Slicing-Based Resource Management for Satellite-Terrestrial Integrated Vehicular Networks

by

Mingcheng He

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Mingcheng He 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Amiya Nayak Professor School of Electrical Engineering and Computer Science University of Ottawa
Supervisor(s):	Xuemin (Sherman) Shen University Professor Department of Electrical and Computer Engineering University of Waterloo
Internal Member:	Kshirasagar Naik Professor Department of Electrical and Computer Engineering University of Waterloo
Internal-External Member:	Amir Khajepour Professor Department of Mechanical and Mechatronics Engineering University of Waterloo
Internal-External Member:	Wei-Chau Xie Professor Department of Civil and Environmental Engineering University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Connected autonomous vehicles (CAVs), which enable information exchange and content delivery through vehicular networks to support multifarious vehicle-to-everything (V2X) applications, are expected to play a very important role in the safety and efficiency of future vehicular transportation systems. However, V2X applications necessitate stringent requirements on low delay, high reliability, and global connectivity, which pose challenges to existing terrestrial networks due to the geographic-constrained fixed infrastructure deployment and scarce resources. In this thesis, we consider satellite-terrestrial integrated vehicular network (STIVN) for supporting V2X applications with reliable, flexible, and ubiquitous service provisioning by jointly managing communications, computing, and caching resources from both terrestrial and low Earth orbit (LEO) satellite networks. Particularly, we will investigate the following three research issues: 1) how to manage resources with low complexity for the coupling decision-making in STIVN in terms of diverse communication modes with different resource costs; 2) how to flexibly coordinate resources from satellite and terrestrial networks with distinct and time-varying coverage to find the global optimal resource utilization across large-scale regions; and 3) how to adaptively schedule resources in STIVN to respond to dynamic network environments resulting from the mobility of satellites and CAVs, as well as the spatiotemporal fluctuations in service demand.

We propose a novel network slicing-based resource management framework to address the three issues. For the first issue, we investigate reservation-stage resource management in STIVN by jointly optimizing coupling decisions to support high-definition (HD) map distribution, where both satellites and terrestrial base stations (BSs) can deliver HD map files to CAVs. By reserving both communication and caching resources from both terrestrial and satellite networks, the goal is to minimize the overall cost of STIVN in terms of resource usage, delay, and slice reconfiguration cost while accommodating more service demand. Considering both multicast and unicast communication modes can support the HD map distribution in STIVN with different characteristics, we design a resource slicing architecture by establishing two slices accordingly to jointly support the HD map distribution application and formulate the resource management problem as a sequential decision-making problem. To solve the problem, we propose a two-layer resource slicing approach, which is a data-model co-driven approach, to find the optimal resource reservation and slicing window length decisions with low complexity. Specifically, considering the time-varying available resources and dynamic service demand, we design a reinforcement learning-based algorithm to decide the communication resources reserved in each network slice in each access point and the slicing window length to find the long-term optimal

performance. Then, according to the reserved communication resources in each slice, a swap-based matching algorithm is derived to determine the optimal multicast and caching decisions in each slicing window. To address the second issue, we study reservation-stage resource management across multiple cells in STIVN to coordinate satellite and terrestrial networks. The cell has the same coverage of satellite beams, and the controller for each cell establishes network slices for each application and manages resources. Considering the distinct and time-varying coverage of satellite and terrestrial networks, it is difficult to achieve the global optimum, where each cell has different available resources, and resources from each satellite can be shared across different cells. To reserve resources for each slice, we formulate a resource slicing problem with cross-cell coordination in STIVN to minimize the overall cost related to resource usage and delay performance and propose a distributed resource slicing scheme for flexible resource management to solve the problem. To alleviate the competition of resources among different cells, an asynchronous multi-agent proximal policy optimization-based algorithm is presented to select satellites serving each cell in each slicing window, and then a distributed optimization algorithm is employed to determine the resource reservation decisions for each slice. To address the third issue, we investigate operation-stage resource management in STIVN by adaptively scheduling resources in real-time to cope with dynamic network environments. Specifically, we focus on the infrastructure-assisted environment sensing application where CAVs in the terrestrial cell require fresh environment data from satellites and roadside units (RSUs) to increase the trust level of the observation. To support satellite- and RSU-assisted environment sensing, we formulate a long-term resource scheduling problem in STIVN to satisfy sensing data freshness and reliability requirements with efficient resource usage. We propose a cooperative satellite-terrestrial resource scheduling scheme that jointly optimizes the sensing interval and resource allocation in STIVN. Leveraging the multicast capabilities of satellites to simultaneously serve multiple CAVs in a group, CAV partitioning and sensing interval determination methods are designed to minimize the cost of computing and communication resources. Based on the CAV grouping and sensing interval, we then develop a reinforcement learning-based algorithm to make real-time decisions on computing and communication resource allocation for different CAV groups to cope with highly dynamic environments. In summary, the proposed slicing-based resource management scheme can support V2X applications with reliable, flexible, and ubiquitous service provisioning and provide useful guidelines for network resource management in future vehicular transportation systems.

Acknowledgements

Pursuing my Ph.D. degree at the University of Waterloo was one of the best decisions I have ever made and has been a truly rewarding experience for me. During my Ph.D. study, I am deeply grateful to all those who have supported and guided me, without whom this thesis would not be possible. I would like to take this opportunity to give my sincere thanks to them.

First of all, I would like to express my deepest and sincerest appreciation to my supervisor, Prof. Xuemin (Sherman) Shen, for his continuous support and patient guidance during my Ph.D. study at the University of Waterloo. From Prof. Shen, I have learned not only the rigorous research attitude and independent thinking but also the importance of balancing work and life. His insightful advice always inspired me to be a better researcher and helpful person. I have been extremely fortunate to have Prof. Shen as my supervisor and he will always play a significant role model in my life.

I would like to thank Prof. Kshirasagar Naik, Prof. Amir Khajepour, Prof. Wei-Chau Xie, and Prof. Amiya Nayak for serving on my thesis examination committee. Their insightful comments and valuable suggestions have significantly improved the quality of my thesis. I would also like to express my gratitude to Prof. Weihua Zhuang for helping me to build the knowledge base from her courses and improve my research skills, which greatly benefits my research in this thesis.

In the past four years, precious friendship, support, and help from BBCR members have made my memories at the University of Waterloo unforgettable and enjoyable. I would like to express my sincere thanks to Prof. Huaqing Wu, Dr. Conghao Zhou, Prof. Mushu Li, Prof. Jie Gao, Prof. Wenchao Xu, Dr. Kaige Qu, Dr. Weisen Shi, Yingying Pei, Shisheng Hu, Yannan Wei, Ruoxu Wang, Xinyu Huang, Zhixuan Tang, Yixiao Zhang, Man Ouyang, and Bicheng Guo for their great help in both my research and life. I am also grateful for the wonderful time spent in Canada with Prof. Dongxiao Liu, Prof. Cheng Huang, Prof. Liang Xue, Han Yin, Lingshuang Liu, Prof. Yilong Hui, Jianzhe Xue, and all current and former BBCR members.

Finally, I would like to greatly thank my parents for their love and encouragement during my Ph.D. study. I would like to extend my special thanks to Mengxue Zhang, for accompanying, understanding, and supporting me in these four years.

Mingcheng He
Oct. 10, 2024
Waterloo, Ontario, Canada

Dedication

This Ph.D. dissertation is dedicated to my beloved parents, Xuelian He and Lei Chen.

Table of Contents

List of Figures xii

List of Tables xiv

List of Abbreviations xv

1 Introduction 1

1.1 Overview of Vehicular Networks 1

1.2 Overview of STIVN 6

1.2.1 Terrestrial Vehicular Networks 7

1.2.2 LEO Satellite-Assisted Vehicular Networks 8

1.2.3 Satellite-Terrestrial Integrated Vehicular Networks 9

1.3 Slicing-Based Resource Management in STIVN 11

1.3.1 Overview of Resource Slicing Architecture 12

1.3.2 Slicing-Based Resource Management Procedure in STIVN 14

1.4 Research Motivations and Contributions 17

1.4.1 Challenges of Slicing-Based Resource Management in STIVN 17

1.4.2 Research Problems and Contributions 19

1.5 Thesis Outline 21

2 Literature Review **23**

2.1 LEO Satellite Networks . . . 23

2.1.1 Background of LEO Satellite Networks . . . 23

2.1.2 Standardization of LEO Satellite Networks . . . 25

2.1.3 Performance Analysis on LEO Satellite Networks . . . 26

2.2 Slicing-Based Resource Management in STIVN . . . 26

2.2.1 Operation-Stage Resource Management . . . 27

2.2.2 Reservation-Stage Resource Management . . . 31

3 Reservation-Stage Multi-Dimensional Resource Management **33**

3.1 Background and Motivations . . . 33

3.2 Network Scenario and System Model . . . 37

3.2.1 Scenario Description . . . 38

3.2.2 Resource Slicing Architecture . . . 39

3.2.3 Effective Capacity Analysis . . . 41

3.2.4 Performance Analysis for Different Slices . . . 42

3.3 Problem Formulation and TLRS Scheme Design . . . 43

3.3.1 Problem Formulation . . . 43

3.3.2 Design of Two-Layer Resource Slicing Scheme . . . 46

3.4 Matching-Based Caching and Multicast File Selection . . . 47

3.4.1 Accommodated Service Demand of Slices . . . 48

3.4.2 Many-to-Many Matching Game Formulation . . . 49

3.4.3 Swapping-Based Matching Algorithm for Caching and Multicast File Selection . . . 51

3.5 HPPO-Based Slicing Window Length and Resource Reservation . . . 53

3.5.1 Markov Decision Process Formulation . . . 53

3.5.2 HPPO-Based Resource Slicing Algorithm . . . 54

3.6 Performance Evaluation . . . 58

3.6.1 Simulation Setting . . . 58
3.6.2 Simulation Results . . . 59
3.7 Summary . . . 67

4 Reservation-Stage Resource Management with Cross-Cell Coordination 68
4.1 Background and Motivations . . . 68
4.2 Network Scenario and System Model . . . 71
4.2.1 Scenario Description . . . 71
4.2.2 Communication Model . . . 72
4.3 Problem Formulation and Decomposition . . . 74
4.3.1 Problem Formulation . . . 74
4.3.2 Problem Decomposition . . . 75
4.4 Distributed Resource Slicing Scheme with Cross-Cell Coordination . . . 76
4.4.1 Distributed Optimization Algorithm for Resource Reservation . . . 77
4.4.2 Reinforcement Learning-Based Access Point Selection . . . 79
4.5 Performance Evaluation . . . 82
4.6 Summary . . . 86

5 Operation-Stage Resource Management for Environment Sensing 87
5.1 Background and Motivations . . . 87
5.2 Network Scenario and System Model . . . 90
5.2.1 Scenario Description . . . 90
5.2.2 Age of Information Model for Environment Sensing . . . 93
5.2.3 Computing and Communication Model . . . 95
5.3 Problem Formulation and Decomposition . . . 96
5.4 CAV Partitioning and Sensing Interval Determination . . . 99
5.4.1 PSO-based Sensing Interval Determination . . . 100
5.4.2 Coalition Game-Based CAV Partitioning Algorithm . . . 102

5.4.3 Computational Complexity Analysis . . . 105
5.5 PPO-Based Real-Time Resource Allocation Algorithm . . . 106
5.5.1 Markov Decision Process Formulation . . . 106
5.5.2 PPO-Based Algorithm for Resource Allocation . . . 107
5.6 Performance Evaluation . . . 109
5.6.1 Simulation Setting . . . 109
5.6.2 Simulation Results . . . 110
5.7 Summary . . . 115

6 Conclusions and Future Works **116**
6.1 Main Research Contributions . . . 116
6.2 Future Works . . . 118

References **120**

APPENDICES **132**

A Proof of Proposition 1 **133**

List of Figures

1.1 An overview of V2X applications in vehicular communication networks. . . 3
1.2 Satellite-terrestrial integrated vehicular network architecture. 9
1.3 Illustration of resource slicing architecture in STIVN. 12
1.4 Slicing-based two-stage resource management scheme in STIVN. 13
1.5 Slicing-based resource management procedure in STIVN. 16

3.1 Illustration of STIVN scenario. 34
3.2 Resource slicing architecture in STIVN. 39
3.3 Working diagram of the proposed TLRS scheme in STIVN. 46
3.4 Illustration of proposed HPPO-based algorithm. 55
3.5 Convergence performance of the proposed TLRS scheme. 59
3.6 Performance under different policies for slicing window length. 60
3.7 System cost under different policies for slicing window length. 62
3.8 Average system cost under different minimum elevation angles of LEO satellites. 63
3.9 Performance under different resource reservation strategies. 64
3.10 Average system cost with different resource reservation policies. 65
3.11 Performance for different caching and multicast policies. 66

4.1 Resource slicing across multiple cells in STIVN. 69
4.2 Service demand in different cells. 83
4.3 Convergence performance. 83

4.4 CDF of system cost in each time slot. 84
4.5 Performance under different minimum elevation angles of satellites. 85

5.1 Considered environment sensing scenario in STIVN. 92
5.2 AoI illustration of CAV. 94
5.3 Working diagram of the proposed CSTRS scheme for environment sensing. 97
5.4 Convergence performance of training process. 111
5.5 Performance of different resource allocation algorithms. 112
5.6 Resource cost in LEO satellite with different reliability requirements. . . . 113
5.8 Resource cost in LEO satellite under different RSU densities in rural area. 113
5.7 Performance of different CAV partition policies. 114
5.9 Resource cost in rural areas from terrestrial and LEO satellite networks. . 115

List of Tables

1.1 V2X applications, use cases, and service requirements 5

3.1 Summary of Notations 38
3.2 Simulation Parameters 58

5.1 Summary of Notations 91
5.2 Simulation Parameters 110

List of Abbreviations

3C	Communication, Computing, and Caching
3GPP	3rd Generation Partnership Project
5G	Fifth-Generation
6G	Sixth-Generation
ADMM	Alternating Direction Method of Multipliers
AI	Artificial Intelligence
AoI	Age of Information
AP	Access Point
BS	Base Station
CAV	Connected Autonomous Vehicle
CDF	Cumulative Distribution Function
DNN	Deep Neural Network
DQL	Deep Q-Learning
DRL	Deep Reinforcement Learning
ETSI	European Telecommunications Standards Institute
GEO	Geostationary Earth Orbit
HD	High-Definition
HPPO	Hybrid Proximal Policy Optimization
IoT	Internet of Things
LEO	Low Earth Orbit
MDP	Markov Decision Process
MEC	Mobile Edge Computing
MEO	Medium Earth Orbit
NHPP	Non-Homogeneous Poisson Process
NOMA	Non-Orthogonal Multiple Access
NTN	Non-Terrestrial Networks
OBU	Onboard Unit
PPO	Proximal Policy Optimization

PSO	Particle Swarm Optimization
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Networks
RL	Reinforcement Learning
RoI	Range of Interest
RSU	Road Side Unit
SDN	Software-Defined Networking
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SSG	Satellite Service Group
STIN	Satellite-Terrestrial Integrated Networks
STIVN	Satellite-Terrestrial Integrated Vehicular Networks
TSG	Terrestrial Service Group
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2S	Vehicle-to-Satellite
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VM	Virtual Machine

Chapter 1

Introduction

Connected and Autonomous Vehicles (CAVs) are playing an increasingly vital role in enhancing road safety, improving transportation efficiency, and elevating the experience of passengers, which enables vehicles to interact with environments through vehicular networks [1]. To empower vehicle-to-everything (V2X) applications, especially in the future driverless era, reliable network accessibility and improved quality of service (QoS) are essential. However, the limitations of current terrestrial vehicular networks, particularly in terms of coverage, pose great issues in ensuring service continuity. Satellite-terrestrial integrated vehicular networks (STIVN), which integrate terrestrial networks with satellite networks, provide a promising solution for ubiquitous network coverage with boosted network capacity and flexible resource provisioning. In this case, efficient resource management in STIVN is crucial to support diversified V2X services but also poses a range of challenges in handling highly dynamic environments and differentiated service requirements. To address the challenges, a resource slicing architecture in STIVN is envisioned to enable flexible and adaptive resource management while satisfying different QoS requirements. In this chapter, we first provide an overview of vehicular networks. Then, we elaborate on the characteristics of STIVN and introduce the potential of slicing-based resource management. Finally, we present the three key research problems investigated in this thesis.

1.1 Overview of Vehicular Networks

As the number and density of vehicles increase on the road, three major issues occur, namely safety, congestion, and environment. For the safety issue, according to the statistics from the US National Highway Traffic Safety Administration (NHTSA), 94% of traffic accidents

are caused by human mistaken operation [2] and the number of road traffic deaths on the roads reaches 1.19 million a year [3]. In addition, the traffic congestion problem, especially during rush hours, can significantly extend the commute time, where there are about 51 hours of congestion per year. The low transportation efficiency also causes an environmental problem with lots of carbon dioxide emissions. For example, the wasted fuel and increasing vehicle emissions decrease the air quality.

Benefiting from advances in artificial intelligence (AI) and high-performance computing, autonomous vehicles have gained increased attention due to their ability to make decisions without human intervention with onboard computational devices and sensors, offering faster responses to emergent events and preventing accidents caused by human errors for a safer and more efficient driving [4]. However, relying solely on single-vehicle intelligence without proper communication means that the vehicle can only obtain limited information through its onboard sensors. This limitation leads to decision-making that often reaches only local optima, ignoring the needs of the broader transportation system, and making it challenging to achieve coordinated driving among multiple vehicles. Moreover, in certain extreme situations such as heavy snow or fog, the onboard sensors of an autonomous vehicle may struggle to accurately perceive the surrounding environment, compromising safety and reliability.

To address these limitations, enabling vehicles to communicate and form a network on the road, known as vehicular networks, is a potential solution to allow vehicles being able to communicate by leveraging vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), or vehicle-to-network (V2N) communication, collectively called vehicle-to-everything (V2X) communications. In this case, connected autonomous vehicles (CAVs) are introduced, which allow autonomous vehicles to exchange information or receive content in real-time using intelligent onboard communication devices in the vehicular networks. This capability not only supports safer and more efficient driving but also enhances coordination among vehicles, optimizing the entire transportation system rather than just individual vehicles. The technique of CAVs enables interaction with environments to improve road safety, transportation efficiency, and the driving experience with fewer environmental impacts [1, 5], as well as provide more accurate and reliable information for traffic signal control [6, 7]. As network-attached vehicles, CAVs can leverage V2X communications to support a wide range of applications, known as V2X applications, allowing for a more comfortable and efficient travel experience for all road users, as shown in Fig. 1.1. Through such vehicle interactions, CAVs help improve both individual autonomous vehicle decision-making and the overall transportation system's performance, especially in challenging conditions where sensor capabilities are limited. Here, we present some typical V2X applications [8]:

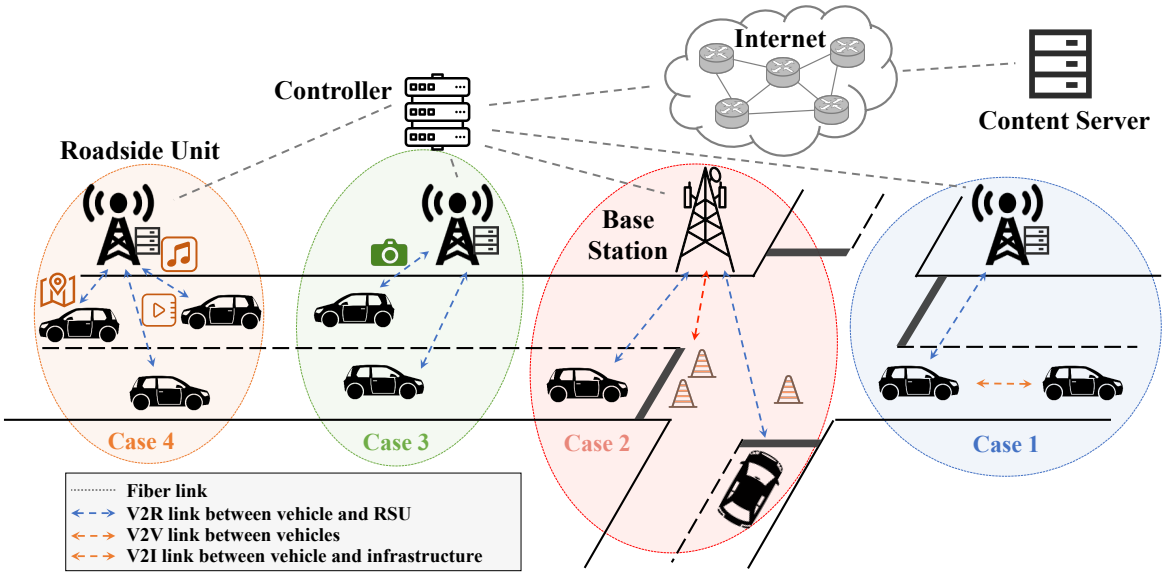


Figure 1.1: An overview of V2X applications in vehicular communication networks.

1. **Autonomous driving information -** CAVs can share driving intentions and operations, known as autonomous driving signaling information, with other traffic participants potentially involved in the cooperative maneuver, based on self-sensing and received traffic information, shown as **Case 1** in Fig. 1.1. By exchanging data with other vehicles or infrastructures, a cooperative operation can be realized for lane changing or platooning to optimize the trajectory, which can enhance transportation efficiency with less congestion. Moreover, for some emergent situations such as the CAV needing to brake suddenly, the CAV can receive or send the emergent information for safety.

2. **Road safety information -** Road safety information is of the utmost importance for connected vehicles to ensure driving safety. To reduce the frequency and severity of vehicle collisions, vehicles need real-time surrounding safety information, such as the collision of other vehicles or some road hazard warning. With V2X communications, the vehicle can deliver alarm information when necessary, perform accident forewarning, and make proper maneuver decisions, shown as **Case 2** in Fig. 1.1.

3. **Traffic efficiency information -** To facilitate traffic management and transportation efficiency, traffic efficiency data, such as vehicles' locations and congestion status, are essential for CAVs to plan the trajectory and avoid collisions. Generally, onboard

sensors on vehicles, including cameras, radar, and LiDAR sensors, can provide surrounding environment information. Sensors deployed on the infrastructures can provide enhanced environment information regarding dynamic traffic on the road to increase the viewing range of each CAV, shown as **Case 3** in Fig. 1.1. In this case, the traffic efficiency data frequently transmitted to CAVs can increase their driving trust level and enable a smooth and prescient driving experience. For example, trajectory planning and real-time navigation for vehicles can be automatically updated according to the received traffic condition information to avoid congestion areas and enhance traffic efficiency.

4. **Infotainment content -** As a key feature of CAV, infotainment content, such as multimedia content, high-definition (HD) maps, and some location-based content, can be obtained by downloading from the infrastructure to better support autonomous driving and provide a comfortable and convenient travel experience for passengers. To enhance localization and provide better navigation, HD maps are essential for CAVs with their high-accuracy and detailed road models to represent static or semi-dynamic objects [7]. Moreover, in some parking lots or tolls, the location-based information exchanges between CAVs and infrastructures can simplify the parking or paying process to enhance convenience, with short-range communication. In addition, video streaming, music, games, or augmented reality (AR) applications can be delivered and shared to provide informative or entertaining services [9]. For instance, passengers may request some location-based content during their traveling to obtain updated restaurant or sight information with some audio or video content, shown as **Case 4** in Fig. 1.1.

Service requirements of different applications are diversified in terms of content size, delay, and reliability [8, 10–12], as summarized in Table 1.1. The definitions of network requirements are given as follows:

- *Content size*: The size of the data needs to be transmitted each time.
- *Delay*: Time from the occurrence of the event in a scenario application zone to the completion of content delivery, including communication and processing time, which refers to the end-to-end delay [8, 13].
- *Reliability*: The probability that information or content is corrected received within a specified delay or freshness requirement [13].

