 以最少的缓存

节点实现簇平均链路权重最大化的节点全覆盖问题。设缓存节点  $m_k$  的最优邻居集合为  $\mathcal{X}_{m_k}^{*t+1}$ , 对应的簇平均链路权重为  $\bar{\mathbf{W}}_{m_k}^{*t+1}$ , 则目标函数为

$$\text{P2: } \max_{\{\mathcal{X}_{m_k}^{*t+1}\}} \text{E} \left[ \sum_{k=1}^{N_M} \bar{\mathbf{W}}_{m_k}^{*t+1} \right], \quad \forall m_k \in \mathcal{M}^{t+1} \quad (6)$$

式(6)表示每个缓存节点应选择最优邻居, 以使簇平均链路权重的期望最大化。

综上所述, 可构建待优化的目标函数, 如式(7)所示。

$$\text{P: } \begin{cases} \text{P1: } \min_N |\mathcal{M}^{t+1}| \\ \text{P2: } \max_{\{\mathcal{X}_{m_k}^{*t+1}\}} \text{E} \left[ \sum_{k=1}^{N_M} \bar{\mathbf{W}}_{m_k}^{*t+1} \right], \quad \forall m_k \in \mathcal{M}^{t+1} \end{cases} \\ \text{s.t. } \text{C1, C2, C3} \quad (7)$$

该优化问题具有 NP-hard 性质, 在高度动态的车辆拓扑环境下, 为满足低时延的传输要求, 求解最优解将导致算法复杂度和计算时延恶化, 因此, 本文借鉴贪婪思想, 提出负载约束下的最小支配集算法, 快速计算可行次优解。仿真表明, 所提算法的缓存节点个数和簇平均链路权重接近穷举法计算的最优结果。

## 4 算法流程及关键步骤

以车辆节点轨迹信息为基础, 以最小化缓存节点个数以及最大化簇平均链路权重的期望为目标, 采用负载约束的最小支配集算法, 实现车辆节点的全覆盖。首先, 基站收集当前周期内所有节点的行驶信息, 以预测得到所覆盖的范围内下一周期的拓扑; 然后, 定义链路稳定性度量, 以构建预测权重邻接矩阵; 最后, 根据预测权重邻接矩阵实现下个周期的缓存节点选择。

### 4.1 链路稳定性度量

定义链路稳定性度量  $\omega_{i,j}^t, \forall i, j \in [1, N_{\text{veh}}]$  表示在第  $t$  周期车辆节点  $i$  和  $j$  之间归一化通信距离容差和归一化链路持续时间的加权和。在第  $t$  周期, 设车辆节点  $i$  和  $j$  在相互的通信范围内且距离为  $\varepsilon_{i,j}^t$ , 车辆节点的通信半径为  $R_{\text{veh}}$ , 则通信距离容差为  $R_{\text{veh}} - \varepsilon_{i,j}^t$ , 从而可得归一化通信距离容差为

$$\tilde{\varepsilon}_{i,j}^t = \begin{cases} \frac{R_{\text{veh}} - \varepsilon_{i,j}^t}{R_{\text{veh}}}, & d_{\min} \leq \varepsilon_{i,j}^t < R_{\text{veh}} \\ 0, & \varepsilon_{i,j}^t \geq R_{\text{veh}} \end{cases} \quad (8)$$

其中,  $d_{\min}$  表示在安全距离限制下的两车最小距离。该值越大说明两车距离越近, 相同遮挡条件下的信道质量越好, 并且在两车的车速均不变时, 链路持续时间越长。当车间距大于或等于车辆通信半径时, 两车通信距离容差均视为 0, 即无法通信。

设周期间隔为  $\Delta t$ , 车辆节点  $i$  和  $j$  在第  $t$  周期中, 将相互处于对方覆盖范围内的时间长度定义为链路持续时间  $\tau_{i,j}^t$ , 该值由轨迹预测确定<sup>[28]</sup>。由于链路持续时间每周期更新, 当值大于周期间隔时, 将链路持续时间的上限设为  $\Delta t$ , 再将其对周期间隔进行归一化, 得到归一化链路持续时间为

$$\tilde{\tau}_{i,j}^t = \begin{cases} \frac{\tau_{i,j}^t}{\Delta t}, & \tau_{i,j}^t < \Delta t \\ 1, & \tau_{i,j}^t \geq \Delta t \end{cases} \quad (9)$$

链路持续时间越长, 说明两车的拓扑关系越稳定, 但并不意味着两车之间的信道质量越好, 链路稳定性还要考虑两车之间的传输距离、遮挡情况以及干扰等。本文着重研究缓存节点选择, 将链路稳定性度量简化为归一化通信距离容差和归一化链路持续时间的加权和, 即

$$\omega_{i,j}^t = \rho \tilde{\epsilon}_{i,j}^t + (1 - \rho) \tilde{\tau}_{i,j}^t \quad (10)$$

其中,  $\rho \in [0, 1]$  为加权因子。基站根据第  $t$  周期的轨迹信息完成第  $t+1$  周期的轨迹预测并计算  $\omega_{i,j}^{t+1}$ ,

以此作为链路权重, 并根据式(1)得到  $\mathbf{W}^{t+1}$ 。

#### 4.2 基于负载约束的最小支配集算法

针对预测权重邻接矩阵  $\mathbf{W}^{t+1}$ , 定义 3 种节点的状态, 即状态未定节点、服务邻居节点以及缓存节点, 分别对应状态标志位 0、1、2, 从而可构建第  $t+1$  周期的节点标志位向量为

$$\mathbf{G}_{1 \times N_{\text{veh}}}^{t+1} = (g_1^{t+1}, g_2^{t+1}, \dots, g_{N_{\text{veh}}}^{t+1}) \\ \forall g_i^{t+1} \in [0, 1, 2], \quad i \in \{1, 2, \dots, N_{\text{veh}}\} \quad (11)$$

