



硕士学位论文

(专业学位)

车联网数据传输与缓存策略研究

作者姓名 : 薛拯

导师姓名 : 韩国军

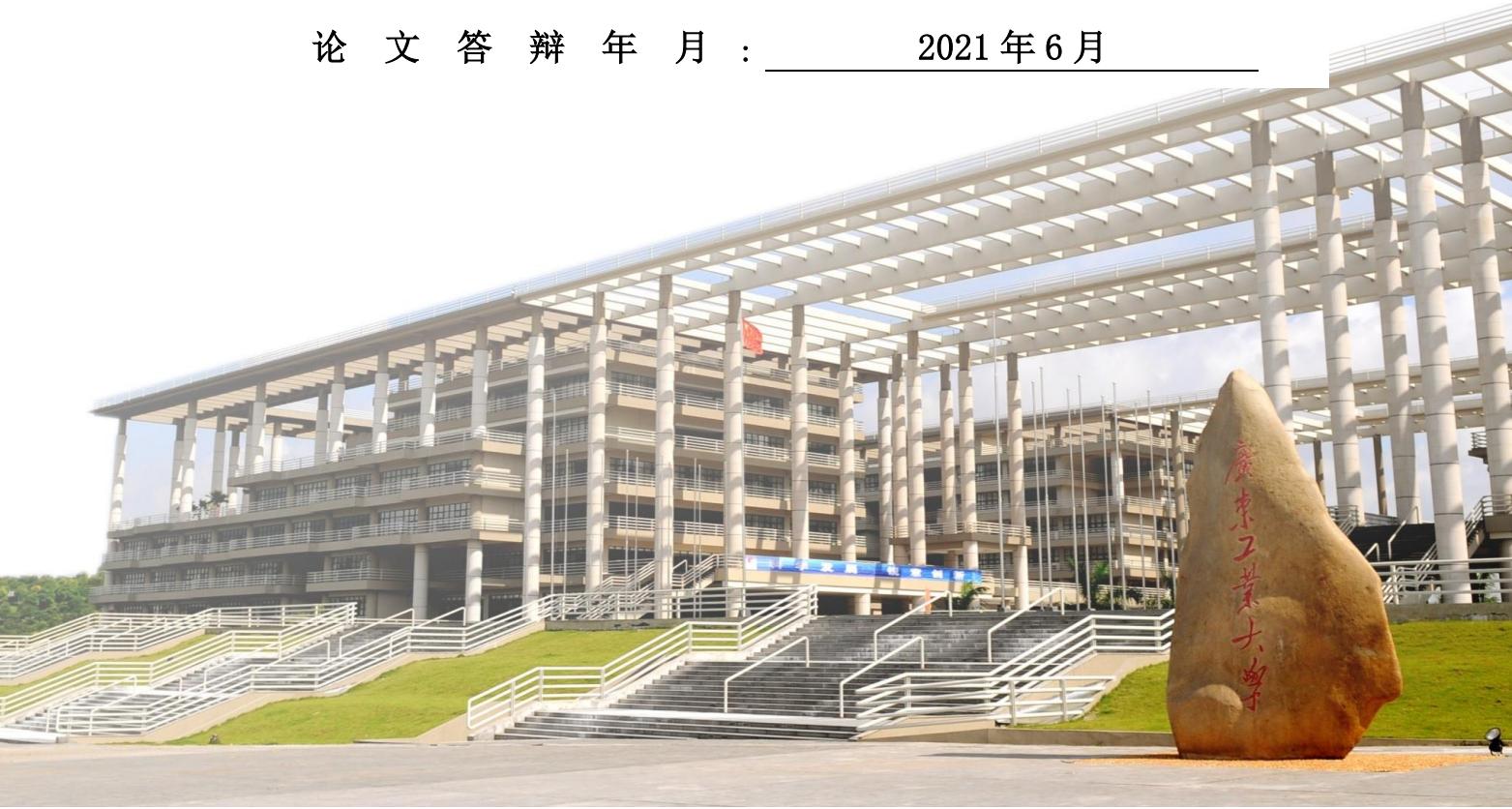
学科(专业)或领域名称 : 电子与通信工程

论文答辩年月 : 2021年6月



薛拯

二零二一年六月



分类号： 学校代码： 11845
UDC： 密级： 学号： 2111803040

广东工业大学硕士学位论文
(工程硕士)

车联网数据传输与缓存策略研究

薛拯

导师姓名（职称）： 韩国军教授

学科（专业）或领域名称： 电子与通信工程

学生所属学院： 信息工程学院

答辩委员会主席： 蔡念

论文答辩日期： 2021年5月29日

A Dissertation Submitted to Guangdong University of Technology
for the Degree of Master
(Master of Engineering)

**Research on Data Transmission and Caching
Strategies in Vehicular Networks**

Candidate: Zheng Xue
Supervisor: Prof. Guojun Han

May 2021
School of Information Engineering
Guangdong University of Technology
Guangzhou, Guangdong, P. R. China, 510006

摘要

车联网作为5G、交通和汽车领域跨界融合且最具潜力的应用，已成为我国战略性新兴产业的重要发展方向。道路安全、交通效率、自动驾驶和信息娱乐等各类车联网应用提出了低时延、高可靠、大带宽、高移动性等新的移动通信需求与挑战。本文主要解决车联网中的三个关键挑战：如何确保与交通相关的数据安全？随着移动互联网数据流量激增，如何保证车联网数据安全、可靠、稳定传输？如何利用有限的存储空间设计高效的缓存策略？

本文主要聚焦解决三个挑战中的关键问题：第一，基于机器学习的入侵检测技术逐步应用到智能车联网上，但仅仅是对异常消息记录的检测识别并不一定能识别出真正的恶意车辆。第二，车联网信息交互带来海量数据，分簇能够改善交通信息收集与分发效率，但簇内数据传输优化的研究较少。此外，热门内容多次重复下载会增加回程链路负载，缓存技术通过将热门内容缓存在距离用户终端更近的路边单元（Road Side Unit, RSU）和车辆上，能够有效降低获取时延和传输成本。但现有的车联网缓存方案大多考虑RSU和基站（Base Station, BS）缓存，没有充分利用移动的车辆本身的缓存空间。本文针对以上问题，通过不同场景下的建模、仿真分析，深入开展车联网数据传输与缓存策略研究。主要贡献如下：

(1) 基于仿真数据搭建并训练好入侵检测系统。针对RSU部署少的场景，提出基于模糊逻辑的智能检测系统部署方案，基于设定阈值能有效识别出恶意车辆。

(2) 基于复杂网络理论，针对城市交通数据业务收集与分发场景，为了减少整个网络信息传输负载，提出基于广义距离的分簇算法、基于模糊逻辑的簇头选择算法和簇内数据传输优化模型的车联网综合通信优化方案，能有效提高簇内网络吞吐量并降低传输时延。

(3) 拓展现有车联网单层缓存方案，推导平均内容传输时延和平均内容传输成本。充分利用车辆及RSU的缓存容量，建立最小化平均传输时延和成本的优化问题。提出交替动态规划搜索算法和低复杂度合作贪婪算法求解优化问题。同时，基于MK5平台开发内容传输系统并实现系统原型。经过仿真和实测数据验证，所提的缓存策略能够有效降低内容传输时延和成本。

关键词：车联网；入侵检测；模糊逻辑；缓存策略

Abstract

As one of the most promising cross-disciplinary applications involving 5G, transportation and automobile, the Internet of Vehicles has become an important development direction of strategic emerging industries in China. Various Internet of Vehicles applications include road safety, traffic efficiency, autonomous driving, infotainment, etc. These applications put forward new requirements for mobile communications such as low latency, high reliability, large bandwidth, and high mobility. In this thesis, we aim to address three key challenges of Internet of Vehicles: (1) How to guarantee data security which is crucial for safety? (2) With the rapid increase of mobile Internet traffic, how to ensure safety, reliability, and stability of data transmission? (3) How to design efficient caching strategies using limited storage spaces?

This thesis mainly focuses to address three issues: First of all, intrusion detection technologies based on machine learning has been gradually applied to the intelligent vehicle networks. However, the actual detection performances of the intrusion detection system implemented on vehicles has not been tested. Secondly, vehicle networking for information interaction generates massive amounts of data. Clustering can improve the efficiency of traffic information collection and distribution, but there are few studies on data transmission optimization within clusters. In addition, repeated downloading of popular contents will increase backhaul loads. Caching technologies can effectively reduce transmission delays and costs by caching popular contents on road side units (RSUs) and vehicles close to the requester. Most of the existing caching schemes consider RSU and base station (BS) caching. Without taking cache capacity on vehicles into consideration. To address the aforementioned issues, this thesis conducts in-depth research on data transmission and caching strategy of Internet of Vehicles through modeling and simulation analysis in different scenarios. The main contributions are as follows:

(1) We build and train an intrusion detection system based on simulation data. For scenarios where RSUs are less deployed, we propose a fuzzy logic-based intelligent detection system deployment scheme is proposed, which can effectively identify malicious vehicles

based on the set threshold values.

(2) Based on complex network theory, we propose a clustering algorithm based on generalized distance, a cluster head selection algorithm based on fuzzy logic, and an optimization model of intra-cluster data transmission. Our work aim to reduce the information transmission load of the entire network for urban traffic data service collection and distribution scenarios. The comprehensive communication scheme can effectively improve the network throughput in clusters and reduce transmission delays.

(3) We expand an existing single-layer caching scheme for the Internet of Vehicles, and derive the average content delivery delay and average content delivery cost. We then propose a caching scheme so that the caching capacity at both vehicles and RSUs can be fully utilized to minimize the overall transmission delay and cost. We further propose an alternate dynamic programming search (ADPS)-based algorithm and a low complexity cooperative greedy algorithm to solve the optimization problem in this scheme. In addition to numerical simulation, we develop an infotainment content transmission system on the MK5 platform and implement the prototype model. Results show that the proposed caching strategy can effectively reduce content transmission delays and costs.

Keywords: Internet of Vehicles; intrusion detection system; fuzzy logic; caching strategy

目录

摘要	I
Abstract	III
目录	V
Contents.....	VIII
第一章 绪论	1
1.1 课题研究背景及研究意义	1
1.2 国内外相关研究现状	2
1.2.1 车联网入侵检测系统研究	2
1.2.2 车联网簇头选择与数据传输策略研究	3
1.2.3 车联网内容缓存策略研究	4
1.3 论文主要内容及结构安排	5
第二章 技术基础概述	7
2.1 车联网技术标准	7
2.1.1 DSRC 技术	7
2.1.2 C-V2X 技术	7
2.2 模糊逻辑	8
2.2.1 模糊化和模糊规则	8
2.2.2 模糊逻辑推理流程	8
2.3 复杂网络理论	9
2.4 MK5 硬件平台介绍	10
2.5 本章小结	11
第三章 面向数据安全的智能检测系统与部署技术研究	12
3.1 数据获取	12
3.1.1 车辆移动模型	12
3.1.2 仿真环境	13
3.1.3 车辆恶意行为	13

3.1.4 特征数据集	14
3.2 智能检测系统	14
3.3 基于模糊逻辑的智能检测系统部署方案	15
3.4 性能分析	18
3.5 本章小结	20
第四章 车联网分簇及簇内数据传输策略研究.....	21
4.1 系统模型	21
4.1.1 数据集	21
4.1.2 无向图模型	22
4.1.3 复杂网络验证	22
4.2 分簇算法	25
4.3 簇头选择算法	25
4.4 簇内数据传输优化模型	28
4.5 仿真及性能分析	30
4.6 本章小结	32
第五章 双层车联网架构中的缓存策略研究.....	33
5.1 缓存系统模型	33
5.1.1 网络模型	33
5.1.2 车辆移动模型	34
5.1.3 内容分布与获取模型	34
5.2 优化问题建模	35
5.2.1 平均内容传输时延	35
5.2.2 平均内容传输成本	37
5.2.3 平均内容传输命中率	38
5.2.4 缓存优化问题建模	38
5.3 交替动态规划搜索算法	39
5.3.1 动态规划算法	39
5.3.2 交替动态规划搜索算法	42

5.4 低复杂度合作贪婪算法	43
5.5 数值仿真结果及分析	46
5.5.1 合作贪婪算法的有效性	47
5.5.2 网络参设设置对缓存方案性能的影响	49
5.6 测试床验证	53
5.6.1 系统原型实现	53
5.6.2 传输速率测试与缓存方案验证	55
5.7 本章小结	57
总结与展望	58
参考文献	60
攻读学位期间取得与学位论文相关的成果	68
学位论文独创性声明	69
致 谢	70

Contents

Abstract(In Chinese)	I
Abstract(In English).....	III
Contents(In Chinese).....	V
Contents(In English)	VIII
Chapter 1 Introduction	1
1.1 Background and significance of research.....	1
1.2 Domestic and international research situation.....	2
1.2.1 Research on Intrusion Detection System for Internet of Vehicles.....	2
1.2.2 Research on cluster head selection and data transmission strategy in VANETs	3
1.2.3 Research on content caching strategies in Internet of Vehicles.....	4
1.3 Contents and structure of this thesis.....	5
Chapter 2 Overview of the technical foundation.....	7
2.1 Technical Standards for Internet of Vehicles	7
2.1.1 DSRC technology.....	7
2.1.1 DSRC technology.....	7
2.1.2 C-V2X technology	7
2.2 Fuzzy logic	8
2.2.1 Fuzzification and fuzzy rules	8
2.2.2 The reasoning process of fuzzy logic technology	8
2.3 Complex network theory	9
2.4 Introduction of MK5 hardware platform.....	10
2.5 Chapter summary	11
Chapter 3 Research on Intelligent Detection System and Deployment Technology for Data Security.....	12
3.1 Data collection.....	12

3.1.1 Vehicle mobility model	12
3.1.2 Simulation scenario	13
3.1.3 Malicious behavior of vehicles.....	13
3.1.4 Feature data set.....	14
3.2 Intelligent detection system.....	14
3.3 The deployment strategy of intelligent detection system based on fuzzy logic.....	15
3.4 Performance analysis.....	18
3.5 Chapter summary	20
Chapter 4 Research on Clustering and Data Transmission Strategies in VANETs.....	21
4.1 System model	21
4.1.1 Data set.....	21
4.1.2 Undirected graph model	22
4.1.3 Complex network verification.....	22
4.2 Clustering algorithm.....	25
4.3 Cluster head selection algorithm	25
4.4 Data transmission optimization model within a cluster	28
4.5 Simulation and performance analysis.....	30
4.6 Chapter summary	32
Chapter 5 Research on caching strategies in the two-layer vehicular networks architecture	33
5.1 System model	33
5.1.1 Network model	33
5.1.2 Vehicle mobility model	34
5.1.3 Content distribution and retrieve process model.....	34
5.2 Problem formulation.....	35
5.2.1 Average content delivery delay	35
5.2.2 Average content delivery cost.....	37
5.2.3 Average content delivery hit ratio.....	38

5.2.4 Problem formulation.....	38
5.3 Alternate dynamic programming search algorithm	39
5.3.1 Dynamic programming algorithm	39
5.3.2 Alternate dynamic programming search algorithm	42
5.4 Low-complexity cooperative greedy algorithm	43
5.5 Numerical simulation results and analysis	46
5.5.1 The effectiveness of cooperative greedy algorithm.....	47
5.5.2 The Impact of Network Parameters on Caching Schemes	49
5.6 Test bed evaluation	53
5.6.1 System prototype implementation.....	53
5.6.2 Transmission rate test and caching scheme verification	55
5.7 Chapter summary	57
Conclusion and prospect.....	58
References	60
Publication and patents during study.....	68
Statement of original authorship and copyright licensing declaration	69
Acknowlegements	70

第一章 绪论

1.1 课题研究背景及研究意义

近年来，第五代（The fifth Generation, 5G）移动通信网络的快速发展建设，推动了新型信息通信技术在智能网联汽车领域的应用。5G 的三大特性可支持高精度地图数据以及车辆、行驶环境数据的传输，能够支持实现汽车自主的感知智能驾驶，而 5G/蜂窝车联网（Cellular Vehicle to Everything, C-V2X）为自动驾驶落地铺平道路，全面助推了车联网产业商用加速。目前，上汽集团已推出全球首款可支持 5G V2X 应用的 5G 智能电动汽车 MARVEL R，获得中国 5G 终端电信设备进网许可证和无线电管理委员会（State Radio Regulation Committee, SRRC）车载车规 5G/C-V2X 终端产品认证，为 5G C-V2X 场景的落地提供了解决方案。

在车联网通信技术标准方面，主流技术有电气和电子工程师协会（Institute of Electrical and Electronics Engineers, IEEE）基于 IEEE 802.11p 标准制定的专用短程通信技术（Dedicated Short Range Communication, DSRC）和第三代合作伙伴计划（3rd Generation Partnership Project, 3GPP）通过拓展 LTE 系统网络制定的 C-V2X，包含 LTE-V2X、5G-V2X 及后续演进，支持 LTE-V2X 向 5G-V2X 演进。DSRC 已经存在多年，发展过程并不顺利。从技术角度看，根据 5G 汽车联盟（5G Automotive Association, 5GAA）的测试报告，C-V2X 在很多方面的性能优于 DSRC，支持更好的非视距性能、在拓展的通信范围内增强了可靠性、更高的抗干扰能力和更好的拥塞控制等^[1]。从成本和标准化的角度看，C-V2X 基于蜂窝网络，复用现有的 4G、5G 网络，覆盖范围广，部署成本较低，而 DSRC 技术组网需要新建大量路侧单元（Road Side Unit, RSU），部署成本高。C-V2X 在全球具备更佳的兼容性且演进路线清晰。从政策支持角度看，美国之前一直支持 DSRC。2020 年 11 月，美国联邦通信委员会针对 5.9GHz 频段重新分配进行了投票，并划分 30MHz 给 C-V2X 专用，表明转向 C-V2X^[2]。中国拥有全球最大的 LTE 网络，政府层面非常鼓励包括 C-V2X 在内的车联网技术的发展。目前已经建成多个国家级/省级车联网先导区，逐渐丰富车联网出行应用场景和模式^[3]。中国拥有较为完整的生态产业链，是 V2X 产业化繁荣发展的基础。目前 C-V2X 还在演进和迭代中，未来发展路径清晰，前景可期。

基于车联网通信技术，车载终端也将接入无线网络，智能网联汽车之间通过信息交互提高交通效率，大量车辆之间的数据共享对车联网系统的安全性和可靠性带来巨大挑战。在数据传输共享的过程中，如何在大规模快速移动的车载网络中保证数据高效可靠的传输有待解决。与此同时，人工智能技术的应用带来更加智能的乘驾体验，例如：丰富的车载娱乐服务（比如：虚拟现实、在线游戏等），这也将激发移动数据流量的爆炸式增长。据预测，到 2023 年，5G 连接将比 4G 连接产生近 3 倍的数据流量^[4]。虽然有各种新技术用来提升无线空口传输速率，但网络容量的提升会给基站和核心网之间的回程链路带来非常大的负载压力，热门的内容会通过回程链路重复冗余传输^[5]。缓存技术通过将受欢迎的内容存储到边缘设备比如 RSU 中，车辆用户获取内容的时延更低，采取何种缓存方案缓存内容也是值得研究的课题。因此，本论文对车联网数据安全检测、数据传输优化与内容缓存策略进行了研究，具有一定的理论和现实意义。

1.2 国内外相关研究现状

1.2.1 车联网入侵检测系统研究

随着自动驾驶技术趋于成熟，车联网与自动驾驶技术结合成为必然，智能网联汽车互相连接交换数据提高交通效率的同时，也带来明显的安全风险。当黑客入侵车辆变为恶意车辆后，正常的数据无法进行转发，这会影响道路安全甚至危及生命。入侵检测系统能够主动检测并识别出恶意车辆，保障车辆驾驶安全。

许多学者针对车联网入侵检测系统进行了研究。Alheeti K M A 等人在文献^[6,7]中对自动驾驶汽车的安全性进行一系列研究，提出一种智能入侵检测体系结构，针对灰洞攻击、冲击攻击、恶意攻击、黑洞攻击提出基于神经网络、支持向量机、线型判别法来检测攻击或恶意行为，提高检测精度。Zaidi K 等人在文献[8]提出基于假设检验的虚假信息攻击检测，通过减少消息传输控制网络信息拥塞，避免广播风暴。Sharma S 等人在文献[9]对车联网入侵检测系统进行了详细的分类介绍及技术分析比较，并提出充当诱饵的蜜罐入侵检测体系，充当诱饵的蜜罐节点用来收集恶意节点的攻击记录并更新数据库规则集，对现有攻击及未知的新攻击检测有更好的策略应对。文献[10]提出一种协作入侵检测系统，利用多判别器生成对抗网络，让多个分布式软件定义网络（Software Defined Networking, SDN）控制器共同训练整个网络的检测模型，与集中式检测相比，能够减少系统计算和通信负载。文献[11]提出基于隐式广义混合过渡分布模型用于快速

过滤来自相邻车辆的消息，减少入侵检测系统的开销和检测时间。文献[12]在不引入太多额外计算和存储开销的情况下，基于机器学习提升了车联网入侵检测模型的分类能力。文献[13]提出基于博弈论的混合模式入侵检测方法，提高了异常检测的准确性。上述研究主要关注如何更好训练检测模型，提高检测精度，而没有考虑入侵检测系统在实际过程中的部署问题，检测精度也是对消息记录的检测精度，而根据这些消息记录并不一定能识别出真正的恶意车辆。在实际的车辆运动过程中，入侵检测系统能否准确检测出恶意行为的车辆并隔离恶意车辆也是值得关注的。

1.2.2 车联网簇头选择与数据传输策略研究

智能网联汽车互相连接组成车载自组织网络(Vehicular Ad hoc NETwork, VANET)，车辆的快速移动对车辆之间建立畅通的通信和稳定的数据传输以支持流媒体应用充满挑战。分簇能够有效改善 VANET 中的数据传输问题。分簇算法的原理是根据一些规则集将移动节点关联为一个簇，并选择一个称为簇头(Cluster Head, CH)的节点，以便簇和网络的其余部分之间进行调解^[14]。簇头负责融合簇内车辆的数据，而其他节点(非簇头)将自己的数据传输至各自的簇头。由于簇头承担了数据融合的任务，负责与外界通信，簇内车辆成员的变动导致网络拓扑动态变化，加速通信链路的断裂，簇结构的稳定性及簇头的选择直接影响数据传输性能^[15]。

分簇及簇头选举用于稳定网络结构及改善车联网通信性能^[16-22]。文献[16]以减少车辆 MAC 层竞争为目的综合考虑速度、密度、信道因素来选取簇头从而提高网络吞吐量等性能。文献[18]提出分布式聚类算法和基于进化博弈论的簇头选举算法，能够减轻基站负载并提高用户体验质量。文献[20]针对荒漠环境提出一种分簇路由算法，建立可靠路由，提高簇的结构稳定性并降低路由开销。文献[21]提出一种基于移动性的稳定聚类方法，利用车辆速度，位置和方向以及链路质量等度量用于簇头选择，该算法在城市和高速公路场景具有可靠的稳定性。文献[22]提出一种计算节点稳定性值的模型，根据距离、速度和概率参数的差异来计算节点稳定性，选取的簇头具有很高的价值稳定性。

文献[23]提出一种基于分布式集群的传输调度，建立无竞争传输调度机制，用来解决最大最小流量分配问题，改善集群内车辆数据传输。文献[24]提出基于集中式聚类的混合车载网络框架，实现聚类和协调消息传输，以减少开销并改善安全数据传输。文献[25]提出一种新的 VANET 数据传播策略，选择最大效用的节点用作数据中继，所提出的方法在数据传播延迟方面表现更好。

分簇及选择簇头虽然改善了簇内车辆通信状况，但簇内通信往往是单跳的方式。当某一车辆脱离与簇头的通信范围时，可能出现通信链路断裂，重连质量不稳定的情况，容易导致数据分组丢失。特别是在大型城市存在大量业务数据分发业务时，容易造成冲突和分组次序混乱^[26]。因此对簇内数据传输优化是非常必要的。

1.2.3 车联网内容缓存策略研究

随着 5G V2X 智能电动汽车的落地，车载系统也会更加智能，用户对网联汽车的需求不仅仅满足于智能安全驾驶功能，车内还需要丰富的娱乐服务。当大量用户同时对热门内容请求下载时，就会给核心网和远程服务器之间的回程链路造成负载压力。为了避免用户对车载娱乐服务的大量需求导致热门内容的数据流量激增，RSU 或者用户终端将热门内容提前缓存来避免受欢迎内容重复从核心网下载。边缘缓存的内容离用户更近，请求用户获取内容的时延更低。对于网络运营商而言，除了满足用户获取内容的时延，用户从不同节点获取内容的传输成本也需要控制，通过车与车，车与 RSU 之间的传输成本远远低于基站的传输成本，将基站的业务流量卸载到边缘服务器或者车辆节点这些链路能够进一步降低内容传输成本。

许多学者对移动终端设备（Device-to-Device，D2D）缓存策略进行了大量研究^[27-38]。文献[28]提出移动感知缓存策略最小化网络时延。文献[31]采用不同用户接触间隔时间模型模拟用户移动模式，提出移动感知缓存放置策略最大化数据卸载率。在这些工作中，广为采用的间隔接触模型用来描述缓存节点的移动模式仅适用于移动用户，对于车辆移动采用该模型是明显不合适的。对于车载网络缓存，根据缓存内容的位置可以分为缓存在 RSU 层、车辆层、跨层缓存。大部分研究者集中在 RSU 缓存。文献[39]提出集中式 RSU 缓存算法最小化车载单元（OnBoard Unit, OBU）下载一个文件的平均时间。文献[40]研究 RSU 缓存场景，多个内容提供商利用 RSU 的存储空间提高自身内容的数据传输效率。文献[41]提出模型来决定请求车辆是否获取内容副本，在哪里获取内容副本，提出基于交叉熵的动态内容缓存方案，提高内容命中率。文献[42]提出分层终端边缘框架，通过计算负载和内容缓存来最小化网络开销。文献[43]提出 RSU 合作主动缓存方案最小化通信时延并提高 QoS。文献[44]提出基于车载云网络架构的缓存放置方案最小化内容获取的平均时延。文献[45]提出由长短期记忆神经网络支持的 Q 学习主动 RSU 缓存策略，旨在提高车载网络中非安全相关服务的 QoS。文献[46]提出联合任务缓存、D2D 任务卸载和边缘计算平台任务卸载的框架，其优化目标为最小化