Table 1.1: V2X applications, use cases, and service requirements

Application Categories	**Use Case Examples**	**Content Size**	**Delay Requirement**	**Reliability Requirement**
Autonomous Driving Signaling	Cooperative maneuvers: lane changing, platooning	Small	$\leq$ 20 ms	99.9%
	Emergent situations	Small	$\leq$ 10 ms	99.9999%
Road Safety Information	Collision warning	Small	$\leq$ 20 ms	99.9 %
	Road hazard warning: signal violation, roadwork warning	Small	$\leq$ 100 ms	99.9 %
Traffic Efficiency Data	Infrastructure-assisted environment perception	Variable	$\leq$ 100 ms	99 %
Infotainment Content	Automatic parking, electronic toll collect	Variable	$\leq$ 500 ms	99 %
	Delay-sensitive applications: live video streaming	Variable	$\leq$ 150 ms	90 %
	Delay-tolerant applications: HD map distribution, vehicle software updating	Variable	Best effort	Best effort

For autonomous driving signaling and collision warning information, the delay requirement should be stringent since the cooperation among nearby CAVs is a real-time action, which needs a fast reaction for safety considerations. Moreover, for some V2X applications, the delay and reliability requirements are coupled for the safety-critical applications. For instance, when the average delay is smaller than the delay requirement, there are still parts of delays of the service exceeding the delay requirement, which may affect the real-time decision and introduce some safety issues. In this case, the reliability requirement is important to ensure the safety level by controlling the upper bound of the delay distribution. Therefore, the joint delay and reliability requirement can ensure the effectiveness of V2X applications to improve driving safety and experience.

In summary, CAVs can improve road safety and provide better travel experiences by exchanging information with the help of vehicular networks to support multifarious V2X applications. Meanwhile, different types of service requirements from diversified V2X applications should be satisfied to fully take the benefits of vehicular networks for the

transportation system.

1.2 Overview of STIVN

Generally, each V2X application needs to consume different kinds of resources to satisfy the network requirements. For example, for autonomous driving information delivery, communication resources are needed in transmitting content in small sizes with ultra-low delay and high reliability, especially in emergent situations. For road safety information, the network requirement is similar to autonomous driving control information but with a loose delay requirement for some road hazard warnings. The traffic efficiency information can provide the environment information for CAVs, which requires low delay and high reliability for the up-to-date data, where communication and computing resources need to be jointly used for data processing and transmission. Moreover, different infotainment contents may have different delay and reliability requirements. For delay-sensitive infotainment applications, such as live video streaming, the low delay and high reliability can guarantee smoothness; while for delay-tolerant applications, such as content downloading, HD map distribution, or software updating, delay with the best effort is acceptable. For some scenarios, data processing may be required with computing resources, such as the transcoding of multimedia streaming. Moreover, popular content can be cached at the mobile edge to reduce the backhaul transmission delay. Therefore, the combination of communication, computing, and caching (3C) resources needs to be jointly considered for multifarious V2X applications. In this case, effectively managing multi-dimensional resources in vehicular networks is essential to support V2X applications with the information processing and exchange between vehicles and infrastructures. In addition, to guarantee the service continuity of V2X applications, seamless vehicular networks are essential, where STIVN is considered to expand the network coverage by integrating satellite and terrestrial networks for ubiquitous network connection.

In this section, we will first introduce the network architecture of the current terrestrial vehicular networks and their challenges in supporting V2X applications. To cope with limited coverage and intermittent network service in terrestrial vehicular networks, we then present satellite networks as a complement to terrestrial vehicular networks and introduce their advantages. Being able to access both terrestrial and satellite vehicular networks, the benefits from STIVN are then presented for supporting V2X applications.

1.2.1 Terrestrial Vehicular Networks

To support multiple V2X applications and satisfy different service requirements, terrestrial vehicular networks, as the main provider for wireless networks in most scenarios, as shown in Fig. 1.1, can support a high transmission rate and low latency service for V2X applications. By deploying onboard units (OBUs), V2V communications can be realized among CAVs within a small communication range to satisfy the high-speed and high-efficiency requirements. The safety-critical information exchange can be supported by V2V links with a low delay. However, when heavy data needs to be transmitted, the limited transmission power of vehicles may constrain the data rate resulting in an unsatisfactory delay. The new standard of NR-V2X for the link between roadside units (RSUs) or base stations (BSs) and vehicles can support V2X applications to improve transmission reliability and reduce delay with a high transmission data rate [14, 15]. For example, infotainment content can be delivered from RSUs or BSs within their coverage area. Moreover, since some V2X applications, such as environment sensing and video streaming, require a tremendous quantity of computing demand for data processing before transmitting the data to CAVs, traditional cloud-based computing introduces a large delay on the backhaul link. By deploying mobile edge computing (MEC) servers on network components in terrestrial vehicular networks, RSUs or BSs have powerful computing capabilities and massive caching storage to enhance network operations and support more applications with satisfied service requirements in vehicular networks. To summarize, with the dense deployment of network components, V2X applications can be supported by terrestrial vehicular networks, where CAVs can reach a high transmission data rate and low delay with V2X communications, and leverage computing capability for processing data on MEC servers and edge caching to reduce backhaul transmission delay.

The terrestrial vehicular networks still have some limitations in the content delivery of V2X applications. First, in scenarios such as highways, rural areas, or remote areas, the deployment of terrestrial RSUs or BSs may be limited due to the high deployment, maintenance cost, or some geographical constraints. Therefore, the terrestrial vehicular network cannot provide ubiquitous network service for CAVs, resulting in intermittent network connections. Without network connections, some of the V2X applications, such as traffic efficiency information and infotainment content, cannot be continuously delivered to CAVs, which will significantly degrade the transportation efficiency and user experience. Second, considering the uneven service demand, including the temporal dynamic traffic in a rush and spare time, and the spatial dynamic traffic in urban and rural areas, fixed network entities in terrestrial vehicular networks are not efficient for resource provisioning. Therefore, relying solely on terrestrial networks can hardly guarantee ubiquitous and reliable network

connectivity for CAVs, as well as efficiently managing resources in dynamic environments.

1.2.2 LEO Satellite-Assisted Vehicular Networks

To enlarge the network coverage and continuity of terrestrial networks, satellite networks, as indispensable components of 6G networks [16], are able to assist terrestrial vehicular networks to support CAVs with seamless and ubiquitous network connections, especially in rural or remote areas. Satellite networks are composed of satellites in different orbits, including geostationary-earth-orbit (GEO), medium-earth-orbit (MEO), and low-earth-orbit (LEO) satellites [17]. Although GEO and MEO satellites can provide large network coverage with a low relative velocity between satellites and CAVs, the high altitude will introduce a large propagation delay that exceeds the service requirements of most time-sensitive V2X applications.

In order to satisfy the low delay requirements, LEO satellite communications are deemed as promising solutions to support V2X applications with a low propagation delay. Specifically, LEO satellites can be considered in a regenerative mode as a 5G gNB to provide network access for CAV, which can support high-bandwidth V2N connectivity in a high-frequency band. Benefiting from global and dense deployment in different altitudes, LEO satellite networks have the capability to provide worldwide seamless coverage anytime anywhere. Moreover, since the LEO satellites operate less independently from terrestrial infrastructure, they are less vulnerable to natural disasters and can provide ultra-reliable and robust network connections for CAVs. Similar to terrestrial network components, by deploying edge servers, LEO satellites can support computing and caching to enhance network operations. In this case, LEO satellites have attracted substantial attention from academia [18] and industry [19–21] from the global network availability and flexible resource provisioning.

However, due to the constrained power and high-cost MEC servers on LEO satellites, each satellite can only provide limited processing capability and caching resources for users. In addition, although the propagation delay of LEO satellites is much smaller than that of satellites in higher orbits, it still can not be neglected for long-distance transmission compared with terrestrial wireless transmission. Moreover, due to the high speed of LEO satellites, which can be 7.56 km/s, the coverage area of satellites changes over time, resulting in a highly dynamic satellite network topology. In this case, solely leveraging satellite networks cannot effectively and efficiently support the huge service demand of V2X applications, and complementing satellite networks to terrestrial vehicular networks is a potential solution.

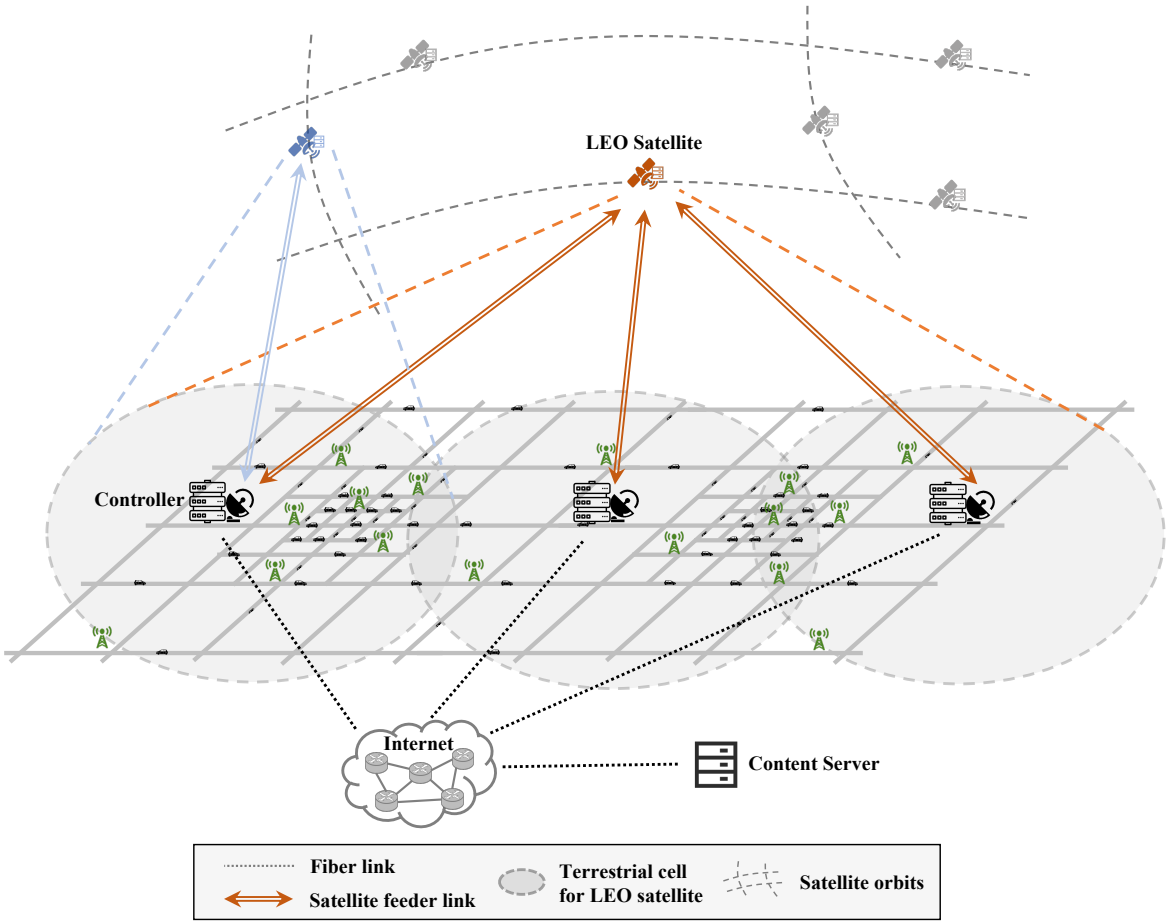


Figure 1.2: Satellite-terrestrial integrated vehicular network architecture.

1.2.3 Satellite-Terrestrial Integrated Vehicular Networks

With the assistance of satellite networks, STIVN can provide seamless, flexible, and cost-effective network services [22, 23], by leveraging distinct characteristics and exploiting complementary advantages from LEO satellite and terrestrial networks. Generally, LEO satellites have two different scenarios for the coverage on the terrestrial area, namely Earth-moving cell and Earth-fixed cell scenarios [24]. In the Earth-moving cell scenario, LEO satellites have fixed beam directions, where the beam footprint on the ground moves constantly, while in the Earth-fixed cell scenario, LEO satellites have steerable beams so that the beam footprint of each satellite points to a fixed cell on the ground when the cell is in its coverage.

In this thesis, we consider an Earth-fixed cell scenario, where the covered area for each satellite changes less frequently, which can thus improve the stability of satellite network service. For the STIVN, the terrestrial network consists of multiple cells, with the area of each cell being equal to the coverage area of a satellite beam. Each terrestrial cell has a central controller to cooperatively manage terrestrial and satellite networks serving this cell. The connection between RSUs or RSUs and the central controller is maintained by a fiber-based wired link, while a ground station co-located with the central controller is responsible for sending or receiving signals or contents via satellite feeder links. The detailed STIVN architecture is shown in Fig. 1.2, where CAVs are able to access and exchange data from both BSs or RSUs and LEO satellites under their coverage through wireless links, with single connectivity or dual connectivity for enhanced reliability [25, 26]. In addition, when some required contents are not cached in LEO satellites or BSs/RSUs, CAVs will fetch the content from the content server.

The main benefits of STIVN in supporting V2X applications are provided in the following aspects:

- *Ubiquitous and seamless network service:* With the dense deployment, LEO satellites can complement terrestrial vehicular networks to provide ubiquitous network services for CAVs. For example, LEO satellites are densely deployed or planned to be launched in different altitudes with different orbit inclination angles for global network services. Therefore, multifarious V2X applications can be empowered anywhere at any time in STIVN.

- *Flexible resource provisioning:* With the assistance of LEO satellites, the STIVN can provide more resources for V2X applications with more flexible resource management. For example, to transmit delay-sensitive information, such as safety messages, V2R communications in terrestrial networks can guarantee the stringent delay requirement in a small transmission distance. At the same time, applications that are not sensitive to delays, such as video or audio-based infotainment content, can be served by both terrestrial and satellite networks by flexibly scheduling resources from different networks. In addition, for some high-demand areas, data can also be offloaded to LEO satellites for computation to reduce the network burden of terrestrial networks.

- *Distinct transmission features:* In STIVN, both LEO satellites and terrestrial RSUs/BSs can communicate with CAVs within their coverage. Most of the content can be unicast through terrestrial network components to CAVs with a high transmission data rate and low delay. By exploiting different transmission features from satellite networks, satellite communications can effectively support vehicular networks. For example,

considering the mobility of CAVs, due to the wide coverage of LEO satellites, content with large data size can be delivered through LEO satellites to save terrestrial network resources and experience fewer handovers. In addition, considering the top-down nature of vehicle-to-satellite (V2S) communication, LEO satellites can serve multiple CAVs simultaneously by multicasting or broadcasting the content, which can reduce repeated transmission and consume less communication resources to accommodate more service demand. With the caching capacity, popular contents can be cached in LEO satellites to reduce the backhaul transmission delay and save caching resources in terrestrial networks.

1.3 Slicing-Based Resource Management in STIVN

To support V2X applications with diversified service requirements and ever-increasing service demands with multiple combinations of resources, efficiently managing resources in STIVN is crucial. However, traditional one-fits-all resource management schemes struggle to guarantee the network requirements of different V2X applications, potentially degrading transportation efficiency and safety levels. To address this issue, a service-oriented resource orchestration framework, namely slicing-based resource management, holds great potential for supporting various V2X applications. As an innovation in service-oriented resource management techniques for the 5G era, resource slicing will continue to play a pivotal role in meeting the requirements of diversified services through efficient resource provisioning [27, 28].

Specifically, resource slicing enables the establishment of multiple virtual networks, known as network slices, on the same physical network infrastructures. Different slices can support different applications with diverse requirements by isolating services and partitioning resources, which allows for service differentiation and guarantees service level agreements (SLA) for individual applications. Moreover, considering the dynamic service demands of different applications, resource slicing can dynamically adjust the resources targeting various applications. In contrast, traditional resource management approaches can only provide a fixed set of resources to all applications, lacking the flexibility to manage resources efficiently in the face of varying demands.

Different from resource slicing in 5G targeting fixed network infrastructures, resource slicing in STIVN needs the construction of network slices upon satellites with high mobility and RSUs or BSs in terrestrial networks while satisfying diverse SLA for different services in different areas. In this section, we will first provide an overview of resource slicing architecture in STIVN and then illustrate the slicing-based resource management.

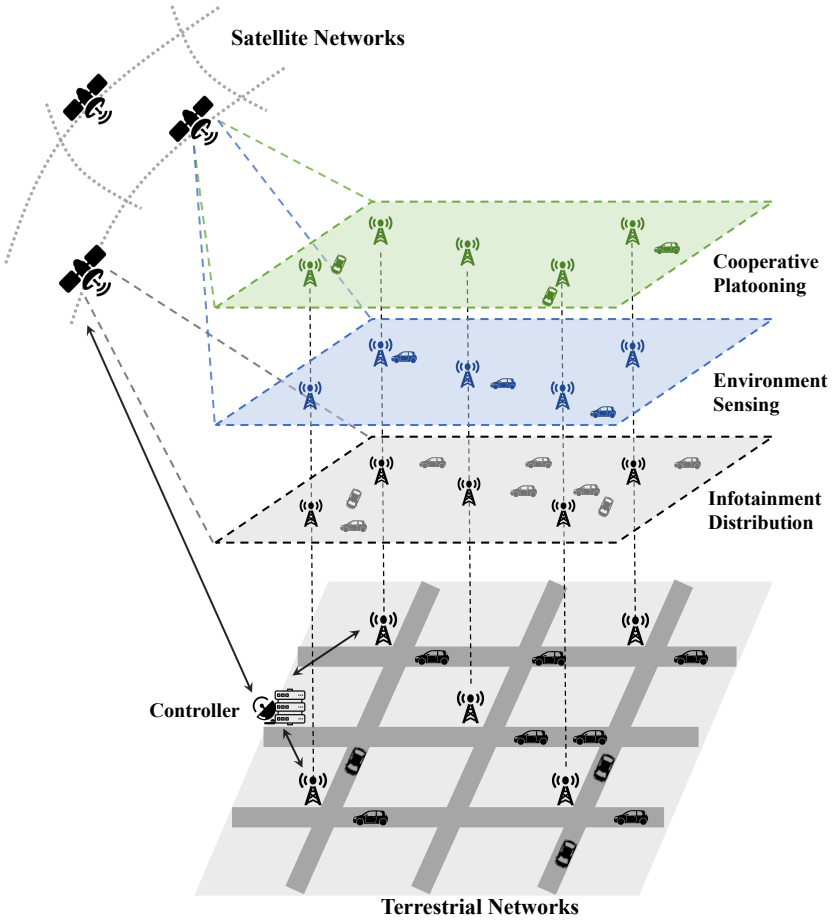


Figure 1.3: Illustration of resource slicing architecture in STIVN.

1.3.1 Overview of Resource Slicing Architecture

The illustration of resource slicing architecture in STIVN is given in Fig. 1.3, where both resources from terrestrial and satellite networks can be partitioned for different slices. A controller in the terrestrial cell is responsible for managing slices in different network components, i.e., satellites, RSUs, BSs, etc., to accommodate the service demands of different V2X applications. By adopting resource slicing, STIVN can support different V2X applications with diversified requirements by service isolation. Moreover, in STIVN, considering the predictable trajectory of LEO satellites, resource slicing can be used to further adapt to dynamic service demands by flexible satellite resource provisioning. For example, as shown in the figure, there are three different slices and each slice supports one

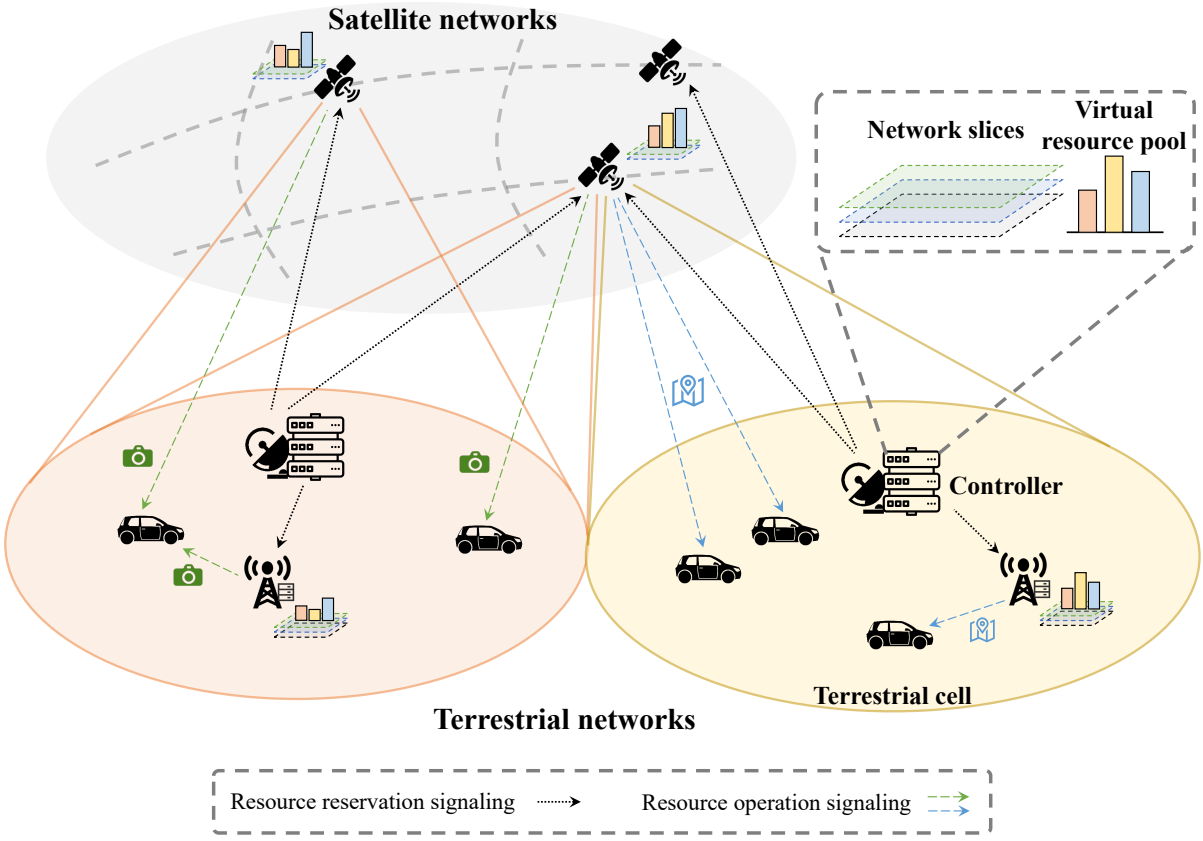


Figure 1.4: Slicing-based two-stage resource management scheme in STIVN.

service with specified service requirements. The three slices are built on the same physical network, but multi-dimensional resources in each slice can be managed and controlled separately and independently. Considering the high service demand for infotainment distribution in the following duration, in this case, only one slice can be built on the LEO satellite so that all resources can be used for serving one application.

With the assistance of resource slicing, a slicing-based two-stage resource management scheme for STIVN can be developed, including the reservation stage in the long term (e.g., minutes to several hours) and the operation stage in real-time (e.g., several milliseconds) [29, 30], as shown in Fig. 1.4. The two-stage design can effectively reduce resource management complexity by first reserving resources for each slice in the controller and then allocating resources for individual CAVs from the slice.

Specifically, a central controller located at the terrestrial networks is responsible for

constructing network slices and making decisions for both terrestrial RSUs and LEO satellites with global information in the reservation stage. By collecting information on available resources from RSUs and LEO satellites, a virtual resource pool, constituted by resources from each physical network component in STIVN, i.e., RSUs and LEO satellites, can be established at the central controller. With the predicted aggregated service demand of each V2X application, the central controller can then reserve multi-dimensional resources in each slice according to the available resources at each network component to satisfy different service requirements. Generally, the slicing decisions for the resource reservation should be adjusted from time to time to guarantee high performance of resource reservation decisions, which is referred to as a slicing window, related to the available resources and the variation of service demand.