所有节点状态标志位均初始化为 0, 根据预测权重邻接矩阵计算每个节点连接状态未定节点的连接度, 即该节点通信覆盖范围内连接的状态未定节点的个数。在服务邻居个数上限  $N_{\max}$  (即负载约束) 下, 综合考虑节点连接度和链路稳定性度量, 筛选最少的缓存节点  $\mathcal{M}^{t+1}$  构成最小支配集, 并择优筛选各缓存节点的服务邻居节点, 实现无重叠的全覆盖, 并最大化簇平均链路权重均值, 即最优化式(7)的目标函数。具体步骤如算法 1 所示。

#### 算法 1 基于负载约束的最小支配集算法

**输入** 预测权重邻接矩阵  $\mathbf{W}^{t+1}$ , 负载约束  $N_{\max}$

**输出** 缓存节点集合  $\mathcal{M}^{t+1}$ , 缓存节点的服务邻居节点集合  $\{\mathcal{X}_{m_1}^{t+1}, \dots, \mathcal{X}_{m_{N_M}}^{t+1}\}$

1) 初始化  $\mathbf{G}_{1 \times N_{\text{veh}}}^{t+1}$ 。

2) 计算每个节点连接状态未定节点的连接度  $Z_i$ ,  $i \in \{1, 2, \dots, N_{\text{veh}}\}$ 。

3) 将  $Z_i = 0$  的孤立节点作为缓存节点, 节点对应的标志位  $g_i^{t+1}$  置为 2。

4)  $Z_i = 1$  节点的标志位  $g_i^{t+1}$  置为 1, 其邻居节点作为缓存节点, 对应的标志位  $g_i^{t+1}$  置为 2。若有多个  $Z_i = 1$  的节点, 则优先选择链路权重最大的节点的邻居节点作为缓存节点纳入  $\mathcal{M}^{t+1}$ , 并根据步骤 6)~步骤 8)选择服务邻居节点。

5) 若不存在  $Z_i = 0, 1$  的节点, 则选择  $Z_i$  最大的节点作为缓存节点。若有多个节点都满足该条件, 则根据  $\mathbf{W}^{t+1}$  选择平均邻居链路最大的节点作为缓存节点纳入  $\mathcal{M}^{t+1}$ , 并将其标志位置 2。

6) 若与缓存节点  $m_k$ ,  $k \in \{1, 2, \dots, N_M\}$  相连的邻居节点个数小于  $N_{\max}$ , 则将其邻居节点均纳入服务邻居节点集合  $\mathcal{X}_{m_k}^{t+1}$ 。否则, 将  $m_k$  的邻居节点按照  $Z_i$  值从小到大排序得到序列  $Z^{\text{sort}}$ , 选择前  $N_{\max}$  个邻居作为服务邻居节点纳入  $\mathcal{X}_{m_k}^{t+1}$ , 并置对应的标志位为 1。

7) 设  $\mathcal{X}_{m_k}^{t+1}$  已纳入  $Z^{\text{sort}}$  中排序前  $u$  个邻居节点, 则  $m_k$  还可选择  $N_{\max} - u$  个邻居节点作为服务节点。若在  $Z^{\text{sort}}$  中, 有多于  $N_{\max} - u$  个邻居节点的连接状态未定节点的连接度等于排序中第  $u+1$  位  $Z_{u+1}^{\text{sort}}$  值时, 计算  $m_k$  与邻居节点的链路权重在该邻居节点所有链路权重按从大到小排序中的位次, 选择位次最高的前  $N_{\max} - u$  个邻居节点作为服务节点纳入  $\mathcal{X}_{m_k}^{t+1}$ , 并置对应的标志位为 1。

8) 若  $m_k$  与多个邻居节点的链路权重在邻居节点所有链路权重排序中位次相同, 则选择  $m_k$  与邻居节点的链路权重最大的邻居节点作为服务节点纳入  $\mathcal{X}_{m_k}^{t+1}$ , 并置对应的标志位为 1。

9) 重复执行步骤 2)~步骤 8), 直到所有节点都被标记为 1 或 2, 即不存在标志位等于 0 的节点, 算法结束, 输出  $\mathcal{M}^{t+1}$  和  $\{\mathcal{X}_{m_1}^{t+1}, \dots, \mathcal{X}_{m_{N_M}}^{t+1}\}$ 。

在上述算法中, 与传统的连接度定义不同, 算法中定义的连接度  $Z_i$  ( $i \in \{1, 2, \dots, N_{\text{veh}}\}$ ) 是指节点

连接状态未定节点的个数。使用重新定义的连接度，可以避免不完全覆盖，并且算法执行过程中着重处理未被覆盖的节点，从而提高算法效率。

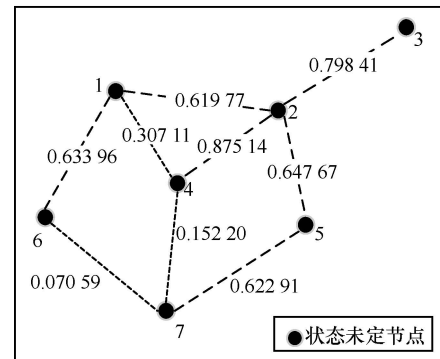
在缓存节点选择方面，优先考虑  $Z_i=0$  的孤立节点，将其设置为缓存节点，一方面可以获取自身所需的文件，另一方面还可以作为携带转发的锚点。其次，考虑  $Z_i=1$  的节点，因为这些节点的邻居节点有且仅有一个，借鉴传统最小支配集的构建规则，将其邻居节点置为缓存节点。若多个节点的连接度为 1，则优先处理链路权重大的节点，因为这些节点能由邻居节点提供更稳定的数据传输，提高数据传输成功率。在没有  $Z_i=0,1$  的节点时选择连接度最大的节点作为缓存节点，借鉴贪婪思想，最大程度地完成节点覆盖。