总时延和总能耗。

与 RSU 缓存相比，缓存在车辆层通过车车（Vehicle-to-Vehicle，V2V）通信能够减少传输时延和带宽竞争。文献[47]以最小化缓存时延和最大化能量效率为目标，提出基于深度确定性策略梯度算法的边缘缓存策略。文献[48]提出一种面向边缘计算的协作式车联网数据分发问题，在保证数据请求的截止期限下，最大程度减少从基站流量的开销。文献[49]提出车载缓存框架，通过整合分布式存储和基于单跳 V2V 链接的缓存内容中继，设计动态分布式存储中继机制，确保缓存的内容在请求用户感兴趣时间内以高概率存在。文献[50]针对车辆内容中心网络提出一种基于社交属性和移动预测的协作缓存方案，与现有技术方案相比具有更高的缓存命中率和更低的内容访问延迟。文献[51]提出最优保留感知缓存方案来最小化网络成本。文献[52]通过二位马尔科夫过程对缓存车辆和移动用户之间的交互进行建模，并通过优化网络能效提出在线车辆缓存方案。文献[53]针对蜂窝车联网中的内容共享提出基于动态定价的激励机制，提高内容交付成功率和能源效率。文献[54]提出安全内容缓存放置与传输算法最小化所有车辆用户的最大脆弱性。

少数研究集中在跨层合作缓存方案，比如 RSU 和基站（Base Station，BS）之间的异构车载网络（Heterogeneous Vehicular Networks，H-VNets）合作缓存^[55]、RSU、BS 和车辆三层合作缓存等。文献[56]设计了一种基于 BS、RSU 和车辆合作的边缘缓存框架，提出联合车辆调度和带宽分配方案最小化满足内容传输时延约束时的内容获取成本。

现有的研究大多数集中在车辆或者 RSU 缓存的方案，而跨层合作缓存只是停留在框架，即不同层之间能够合作，该层没有缓存请求内容则向其他层继续请求内容，是一种请求机制，并没有对不同层具体缓存的内容进行研究。因此，如何充分利用不同层有限的缓存空间来最大化缓存增益对缓解内容冗余下载是十分必要的。

1.3 论文主要内容及结构安排

本文针对车联网中数据安全、传输可靠以及热门内容冗余下载问题，分别研究了车联网面向数据安全的智能检测系统与部署技术、簇内数据传输优化策略和双层缓存策略。具体是，针对网络入侵检测数据集过时不适用于车联网，通过仿真得到数据集用于训练检测系统并且考虑检测系统的部署问题，最终能否识别出异常行为的车辆进行研究。针对大型城市大量业务数据分发业务，基于复杂网络理论对车联网进行建模，完成分簇、簇头选择以及簇内数据传输优化工作。针对数据流量激增，内容下载冗余，提

出车联网双层缓存模型，充分利用车辆和 RSU 缓存空间最小化内容传输平均时延和成本总和，提出两种算法仿真求解，搭建硬件测试平台实现系统模型并对缓存方案进行验证。本文的章节安排如下：

第一章：绪论。首先简要介绍论文研究背景，分别对车联网入侵检测系统、分簇与数据传输和缓存策略研究现状及存在的问题进行了总结。最后介绍研究内容及章节安排。

第二章：技术基础概述。主要介绍两个主流的车联网通信技术标准、本文研究所需的技术理论和系统原型搭建所使用的硬件平台。

第三章：面向数据安全的智能检测系统与部署技术研究。介绍车联网入侵检测数据集获取的仿真场景、仿真工具、仿真流程、数据格式、数据预处理，智能检测系统流程及训练结果，随后研究在不同部署方案下的检测结果。仿真结果表明所提的部署方案对检测效果有一定的提升。

第四章：车联网分簇及簇内数据传输策略研究。从复杂网络理论角度对真实出租车数据集进行分析，验证符合复杂网络的某些特性，建立车联网模型，完成分簇、簇头选择以及簇内数据传输优化的综合通信方案。数值仿真结果表明，所提的方案能够改善簇内车辆通信状况。

第五章：双层车联网架构中的缓存策略研究。考虑网络模型、车辆移动模型和内容分布与获取流程模型，推导内容获取平均时延和平均成本。缓存优化方案旨在充分利用车辆和 RSU 有限的缓存空间来最小化网络平均时延和成本。为了求解该问题，提出交替动态规划搜索算法和低复杂度的合作贪婪算法。仿真结果表明，所提缓存方案在时延、成本上远低于其他方案，同时内容命中率提高。随后，用硬件搭建系统模型原型，开发娱乐内容传输系统，并对缓存方案进行验证。基于真实测试数据，所提的缓存方案仍能实现优于其他方案的效果。

第二章 技术基础概述

本章对车联网现有标准体系、论文用到的相关技术和理论基础以及支持 DSRC 通信的 MK5 硬件平台做介绍。

2.1 车联网技术标准

目前，国际上有两个车联网无线通信技术的标准路线，由 IEEE 主导标准化的 DSRC 技术和由 3GPP 主导标准化的 C-V2X 技术路线。

2.1.1 DSRC 技术

DSRC 技术是一组协议栈的集合，包括 IEEE 802.11p，IEEE 1609.1/.2.3.4 协议家族和应用层由美国汽车工程师协会 SAE 制定的 SAE J2735 消息集，并由美国联邦通信委员会划分 5.9GHz 频带中的 75MHz 带宽作为 DSRC 专属的交通安全和车辆通信频谱，频谱分为一个控制信道和六个服务信道^[57]，支持运动环境下的车车和车路间的直通通信，但存在连续覆盖性差、车辆密集时通信时延大和可靠性低等缺点。DSRC 技术已在北美、欧洲进行了十余年的研究和测试评估，由于其通信性能、部署成本原因，商用进展不理想。

2.1.2 C-V2X 技术

C-V2X 技术包括 LTE-V2X 和 NR-V2X，是将蜂窝通信技术和直通通信技术有机结合的车联网无线通信技术，支持车车、车路、车人以及车网等各类应用。LTE-V2X 技术最早由陈山枝教授及大唐团队提出，并积极推动 C-V2X 国际标准化及其演进^[58]。C-V2X 的系统架构原理包括蜂窝通信方式、直通通信方式及其对应的 Uu 接口、PC5 接口。Uu 接口主要用来实现低时延的业务，提高非视距条件下的可靠性，只能工作在蜂窝网络覆盖内。PC5 接口进一步分为两种工作模式，模式 4 和模式 3，模式 4 是用户终端自主资源选择，V2V 数据调度和接口的管理基于车车间的分布式算法实现。模式 3 是基站资源调度分配，通过控制信令接口 Uu 实现 V2V 数据的调度和接口管理。基于 Uu 接口的协议栈与传统的 5G 协议栈一样，而基于 PC5 接口的协议栈，3GPP 定义了其中的物理层和媒体接入层，完全重用 DSRC 的高层协议规范，用户从 DSRC 迁移到 C-V2X 的成本会相对较低。LTE-V2X 用于基本道路安全业务，而 NR-V2X 主要面向增强的 V2X 业务，两者互相补充。

2.2 模糊逻辑

模糊逻辑是一种使用非数字语言，模仿人类决策来处理不确定信息的一种方法。不同于布尔逻辑，计算机可以直接根据精确的描述数值与阈值进行判断和决策，而人类进行判断和决策时往往根据一个模糊不确定的相对信息来进行决策。生活中的实际问题大多都是模糊的，没有明确的分界线，比如个人的财富、身高，多少算富有？多高算高？等等，为了让计算机理解这种描述语言并帮助作出决策，模糊逻辑应运而生。

2.2.1 模糊化和模糊规则

为了描述带有模糊性质的变量，比如“速度”，可以使用“快速”、“中速”、“慢速”这样的模糊集合来表示速度的大小（模糊等级），某一速度并不完全归属于其中一个类，而是以隶属度来衡量，不同的模糊等级与速度对应不同的隶属度，其关系即为隶属度函数。比如当输入某一速度 15m/s 时，根据隶属度函数可以将输入速度描述为{快速：0，中速：0.2，慢速：1}。将输入数值转化为模糊等级隶属度的过程称为模糊化。

完成模糊化后，需要通过模糊规则进行模糊推理。常使用 IF-THEN 规则描述，输入的不同变量最后需要通过模糊规则来进行运算，通常采用最大最小准则，最后的输出值也是模糊集合表示的变量，但这个变量并不能解决实际问题，此时需要通过去模糊化方法得到模糊逻辑输出值。

2.2.2 模糊逻辑推理流程

本节通过一个例子来详细说明模糊逻辑的推理流程。一个完整的模糊逻辑推理流

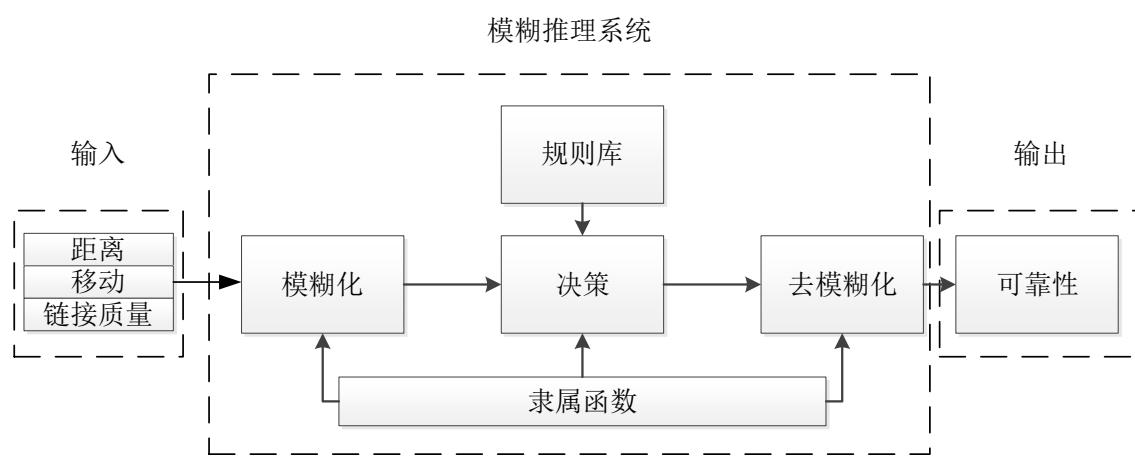


图 2-1 模糊逻辑系统

Fig.2-1 Fuzzy logic system

程如图 2-1 所示,主要包括模糊化、规则库、决策系统和去模糊化。输入一般是多个因素,而模糊逻辑输出值常代表概率或者可靠性,是精确的值。模糊规则和推演过程示例如图 2-2 所示。输入变量模糊化后,按照模糊规则进行推演,为了整合所有规则推演的结果,采用 Min-Max 方法,按每条规则推演时采用 Min 方法,所有规则推演的结果进行整合时采用 Max 方法,可以看到,最后得到的结果仍然是一个模糊化的变量。为了得到精确的值,需要进行去模糊化操作,采用重心法进行去模糊化结果较为准确,将模糊化的结果表示在输出隶属函数上,并求解表示图形的重心,其横坐标即为去模糊化后的输出值。

	Distance	Mobility	Link quality	Rank
Rule 1	Large	Slow	Good	Perfect
Rule 2	Large	Slow	Medium	Good
Rule 3	Large	Slow	Bad	Unpreferable
Rule 4	Large	Medium	Good	Good
Rule 5	Large	Medium	Medium	Acceptable
...

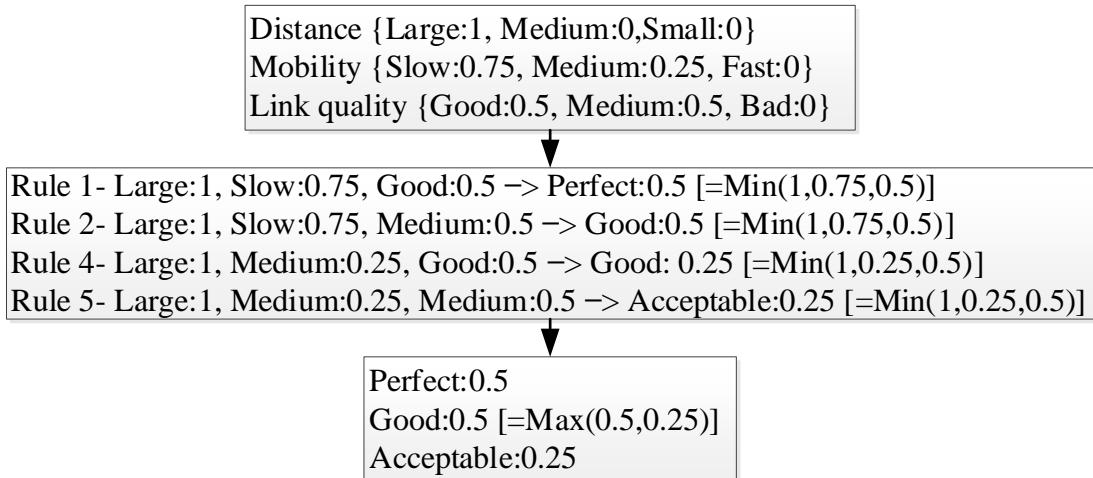


图 2-2 模糊规则推演示例

Fig.2-2 Example for fuzzy rule evaluations

2.3 复杂网络理论

复杂网络是将现实世界复杂系统抽象成网络来进行研究的一种理论工具。互联网可以看做计算机通过通信介质互相连接形成的一个无标度复杂网络,再比如社交网络、电力网络、物流网络等等。在 5G 通信环境中的车联网和互联网类似,可以看做通过某种关系将不同车辆节点连接的复杂网络,其连接具有多样性,信息交互速度快,并且在

不断发展演化。

复杂网络使用图模型 $G(V, E)$ 进行抽象，由顶点集 V 和边集 E 组成。为了刻画复杂网络的网络拓扑，其结构和性质通过下面介绍的参数进行描述。

(1) 度 (Degree) 及度分布 (Degree Distribution)。节点 i 周围共有几个节点与它直接相连定义为度 k_i 。度分布即网络中节点度为 k 的节点数在总节点数中所占的比率。复杂网络的度分布符合幂律分布，具有大多数节点度值较小和少数节点度值较大的特点，具有这样性质的网络也称为无标度网络。

(2) 介数 (Betweenness)。节点或者边的重要程度通常使用介数来描述，分为点介数和边介数。网络中任意两个节点经过节点 i 的最短路径数量占所有两点最短路径总数的比例定义为点介数，边介数与点介数类似，定义为网络中任意两个节点经过边 e_i 的最短路径数量占所有两点最短路径总数的比例。

(3) 平均路径长度 (Average path length)。定义为任意两个节点间最短距离的平均值，网络的平均路径越小，代表网络的连通性越好。

(4) 聚类系数 (Clustering coefficient)。反映节点的聚集程度，任意一个节点的相邻节点也有可能是相邻的。

复杂网络具有无标度特性和小世界性，无标度特性反映在度值高的节点占少数，而度值低的节点占大多数。小世界性是指网络具有大的聚类系数和小的平均路径。

2.4 MK5 硬件平台介绍

MK5 是 Cohda Wireless 公司推出的面向车联网的 DSRC 设备，提供 MK5-OBU 和 MK5-RSU 产品，主要应用 802.11p 协议栈，在恶劣户外环境也能拥有较强无线通信能力。MK-5 RSU 和 MK-5 OBU 拥有一样的基础功能，内部结构一样，处理器单元搭载恩智浦公司的 i.MX 6Quad 四核处理器，DSRC 芯片其射频前端采用的是该公司的 RoadLINK 产品 SAF510x，适用于 V2X 应用，收发信机及基带采用 SAF5100。MK5 运行 Linux 操作系统，用户通过应用程序接口 (Application Programming Interface, API) 开发 ITS 应用程序在 MK5 中编译运行，并提供高精度定位系统。Cohda Wireless 最新推出的 C-V2X 产品 MK6C 符合中国标准和规范，并积极推动 C-V2X 实验，推动中国智能网联车辆技术发展，这也表明 C-V2X 技术后来居上，围绕 C-V2X 的通信芯片、模组、终端等上下游厂商都在积极布局，抢占市场先机。

MK5 系统支持欧洲 V2X 标准 ETSI ITS-G5 和 DSRC 标准，其应用层通过定义 DSRC 标准消息集、多种数据帧格式及数据元素，采用紧凑的消息编码方式。标准消息集分为消息、数据帧和数据元素三级并进行定义，消息集采用 ASN.1 (Abstract Syntax Notation 1, ASN.1) 的非对齐压缩编码规则。掌握 SAE J2735 定义的基本安全消息(Basic Safety Message, BSM) 是实现本文 V2X 的基础。BSM 的消息分为两部分，必选内容部分包括车辆信息即车辆的位置、运行状态信息等，可选内容部分包括车辆的事件信息。BSM 数据包是最基础的 V2X 消息，涉及到汽车的安全控制，不允许被监听，因此在发送和接收过程中需要加密编码和解密译码，BSM-Shell 是传输和接收 BSM 消息的应用程序，发送时间间隔为 50ms，DSRC 设备在发送时会将上位机传输的 UDP 报文等编码为 BSM 消息通过 DSRC 协议广播出去，而在接收时解码所有收到的 BSM 消息，并将解码的 BSM 记录到日志文件或者转到相应的 UDP 接口。

许多研究基于 MK5 平台开发了不同的 DSRC 车联网应用场景或程序，文献[59]提出碰撞预警纠错算法提高前向碰撞预警应用性能。文献[60]设计基于云计算的碰撞预警系统，支持车联网低时延和关键安全服务。文献[61,62]基于车辆雾计算架构设计和实现交通异常检测与预警系统，系统的目标是基于实时视频检测异常并给其他车辆发送预警消息，其他车辆能够及时回应减少交通拥堵的概率，提高交通效率。还有许多关于防碰撞预警的研究^[63-65]。

2.5 本章小结

本章介绍了车联网通信技术标准以及本文所采用模糊逻辑技术、复杂网络理论以及系统模型原型实现所使用的 MK5 平台，是后续章节的技术理论和原型实现基础。

第三章 面向数据安全的智能检测系统与部署技术研究

本章通过仿真车辆通信异常行为获取车联网通信记录数据，用于智能检测系统训练。同时考虑检测系统部署成本提出基于模糊逻辑的部署方案，仿真结果验证了检测系统的准确性和部署方案的合理性。

3.1 数据获取

KDD 99 数据集是有线网络入侵检测领域经典、权威、使用最广泛、公认的标准数据集。自 1999 年至今已有 21 年的历史，在现在看来已经过时，并且车联网中的一些特定攻击在数据集中并没有体现，无法代表真实的车联网攻击特征。因此，本章通过仿真来产生车联网入侵检测数据集。

3.1.1 车辆移动模型

本章通过使用城市交通模拟器（SUMO，版本：0.12.3）和车载网络移动模型生成器（MOVE，版本：2.91）来产生城市车辆移动场景，车辆移动采用曼哈顿移动模型，选择该类型的移动模型原因是车辆对方向的选择更加灵活适用。SUMO 与 MOVE 的城市交通仿真如图 3-1。MOVE 设定车辆数量、线路（file name.rou.xml）和地图（file name.net.xml）文件，进一步产生 SUMO 跟踪文件（file name.sumo.tr）并转化为 NS2（版本：2.35）格式文件（file name.tcl）。

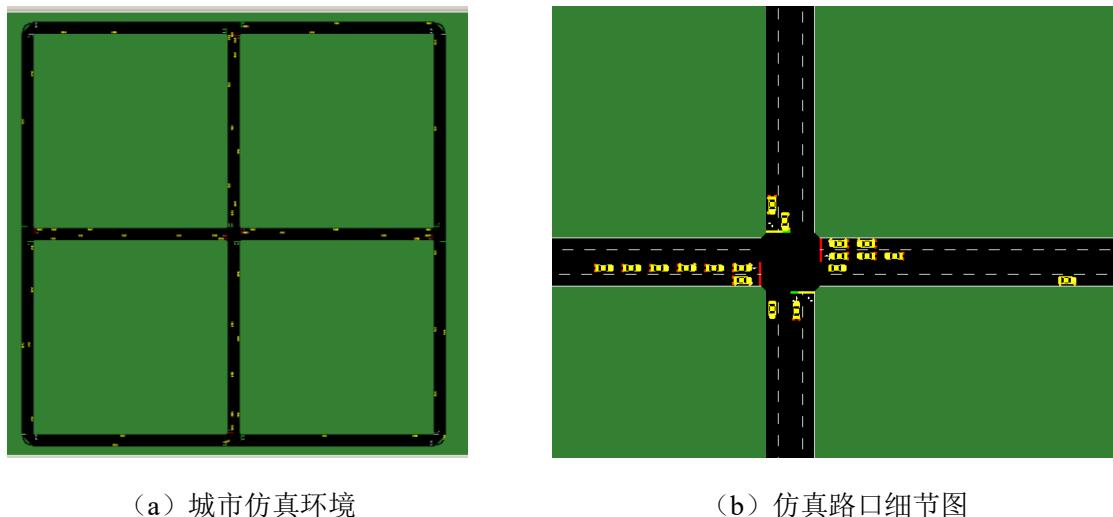


图 3-1 SUMO 与 MOVE 城市仿真图

Fig.3-1 Screenshot of SUMO and MOVE city simulation

3.1.2 仿真环境

通过 NS2 创建车载自组织网络环境如图 3-2，仿真参数设置如表 3-1。

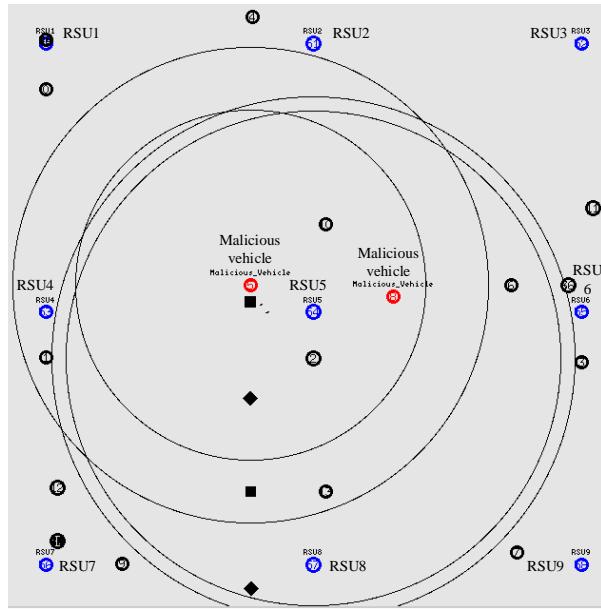


图 3-2 NS2 NAM 仿真图

Fig.3-2 Screenshot of simulation in NS2 NAM

表 3-1 仿真环境与参数

Table.3-1 Simulator environmental and parameters

参数名称	参数值
实验场景	2 车道 500×500 (m) 城市道路
最大车辆速度	30 m/s
车辆数量	50
RSU 数量	9
无线通信协议	802.11p
路由协议	AODV
传输范围	250 m
仿真时间	300 s

3.1.3 车辆恶意行为

为了衡量检测系统的性能，车辆需要产生正常和恶意类型的行为，当车辆丢掉所有接收到的数据包或者在转发其他车辆发送到目的车辆的数据包时将其丢掉，该车辆

定义为恶意车辆。这种恶意行为通过修改 AODV 路由协议来实现，车辆表现为丢包而不是将数据包转发到目的节点。如图 3-2 中恶意车辆 5 丢掉车辆 2 发送的数据包而不是转发到目的节点。由于车辆恶意行为的存在导致产生的跟踪文件与正常行为具有不同的特征。在仿真场景中设置了两个恶意车辆不断产生异常行为。

3.1.4 特征数据集

NS2 仿真后产生的跟踪文件有新旧两种格式，本次仿生成的为新跟踪文件格式，跟踪文件的特征如表 3-2。从跟踪文件中提取特征并进行归一化预处理。

表 3-2 新跟踪文件特征

Table.3-2 Features of new trace file

特征名称	解释	特征名称	解释
Event (s/r/d/f)	事件类型	Is	源端口号
Time (t)	时间	Id	目的端口号
Hs	源节点 id	It	分组类型
Hd	目的节点 id	Il	分组大小
Ni	802.11p	If	流 id
Nx Ny Nz	节点坐标	Ii	唯一 id
Ne	节点能量级别	Iv	TTL 值
Nl	节点跟踪级别	P	应用层类型
Nw	事件原因	Pt	数据包标记
Ma	持续时间	Ph	生存计数
Md	MAC 源地址	Pb	广播 id
Ms	MAC 目的地地址	Pd	目的地址
Mt	MAC 类型	Ps	源地址

3.2 智能检测系统

智能检测系统采用前馈神经网络来检测识别车载自组织网络中的异常行为车辆。数据集为超过 53000 条跟踪特征记录，描述了车联网中的正常异常行为，数据集划分

为训练集（70%）和测试集（30%）。初始参数对于智能检测系统的性能非常重要，经过反复尝试确定最佳训练比例，训练阶段设置的参数 TrainParam.EPOCHS=100，TrainParam.LR=1*10⁻⁵，TrainParam.GOAL=0 和 TrainParam.MIN_GRAD=1*10⁻¹²。前馈神经网络包括输入层、隐藏层和输出层，输入层包含 30 个神经元，与提取特征数相等，最好的训练效果对应隐藏层神经元数为 6 个，输出层包含 2 个神经元，即正常和异常，其结构如图 3-3。

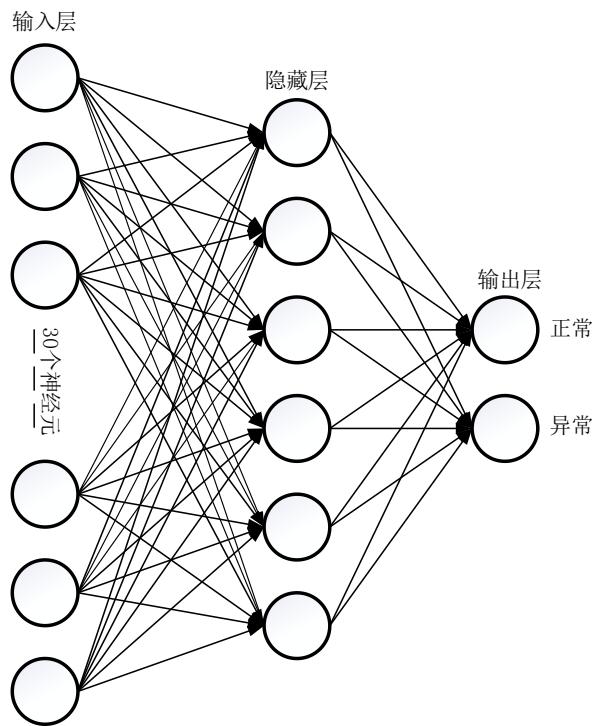


图 3-3 人工神经网络结构

Fig.3-3 Structure of the artificial neural network

3.3 基于模糊逻辑的智能检测系统部署方案

在 RSU 部署密集的城市街道场景，智能检测系统可以部署在 RSU 上来进行实时检测功能，保证该区域的车联网通信安全。考虑到 RSU 可能还具备其他比如预警功能，当 RSU 计算能力不够、通信受限、存储能力弱时，模型的训练部署在云端，当车辆与 RSU 通信时，RSU 收集这些数据上传到云端，进一步训练，并不断优化模型。在 RSU 分布较少的区域，考虑 RSU、智能检测系统的部署成本以及安全问题，提出基于模糊逻辑的代理选择方案，将 RSU 的智能检测功能进一步下沉到车辆，选择合适的车辆作为代理协助 RSU 收集数据并完成异常检测功能。