Given the reserved resources in each network slice at each network component, the local controllers at LEO satellites and RSUs are responsible, during the operation stage, for allocating these reserved resources to each CAV in real-time to cope with the dynamic network environment. This allocation aims to meet the QoS requirements of different applications, ensuring that each vehicle is allocated the appropriate 3C resources based on its specific needs. The scheduling policy is highly task-dependent, as different tasks may have distinct QoS requirements. For instance, safety-critical applications demand ultra-reliable, low-latency communication, whereas infotainment services may prioritize high throughput. Therefore, the resource allocation strategy should be designed to accommodate such diverse V2X requirements with efficient resource consumption for guaranteeing QoS requirements by dynamically adjusting resource allocation according to the current network conditions and CAV demands. Moreover, the local controllers also gather information on the status of each network slice and CAV to analyze service satisfaction and monitor resource utilization for possible network slice adjustments. With the slicing-based resource management scheme for STIVN, flexible network configuration, customized service requirements, and isolated resource partition for different V2X applications can be realized by jointly managing resources from LEO satellites and RSUs.

1.3.2 Slicing-Based Resource Management Procedure in STIVN

The detailed procedure of the proposed slicing-based two-stage resource management scheme for STIVN is shown in Fig. 1.5. In order to satisfy diversified service requirements for V2X applications in STIVN in the reservation stage, which is a long-term decision-making stage, resources are served for different network slices corresponding to different V2X applications by the central controller in a large timescale based on a time-varying virtualized resource

pool, which includes the resources from LEO satellites and RSUs, and the mobility and traffic prediction. The detailed procedure of the reservation stage is shown as follows:

- *Mobility and traffic prediction*: In the central controller, by collecting the mobility information, service demand, and historical requirements from CAVs, the CAV mobility and service demand can be predicted through model-based, AI-based, or hybrid algorithms.
- *Virtualized resource pool*: Due to the mobility of LEO satellites in STIVN, a constantly changing network topology results in dynamically available resources. Resource virtualization can be adopted to abstract virtualized resources from the physical resources in BSs, RSUs, and LEO satellites. A collection of time-varying virtualized resources, instead of specific physical resources, can be reserved in each slice to serve different V2X applications to facilitate the management of network slices.
- *Network slice management*: Based on the long-term prediction of CAV mobility and service demands, the established virtualized resource pool, and the information on slice resource utilization from BSs, RSUs, and LEO satellites, the central controller can manage network slices by reserving and adjusting virtual resources for each slice in each slicing window. The reserved virtual resources of each slice can then map to the physical resources in each RSU and LEO satellite and be reserved for each V2X application for resource allocation in the operation stage.

Therefore, the reservation stage focuses on long-term resource planning for multiple V2X applications based on the network demand prediction and different service requirements of each application. In order to satisfy the delay and reliability requirement of different V2X applications, the resource reservation scheme needs to be investigated for reserving resources in each slice, which should avoid the over-provision or under-provision to achieve efficient resource utilization in STIVN.

Furthermore, in order to guarantee the service requirement for each CAV, RSUs, and LEO satellites need to allocate resources reserved in each slice to adapt to a highly dynamic network environment in the operation stage, which is a short-term decision-making stage. The resource allocation decisions for each CAV will be specified at each local controller at BSs, RSUs, and satellites according to the reserved resources in each slice and the service demand of each CAV. The detailed procedure of the operation stage is shown as follows:

- *Session creation*: When the CAV aims to contact RSUs, BSs, or LEO satellites for information uploading or downloading, a session is created to start the data transmission for V2X applications.

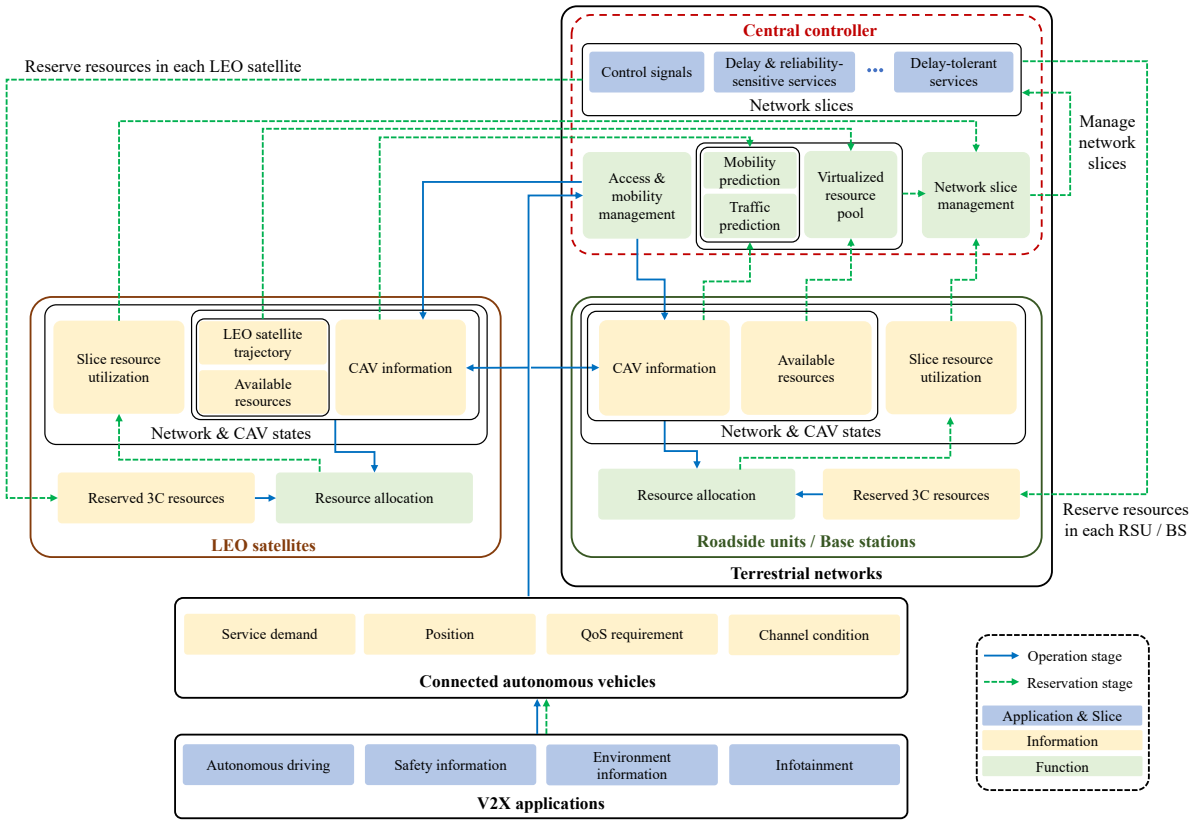


Figure 1.5: Slicing-based resource management procedure in STIVN.

- *CAV information collection*: When the session is created, the central controller, as well as RSUs, BSs, and satellites collect diverse information, including mobility of the CAV, service demand, the corresponding service requirement, and channel condition for the resource management and access management decision.

- *Access and mobility management*: When the central controller receives the information of the CAV, it can make the decision of access selection based on the service requirement and CAV information, and note the location and mobility of the CAV. The access selection result will be used for further resource allocation for different applications.

- *Resource allocation decision making*: Based on the access selection result, service information, CAV information, and real-time network status, including LEO satellite mobility and available resources in each slice, RSUs, BSs, and satellites can make the real-time decision of resource allocation for CAVs. By receiving the feedback or

latest CAV states, such as the QoS performance, channel conditions, etc., the central controller can further optimize the access selection, and RSUs or LEO satellites can adjust the real-time resource allocation to reach a better user-level network performance.

- *Available resources update*: Based on the resource reservation strategy from the central controller, the available resources will be updated for different slices for resource allocation in the operation stage.

To summarize, the operation stage mainly focuses on real-time resource allocation for each CAV given the reserved resources in each slice. In order to satisfy the delay and reliability requirement for individual CAVs, a proper resource management scheme is necessary to adapt to the highly dynamic network environment, such as CAV mobility, varying service demand, as well as the movement or trajectory of LEO satellites, aiming to satisfy service requirements with efficient resource utilization.

1.4 Research Motivations and Contributions

Although the slicing-based resource management in STIVN provides possible solutions to guarantee different V2X service requirements, enhance the flexibility for resource provisioning, and improve the adaptivity to dynamic service demand, the design of slicing-based resource management solutions in STIVN still meets several challenges. In this section, we first introduce the challenges of developing slicing-based resource management solutions in STIVN. Then, we outline three research efforts to address the challenges.

1.4.1 Challenges of Slicing-Based Resource Management in STIVN

In STIVN, it is essential to investigate how to design slicing-based resource management solutions to support the diversified service demands of V2X applications while satisfying service requirements. However, there are still several challenges, given as follows:

- **Different service requirements -** In STIVN, the slicing-based resource management should satisfy the service requirements of various V2X applications. Specifically, different V2X applications have different service requirements, such as delay, throughput, and reliability requirements as shown in Table 1.1. It is challenging to enable customized network configurations for multiple services while meeting their distinct

QoS requirements, as the configuration used for one service can impact those used for others. For instance, 3C resources consumed for one service may interfere with the available resources of other services, which increases the difficulties in resource slicing. Additionally, different applications may have varying preferences for satellite or terrestrial networks. For example, the HD map with high demand can be multicast by satellites while signaling information for autonomous driving prefers the terrestrial network with low delay. While the slicing-based resource management approach in STIVN has the potential to address these issues by isolating and optimizing resources for each application, efficiently managing these resources across such diverse service requirements remains a significant challenge.

- **Complicated decision-making -** The decision variables of resource management in STIVN are highly coupled with each other. First, both satellite and terrestrial networks in the STIVN have their specific features in terms of coverage, throughput, propagation delay, etc. Specifically, both multicast and unicast transmissions can be adopted in STIVN from different access points, which have different resource usage to accommodate different service demands. In this case, the unprecedented heterogeneity in STIVN renders the traditional resource management schemes inadequate where decisions from different networks may jointly affect the performance of serving V2X applications. Second, the STIVN should provide communication, computing, and caching resources for supported V2X applications, however, multi-dimensional resources are coupled for applications as illustrated in Section 1.2, which jointly affect network performances. Meanwhile, the cost of different resources in satellite and terrestrial networks may be different resulting from different expenses of equipment or band usage. Therefore, it is critical to efficiently customize the resource management policy by considering multi-dimensional resources in heterogeneous networks.

- **Difficult optimal control -** In STIVN, considering the mobility of satellites and distinct coverage of satellite and terrestrial networks, it is difficult to implement a globally centralized control for the global optimum resource management. Specifically, satellite coverage spans multiple and different cells for resource management across cells while the terrestrial infrastructures in each cell only serve vehicles within the cell. In this case, implementing a globally centralized controller to derive the optimal decisions will encounter formidable complexity since it requires extensive interactions between different controllers and satellites. Besides, it is complex to achieve a global consensus for each controller to manage resources from satellites and terrestrial infrastructures serving the cell, which introduces resource sharing and competition from different cells in one satellite. Therefore, it is essential to develop a slicing-based

method tailored for the heterogeneity in STIVN with distinct control coverage of different resources by leveraging resource virtualization.

- **Dynamic network environments -** The highly dynamic network environments for serving V2X applications, including the mobility of CAVs and satellites, spatiotemporal fluctuations of service demand, and dynamic channel conditions, bring a two-fold challenge. First, LEO satellites in the constellation move worldwide resulting in varying availability and quality of service in different cells. Additionally, in an Earth-fixed cell scenario, the coverage time of each satellite varies for a cell due to the movement of LEO satellites in different orbits and the varying position of orbits across the cell with Earth's rotation. Meanwhile, the terrestrial network components within each terrestrial cell may differ due to the variations in the deployment of terrestrial infrastructure, resulting in distinct available resources across different cells. Consequently, the available resources for different cells may be distinct and fluctuate over time. Second, due to different connectivity patterns for different infrastructures, such as distinct connection durations with satellites and RSUs, it is crucial to conduct real-time resource management to guarantee seamless service provisioning. Third, the reconfiguration of networks, such as the adjustment of resource reservation decisions in each slice, is required to adapt to dynamic service demands, which however yields additional costs of resource usage. In this case, the adaptive resource management for balancing the network performance and re-configuration cost is challenging. However, achieving optimal performance should take future network states into account, leading to a sequential decision-making problem, where solving such a problem with uncertainty and dynamic network environments is complicated.

1.4.2 Research Problems and Contributions

In this thesis, we investigate the slicing-based resource management in STIVN to tackle the aforementioned challenges. The objective is to develop low-complexity, flexible, and adaptive resource management schemes for STIVN to improve resource utilization while satisfying diversified service requirements of V2X applications. Specifically, we focus on the following three research problems in STIVN and contribute to designing resource management solutions to improve service performance and resource utilization.

1. We first investigate the resource management in a single terrestrial cell for HD map distribution in STIVN in the reservation stage to satisfy the service requirements by jointly optimizing coupling decisions, where both LEO satellites and terrestrial

BSs can deliver HD map files to CAVs. The goal is to configure network slices from both terrestrial and satellite networks in a low complexity to minimize the overall cost in terms of resource usage, delay, and slice reconfiguration cost while maximizing the service demand that can be accommodated to meet the service requirements. Taking the benefit of the satellite multicast and considering the different features of multicast and unicast transmission in STIVN, we design a resource slicing architecture in STIVN by establishing two subslices accordingly to jointly support the HD map distribution application and formulate the resource management problem as a sequential decision-making problem. Considering the interdependency of multi-dimensional decisions, a two-layer resource slicing approach is developed, which is a data-model co-driven approach, to find the optimal resource slicing decisions with less complexity. Specifically, considering the time-varying available resources and spatial-temporal dynamic service demand, we design a reinforcement learning-based algorithm to decide the communication resources reserved in each network slice in each access point and the slicing window length to find the long-term optimal performance. Then, according to the reserved communication resources in each slice, a swap-based matching algorithm is derived to determine the optimal multicast and caching decisions in each slicing window. Simulation results demonstrate that the proposed scheme in STIVN can effectively guarantee the service requirements with efficient resource usage and lower delay performance.

2. We then investigate the resource management across multiple terrestrial cells for both delay-tolerant and delay-sensitive applications as two slices in STIVN in the reservation stage. In the considered scenario, different cells have diversified service demands, terrestrial BS deployments, and different connections with satellites. In addition, with the large coverage of LEO satellites, resources from each satellite can be shared across different terrestrial cells, introducing the concept of cross-cell coordination of communication resources. The goal is to flexibly reserve resources from different networks to support service demands in different terrestrial cells. To reserve resources in STIVN for each slice, we formulate a resource slicing problem with cross-cell coordination to minimize the overall cost related to resource usage and delay performance and propose a distributed resource slicing scheme for flexible resource management to solve the problem. To alleviate the competition of resources among different cells, an asynchronous multi-agent proximal policy optimization (AMAPPO)-based algorithm is presented to select satellites serving each cell in each slicing window. Then, a distributed optimization algorithm is employed to determine the resource reservation decisions for each slice. Extensive simulations demonstrate that the proposed scheme can approach the optimal benchmark in terms of resource

usage and delay performance with less training time.

3. Lastly, we investigate the resource management for infrastructure-assisted environment sensing in one terrestrial cell in STIVN in the operation stage in highly dynamic environments. In the considered scenario, LEO satellites and roadside units (RSUs) cooperate to provide CAVs with fresh sensing data. To support satellite- and RSU-assisted environment sensing for CAVs, we formulate a long-term resource scheduling problem in STIVN to satisfy sensing data freshness requirements with efficient resource usage. To deal with the challenges posed by the dynamic network environment as well as stringent data freshness requirements, we propose a cooperative satellite-terrestrial resource scheduling scheme to adaptively optimize the sensing interval and resource allocation in STIVN. Leveraging the multicast feature of LEO satellites to serve a group of CAVs simultaneously with less resource consumption, we design sensing interval determination and CAV partitioning methods for reducing the computing and communication resource cost and the resource allocation complexity. Based on the CAV grouping and sensing interval, a reinforcement learning-based algorithm is then developed for the real-time computing and communication resource allocation decisions for different CAV groups to cope with time-varying network dynamics. Simulation results demonstrate that the proposed scheme outperforms benchmark methods in terms of resource usage and reliability performance.

By solving the three research problems, the designed low-complexity, flexible, and adaptive resource management schemes for STIVN can achieve the thesis objective.

1.5 Thesis Outline

The remainder of the thesis is organized as follows: In Chapter 2, we present the literature review of LEO satellite-assisted networks and the state-of-the-art resource management technologies for STIVN. In Chapter 3, reservation-stage multi-dimensional resource management for a single terrestrial cell in STIVN is investigated, where a two-layer resource slicing scheme is proposed to support HD map distribution with coupling decision variables to satisfy service requirements in low complexity. In Chapter 4, reservation-stage resource management across multiple terrestrial cells in STIVN is investigated, where a distributed resource slicing scheme is proposed for flexible resource management across different cells in the reservation stage to support both delay-tolerant and delay-sensitive applications. In Chapter 5, operation-stage resource management is investigated in STIVN, where a cooperative satellite-terrestrial resource scheduling scheme is proposed to support

infrastructure-assisted environment sensing by adaptively optimizing the sensing interval and resource allocation in STIVN for minimizing resource consumption with satisfied service requirements. Finally, we conclude the thesis and discuss future works in Chapter 6.

Chapter 2

Literature Review

In this chapter, the background and related works of STIVN are presented. First, the background, standards, and existing works on the performance analysis of LEO satellite networks are surveyed. Then, different resource management schemes in the STIVN are introduced, including both the reservation stage and operation stage in resource slicing.

2.1 LEO Satellite Networks

As global demand for reliable and widespread connectivity continues to rise, LEO satellite networks have emerged as a critical solution to complement traditional terrestrial networks. These networks offer the potential to overcome limitations such as coverage gaps and high latency in remote areas. This section targets the significance of LEO satellite networks by examining their background, the ongoing standardization efforts, and the performance analysis that underscores their importance in the future of vehicular networks.

2.1.1 Background of LEO Satellite Networks

Satellite networks are composed of satellites in different heights of orbit, including GEO, MEO, and LEO satellites. GEO and MEO satellites are deployed on orbits with high altitudes, which can allow wide coverage with a single satellite and can avoid the relative high-speed movement between a terrestrial user and the satellite. Specifically, GEO satellites have an altitude of 35786 km, and MEO satellites have an altitude from 7000 up to 20000 km. Since the distance is large, the round-trip propagation delay between the user and

the satellite may be over 100 ms [31], which is much larger than the requirement of 5G networks and hard to support some delay-sensitive applications. On the other hand, LEO satellites, with an altitude from 600 to 1500 km, much lower than GEO and MEO satellites, can provide a low delay communication in which the round-trip propagation delay can be less than 10 ms. Hence, LEO satellites are deemed as promising solutions for future network architectures. LEO satellite networks can provide access to users in two modes, namely transparent and regenerative modes. Specifically, for the transparent mode, the satellite only provides connectivity between the users and the terrestrial BS (5G RAN), which simply amplifies and retransmits the signals it receives from the ground without altering or processing the data. For the regenerative mode, the satellite can perform as a base station to provide radio access, which demodulates the received signal, processes the data, and then re-modulates it before transmission. The main properties of the LEO satellite networks are given as follows:

Spectrum: The LEO satellite communications operate mainly in C-band and Ka-band [32]. C-band was the first frequency band allocated for commercial telecommunications via satellites. With appropriate satellite beam layouts, handheld user equipment can be served by LEO in the C-band with a small pathloss. Due to the spectrum scarcity, some satellite operators are moving to the Ka-band, which offers much greater signal bandwidth than the C-band but is much more sensitive to weather conditions. In addition, since the Ka-band has a high frequency, smaller antenna sizes are allowed for users with multi-antenna technology to obtain a large gain. On the other hand, the frequency band for the satellite may coincide with some terrestrial networks' frequency bands. Thus, interference between satellite and terrestrial networks should be considered in future integrated networks.

Coverage: Following the 3GPP release 15, the diameter of a beam for a non-GEO satellite is about 100 - 500 km, in which the footprint of the beams is typically elliptic shape [24]. More specifically, some data from real LEO satellite systems are provided, such as Starlink [20], Telesat [19], and OneWeb [21]. Both satellites in Starlink and Telesat satellite systems have individually shapeable and steerable beams, versus OneWeb which has only fixed beams with the variable diameter of the coverage area [33]. The coverage diameter of a beam from different systems varies from 20 to 500 km. Generally, the coverage of an LEO satellite is several times larger than the beam coverage since each satellite has multiple beams for different adjacent areas.

Mobility and trajectory: Generally, the speed of LEO satellites is up to 7.56 km/s [24], which may be slightly different according to different altitudes. For a beam diameter size of 50 km, the continuous service time for a user is less than 7 s, while for a diameter size of 500 km, the service time is about 65 s. In this case, the speed of the LEO satellites relative to the users is large. Since each satellite operates in a fixed orbit, the trajectory of the

satellites can always be predicted or given directly. However, due to the Earth's rotation, the satellites in one orbit cannot always cover a fixed area even if the deployment of the satellite in such orbit is dense. Therefore, the service time from the satellites in one orbit for one area is limited and the period for the orbit to re-traverse the area is about one day.

2.1.2 Standardization of LEO Satellite Networks

Currently, standards of satellite-terrestrial network integration have been developed to guide the implementation of the satellite-terrestrial network. The 3rd Generation Partnership Project (3GPP) has carried out substantial standardization work on satellite-terrestrial network integration. Technical specification [34] and technical report [35] have been developed to explore the scenarios and identify service requirements for a new architecture where satellite networks act as an extension to terrestrial networks. In technical reports [24, 31], the new radio (NR) is studied to support non-terrestrial networks (NTN). The architecture principles and channel model differences of satellite networks and terrestrial cellular networks are considered and compared. In [36], the service application scenarios are divided into three categories when integrating satellite access networks into 5G systems: continuity, ubiquity, and scalability. The preceding 3GPP standards indicate that the satellite network is an indispensable part of 5G and beyond communication systems to provide ubiquitous access capability and ensure service quality.

Moreover, the European Telecommunications Standards Institute (ETSI) has proposed some standards related to the integration of satellite and terrestrial networks. The definition and classification of scenarios integrating satellite and terrestrial networks are presented in [37]. The role of satellite networks in disaster management and resource requirements for applications such as Earth observation, satellite navigation, and satellite communications are identified [38]. In [39], the regulations that should be followed when introducing cognitive radio (CR) technology in satellite communications are determined, and different scenarios for using CR in the Ka-band are emphasized. The traffic distribution in wireless access networks is considered and a typical scenario of satellite-terrestrial network integration is studied, with satellite backhauling in rural areas [40]. The implementation of the cooperation between satellite networks and femto base stations is refined in [41], where some backhaul solutions using satellite-terrestrial networks are also provided. In addition, [42] proposes the Satellite Independent Service Access Point (SI-SAP) and regulates the physical air interfaces for broadband services in satellite-terrestrial networks.

2.1.3 Performance Analysis on LEO Satellite Networks

To exploit the performance of LEO satellite networks, there exist some research works on network modeling and analysis in various metrics.

The authors in [43], the outage probability is derived for a downlink LEO satellite communication system where multiple LEO satellites are uniformly distributed over a sphere following a homogeneous binomial point process (BPP). In addition, a system throughput maximization problem is addressed using an iterative algorithm to obtain near-optimal solutions. An analytic framework for the uplink coverage probability in hybrid satellite-terrestrial networks is provided in [44], and the impact of the satellite constellation size and the terrestrial base station density on the coverage of the uplink traffic is analyzed. In [45], the meta distribution (MD) of signal-to-interference-and-noise ratio (SINR) for the downlink integrated LEO satellite-terrestrial millimeter-wave decode-and-forward relay network is investigated.