在服务邻居节点选择方面，由于数据帧大小有限，每个缓存节点同一个周期只能为  $N_{max}$  个邻居节点提供服务。在筛选邻居节点的过程中，优先选择  $Z_i$  较小的邻居节点是为了解决节点全覆盖性，因为  $Z_i$  较大的邻居节点被其他缓存节点选择为服务邻居节点的可能性更大。考虑缓存节点与邻居节点链路权重在该邻居节点所有链路权重中的排序位次，是为了判断缓存节点对于该邻居节点的重要性以及链路的稳定性。

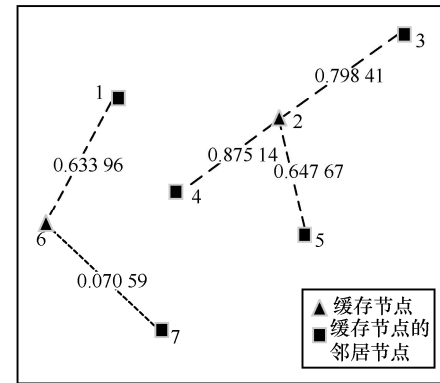
例如，假设有  $\{1,2,3,4,5,6,7\}$  共 7 个节点，其拓扑关系及链路权重如图 2(a)所示，设  $N_{max}=3$ ，则根据所提算法执行缓存节点及其服务邻居节点的筛选，具体过程如下。

- 1) 将连接度为 0 的孤立节点设置为缓存节点，本例中无孤立节点，可跳过此步骤。
- 2) 处理连接度为 1 的节点，即当某节点的邻居节点只有一个时，则考虑将该节点的邻居节点设置为缓存节点。如图 2(a)所示，节点 3 的连接度为 1，其邻居节点有且仅有节点 2，因此将节点 2 设为缓存节点，节点 3 纳入节点 2 的服务邻居节点集合。
- 3) 由于  $N_{max}=3$ ，缓存节点 2 还可以在剩余的邻居节点  $\{1,4,5\}$  中选择 2 个节点作为服务邻居节点。其中，连接状态未定节点的度最小的是节点 5，因此将其纳入节点 2 的服务邻居节点集合。
- 4) 对于节点 1 和节点 4，链路 1—2 在节点 1 的所有链路从大到小的排序中位次为 2，链路 2—4 在节点 4 的链路排序位次为 1，因此，将节点 4 纳入节点 2 的服务邻居节点集合。至此，完成缓存节点 2 的服务邻居节点筛选，即节点  $\{3,4,5\}$ 。
- 5) 剩余的状态未定节点为节点  $\{1,6,7\}$ ，其中节点 6 连接状态未定节点的度最大，因此将其选为缓

存节点。其邻居节点个数为 2，即节点  $\{1,7\}$ ，该值小于  $N_{max}$ ，因此，均纳入节点 6 的服务邻居节点集合。对应结果如图 2(b)所示。



(a) 节点状态未定时的拓扑关系及链路度量值



(b) 节点状态确定后的拓扑关系及链路度量值

图 2 基于负载约束的最小支配集算法案例

## 5 仿真分析

### 5.1 仿真场景及参数设置

通过对市区中心某十字路口早高峰实地考察，仿真场景设定为双向六车道的单个十字路口，每条支路长度为 1 km，总面积为 2 km×2 km，车流量在交通灯作用下呈现波动上升趋势，如图 3 所示。设定仿真软件 SUMO (simulation of urban mobility) 的参数，从而生成交通流数据，即提取以十字路口为圆心、以 500 m 为半径的车辆轨迹，再采用 NS3 (network simulator) 完成算法性能的分析 and 评估，关键仿真参数如表 1 所示。

基站的服务范围为 500 m，车辆之间的通信半径为 50 m。可请求的文件种类总数为 20，后续将评估每车的缓存空间分别为 1、3、5 时的性能，且所有车辆的缓存空间保持一致。文件流行度服从 Zipf 分布，节点请求也服从 Zipf 分布。此外，缓存节点可以发出请求，且可以被自身响应，每个缓存

节点缓存最流行的前  $C$  个文件。C-V2X mode-3 参数设定中, 设子载波带宽为 180 kHz, 噪声功率为  $-113$  dBm, 数据传输率为 1 Mbit/s<sup>[3]</sup>。信道采用 Winner+ B1 城市场景模式<sup>[29]</sup>。为匹配车辆间通信半径设定, 发射功率设为 10 dBm, 并限定数据分组到达率容限为 90%。