部署方案基于车辆的历史数据，考虑车辆的速度、领导力、路线因素作为选择代理的标准。速度因素包括车辆平均速度（VFm）及其速度方差（VFv），领导力因素包括平均领导力（LFm）及其领导力方差（LFv），领导力指车辆在其通信距离范围内其他车辆的数量，路线因素（RF）指车辆行驶的距离。其计算公式如下

$$VFm(s)_{s \in N_s} = \frac{1}{T} \sum_{t=1}^T v(s, t) \quad (3.1)$$

其中 $v(s, t)$ 表示车辆 s 在 t 时刻的速度， T 表示车辆移动的总时间。 N_s 表示所有车辆的集合。

$$VFv(s)_{s \in N_s} = \frac{1}{T} \sum_{t=1}^T (v(s, t) - VFm(s))^2 \quad (3.2)$$

$$LFm(s)_{s \in N_s} = \frac{1}{T} \sum_{t=1}^T C(s, t) \quad (3.3)$$

$$LFv(s)_{s \in N_s} = \frac{1}{T} \sum_{t=1}^T (C(s, t) - LFm(s))^2 \quad (3.4)$$

其中 $C(s, t)$ 表示车辆 s 通信范围内的所有车辆。

$$RF(s)_{s \in N_s} = \sqrt{\sum_{t=1}^T ((x(s, t) - x(s, t+1))^2 + (y(s, t) - y(s, t+1))^2} \quad (3.5)$$

其中 $x(s, t); y(s, t)$ 表示车辆 s 在时刻 t 的横纵坐标。

部署方案计算车辆的不同因素值，由于各个因素大小单位不同，统一进行归一化处理。由于各个因素没有一个标准来衡量好坏，将各个因素划分为不同的模糊等级，比如 VFm 对应 {Slow, Medium, Fast}，当计算的 VFm 值为 0.75，根据输入隶属函数对应的模糊输入变量为 {Slow:0, Medium:0.5, Fast:0.5}（如图 3-4 VFm）。选择速度因素适中、领导力因素强、路线范围广的车辆作为代理来部署检测系统，能够收集更多车辆通信数据，协助 RSU 提高检测精度和效率。各因素对应的输入输出隶属函数如图 3-4。根据每个因素有不同的等级，制定模糊规则表，根据模糊规则表以及输出隶属函数来计算最后输出值。每个输入因数对应三个等级，因此规则表共有 243 条规则，部分规则见表 3-3。所有输入因素对应的模糊等级按照对应模糊规则表中的规则通过最小最大（Min-Max）原则计算出最后的输出模糊等级值，通过求解模糊等级值对应图形的重心去模糊化，重心的横坐标即为模糊输出值，作为是否选择该车辆部署检测系统的评估标准。

本节提出的智能检测系统以及部署代理选择方案可以总结为图 3-5。MOVE、SUMO、

NS2 等仿真工用来产生特征数据集，数据的预处理、智能检测系统的训练、测试以及部署代理的模糊逻辑选择方案由 Matlab 完成。

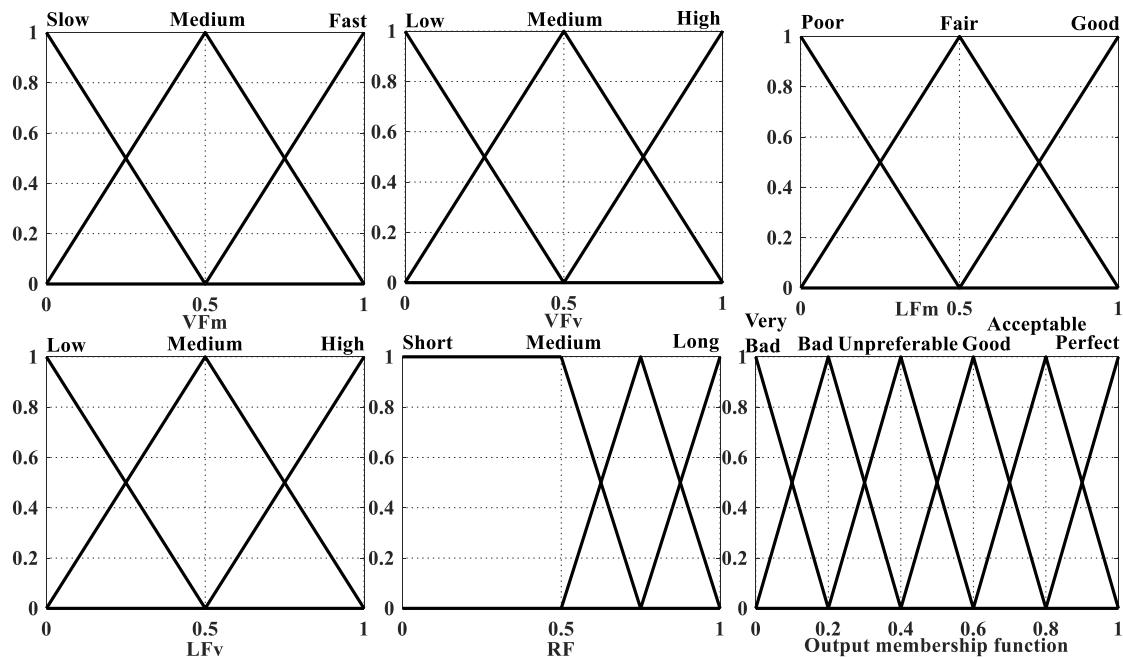


图 3-4 输入输出隶属函数

Fig.3-4 Input and output membership function

表 3-3 模糊规则

Table.3-3 Fuzzy rule

Rules	VFm	VFv	LFm	LFv	RF	Rank
Rule1	Slow	Low	Good	Low	Long	Perfect
Rule2	Slow	Low	Good	Low	Medium	Good
Rule3	Slow	Low	Good	Low	Short	Unpreferable
Rule4	Slow	Low	Good	Medium	Long	Good
Rule5	Slow	Low	Good	Medium	Medium	Acceptable
Rule6	Slow	Low	Good	Medium	Short	Acceptable
Rule7	Slow	Low	Good	High	Long	Unpreferable
Rule8	Slow	Low	Good	High	Medium	Good
...
Rule243	Fast	High	Poor	High	Short	VeryBad

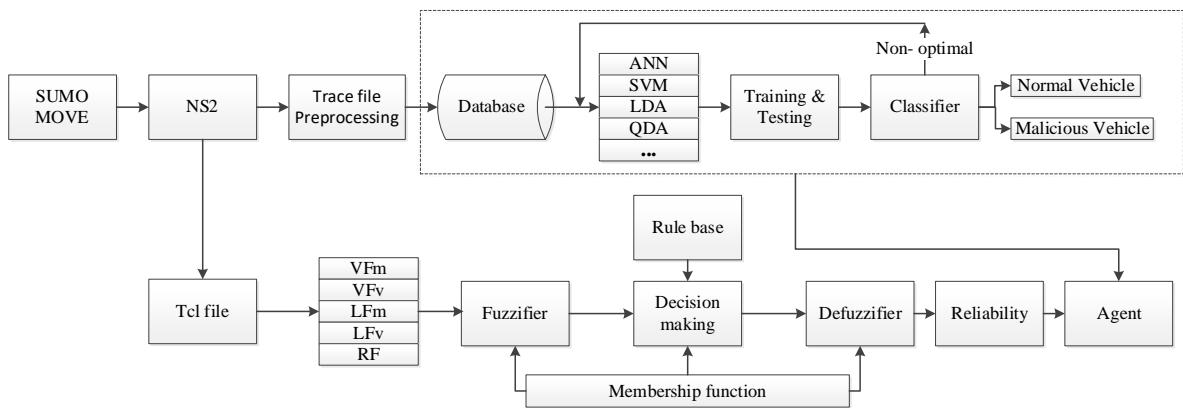


图 3-5 智能检测系统及部署代理选择流程

Fig.3-5 Intelligent detection system and deployment agent selection process

3.4 性能分析

为了评估检测系统的性能，主要计算的指标包括： TP （真阳）：正常记录识别为正常； FP （假阳）：正常记录识别为异常； TN （真阴）：异常记录识别为异常； FN （假阴）：异常记录识别为正常。准确率 (*Accuracy*)、精确率 (TP_{Rate})、召回率 (TN_{Rate})、漏报率 (FN_{Rate})、误报率 (FP_{Rate})，计算公式如下

$$Accuracy = \frac{\text{正确识别的类别数 (正常/异常)}}{\text{类别总数 (正常/异常)}} \quad (3.6)$$

$$TP_{Rate} = \frac{TP}{TP+FN} \quad (3.7)$$

$$TN_{Rate} = \frac{TN}{TN+FP} \quad (3.8)$$

$$FN_{Rate} = 1 - TP_{Rate} = \frac{FN}{FN+TP} \quad (3.9)$$

$$FP_{Rate} = 1 - TN_{Rate} = \frac{FP}{FP+TN} \quad (3.10)$$

智能检测系统训练后分类准确率以及识别率如表 3-4 和表 3-5。从结果来看，该测试集包括 19056 条正常记录和 2433 条异常记录，匹配记录表示真实记录和人工神经网络预测的结果相同，未匹配记录则表示预测结果与真实记录不同。智能检测系统对正常记录的识别率达到 99.59%，异常记录达到 96.75%，系统漏报率为 0.41%，预警率在 96.79% 与 99.59% 之间，具有不错的识别精度，除了人工神经网络外，一些其他的检测方法如图 3-5 中所示，比如支持向量机 (SVM)、线性判别分析 (LDA)、二次判别分析 (QDA) 也可以实现异常检测。

表 3-4 分类准确率

Table.3-4 Accuracy of classification

记录类别	智能检测系统				
	真实记录	人工神经网络	匹配记录	未匹配记录	准确率
正常	19056	19057	18978 (<i>TP</i>)	79 (<i>FP</i>)	99.59%
异常	2433	2432	2354 (<i>TN</i>)	78 (<i>FN</i>)	96.75%

表 3-5 识别率

Table.3-5 Recognition rate

预警类型	TP_{Rate}	TN_{Rate}	FN_{Rate}	FP_{Rate}
准确率	99.59%	96.79%	0.41%	3.21%

为评估检测系统部署方案的性能，采用不同于检测系统训练及测试的数据，新的数据记录共 105338 条，设置 3 个异常节点（节点编号：5, 8, 13）产生异常行为。通过部署在不同车辆节点对检测结果进行分析及对比，部署在不同车辆节点识别的异常记录检测结果如图 3-6、图 3-7。根据检测系统的性能具有一定的误报率，并且检测系统只识别记录并没有检测出产生异常行为的车辆，通过不同代理识别的异常记录占比可以发现一些不同的识别特点，识别出来的异常记录对应的节点即为检测系统的怀疑对象，占比越高则是异常车辆的可能性更高。基于模糊逻辑方案选择的代理（Fuzzy Agent），其识别出的异常记录占比前三为 5,8,13，如果设置占比阈值为 10%，则可以准确识别出异常节点，如果阈值更小，则只需要另外考虑其他四个节点，而通过随机选择出来的节点部署（Random Agent 1,Random Agent 2,Random Agent 3）可能就会出现每个车辆都具有怀疑的可能性，依据占比排行或设置阈值也会出现检测错误的情况，比如 Random Agent 1 按照占比会检测出 14、17 节点为异常节点，Random Agent 2, Random Agent 3 也会出现检测错误的情况，并且怀疑对象涵盖多数节点，这也说明检测系统部署在不同节点对检测系统性能有不同的影响。基于模糊逻辑的检测系统部署方案能够综合考虑节点速度、领导力、路线因素，协助 RSU 收集更多更广泛更全面的数据达到一个较好的检测效果，比随机部署具有更少的怀疑对象，能够提高识别效率和精度。

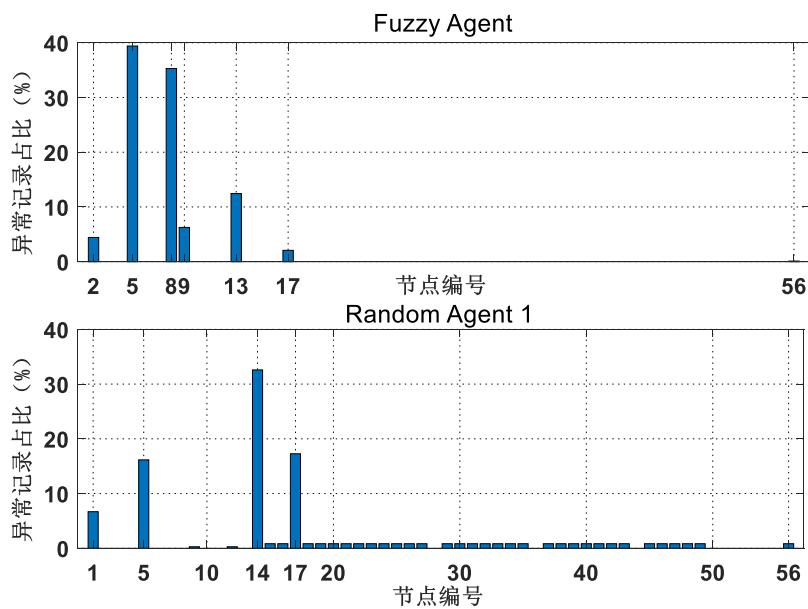


图 3-6 异常记录识别占比

Fig.3-6 Percentage of abnormal record identification

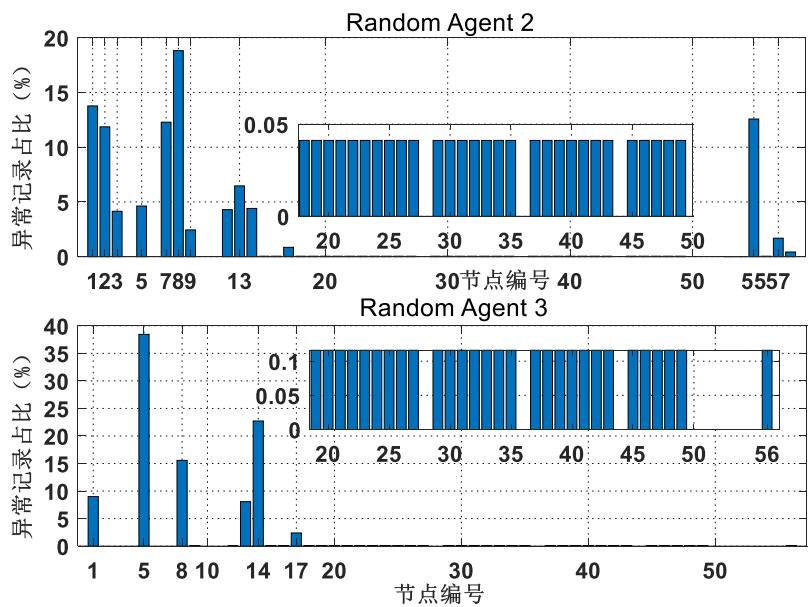


图 3-7 异常记录识别占比

Fig.3-7 Percentage of abnormal record identification

3.5 本章小结

本章通过车联网仿真软件在小场景下仿真了车辆正常异常通信行为，提出基于人工神经网络的智能检测系统及基于模糊逻辑的代理选择方案，相比有线网络入侵检测，存在数据集过时，缺少车联网特定攻击等缺点，本章是对车联网仿真数据入侵检测的一个尝试，现有的工作证明了仿真数据用于检测的可行性并且具有一定的检测效果。

第四章 车联网分簇及簇内数据传输策略研究

本章通过分析真实出租车 GPS 数据集,从复杂网络理论角度构建车联网通信模型,提出基于车辆广义距离的分簇及基于模糊逻辑的簇头选择方案,在此基础上建立簇内数据传输优化模型,能够有效改善簇内车辆通信状况。

4.1 系统模型

在车载网络中,车-车、人-车、车-基础设施通信方式众多以及车辆快速移动导致网络拓扑动态变化,尤其是在大型密集城市,车辆与其他通信设备的通信构成复杂的城市车载网络,链路具有多样性和时变性。复杂网络理论能够提供精确的网络描述模型帮助网络设计、优化及管理,本节基于复杂网络理论分析车载网络的特征并建立系统模型。

4.1.1 数据集

本节数据集基于成都市出租车 2014 年 8 月 3 日到 8 月 23 日真实 GPS 数据,其数据格式及示例为:出租车 ID,纬度,经度,载客状态(1 表示载客,0 表示无客),时间点;1,30.4996330000,103.9771760000,1,2014/08/03 06:01:22。本节选取经度[103.90,104.20]和纬度[30.55,30.80]范围内的车辆 GPS 数据用于构建车联网模型。图 4-1 为成都市某

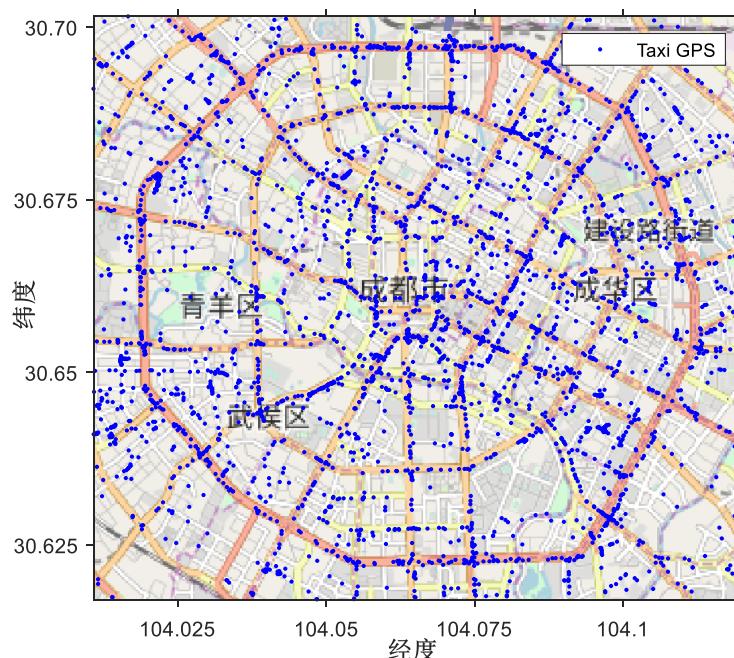


图 4-1 成都市出租车 GPS 坐标分布

Fig.4-1 Taxis GPS distribution in Chengdu

一时刻出租车的 GPS 坐标分布。

4.1.2 无向图模型

建立车辆通信无向图模型 $G(V,E)$, 其中 V 为所有车辆集合, E 为车辆间通信链路集合。所有车辆和 RSU 具有相同的通信能力, 一对节点 (车辆) 只要在最大通信范围 r 内就能实现彼此通信, 假设加性高斯白噪声, 功率为 $N_0^{[26]}$ 。

4.1.3 复杂网络验证

节点的度分布如图 4-2 所示。节点度越大, 即拥有大量通信链接, 这种情况出现的概率较低, 而大部分节点拥有少量的通信链接, 概率较高。节点的度分度符合幂律分布, 反应了复杂网络的无标度特性。

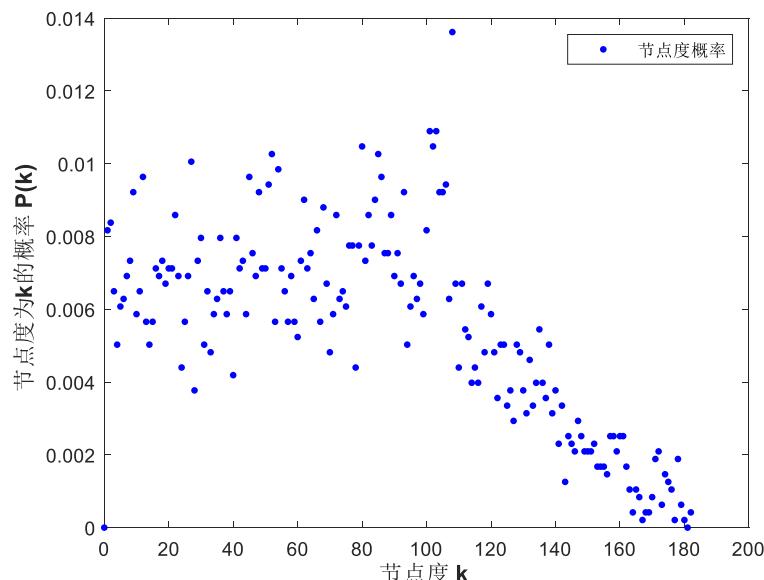


图 4-2 节点度分布

Fig.4-2 Node degree distribution

聚类系数: 反映车辆 i 的邻居车辆与其他车辆通信的特征, 是网络的局部特征, 车辆 i 的聚类系数定义如下

$$\alpha_i = \frac{E_i}{k_i(k_i-1)/2} \quad (4.1)$$

其中 E_i 为所有邻居车辆的通信链路数量总和。

中介中心性: 量化节点的重要性, 车辆 i 的中介中心性定义如下

$$B_i = \frac{2}{(N-1)(N-2)} \sum_{s \neq i \neq t} \frac{n_{st}^i}{g_{st}} \quad (4.2)$$

其中 g_{st} 为节点 s 到节点 t 的最短路径的数量, n_{st}^i 为节点 s 到节点 t 最短路径途中经过节点 i 的路径数量, N 为所有节点的数量。

特征路径长度: 定义为节点对之间最短路径的平均值, 公式如下

$$L = \frac{2}{N(N-1)} \sum_{i,j=1; i \geq j}^N h_{ij} \quad (4.3)$$

其中 h_{ij} 表示节点 i 到节点 j 的最短路径经过的跳数。

每个节点的中介中心性与聚类系数如图 4-3 所示。网络的平均聚类系数 $\bar{o} = \frac{1}{N} \sum_{i=1}^N o_i = 0.6742$, 特征路径长度 $L=6.7251$, 网络的特征路径长度符合六度分割理论 (网络中两个节点的最大连接间隔不超过六个节点), 具有较高的聚类系数, 基于以上参数分析, 车载网络具备小世界特性和无标度特性。在具有这样特性的系统中, 数据传输快, 少量节点在信息传输和分发过程中扮演重要角色。

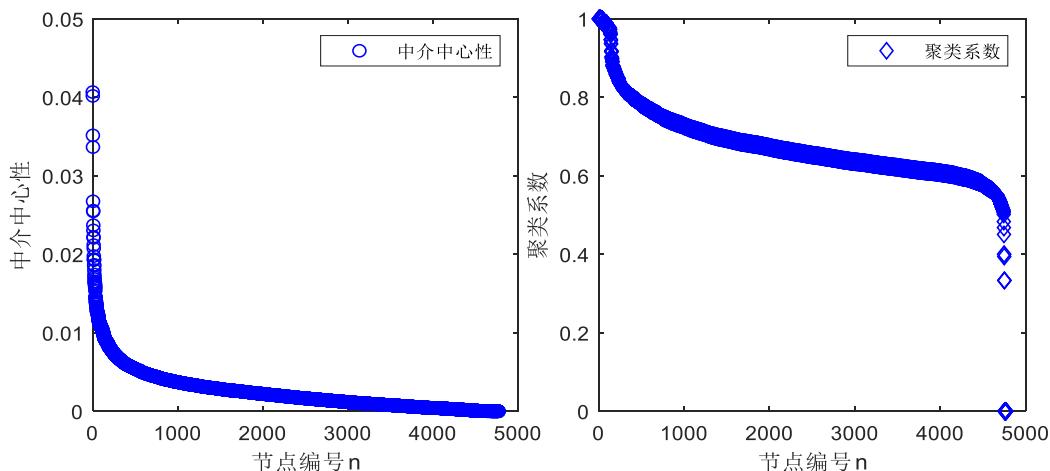


图 4-3 中介中心性与聚类系数

Fig.4-3 Betweenness centrality and clustering coefficient

随着时间的变化, 复杂网络的相关特征也会发生变化。表 4-1 显示了网络拓扑参数 (如车道上的车辆数、平均节点度、中介中心性、聚类系数) 在一天中不同时刻的变化。具体来说, 成都早上 8:00, 许多出租车司机可能因为交通拥堵而不愿在路上浪费时间, 车辆数量最少, 上午 10:00 之后, 道路上出租车数量越来越多, 导致平均节点度增加, 尽管节点度随时间变化, 但复杂网络的参数基本保持稳定。在考虑中介中心性和聚类系数的基础上, 可以认为车载网络拓扑是时不变的以及具备小世界和无标度的特性。

表 4-1 拓扑时不变验证

Table 4-1 Topological time-invariance verification

时间	车辆数	平均节点度	平均距离	边介数	聚类系数
8:00 a.m.	2024	4.7273	838	0.0040	0.5869
10:00 a.m.	2647	4.6889	816	0.0043	0.6970
12:00 noon	2759	7.0476	749	0.0347	0.6983
14:00 p.m.	2863	5.2273	817	0.0071	0.5653
16:00 p.m.	2880	4.3889	802	0.0026	0.6842
18:00 p.m.	2731	4.0500	841	0.0123	0.6123
20:00 p.m.	2842	3.0667	829	0.0069	0.5500

在空间分布上，通过分析不同区域，选取了交通密集的 40 个区域内的数据（40 个半径为 250m 的圆形非重叠区域）。交通密度图如图 4-4 所示，统计了不同时刻不同区域内的车辆数量，将结果按升序显示如图 4-5。可以看出，在中午 12:00 以后，道路上出租车的数量保持相对稳定，遵循相似的空间分布，尽管早上节点数较少，但归一化后的空间分布与其他时间记录类似。