Taking the Earth's rotation and satellite rotation into account, a non-stationary satellite communication system is modeled in [46]. Considering the variations in the distance between satellites and ground users, the instantaneous outage probability (OP) and channel capacity are calculated, as well as the system throughput within finite communication time. In [47], the outage probability is analyzed for an overlay satellite-terrestrial network (OSTN), where an opportunistically selected terrestrial Internet-of-Things (IoT) network assists the primary satellite communications in the presence of hybrid interference from extra-terrestrial sources and terrestrial sources. In [48], network performance is analyzed in different scenarios in STIVN with different satellite constellation settings and terrestrial network deployments.

2.2 Slicing-Based Resource Management in STIVN

Effective resource management is crucial in STIVN due to the dynamic and heterogeneous nature of these networks. As STIVN aims to provide seamless and reliable network services for CAVs across diverse environments, efficiently managing resources becomes essential to meet the varying service demands. This section explores the importance of resource management within general satellite-terrestrial integrated networks (STIN), as well as STIVN, focusing on strategies implemented during the operation stage and the reservation stage.

2.2.1 Operation-Stage Resource Management

In studying operation-stage resource management, there are a few research works focusing on the challenges of distinct features in satellite and terrestrial networks. In this subsection, we review the state of the arts in resource allocation in the operation stage in STIVN, including user association, radio resource allocation, and multi-dimensional resource orchestration in radio access networks (RAN).

User Association

As heterogeneous networks where the satellite and the terrestrial networks have different properties in providing the services, the radio access mode selections are studied in a wide range. The authors in [49] investigate the access mode selection (AMS) in STIVN to support multifarious vehicular services with enhanced network flexibility and reliability. A learning-based algorithm for an adaptive AMS scheme is proposed to support different services in highly dynamic vehicular networks by exploiting the advantages of different networks, such as low propagation delay for terrestrial networks and less handover for satellite networks. In [50], a network access selection scheme is proposed for the high-speed railway wireless system in STIN to guarantee network continuity for safety services, which can effectively reduce the impact of service interruptions resulting from frequent handover. A Q-learning algorithm is proposed to select the access mode for each service according to the channel condition of each access node, the current bit error rate, and the end-to-end latency.

Inevitably, the direct links between the satellite and users do not always have desirable channel conditions or accessible links. Therefore, a relay selection in the STIVN has been studied, where the satellites are part of the backhaul networks. Several studies are focusing on improving the network capacity by leveraging satellites to provide a backhaul connection. In [51], the authors study a data offloading problem in the STIN, where uplink transmissions are considered for users to access the network via the macro BS, traditional small cell BSs, or LEO-based terminals. The satellite is considered as a relay node to provide a coverage extension and backhaul connectivity. In this case, it is impossible to construct all fiber-equipped or stable wireless backhaul links for every small cell in the STIN. The formulated optimization problem can maximize the sum rate of all users and the number of accessed users in the terrestrial networks by selecting the user association and resource allocation of multiple cells. Moreover, some studies focus on IoT networks, where the designed scheme should be cost-efficient as well as support high data rates and low delay. Based on deep actor-critic reinforcement learning, the network control and resource

allocation for the STIN were jointly optimized in [52]. The authors propose a negotiation of a collective bargaining policy with the spectrum provider from IoT networks to maximize the long-term utility by selecting either terrestrial networks or satellite networks. An auction-based relay node selection algorithm is proposed in [53] for IoT nodes in the STIN, where the satellite networks have large coverage and can alleviate the spectrum scarcity of terrestrial networks. Since some IoT nodes cannot be directly linked to the satellites, the base stations can be regarded as relay nodes between the connection of IoT nodes and the satellite and use the non-orthogonal multiple access (NOMA) scheme for cooperative transmission. The objective is to maximize the sum transmission rate for all primary IoT nodes by using VCG auction to choose the relay nodes among terrestrial base stations with the constraint of the minimum requirement for the remaining IoT nodes.

Radio Resource Allocation

In the field of wireless communications, many studies are focusing on radio resource management to improve the network performances for the STIN.

In [54], two service function chain (SFC) orchestration schemes are designed to maximize service acceptance and ensure satellite load fairness. Specifically, a load-aware heuristic algorithm and a hierarchical reinforcement learning approach are proposed based on Graph Attention Networks (GAT) to improve service acceptance, satellite load fairness, and the robustness of dynamic LEO satellite networks. In [55], the authors propose a cooperative multigroup multicast transmission scheme in the STIN where the pre-grouping for users is applied to different content, and the multicast is adopted for each group with the same transmission signal. Multiple BS and one satellite provide multicast services in a cooperative manner and each user is able to connect to the base station and the satellite simultaneously. To cope with the challenges of intra-cell interference, inter-cell interference, and interference from the satellite, the authors propose a beamforming vector adjustment of the terrestrial base station and the satellite to maximize the minimum signal-to-interference and noise ratio among all users. The authors in [56] propose a cooperative resource management algorithm for IoT devices. The satellite and terrestrial networks support massive IoT devices by sharing the same spectrum resource, which can effectively improve spectrum efficiency. Considering the interference, an optimization problem is formulated to minimize the total transmit power with the constraint of maximal outage probability by controlling beamforming vectors. The authors in [57] propose a resource allocation scheme for NOMA-based STIN. Specifically, the user can only connect to one access node, i.e. the BS or the satellite, where BSs have limited resources, and cannot serve all users within its coverage. To maximize the system capacity with minimal overall power consumption, joint user

association, user pairing for NOMA, and beamforming vector are optimized.

An intelligent spectrum management framework for the STIN is proposed in [58] based on software-defined networking (SDN) and artificial intelligence (AI) to deal with the difficulties in obtaining an accurate and timely perception of satellite communications. Specifically, SDN transforms heterogeneous networks into integrated networks with interoperability for two networks. Therefore, cooperative management for the STIN can be designed through a centralized controller with AI-enabled environment prediction to configure the networks for optimal resource allocation. A load- and mobility-aware cooperative content delivery scheme is proposed in [59] for STIVN where the integrated networks can provide global connectivity, enhanced flexibility, and reliability for vehicular networks. The vehicle can connect to multiple nodes to receive the content with a multi-connectivity scheme. Given the information on cached content, the authors jointly optimize the association, bandwidth allocation, and content delivery strategy to minimize the overall transmission delay by considering user fairness, load balancing, and vehicle mobility. The authors in [60] provide a stochastic delay analysis to satisfy the delay requirement of the space traffic and guide the network configuration in the STIN. In this work, the satellites can either serve in a direct mode or a relay node for the indirect mode due to the intermittent connection. The authors model a heterogeneous offloading process and build a series-parallel queuing model with the consideration of the propagation delay. The upper bounds of the stochastic delay can therefore be derived according to the geometric relationship between the arrivals and leftover services. In [61], a non-orthogonal multiple access-based age of information minimization is investigated in STIN by selecting base station-terrestrial user pairs and optimizing transmit power. By jointly considering mobility patterns and fluctuating loads in STIVN, a time-varying resource graph is proposed in [62] to model dynamic resources for better resource allocation. In [63], the authors introduced a cyber twin-assisted joint mode selection and dynamic pricing scheme for efficient network management in ultra-dense integrated satellite-terrestrial networks, where the cyber twin acts as an intelligent agent aiding in mode selection decisions.

Multi-Dimensional Resource Orchestration

Some recent works study not only the management of radio resources but also the combination of communication, computing, and caching resources, which are called 3C resources. By jointly considering the three types of resources in networks, the STIVN architecture can provide more flexible resource allocations to enhance network performance by leveraging different advantages in integrated networks.

Focusing on communication and caching resource management, aiming to reduce the

content delivery delay in caching-enabled networks, caching node selection and the content placement in both satellites and terrestrial network nodes are studied in [64–68], where popular contents can be cached in different places, taking advantage of the large coverage of satellites and high transmission data rate of terrestrial nodes, to reduce the backhaul utilization, as well as enhance users' quality of experience. Specifically, the authors in [64] consider cache-enabled networks with one LEO satellite and several access points, which can provide the traffic offloading from terrestrial networks to reduce network stress. Some popular contents can be cached in the access points (APs) and LEO satellite in a coded block-based format which can reduce the backhaul utilization. The objective is to maximize energy efficiency by adjusting the cached block placement, power allocation for APs, and whether the content is cached on the satellite. In [65], the authors consider the dynamics of the satellites and user demands in the STIN, where the satellites only provide limited service time for each user. To solve this problem, a learning-based joint cache placement and content delivery algorithm is proposed to minimize the long-term overall transmission delay by providing a long-term decision of cache placement and a short-term decision of user association with either satellites or terrestrial base stations. In [66], the authors propose a cooperative strategy between nodes for both downlink and uplink transmission in the STIN. In order to decrease the transmission delay, the users can be selected as relays in the uplink to ensure the quality of communications or the caching devices for downlink by caching popular contents A joint cache placement and content delivery algorithm is proposed in [67]. Considering the cache-enabled centralized-RAN (C-RAN) architecture, the authors jointly discover the impact of the cache placement, user clustering in a large timescale, and content delivery by controlling the multicast beamforming in a short timescale and aim to minimize the long-term power consumption by decoupling and solving the optimization problems. The authors in [68] study the caching strategy in the STIN where the satellites can provide spacious coverage, but also have dynamic propagation links and time-varying topology. To solve this problem, a novel caching-assisted content distribution scheme is proposed in the STIN to guarantee users' quality of experience. In [69], a data-model co-driven approach is proposed for cooperatively scheduling communication and computing resources in both terrestrial and satellite networks to satisfy stringent delay and reliability requirements of vehicular services with efficient resource usage.

In addition, several studies aim to jointly manage 3C resource for better network performance in STIVN. An SDN-defined STIN is proposed in [70] to jointly orchestrate 3C resources in integrated networks to maximize resource efficiency, leveraging the global coverage, continuous services, and increased resources from the STIN. Considering the network dynamic, heterogeneity, and complexity, it is hard for network controllers to make resource allocation decisions through traditional optimization methods. Therefore,

a deep Q-learning approach is proposed to solve the resource allocation problems in the SDN-defined STIN. A cost-effective hybrid computation offloading paradigm in STIN is proposed in [71], which aims to minimize total computation offloading costs while satisfying diverse user latency requirements and adhering to satellite energy constraints, while a hybrid proximal policy optimization-based algorithm to support task offloading in STIN for remote IoT users is developed in [72] for minimizing offloading delay. The scheduling of tasks in a space-air-ground integrated network is investigated in [73]. In the scenario considered, tasks are collected and offloaded by UAVs, and tasks need to be processed by the UAV, based station, or remote satellite using a deep reinforcement learning approach. In [74], the authors explore Internet service provisioning in space-air-ground integrated networks through dynamic network function mapping and scheduling. The authors in [75] consider resource management with 3C resources to support IoT networks. Due to the difficulties of meeting all industrial IoT requirements from terrestrial networks and leveraging the different properties of the satellites and terrestrial base stations in coverage and transmission rate, the STIN is considered for industrial IoT networks. The proposed algorithm aims to minimize the total system delay while maximizing resource utilization by matching IoT nodes to the proper base station or satellite for the 3C resource allocation. In [76], the authors investigate the integrated 3C resource management, where the blockchain technique and Nash bargaining game were adopted, considering the serving period of the satellites, to improve the throughput of users while guaranteeing fairness and data security. Furthermore, the authors propose a relay model in [77] for downlink transmission considering the intermittent communication of satellites, which can improve the throughput performance.

2.2.2 Reservation-Stage Resource Management

In this subsection, we will discuss the related work on reservation-stage resource management in STIVN, particularly within the context of STIVN. Unlike terrestrial networks, resource slicing in STIVN presents unique challenges due to the heterogeneous nature of the network, which combines satellite and terrestrial components. This heterogeneity introduces additional complexity in managing and reserving resources effectively. Many researchers focus on developing resource slicing techniques tailored to the distinct characteristics of STIVN, aiming to develop resource management frameworks and optimize the reservation of resources.

Significant progress is emphasized in [78] for network slicing frameworks designed to improve performance and resilience in satellite networks, including the definition, modeling, orchestration, and deployment of multiple slices and infrastructures. To realize flexible,

reliable, and scalable network resource management, an SDN-based framework for the space-air-ground integrated networks is proposed in [79] where the framework explores AI-based solutions to facilitate efficient resource slicing. Specifically, to ensure user satisfaction with different services, resource slicing enables the coexistence of multiple slices using independent virtual networks with a shared physical network infrastructure. In [80], a multi-domain dynamic network slicing framework is proposed utilizing a multi-path, multi-hop delay model specifically designed for delay-sensitive applications. Moreover, [81] introduces a resilient slicing design that incorporates the planning and provisioning of redundant resources to mitigate potential failures. In [82], the authors present a resource slicing solution on enhanced Mobile Broadband (eMBB) services within STIVN by ensuring different grades of QoS in dynamic scenarios, along with the queueing analysis to minimize the latency for resource allocation by adjusting the network selection. The authors in [83] study resource slicing in the STIVN by developing an online control framework to make decisions on the request admission and scheduling, UAV dispatching, and resource slicing for different services. Specifically, a dynamic slicing algorithm can allocate the resources from different infrastructures for different slices with the online decision aiming to maximize the time-averaged throughput with minimum UAV dispatching cost while stabilizing the time-average queue. In [84], the authors investigate a resource slicing problem to reserve resources in satellites to achieve efficient resource provisioning. To address the challenges of non-stationary service demands, inaccurate prediction, and satellite mobility, an adaptive digital twin-assisted resource slicing scheme is proposed for robust and adaptive resource management. Based on the two-stage resource management framework, a learning-based network management method is proposed in [85]. With the characteristic of the STIVN, where the available network resources and user demands are dynamic, the authors investigate the joint resource slicing and scheduling algorithm to manage the spectrum resources in the STIVN for supporting two different services to minimize the long-term overall system cost. A hybrid data-model co-driven resource slicing mechanism has been proposed in [86] to cope with spatiotemporal dynamics in service demands considering multiple beam-coverage areas. The authors in [87] create low-delay, high-throughput, and wide-coverage slices within space-air-ground integrated networks where both centralized and distributed multi-agent deep deterministic policy gradient (DDPG)-based algorithms are developed to jointly optimize service delay, throughput, and coverage area.

Chapter 3

Reservation-Stage Multi-Dimensional Resource Management

In this chapter, we investigate reservation-stage resource management in STIVN, in which both communication and caching resources are managed in LEO satellites and terrestrial BS to satisfy the service requirements of HD map distribution in vehicular networks. Leveraging the heterogeneity of satellite and terrestrial networks, a novel resource slicing architecture in STIVN is developed with the establishment of two slices for multicast and unicast transmissions with different transmission requirements. To address the challenges posed by spatiotemporal dynamics in service demands and time-varying satellite availability, we formulate the resource slicing problem into a long-term optimization problem and propose a two-layer resource slicing (TLRS) scheme for multi-dimensional resource management in low complexity. Specifically, a hybrid data-model co-driven approach is developed, including a swap-based matching algorithm to determine the multicast and caching decisions within each slicing window in the inner layer, and a hybrid proximal policy optimization (HPPO)-based reinforcement learning algorithm to adaptively adjust the slicing window length and reserved communication resources in the outer layer. Simulation results demonstrate that the proposed TLRS scheme in STIVN can effectively guarantee the service requirements with efficient resource usage and lower delay performance.

3.1 Background and Motivations

With the rapid development of intelligent transportation systems, CAVs have been regarded as important components to enhance transportation efficiency and safety by enabling

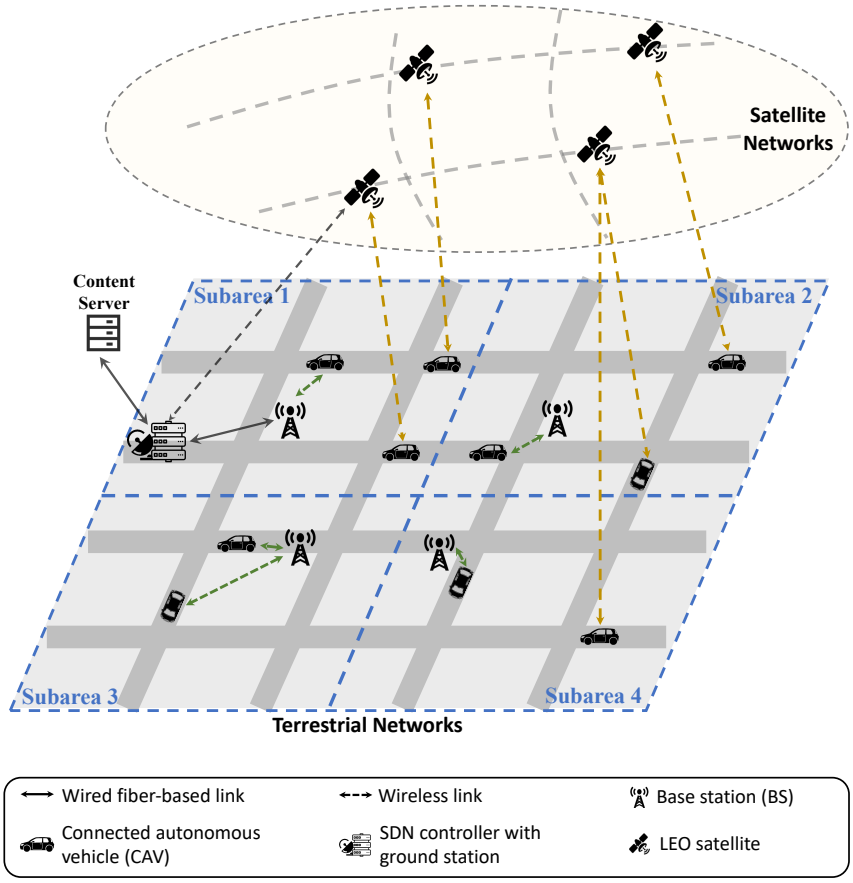


Figure 3.1: Illustration of STIVN scenario.

communication between vehicles and infrastructure [1, 5]. This advancement necessitates the development and implementation of HD maps, which provide detailed and precise geographical information, including lane models, traffic signs, and road geometry, essential for the accurate localization and enhanced navigation of CAVs [88]. However, the data volume of HD maps is relatively large in comparison with traditional maps, which makes it impractical to store the entire HD maps onboard [89]. In this case, the full potential of HD maps can only be realized with robust network support to ensure real-time HD map delivery, emphasizing the crucial role of vehicular networks to support network-reliant services [8].

To support HD map distribution, while effective in urban areas, terrestrial vehicular networks may fail to provide adequate coverage in remote or rural areas due to high

deployment and maintenance costs, or some geographical constraints. LEO satellites, thanks to their large communication coverage and dense deployment, show great potential in assisting the existing terrestrial vehicular networks in providing ubiquitous connectivity and seamless service provision, especially in remote areas. By cooperatively leveraging the complementary advantages of satellite and terrestrial networks, STIVN, as shown in Fig. 3.1, enables CAVs to access necessary data anywhere at any time [16]. In addition, by deploying edge servers along with both satellites and BSs, HD map files can be stored in the edge server to reduce the transmission delay instead of fetching files from the content server. In this case, to facilitate HD map distribution service, joint management of communication and caching resources from both satellite and terrestrial networks are necessary in vehicular networks.

In order to guarantee the service requirement of HD map distribution, resource slicing will play a pivotal role in STIVN [90]. Unlike conventional one-fit-all resource management, resource slicing enables the construction of multiple virtual networks, referred to as network slices, on top of the shared physical network infrastructure to realize resource isolation for satisfying different service requirements under dynamic service demand. Specifically, the reservation stage is crucial for slicing-based resource management, which configures network slices by reserving resources in each slice in a large timescale for different services, to avoid resource competition among different slices. To maintain the effectiveness of slices, ensuring that they can satisfy service requirements in dynamic network conditions with efficient resource utilization, adaptive adjustments to slices at specific intervals are crucial, where the adjustment intervals are referred to as slicing windows. On the other hand, within each slicing window, infrastructures can further schedule the resources in real-time operation stage to adapt to dynamic network environments. With resource slicing, the complexity of overall resource management can be reduced by introducing two time-scale orchestrations, i.e., reserving resources in each slice in infrastructures and scheduling resources to individual CAVs.

In this work, we investigate resource slicing for HD map distribution in STIVN and focus on the reservation stage to guarantee long-term service requirements. However, adaptive multi-dimensional resource slicing in STIVN remains an open problem with the following challenges. First, satellite networks have distinct characteristics compared to terrestrial networks. Consequently, satellites and terrestrial networks can leverage their respective advantages to provide different transmission modes for HD map distribution. Specifically, the top-down transmission feature of LEO satellites can facilitate the distribution of popular HD maps by adopting multicast transmission, which allows satellites to periodically serve multiple CAVs simultaneously with similar channel conditions instead of the traditional request-response mode. Considering the multicast interval time of the periodicity, the

transmission delay should be more stringent compared with unicast transmission to satisfy the service requirement. Meanwhile, the unicast transmission from satellites is still applicable to individual CAVs. On the other hand, since channel conditions can vary significantly among different CAVs within the coverage area of a BS, terrestrial networks provide unicast transmission for HD map distribution. Therefore, considering the coupling relationships between the caching resources and communication resources, jointly managing multi-dimensional resources for multicast and unicast transmission from both satellite and terrestrial networks is complex to efficiently reserve resources in each slice. Second, due to the mobility of LEO satellites, the number and positions of satellites covering a given area change over time, which results in the time-varying available resources in STIVN, posing a challenge to conventional resource slicing methods designed for cases with fixed resources. Third, the slicing window length should be adjusted to adapt to both the temporal dynamics of service demand and time-varying available resources in STIVN. Generally, a large slicing window may incur low resource utilization, especially with some burst traffic. For example, the resources should be reserved to accommodate the service demand with the highest traffic within the slicing window to satisfy service requirements, where the resources may therefore be wasted most of the other time. On the other hand, a short slicing window means that the resource reservation can adapt to a dynamic environment more precisely, with high slicing reconfiguration costs. With the LEO satellites, leveraging dynamically available resources has the potential to accommodate dynamic traffic with a lower reconfiguration cost and high resource utilization. However, the resource reservation and slicing window length mutually affect resource slicing in STIVN, resulting in complex decision-making.

To solve the above-mentioned challenges, we propose a multi-dimensional resource slicing scheme for STIVN in low complexity to satisfy the service requirements of HD map distribution. Our objective is to reserve resources from both terrestrial and satellite networks to minimize the overall cost related to resource usage and delay while maximizing the service demand that can be accommodated to meet the service requirements. Considering the different features of multicast and unicast transmission, we design a novel resource slicing architecture in STIVN by establishing two slices accordingly to jointly serve the HD map distribution service. Due to the interdependency of multi-dimensional resources in each slice, it is highly complex for the joint optimization of different types of resources and the slicing window length. Therefore, a two-layer resource slicing approach is developed, which is a data-model co-driven approach, to find the optimal solutions with less complexity. Specifically, considering the time-varying available resources and spatial-temporal dynamic service demand, we design a reinforcement learning-based algorithm in the outer layer to decide the communication resources reserved in each network slice in each access point and the slicing window length to find the long-term optimal performance. Then, in the

inner layer, according to the reserved communication resources in each slice, a swap-based matching algorithm is derived to determine the optimal multicast and caching decisions in each slicing window. The main contributions of this work are summarized as follows:

- We investigate the resource slicing problem in STIVN to support HD map distribution service, where both satellites and BSs in terrestrial networks can accommodate service demands from CAVs. To satisfy the service requirement while leveraging the multicast feature of satellites, we develop a resource slicing architecture in STIVN with two slices for the service. Then, we formulate the multi-dimensional resource slicing problem in STIVN, aiming to satisfy the service requirement with efficient resource usage.
- We propose the TLRS scheme to realize an adaptive multi-dimensional resource slicing in STIVN. In this scheme, the original problem is decoupled into two subproblems to reduce the complexity of addressing the sequential decision-making problem. Then, the slicing window length and communication resource reservation are determined in the outer layer and multicast and caching decisions are made in the inner layer.
- We develop a hybrid proximal policy optimization (HPPO)-based reinforcement learning algorithm to determine the non-homogeneous actions, i.e., the discrete action for slicing window length and continuous action for communication resource reservation, at the beginning of each slicing window, for minimizing the long-term system cost. Then, a swap-based matching algorithm is designed to optimize the multicast and caching decisions within each slicing window given the slicing window length and reserved communication resources.