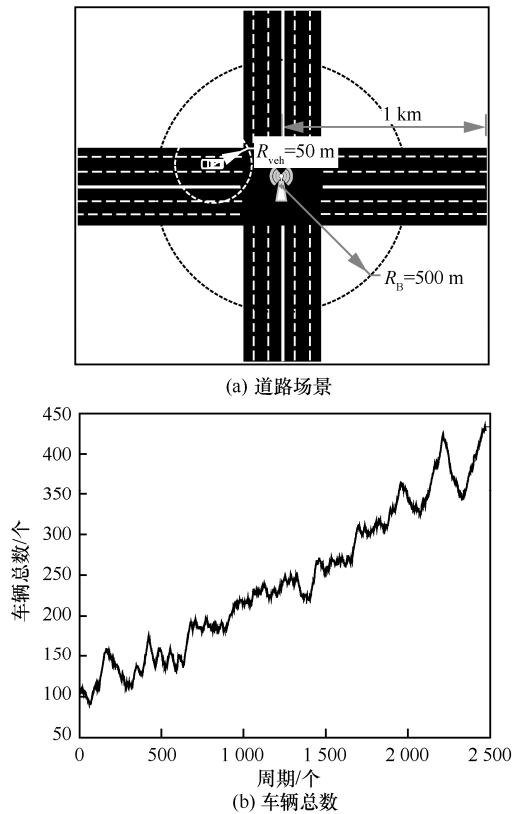


图 3 仿真场景

表 1 关键仿真参数

参数	值
基站覆盖半径/m	500
车辆通信半径/m	50
车辆密度/(辆·(千米·车道) <sup>-1</sup> )	4~15
文件种类个数/个	20
车辆缓存容量/个	5
仿真周期/个	2 473
周期时长/s	5
权重因子 $\rho$	0.5
服务邻居个数上限 $N_{max}$ /个	1,3,5
Zipf 分布参数	0.9
噪声功率/dBm	-113
数据传输速率/(Mbit·s <sup>-1</sup> )	1
信道模型	Winner+ B1
节点发射功率/dBm	10

为了评估算法的性能, 将所提算法(下文简称 NmaxMDS 算法)与穷举法、CCMP 算法<sup>[6]</sup>、k-means 无监督聚类算法(下文简称 k-means 算法)<sup>[30]</sup>、最大连接度算法(下文简称 MaxDegree 算法)<sup>[31]</sup>进行对比。

### 5.2 性能指标

将本文所提算法与穷举法对比, 通过大量随机拓扑的仿真并统计, 以验证筛选出的缓存节点个数和簇平均链路权重均值接近全局最优解的程度。其中, 缓存节点个数的准确性通过给定拓扑下, 本文所提算法与穷举法计算的最优缓存节点个数差值, 即缓存节点个数偏差的分布衡量; 同理, 簇平均链路权重均值也通过与最优结果偏差的均值和标准差衡量。

此外, 将所提算法与 3 种代表性算法的缓存节点选择部分进行比较, 由于这些对比算法并不以节点的全覆盖作为首要优化目标, 因此所比较的性能指标不同于上述与穷举法比较的情况, 而主要衡量缓存节点的利用率以及为基站负载分流的程度, 所采用的具体性能指标如下。

1) 请求应答率。定义为一个周期内, 所有普通节点发出的文件请求中, 能被车辆缓存节点响应的请求占据所有请求的比例。该指标反映的是缓存节点为基站分担负载的程度, 其值越大说明越少的文件请求需要从基站获得响应, 即基站负荷越小。同时, 该指标还能体现缓存分布的合理性和多样性。另外, 为了更清晰体现该指标性能差异, 本节将直观地比较所提算法与对比算法请求应答率差值, 差值越大说明所提算法的优势越明显。

2) 重复应答率。定义为一个周期内, 被缓存源应答的请求中, 同时被多个缓存源应答的请求的比例。该指标体现了缓存分布的冗余度, 当节点的请求同时被多个缓存节点响应时, 将导致响应冗余和带宽浪费。

3) 缓存节点应答次数均值。定义为一个周期内, 每个缓存节点平均的响应次数。其值越大, 说明平均意义下, 缓存节点能够为越多普通节点提供文件共享。该指标体现了缓存节点选择的合理性, 同时也侧面反映了分担基站负载的程度。

### 5.3 NmaxMDS 算法与穷举法的性能比较

穷举法采用遍历尝试的方式。缓存节点个数以 1 为初始值逐步增加, 并且每个缓存节点遍历尝试所有不超过  $N_{max}$  个邻居的邻居节点组合, 直至找到可实现无重叠全覆盖的最少的缓存节点个数, 并且每个缓存节点的服务邻居节点个数不超过  $N_{max}$ 。随着缓存节点个数的增加, 穷举法需要尝试的缓存节点组合以及缓

存节点的服务邻居组合增量巨大，但总节点数过少无法体现算法间性能优劣，因此设节点总数为 4~9，服务邻居个数上限分别为 2 和 3。为减少随机误差，每组参数均重复执行 300 次后计算统计平均值。

图 4 反映了不同的拓扑节点总个数和不同  $N_{max}$  约束下，NmaxMDS 算法和穷举法的缓存节点个数的比较。统计 2 种算法得到缓存节点个数差值出现的次数，再除以总的仿真重复执行次数，即可得到各缓存节点个数偏差占比。从图 4 中可以看出，在不同的节点总数和  $N_{max}$  约束下，缓存节点个数偏差占比趋于稳定，无偏差比例平均值为 95.61%。随着节点个数增多，缓存节点个数偏差占比没有显著差异变化。NmaxMDS 算法最终的缓存节点个数较大程度上与穷举法吻合，有偏差的比例平均值仅为 4.39%，且仅比穷举法的缓存节点个数多一个，但显然 NmaxMDS 算法的时间复杂度远小于穷举法。