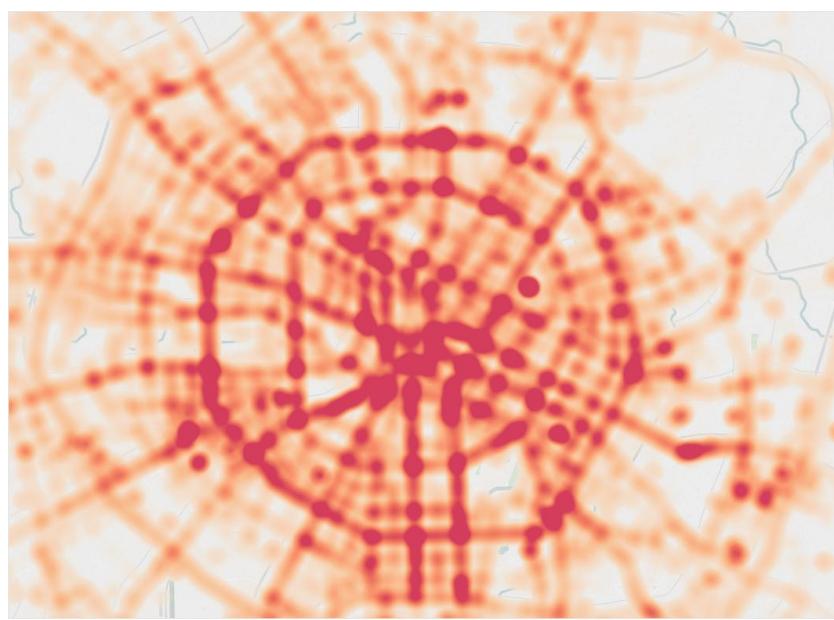


图 4-4 交通密度图

Fig. 4-4 Traffic density map

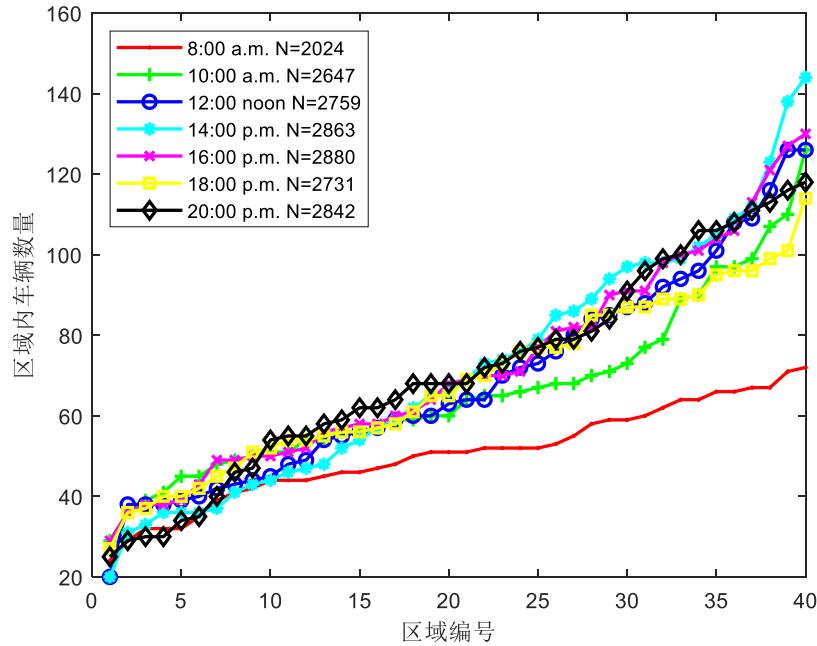


图 4-5 空间分布时不变

Fig.4-5 The time-invariant spatial distribution

4.2 分簇算法

通过复杂网络理论分析，少数节点能够起到数据收集分发的关键作用。整个网络需要划分不同的簇群来减轻网络负载，每个簇群选择出关键的节点承担数据收集分发功能，而簇群需要尽可能持续稳定的维持，因此提出辅助数据传输的分簇算法。车辆之间在一定范围内形成相对稳定的簇需要考虑车辆周围的邻居车辆数量、车辆与邻居车辆的速度差以及距离差，定义车辆之间的广义距离如公式（4.4）所示。广义距离反映两个车辆之间的相似度，广义距离越大，两个车辆之间形成簇的可能性越小^[26]。

$$D_{ij} = \xi_1(k_i B_i + k_j B_j) + \xi_2 v_{ij} + \xi_3 d_{ij} \quad (4.4)$$

其中 k_i, k_j 为车辆 i, j 的节点度， B_i, B_j 为车辆 i, j 的中介中心性， v_{ij} 为车辆 i, j 之间的速度差， d_{ij} 为车辆 i, j 之间的距离， ξ_1, ξ_2, ξ_3 为权重系数。

基于广义距离的分簇算法见表 4-1（算法 4.1）。

4.3 簇头选择算法

簇头在簇群里扮演重要角色，如何选择出合理的簇头是保证簇群内数据稳定收集与分发的关键。为了选出相对稳定的簇头，本节基于模糊逻辑考虑车辆的领导力、速

表 4-1 基于广义距离分簇

Table 4-1 Clustering based on generalized distance

算法 4.1 基于广义距离分簇**初始化**

- 1: 计算车辆之间的欧氏距离 d_{ij} ;
- 2: 根据 $r=1000m$ 产生邻接矩阵和度矩阵 d_M ;
- 3: 计算 $k_i, k_j, B_i, B_j, v_{ij}$;
- 4: 计算广义距离矩阵 D_{ij} ($\xi_1 = \xi_2 = 0.3, \xi_3 = 0.4$)。

谱聚类

- 5: 生成拉普拉斯矩阵 $L_M, L_M = d_M - D_{ij}$;
- 6: 计算归一化拉普拉斯矩阵 \bar{L}_M ;
- 7: 计算 \bar{L}_M 特征值和特征向量，按特征值从小到大排序;
- 8: 使用 K-means 对新排序的向量空间进行聚类。

输出

- 9: 分簇结果

度、距离三个因素 (LF, VF, DF)。领导力因素主要考虑车辆节点度和中介中心性，领导力因素指车辆与邻居车辆的综合速度差 (根据节点度所占的权重计算)。距离因素指的是车辆与邻居车辆的综合欧氏距离，只要车辆周围的邻居节点越多，综合距离越小，选取的簇头生存周期更长，在簇内负责协调与其他车辆的通信更加稳定可靠。车辆 i 的领导力因素、速度因素、距离因素计算如下

$$LF_i = k_i B_i \quad (4.5)$$

$$VF_i = \sum_{j=1}^N \frac{c_{ij} k_j v_{ij}}{k_i^2} \quad (4.6)$$

$$DF_i = \sum_{j=1}^N \frac{c_{ij} k_j d_{ij}}{k_i^2} \quad (4.7)$$

其中 N 为簇内车辆总数， $c_{ij} \in \{0,1\}$ 表示车辆 i 与簇内车辆 j 是否通信。

由于各因素大小量级不同，所有因素都需要进行归一化处理。归一化的因素可以通过隶属函数写出对应的模糊语言变量，不同因素的模糊语言变量通过规则库定义的模糊规则进行计算，输出的模糊语言变量通过输出隶属函数并进行去模糊化得到模糊

逻辑输出值。输入因素对应的输入隶属函数如图 4-6。三个输入因素，每个因素对应三个模糊等级，所以共有 27 条模糊规则，模糊规则见表 4-2。

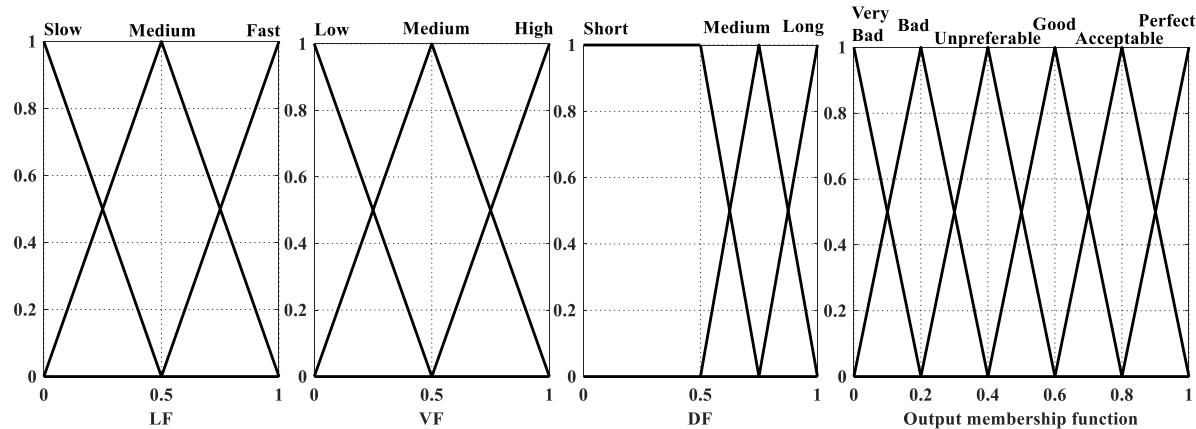


图 4-6 输入输出隶属函数

Fig.4-6 Input and output membership function

表 4-2 模糊规则

Table 4-2 Fuzzy rule

Rules	LF	VF	DF	Rank
Rule 1	Good	Slow	Close	Perfect
Rule 2	Good	Slow	Medium	Good
Rule 3	Good	Slow	Far	Unpreferable
Rule 4	Good	Medium	Close	Good
Rule 5	Good	Medium	Medium	Acceptable
Rule 6	Good	Medium	Far	Bad
Rule 7	Good	Fast	Close	Unpreferable
Rule 8	Good	Fast	Medium	Bad
Rule 9	Good	Fast	Far	Very Bad
Rule 10	Fair	Slow	Close	Good
Rule 11	Fair	Slow	Medium	Acceptable
Rule 12	Fair	Slow	Far	Bad
Rule 13	Fair	Medium	Close	Acceptable

Rule 14	Fair	Medium	Medium	Unpreferable
Rule 15	Fair	Medium	Far	Bad
Rule 16	Fair	Fast	Close	Bad
Rule 17	Fair	Fast	Medium	Bad
Rule 18	Fair	Fast	Far	Very Bad
Rule 19	Poor	Slow	Close	Unpreferable
Rule 20	Poor	Slow	Medium	Bad
Rule 21	Poor	Slow	Far	Very Bad
Rule 22	Poor	Medium	Close	Bad
Rule 23	Poor	Medium	Medium	Bad
Rule 24	Poor	Medium	Far	Very Bad
Rule 25	Poor	Fast	Close	Bad
Rule 26	Poor	Fast	Medium	Very Bad
Rule 27	Poor	Fast	Far	Very Bad

根据文献[66]中最大化网络容量来选择网关节点（簇头），其问题形式如下

$$\begin{aligned}
 & \min \quad \Omega \\
 & s.t. \quad RAP - \Omega \mathbf{1} + y = \mathbf{0} \\
 & \quad P^T \mathbf{1} = \mathbf{1} \\
 & \quad y \geq \mathbf{0} \\
 & \quad P \geq \mathbf{0}
 \end{aligned} \tag{4.8}$$

其中 $P = [p(t), t = 1, 2, \dots, N]^T$ 代表每个节点成为簇头的概率， $\mathbf{1} = [1, 1, \dots, 1]^T$ 。 Ω ， y 均为辅助变量，其优化目标为根据节点的峰值负载能力来最大化网络容量，经过松弛、引入辅助变量得到式 (4.8)。

基于本文的通信模型，使用模糊逻辑与最大化网络容量选择簇头的概率如图 4-7。根据各个节点的速度、周围节点数、链路通信能力等状态，得到不同节点成为簇头的概率，只有个别车辆能够被选择为簇头。

4.4 簇内数据传输优化模型

针对簇内大量流媒体数据传输的需求，由于车辆高速移动以及物理层信道衰落效

应会影响分簇后簇群的可靠性以及吞吐量性能，即使在 MAC 层无传输冲突的条件下，物理层信道条件也会对系统数据包丢失产生很大影响。本节从考虑物理层解码失败而引起数据包丢失的角度，基于已经完成的分簇和簇头选择工作，对簇内数据传输进行优化^[26]。

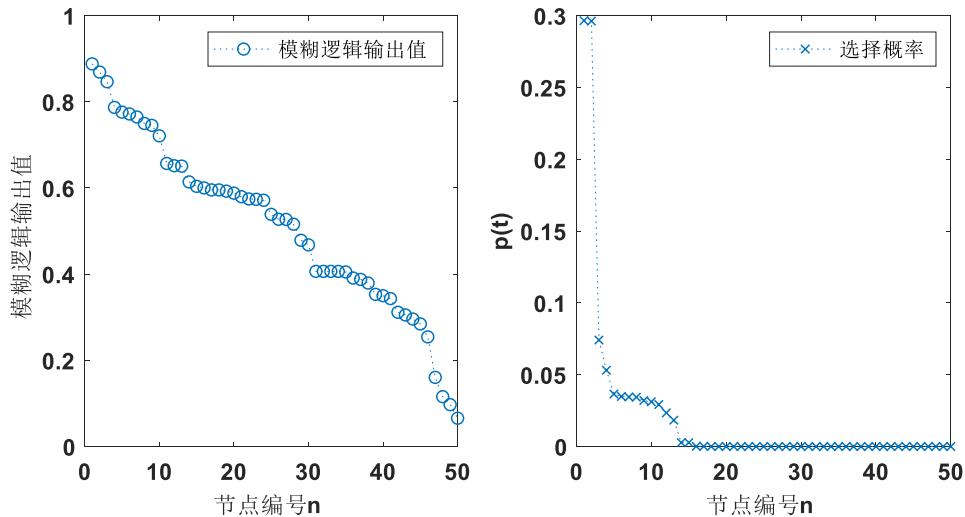


图 4-7 两种方法的簇头选择概率（数据范围为精度[104.03,104.10]和纬度[30.63,30.67]）

Fig.4-7 Cluster head selection probability based on two methods (longitude from 103.9 to 104.2 and latitude from 30.55 to 30.80)

在集群 C_j (j 为簇头) 内，车辆 i 的数据包到达簇头丢失的概率为

$$Pr_i = 1 - \exp(-L_{p_i} K_3 d_{ij}^{K_2}) \quad (4.9)$$

其中 K_2 为路径损失指数， $L_{p_i} = \phi_i/[T_c(v_i) \log_2(1 + \gamma_0)]$ ， ϕ_i 为车辆 i 发送数据包的大小， γ_0 为接收 SNR 阈值。 $T_c(v_i) = \frac{3cf_0}{4\sqrt{\pi}(c+v_i)}$ ， c 为光速， f_0 为载波频率， v_i 为车辆 i 的速度^[67]。

$$K_3 = \frac{K_1 N_0 \gamma_0}{P_t} \quad (4.10)$$

其中 N_0 为加性噪声功率， P_t 为发射功率。

$$K_1 = \frac{(4\pi)^2}{G_t G_r \lambda_0^2} \quad (4.11)$$

其中 $\lambda_0 = c/f_0$ ， G_t ， G_r 为发射机和接收机的天线增益。

集群内每个车辆的吞吐量计算如下

$$\varphi_i = \lambda_i(1 - Pr_i)L_{p_i} \quad (4.12)$$

其中 λ_i 为车辆 i 的数据包到达率。

簇内数据传输优化模型见式 (4.13)

$$\begin{aligned} & \max \quad \sum_{i \in C_j} \varphi_i \\ & s.t. \quad \varphi_i \leq \varphi_0 \\ & \quad \sum_{i=1}^{C_j} \lambda_i \leq \mu \end{aligned} \quad (4.13)$$

φ_0 为每个车辆之间的最大传输速率，每个簇群的数据包到达率上限为 μ 。

4.5 仿真及性能分析

利用 Matlab 对簇内数据传输优化模型进行数值仿真，参数设置见表 4-3。

表 4-3 仿真参数

Table 4-3 Simulation parameters

仿真参数	数值
G_t, G_r : 发送和接收天线增益	1
f_0 : 载波频率	5.9GHz
P_t : 传送功率	2W
γ_0 : 接收 SNR 阈值	3dB
N_0 : 加性噪声功率	3.98×10^{-12} mW
ϕ_i : 每辆车发送数据包大小	400~500Bytes
φ_0 : 吞吐量阈值	400Kbps
μ : 簇群分组到达率上限	200packets/s
K_2 : 路径损失指数	2

比较基于模糊逻辑 (Fuzzy) 和最大化网络容量 (MNC) 两种选择簇头选择方式对簇内数据传输模型中的吞吐量及时延的影响，归一化时延性能如图 4-8，吞吐量性能如图 4-9，簇内每辆车的最大吞吐量为 400kb/s。可以看到，基于模糊逻辑的簇头选择对簇内数据传输的平均吞吐量与平均时延性能优于另一种方法，基于模糊逻辑的方案选择的簇头能够维持与簇内车辆之间稳定的通信，提高与簇头通信的吞吐量并降低数据传输时延。

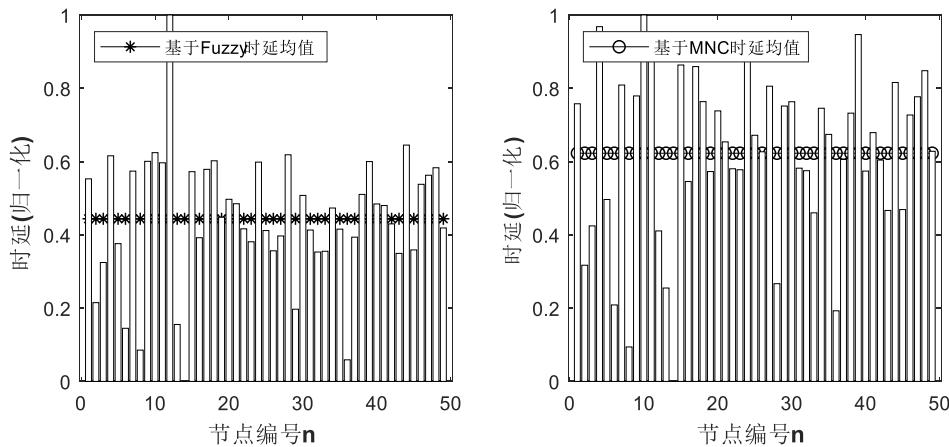


图 4-8 两种方法的时延性能比较

Fig.4-8 The comparision between two methods in terms of the actual time delay

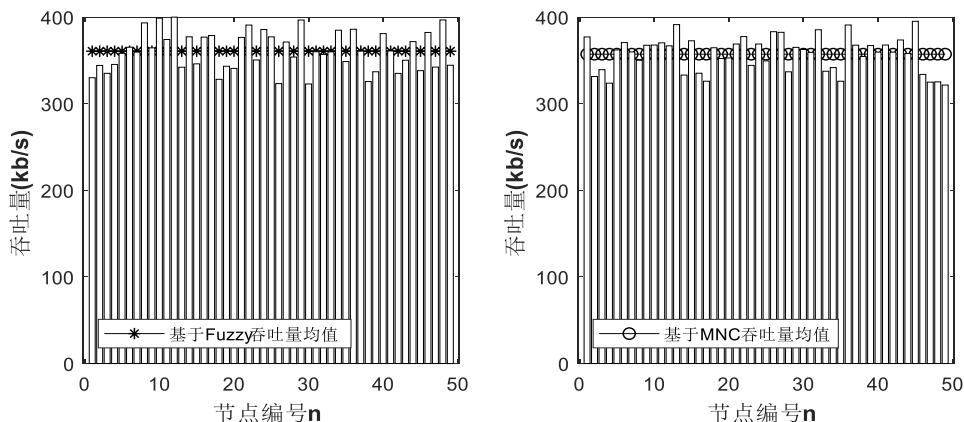


图 4-9 两种方法的吞吐量性能比较

Fig.4-9 The comparision between two methods in terms of the actual throughput

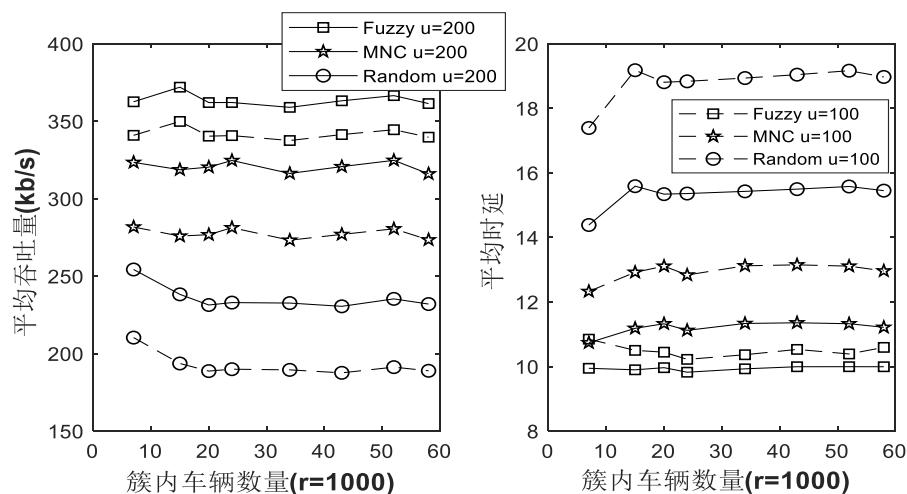


图 4-10 簇内平均吞吐量与平均时延性能比较 ($r=1000$)

Fig.4-10 Comparison of average throughput and average delay in clusters

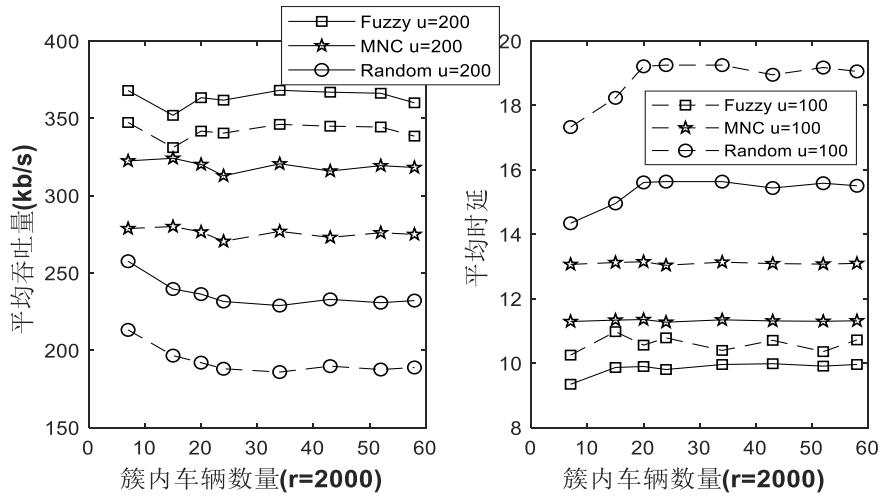
图 4-11 簇内平均吞吐量与平均时延性能比较 ($r=2000$)

Fig.4-11 Comparison of average throughput and average delay in clusters

当簇内车辆数量增加时，簇内平均吞吐量和平均时延性能随簇内车辆数量变化如图 4-10 所示。随簇内车辆数增加，平均吞吐量及平均时延性能变化均维持在稳定的范围内，数据传输模型能够适应车辆数的变化并保持稳定的通信状况。当簇内分组到达率 $\mu = 100$ 时，基于模糊逻辑的方案簇内平均吞吐量高于 MNC 和随机方案，同时能够达到较低的平均时延，当 $\mu = 200$ 时，簇内分组到达率上限增加，簇内通信状况改善，三种方法的平均吞吐量均有提高，同时平均时延降低，但基于模糊逻辑方案的簇内平均吞吐量及平均时延性能均优于其他方案。当通信半径增加为 $r=2000$ 时，其性能变化与 $r=1000$ 时类似，数据传输优化模型随通信半径变化保持良好性能。

簇内时延均值 \bar{t}_d 计算见式 (4.14)

$$\bar{t}_d = 1/C_j \sum_{i \in C_j} \phi_i / \varphi_i = 1/C_j \sum_{i \in C_j} \exp(L_{p_i} K_3 d_{ij}^{K_2}) / \lambda_i \quad (4.14)$$

L_{p_i} 的计算式中，由于 c 为光速，所以车辆速度可以忽略不计，同一簇内车辆节点与簇头之间的传输时延主要取决于簇头与簇内车辆的距离，传输数据包大小和分组到达率。

4.6 本章小结

本节基于真实数据集分析，验证了车载网络符合复杂网络理论特征，并建立车载网络通信模型，提出的基于广义距离的分簇算法划分相对稳定的簇，在簇内选择出稳定的节点作为簇头，保证簇头与簇内成员之间数据传输的稳定性。在较大规模车辆簇内通信环境，所提的综合通信方案能够有效提高簇内网络吞吐量并降低数据传输时延。

第五章 双层车联网架构中的缓存策略研究

本章拓展车载网络现有的单层缓存策略，提出双层车联网缓存架构。单层 RSU 缓存中，由于车辆的移动，难以对其进行建模，而车辆层的缓存空间不容忽视。提出的双层缓存架构考虑联合车辆层和 RSU 层缓存，充分利用车辆和 RSU 的缓存能力来最大化缓存收益。优化目标为最小化用户获取缓存内容的时延以及运营商传输内容的成本，通过分析得到该问题为非线性整数规划问题。为求解该问题，提出交替动态规划搜索算法求解，该算法随 RSU 数量呈指数增长，通过分析以及受该算法求解结果启发，为了减少复杂度，提出适用于大规模场景的合作贪婪算法。通过室内搭建车联网设备平台，实现了提出的系统模型原型。仿真和实测数据验证了提出的缓存策略的有效性。

5.1 缓存系统模型

5.1.1 网络模型

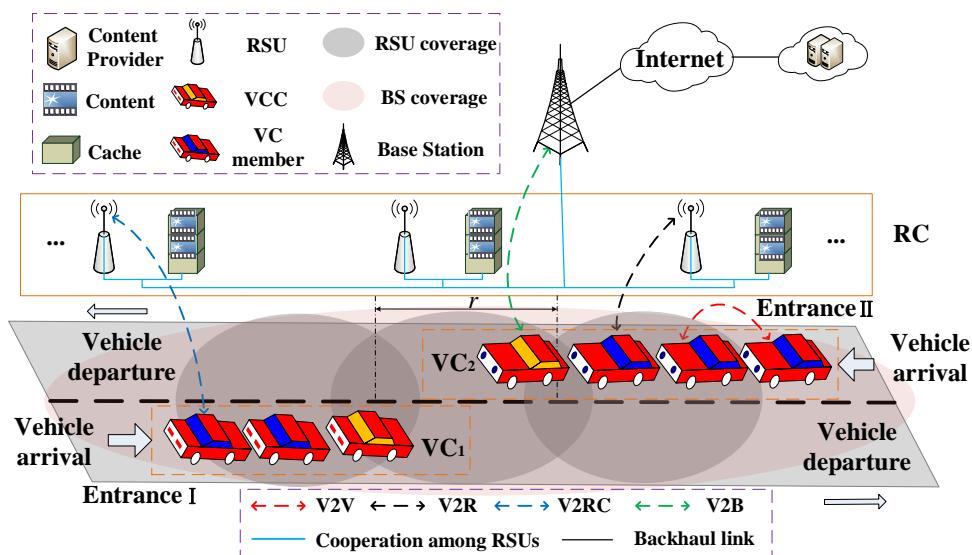


图 5-1 系统模型

Fig.5-1 System model

如图 5-1 所示，考虑双向道路场景，包括车辆，RSU，基站（Base Station, BS），整个车载网络分为两层，RSU 云层（RSU Cloud, RC），车辆云层（Vehicle Cloud, VC）。在 RC 中，RSU 通过有线网络连接形成计算云。VC 由特定区域的移动车辆形成云层，每个 VC 有一个管理员（VC Controller, VCC），负责管理云层的缓存资源，VCC 通过合适的算法进行选择。BS 和内容提供商（Content Provider, CP）通过回程链路连接到