The remainder of this chapter is organized as follows. The considered network scenario, designed resource slicing architecture, and system models are introduced in Sec. 3.2. The problem is formulated in Sec. 3.3, followed by the proposed two-layer resource slicing scheme. Then, a swap-based matching algorithm is designed for determining multicast and caching files in Sec. 3.4, and a reinforcement learning-based algorithm for resource reservation and slicing window length is provided in Sec. 3.5. Performance evaluation is given in Sec. 3.6. Finally, in Sec. 3.7, we conclude this chapter.

3.2 Network Scenario and System Model

In this section, we first introduce our considered STIVN scenario with the slicing-based resource management architecture and then present the system models. Some useful

Table 3.1: Summary of Notations

Notation	Definition
$\mathcal{BS}, \mathcal{S}, \mathcal{AP}, \mathcal{F}$	Sets of terrestrial base stations, LEO satellites, all access points, and HD map files
B_{ap}^{r}	Available resource r at AP ap
$\lambda_{b,f}^{k,t}$	Service demand intensity for file f within the range of BS b at time slot (k,t)
ω_k	Length of slicing window k
$v_{ap,x}^{k}$	Reserved resource ratio of resource r in AP ap to slice x in slicing window k
$u_{ap,f}^{k}$	Caching indicator for file f in AP ap in slicing window k
$m_{ap,f}^{k}$	Indicator of multicast for file f in AP ap in slicing window k
$\tilde{\lambda}_{ap,x,f}^{k,t}$	Accommodated service demand of file f from slice x of AP ap at time slot (k,t)
$D_{ap,x}^{\mathrm{Q},k,t}$, $D_{ap}^{\mathrm{P},k,t}$, D^{BH}, $D_{ap,x,f}^{k,t}$	Queueing and transmission delay, propagation delay, backhaul delay, and overall delay
β_1, β_2, β_3, β_4	Weight of resource reservation cost, delay cost, slice reconfiguration cost, and demand penalty

notations are listed in Table 3.1 for easy reference.

3.2.1 Scenario Description

The considered STIVN scenario is shown in Fig. 3.1, where CAVs in the target area can be served by either terrestrial BSs or LEO satellites to receive HD map files. LEO satellites are equipped with steerable antennas where one beam can point to the target area when the elevation angle constraint is met. The target area consists of different subareas. When CAVs enter one subarea, they will request one of the HD map files associated with that subarea, where different CAVs may have different HD map file demands. The overall delay of receiving the corresponding HD map file should be less than $D^{\max}$ with the maximum service violation probability ε. The set of all HD map files for the target area is denoted by $\mathcal{F} = \{1, 2, ..., F\}$, where each HD map file has a data size of κ. A content server is deployed away from the area, which stores all HD maps for the area.

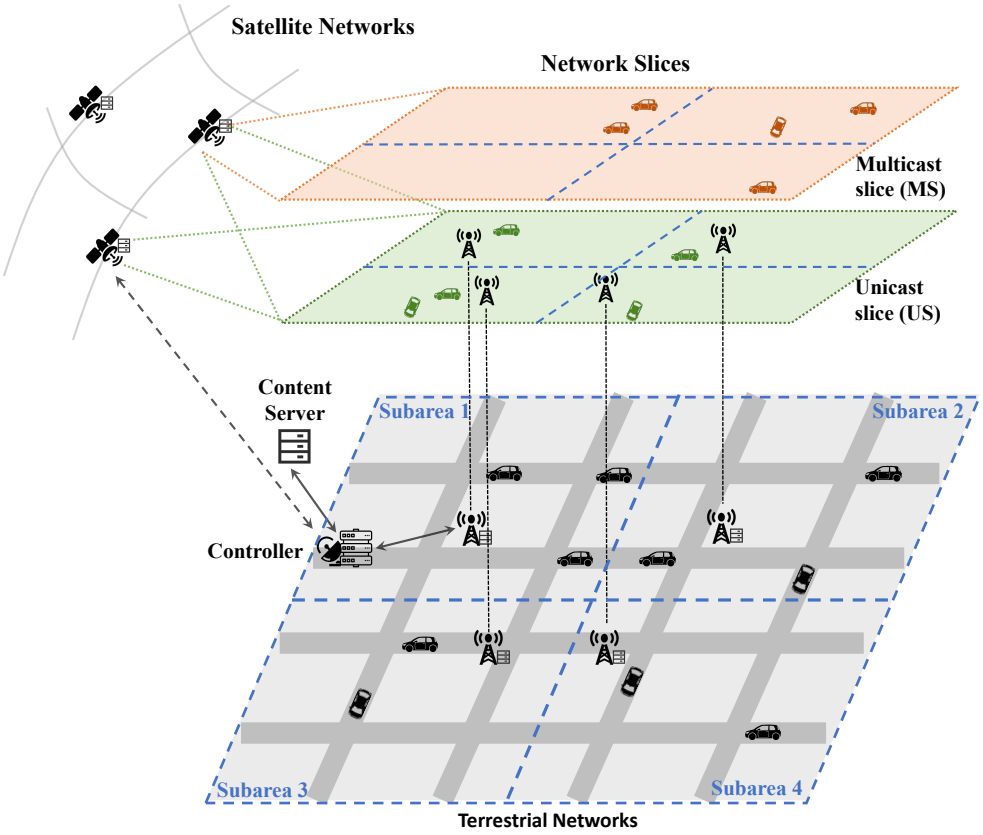


Figure 3.2: Resource slicing architecture in STIVN.

Denote the set of BSs in the area and LEO satellites in a constellation by $\mathcal{BS}$ and $\mathcal{S}$, respectively. Let $\mathcal{AP} = \mathcal{BS} \cup \mathcal{S}$ represent the set of all access points (APs), where each AP is equipped with caching capabilities for data storage. The total available resource r at AP ap are denoted by B_{ap}^{r}, where $r \in \mathbf{r} = \{\mathrm{cm}, \mathrm{ch}\}$ indicating spectrum bandwidth of communication and caching capacity, respectively. An SDN controller is deployed to manage the resources of all APs serving the area [79].

3.2.2 Resource Slicing Architecture

To satisfy the service requirements of HD map distribution, resource slicing is adopted to provide isolated resources for the individual service. Leveraging the advantages of satellite and terrestrial networks, two different transmission modes in STIVN can be derived, namely

multicast and unicast transmission modes. Specifically, benefiting from the high altitude and large coverage of LEO satellites, a periodical multicast transmission of HD map files can be leveraged to deliver one file to multiple CAVs simultaneously to reduce the resource cost, especially for popular files. However, due to the periodic multicast of HD map files where all file requests from CAVs within the multicast interval should satisfy the service requirement, the transmission delay is stringent. In addition, the unicast transmission through conventional request-response mode is still applicable for satellites to deliver HD map files. On the other hand, considering the differentiated channel conditions between the BS and CAVs within its coverage, the multicast is not proper for terrestrial networks to transmit files. In this case, only unicast mode is used for BSs to deliver HD map files to individual CAVs, which can provide a high transmission rate with low delay. Since different transmission modes have different features, establishing one slice for both two transmission modes can hardly guarantee the service requirements by sharing resources in the same slice [91]. Therefore, as shown in Fig. 3.2, we build two slices for HD map distribution to realize the resource isolation for different transmission modes, namely multicast slice (MS) and unicast slice (US), respectively.

The resource slicing in STIVN operates in a time-slotted system. Considering that the slicing decisions should be timely adjusted to maintain the high slicing performance, the time is separated into a number of slicing windows $k \in \{1, 2, ...\}$. Each slicing window consists of ω_k time slots, where time slot $t \in \{1, 2, ..., \omega_k\}$ has the duration τ. The maximum length of a slicing window is set as $\omega_{\max}$ time slots. For simplicity, let (k, t) represent the t-th time slot in slicing window k. Due to the time-varying available resources resulting from the mobility of satellites as well as the spatiotemporal dynamic of service demand in the STIVN, the slicing window length should be adaptively adjusted for each slicing window accordingly to adapt to the dynamic network environment.

Let $P_{ap}^{k,t}$ denote the coordinate of AP ap at time (k, t) in an Earth-centered Earth-fixed (ECEF) coordinate system, where the center of the Earth at its origin. The coordinates of each LEO satellite remain the same in each time slot and change over time due to their mobility. Let $a_{ap}^{k,t}$ denote the indicator of the accessibility of AP ap for the target area at time slot (k, t), where $a_{ap}^{k,t} = 1$ indicates that the AP ap can serve the area and $a_{ap}^{k,t} = 0$ if otherwise. In this case, let $\mathcal{S}^k = \{ap \mid \sum_{t=1}^{\omega_k} a_{ap}^{k,t} > 0, \forall ap \in \mathcal{S}\}$ and $\mathcal{AP}^k = \mathcal{BS} \cup \mathcal{S}^k$ denote the set of satellites being able to cover the target area and all APs being able to serve CAVs in the area in slicing window k, respectively.

Assuming that service demands of HD maps follow a non-homogeneous Poisson process (NHPP) [92], where the service demand intensity (in packets per second) varies over time to depict the busy and non-rush periods and is fixed within each time slot. Let $\lambda_f^{k,t}$ denote the entire service demand intensity of file f in the target area at time slot (k, t), which is

the sum of the service demand intensity within the coverage of each BS and the area not covered by terrestrial networks, represented by

$$\lambda_f^{k,t} = \sum_{b \in \mathcal{BS} \cup \{0\}} \lambda_{b,f}^{k,t}. \tag{3.1}$$

Here, $\lambda_{b,f}^{k,t}$ is the service demand intensity for file f within the coverage of BS b at time slot (k,t), where $\lambda_{0,f}^{k,t}$ indicates the intensity of file f that cannot be served by any BSs.

In this work, we mainly focus on the reservation-stage resource management in each slice to guarantee service requirements in long-term. At the beginning of each slicing window, the SDN controller can predict the future service demand[1]and make the communication and caching resource reservation decisions for each AP involved in the slicing window. The decisions remain unchanged within each slicing window. Let $v_{ap,x}^k \in [0,1]$ denote the reserved communication resource ratio to all available communication resources from AP ap to slice $x \in \mathbf{x} = \{0,1\}$ in slicing window k, where $x = 0$ representing MS and $x = 1$ for US, respectively. Let $u_{ap,f}^k \in \{0,1\}$ denote the caching decision, where $u_{ap,f}^k = 1$ if file f is cached in AP ap and $u_{ap,f}^k = 0$ for otherwise.

Given the communication resource reservation decision $v_{ap,x}^k$, the transmission rate for delivering packets (in packets per second) from AP ap to CAV g within its coverage can be expressed as

$$\xi_{ap,x,g}^{k,t} = \frac{a_{ap}^{k,t} v_{ap,x}^k B_{ap}^{\mathrm{cm}}}{\kappa} \log_2(1 + \frac{p_{ap} h_{ap,g}^{k,t}}{\sigma^2}), \tag{3.2}$$

where p_{ap} denotes the transmission power of ap, σ^2 is the noise power, and $h_{ap,g}^{k,t}$ is the channel gain.

3.2.3 Effective Capacity Analysis

The performance of each slice in each time slot within a slicing window is significantly influenced by resource reservation decisions, which directly affect metrics such as delay and resource usage. To characterize the HD map file delivery procedure, We model the service in each slice in each AP as a queueing system. Specifically, if the required file is cached in the AP, a transmission queue is for transmitting the file to CAVs; otherwise, a backhaul transmission should also be included for non-cached files transmitted from the

[1]Note that the service demand prediction is not the primary focus of this work and we assume that the service demand in the following $\omega_{\max}$ time slots can be perfectly predicted by adopting several algorithms, such as recurrent neural networks (RNN) [93] and long-short term memory (LSTM) [94].

content server to the AP followed by a transmission queue for transmitting the file to CAVs. Let $D_{ap,x}^{\mathrm{Q},k,t}$ denote the queueing and transmission delay from AP ap to CAVs in the target area at time slot (k,t). Due to the time-varying nature of wireless channels and stochastic service demand, it is difficult to guarantee the deterministic delay performance of queues in each slice while the average delay performance is insufficient for depicting the service requirement satisfaction within each slice.

To model a statistical delay performance of the service with a violation probability, the effective capacity emerges as a valuable approach to model the statistical performance [95]. Specifically, it can facilitate the evaluation of delay-bounded violation probability that delay D exceeds the bounded queueing and transmission delay $D_{ap,x}^{\mathrm{Q},k,t}$, presented by

$$P(D \geq D_{ap,x}^{\mathrm{Q},k,t}) \approx e^{-\theta C_{ap,x}^{k,t} D_{ap,x}^{\mathrm{Q},k,t}}, \tag{3.3}$$

where θ is the exponential decay rate of the violation probability defined in the context of large derivation theory [95], where we set the identical value of θ for queues in different APs. In addition, the effective capacity $C_{ap,x}^{k,t}$, which can quantify the maximum service demand that can be accommodated by reserved resources, can be defined as

$$C_{ap,x}^{k,t} = -\frac{1}{\theta} \log \left(\mathbb{E} \left\{ e^{-\theta \xi_{ap,x,g}^{k,t}} \right\} \right). \tag{3.4}$$

Hence, with a given resource reservation decision $v_{ap,x}^{k}$ and the delay violation probability ε, the bounded queueing and transmission delay can be obtained, given by

$$D_{ap,x}^{\mathrm{Q},k,t} = -\frac{\log(\varepsilon)}{C_{ap,x}^{k,t}\theta}. \tag{3.5}$$

3.2.4 Performance Analysis for Different Slices

Based on the effective capacity theory, we analyze the delay performance of different slices. First, MSs are only established in satellites and provide multicast service. In this case, the propagation delay can not be negligible. Let $m_{ap,f}^{k}$ denote the indicator of multicast service, where $m_{ap,f}^{k} = 1$ represents that file f is multicast by satellite ap in slicing window k, and $m_{ap,f}^{k} = 0$, otherwise. Considering the periodical multicast mechanism, when a CAV requests a file, the request will be processed at the next multicast starting time. Let T^{M} denote the interval time of two multicasts. The average overall delay of receiving file f from satellite ap in MS at time slot (k,t) can be represented by

$$D_{ap,0,f}^{k,t} = D_{ap,0}^{\mathrm{Q},k,t} + \frac{T^{\mathrm{M}}}{2} + D_{ap}^{\mathrm{P},k,t} + (1 - u_{ap,f}^{k}) D^{\mathrm{BH}}. \tag{3.6}$$

Here, we consider an average waiting duration of $T^{\mathrm{M}}/2$ for processing the request between two multicast events according to the Poisson feature of the request. Let $D_{ap}^{\mathrm{P},k,t}$ indicate the propagation delay from satellite ap, and D^{BH} represent the backhaul delay between the content server and satellite.

Then, to model the delay performance of US in the area, the overall delay from US in AP ap to CAVs at time slot (k,t) is denoted by

$$D_{ap,1,f}^{k,t} = D_{ap,1}^{\mathrm{Q},k,t} + \mathbb{1}_{ap\in\mathcal{S}} D_{ap}^{\mathrm{P},k,t} + (1-u_{ap,f}^{k}) D^{\mathrm{BH}}, \tag{3.7}$$

where $\mathbb{1}_{condition} = 1$ if *condition* is true and $\mathbb{1}_{condition} = 0$ if otherwise.

The accommodated service demand of file f from AP ap in slice x can be denoted by $\tilde{\lambda}_{ap,x,f}^{k,t}$, where the details will be presented in Sec. 3.4.1. Note that BSs only establish US, i.e., $\tilde{\lambda}_{ap,0,f}^{k,t} = 0, \forall ap \in \mathcal{BS}$.

3.3 Problem Formulation and TLRS Scheme Design

3.3.1 Problem Formulation

Considering the time-varying service demand and available resources, we aim to minimize the long-term overall system cost while satisfying the service requirement of HD map distribution. The system cost includes the resource reservation cost from both terrestrial BSs and LEO satellites, the delay cost, the slice reconfiguration cost, and the service demand penalty.

Resource Cost

The resource reservation cost is the time slot-averaged cost of all resources in different slices from BSs and LEO satellites serving the target area. Specifically, the resource reservation cost in slicing window k is given as

$$H_{k}^{\mathrm{res}} = \frac{1}{\omega_k} \sum_{t=1}^{\omega_k} \sum_{ap\in\mathcal{AP}^k} a_{ap}^{k,t} \left(\sum_{x\in\mathbf{x}} \alpha_{ap}^{\mathrm{cm}} v_{ap,x}^{k} B_{ap}^{\mathrm{cm}} + \alpha_{ap}^{\mathrm{ch}} \sum_{f\in\mathcal{F}} u_{ap,f}^{k} \kappa \right), \tag{3.8}$$

where α_{ap}^{r} is the unit cost of resource r for AP ap.

Delay Cost

Considering that the delay requirement for HD map distribution is not stringent, however, the CAVs always pursue a lower delay for receiving the content when requesting. In this case, we aim to find a minimize the delay for HD map distribution by efficiently reserving resources and effectively developing multicast policies to enhance the user experience in terms of delay. Therefore, the delay cost in slicing window w is defined by

$$H_k^{\mathrm{d}} = \frac{1}{\omega_k} \sum_{t=1}^{\omega_k} \frac{\sum_{f\in\mathcal{F}} \sum_{ap\in\mathcal{AP}^k} a_{ap}^{k,t} \sum_{x\in\mathbf{x}} D_{ap,x,f}^{k,t} \tilde{\lambda}_{ap,x,f}^{k,t}}{\sum_{f\in\mathcal{F}} \sum_{ap\in\mathcal{AP}^k} a_{ap}^{k,t} \sum_{x\in\mathbf{x}} \tilde{\lambda}_{ap,x,f}^{k,t}}, \tag{3.9}$$

which represents the average delay of HD map distribution service in the target area supported by each slice of all APs.

Slice Reconfiguration Cost

Between different slicing windows, the controller can adjust the amount of communication resources reserved in slices from different APs to adapt to dynamic service demand and satellite mobility. However, the process of resource adjustment incurs slice reconfiguration costs. Since releasing resources is easily accomplished, the cost of releasing resources in slices can be neglected. In slicing window k, let H_k^{slice} represent the reconfiguration cost, which quantifies the discrepancy in resource reservation decisions between adjacent slicing windows, given by

$$H_k^{\mathrm{slice}} = \frac{1}{\omega_k} \sum_{ap\in\mathcal{AP}^{k-1}\cup\mathcal{AP}^k} \sum_{x\in\mathbf{x}} \left[v_{ap,x}^k - v_{ap,x}^{k-1}\right]^+, \tag{3.10}$$

where function $[\cdot]^+$ represents $\max\{\cdot, 0\}$.

Service Demand Penalty

The designed slices for HD map distribution aim to accommodate as much service demand with satisfied service requirements in the target area. To this end, we introduce a service demand penalty, which indicates the time-slot-averaged remaining service demand not being accommodated through two slices in each slicing window, given by

$$H_k^{\mathrm{pen}} = \frac{1}{\omega_k} \sum_{t=1}^{\omega_k} \left[\lambda_f^{k,t} - \left(\sum_{ap\in\mathcal{BS}} \min\left\{\tilde{\lambda}_{ap,1,f}^{k,t}, \lambda_{ap,f}^{k,t}\right\} + \sum_{ap\in\mathcal{S}_k} \left(\tilde{\lambda}_{ap,0,f}^{k,t} + \tilde{\lambda}_{ap,1,f}^{k,t}\right)\right)\right]^+. \tag{3.11}$$

Specifically, BSs can only serve the service demand within their coverage while satellites can support all service demand in the target area.

Combining all terms, the overall system cost in slicing window k is defined as

$$H_k^{\text{sys}} = \beta_1 H_k^{\text{res}} + \beta_2 H_k^{\text{d}} + \beta_3 H_k^{\text{slice}} + \beta_4 H_k^{\text{pen}}, \tag{3.12}$$

where parameters β_1, β_2, β_3, and β_4 indicate the relative importance of different costs and penalty.

To this end, we formulate the problem of resource slicing in STIVN with two slices to support HD map distribution. Our objective is to minimize the long-term overall system cost, formulated as

$$\mathbf{P0} \quad \min_{\substack{\omega_k, m_{ap,f}^k, \\ v_{ap,x}^k, u_{ap,f}^k}} \quad \mathbb{E}\left[\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} H_k^{\text{sys}}\right] \tag{3.13}$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} m_{ap,f}^k \le d_{ap}^k, \tag{3.13a}$$

$$\sum_{x\in\mathbf{x}} v_{ap,x}^k \le \mathbb{1}_{ap\in\mathcal{AP}^k}, \tag{3.13b}$$

$$\sum_{f\in\mathcal{F}} u_{ap,f}^k \kappa \le \mathbb{1}_{ap\in\mathcal{AP}^k} B_{ap}^{\text{ch}}, \tag{3.13c}$$

$$\omega_k \in \{1, ..., \omega_{\max}\}, \tag{3.13d}$$

$$m_{ap,f}^k \in \{0, 1\}, \tag{3.13e}$$

$$u_{ap,f}^k \in \{0, 1\}, \tag{3.13f}$$

$$v_{ap,x}^k \in [0, 1]. \tag{3.13g}$$

In problem **P0**, constraint (3.13a) constrains that the number of files served in MS should be less than the maximum availability, where the detailed expression will be given in Sec. 3.4. Constraints (3.13b) and (3.13c) indicate that the overall ratio of reserved communication resources and the size of cached files in each AP cannot exceed the limitation if the AP is accessible within the slicing window. Constraint (3.13d) limits the maximum length of each slicing window. Constraints (3.13e)-(3.13f) ensure the binary decisions for multicast and caching, and constraint (3.13g) represents the continuous decisions between 0 and 1 for communication resource reservation.

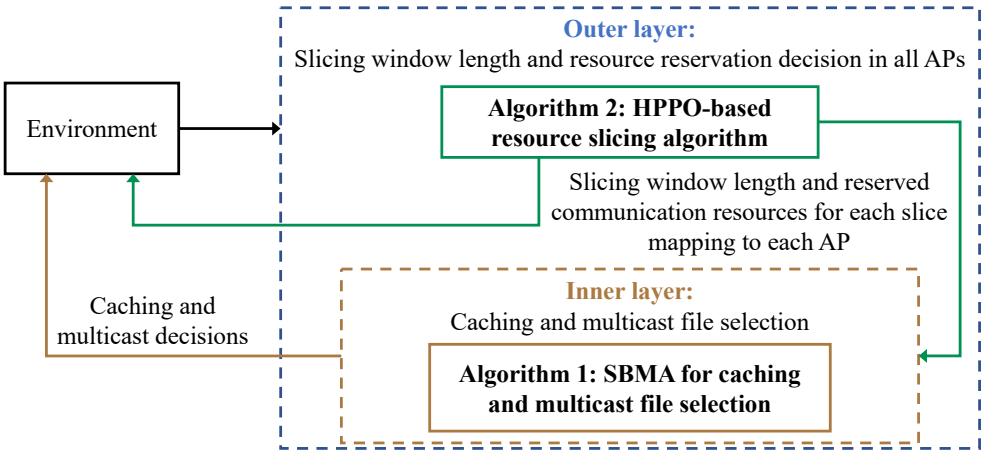


Figure 3.3: Working diagram of the proposed TLRS scheme in STIVN.