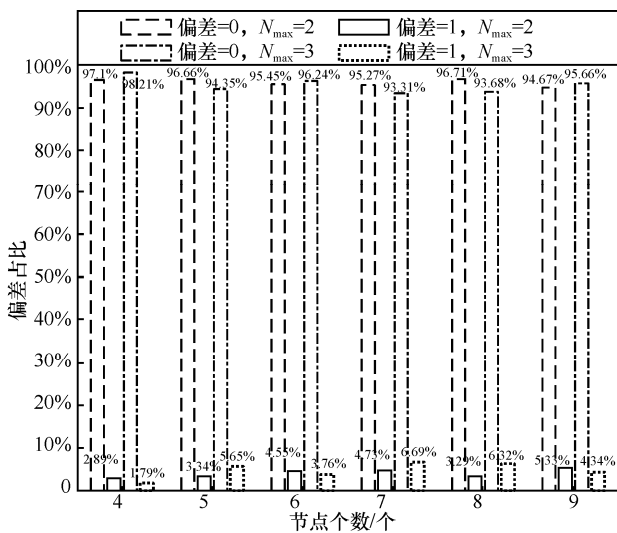


图 4 缓存节点个数比较

在不同节点个数且不同  $N_{max}$  约束下，NmaxMDS 算法和穷举法之间的簇平均链路权重均值偏差如图 5 所示。从图 5 可以看出，NmaxMDS 算法的簇平均链路权重均值比穷举法平均约小 11.88%，平均标准差约为 8.08%。这是由于为了适应 C-V2X 低时延的需求，NmaxMDS 算法采用贪婪思想的思想，以牺牲准确度为代价，提高执行速度，但 NmaxMDS 算法所得结果较接近穷举法的最佳结果。

### 5.4 NmaxMDS 算法与其他缓存节点选择算法的性能比较

图 6 呈现了不同周期下，NmaxMDS 算法和 MaxDegree 算法、CCMP 算法以及 k-means 算法之

间的性能比较，其中设  $C=5$ ， $N_{max}=3$ ，并且为减少随机误差，各算法重复执行 300 次后计算统计平均值。

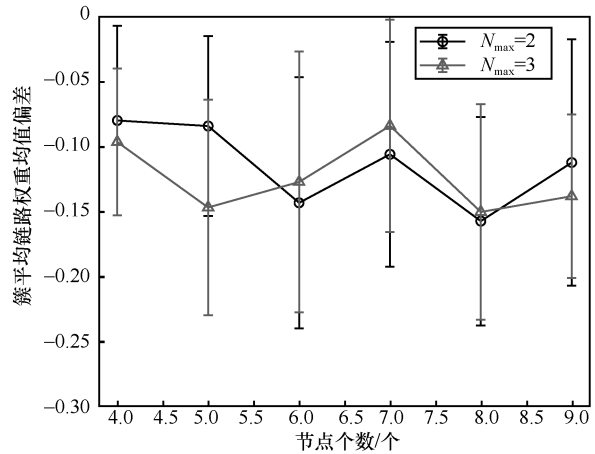


图 5 簇平均链路权重均值偏差

从图 6(a)可以看出，NmaxMDS 算法的请求应答率均值保持在 58.70%，标准差为 1.07%。图 6(a)中的理论值由相同参数设定下，Zipf 分布的 CDF 计算得到，即文件种类数为 20、最流行的前 5 个文件的 CDF 值为 59.32%，NmaxMDS 算法与之偏差约为 0.62%。这是由于非理想信道导致的分组丢失引起部分性能损失。NmaxMDS 算法中每个节点均有且仅有一个缓存节点管辖，且缓存节点存储最流行的前 5 个文件，只要请求这 5 个文件必然获得响应，因此理论上可达请求应答率的理论值。其他 3 种算法随着周期的增多都呈现波动且略微下降的趋势，这是因为在如图 3(b)所示的车流量波动上升作用下，由于不完全覆盖导致更多的节点无法从缓存节点获得所需文件。CCMP 算法和 k-means 算法的请求应答在 0.37~0.46 波动，且 k-means 算法略优于 CCMP 算法。CCMP 算法在进行缓存节点选择时，考虑了对于节点轨迹的预测和节点在热区的逗留时长，未考虑节点的覆盖面。k-means 算法中，设各周期的  $K$  值与 NmaxMDS 算法的缓存节点个数对应一致，虽然该算法也考虑了链路稳定性度量，但对缓存节点的筛选不够完善。MaxDegree 算法的请求应答最低，且在 0.17~0.35 波动，这是因为在进行缓存节点选择时，仅根据节点的连接度进行选择，虽然连接度大的节点可以覆盖更多节点，但是对于节点的服务能力没有进行适度考虑，容易造成大量的覆盖冗余，也导致了覆盖不完全。NmaxMDS 算法综合考虑连接度、负载约束和链路稳定性，提升了缓存节点请求应答，提升网络的服务性能。