互联网，内容提供商也可称为远程服务器，负责产生娱乐内容。基于用户的兴趣，请求的内容如果缓存在 VC 或者 RC 中，内容能够传输到请求车辆，取决于车辆的运动轨迹以及内容缓存的位置。

为了不失一般性，场景集中在一个基站的覆盖区域，道路依次排列 N_{RC} 个 RSU，采用 DSRC 接入技术与车辆通信，不同 RSU 的覆盖区域重叠。单个 RSU 的通信范围为 r ，RSU 和车辆均具备一定的缓存能力，单个 RSU 和车辆的缓存容量分别为 S^R ， S^V 。基站布置有网络管理控制器，负责收集车辆及 RSU 的信息做内容缓存决策^[68]。

5.1.2 车辆移动模型

在系统模型中，车辆采取基于排的运动模式^[69]，这是针对具有相同兴趣的车辆团体的驾驶模式，车辆群体的排头 VCC 的驾驶速度在 $[v_{min}, v_{max}]$ 之间^[70]，其他车辆能够调整速度，跟上 VCC 的速度并保持一定的车辆间隔^[71]，同时能够满足通信范围内的数据传输。车辆的平均速度为 $v = (v_{max} + v_{min})/2$ 。道路的两端分别为一个入口一个出口，进入入口 j ($j=1,2$) 的车辆数量为 $V_j(t)$ ，是时变的并服从参数为 λ_j 的泊松分布，参数代表车辆的平均到达率。

$$P(V_j(t) = k) = \frac{(\lambda_j t)^k}{k!} e^{-\lambda_j t}, k = 1, 2, \dots \quad (5.1)$$

其中 $V_j(t)$ 的期望为 $E[V_j(t)] = \lambda_j t$ ， $VC_j(t)$ 表示 VC_j 进入入口 j 的数量， N_{VC_j} 表示 VC_j 中车辆的数量，显然， $VC_j(t) = V_j(t)/N_{VC_j}$ ， μ_j 表示 $VC_j(t)$ 的期望见式 (5.2)。

$$\mu_j = E[VC_j(t)] = E\left[\frac{V_j(t)}{N_{VC_j}}\right] = \frac{\lambda_j}{N_{VC_j}} t, j = 1, 2 \quad (5.2)$$

5.1.3 内容分布与获取模型

假设内容库共有 L 个内容， $\{1, 2, \dots, L\}$ 表示所有内容的集合，每个内容的大小为 S_q ，内容库可不定时更新增加新的内容。考虑到一些受欢迎的内容会比其他内容请求频繁，假设内容的流行度服从齐普夫 (Zipf) 分布^[72]，内容 l 的请求概率见式 (5.3)。

$$p(l) = \frac{l^{-\gamma}}{\sum_{i=1}^L i^{-\gamma}} \quad (5.3)$$

其中 γ 为齐普夫分布参数。

对 VC_j 中的车辆 V_i ($1 \leq i \leq N_{VC_j}$) 和 RC 中的 RSU_m ($1 \leq m \leq N_{RC}$)，定义车辆及

RSU 的二进制缓存矩阵分别为 C^{VC_j} 和 C^{RC} 。

$$C^{VC_j} = [c_{i,l}^{VC_j}]^{N_{VC_j} \times L}, C^{RC} = [c_{m,l}^{RC}]^{N_{RC} \times L} \quad (5.4)$$

其中 $c_{i,l}^{VC_j}$, $c_{m,l}^{RC}$ 分别代表内容 l 在车辆 V_i 和 RSU m 的缓存放置指示因子, 见式 (5.5)。

$$c_{i,l}^{VC_j}, c_{m,l}^{RC} = \begin{cases} 1, & \text{内容 } l \text{ 缓存在 } VC_j \text{ 中的车辆 } V_i \text{ 或者 RSU}_m \\ 0, & \text{其他} \end{cases} \quad (5.5)$$

车辆请求内容 l 时, 首先会搜索自己的缓存空间, 如果没有找到该内容, 则通过 V2V 方式从 VC 中的其他成员获取。如果 VC 中没有, 则从通信范围内的 RSU 获取。如果最近的 RSU 也没有, RSU 之间有线连接可以互相协作, RC 中的其他成员协作传输内容。如果仍然没有, 请求车辆只能通过基站从内容提供商获取。每个请求车辆最终都能接收请求的内容。如果请求内容从自身或者 VC 或者 RC 获取, 则称为命中, 所以命中取决于请求内容与 VC 或者 RC 中缓存的内容是否匹配。当请求内容被 VC、通信范围内的 RSU、RC 成员和内容提供商满足时, 对应的数据传输速率为 $R_{VV}, R_{RV}, R_{RCV}, R_{PV}$, 同时会产生传输成本, 传输单位数据对应的价格为 $P_{VV}, P_{RV}, P_{RCV}, P_{PV}$ 。合理假设 $R_{VV} > R_{RV} > R_{RCV} > R_{PV}$, $P_{VV} < P_{RV} < P_{RCV} < P_{PV}$ 。如果车辆请求内容被自身满足, 则传输的时延和产生的成本为 0。

5.2 优化问题建模

5.2.1 平均内容传输时延

当 VC_j 成员进入入口 j 时, VC_j 中的车辆开始下载请求的内容 l , 定义 $\tau_{i,j,l}$ 为 VC_j 中的车辆 V_i 完整下载完内容 l 所需的时延, 整个 VC_j 下载内容 l 的时延定义为 $\tau_{j,l}$, 考虑到每个内容具有不同的请求概率, $\tau_{j,l}$ 的期望为

$$E[\tau_{j,l}] = \sum_{l=1}^L p(l) \frac{1}{N_{VC_j}} \sum_{i=1}^{N_{VC_j}} \tau_{i,j,l} \quad (5.6)$$

对于任意内容请求下载产生的平均时延定义为 τ_{av} , 由式 (5.7) 计算。

$$\tau_{av} = \sum_{j=1}^2 \frac{\mu_j E[\tau_{jl}]}{\mu_1 + \mu_2} \quad (5.7)$$

当车辆经过 RSU m 时, 需要根据请求内容所缓存的位置, 从不同的节点获取对应不同的传输速率, VC_j 中的车辆 V_i 经过 RSU m 的下载速率为 $R_{i,m,l,j}$, 在 VC_j 中, 传输速率为 $R_{i,m,l}^{VC_j}$, 由式 (5.8) 计算。

$$R_{i,m,l}^{VC_j} = \left(1 - c_{i,l}^{VC_j}\right) \left\langle \sum_{n=1, n \neq i}^{N_{VC_j}} c_{i,l}^{VC_j} \right\rangle_0 R_{VV} \left(c_{i,l}^{VC_j} \neq 1\right) \quad (5.8)$$

其中

$$\langle f(\cdot) \rangle_0 = \begin{cases} 0, & f(\cdot) = 0, \\ 1, & f(\cdot) > 0. \end{cases} \quad (5.9)$$

在式 (5.8) 中, $\left(1 - c_{i,l}^{VC_j}\right) \left\langle \sum_{n=1, n \neq i}^{N_{VC_j}} c_{i,l}^{VC_j} \right\rangle_0$ 表示请求的内容没有缓存在车辆 V_i 中, 但能够从 VC_j 中的其他成员获取。

当请求的内容无法在 VC_j 中满足时会将请求传输到 RC, RC 中, 传输速率 $R_{i,m,l}^{RC}$ 为

$$R_{i,m,l}^{RC} = c_{m,l}^{RC} R_{RV} + \left(1 - c_{m,l}^{RC}\right) \left\langle \sum_{k=1, k \neq m}^{N_{RC}} c_{m,l}^{RC} \right\rangle_0 R_{RCV} + \left(1 - \left\langle \sum_{k=1}^{N_{RC}} c_{m,l}^{RC} \right\rangle_0\right) R_{PV} \quad (5.10)$$

考虑 VC_j 和 RC 中的传输速率, 总的传输速率 $R_{i,m,l,j}$ 计算为

$$R_{i,m,l,j} = R_{i,m,l}^{VC_j} + \left(1 - \left\langle \sum_{n=1}^{N_{VC_j}} c_{i,l}^{VC_j} \right\rangle_0\right) R_{i,m,l}^{RC} \quad (5.11)$$

其中 $\left(1 - \left\langle \sum_{n=1}^{N_{VC_j}} c_{i,l}^{VC_j} \right\rangle_0\right)$ 表示在 VC_j 中未获取到请求内容。

车辆经过不同 RSU 根据请求内容缓存位置不同会有不同的传输速率, 通过不同的传输速率持续下载内容直至下载完整个内容, 定义 $\delta_{i,m,l}^{VC_j}$ 指示 VC_j 中的车辆 V_i 在 RSU_m 时是否完整获取内容 l 。

$$\delta_{i,m,l}^{VC_j} = \begin{cases} 1, & VC_j \text{ 中的车辆 } V_i \text{ 完整获取内容 } l, \\ 0, & \text{其他.} \end{cases} \quad (5.12)$$

当 VC_j 中的车辆 V_i 离开 RSU_m 的覆盖区域时, 已经累计下载内容 l 的数据量 $D_{i,m,l}^{VC_j}$ 为

$$D_{i,m,l}^{VC_1} = \sum_{1,2,\dots,m} R_{i,m,l,1} \frac{r}{v}, D_{i,m,l}^{VC_2} = \sum_{m,m+1,\dots,N_{RC}} R_{i,m,l,2} \frac{r}{v} \quad (5.13)$$

从入口 1 进入的车辆 i 完全下载完内容 l 所需的时延 $\tau_{i,1,l}$ 可以表示为

$$\tau_{i,1,l} = \begin{cases} 0, & c_{i,l}^{VC_1} = 1 \\ \frac{s_q}{R_{i,m,l,1}}, & \delta_{i,m,l}^{VC_1} = 1, m = 1, c_{i,l}^{VC_1} \neq 1 \\ \frac{r(m-1)}{v} + \frac{s_q - D_{i,m-1,l}^{VC_1}}{R_{i,m,l,1}}, & \delta_{i,m,l}^{VC_1} = 1, m \in \{2, 3, \dots, N_{RC}\}, c_{i,l}^{VC_1} \neq 1 \\ \frac{rN_{RC}}{v} + \frac{s_q - D_{i,N_{RC},l}^{VC_1}}{R_{i,m,l}^{VC_1} + \left(1 - \left\langle \sum_{n=1}^{N_{VC_1}} c_{i,l}^{VC_1} \right\rangle_0\right) R_{PV}}, & \forall m \in \{1, 2, 3, \dots, N_{RC}\}, \delta_{i,m,l}^{VC_1} = 0, c_{i,l}^{VC_1} \neq 1 \end{cases} \quad (5.14)$$

从入口 2 进入的车辆 i 下载内容 l 所需的时延 $\tau_{i,2,l}$ 可以表示为（访问 RSU 顺序不同）式 (5.15)。

$$\tau_{i,2,l} = \begin{cases} 0, & c_{i,l}^{VC_2}=1 \\ \frac{s_q}{R_{i,m,l,2}}, & \delta_{i,m,l}^{VC_2}=1, m=N_{RC}, c_{i,2}^{VC_1}\neq 1 \\ \frac{r(N_{RC}-m)}{v} + \frac{s_q-D_{i,m+1,l}^{VC_2}}{R_{i,m,l,2}}, & \delta_{i,m,l}^{VC_2}=1, m\in\{1,2,3,\dots,N_{RC}-1\}, c_{i,2}^{VC_1}\neq 1 \\ \frac{rN_{RC}}{v} + \frac{s_q-D_{i,1,l}^{VC_2}}{R_{i,m,l}^{VC_2} + \left(1 - \left(\sum_{n=1}^{N_{VC_2}} c_{i,l}^{VC_n}\right)_0\right) R_{PV}}, & \forall m\in\{1,2,3,\dots,N_{RC}\}, \delta_{i,m,l}^{VC_2}=0, c_{i,2}^{VC_1}\neq 1 \end{cases} \quad (5.15)$$

5.2.2 平均内容传输成本

对于任意内容请求下载产生的平均传输成本 $Cost_{av}$ 计算如式 (5.16)

$$Cost_{av} = \sum_{j=1}^2 \sum_{l=1}^L \sum_{i=1}^{N_{VC_j}} \frac{\mu_j}{\mu_1 + \mu_2} \frac{p(l)}{N_{VC_j}} Cost_{i,j,l} \quad (5.16)$$

当 VC_j 中的车辆 V_i 经过 RSU_m 下载内容 l 时，单位数据传输价格 $Cost_{i,m,l,j}$ 为

$$Cost_{i,m,l,j} = \left(1 - c_{i,l}^{VC_j}\right) \left\langle \sum_{n=1, n \neq i}^{N_{VC_j}} c_{i,l}^{VC_n} \right\rangle_0 P_{VV} + \left(1 - \left\langle \sum_{n=1}^{N_{VC_j}} c_{i,l}^{VC_n} \right\rangle_0 \right) [c_{m,l}^{RC} P_{RV} + (1 - c_{m,l}^{RC}) \left\langle \sum_{k=1, k \neq m}^{N_{RC}} c_{m,l}^{RC} \right\rangle_0 P_{RCV} + (1 - \left\langle \sum_{k=1}^{N_{RC}} c_{m,l}^{RC} \right\rangle_0) P_{PV}] \quad (5.17)$$

经过 RSU_m 下载内容 l 产生的传输成本 $Cost_{i,m,l}^{VC_j}$ 为

$$Cost_{i,m,l}^{VC_j} = \frac{r}{v} R_{i,m,l,j} Cost_{i,m,l,j} \quad (5.18)$$

当 VC_j 中的车辆 V_i 离开 RSU_m 的覆盖区域时，下载内容 l 累计产生的传输成本 $Cost_{i,m,l,j}^{sum}$ 为

$$Cost_{i,m,l,1}^{sum} = \sum_{1,2,\dots,m} Cost_{i,m,l}^{VC_1}, Cost_{i,m,l,2}^{sum} = \sum_{m,m+1,\dots,N_{RC}} Cost_{i,m,l}^{VC_2} \quad (5.19)$$

从入口 1 进入的车辆 i 完全下载完内容 l 所需的成本 $Cost_{i,1,l}$ 可以表示为

$$Cost_{i,1,l} = \begin{cases} S_q Cost_{i,m,l,1}, & \delta_{i,m,l}^{VC_1} = 1, m = 1 \\ Cost_{i,m-1,l,1}^{sum} + (S_q - D_{i,m-1,l}^{VC_1}) Cost_{i,m,l,1}, & \delta_{i,m,l}^{VC_1} = 1, m \in \{2, 3, \dots, N_{RC}\} \\ (1 - c_{i,l}^{VC_1}) \left\langle \sum_{n=1, n \neq i}^{N_{VC_1}} c_{i,l}^{VC_n} \right\rangle_0 P_{VV} + \left(1 - \left\langle \sum_{n=1}^{N_{VC_1}} c_{i,l}^{VC_n} \right\rangle_0\right) P_{PV}, & \forall m \in \{1, 2, 3, \dots, N_{RC}\}, \delta_{i,m,l}^{VC_1} = 0 \end{cases} \quad (5.20)$$

从入口 2 进入的车辆 i 完全下载完内容 l 所需的成本 $Cost_{i,2,l}$ 可以表示为

$$Cost_{i,2,l} =$$

$$\left\{ \begin{array}{ll} S_q Cost_{i,m,l,2}, & \delta_{i,m,l}^{VC_2} = 1, m = N_{RC} \\ Cost_{i,m+1,l,2}^{sum} + (S_q - D_{i,m+1,l}^{VC_2}) Cost_{i,m,l,2}, \delta_{i,m,l}^{VC_2} = 1, m \in \{1, 2, 3, \dots, N_{RC} - 1\} \\ Cost_{i,1,l,2}^{sum} + (S_q - D_{i,1,l}^{VC_2}) [& \forall m \in \{1, 2, 3, \dots, N_{RC}\}, \delta_{i,m,l}^{VC_2} = 0 \\ (1 - c_{i,l}^{VC_2}) \left(\sum_{n=1, n \neq l}^{N_{VC_2}} c_{i,n}^{VC_2} \right)_0 P_{VV} + \left(1 - \left(\sum_{n=1}^{N_{VC_2}} c_{i,n}^{VC_2} \right)_0 \right) P_{PV}], & \end{array} \right. \quad (5.21)$$

5.2.3 平均内容传输命中率

VC_j 中的车辆 V_i 完整下载完内容 l , 会经过不同的 RSU 根据缓存的位置以不同的下载速率进行下载, 定义经过 RSU 发生命中的次数与经过 RSU 的总数之比为 $hitratio_{i,l,j}$ 。对于任意内容请求, 平均内容传输命中率 $hitratio_{av}$ 计算如下

$$hitratio_{av} = \sum_{j=1}^2 \sum_{l=1}^L \sum_{i=1}^{N_{VC_j}} \frac{\mu_j}{\mu_1 + \mu_2} \frac{p(l)}{N_{VC_j}} hitratio_{i,l,j} \quad (5.22)$$

5.2.4 缓存优化问题建模

对于车载网络中的娱乐应用, 用户获取内容的时延以及运营商的传输成本至关重要。本章考虑充分利用车辆及 RSU 的缓存容量最小化时延与成本总和, 联合车辆及 RSU 缓存优化问题如下

$$\begin{aligned} P_C: \quad & \min \quad U(\mathbf{C}^{VC_1}, \mathbf{C}^{VC_2}, \mathbf{C}^{RC}) \\ & s.t. \\ & \sum_{l=1}^L c_{i,l}^{VC_1} \leq S^V, c_{i,l}^{VC_1} \in \{0, 1\}, i = 1, 2, \dots, N_{VC_1}, \\ & \sum_{l=1}^L c_{i,l}^{VC_2} \leq S^V, c_{i,l}^{VC_2} \in \{0, 1\}, i = 1, 2, \dots, N_{VC_2}, \\ & \sum_{l=1}^L c_{m,l}^{RC} \leq S^R, c_{m,l}^{RC} \in \{0, 1\}, m = 1, 2, \dots, N_{RC}, \\ & l = 1, 2, \dots, L. \end{aligned} \quad (5.23)$$

$$U(\mathbf{C}^{VC_1}, \mathbf{C}^{VC_2}, \mathbf{C}^{RC}) = \alpha \frac{\tau_{av}}{\tau_{max}} + \beta \frac{Cost_{av}}{Cost_{max}} = \sum_{l=1}^L \sum_{j=1}^2 \sum_{i=1}^{N_{VC_j}} \frac{\mu_j}{\mu_1 + \mu_2} \frac{p(l)}{N_{VC_j}} \left(\alpha \frac{\tau_{i,j,l}}{\tau_{max}} + \beta \frac{Cost_{i,j,l}}{Cost_{max}} \right) \quad (5.24)$$

其中 α , β 为时延和成本的权重参数, τ_{max} , $Cost_{max}$ 为获取内容所需最大容忍时延和传输成本, 在这种情况下, 内容均从通过基站从内容提供商获取, $\tau_{max} = S_q / R_{PV}$, $Cost_{max} = S_q P_{PV}$ 。式 (5.23) 中的约束为每个车辆和 RSU 的缓存内容不超过其本身的缓存能力。时延和成本的单位不同, $\frac{\tau_{av}}{\tau_{max}}$ 和 $\frac{Cost_{av}}{Cost_{max}}$ 用于归一化。

很明显, P_C 问题是 0-1 非线性整数规划 (Non-Linear Integer Programming, NLIP) 问题, 通常为 NP 难问题, 时延和成本的目标函数非常复杂。目标函数中由于有式 (5.9) 中阶跃函数的存在, 使得问题难以进行凸优化松弛。在 5.3 节中将提出有效的算法对优

化问题进行求解。

5.3 交替动态规划搜索算法

受交替凸搜索算法启发^[73,74], 交替动态规划搜索 (Alternate Dynamic Programming Search, ADPS) 算法用来求解问题 P_C , 将原问题分解为三个子问题, 使用动态规划 (Dynamic Programming, DP) 算法求解每个子问题, 在分解步骤, 每次固定其中两个缓存矩阵创建子问题, 简化的子问题如下

$$\begin{aligned} P_{C1}: \quad & \min \quad U(\mathbf{C}_{fix}^{VC_1}, \mathbf{C}_{fix}^{VC_2}, \mathbf{C}^{RC}) \\ & s.t. \sum_{l=1}^L c_{m,l}^{RC} \leq S^R, \quad c_{m,l}^{RC} \in \{0,1\}, \\ & \quad m = 1, 2, \dots, N_{RC}, \quad l = 1, 2, \dots, L, \end{aligned} \quad (5.25)$$

$$\begin{aligned} P_{C2}: \quad & \min \quad U(\mathbf{C}^{VC_1}, \mathbf{C}_{fix}^{VC_2}, \mathbf{C}^{RC}) \\ & s.t. \sum_{l=1}^L c_{i,l}^{VC_1} \leq S^V, \quad c_{i,l}^{VC_1} \in \{0,1\}, \\ & \quad i = 1, 2, \dots, N_{VC_1}, \quad l = 1, 2, \dots, L, \end{aligned} \quad (5.26)$$

$$\begin{aligned} P_{C3}: \quad & \min \quad U(\mathbf{C}_{fix}^{VC_1}, \mathbf{C}^{VC_2}, \mathbf{C}^{RC}) \\ & s.t. \sum_{l=1}^L c_{i,l}^{VC_2} \leq S^V, \quad c_{i,l}^{VC_2} \in \{0,1\}, \\ & \quad i = 1, 2, \dots, N_{VC_2}, \quad l = 1, 2, \dots, L. \end{aligned} \quad (5.27)$$

问题 P_{C1} , P_{C2} , P_{C3} 仍然为 NLIP 问题并具有类似的结构, 在 5.3.1 节使用最优动态规划算法分别求解每个子问题。

5.3.1 动态规划算法

动态规划算法是处理整数变量问题的一种有效算法, 通过把原整数问题划分为多阶段决策问题然后分阶段解决问题。每一个阶段关联多个状态, 每个阶段的最优决策基于前一阶段的最优决策以及阶段间的关系。通过第一阶段的最优决策以及阶段间的递归关系, 每一阶段都能做出最优决策, 到最后一阶段的问题构建等同于原问题, 这样可以找到原问题的最优解。下面以求解 P_{C1} 问题为例说明动态规划算法的关键步骤, 这些步骤包括重写问题 P_{C1} 为多阶段决策问题及找到相邻阶段间的递归关系。动态规划算法见表 5-1 (算法 5.1)。

1) 多阶段决策问题建模

将问题 P_{C1} 分解为 L 个阶段, 且在第 l 阶段优化前 l 个内容的缓存方案。 P_{C1} 问题中只需要优化 RSU 的缓存内容方案。令 $\mathbf{s}_l = [s_{m,l}]^{L \times 1} (0 \leq s_{m,l} \leq S^R)$ 表示第 l 阶段的状态

表 5-1 动态规划算法求解问题 P_{C1} Table.5-1 The DP Algorithm to solve problem P_{C1} **算法 5.1:** 动态规划算法求解问题 P_{C1}

```

1:   初始化;
2:   for 所有  $s_1 \leq [S_m]^{N_{RC} \times 1}$  do
3:      $J_1(s_1) = U_1(\min\{1, s_1\});$ 
4:   end for
5:   for  $l=2$  to  $L$  do
6:     for 所有  $s_l \leq [S_m]^{N_{RC} \times 1}$  do
7:       将  $J_l(s_l)$  初始化为较大的正数
8:       for 所有可能的  $c_l^{RC}$  do
9:         if  $J_l(s_l) > U_l(c_l^{RC}) + J_{l-1}(s_{l-1})$  then
10:           $J_l(s_l) = U_l(c_l^{RC}) + J_{l-1}(s_{l-1});$ 
11:           $c_l^{RC*} = c_l^{RC}$ 
12:        end if
13:      end for
14:    end for
15:  end for
16:  return 最优缓存放置矩阵  $C^{RC*}$ 

```

向量，代表所有 RSU 用于存储前 l 个内容的存储容量。

在第 l 阶段的 s_l 状态，对于前 l 个内容的缓存放置问题为

$$\begin{aligned}
 P_{C1_l}: J_l(s_l) = \min_{\{c_n^{RC}, 1 \leq n \leq l\}} & U_n(c_n^{RC}) \\
 \text{s.t. } \sum_{m=1}^l c_{m,n}^{RC} \leq s_{m,l}, m = 1, 2, \dots, N_{RC}, & \\
 c_{m,n}^{RC} \in \{0, 1\}, \forall m, 1 \leq n \leq l. &
 \end{aligned} \tag{5.28}$$

其中 $c_n^{RC} = [c_{m,n}^{RC}]^{N_{RC} \times 1}$ 为内容 l 的缓存放置向量，且

$$U_n(c_n^{RC}) = \sum_{j=1}^2 \sum_{i=1}^{N_{VC_j}} \frac{\mu_j}{\mu_1 + \mu_2} \frac{p(n)}{N_{VC_j}} \left(\alpha \frac{\tau_{i,j,n}}{\tau_{max}} + \beta \frac{Cost_{i,j,n}}{Cost_{max}} \right) \tag{5.29}$$

对于最后一阶段的决策问题， $l = L$, $s_{n,L} = S^R$, $s_L = [S_m]^{N_{RC} \times 1} = [1S^R]^{N_{RC} \times 1}$, 问

题 P_{C1_L} 可以表示为式 (5.30)。

$$\begin{aligned}
 P_{C1_L}: J_L(\mathbf{s}_L) = \min_{\substack{\{c_n^{RC}, \\ 1 \leq n \leq L\}}} & \sum_{n=1}^L \sum_{j=1}^2 \sum_{i=1}^{N_{VC_j}} \frac{\mu_j}{\mu_1 + \mu_2} \frac{p(n)}{N_{VC_j}} \left(\alpha \frac{\tau_{i,j,n}}{\tau_{max}} + \beta \frac{Cost_{i,j,n}}{Cost_{max}} \right) \\
 \text{s.t. } & \sum_{n=1}^L c_{m,n}^{RC} \leq S_m, m = 1, 2, \dots, N_{RC}, \\
 & c_{m,n}^{RC} \in \{0, 1\}, \forall m, 1 \leq n \leq L.
 \end{aligned} \tag{5.30}$$