3.3.2 Design of Two-Layer Resource Slicing Scheme

The formulated problem **P0** is a stochastic optimization problem due to the spatiotemporal dynamic service demand and time-varying available resources in STIVN. Specifically, the limited information on future service demands, as well as hybrid variable types including both continuous and discrete decisions, make it highly complex and hard to find a globally optimal solution from traditional optimization approaches. One potential solution is to apply reinforcement learning-based methods to solve the problem without requiring knowledge of future information. However, the coupling relation of different decisions and the huge action dimension in adopting a fully data-driven method renders inefficiency and difficult converging. To solve the multi-dimensional resource slicing problem in STIVN, we propose a two-layer resource slicing (TLRS) scheme. Specifically, the problem **P0** is first decoupled into two subproblems: caching and multicast file selection subproblem, and slicing window and communication resource reservation determination subproblem, corresponding to the two layers of the proposed scheme. In the inner layer, a matching-based caching and multicast file selection algorithm is designed to determine the caching and multicast decisions in each slice of APs in each slicing window, given the slicing window length and reserved communication resources. In the outer layer, we leverage an HPPO-based reinforcement algorithm to make resource slicing decisions, including slicing window length and reserved communication resources in each AP. The working diagram of the proposed TLRS scheme is shown in Fig. 3.3.

According to Eq. (3.12), with a given slicing window length ω_k and reserved communication resources $v_{ap,x}^k$ for each slice, the slice reconfiguration cost is determined. In this

case, the long-term impact from slice reconfiguration is stripped away in the caching and multicast file selection subproblem. Therefore, the objective of the first subproblem is to minimize the system cost with maximum accommodated service demand in each slicing window to find the tradeoff between caching resource cost and the delay performance by optimizing the caching and multicast policy, which can be formulated as

$$\mathbf{P1} \quad \min_{m^k_{ap,f}, u^k_{ap,f}} \quad H^{\text{sys}}_k \tag{3.14}$$

$$\text{s.t.} \quad \omega_k = \hat{\omega}_k, \tag{3.14a}$$

$$v^k_{ap,x} = \hat{v}^k_{ap,x}, \forall x \in \mathbf{x}, \forall ap \in \mathcal{AP}^k, \tag{3.14b}$$

$$(3.13a), (3.13c), (3.13e), (3.13f),$$

where $\hat{\omega}_k$ and $\hat{v}^k_{ap,x}$ are the given optimal slicing window length and communication resource reservation decisions in slicing window k, respectively.

Then, for the second subproblem, the slicing window length and communication resource reservation decisions are optimized for each slicing window to minimize the long-term overall system cost, formulated as

$$\mathbf{P2} \min_{\omega_k, v^k_{ap,x}} \mathbb{E}\left[\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} H^{\text{sys}}_k\right] \tag{3.15}$$

$$\text{s.t.} \quad (3.13b), (3.13d), (3.13g).$$

Note that for any given slicing window and communication resource reservation decisions in each slicing window, the optimal caching and multicast decisions can be obtained by solving subproblem **P1**, and the corresponding overall system cost in each slice can be calculated. In the following, we present our designed algorithms for solving subproblems **P1** and **P2** in Sections 3.4 and 3.5, respectively.

3.4 Matching-Based Caching and Multicast File Selection

In this section, we first derive the accommodated service demand in different slices for calculating the system cost. Then, we reformulate the caching and multicast file selection subproblem as a many-to-many two-sided matching problem and develop a swap-based matching algorithm to solve this problem.

3.4.1 Accommodated Service Demand of Slices

For each slice in APs, the accommodated service demand can be determined according to the resource reservation decision. Specifically, if file f is multicast by a satellite ap, considering the large coverage of satellite, the accommodated service demand in time slot (k,t) is all required service demand in the area if the delay can be satisfied, given by

$$\tilde{\lambda}_{ap,0,f}^{k,t} = \begin{cases} m_{ap,f}^{k}\lambda_{f}^{k,t}, & \text{if } D_{ap,0,f}^{k,t} + \frac{T^{\mathrm{M}}}{2} \leq D^{\max} \\ 0, & \text{otherwise.} \end{cases} \tag{3.16}$$

Here, service demand can be fully accommodated only if the overall delay of the last CAV request within the multicast interval should be satisfied for the delay requirement of HD map distribution, where $D_{ap,0,f}^{k,t}$ is determined in Eq. (3.6) according to resource reservation decision $v_{ap,0}^{k}$ and caching decision $u_{ap,f}^{k}$.

For the US in each AP, the accommodated service demand is impacted by both the reserved resources in the slice and the caching decisions of each AP. According to the effective theory, given the reserved resources for US in AP ap, the maximum accommodated intensity of service demand to satisfy the stability of the transmission queue can be defined as

$$\hat{\lambda}_{ap}^{k,t} = \frac{\theta C_{ap,1}^{k,t}}{e^{\theta}-1}. \tag{3.17}$$

Then, considering the caching decisions in each AP, let $\hat{\lambda}_{ap,\mathrm{C}}^{k,t}$ and $\hat{\lambda}_{ap,\mathrm{NC}}^{k,t}$ denote the maximum accommodated service demand intensity for cached and non-cached files, respectively, given by

$$\hat{\lambda}_{ap,\mathrm{C}}^{k,t} = \begin{cases} \hat{\lambda}_{ap}^{k,t}, & \text{if } D_{ap,1}^{\mathrm{Q},k,t} + \mathbb{1}_{ap\in\mathcal{S}} D_{ap}^{\mathrm{P},k,t} \leq D^{\max}, \\ 0, & \text{otherwise,} \end{cases} \tag{3.18}$$

and

$$\hat{\lambda}_{ap,\mathrm{NC}}^{k,t} = \begin{cases} \hat{\lambda}_{ap}^{k,t}, & \text{if } D_{ap,1}^{\mathrm{Q},k,t} + \mathbb{1}_{ap\in\mathcal{S}} D_{ap}^{\mathrm{P},k,t} + D^{\mathrm{BH}} \leq D^{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.19}$$

In the resource reservation stage, we consider that each AP can serve multiple files that are either cached locally at the AP or fetched from the content server. For instance, CAVs within the range of an AP might request different files. If one of these files is cached at the AP, a likely scenario is that the AP will serve more CAVs requesting the cached file with less delay. In this situation, CAVs requesting other files might be supported by other APs. However, it is difficult to ascertain the accommodated service demand for each file at an AP, which is mainly determined in real-time resource scheduling. In this case, for the

resource reservation in a large timescale, we only pursue the maximum service demand that can be accommodated given the reserved resources and caching decisions among all APs. Specifically, the optimal accommodated service demand for each US can be obtained by solving the following linear programming problem to maximize the overall accommodated service demand according to the given resource reservation and caching decisions in each time slot for US from all APs, given by

$$\mathbf{P3} \quad \max_{\tilde{\lambda}^{k,t}_{ap,1,f}} \sum_{ap\in\mathcal{AP}^k} \sum_{f\in\mathcal{F}} \tilde{\lambda}^{k,t}_{ap,1,f} \tag{3.20}$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} \tilde{\lambda}^{k,t}_{ap,1,f} \leq a^{k,t}_{ap} \hat{\lambda}^{k,t}_{ap}, \tag{3.20a}$$

$$\sum_{ap\in\mathcal{AP}^k} (\tilde{\lambda}^{k,t}_{ap,0,f} + \tilde{\lambda}^{k,t}_{ap,1,f}) \leq \lambda^{k,t}_{f}, \tag{3.20b}$$

$$\tilde{\lambda}^{k,t}_{ap,1,f} \leq u^{k}_{ap,f} \hat{\lambda}^{k,t}_{ap,\mathrm{C}} + (1 - u^{k}_{ap,f}) \hat{\lambda}^{k,t}_{ap,\mathrm{NC}}. \tag{3.20c}$$

Here, constraint (3.20a) limits the available accommodated service demand in each AP; (3.20b) constrains the service demand request of each file; and (3.20c) regulates the maximum accommodated service demand in each AP for the file according to the delay performances given by Eqs. (3.18) and (3.19). Since **P3** is a linear programming program, it can be directly solved by some convex optimization solvers, such as CVX [96], to achieve the accommodated service demand from the US in all APs under the given resource reservation and caching decisions.

3.4.2 Many-to-Many Matching Game Formulation

In order to solve the subproblem **P1**, we aim to find the optimal solution of multicast and caching files $m^{k}_{ap,f}$ and $u^{k}_{ap,f}$ in each AP. Considering that each BS has sufficient storage capacity to store all HD map files related to the corresponding subarea, we mainly investigate the caching and multicast file selection from LEO satellites for both MS and US. Specifically, with the given reserved communication resources, we aim to find the optimal caching and multicast decisions for solving **P1**. The considered decisions within the slicing window include whether to multicast an HD map file and whether to cache the file in each satellite.

Considering the combination of two binary decisions, we transform problem **P1** as a many-to-many matching problem to find the matching between caching and multicast files in different satellites. Specifically, we define (ap, m) to indicate the service mode m

of satellite ap within the slicing window, where $m = 1$ is multicast with caching, $m = 2$ indicates the multicast without caching, $m = 3$ represents unicast with caching, and $m = 4$ represents unicast without caching. For simplicity, we omit the notation k in this section, as the decisions are made for each slicing window with a known slicing window length. Let $\Phi = \{(ap, m) \mid \forall ap \in \mathcal{S}^k, m = 1, 2, 3, 4\}$ denote the set of satellite service modes within the slicing window. In this case, $\mathcal{F}$ and Φ represent the HD map file set and the satellite service mode set as two disjoint sets of selfish and rational players, aiming to minimize the overall system cost within the slicing window. Specifically, if file f and (ap, m) are matched with each other, they form a matching pair $(f, (ap, m))$, representing that file f is served by satellite ap with service mode m. Each player tends to seek the most preferred partners for matching. The matching can be formally presented as follows.

Definition 1. Given two disjoint sets, $\mathcal{F}$ for HD map files and Φ for the satellite service modes, a many-to-many matching Ψ is a mapping from the set $\mathcal{F} \cup \Phi$ into the set of all subsets of $\mathcal{F} \cup \Phi$, such that for every $f \in \mathcal{F}$ and $(ap, m) \in \Phi$:

1). $\Psi((ap, m)) \subseteq \mathcal{F}$ and $\Psi(f) \subseteq \Phi$;

2). $|\Psi((ap, 1))| \leq d_{ap,1}$ and $|\Psi((ap, 2))| \leq d_{ap,2}$;

3). $|\Psi((ap, 1))| + |\Psi((ap, 3))| \leq \lfloor B_{ap}^{\mathrm{ch}}/\kappa \rfloor$;

4). $|\Psi(f)| = |\mathcal{S}^k|$;

5). $(ap, m) \in \Psi(f)$ if and only if $f \in \Psi((ap, m))$;

6). $\Psi((ap, m)) \cap \Psi((ap, m')) = \emptyset, m \neq m'$.

Condition 1) states that each service mode is matched with a subset of HD map files, and each file is matched with a subset of satellite service modes. In condition 2), to satisfy the delay requirement, each satellite can only support limited files in MS. Let d_{ap}, $d_{ap,1}$ and $d_{ap,2}$ denote the maximum multicast file number, the number of available cached files, and available non-cached files for MS in satellite ap, respectively, expressed as

$$d_{ap,1} = d_{ap} = \begin{cases} \min_t \left\{ \left\lfloor C_{ap,0}^{k,t} T^{\mathrm{M}} \right\rfloor \;\middle|\; a_{ap}^{k,t} = 1 \right\}, & \text{if } \exists t, D_{ap,0}^{\mathrm{Q},k,t} + T^{\mathrm{M}} + D_{ap}^{\mathrm{P},k,t} \leq D^{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.21}$$

$$d_{ap,2} = \begin{cases} d_{ap} - |\Psi((ap, 1))|, & \text{if } \exists t, D_{ap,0}^{\mathrm{Q},k,t} + T^{\mathrm{M}} + D_{ap}^{\mathrm{P},k,t} + D^{\mathrm{BH}} \leq D^{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.22}$$

The maximum multicast file number is determined by considering the presence of cache scenarios given multicast interval T^{M}, which has a loose queuing and transmission delay

requirement compared with non-cache scenarios to support more files to be multicast. In this case, the number of available cached files $d_{ap,1}$ in MS is equal to d_{ap}, given in Eq. (3.21). On the other hand, the number of available non-cached files is obtained by subtracting the number of selected files in mode 1 from the number of total available files, given by Eq. (3.22) As the satellite moves over time slots, the number of available files varies over time. In this scenario, we select the minimum number of available files within the slicing window for MS in each satellite. Note that given different slicing windows, d_{ap}^k in constraint (3.13a) is the same as d_{ap} in this section, indicating the available files served by MS within slicing window k. For condition 3), since the MS and US share the same storage in each satellite, the cached files cannot exceed the maximum available caching resources in each satellite. Condition 4) states that the size of $\Psi(f)$ should equal the satellite number $|\mathcal{S}^k|$ in the slicing window, indicating that each satellite needs to select one service mode to each file. Note that the selection of the service mode does not directly correspond to the accommodated service demand, which should be determined according to Sec. 3.4.1, i.e., even if file f is selected to be served with mode m in satellite ap, the accommodated service demand can be 0. Condition 5) states that if satellite service mode (ap, m) is matched to file f, file f is also matched to satellite service mode (ap, m), and vice versa. Condition 6) states that a file can only choose one service mode in each satellite.

With the matching pair $(f, (ap, m))$ in the many-to-many matching Ψ, we can obtain the caching and multicast file decisions for each satellite ap respectively. Specifically, a matching pair $(f, (ap, 1))$ can refer to the decisions $m_{ap,f}^k = 1$ and $u_{ap,f}^k = 1$; matching pair $(f, (ap, 2))$ can refer to the decisions $m_{ap,f}^k = 1$ and $u_{ap,f}^k = 0$; matching pair $(f, (ap, 3))$ can refer to the decisions $m_{ap,f}^k = 0$ and $u_{ap,f}^k = 1$; and matching pair $(f, (ap, 4))$ can refer to the decisions $m_{ap,f}^k = 0$ and $u_{ap,f}^k = 0$. Through the caching and multicast file decisions from matching policy Ψ, we can obtain the utility $H(\Psi)$, which is the value of H_k^{sys} for the slicing window k with the solution from **P3**.

3.4.3 Swapping-Based Matching Algorithm for Caching and Multicast File Selection

Generally, the outcome of a matching game depends on the dynamic interaction among players. The formulated matching game is a many-to-many matching game with externalities [97], which indicates that the adjustment of a matching pair will affect the preference of other players, also known as the peer effects. Due to externalities, each player may keep changing the preference order in response to the adjustment of other matching pairs. Therefore, the static preference list is no longer appropriate in this matching game.

Algorithm 1 SBMA for caching and multicast file selection

1: **Initialization Phase**
2: Let $\Psi = \emptyset$;
3: **for** $f \in \mathcal{F}$ **do**
4: Match f with the service mode $(ap, 4)$ of each satellite, $\Psi = \Psi \cup (f, (ap, 4))$;
5: **end for**
6: **Swap Matching Phase**
7: **for** $ap \in \mathcal{S}^k, m \in \{1, 2, 3, 4\}$ **do**
8: Calculate $H(\Psi)$, set $\Psi_{\text{temp}} = \Psi$;
9: For a file $f \in \Psi((ap, m))$, the controller searches the satellite service mode $(ap', m') \in \Phi \setminus \{(ap, m)\}$ with $f' \in \Psi((ap', m'))$ for a swap operation ;
10: **if** $H(\Psi^{ap,m,f}_{ap',m',f'}) > H(\Psi_{\text{temp}})$ **then**
11: Swap the pair according to **Definition 2** and obtain Ψ' as new matching policy;
12: $\Psi_{\text{temp}} = \Psi'$, and go back to Line 9;
13: **end if**
14: Update the matching with $\Psi = \Psi_{\text{temp}}$;
15: **end for**
16: Go back to Line 7 and repeat until no swapping pairs can be found;
17: **Output:** Caching indicator $u^k_{ap,f}$ and multicast decision $m^k_{ap,f}$ in the slicing window according to matching policy Ψ;

To solve this matching problem, we propose a swap-based matching algorithm (SBMA) for the caching and multicast file selection. Specifically, the swap operation in a matching allows two players in the same set to exchange their matches while keeping other players' matching [98], which can be presented as

Definition 2 (Swap operation)**.** Given a matching Ψ and two different pairs $(f, (ap, m)), (f', (ap', m')) \in \Psi$, a swap matching is

$$\Psi^{ap,m,f}_{ap',m',f'} = \Psi \setminus \{(f, (ap, m)), (f', (ap', m'))\} \cup \{(f', (ap, m)), (f, (ap', m'))\}. \tag{3.23}$$

Based on the swap operation, the procedure of our proposed SBMA for caching and multicast file selection is shown in Algorithm 1. In the initialization phase, each file is matched with the unicast and non-caching mode of each satellite. Then, in the swap matching phase, each satellite service mode checks whether it can switch its currently selected file with other files selected by other service modes to increase the utility. The iterations stop when no swapping can be found to further increase the utility value. In this

case, the final matching Ψ can be found with the given slicing window length and reserved communication resources for caching and multicast decisions.

3.5 HPPO-Based Slicing Window Length and Resource Reservation

In this section, we investigate the optimal slicing window length and communication resource reservation decisions to solve subproblem **P2** in the outer layer. Considering the long-term impacts on sequential decision-making, we reformulate the subproblem by leveraging the Markov decision process (MDP) and propose a reinforcement learning (RL)-based algorithm to solve the problem.

3.5.1 Markov Decision Process Formulation

Due to the dynamic nature of service demand and the impact of slicing reconfiguration cost from the previous decisions for long-term optimization, finding the optimal solution for the slicing window length and communication resource reservation problem requires the consideration of sequential decisions. Deterministic optimization algorithms are inadequate for solving such problems. Thus, we reformulate subproblem **P2** by leveraging MDP to model the network dynamics. Let $\zeta^k = [\Lambda^k, P^k, \mathbf{v}^{k-1}] \in \mathfrak{S}$ denote the state of the STIVN in slicing window k with the dimension $1 \times |\zeta|$. Specifically, Λ^k is the set consisting of the predicted service demand in each time slot during the future $\omega_{\max}$ time slots of all subareas; P^k is the set of all positions of APs $\mathcal{AP}^k_{\max} = \mathcal{BS} \cup \mathcal{S}^k_{\max}$ in the next $\omega_{\max}$ time slots, where $\mathcal{S}^k_{\max} = \{ap \mid \sum_{t=1}^{\omega_{\max}} a^{k,t}_{ap} > 0, \forall ap \in \mathcal{S}\}$; and $\mathbf{v}^{k-1}$ is the set of resource reservation decision in the previous slicing window, where $\mathbf{v}^k = \{v^k_{ap,x} \mid \forall ap \in \mathcal{AP}^k_{\max}, \forall x \in \mathbf{x}\}$.

Let $\mathfrak{A} = \mathfrak{A}_1 \otimes \mathfrak{A}_2$ denote the action space, which is the combination of the discrete action space of the slicing window length and continuous action space of communication resource reservation, where $\otimes$ is the Cartesian product. Thus, the action in slicing window k is denoted by $\mathcal{A}^k = (\mathcal{A}^k_1, \mathcal{A}^k_2) = (\omega^k, \mathbf{v}^k) \in \mathfrak{A}$ for the slicing window length and reserved communication resources in each slice. Note that the action space contains both discrete and continuous actions with a dimension of $\mathcal{A}^k \in \{1, ..., \omega_{\max}\} \times [0, 1]^{|\mathcal{AP}^k_{\max}|}$.

To evaluate an action in a state, the set of rewards is denoted by $\mathfrak{R} := \mathfrak{S} \times \mathfrak{A} \to \mathbb{R}$. We define the obtained reward function of taking action $\mathcal{A}^k$ in state ζ^k as

$$\mathcal{R}(\zeta^k, \mathcal{A}^k) = -H^{\text{sys}}_k. \tag{3.24}$$

Let $\mathbf{\Pi}$ denote the set of all policies, which is the function mapping from state space $\mathfrak{S}$ to the action space $\mathfrak{A}$. Specifically, an accumulated discounted reward function is considered to estimate the expected return of the current decision-making. Therefore, the problem of minimizing the long-term overall system cost in subproblem **P2** can be approximated by minimizing the long-term accumulated discounted reward, given by

$$\mathbf{P2'} \max_{\pi \in \mathbf{\Pi}} \ \mathbb{E}\left[\lim_{K\to\infty} \sum_{k=1}^{K} \gamma^k \mathcal{R}(\zeta^k, \mathcal{A}^k) \middle| \pi\right] \tag{3.25}$$
$$\text{s.t.} \quad (3.13b), (3.13d), (3.13g).$$

3.5.2 HPPO-Based Resource Slicing Algorithm

Considering the lack of information on state transition probabilities for the MDP-based problem, the RL-based algorithm has the potential to solve the problem without requiring state transition probabilities. However, determining the non-homogeneous actions in subproblem **P2′**, i.e., discrete action for slicing window length and continuous action for communication resource reservation, is challenging, where the discrete action may also affect the policy for determining the continuous action. Some related studies investigate methods to output two kinds of actions from one neural network including the discretization of the continuous action or continualization of the discrete action [99], which however cause scalability issues or difficulties in approximation and generalization of the continuous action in piecewise-function action subspaces.

Inspired by the studies about parameterized action space in RL [100] with both discrete action and continuous action, the design for the joint action where the discrete action can be obtained before generating the continuous parameters [101], and the hybrid actor-critic RL in parameterized action space [102], we hereby developed an HPPO-based algorithm to solve this problem, as shown in Fig. 3.4. Specifically, the HPPO-based algorithm is designed based on the proximal policy optimization (PPO)-based algorithm [103], which can reuse the historical samples and handle a large-scale problem, to find the optimal policy, with two actor networks to tackle the hybrid action issue. The key idea is to jointly train two separate actor networks as two policies for the slicing window length decision and resource reservation, respectively. For determining the slicing window length, the output is generated from the discrete actor network according to the input state. Then, for the resource reservation policy, the decision to select continuous actions for each slice is obtained with the given slicing window length decision to find the optimal long-term resource reservation decisions for this slicing window. In this case, the overall policy consisting of two actor networks is

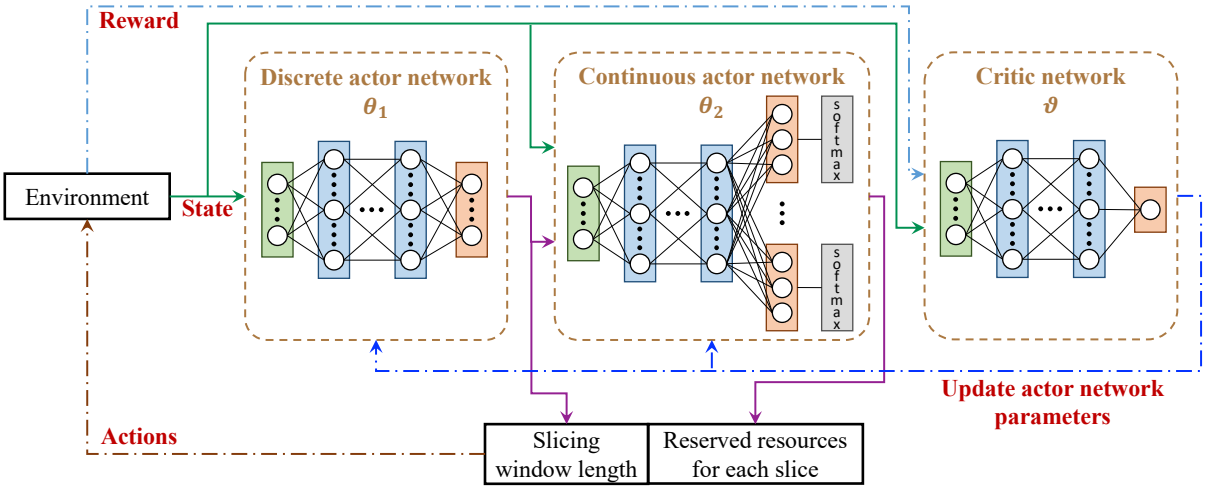


Figure 3.4: Illustration of proposed HPPO-based algorithm.

the multiplication of two policies with conditional probability. Similar to the traditional PPO algorithm, the critic network as a state-value function is leveraged to generate the value of the given state for updating the actor networks. This kind of design can effectively extract the relationship between slicing window length and resource reservation decisions to reach a better performance.