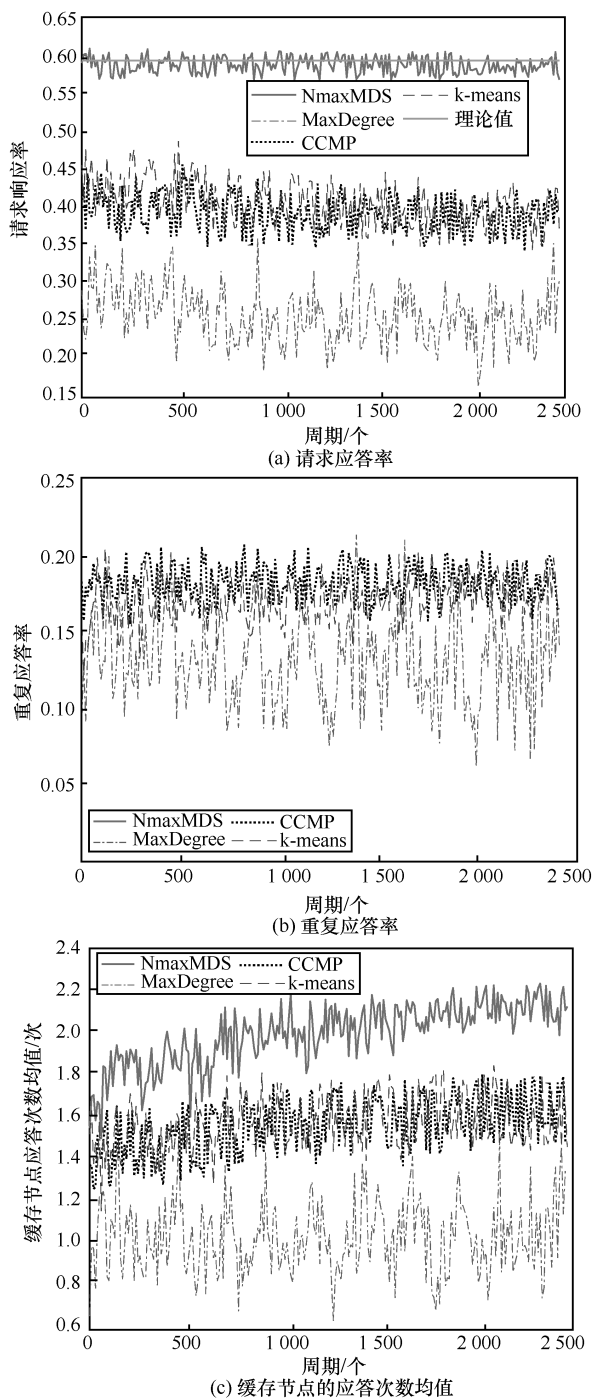


图 6 不同周期下算法性能分析

图 6(b)显示了 4 种算法的重复应答率。3 种对比算法在考虑节点分配成簇的情况下没有考虑节点本身的负载约束，因此一个节点可能会被多个簇头共同覆盖，从而导致了较高的重复应答率。其中，MaxDegree 算法的全覆盖性不佳，导致了其重复应答率比例较低，其波动一方面由拓扑的随机性引起，另一方面由于车流量的起伏变化导致。NmaxMDS 算法在服务邻居节点个数上限约束下，

实现了无重叠覆盖，因此该性能指标恒等于 0，可以降低缓存节点之间的信道竞争，减少冗余响应所耗费的带宽资源。

图 6(c)显示了 4 种算法的缓存源应答次数均值。从图 6(c)可以看出，MaxDegree 算法的缓存源应答次数均值最小，波动区间位于 0.7~1.4 次/缓存节点，CCMP 算法和 k-means 算法变化趋势和值都较为接近，且随着周期变化均呈逐步上升趋势，在仿真周期末期趋于平缓，达到约 1.5 次/缓存节点。算法随着周期变化，缓存应答次数均值明显上升且相对于其他算法的优势逐步增大，仿真周期末期趋于平缓，达到约 2.2 次/缓存节点。MaxDegree 算法只考虑了缓存节点可连接节点的数量，忽视了其服务能力，造成了节点的不完全覆盖，也出现较多的重叠覆盖，从而降低了缓存应答次数均值。CCMP 算法和 k-means 算法均从成簇的角度以逗留时长和链路稳定性度量等为性能进行簇头及其邻居的筛选，故减少了部分冗余响应，但是全覆盖性仍不足。NmaxMDS 算法除了对车辆轨迹进行提前预测外，在分簇阶段也从微观本质上考虑链路的连接实质，提高了缓存节点的应答次数，减轻了基站负担。具体地，在仿真周期初期，车辆密度较小，孤立节点以及连接度低的节点较多，因此有较多的节点无法相互提供文件共享，从而导致缓存应答次数较小，且与对比算法差距不大。随着车辆密度增加，链路资源逐渐增多，更合理的缓存节点及其服务邻居节点的选择逐步突显了 NmaxMDS 算法的优势。而在仿真周期末期车辆密集，缓存节点的服务能力趋于饱和，因此缓存应答次数趋于平缓，但 NmaxMDS 算法仍然保持较大优势。

图 7 反映了 3 种对比算法与 NmaxMDS 算法在不同车辆缓存空间条件下的请求应答率差值，其中设  $N_{max}=3, C=5,10,15$ 。从图 7 可以看出，该差值均大于零，即 NmaxMDS 算法的请求应答率性能在不同车辆缓存空间下均优于对比算法，并且随着车辆缓存空间的增加，优势更加明显。当缓存空间为 5 时，NmaxMDS 算法与 MaxDegree 算法、CCMP 算法和 k-means 算法的差值中位数分别为 0.32、0.18 和 0.17。当缓存空间增加到 15 时，对应的差值中位数增加到 0.49、0.28 和 0.26。因为更大的缓存空间意味着能存储更多文件，普通节点能够以更大概率从缓存节点获取文件。缓存节点的全覆盖性越好，请求应答率就越高，并且每个缓存节

点更大的缓存空间更加突显了 NmaxMDS 算法在全覆盖性方面的优势。

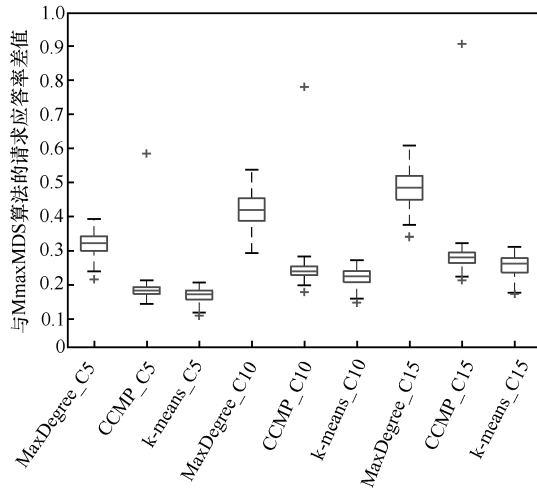


图7 不同车辆缓存空间的请求应答率差值

NmaxMDS 算法与 3 种对比算法在请求应答率差值的 75%分位点和 25%分位点的变化趋势与中位数类似，但在这 2 个分位点之间的间距方面，与 MaxDegree 算法差值的间距最大，k-means 算法的间距次之，CCMP 算法的间距最小。在 NmaxMDS 算法中，每个普通节点都与一个缓存节点连接，保持接近理论值的请求应答率，2 个分位点的间距越大说明算法效果的稳定性受节点密度和分布的影响越大，其性能的波动明显。