很显然，在第 L 阶段，问题 P_{C1_L} 与原问题 P_{C1} 相同，通过逐步求解问题 P_{C1_l} ，可以得到原问题 P_{C1} 的最优解。

2) 递归关系

在第一阶段，当 $s_{m,1} \geq 1$ 时，RSU_m的最优决策是缓存第一个内容，因此， $\mathbf{s}_1 = \min\{1, \mathbf{s}_1\}$ 。

$$J_1(\mathbf{s}_1) = U_1(\min\{1, \mathbf{s}_1\}) \tag{5.31}$$

接下来，第 l 阶段的最优决策基于 $l-1$ 阶段的最优决策，阶段间的递归关系见式 (5.32)，其示例如图 5-2 所示。分为 3 个阶段，阶段 1 到阶段 2 的路径部分忽略，阶

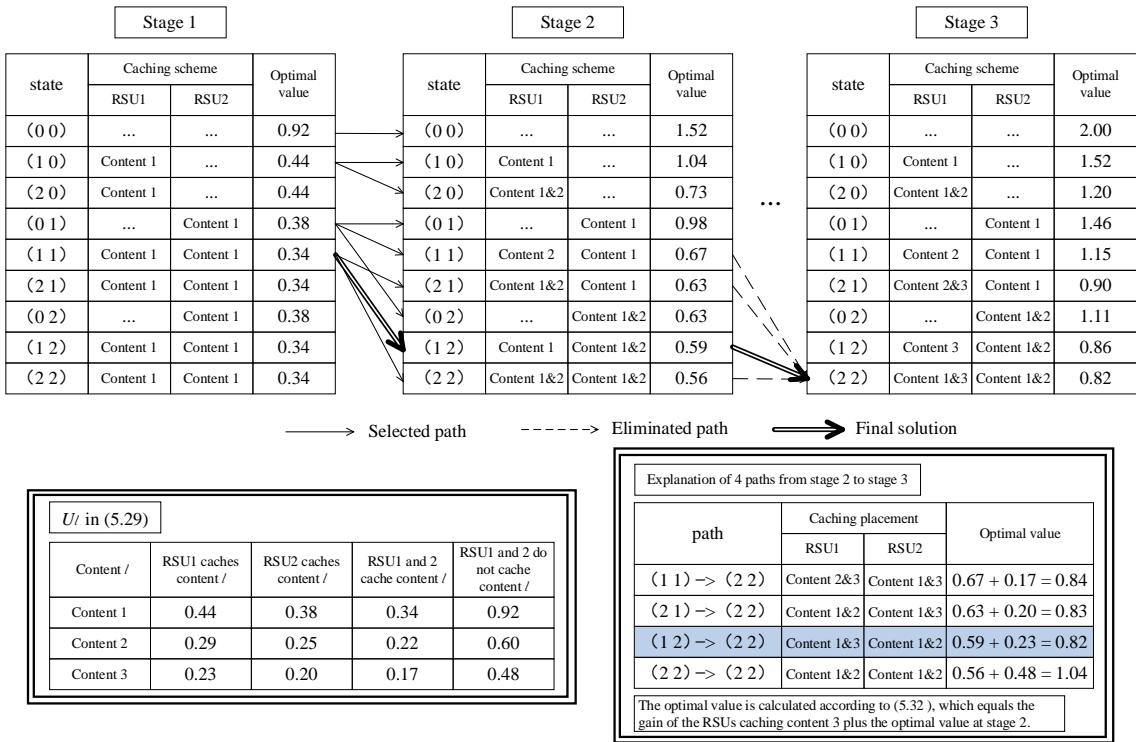


图 5-2 动态规划算法示例， $\mathbf{C}_{fix}^{VC_1} = \mathbf{0}$, $\mathbf{C}_{fix}^{VC_2} = \mathbf{0}$, $N_{RC} = 2$, $S^R = 2$, $L = 3$

Fig.5-2 An illustration of the proposed DP algorithm with $\mathbf{C}_{fix}^{VC_1} = \mathbf{0}$, $\mathbf{C}_{fix}^{VC_2} = \mathbf{0}$, $N_{RC} = 2$, $S^R = 2$,

$$L = 3$$

段 l 的状态代表缓存前 l 个内容的缓存能力，例如，阶段 2 的状态 (12) 代表 RSU1 缓存内容 1 或者 2，RSU2 缓存内容 1 和 2。最后的解法为 RSU1 和 RSU2 缓存内容 1，RSU2 缓存内容 2，最后 RSU1 缓存内容 3。

$$\begin{aligned} J_l(\mathbf{s}_l) = \min_{\{c_l^{RC}\}} \quad & U_l(c_l^{RC}) + J_{l-1}(\mathbf{s}_{l-1}) \\ \text{s. t. } & c_{m,l}^{RC} \in \{0,1\}, \\ & c_{m,l}^{RC} \leq s_{m,l}, \\ & m = 1, 2, \dots, N_{RC} \end{aligned} \quad (5.32)$$

其中 $\mathbf{s}_{l-1} = \mathbf{s}_l - \mathbf{c}_l^{RC}$ 是相邻阶段间的状态转移方程。

具体来说，在第 l 阶段的 \mathbf{s}_l 状态，最优缓存决策 \mathbf{c}_l^{RC} 通过搜索内容 l 的所有缓存放置状态，找到使得 $U_l(c_l^{RC}) + J_{l-1}(\mathbf{s}_{l-1})$ 最小的方案， $J_l(\mathbf{s}_{l-1})$ 是通过 $l-1$ 阶段得到。因此，据 $J_1(\mathbf{s}_1)$ 和递归关系式 (5.32)，可以找到问题 P_{C1} 的最优解。

3) 复杂度分析

根据算法 5.1，阶段 1 的计算复杂度（行 2-4）为 $O((S^R + 1)^{N_{RC}})$ ，对于阶段 l ($1 < l \leq L$)，每个 RSU 有 $S^R + 1$ 个状态，每个状态需要决定是否缓存内容 l 除了 0 状态，一个 RSU 的计算复杂度为 $O(2S^R + 1)$ ， N_{RC} 个 RSU 的计算复杂度（行 6-14）为 $O((2S^R + 1)^{N_{RC}})$ ，所以整体复杂度为 $O((S^R + 1)^{N_{RC}}) + O((L - 1)(2S^R + 1)^{N_{RC}})$ ，也可以表示为 $O((L - 1)(2S^R + 1)^{N_{RC}})$ 。

5.3.2 交替动态规划搜索算法

问题 P_{C1} , P_{C2} , P_{C3} 能够分别使用动态规划算法求解，受交替凸搜索算法启发提出交替动态规划搜索方法，见表 5-2 (算法 5.2)。该算法求解后的结果称为交替动态规划搜索缓存方案。

交替动态规划搜索算法依次解决问题 P_C 通过固定其中两个缓存矩阵变量，迭代的过程直至目标函数收敛，需要注意到不同的初始点可能导致不同的收敛结果。设置 $\mathbf{C}_0^{VC_1}$, $\mathbf{C}_0^{VC_2}$ 和 \mathbf{C}_0^{RC} 为零矩阵，即车辆和 RSU 不缓存任何内容，作为初始点，算法的性能在 I_{max} 迭代次数后稳定。给定所有参数后，算法在 $I_{max} = 2$ 时稳定。

复杂度分析：根据算法 5.2，因为动态规划算法的复杂度在上一节已经分析过，所以使用动态规划算法（行 4-6）的复杂度分别为 $O((L - 1)(2S^R + 1)^{N_{RC}})$, $O((L -$

$1)(2S^V + 1)^{N_{VC1}}$ 和 $O((L - 1)(2S^V + 1)^{N_{VC2}})$ 。整个算法的计算复杂度为 $O(I_{max}(L - 1)((2S^R + 1)^{N_{RC}} + (2S^V + 1)^{N_{VC1}} + (2S^V + 1)^{N_{VC2}}))$, 该算法随车辆和 RSU 的数量呈指数增长。

表 5-2 交替动态规划算法求解问题 P_C Table.5-2 The ADPS algorithm to solve problem P_C **算法 5.2:** 交替动态规划算法求解问题 P_C 输入: $Acc, Maxiter, \mathbf{C}_0^{VC_1}, \mathbf{C}_0^{VC_2}, \mathbf{C}_0^{RC}$ 输出: $\mathbf{C}_j^{VC_1}, \mathbf{C}_j^{VC_2}, \mathbf{C}_j^{RC}$

```

1:       $i = 1$ 
2:       $\Delta = \infty$ 
3:      while ( $\Delta > Acc$ )  $\wedge (i < Maxiter)$  do
4:           $\mathbf{C}_i^{RC} = \operatorname{argmin}_{\mathbf{C}^{RC}} U(\mathbf{C}_{i-1}^{VC_1}, \mathbf{C}_{i-1}^{VC_2}, \mathbf{C}_i^{RC})$  [使用 DP 算法求解问题  $P_{C1}$ ]
5:           $\mathbf{C}_i^{VC_1} = \operatorname{argmin}_{\mathbf{C}^{VC_1}} U(\mathbf{C}_i^{VC_1}, \mathbf{C}_{i-1}^{VC_2}, \mathbf{C}_i^{RC})$  [使用 DP 算法求解问题  $P_{C2}$ ]
6:           $\mathbf{C}_i^{VC_2} = \operatorname{argmin}_{\mathbf{C}^{VC_2}} U(\mathbf{C}_i^{VC_1}, \mathbf{C}_i^{VC_2}, \mathbf{C}_i^{RC})$  [使用 DP 算法求解问题  $P_{C3}$ ]
7:           $\Delta = |U(\mathbf{C}_i^{VC_1}, \mathbf{C}_i^{VC_2}, \mathbf{C}_i^{RC}) - U(\mathbf{C}_{i-1}^{VC_1}, \mathbf{C}_{i-1}^{VC_2}, \mathbf{C}_{i-1}^{RC})|$ 
8:           $i = i + 1$ 
9:      end while
10:      $I_{max} = i$ ; \总迭代次数
11:      $j = \operatorname{argmin}_{n=1,2,\dots,i-1} U(\mathbf{C}_n^{VC_1}, \mathbf{C}_n^{VC_2}, \mathbf{C}_n^{RC})$ 
12:     return  $\mathbf{C}_j^{VC_1}, \mathbf{C}_j^{VC_2}, \mathbf{C}_j^{RC}$ 

```

5.4 低复杂度合作贪婪算法

由于交替动态规划搜索算法复杂度随节点数量指数增长。对于大规模问题, 提出更为实际的合作贪婪 (Cooperation-Based Greedy, CBG) 算法来求解原问题。通过车辆请求获取内容的流程, 车辆请求的内容能够被 VC 或者 RC 成员满足。VC 或者 RC 中只要其中一个成员缓存该内容, 那么请求车辆就能够获取该内容。VC 中一个成员缓存

了该内容，RC 就不需要重复缓存该内容。请求内容的流程是先从 VC 中的车辆缓存中检索，所以，VC 中缓存该内容后，RC 中无需再进行重复缓存，这样有限的缓存空间能够被充分利用起来，同时还能提高缓存命中率。

表 5-3 交替动态规划搜索缓存方案， $L = 25$, $S^V = 1$, $S^R = 3$, $S_q = 150$, $\gamma = 0.6$, $N_{VC_1} = 5$,

$$N_{VC_2} = 6$$

Table 5-3 ADPS caching scheme with $L = 25$, $S^V = 1$, $S^R = 3$, $S_q = 150$, $\gamma = 0.6$, $N_{VC_1} = 5$ and

$$N_{VC_2} = 6$$

Caching matrix	$N_{RC} = 2$	$N_{RC} = 3$	$N_{RC} = 4$	$N_{RC} = 5$
$[\mathbf{C}^{VC_1}]^T$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0
$[\mathbf{C}^{VC_2}]^T$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ <hr/> 0
$[\mathbf{C}^{RC}]^T$	$\begin{bmatrix} 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ <hr/> 0	$\begin{bmatrix} 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ <hr/> 0
U	0.9134	0.7951	0.7217	0.6752

通过交替动态规划搜索算法在小规模场景求解的缓存方案结果见表 5-3，参数设置 $L = 25$, $S^V = 1$, $S^R = 3$, $S_q = 150$, $\gamma = 0.6$, $N_{VC_1} = 5$ 和 $N_{VC_2} = 6$ ，其他参数设置和

5.5 节中一样。表 5.1 中所有矩阵的行代表内容 ID，**0**代表零矩阵。从结果可以看出，

表 5-4 合作贪婪缓存算法求解问题 P_C Table.5-4 The cooperation-based greedy algorithm to solve problem P_C **算法 5.4:** 合作贪婪缓存算法求解问题 P_C 输入: $\mathcal{C}_0^{VC_1}, \mathcal{C}_0^{VC_2}, \mathcal{C}_0^{RC}$ 输出: $\mathcal{C}_{l_{max}-1}^{VC_1*}, \mathcal{C}_{l_{max}-1}^{VC_2*}, \mathcal{C}_{l_{max}-1}^{RC*}$

- 1: 所有车辆和 RSU 的最大缓存能力 $l_{max} = S^V * max\{N_{VC_1}, N_{VC_2}\} + S^R * N_{RC}$
- 2: **for** $l=1$ to l_{max} **do**
- 3: 按顺序依次从 VC_1, VC_2, RC 中选择具有缓存空间的车辆 $i, j, RSU k$ 。
- 4: 为了 VC 与 RC 层间进行合作，车辆中如果缓存有该内容，则无需缓存在 RSU 中，如果没有车辆缓存该内容，则需要缓存在 RSU 中。
- 5: 考虑所有层间合作情况，求解式 (5.24)，得到最优值 U 对应的缓存方案 $\mathcal{C}_{l-1}^{VC_1*}, \mathcal{C}_{l-1}^{VC_2*}, \mathcal{C}_{l-1}^{RC*}$ 。
- 6: **end for**
- 12: **return** $\mathcal{C}_{l_{max}-1}^{VC_1*}, \mathcal{C}_{l_{max}-1}^{VC_2*}, \mathcal{C}_{l_{max}-1}^{RC*}$

VC1	VC2	RC		
V1, V2...VN _{VC1}	V1, V2...VN _{VC2}	R1, R2...RN _{RC}		
Select vehicle i with extra cache space (regardless of vehicle number sequence)	Select vehicle j with extra cache space (regardless of vehicle number sequence)	Select RSU k with extra cache space (from R1 to RN _{RC})		
Selected vehicles and RSU	Caching scheme (for content l)		Optimal value and scheme	
Cooperating cache for content l (stage l)	V_i	V_j	R_k	Choose the optimal caching scheme for content l
...	...	Content l	...	
Content l	Content l	
Content l	No cache space	
No cache space	Content l	

图 5-3 算法 5.4 中行 3-5 的详细过程

Fig.5-3 Conceptual illustration of line 3-5 in Algorithm 5.4

随着 RSU 的数量 N_{RC} 增加，缓存在 VC 中的内容几乎不变，RSU 中的内容随 RSU 增加而增加。VC 和 RC 开始从最流行的内容缓存，VC 中的成员缓存不同的内容，RC 与 VC 中也缓存不同的内容，这种缓存方案能够充分利用不同节点的缓存空间，请求车辆能够从边缘节点获取更多的内容。受结果启发以及以上分析，为了减少算法的复杂度，适应大规模场景，提出合作贪婪算法。利用 VC 和 RC 层内成员互相合作，层间互相合作，贪婪体现在考虑层间合作的不同情况，选取最优目标函数值对应的缓存情况。合作贪婪算法见表 5-4（算法 5.4），算法 5.4 中的行 3-5 详细过程见图 5-3。该算法求解的结果称为合作贪婪缓存方案。

复杂度分析：合作贪婪算法的计算复杂度为 $O\left(2(S^V * \max\{N_{VC_1}, N_{VC_2}\} + S^R * N_{RC})\right)$ ，随节点数量线性增长。

5.5 数值仿真结果及分析

本节通过数值仿真评估不同缓存方案的性能。本章提出的两种缓存方案均由基站以集中式的方式执行，最优双层缓存放置指示因子由基站分配给每个 RSU 和 VCC，VCC 分配给每辆车。本次仿真中，入口 1/2 的车流量为 [40,36]/min，每个内容的大小为 150Mb，每个 RSU 的覆盖范围为 200m，车辆的平均速度为 20m/s，从车辆/RSU/RC 成员/内容提供商传输内容到目标车辆的传输速率分别为 [8,6,4,2]Mbps，相对应的传输单位数据的成本价格为 [1,4,6,10]，除此外，设定时延和成本的权重相等，即 $\alpha = \beta = 1$ 。

将所提的算法 5.2 和 5.3 求解的结果即两种缓存方案分别标记为 ADPS 和 CBG 缓存方案。此外，还将所提的缓存方案与以下缓存方案进行比较。

- (1) 基于流行度的缓存方案 (Popularity-Based Caching Scheme, PoBCS): 每个节点从最流行的内容开始缓存直到存储空间填满。
- (2) 基于流行度的概率缓存方案 (Probability-Based Caching Scheme, PrBCS): 每个节点缓存内容基于概率，概率与流行度相关，流行度概率越大，越可能被缓存。
- (3) 随机缓存方案 (Random-Based Caching Scheme, RBCS): 每个节点随机决定缓存内容，与流行度或其他参数无关。
- (4) 无缓存方案 (No Caching): 每个节点不缓存任何内容。
- (5) 低时延缓存方案 (Low Latency Caching Placement, LLCP): 文献[44]提出的最小化时延缓存方案，采用模拟退火 (Simulated Anneal, SA) 算法，该算法能够用来

求解问题 P_C 。算法流程见表 5-5 (算法 5.5)。该算法确定初始温度 T_{max} , 终止温度 T_{min} , 每次温度下的迭代次数 K_{in} 和冷却系数 θ 。在每次迭代中, m ($m = 1, 2, \dots, N_{RC}$) 随机选择。换句话说, 改变RSU_m的缓存内容且满足问题 P_C 的约束, 计算得到 \mathbf{C}^{old} 和 \mathbf{C}^{new} 对应的性能差 (行 4-5), \mathbf{C}^{new} 会以一定的概率 $Pe = exp(-\delta/T)$ 来决定是接受还是拒绝。经过 K_{in} 次迭代后, 当前温度 T 降至 θT , 下一次 K_{in} 次迭代开始。当 $T < T_{min}$ 时算法终止, \mathbf{C}^{new} 为最终 RSU 的缓存方案。需要注意的是, 算法性能取决于初始值, 算法的总迭代次数为 $\frac{T_{max}-T_{min}}{\theta} \times K_{in}$ 。

表 5-5 模拟退火算法求解问题 P_C Table 5-5 The simulated anneal algorithm to solve Problem P_C

算法 5.5: 模拟退火算法求解问题 P_C

输入: $T_{max}, T_{min}, K_{in}, \theta, \mathbf{C}_0^{VC_1}, \mathbf{C}_0^{VC_2}, \mathbf{C}_0^{RC}$

输出: \mathbf{C}^{new}

初始化 $T = T_{max}; \mathbf{C}^{old} = \mathbf{C}_0^{RC}$, 求解式 (5.24) 得到 $U^{old} = U(\mathbf{C}_0^{VC_1}, \mathbf{C}_0^{VC_2}, \mathbf{C}^{old})$

- 1: **while** $T \geq T_{min}$ **do**
- 2: **for** $k = 1: K_{in}$ **do**
- 3: 从 1 到 N_{RC} 随机选择 m
- 4: 改变 \mathbf{C}^{old} 中第 m 行的元素得到新的缓存放置矩阵 \mathbf{C}^{new} , $U^{new} = U(\mathbf{C}_0^{VC_1}, \mathbf{C}_0^{VC_2}, \mathbf{C}^{new})$
- 5: $\delta = U^{new} - U^{old}$
- 6: **if** $\delta > 0$
- 7: $Pe = exp(-\delta/T), A = rand(1)$
- 8: **if** $A > Pe$
- 9: $\mathbf{C}^{new} = \mathbf{C}^{old}, U^{new} = U^{old}$
- 10: **end if**
- 11: **end if**
- 12: $\mathbf{C}^{old} = \mathbf{C}^{new}, U^{old} = U^{new}$
- 13: **end for**
- 14: $T = \theta T$
- 15: **end while**

5.5.1 合作贪婪算法的有效性

本节比较所有缓存方案的时延、成本、命中率和计算时间。由于 ADPS 缓存方案复杂度高, 考虑小规模场景, 参数设置 $L = 25, S^V = 1, S^R = 3, \gamma = 0.85, N_{VC_1} = 5$ 和

$N_{VC_2} = 6$, 其他参数设置和前面一样。

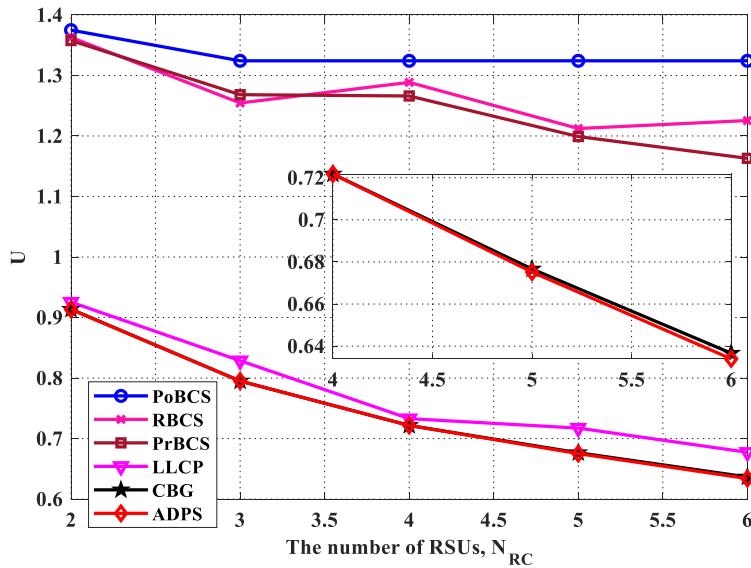


图 5-4 U 随 RSU 数量变化, $L = 25, S^V = 1, S^R = 3, \gamma = 0.85, S_q = 150, N_{VC_1} = 5$ 和 $N_{VC_2} = 6$

Fig.5-4 U versus the number of RSUs with $L = 25, S^V = 1, S^R = 3, \gamma = 0.85, S_q = 150, N_{VC_1} =$

5 and $N_{VC_2} = 6$

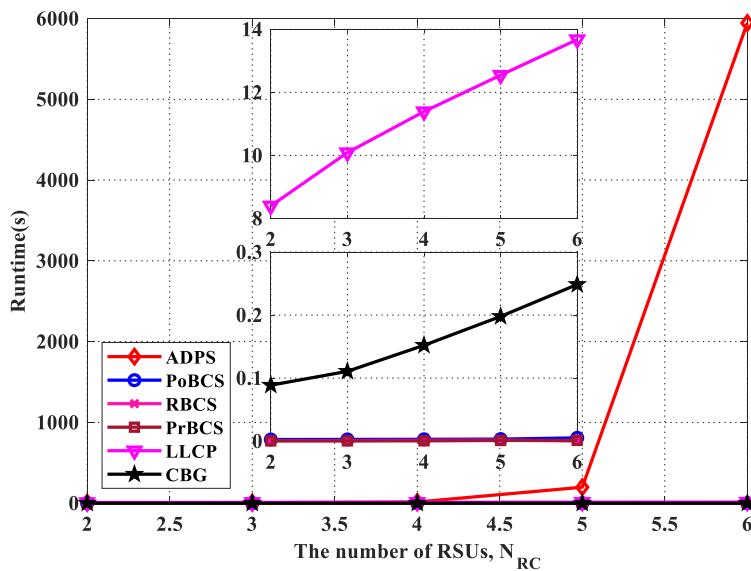


图 5-5 不同缓存方案随 RSU 数量变化的运行时间, $L = 25, S^V = 1, S^R = 3, \gamma = 0.85, S_q =$

150, $N_{VC_1} = 5$ 和 $N_{VC_2} = 6$

Fig.5-5 The runtime of caching schemes versus the number of RSUs with $L = 25, S^V = 1, S^R = 3, \gamma =$

0.85, $S_q = 150, N_{VC_1} = 5$ and $N_{VC_2} = 6$

如图 5-4 所示, 在不同缓存方案下, RSU 的数量对 U 的影响, 随 RSU 的数量增加, U 即时延与成本和减少除了 RBCS 和 PoBCS, 原因是网络总体缓存容量增加, 车

辆有更多机会获取感兴趣的内容。对 PoBCS，尽管总体缓存容量增加，因为每个 RSU 缓存有相同的流行内容，导致 RSU 之间的非合作性质，有限的空间缓存相同的内容导致从 RSU 端获取到的内容更少，更多的内容只能从内容提供商的远程服务器获取。对 RBCS，每个 RSU 随机决定，由于随机性，所以其性能不稳定，而 ADPS 和 CBG 缓存方案优于其他方案归结于层间的相互合作，并且能够在小场景下实现几乎相同的性能。

在图 5-5 中，展示了 RSU 数量对不同缓存方案运行时间的影响，图中的所有仿真结果基于处理器为 3.20 GHz Intel Core i7 和内存为 16GB 的 windows 计算机。可以看到，ADPS 缓存方案运行时间随 RSU 数量呈指数增长，CBG 和 LLCP 缓存方案随 RSU 数量呈线性增长，这与 5.3 节和 5.4 节中的计算复杂度分析相匹配。此外，综合图 5-4 和图 5-5，在实现几乎相同时延与成本和的性能下，CBG 缓存方案的运行时间远低于 ADPS 缓存方案。例如，当 $N_{RC} = 6$ 时，CBG 缓存方案的运行时间为 0.25s，而 ADPS 缓存方案的运行时间高达 5945s，这说明了 CBG 缓存方案的有效性与实用性，在接下来的大规模场景仿真中，仅将 CBG 缓存方案与其他缓存方案作为比较。

5.5.2 网络参设设置对缓存方案性能的影响

如图 5-6 所示，在不同缓存方案下，车辆和 RSU 的缓存能力对 U 的影响。随缓存容量 S^R 和 S^V 增加， U 减少，原因是总的缓存容量增加，更多的内容缓存在 VC 和 RC 成员中，具有更快的传输速率和更低的传输成本价格，传输时延和成本减少。用 U_τ 和 U_{cost}