Considering that the AP number serving the target area in each slicing window may vary over time due to satellite mobility, the variability in state and action space poses a challenge in solving the problem using an HPPO-based algorithm. Therefore, we use the maximum number of M APs to regulate the overall state and action spaces. As shown in Fig. 3.4, the output of the discrete actor network, parameterized by θ_1, is the slicing window length decision according to the input state. In addition, to obtain the optimal resource reservation decision of each AP, the input of the continuous actor network includes the state and the slicing window length decision from the discrete actor network. The output of the continuous actor network, parameterized by θ_2, is the resource reservation decisions for each slice of each AP. Since the direct output of resource reservation decisions for each slice cannot guarantee constraints (3.13b) and (3.13g) with an accessible resource reservation ratio for each AP. Therefore, we design a softmax-based actor network to regulate the output decision of each slice in APs. The basic idea is that the output layer adopts a softmax activation function to activate each 3-dimension vector indicating the ratio of reserved resources in MS and US, and the ratio of unreserved resources, where the overall output vector from the continuous actor network has the dimension of $3M$. In this case,

a normalized output vector for each AP can be obtained, given by $\sum_{x=0}^{2} v_{ap,x}^{k} = \mathbb{1}_{ap \in \mathcal{AP}_k}$, where $\{v_{ap,0}^{k}, v_{ap,1}^{k}, v_{ap,2}^{k}\}$ is the output vector of each AP. Here, $v_{ap,2}^{k}$ is the output indicating the ratio of unreserved resources in AP ap. Therefore, the sum ratio of all reserved resource $\sum_{x=0}^{1} v_{ap,k}^{x} \leq \mathbb{1}_{ap \in \mathcal{AP}_k}$ satisfying the constraints.

To train the HPPO-based algorithm, the goal is to find the optimal parameter θ_1 and θ_2 for two actor networks, where the overall policy, parameterized by $\Theta = [\theta_1, \theta_2]$, is the combination of two policies, given by

$$\pi_{\Theta}(\mathcal{A}|\zeta) = \pi_{\theta_2}(\mathcal{A}_2|\mathcal{A}_1, \zeta)\pi_{\theta_1}(\mathcal{A}_1|\zeta), \tag{3.26}$$

where the policy of θ_2 is jointly related to the state ζ and action $\mathcal{A}_1$ from the discrete actor. The goal is to maximize the clipped surrogate objective

$$L^{\text{clip}}(\Theta) = \mathbb{E}\Big\{ \min(\xi_{\Theta} A^{\pi_{\Theta_{\text{old}}}}, \text{clip}(\xi_{\Theta}, 1-\delta, 1+\delta) A^{\pi_{\Theta_{\text{old}}}}) \Big\}. \tag{3.27}$$

The policy updating ratio ξ_{Θ} based on current policy for Θ is defined as

$$\xi_{\Theta} = \frac{\pi_{\Theta}(\mathcal{A}|\zeta)}{\pi_{\Theta_{\text{old}}}(\mathcal{A}|\zeta)} = \frac{\pi_{\theta_2}(\mathcal{A}_2|\mathcal{A}_1, \zeta)\pi_{\theta_1}(\mathcal{A}_1|\zeta)}{\pi_{\theta_{\text{old},2}}(\mathcal{A}_2|\mathcal{A}_1, \zeta)\pi_{\theta_{\text{old},1}}(\mathcal{A}_1|\zeta)}. \tag{3.28}$$

Here, $\pi_{\theta_{\text{old,i}}}$ is the old policy of actor network i. Moreover, $A^{\pi_{\Theta_{\text{old}}}}$ is the estimated advantage from policy $\pi_{\Theta_{\text{old}}}$ as the combination of two policies, which can be expressed as

$$A^{\pi_{\Theta_{\text{old}}}}(\zeta^k, \mathcal{A}^k) = \mathcal{R}(\zeta^k, \mathcal{A}^k) + \gamma V_{\vartheta}(\zeta^{k+1}) - V_{\vartheta}(\zeta^k), \tag{3.29}$$

where $V_{\vartheta}(\zeta^k)$ is a state-value function of the critic network parameterized by ϑ, given by

$$V_{\vartheta}(\zeta^k) = \mathbb{E}\left\{ \sum_{l=0}^{\infty} \gamma^l \mathcal{R}(\zeta^{k+l+1}, \mathcal{A}^{k+l+1}) \,|\, \zeta^k \right\}. \tag{3.30}$$

The clipping function clips the ratio ξ_{θ_i} between $1-\delta$ and $1+\delta$ with a clipping hyperparameter δ, which removes incentives for the new policy to get far from the old policy to improve the learning stability. The measure of the accuracy of the critic network for the state-value function is based on the mean square error loss function, given by

$$L^{\text{critic}}(\vartheta) = \mathbb{E}\left\{ \left(\mathcal{R}^k + V_{\vartheta}(\zeta^{k+1}) - V_{\vartheta}(\zeta^k) \right)^2 \right\}. \tag{3.31}$$

The overall HPPO-based resource reservation algorithm is given in Algorithm 2. Considering the slicing window length of each slicing window varies in each training interaction,

Algorithm 2 HPPO-based resource slicing algorithm

1: **Initialization:** Parameters $\Theta = [\theta_1, \theta_2]$ for policy, ϑ for state-value function, and buffer memory $\mathcal{B}$;
2: **for** $j \in \{1, 2, ...\}$ **do**
3: Obtain initial state ζ^1;
4: $k = 1$;
5: **while** $\sum_{l=1}^{k} \omega_l < T_{\text{all}}$ **do**
6: Execute action $\mathcal{A}^K$ based on $\pi_\theta(\zeta^k|\mathcal{A}^k)$;
7: Compute reward $\mathcal{R}(\zeta^k, \mathcal{A}^k)$;
8: Obtain new state ζ^{k+1};
9: Record transition $(\zeta^k, \mathcal{A}^k, \mathcal{R}^k, \zeta^{k+1})$ into $\mathcal{B}$;
10: $k = k + 1$;
11: **end while**
12: **for** $e \in \{1, 2, ..., E\}$ **do**
13: Sample a minibatch of $\mathcal{B}_N$ transitions from buffer $\mathcal{B}$;
14: Update parameters $\Theta \leftarrow \arg\max_\Theta L^{\text{clip}}(\Theta)$ and $\vartheta \leftarrow \arg\min_\vartheta L^{\text{critic}}(\vartheta)$;
15: **end for**
16: Empty buffer $\mathcal{B}$ for a new training episode;
17: **end for**
18: **Output:** Parameters θ_1, θ_2;

to make the training process fair, we set that the accumulated time slots until the end of the last slicing window in each episode should be just larger than T_{all}. For the model training, first, the policy and the value functions are initialized with parameters θ_1, θ_2, and ϑ, respectively. The buffer $\mathcal{B}$ is used to store the transitions of the interaction with the environment. Then, at each slicing window, the controller collects the state from the environment, takes actions based on the current policy, computes the reward, observes the new state, and records transitions into buffer based on Line 6 to 9. At the end of each episode, the policy and value function will be updated based on the clipped surrogate objective and the mean-square error based on Line 13 to 14. Specifically, to ensure the stability of the strategy updating, the policy and value functions are updated E times in each episode by taking full use of the samples generated by the current strategy.

Table 3.2: Simulation Parameters

Velocity of satellites	7.56 km/s
LEO satellite transmission power	40 dBm
LEO satellite carrier frequency	40 GHz
LEO satellite antenna gains	32 dBi
LEO satellite bandwidth	250 MHz
LEO satellite path-loss exponent	2.5
Noise power	-174 dBm/Hz
BS transmission power	32 dBm
BS carrier frequency	28 GHz
BS antenna gains	10 dBi
BS bandwidth	100 MHz
BS path-loss exponent	3.5
Parameters β_1, β_2, β_3, and β_4	0.01, 1, 0.5, 1.5

3.6 Performance Evaluation

In this subsection, extensive simulations are carried out to evaluate the proposed TLRS scheme. Specifically, we first elaborate on the simulation settings. Afterward, the simulation results are presented to demonstrate the effectiveness of our proposed scheme.

3.6.1 Simulation Setting

In our simulation scenario, we consider the target area with a radius of 15 km, where the central point is at the latitude of 30°N and the longitude of 82.5°W. The area is divided into 4 subareas and each subarea consists of 2 kinds of HD map demand requested by CAVs. The size of each HD map is 10 Mb. We adopt the Starlink Phase 1 for the satellite constellation parameters, including 72 orbits with 22 satellites in each orbit. The inclination angle is 53 degrees and the altitude of the satellite is 550 km. By default, we set the minimal access elevation angle to be 35 degrees. For the terrestrial networks, BSs are distributed in the area with a density of 0.1 /km^2, each with a coverage radius of 1 km. The unit cost of the communication resources and caching resources $\alpha_{ap}^{\mathrm{cm}}$ and $\alpha_{ap}^{\mathrm{ch}}$ of satellites are given by 0.1 and 1 respectively, while the unit costs of BSs are 0.02 and 0.2. The total duration of

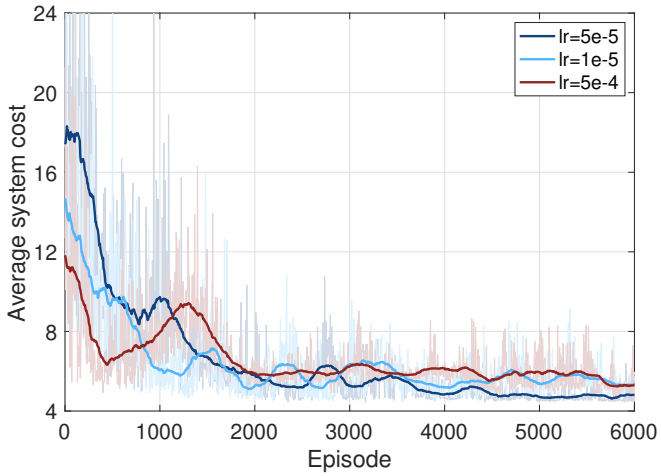


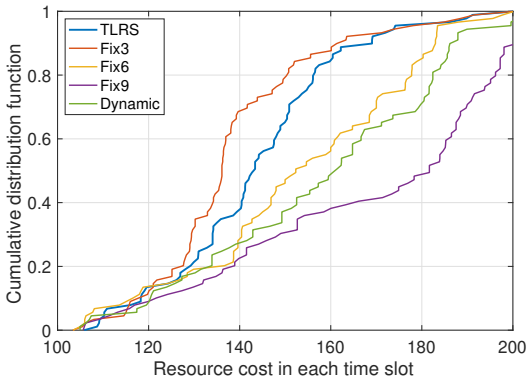
Figure 3.5: Convergence performance of the proposed TLRS scheme.

the simulation comprises 30 min and the length of each time slot is set to be 20 s. The delay requirement of the HD map distribution is 2.5 s with a reliability of 0.99. The default multicast interval is set as 1.5 s. Some main simulation parameters are listed in Table 3.2.

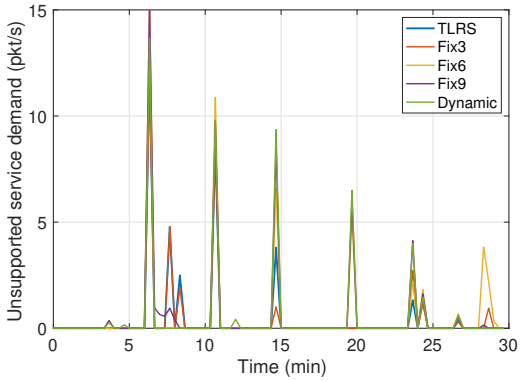
For both actor and critic networks in the HPPO-based reinforcement learning model, we employ three fully connected hidden layers with [256, 256, 128] neurons. The Tanh activation function is applied after each hidden layer to realize a nonlinear approximation. Furthermore, the Adam optimizer is adopted to train the parameters of networks.

3.6.2 Simulation Results

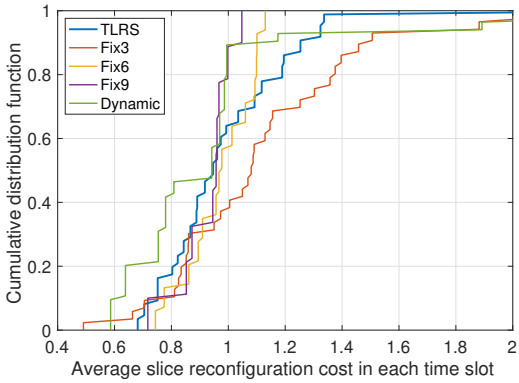
The convergence performance of the proposed TLRS scheme is shown in Fig. 3.5 to illustrate the average cost of each episode. To highlight the convergence trend, the dark line is the 150-point moving average of the average system cost and the shadowed lines are the performance of each episode. We compare the convergence performance with different learning rates, where an identical learning rate is set for the discrete actor network, continuous actor network, and critic network. We can observe that all curves decrease with the increase of training episodes and tend to converge with a large episode number, which verifies the convergence property of the proposed HPPO-based algorithm. In addition, a large learning rate leads to a fast convergence speed, which however results in an unstable performance. Therefore, in the remaining part of the section, the learning rate for the HPPO-based algorithm is set to be 5×10^{-5}.



(a) CDF of resource cost.



(b) Unsupported service demand.



(c) CDF of average slice reconfiguration cost.

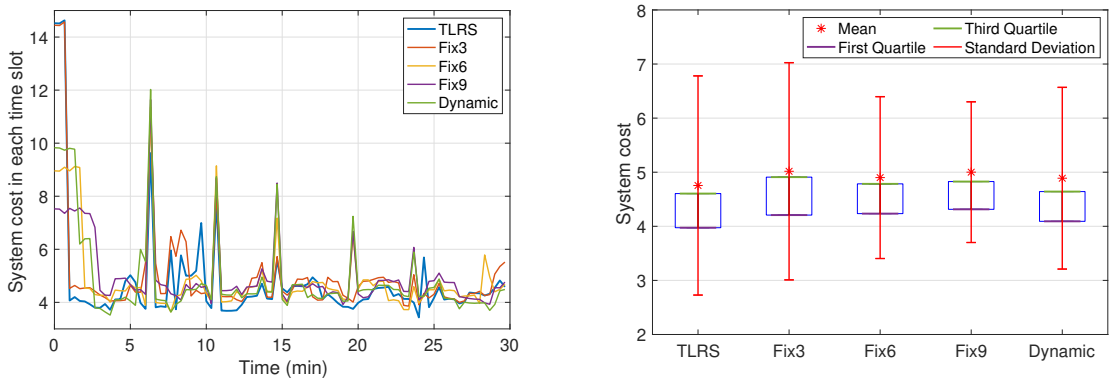
Figure 3.6: Performance under different policies for slicing window length.

Next, we perform several experiments to validate the efficacy of different components in the TLRS scheme. First, we evaluate the effectiveness of the slicing window length decision from the HPPO-based resource reservation algorithm. Specifically, we adopt two benchmark algorithms: 1) *Fix*n: the length of each slicing window length is fixed with n time slots, and 2) *Dynamic*: the slicing window length is determined by the fluctuation of service demand, ensuring that the variance of service demand within each slicing window remains below a threshold. For all benchmark algorithms, the resource reservation decisions are determined with a conventional PPO algorithm, and caching and multicast decisions are made via SBMA.

The performances of different metrics under different slicing window length determination

policies are shown in Fig. 3.6, including the resource cost, unsupported service demand, and slice reconfiguration cost in each time slot within the simulation duration. The cumulative distribution function (CDF) of the resource cost in each time slot is depicted in Fig. 3.6(a). The resource cost of *Fix*3 is the lowest since an efficient resource reservation can adapt to the dynamic service demand in a short time duration with less waste of resources. The proposed TLRS scheme has the comparative resource cost performance with *Fix*3 due to the adaptive slicing window length, which can effectively fit dynamic service demand. On the other hand, the resource cost of *Fix*6 and *Fix*9, due to their long slicing window length, the resource cost is much higher to compromise the highly burst service demand within the slicing window. The *Dynamic* policy mainly focuses on the variation of the service demand, which however neglects the time-varying available resources caused by satellite mobility. Therefore, in some cases, a long slicing window length may result in a high resource cost by reserving resources in some satellites with limited communication capability, e.g. satellites in a low elevation angle. Figure 3.6(b) shows the remaining unsupported service demand in the target area, which indicates different slicing window length policies have a limited impact on the accommodation of service demand. Specifically, the proposed TLRS scheme and *Fix*3 show superior performance in accommodating service demand, which consistently maintains lower levels of unsupported service demand throughout the duration. The performance of slice reconfiguration cost is shown in Fig. 3.6(c). Note that the average slice reconfiguration cost in each time slot is the time-averaged slice reconfiguration cost of the slicing window, which means the average slice reconfiguration cost in each time slot within a slicing window is the same. From the figure, we can observe that the *Fix*3 has the largest slice reconfiguration cost since it adjusts the slice frequently resulting in a high cost, while *Fix*9 and *Dynamic* policies maintain a low cost due to the large duration of the slice. The proposed TLRS scheme has a medium performance since the slices are adjusted according to both dynamic service demand and available resources.

By combining different impacts, the performance of the system cost is illustrated in Fig. 3.7. The time-slotted system cost is shown in Fig. 3.7(a). The proposed TLRS scheme has almost the lowest system cost compared with other schemes, which shows the effectiveness of the adaptive slicing window length for resource reservation in STIVN. In addition, some statistical results are provided in Fig. 3.7(b), including the mean value, standard deviation, and the first and third quartile of the system cost for each policy. From the figure, we can see that the proposed TLRS has the lowest mean value, which indicates the best performance among all slicing window policies. Furthermore, the TLRS has a better performance in both the first and third quartile of overall cost, which represents that TLRS can achieve better performance at most time slots. The *Dynamic* policy has a similar third quartile to the TLRS, which however has a larger mean value since the



(a) System cost.

(b) Statistical results.

Figure 3.7: System cost under different policies for slicing window length.

slicing window length is only determined by the variation of service demand and does not consider the available resources over time. In this case, some large cost value occurs for some unnecessary resources and unsupported service demands. For the fixed slicing window policies, the standard deviation is smaller for the longer slicing window, due to closer resource consumption within each slicing window.

Then, we show the average system cost performance of our proposed scheme under different minimum satellite elevation angles in Fig. 3.8 From the figure, we can see that when the minimum elevation angle is small, indicating that more satellites are able to serve CAVs in the target area, the system cost becomes small due to efficient resource reservation decisions under different slicing window lengths. Specifically, with more satellites being able to serve the target area, the time-varying resources from the mobility of satellites can be effectively leveraged for a longer slicing window to satisfy the dynamic service demand. In this case, the *Fix*9 has a relatively small cost compared with other fixed slicing window length policies, while *Fix*3 achieves the largest cost value considering the cost from frequent slice reconfiguration. With the increment of the elevation angle, the overall cost increases correspondingly. Specifically, with a large elevation angle, the resources from satellite networks may not accommodate all service demands from the areas not covered by terrestrial areas, resulting in a high cost. Our proposed TLRS scheme can always achieve the lowest cost compared with other slicing window length policies.

Next, we investigate the effectiveness of the HPPO-based resource reservation strategy in our proposed TLRS scheme. For comparison, we introduce the following benchmark algorithms: 1) *SRS*: Separate resource slicing decisions for satellite and terrestrial networks,

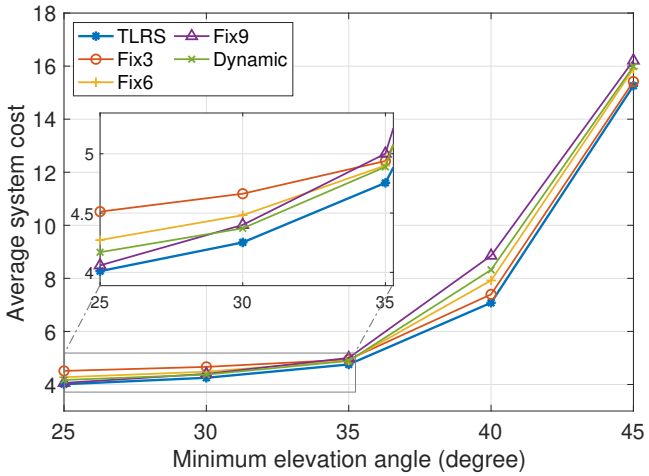
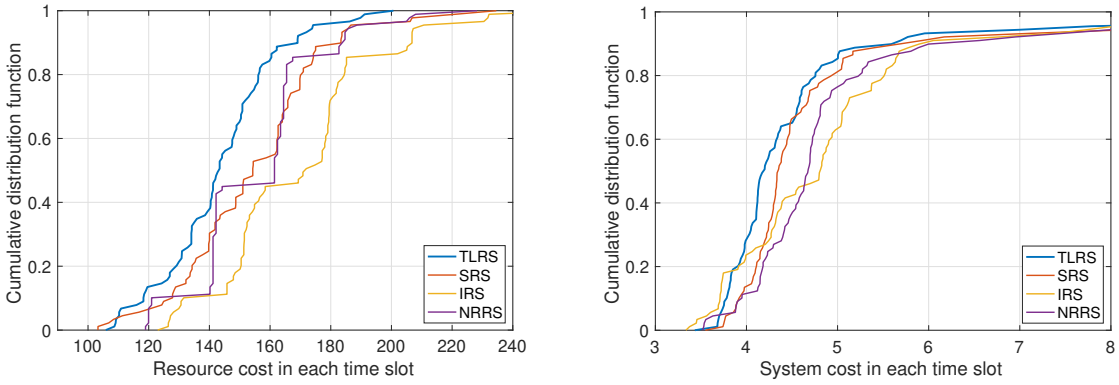


Figure 3.8: Average system cost under different minimum elevation angles of LEO satellites.

i.e., all involved satellites and all BSs have consistent resource reservation decisions within the slicing window, respectively, although the decisions for satellites and BSs may differ from each other; 2) *IRS*: All involved APs within the slicing window have the identical resource reservation decisions; and 3) *NRRS*: Resource slicing decisions for APs are non-reconfigurable and maintain a fixed value in all slicing windows, where the decisions are made according to the average service demand from CAVs. Both *SRS* and *IRS* policies adopt an HPPO-based algorithm with two and one action for resource reservation, respectively.

In Fig. 3.9, we show the CDF of resource cost and system cost performances of different resource reservation strategies. Specifically, we can find that the proposed TLRS scheme reserves the minimum resources to achieve the minimum system cost since the resource reservation decisions are made for each AP in each slicing window, which can result in efficient resource utilization for minimizing the overall resources. However, *SRS* and *IRS* cannot provide efficient resource reservation decisions due to their limited action dimensions, which cannot effectively take into account the spatial dynamic of service demand and the impact of satellite mobility. In addition, since *NRRS* strategy does not adapt to the changes in satellite resources, it only considers the average service demand for all time slots, resulting in high system costs.

Then, we compare the average system cost of different communication resource reservation decisions with different minimum elevation angles of satellite networks in Fig. 3.10. Specifically, the proposed TLRS scheme has the lowest system cost compared with other policies. The advantage stems from the fine-grained resource reservation decisions from the



(a) CDF of resource cost.

(b) CDF of system cost.

Figure 3.9: Performance under different resource reservation strategies.

HPPO-based algorithm in the TLRS scheme by considering the decision for each AP in each slicing window, reducing the system cost. The benchmark algorithms can only make resource reservation decisions by taking into account the average conditions, which results in a higher cost with either more unsupported service demand or more resource cost. With the increment of minimum elevation angle, the overall cost increases accordingly, and the TLRS scheme consistently maintains a low level of cost. Specifically, with a large elevation angle, due to the insufficient available resources, a high cost will occur for all strategies, while the proposed TLRS scheme can always achieve the lowest cost.