图 8 呈现了 NmaxMDS 算法与 3 种对比算法在不同服务邻居节点上限约束下的请求应答率差值，其中， $C=5$ ， $N_{\max}=1,3,5$ 。从图 8 可以看出，请求应答率差值均大于零，说明 NmaxMDS 算法优于各对比算法。随着  $N_{\max}$  增加，NmaxMDS 与 3 种对比算法的请求应答率差值逐渐减小。这是因为每个缓存节点可服务更多的邻居节点时，普通节点可选择的缓存节点更多。对于每个缓存节点的服务邻居节点的准确筛选的要求降低，弱化了各算法之间的性能差距。

随着  $N_{\max}$  增加，NmaxMDS 算法与 MaxDegree 算法的差值最大，中位数分别是 0.37、0.32 和 0.29；与 CCMP 算法的差值次之，中位数分别是 0.31、0.18 和 0.11；与 k-means 算法的差值最小，中位数分别为 0.30、0.17 和 0.09。这是因为 MaxDegree 算法仅从连接度考虑，CCMP 算法从热区逗留时间角度考虑了车辆密集程度，而 k-means 算法从位置和链路的角度综合考虑，所以 k-means 考虑因素更周全，与 NmaxMDS 算法的性能差距更小。

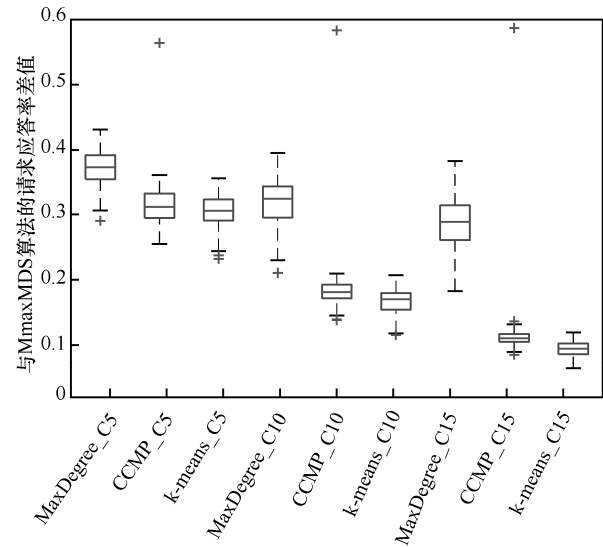


图8 不同服务邻居节点上限的请求应答率差值比较

## 6 结束语

在城市环境下，C-V2X 车辆拓扑快速变化且传输时延要求较高，车辆缓存节点通过 V2V 传输可降低数据传输时延，减轻基站负荷。本文提出 NmaxMDS 算法解决缓存节点及其服务邻居节点的选择问题，定义链路稳定性度量并构建预测权重邻接矩阵；构建最小化缓存节点个数以及最大化簇平均链路权重均值的目标函数；定义 3 种节点状态，在缓存节点负载约束下，求解车辆拓扑的最小支配集，实现无重叠的节点全覆盖。仿真结果表明，所提算法得到的缓存节点个数和簇平均链路权重均值接近穷举法计算的全局最优结果；在请求应答率、重复应答率和缓存源应答次数等方面均优于对比算法，并且重复应答率恒为零，请求应答率可达理论上界，证明了所提算法可有效提升缓存节点利用率，减轻基站负荷。

在后续工作中，考虑适度地在缓存节点之间引入重叠覆盖，并利用缓存节点间的协作关系，进一步提升缓存节点的请求应答。

## 参考文献：

- [1] GUNTHER H, TRAUER O, WOLF L. The potential of collective perception in vehicular ad-hoc networks[C]//International Conference on ITS Telecommunications. Piscataway: IEEE Press, 2015: 1-5.
- [2] CHEN S, HU J, SHI Y, et al. A vision of C-V2X: technologies, field testing and challenges with Chinese development[J]. arXiv Preprint, arXiv:2002.08736, 2020.
- [3] ASLANI R, SABERINIA E, RASTI M. Resource allocation for cellular V2X networks mode-3 with underlay approach in LTE-V stan-