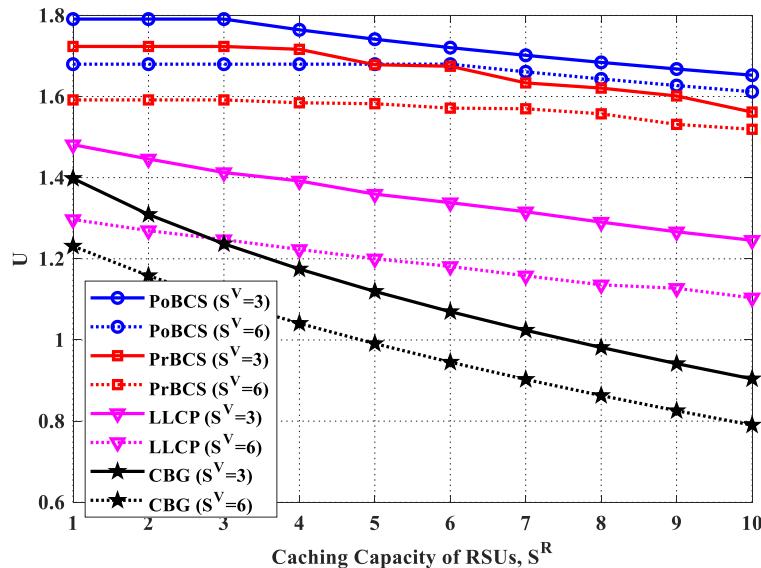
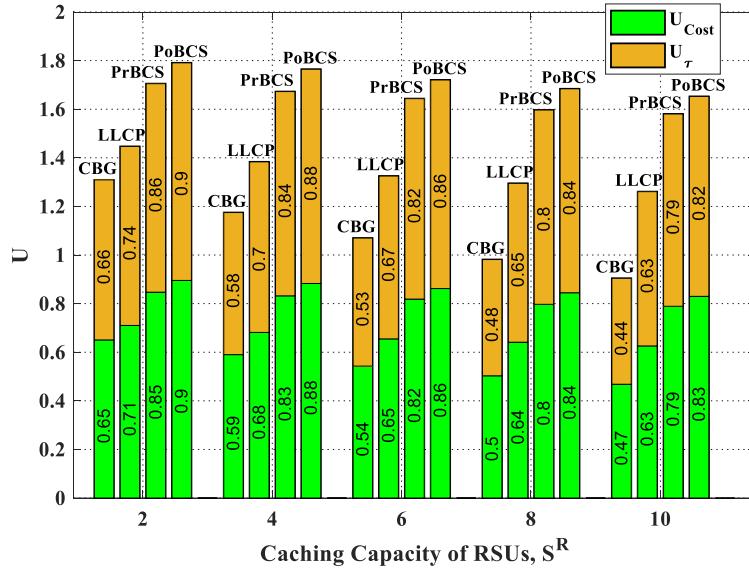
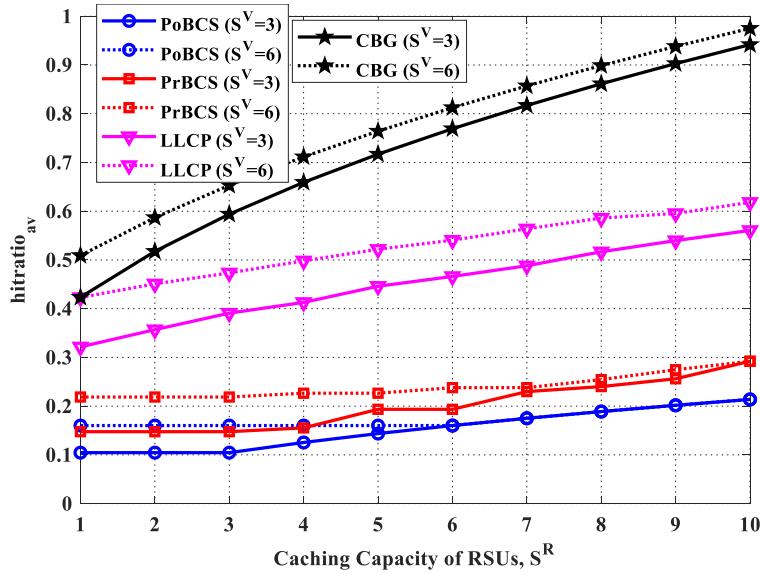


图 5-6 U 随缓存能力变化， $L = 250, \gamma = 0.6, S_q = 150, N_{VC_1} = 5, N_{VC_2} = 6$ 和 $N_{RC} = 20$

Fig.5-6 U versus caching capacity with $L = 250, \gamma = 0.6, S_q = 150, N_{VC_1} = 5, N_{VC_2} = 6$ and $N_{RC} = 20$

图 5-7 U 随缓存能力变化, $L = 250, \gamma = 0.6, S_q = 150, N_{VC_1} = 5, N_{VC_2} = 6$ 和 $N_{RC} = 20$ Fig.5-7 U versus caching capacity with $L = 250, \gamma = 0.6, S_q = 150, N_{VC_1} = 5, N_{VC_2} = 6$ and $N_{RC} = 20$ 图 5-8 平均命中率随缓存能力变化, $L = 250, \gamma = 0.6, S_q = 150, N_{VC_1} = 5, N_{VC_2} = 6$ 和 $N_{RC} = 20$ Fig.5-8 The average hit ratio versus caching capacity with $L = 250, \gamma = 0.6, S_q = 150, N_{VC_1} = 5, N_{VC_2} = 6$ and $N_{RC} = 20$

表示公式(5.24)中的第一部分 $\alpha \frac{\tau_{av}}{\tau_{max}}$ 和第二部分 $\beta \frac{cost_{av}}{cost_{max}}$, 即分别代表时延和成本部分,

其随缓存能力 S^R 的变化如图 5-7 所示。可以非常明显的看到时延和成本都是随 S^R 增加而减少, 导致总体 U 减少。除此之外, CBG 缓存方案优于其他方案。当 $S^R=10$ 时, 与 LLCP 方案相比, 平均传输时延和成本分别减少 30% 和 25%, 总体 U 减少 29%,

这主要归因于 CBG 缓存方案能够充分利用合作关系、缓存空间和内容流行度信息。

如图 5-8 所示，在不同缓存方案下，车辆和 RSU 的缓存能力对平均内容命中率的影响。正如预期那样，更大的缓存容量能够缓存更多内容，内容命中率增加，CBG 缓存方案具有 VC 与 RC 层内与层间合作的特性避免缓存相同的内容，更多的内容在有限的缓存空间缓存，因此提高了内容命中率。另一个有趣的现象需要注意到 PoBCS，在图 5-6 和图 5-8 中，当车辆的缓存能力 S^V 小于 RSU 的缓存能力 S^R ， U 和平均命中率保持不变，因为 PoBCS 缓存相同的流行内容导致 VC 和 RC 成员的非合作性，缓存在 RSU 中的内容是多余且无用的。当 RSU 的缓存能力超过车辆的缓存能力后，获取更多的内容主要取决于 RSU 中缓存的内容，所以时延成本和开始减少，命中率开始增加。CBG 缓存方案比其他方案在有限的缓存能力下能够实现更好的性能。

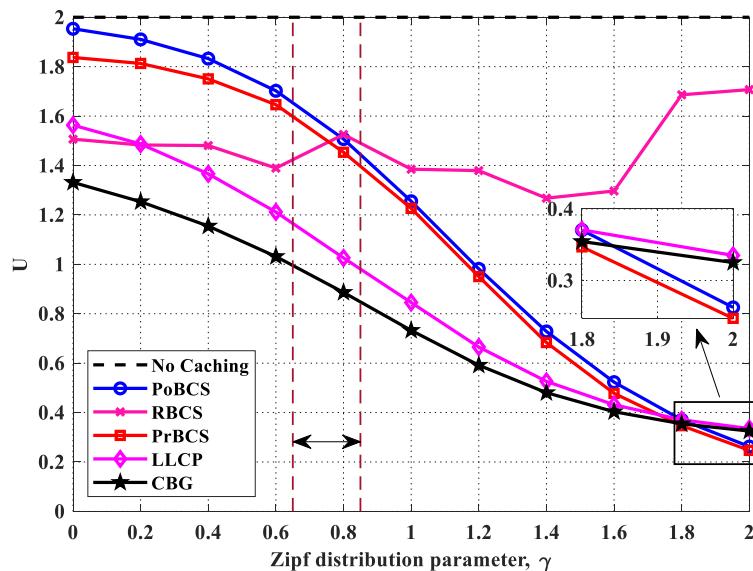


图 5-9 U 随 Zipf 参数变化， $L = 250, S_q = 150, S^V = 2, S^R = 8, N_{VC_1} = 5, N_{VC_2} = 6$ 和 $N_{RC} = 20$

Fig.5-9 U versus Zipf distribution parameter with $L = 250, S_q = 150, S^V = 2, S^R = 8, N_{VC_1} = 5, N_{VC_2} = 6$ and $N_{RC} = 20$

如图 5-9 所示，在不同缓存方案下， U 随 Zipf 参数变化的影响。 U 随 γ 增加而减少，除了 RBCS 和 No Caching 方案外，原因是当 γ 增加时对流行内容的请求占比会增加，车辆和 RSU 的缓存能够提供内容来满足对流行内容的频繁请求。No Caching 方案不考虑任何缓存技术，所有内容的请求通过基站从内容提供商的远程服务器获取，获取内容的时延和成本最高。对于 RBCS，由于每个车辆和 RSU 随机决定缓存内容，与其他

参数比如 γ 无关，性能变化不稳定，但比 No Caching 方案好，也能反映应用缓存技术带来的增益。然而，当 $\gamma \geq 1.8$ 时，PoBCS 和 PrBCS 表现比 CBG 缓存方案更好。原因是 γ 越大最流行的内容请求的概率增加，超过了其他所有内容请求概率的总和，这时都缓

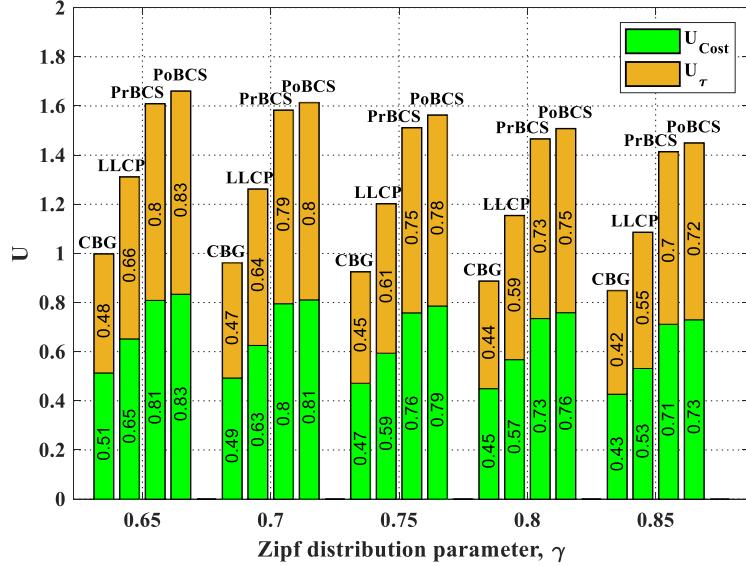


图 5-10 U 随 Zipf 参数变化， $L = 250, S_q = 150, S^V = 2, S^R = 8, N_{VC_1} = 5, N_{VC_2} = 6$ 和 $N_{RC} = 20$

Fig.5-10 U versus Zipf distribution parameter with $L = 250, S_q = 150, S^V = 2, S^R = 8, N_{VC_1} =$

$5, N_{VC_2} = 6$ and $N_{RC} = 20$

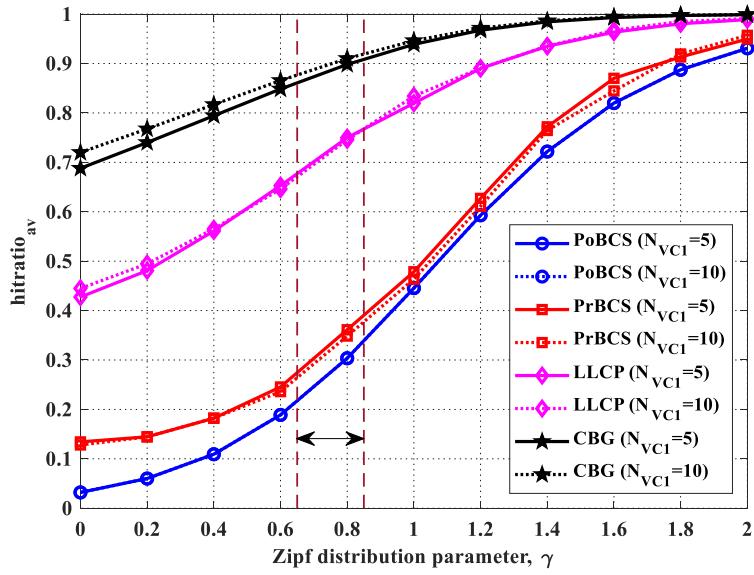


图 5-11 平均命中率随 Zipf 参数变化， $L = 250, S_q = 150, S^V = 2, S^R = 8, N_{VC_2} = 6$ 和 $N_{RC} = 20$

Fig.5-11 The average hit ratio versus Zipf distribution parameter with $L = 250, S_q = 150, S^V = 2, S^R =$

$8, N_{VC_2} = 6$ and $N_{RC} = 20$

存最流行的内容比如 PoBCS 和 PrBCS 会带来高的增益。

现有的研究[41][72]表明 γ 可能根据不同的网络情形从 0.65 到 0.85 变化, 图 5-10 展示了 $\gamma(0.65 \leq \gamma \leq 0.85)$ 对平均时延和成本在不同缓存方案下的实际影响。随着 γ 增加, 流行内容请求占比上升, 采用 CBG 缓存方案考虑内容流行度的变化, 请求车辆无需从其他高时延和高成本的地方获取到流行内容, 因此平均时延和成本降低, 总体 U 减少。图 5-10 中, CBG 缓存方案性能优于其他方案, 当 $\gamma = 0.85$ 时, 与 LLCP 方案相比, 平均时延和成本能够分别减少 24% 和 19%, 总体 U 减少 21%。

如图 5-11 所示, 展示了不同缓存方案下, γ 和 VC 中的车辆数量对平均内容命中率的影响。结果显示 CBG 缓存方案相比其他缓存方案, 能够达到最高的内容命中率。除此之外, 随 γ 增加, 所有方案的平均命中率增加。当 VC 层中的车辆数增加时, 换句话说 N_{VC_1} 增加, PoBCS 在 $N_{VC_1} = 5$ 和 $N_{VC_1} = 10$ 时, 平均内容命中率的曲线完全重合, 这也再一次说明了 PoBCS 的非合作性。对于 PrBCS 和 LLCP 方案, 其变化不明显。当 γ 从 0 到 1 变化时, CBG 缓存方案能够充分利用层内层间合作关系、存储能力、内容流行度等信息提高内容命中率。当 γ 从 1 到 2 时, 命中率的变化主要受请求概率的影响而不是 VC 层内车辆数量, 这与平均命中率的定义有关。在 $\gamma = 0.65 \sim 0.75$ 时, CBG 缓存方案的平均命中率能够达到 85% 以上。

5.6 测试床验证

本节设计了娱乐内容传输 (Infortainment Content Transmission, ICT) 系统, 基于该系统搭建并实现了双层系统架构原型。

5.6.1 系统原型实现

如图 5-12, 采用 Cohda Wireless 提供的 MK5 OBU 和 RSU 作为通信硬件, 阿里云服务器作为内容提供商, 配置 2.5 GHz CPU 和 4G 内存。车辆 1 (V1) 由 1.6 GHz CPU 和 4G 内存的笔记本 (Laptop1) 连接 MK5 OBU 组成。同理, 车辆 2 (V2) 由 1.6 GHz CPU 和 8G 内存的笔记本 (Laptop2) 连接 MK5 OBU 组成。V3 配置 2.5 GHz CPU 和 8G 内存。同时, Laptop4 配置 2.2 GHz 和 16G 内存连接两个 MK5 RSU。V1, V2, V3 能够与 RSU 和云服务器分别通过 DSRC 接口和 4G 热点进行通信。除此之外, 还准备了特定的视频内容数据集^[75], 来自 50 个好莱坞电影视频片段, 大小为 8968-9790KB。

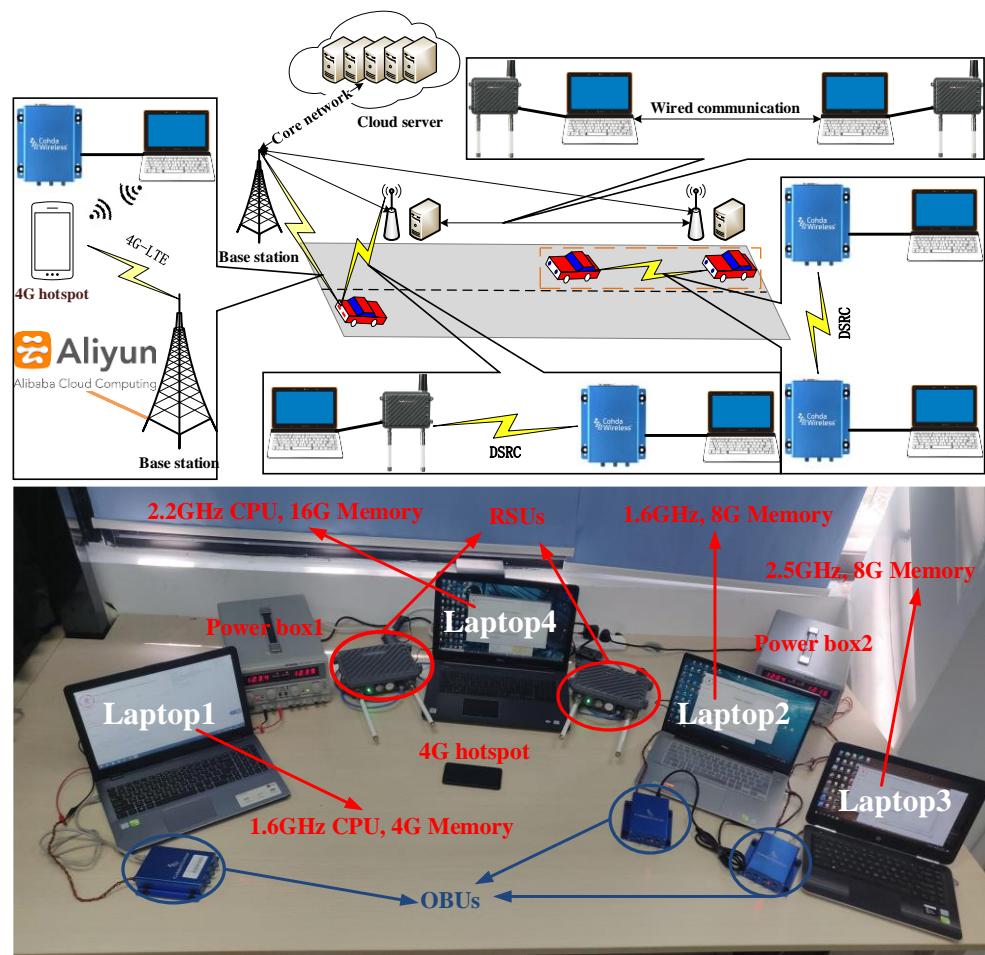


图 5-12 系统原型实现

Fig.5-12 Prototype implementation

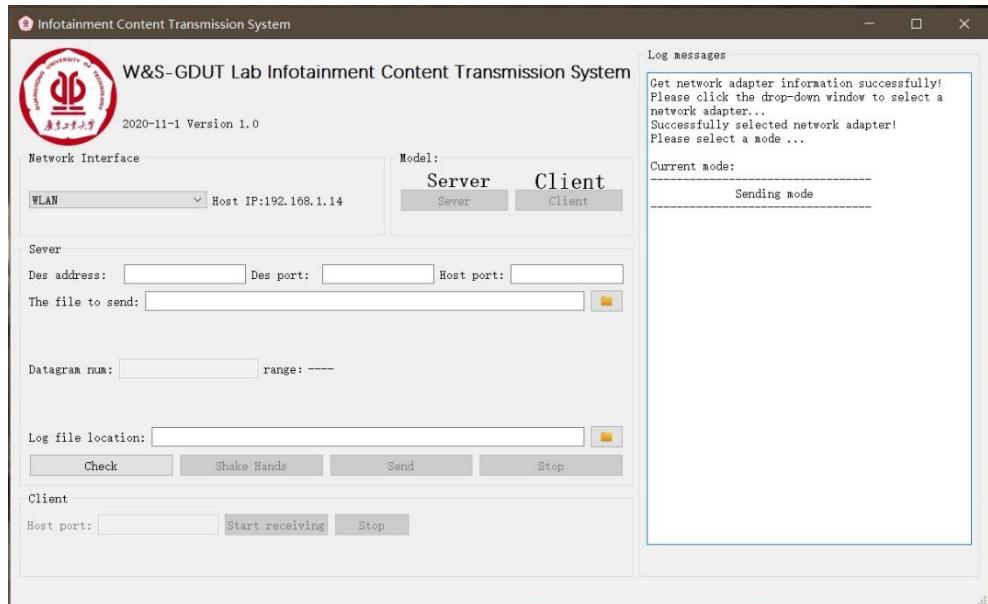


图 5-13 系统界面

Fig.5-13 System interface

娱乐内容传输系统界面如图 5-13，该系统分为三个模块，展示界面模块、内容处理模块和 UDP 通信模块，系统可以作为内容接收端（服务器端）和发送端（客户端）。该系统与 OBU，RSU，云服务器之间通过 UDP 协议通信，UDP 通信模块的程序逻辑如图 5-14。视频内容进行传输时，选择服务器模式，视频数据通过内容处理模块处理为 UDP 数据报文，进而传输到 DSRC 设备。DSRC 设备将 UDP 报文处理为基本安全消息(BSM)通过 DSRC 协议广播到目标车辆的 DSRC 设备，目标车辆选择客户端模式，接收并恢复视频文件，传输完成后展示界面会显示传输时延，传输速率，测试传输时延前会进行时间同步，一次内容的传输过程完成。

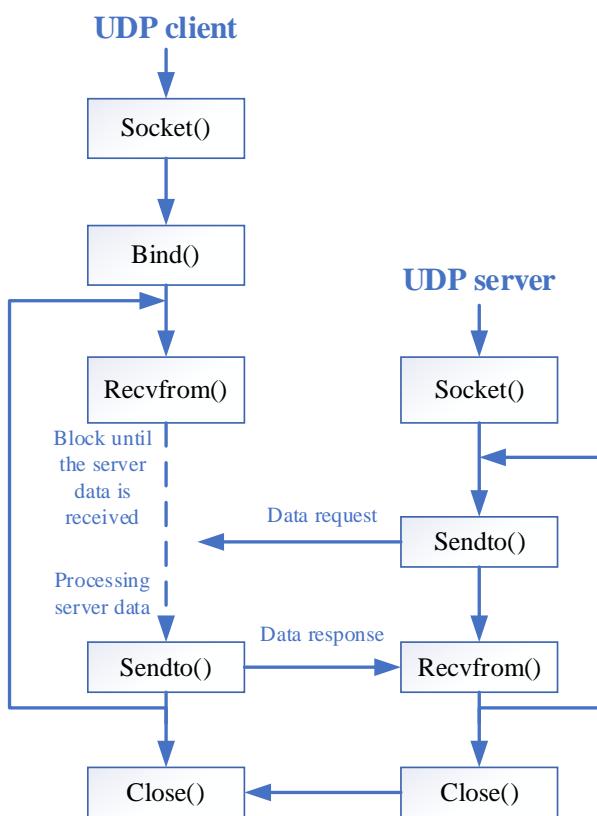


图 5-14 UDP 通信模块程序逻辑

Fig.5-14 Program logic of UDP communication module

5.6.2 传输速率测试与缓存方案验证

在实际测试中，测试了请求车辆从其他车辆、RSU、RSU 协作、云服务器获取不同视频内容的真实传输速率 R_{VV} , R_{RV} , R_{RCV} , R_{PV} 。如图 5-15 所示， R_{VV} 和 R_{RV} 非常接近，因为 MK5 RSU 和 MK5 OBU 的内部结构一样，其平均速率达到 336.3KB/s， R_{RCV} 的平均值约为 202.1KB/s， R_{PV} 的平均值约为 171.4KB/s。所有视频内容通过不同方式获取的

平均传输速率均符合之前的速率假设，通过实现系统原型得到实际的传输速率数据能够有效的验证所提缓存方案的有效性。

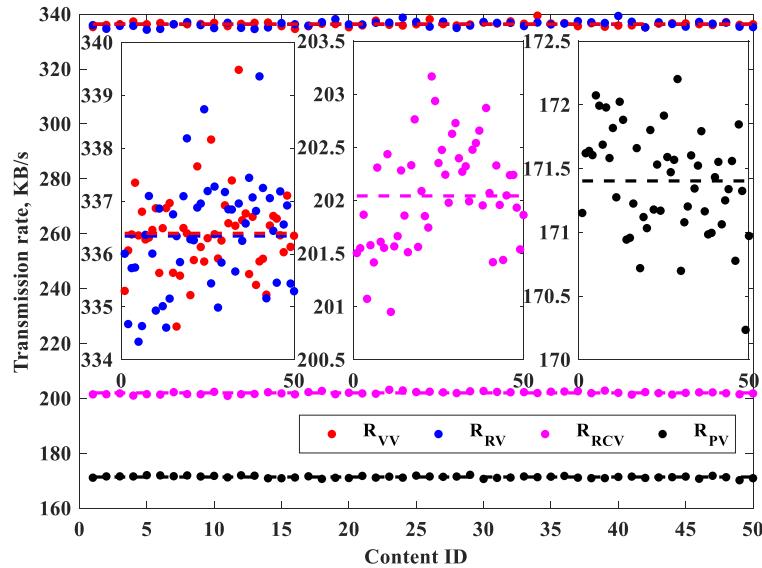


图 5-15 实际测试传输速率

Fig.5-15 Transmission rate under actual test

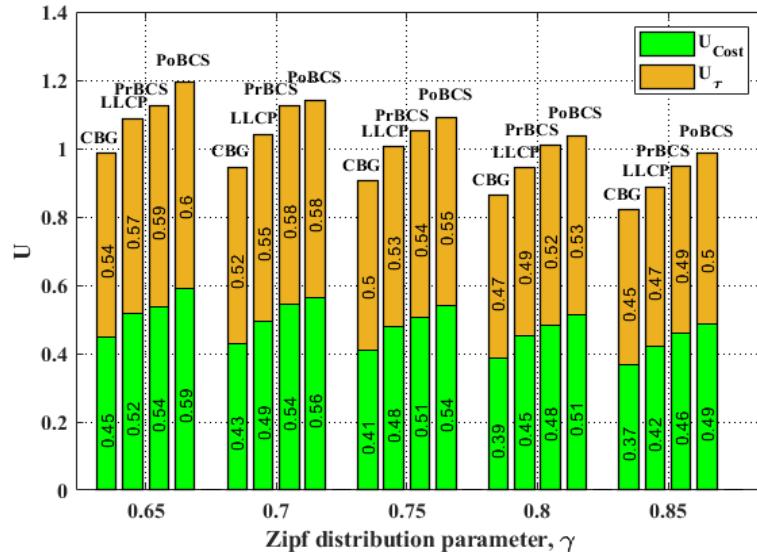


图 5-16 U 随 Zipf 参数基于真实数据集变化， $L = 50, S^V = 5, S^R = 10, N_{VC_1} = 1, N_{VC_2} = 2$ 和 $N_{RC} = 2$