- ard[J]. IEEE Transactions on Vehicular Technology, 2020, 69(8): 8601-8612.
- [4] ZHENG C, PAPPAS N, KOUNTOURIS M. Probabilistic caching in wireless D2D networks: cache hit optimal versus throughput optimal[J]. IEEE Communications Letters, 2016, 21(3): 584-587.
- [5] YI C, HUANG S, CAI J. An incentive mechanism integrating joint power, channel and link management for social-aware D2D content sharing and proactive caching[J]. IEEE Transactions on Mobile Computing, 2017, 17(4): 789-802.
- [6] GIATSOGLOU N, NTONIN K, KARTSAKLI E, et al. D2D-aware device caching in mmWave-cellular networks[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(9): 2025-2037.
- [7] ZHU K, ZHI W, ZHANG L, et al. Social-aware incentivized caching for D2D communications[J]. IEEE Access, 2016, 4: 7585-7593.
- [8] RAY A, DE D. Energy efficient cluster head selection in wireless sensor network[C]//2012 1st International Conference on Recent Advances in Information Technology. Piscataway: IEEE Press, 2012: 306-311.
- [9] QIAO J, ZHANG X. Compressive data gathering based on even clustering for wireless sensor networks[J]. IEEE Access, 2018, 6: 24391-24410.
- [10] SALONI, KAKKAR D. TOPSIS optimized dual-hop routing protocol for homogeneous wireless sensor networks with grid-based clustering[C]//2018 First International Conference on Secure Cyber Computing and Communication. Piscataway: IEEE Press, 2018:776-779.
- [11] GAO W, CAO G, IYENGAR A, et al. Cooperative caching for efficient data access in disruption tolerant networks[J]. IEEE Transactions on Mobile Computing, 2014, 13(3): 611-625.
- [12] RAWASHDEH Z, MAHMUD S. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways[J]. EURASIP Journal on Wireless Communications and Networking, 2012, 1(15): 1-13.
- [13] DAKNOU E, THAALBI M, TABBANE N. Clustering enhancement for VANETs in highway scenarios[C]//2015 International Conference on Communications and Networking. Piscataway: IEEE Press, 2015:1-5.
- [14] FAROOQ W, ALI KHAN M, REHMAN S. A novel real time framework for cluster based multicast communication in vehicular ad hoc networks[J]. International Journal of Distributed Sensor Networks, 2016, 12: 1-18.
- [15] YAO L, CHEN A, DENG J, et al. A cooperative caching scheme based on mobility prediction in vehicular content centric networks[J]. IEEE Transactions on Vehicular Technology, 2017, 6(67): 5435-5444.
- [16] ALSUHLI G H, KHATTAB A, FAHMY Y A. Double-head clustering for resilient VANETs[J]. Wireless Communications and Mobile Computing, 2019, 2019(1): 1-17.
- [17] QI W, LANDFELDT B, SONG Q, et al. Traffic differentiated clustering routing in DSRC and C-V2X hybrid vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(7): 7723-7734.
- [18] CHENG J J, YUAN G Y, ZHOU M C, et al. A connectivity prediction-based dynamic clustering model for VANET in an urban scene[J]. IEEE Internet of Things Journal, 2020, 7(9): 8410-8418.
- [19] SHIN I, KIM M, MUTKA M, et al. MCBT: multi-hop cluster based stable backbone trees for data collection and dissemination in WSNs[J]. Sensors, 2009, 9(8): 6028-6045.
- [20] FATHIAN M, JAFARIAN-MOGHADDAM A R. New clustering algorithms for vehicular ad hoc network in a highway communication environment[J]. Wireless Networks, 2015, 21(8): 2765-2780.
- [21] AHMAD I, NOOR R M, ZABA M R, et al. A cooperative heterogeneous vehicular clustering mechanism for road traffic management[J]. International Journal of Parallel Programming, 2019, 48: 870-889.
- [22] LIU Y, ZHU D, MA W. A novel cooperative caching scheme for content centric mobile ad hoc networks[C]//IEEE Symposium on Computers and Communication. Piscataway: IEEE Press, 2016: 824-829.
- [23] LIU L, XIE D, WANG S, et al. CCN-based cooperative caching in VANET[C]//International Conference on Connected Vehicles and Expo. Piscataway: IEEE Press, 2016: 198-203.
- [24] LI Z, WANG Q, ZOU H. QoE-aware video multicast mechanism in fiber-wireless access networks[J]. IEEE Access, 2019, 7: 123098-123106.
- [25] LIU A, LAU V, CAIRE G. Cache-induced hierarchical cooperation in wireless device-to-device caching networks[J]. IEEE Transactions on Information Theory, 2018, 6(64): 4629-4652.
- [26] LIU Y, ZHI T, XI H, et al. A novel content popularity prediction algorithm based on auto regressive model in information-centric IoT[J]. IEEE Access, 2019, 7: 27255-27564.
- [27] GOLREZAEI N, MANSOURIFARD P, MOLISCH A, et al. Base-station assisted device-to-device communications for high-throughput wireless video networks[J]. IEEE Transactions on Wireless Communications, 2014, 7(13): 3665-3676.
- [28] WIEST J, HOFFKEN M, KRESEL U, et al. Probabilistic trajectory prediction with Gaussian mixture models[C]//2012 IEEE Intelligent Vehicles Symposium. Piscataway: IEEE Press, 2012, DOI: 10.1109/IVS.2012.6232277.
- [29] 3GPP TR 36.885. Study on LTE-based V2X services: Release 14 V14.0.0[S]. (2016-06)[2020-09-23].
- [30] AARON B, TAMIR D, RISHE N, et al. Dynamic incremental k-means clustering[C]//Computational Science and Computational Intelligence. Piscataway: IEEE Press, 2014: 308-313.
- [31] RAMY A, MAJID B M, MEHDI B, et al. Inter-cluster cooperation for wireless D2D caching networks[J]. IEEE Transactions on Wireless Communications, 2018, 7(9): 6108-6121.

## [作者简介]



徐哲鑫 (1985- )，男，福建福州人，博士，福建师范大学副教授，主要研究方向为无线自组织网、车联网。

高楷蒙 (1996- )，女，河南安阳人，福建师范大学硕士生，主要研究方向为无线自组织网、车联网等。

贾文康 (1969- )，男，台湾台北人，博士，福建师范大学教授，主要研究方向为网络协议、路由与转发。

吴怡 (1970- )，女，辽宁葫芦岛人，福建师范大学教授、博士生导师，主要研究方向为无线自组织网、无线视频传输。