Fig.5-16 U versus Zipf distribution parameter in real-world data set with $L = 50, S^V = 5, S^R = 10, N_{VC_1} = 1, N_{VC_2} = 2$ and $N_{RC} = 2$

基于实测的传输速率、视频内容大小，每个车辆在真实 RSU 的停留时间设为 10s 模拟车辆的移动。基于这些数据和设定，不同缓存方案在真实数据下随 γ 变化的性能如图 5-16 和图 5-17。可以直观的看到，随着 γ 增加，平均内容传输时延和成本降低，平均

内容命中率增加。在实测数据中采用 CBG 缓存方案，当 $\gamma = 0.85$ 时，与 PoBCS 相比，平均时延和成本分别减少 10% 和 24%。正如预期的那样，不同节点之间的合作在缓存方案设计中扮演重要角色，CBG 缓存方案在实测数据中的性能表现优于其他方案，也验证了提出的 CBG 缓存方案的有效性。

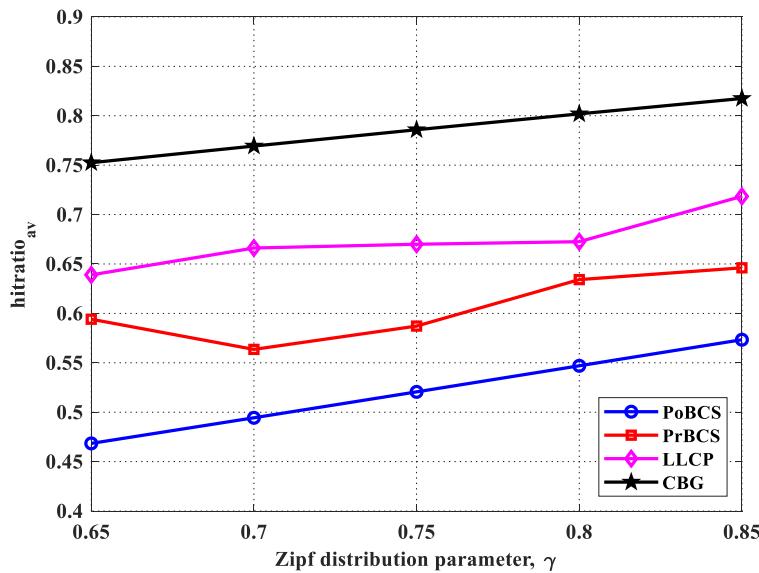


图 5-17 平均命中率随 γ 基于真实数据集变化， $L = 50, S^V = 5, S^R = 10, N_{VC_1} = 1, N_{VC_2} = 2$ 和 $N_{RC} = 2$

Fig.5-17 The average hit ratio versus Zipf distribution parameter in real-world data set with $L = 50, S^V = 5, S^R = 10, N_{VC_1} = 1, N_{VC_2} = 2$ and $N_{RC} = 2$

5.7 本章小结

本章将现有的单层缓存方案拓展为双层。基于双层车载网络架构分析并推导平均时延、成本、命中率表达式，提出联合车辆 RSU 缓存优化问题来最小化时延和成本，提出交替动态规划搜索算法和合作贪婪算法求解优化问题。经过数值仿真表明了提出缓存方案的优势，同时合作缓存方案的运行时间比动态规划搜索算法运行时间更短。除此之外，搭建系统原型并基于实测数据验证，验证了所提缓存方案的有效性。

总结与展望

总结

随着 5G、人工智能、大数据、云计算、区块链等新技术的出现，为汽车与交通行业变革带来强大支撑力，行业普遍认为，基于 C-V2X 的车联网很可能成为 5G 时代最先成功的垂直行业应用场景。未来人们出行时，基于车联网的自动驾驶会逐步适应极端、无法预测的场景，向 L5 级别的自动驾驶发展，智能网联汽车将会成为第二个移动终端。针对车联网通信面临假冒终端、数据丢弃或者销毁等安全风险，本文基于仿真数据提出智能检测系统和部署方案。车联网中车辆之间互联形成交通信息流，如何有效管理交通信息流的收集和分发，提高交通效率，本文从复杂网络的角度基于真实数据集进行分析，确定车载网络符合复杂网络的特性，提出分簇、簇头选择以及簇内数据传输的综合通信方案。针对移动互联网流量激增，智能网联汽车作为出行中的移动终端，为避免热门内容冗余下载，本文研究了双层车联网架构下的缓存放置策略优化问题。

具体工作如下：

(1) 分析国内外车联网数据安全、数据传输和缓存策略研究现状，比较主流的车联网技术标准，介绍本文采用的技术、理论和 DSRC 设备 MK5 及基于该设备进行的相关研究现状。

(2) 利用 MOVE、SUMO 和 NS2 仿真车辆运动通信，引入异常行为车辆，得到车辆数据交互的数据集，使用神经网络训练数据并达到不错的检测效果，同时提出基于模糊逻辑的检测系统部署方案，仿真结果表明这种部署方案能够有效检测出异常车辆。

(3) 从复杂网络理论的角度分析真实车辆 GPS 数据，验证了车载网络符合复杂网络特性，并基于无向图模型提出基于广义距离的分簇算法和基于模糊逻辑的簇头选择算法，同时提出簇内数据传输优化模型，提出的簇头选择方案能够有效提高簇内网络吞吐量并降低传输时延。

(4) 针对双层车联网架构缓存场景，基于车辆运动模型、内容分布模型及内容请求模型，推导平均内容传输时延和平均内容传输成本。提出的优化缓存放置策略旨在充分利用车辆和 RSU 的存储容量来最小化平均时延和成本。为了求解该问题，提出交

替动态规划搜索算法和低复杂度合作贪婪算法的缓存放置策略，此外基于 MK5 硬件平台开发娱乐内容传输系统，实现了系统原型。仿真结果和实测数据验证表明，所提的缓存方案能够降低传输时延和传输成本，同时提高内容命中率。

展望

本文对车联网入侵检测系统、数据传输和缓存策略开展了研究，取得了一定的进展，但仍有一定的改进和提升空间。

(1) 本文基于仿真，模拟了车联网中出现异常车辆造成数据丢弃的行为，仅代表一种异常行为，而真实车联网场景中的攻击类型种类繁多，出现新的攻击类型时，这种检测方式显然不能应对。通过蜜罐节点引诱新的攻击类型进行攻击并提取新的特征加入训练，能够对新的攻击类型起到检测效果。由于车联网的开放性，数据在传输过程中面临窃听、篡改等攻击，通过区块链的公私钥加密技术，能够提高数据传输的安全性。

(2) 在簇内数据传输优化模型中，簇内成员及簇头会随时间变化，虽然分簇及簇头选择算法尽力维持簇结构的稳定性，但如何适应动态环境下的数据稳定传输是研究的另一个重点。

(3) 在双层缓存结构模型中，缓存内容是定期更新的，内容流行度是动态变化的，推导时延和成本的前提是车辆在该假设下能够完全下载完请求内容，车辆请求下载的传输速率在实际情况下会随车辆移动变化，如何构建一个更加符合实际情况的模型并根据内容流行度变化实现动态缓存策略调节还需要进一步深入研究。

参考文献

- [1] 5GAA. V2X Functional and Performance Test Report[EB/OL]. https://5gaa.org/wp-content/uploads/2018/11/5GAA_P-190033_V2X-Functional-and-Performance-Test-Report_final-1.pdf.
- [2] 肖瑶, 刘会衡, 程晓红. 车联网关键技术及其发展趋势[J]. 通信技术与政策, 2020(8):18-21.
- [3] 葛雨明, 翁亚红. 车联网先导性应用进展研究[J]. 信息通信技术与政策, 2020(8):18-21.
- [4] Cisco. Cisco Annual Internet Report (2018–2023) White Paper[EB/OL]. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [5] 孙瑞锦. 基于携能和边缘缓存的无线通信系统传输技术研究[D]. 北京:北京邮电大学, 2019:8-13.
- [6] Alheeti K M A, Gruebler A, McDonald-Maier K. Intelligent intrusion detection of grey hole and rushing attacks in self-driving vehicular networks[J]. Computers, 2016, 5(3):16-16.
- [7] Alheeti K M A, Gruebler A, McDonald-Maier K. Using discriminant analysis to detect intrusions in external communication for self-driving vehicles[J]. Digital Communications and Networks, 2017, 3(3):180-187.
- [8] Zaidi K, Milojevic M B, Rakocevic V, et al. Host-based intrusion detection for VANETs: a statistical approach to rogue node detection[J]. IEEE Transactions on Vehicular Technology, 2016, 65(8):6703-6714.
- [9] Sharma S, Kaul A. A survey on intrusion detection systems and honeypot based proactive security mechanisms in VANETs and VANET cloud[J]. Vehicular Communications, 2018, 12(1):138-164.
- [10] Shu J, Zhou L, Zhang W, et al. Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach[J]. IEEE Transactions on Intelligent

- Transportation Systems, 2020, 3(1):1-12.
- [11] Liang J, Lin Q, Chen J, et al. A filter model based on hidden generalized mixture transition distribution model for intrusion detection system in vehicle ad hoc networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(7): 2707-2722.
- [12] 李宁宁. 基于机器学习的车联网入侵检测技术的研究与实现[D]. 成都:电子科技大学, 2019:20-25.
- [13] 甘子健. 车联网中入侵检测关键技术的研究[D]. 湘潭:湘潭大学, 2018:9-16.
- [14] Cooper C, Franklin D, Ros M, et al. A comparative survey of VANET clustering techniques[J]. IEEE Communications Surveys & Tutorials, 2017, 19(1):657-681.
- [15] 吕争, 尚光龙. FLCHS:面向车联网中的簇头产生算法[J]. 控制工程, 2017, 24(10):195-199.
- [16] Wu C, Yoshinaga T, Chen X, et al. Cluster-based content distribution integrating LTE and IEEE 802.11p with fuzzy logic and Q-learning[J]. IEEE Computational Intelligence Magazine, 2018, 13(1):41-50.
- [17] Wang J, Jiang C, Gao L, et al. Complex network theoretical analysis on information dissemination over vehicular networks[C]. IEEE International Conference on Communications (ICC). IEEE, Kuala Lumpur, Malaysia, 2016:1-6.
- [18] Zhang K, Wang J, Jiang C, et al. Content aided clustering and cluster head selection algorithms in vehicular networks[C]. IEEE Wireless Communications and Networking Conference (WCNC). IEEE, San Francisco, CA, USA, 2017:1-6.
- [19] 彭军, 马东, 刘凯阳, 等. 基于 LTE D2D 技术的车联网通信架构与数据分发策略研究[J]. 通信学报, 2016, 37(7):62-70.
- [20] 默罕莫德·默森, 许凯凯, 夏玮玮, 等. 荒漠场景应用的车联网及其分簇路由算法[J]. 通信学报, 2012, 33(10):166-174.
- [21] Alsuhihi G, Khattab A, Fahmy A. A mobility-based double-head clustering algorithm for dynamic VANET[C]. International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC). IEEE, Alexandria, Egypt, 2018:91-94.

- [22] Marzak B, Toumi H, Talea M, et al. Cluster head selection algorithm in vehicular ad hoc networks[C]. International Conference on Cloud Technologies & Applications. IEEE, Marrakech, Morocco, 2015:1-4.
- [23] Azizian M, Cherkaoui S, Hafid A. A distributed cluster based transmission scheduling in VANET[C]. IEEE International Conference on Communications (ICC). IEEE, Kuala Lumpur, Malaysia, 2016:1-6.
- [24] Shi C, Zhou Y, Li W, et al. A centralized clustering based hybrid vehicular networking architecture for safety data delivery[C]. IEEE Global Communications Conference (GLOBECOM). IEEE, Singapore, 2017:1-6.
- [25] Shen X, Cheng X, Yang L, et al. Data dissemination in VANETs: A scheduling approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5):2213-2223.
- [26] 薛拯, 刘洋, 韩国军, 等. 车联网下基于模糊逻辑的簇头选择与数据传输策略研究[J]. 小型微型计算机系统, 2020, 41(7):1458-1463.
- [27] Su Z, Dai M, Xu Q, et al. Q-learning-based spectrum access for content delivery in mobile networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(1):35-47.
- [28] Sun R, Wang Y, Ling L, et al. Delay-oriented caching strategies in D2D mobile networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(8):8529-8541.
- [29] Sun R, Wang Y, Cheng N, et al. QoE-driven transmission-aware cache placement and cooperative beamforming design in cloud-RANs[J]. IEEE Transactions on Vehicular Technology, 2020, 69(1):636-650.
- [30] Wang R, Zhang J, Song S H, et al. Mobility-aware caching in D2D networks[J]. IEEE Transactions on Wireless Communications, 2017, 16(8):5001-5015.
- [31] Wang R, Zhang J, Song S H, et al. Exploiting mobility in cache-assisted D2D networks: Performance analysis and optimization[J]. IEEE Transactions on Wireless Communications, 2018, 17(8):5592-5605.
- [32] Ma J, Liu L, Shang B, et al. Cache-aided cooperative Device-to-Device (D2D) networks:

- A stochastic geometry view[J]. IEEE Transactions on Communications, 2019, 67(11):7444-7455.
- [33] Ma J, Liu L, Song H, et al. Scalable video transmission in cache-aided device-to-device networks[J]. IEEE Transactions on Wireless Communications, 2020, 19(6):4247-4261.
- [34] Xu Q, Su Z, Wang Y, et al. Secure edge caching for layered multimedia contents in heterogeneous networks[C]. IEEE Global Communications Conference (GLOBECOM). IEEE, Waikoloa, HI, USA, 2019:1-6.
- [35] Deng T, Ahani G, Fan P, et al. Cost-optimal caching for D2D networks with user mobility: Modeling, analysis, and computational approaches[J]. IEEE Transactions on Wireless Communications, 2018, 17(5):3082-3094.
- [36] Ibrahim A M, Zewail A A, Yener A. Device-to-Device coded caching with distinct cache sizes[J]. IEEE Transactions on Communications, 2019, 68(5):2748-2762.
- [37] Lee M C, Molisch A F. Individual preference aware caching policy design in wireless D2D networks[J]. IEEE Transactions on Wireless Communications, 2020, 19(8):5589-5604.
- [38] Chen Y, Gong X, Ou R, et al. Crowd caching: Incentivizing D2D-enabled caching via coalitional game for IoT[J]. IEEE Internet of Things Journal, 2020, 7 (6):5599-5612.
- [39] Ding R, Wang T, Song L, et al. Roadside-unit caching in vehicular ad hoc networks for efficient popular content delivery[C]. IEEE Wireless Communications and Networking Conference (WCNC). IEEE, New Orleans, LA, USA, 2015:1207-1212.
- [40] Hu Z, Zheng Z, Wang T, et al. Roadside unit caching: Auction-based storage allocation for multiple content providers[J]. IEEE Transactions on Wireless Communications, 2017, 16(10):6321-6334.
- [41] Su Z, Hui Y, Xu Q, et al. An edge caching scheme to distribute content in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2018, 67(6):5346-5356.
- [42] Qin Z, Leng S, Zhou J, et al. Collaborative edge computing and caching in vehicular networks[C]. IEEE Wireless Communications and Networking Conference (WCNC). IEEE, Seoul, Korea, 2020:1-6.

- [43] Y Alnagar, Hosny S, El-Sherif A A. Towards mobility-aware proactive caching for vehicular ad hoc networks[C]. IEEE Wireless Communications and Networking Conference Workshop. IEEE, Marrakech, Morocco, 2019:1-6.
- [44] Ma J, Wang J, Liu G, et al. Low latency caching placement policy for cloud-based VANET with both vehicle caches and RSU caches[C]. IEEE Globecom Workshops (GC Wkshps). IEEE, Singapore, 2017:1-6.
- [45] Lu H, Lei L, Kan Z, et al. A Q-learning based proactive caching strategy for non-safety related services in vehicular networks[J]. IEEE Internet of Things Journal, 2019, 6(3):4512-4520.
- [46] 卢昇和. 基于 D2D 的车联网多接入边缘计算和缓存技术研究[D]. 广州:华南理工大学, 2020:33-45.
- [47] 于雯露. 基于能量损耗和时延最小化的车联网边缘缓存策略研究[D]. 哈尔滨:哈尔滨工业大学, 2020:21-49.
- [48] 张玲玲. 基于边缘计算的协作式车联网数据分发和缓存技术研究[D]. 广州:华南理工大学, 2020:14-49.
- [49] Hu B, Fang L, Cheng X, et al. In-Vehicle Caching (IV-Cache) via Dynamic Distributed Storage Relay (D^2SR) in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1):843-855.
- [50] Yao L, Wang Y, Wang X, et al. Cooperative caching in vehicular content centric network based on social attributes and mobility[J]. IEEE Transactions on Mobile Computing, 2021, 20(2): 391-402.
- [51] Deng T, Fan P, Yuan D. Optimizing retention-aware caching in vehicular networks[J]. IEEE Transactions on Communications, 2019, 67(9):6139-6152.
- [52] Zhang Y, Li C, Luan T H, et al. A mobility-aware vehicular caching scheme in content centric networks: Model and optimization[J]. IEEE Transactions on Vehicular Technology, 2019, 68(4):3100-3112.
- [53] Han X, Li X, Luo C, et al. Incentive mechanism with the caching strategy for content sharing in vehicular networks[C]. IEEE Globecom Workshops (GC Wkshps). IEEE,

- Waikoloa, HI, USA, 2019:1-6.
- [54] Yang H, Zhang K, Zheng K, et al. Secure edge caching placement and delivery for ultra-reliable and low-latency vehicular networks[C]. IEEE Globecom Workshops (GC Wkshps). IEEE, Waikoloa, HI, USA, 2019:1-6.
- [55] Chen J, Wu H, Yang P, et al. Cooperative edge caching with location-based and popular contents for vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(9):10291-10305.
- [56] Qiao G, Leng S, Maharjan S, et al. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks[J]. IEEE Internet of Things Journal, 2020, 7(1):247-257.
- [57] Liu K, Xu X, Chen M, et al. A hierarchical architecture for the future internet of vehicles[J]. IEEE Communications Magazine, 2019, 57(7):41-47.
- [58] 陈山枝, 胡金玲, 赵丽等. 蜂窝车联网 (C-V2X) [M]. 北京:人民邮电出版社, 2021:72-103.
- [59] Neto J, Gomes L C, Castanho E M, et al. An error correction algorithm for forward collision warning applications[C]. IEEE International Conference on Intelligent Transportation Systems. IEEE, Rio de Janeiro, Brazil, 2016:1926-1931.
- [60] Xu X, Liu K, Xiao K, et al. Design and implementation of a fog computing based collision warning system in VANETs[C]. IEEE Symposium on Product Compliance Engineering - Asia (ISPCE-CN). IEEE, Shenzhen, China, 2018:1-6.
- [61] Liu C, Liu K, Ren H, et al. Enabling safety-critical and computation-intensive IoV applications via vehicular fog computing[C]. International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). Shenzhen, China, 2019:378-383.
- [62] Liu C, Liu K, Guo S, et al. Adaptive offloading for time-critical tasks in heterogeneous internet of vehicles[J]. IEEE Internet of Things Journal, 2020, 7(9):7999-8011.
- [63] 欧杰. 基于 DSRC 技术的车辆防碰撞语音提示与辅助控制系统的研发[D]. 重庆: 重庆大学, 2018:23-57.
- [64] 杨允新. 基于 V2X 的车路协同系统防撞预警研究与实现[D]. 重庆:重庆邮电大学,

2019:41-76.

- [65] 刘玉蓉. 基于 V2X 通信的车辆防撞预警系统的研究与实现[D]. 武汉:武汉理工大学, 2019:44-67.
- [66] Wang J, Jiang C, Zhu H, et al. Internet of vehicles: Sensing-aided transportation information collection and diffusion[J]. IEEE Transactions on Vehicular Technology, 2018, 67(5):3813-3825.
- [67] Wang H, Liu R P, Ni W, et al. VANET modeling and clustering design under practical traffic, channel and mobility conditions[J]. IEEE Transactions on Communications, 2015, 63(3):870-881.
- [68] Yang P, Zhang N, Bi Y, et al. Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach[J]. IEEE Network, 2017, 31(5):14-20.
- [69] Jia D T, Lu K, Wang J, et al. A survey on platoon-based vehicular cyber-physical systems[J]. IEEE Communications Surveys & Tutorials, 2015, 18(1):263-284.
- [70] Zhou H, Bo L, Luan T H, et al. Chain cluster: Engineering a cooperative content distribution framework for highway vehicular communications[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(6):2644-2657.
- [71] Ucar S, Ergen S C, Ozkasap O. IEEE 802.11p and visible light hybrid communication based secure autonomous platoon[J]. IEEE Transactions on Vehicular Technology, 2018, 67(9):8667-8681.
- [72] Breslau L, Pei C, Li F, et al. Web caching and zipf-like distributions: Evidence and implications[C]. International Conference on Computer Communications (INFOCOM). IEEE, New York, USA, 1999:126-134.
- [73] Gorski J, Pfeuffer F, Klamroth K. Biconvex sets and optimization with biconvex functions: a survey and extensions[J]. Mathematical Methods of Operations Research, 2007, 66(3):373-407.
- [74] H Wang, Jing W, Wen X, et al. Segment-based joint caching and recommendation optimization for mobile video transmission[C]. IEEE Globecom Workshops (GC Wkshps). IEEE, Waikoloa, HI, USA, 2019:1-6.

- [75] Bojanowski P, Lajugie R, F Bach, et al. Weakly supervised action labeling in videos under ordering constraints[J]. Lecture Notes in Computer Science, 2014, 8693(6):628-643.

攻读学位期间取得与学位论文相关的成果

发表和投稿与学位论文相关学术论文

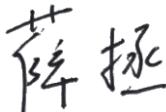
- [1] 薛拯, 刘洋, 韩国军, 闫晶莹. 车联网下基于模糊逻辑的簇头选择与数据传输策略研究[J]. 小型微型计算机系统, 2020, 41(7): 1458-1463.
- [2] **Zheng Xue**, Yang Liu, Guojun Han, Ferheen Ayaz, Zhengguo Sheng and Yonghua Wang. “Two-layer Distributed Content Caching for Infotainment Applications in VANETs,” IEEE Internet of Things Journal, second review after minor revision, 2021.

申请发明专利

- [1] 韩国军, 闫晶莹, 刘洋, 薛拯. 一种失效节点检测方法、装置和相关设备[P]. 中国: 201910888391.8 (实审中)
- [2] 陆俏云, 刘洋, 韩国军, 薛拯, 闫晶莹. 一种基于分簇路由协议的行驶车辆通信方法、装置及介质[P]. 中国: 201910532314.9 (实审中)
- [3] 韩国军, 林旭斌, 刘洋, 陆俏云, 薛拯. 一种车联网路由改进方法[P]. 中国: 201811396033.7 (实审中)

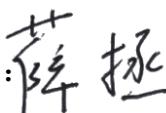
学位论文独创性声明

本人郑重声明：所呈交的学位论文是我个人在导师的指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明，并表示了谢意。本人依法享有和承担由此论文所产生的权利和责任。

论文作者签名:  日期: 2021年6月7日

学位论文版权使用授权声明

本学位论文作者完全了解学校有关保存、使用学位论文的规定：“研究生在广东工业大学学习和工作期间参与广东工业大学研究项目或承担广东工业大学安排的任务所完成的发明创造及其他技术成果，除另有协议外，归广东工业大学享有或特有”。同意授权广东工业大学保留并向国家有关部门或机构送交该论文的印刷本和电子版本，允许该论文被查阅和借阅。同意授权广东工业大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、扫描或数字化等其他复制手段保存和汇编本学位论文。保密论文在解密后遵守此规定。

论文作者签名:  日期: 2021年6月7日

指导教师签名:  日期: 2021年6月7日

致 谢

时光飞逝，转眼三年研究生生涯接近尾声。从科研小白到如今完成毕业论文，五味杂陈，但相比入学前，我成长了许多也多了一些思考。在准备毕业之际，我将诚挚的感谢献给曾今热情指导和帮助过我的老师、同学、朋友以及关心支持我的亲人。

首先，最想要感谢的是我的导师韩国军教授，为我的研究工作提供了优越的软硬件环境，创造了严谨、浓厚、和谐的科研氛围。韩老师在学术上严谨的治学态度、渊博深厚的学识和前瞻性的洞察力，在科研课题上为我指明了方向。在定期的组会和进度汇报上，时刻提醒我们以更高的标准要求自己，互相学习，互相进步。此外，韩老师还不定期分享学术资料，邀请知名学者做学术报告，举办研讨会，参加学术会议，让团队有更多机会接触学术前沿知识，与同行深入交流学习，这大大开阔了我的眼界和学术思路。周末在校园也时常看见韩老师和方毅老师，您对科研事业的执著和热情让我敬佩。在生活中，韩老师平易近人，胸怀宽广，时常叮嘱多运动保持健康的身体，教导为人处世的道理，也会在我迷茫时给出一些人生建议。您既是恩师也是榜样，给了我在科研道路上继续前进的动力和信心，在此，向您致以崇高的敬意。

与此同时，还要感谢一直默默为研究工作提供指导和帮助的刘洋老师，引领我走入科研的大门，打好科研基本功，定期的组会、邮件、微信交流，探讨学术问题，推进课题进展，每一次激烈的讨论直击核心问题，也激励着我不断探索更好的方法解决难题。在我放松懈怠时，鞭策激励我继续前行，在我迷茫困惑时，给出宝贵的指导意见。您无私的传授和耐心的指导让我真正意识到科研该如何做，并不断进取和精进。衷心感谢您。在这里也对其他参与论文修改、提出宝贵意见的许杰老师、盛正国老师、Ferheen Ayaz 博士表示感谢！

感谢编码与存储课题组的小伙伴们提供的支持与帮助，与你们相遇是一种缘分，也从你们身上学习到了很多，一起运动、聚餐、游玩、吐槽，有你们的陪伴为科研生活增添了欢乐。感谢同届的黄三维、胡海华、陈凌鹏、闫晶莹、吴优同学，与你们一起上课、科研度过三年时光非常开心，向你们请教问题也会给予无私的帮助。

感谢多年来一直陪伴、支持和关怀我的亲人们，谢谢理解与包容，你们是我最坚强的后盾！

向所有在论文中引用的文献作者以及出版者致以衷心的感谢，你们的智慧和劳动成果为我提供素材，让我顺利完成学位论文。如有疏漏，向您表示歉意和感谢！

最后，感谢在百忙之中抽出宝贵时间评阅论文的各位评审专家！