Last, we evaluate the effectiveness of the SBMA for caching and multicast decisions. Specifically, we adopt different benchmark algorithms, including the strategies with and without multicast, as well as different caching policies, including matching-based caching (MBC) and popularity-aware caching (PAC) strategies. Specifically, for the non-multicast policy, only unicast slice is considered for each AP.

In Fig. 3.11, we compare performances of different caching and multicast policies with different multicast interval times. From Fig. 3.11(a), we can find that the proposed TLRS scheme can accommodate the most service demand compared with other policies due to the effective caching and multicast strategies in each satellite. Specifically, when the multicast interval time is small, less resources are required for multicasting one file with delay satisfaction. In this case, more service demand of popular files can be benefited through multicast. With the increment of the multicast interval, the required resources increase accordingly due to limited transmission time for each multicast to satisfy the service requirement, the service demand cannot be fully accommodated. The policy with

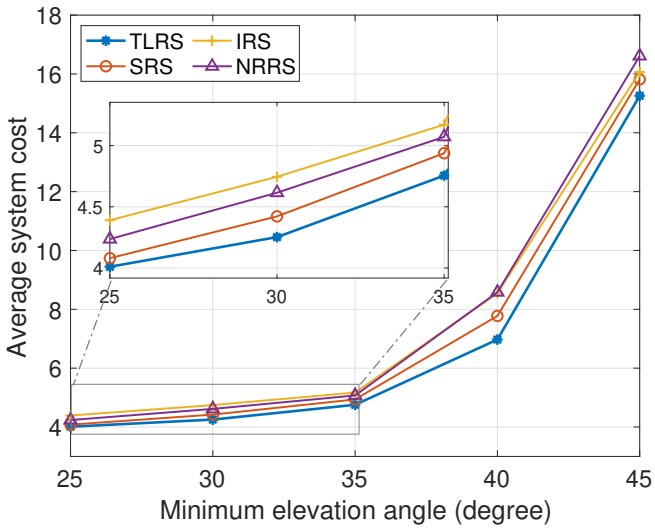
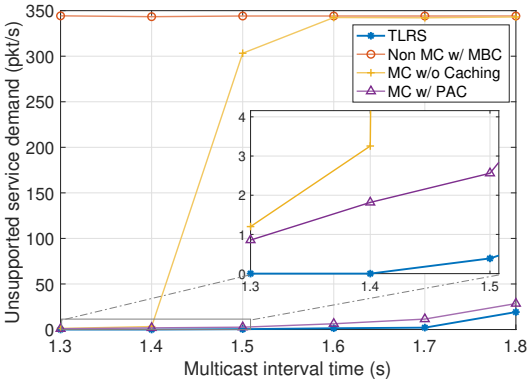


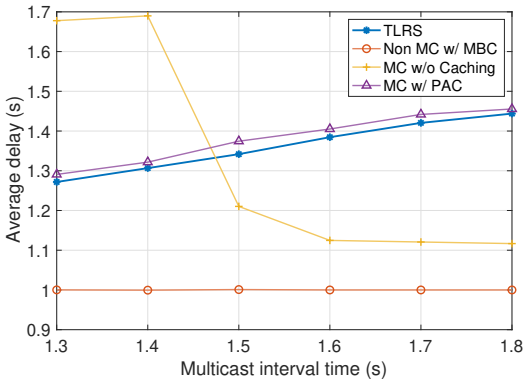
Figure 3.10: Average system cost with different resource reservation policies.

multicast and PAC has a slightly worse performance compared with the TLRS scheme since the caching policy only considering the popularity of the file cannot take full advantage of the time-varying available resources and jointly leverage multicast and unicast feature of satellites. In addition, the policy with multicast but without caching has a sudden increase of unsupported service demand when the multicast interval increases. The main reason is that when the multicast interval time is large, the required queueing and transmission delay in each AP is small. In this case, without caching in satellites, the backhaul delay occurs resulting in difficulties in satisfying the service requirement, which leads to a high unsupported service demand. If the multicast is not adopted, the unsupported service demand is high since the unicast cannot support the huge service demand for some files.

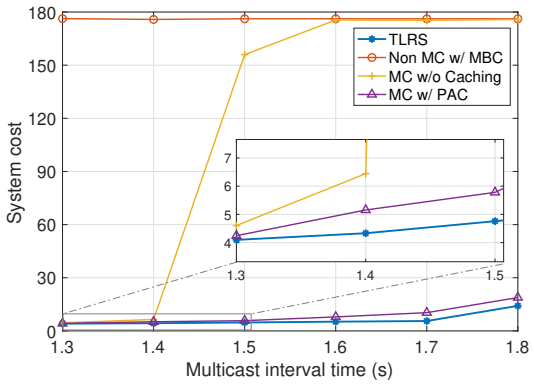
Moreover, the average delay performance is provided in Fig. 3.11(b) with different multicast intervals. When the multicast is not adopted, the average delay is low since there is no waiting time for each multicast event. For the policy adopting multicast without caching, when the multicast interval time is small, the required resources for satisfying delay requirements can support the file fetched directly from the content server. Hence, considering the backhaul delay, the average delay is larger than other strategies. However, when the multicast interval time is large, the delay requirement cannot be satisfied without caching, resulting in a large amount of unsupported service demand. Therefore, the average delay becomes small since it only counts the accommodated service demand mainly from unicast slice. As for the policies consisting of multicast and caching, they have similar delay



(a) Unsupported service demand.



(b) Average delay.



(c) Average system cost.

Figure 3.11: Performance for different caching and multicast policies.

performances while the proposed TLRS scheme can achieve a lower delay.

Finally, the average system cost performance is shown in Fig. 3.11(c), where the proposed TLRS scheme achieves the best performance compared with other strategies. Specifically, we can find that the dominant part of the overall cost is the accommodated service demand. In this case, the performance of the policy consisting of multicast and PAC approaches to the proposed TLRS scheme. On the other hand, the policies without either multicast or caching have a large system cost, especially under a large multicast interval time. Therefore, we can conclude that our proposed TLRS scheme can achieve an adaptive and efficient resource slicing in STIVN.

3.7 Summary

In this chapter, we have investigated a multi-dimensional resource slicing problem for HD map distribution in STIVN. By jointly taking full advantage of satellite and terrestrial networks, we develop a novel resource slicing scheme and establish MS and US considering different transmission modes in STIVN. To cope with time-varying available resources and spatiotemporal dynamics in service demands, we have proposed the TLRS scheme, which is a hybrid data-model co-driven approach for resource slicing in STIVN to jointly find the optimal slicing window length, communication and caching resource reservation, and multicast decisions. The proposed scheme can cooperatively manage resources from both satellite and terrestrial networks to enable reduced delay, low slice reconfiguration cost, enhanced resource utilization efficiency, and adaptability to time-varying service demands while satisfying service requirements.

Chapter 4

Reservation-Stage Resource Management with Cross-Cell Coordination

In this chapter, we investigate a resource management problem in the reservation stage with cross-cell coordination in STIVN to satisfy distinct service delay requirements and efficient resource usage. To address the challenges posed by spatiotemporal dynamics in service demands and satellite mobility, we formulate the resource slicing problem into a long-term optimization problem. Considering the distinct control mechanism between satellite and terrestrial networks in different cells in STIVN, a distributed resource slicing (DRS) scheme is proposed for flexible resource management across different cells. Specifically, an asynchronous multi-agent reinforcement learning-based algorithm is designed to determine the optimal satellite set serving each cell and a distributed optimization algorithm is developed to make the resource reservation decisions for each slice. Simulation results demonstrate that the proposed scheme outperforms benchmark methods in terms of resource usage and delay performance.

4.1 Background and Motivations

The next-generation networks are envisioned to support a wide range of emerging services anytime and anywhere [16]. LEO satellites, thanks to their large communication coverage and global deployment, show great potential in assisting the existing terrestrial networks

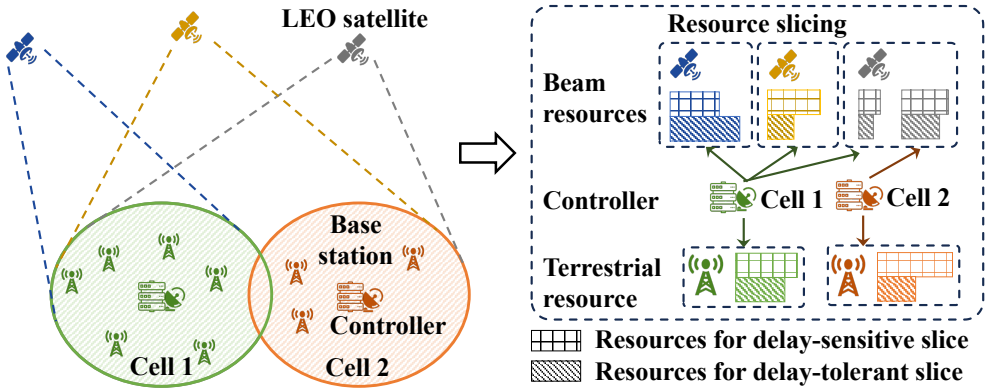


Figure 4.1: Resource slicing across multiple cells in STIVN.

in providing ubiquitous connectivity and seamless service provision, especially in remote areas. Therefore, by cooperatively leveraging the complementary advantages of satellite and terrestrial networks, STIVN emerges as a promising network architecture to facilitate services with various QoS requirements in the 6G era. This encompasses a wide range of network-reliant services, such as V2X applications from delay-sensitive environment sensing services with stringent delay requirements to delay-tolerant services like HD map distribution and multimedia streaming requiring high throughput [8].

To accommodate multifarious services, resource slicing enables the construction of multiple virtual networks, referred to as slices, on top of the shared physical network infrastructure to realize resource isolation while satisfying different service-level agreements. To maintain the effectiveness of slices, ensuring that they can satisfy QoS requirements in dynamic network conditions with efficient resource utilization, it is crucial to make periodic adjustments to slices at specific intervals referred to as slicing windows. In STIVN, the terrestrial network consists of multiple cells, with the area of each cell being equal to the coverage area of a satellite beam. Within each cell, multiple slices can be created to support different services, and there should be a controller responsible for managing both the terrestrial and satellite resources for different slices, as shown in Fig. 4.1. Moreover, with the large coverage of LEO satellites, resources from each satellite can be shared across different terrestrial cells, introducing the concept of cross-cell coordination of communication resources.

To investigate the resource slicing problem with cross-cell coordination in STIVN, three challenges need to be addressed. First, each satellite has large coverage, which can serve different terrestrial cells using multiple beams by adopting the beam-hopping technique to

reuse limited beam resources [104], leading to resource competition among these cells. In addition, the dense satellite deployment enables multiple satellites to concurrently serve the same terrestrial cell with frequency reuse or beamforming techniques [105]. In this case, implementing a centralized resource slicing approach to derive the optimal decisions will encounter formidable complexity since it requires extensive interactions between different controllers and satellites. Second, the spatiotemporal dynamics in service demands across different slices and cells result in different lengths of slicing windows as well as asynchronous decision-making in each cell, which further complicates the decision-making process. Third, due to the mobility of satellites, the amount of resources available for each cell is time-varying and distinct, i.e., for each cell, the number of satellites serving the cell may vary over time and the connection duration to each satellite may be different, posing a challenge to conventional slicing methods designed for cases with fixed resources.

In this work, we propose a distributed resource slicing (DRS) scheme with cross-cell coordination in STIVN to support scalable and flexible resource management in each cell. Our objective is to reserve resources from terrestrial and satellite networks for each slice to minimize the overall cost related to resource usage and delay performance. Considering the distinct service demand and time-varying available resources, we design a novel resource slicing architecture in STIVN for cross-cell coordination. Specifically, a hybrid data-model co-driven approach is developed to find the optimal solutions in a less time-consuming manner, where an asynchronous multi-agent proximal policy optimization (AMAPPO)-based algorithm is presented to select satellites serving each cell in each slicing window, and a distributed optimization algorithm is employed to determine the resource reservation decisions for each slice. The main contributions of this work are summarized as follows:

- We investigate the resource slicing problem across multiple terrestrial cells for both delay-tolerant and delay-sensitive applications as two slices in STIVN. To satisfy diversified service requirements while considering distinct service demand and time-varying available resources, we design a novel resource slicing architecture in STIVN for cross-cell coordination. Then, we formulate the resource slicing problem into a long-term optimization problem, aiming to satisfy distinct delay requirements with efficient resource utilization.

- We propose a DRS scheme to realize scalable and flexible resource management across different cells. In this scheme, considering that the resource reservation decisions remain unchanged within each slicing window, the original problem is decoupled into two subproblems to reduce the complexity of addressing non-linear stochastic sequential-decision making problem. Specifically, the accessible satellites are selected

for the slicing window in each cell and then resource reservation decisions are made with a given satellite set in each slicing window.

- We design a hybrid data-model co-driven approach is developed to find the optimal solutions in a less time-consuming manner, where an asynchronous multi-agent proximal policy optimization (AMAPPO)-based algorithm is presented to select satellites serving each cell in each slicing window, and an ADMM-based distributed optimization algorithm is employed to determine the resource reservation decisions for each slice.

The remainder of this chapter is organized as follows. The considered network scenario and system models are introduced in Sec. 4.2. The problem is formulated in Sec. 4.3. Then, the proposed distributed resource slicing scheme is proposed in Sec. 4.4, including a distributed ADMM-based optimization algorithm for resource reservation and an RL-based satellite selection algorithm. Performance evaluation is given in Sec. 4.5. Finally, in Sec. 4.6, we conclude this chapter.

4.2 Network Scenario and System Model

In this section, we first introduce our considered STIVN scenario with the resource slicing architecture for cross-cell coordination and then present the system models.

4.2.1 Scenario Description

As shown in Fig. 4.1, we consider an STIVN scenario where the terrestrial network consists of multiple cells, with the area of each cell being equal to the coverage area of a satellite beam. The service demand in each terrestrial cell can be served by either terrestrial base stations or LEO satellites. Let $\mathcal{N} = \{1, 2, ..., N\}$ denote the set of terrestrial cells, where a controller is deployed in each cell for managing communication resources for different services in the cell. Denote the set of satellites within a constellation by $\mathcal{S}$. Each satellite has K beams ($K < N$), with each beam covering one terrestrial cell. The available communication bandwidth in each beam is denoted by B^{S}. Moreover, there are A_n terrestrial base stations deployed in cell $n \in \mathcal{N}$ where each base station can provide the communication resources with bandwidth B^{T}. In this work, we conceptualize the provisioning of terrestrial communication resources within a cell through a virtual terrestrial

access point. Note that different spectrum frequencies are used for different satellites and terrestrial communications in each cell to avoid co-channel interference.

Consider a time-slotted system where each time slot $t \in \{1, 2, ..., T\}$ has the same duration, denoted by τ. Due to the mobility of satellites, the set of accessible satellites for cell n in time slot t is time-varying, denoted by $\mathcal{S}_{n,t}^{\text{ts}} \subset \mathcal{S}$. In this case, let $\mathcal{AP}_{n,t}^{\text{ts}} = \{0\} \cup \mathcal{S}_{n,t}^{\text{ts}}$ denote the set of all access points being able to serve cell n in time slot t, where $ap = 0$ indicates the terrestrial access point. Let $a_{n,ap,t}$ denote whether access point ap can serve cell n in time slot t or not, where $a_{n,ap,t} = 1$ indicates that access point ap can serve users in cell n; otherwise, $a_{n,ap,t} = 0$.

In this work, the supported services can be classified into two categories: delay-sensitive services with specific delay and reliability requirements and delay-tolerant services without delay requirements. Two slices, indexed by $l = 1$ and $l = 2$, are created for the delay-sensitive services and delay-tolerant services, respectively. Consider the service demand for each service follows a non-homogeneous Poisson Process (NHPP), i.e., the intensity of the Poisson process varies over time slots. Let $\lambda_{n,l,t}$ denote the service demand intensity of slice l in cell n in time slot t. Specifically, due to the potential intermittent network connection from terrestrial networks in each cell, the service demand can be represented by $\lambda_{n,l,t} = \lambda_{n,l,t}^{\text{C}} + \lambda_{n,l,t}^{\text{NC}}$, where $\lambda_{n,l,t}^{\text{C}}$ is the service demand intensity within the coverage of terrestrial networks, and $\lambda_{n,l,t}^{\text{NC}}$ is the service demand only being able to be served by satellite networks. Due to the variation in service demands, the resource slicing decisions should be periodically adjusted to guarantee the effectiveness of resource management, where the interval of the adjustment is referred to as the slicing window. By analyzing historical data using AI-based prediction methods, in this work, we assume that the controller can predict the service demand of each slice for the following $W_{\max}$ time slots. Therefore, the slicing window length in each terrestrial cell should be determined according to the service demand variation. Let $\mathcal{W}_n^i = \{t_{n,i}, ..., t_{n,i} + w_{n,i} - 1\}$ represent the corresponding time slot set for i-th slicing window of cell n, where $t_{n,i}$ is the initial time slot of the slicing window, and $w_{n,i} \in \{1, ..., W_{\max}\}$ is the slicing window length. Let $\mathcal{AP}_{n,i} = \bigcup_{t \in \mathcal{W}_n^i} \mathcal{AP}_{n,t}^{\text{ts}}$ represents the set of access points that can serve cell n within slicing window i.

4.2.2 Communication Model

For resource slicing in each slicing window, we aim to reserve communication resources from different access points to each slice to satisfy different service requirements. Let $b_{l,t}^{n,ap}$ denote the ratio of reserved resources in access point ap for slice l in cell n at time slot t. Specifically, if $ap = 0$, this refers to the ratio of terrestrial resources in cell n reserved for

the slice, otherwise, it indicates the ratio of the reserved resources in a beam of ap. Once the resources are reserved, they cannot be adjusted within the slicing window, given by

$$b_{l,t+1}^{n,ap} = b_{l,t}^{n,ap}, \forall t, t+1 \in \mathcal{W}_n^i, a_{n,ap,t} = a_{n,ap,t+1} = 1. \tag{4.1}$$

Due to the time-varying nature of stochastic service demands, it is difficult to satisfy the deterministic QoS requirement for every network slice. To model the statistical delay performance, effective capacity emerges as a valuable approach [95]. Specifically, it quantifies the maximum arrival rate that can be accommodated by reserved resources for each slice, facilitating the evaluation of delay-bounded violation probability that the overall delay $D_{l,t}^{n,ap}$ exceeds the delay bound $D_{n,ap,l,t}^{\mathrm{B}}$, presented as [106]

$$P(D_{l,t}^{n,ap} \geq D_{n,ap,l,t}^{\mathrm{B}}) \approx e^{-\theta C(R_{l,t}^{n,ap},\theta) D_{n,ap,l,t}^{\mathrm{B}}}, \tag{4.2}$$

where $C(\cdot, \theta)$ is the effective capacity, and θ is the QoS exponent representing the exponential decay rate of the QoS violation probability according to the large derivation theory [95], in which the value of θ is assumed to be the same for queues in each access point. For a long-term planning problem, we consider the service rate $R_{l,t}^{n,ap}$ to be mainly affected by the average communication distance between users and access points. In this case, the packet buffer at each access point for each slice can be modeled as an M/D/1 queueing system in each time slot, where the service rate counting in packets in the queue can be given by

$$R_{l,t}^{n,ap} = \frac{b_{l,t}^{n,ap} B_{ap}}{L_l} \log_2(1 + \frac{P_{ap}|h_{n,ap,t}|^2}{\sigma^2}), \tag{4.3}$$

where L_l is the packet size for slice l, P_{ap} is the transmission power of ap, $h_{n,ap,t}$ is the channel gain of the transmission between users in cell n and access point ap, which can be presented as $h_{n,ap,t} = (d_{n,ap,t})^{-\delta_{ap}}$, with the pathloss exponent δ_{ap} of access point ap and the distance $d_{n,ap,t}$ between the central point of cell n and access point ap in time slot t. Denote the entire bandwidth of access point ap by B_{ap}. When $ap = 0$, $B_{ap} = B^{\mathrm{T}}$ for analyzing effective capacity in each base station to determine the delay-bounded violation probability served by terrestrial networks, otherwise, $B_{ap} = B^{\mathrm{S}}$. Moreover, $D_{n,ap,l,t}^{\mathrm{B}} = D_l^{\mathrm{B}} - \mathbb{1}_{ap \in \mathcal{S}} D_{n,ap,t}^{\mathrm{Prop}}$ considering the propagation delay for satellite communication with

$$D_{n,ap,t}^{\mathrm{Prop}} = d_{n,ap,t}/c, \tag{4.4}$$

where D_l^{B} is the specified delay bound for services in slice l and c is the light speed.

Due to the deterministic service rate, the effective capacity is defined as

$$C(R_{l,t}^{n,ap}, \theta) = -\frac{1}{\theta} \log\left(\mathbb{E}\left\{e^{-\theta R_{l,t}^{n,ap}}\right\}\right) = R_{l,t}^{n,ap}. \tag{4.5}$$

Based on the effective capacity, with a given reserved resources from access point ap, the maximal supported intensity for the Poisson process can be given by [107]

$$\tilde{\lambda}_{n,ap,l,t} = \begin{cases} \min\left\{\dfrac{\theta C(R_{l,t}^{n,ap},\theta)}{e^{\theta}-1}A_n, \lambda_{n,l,t}^{\mathrm{C}}\right\}, & \text{if } ap = 0, \\ \dfrac{\theta C(R_{l,t}^{n,ap},\theta)}{e^{\theta}-1}, & \text{otherwise}, \end{cases} \tag{4.6}$$

where the maximum supported service demand from terrestrial networks is limited by $\lambda_{n,l,t}^{\mathrm{C}}$. To satisfy the service demand for each slice, the total supported service demand $\tilde{\lambda}_{n,l,t}^{\mathrm{tot}}$ should exceed the service demand for each slice, given by

$$\tilde{\lambda}_{n,l,t}^{\mathrm{tot}} = \sum_{ap \in \mathcal{AP}_{n,t}^{\mathrm{ts}}} \tilde{\lambda}_{n,ap,l,t} \geq \lambda_{n,l,t}. \tag{4.7}$$

4.3 Problem Formulation and Decomposition

4.3.1 Problem Formulation

Considering the dynamic network environments with time-varying service demand and available resources for each cell, we aim to minimize the long-term overall system cost by reserving communication resources from different access points. The overall system cost includes the resource cost from both terrestrial and satellite networks and the dissatisfaction cost for the delay-tolerant service. Specifically, the resource cost in time slot t for cell n is the sum of resource cost in different slices from different access points serving the target area, which can be given by

$$C_{n,t}^{\mathrm{res}} = \sum_{l} \sum_{ap \in \mathcal{AP}_{n,t}^{\mathrm{ts}}} \alpha_{ap} b_{l,t}^{n,ap} B_{ap}, \tag{4.8}$$

where α_{ap} is the cost of a resource unit for access point ap.

Moreover, although there are no stringent delay requirements for the delay-tolerant service, each user may have an expected delay performance given by D_2^{B}. Denote the delay dissatisfaction probability by

$$W_{n,ap,l,t}^{\mathrm{dis}} = \Pr(D_{l,t}^{n,ap} > D_{n,ap,l,t}^{\mathrm{B}}) \mathbb{1}_{b_{l,t}^{n,ap}>0}, \tag{4.9}$$

indicating the probability of the delay exceeding the expectation for slice l of cell n in time slot t. In this case, the dissatisfaction cost in time slot t for cell n can be represented by the maximum delay dissatisfaction probability from all access points serving users in the cell, given by

$$C_{n,t}^{\text{dis}} = \max_{ap \in \mathcal{AP}_{n,t}^{\text{ts}}} W_{n,ap,2,t}^{\text{dis}} \ . \tag{4.10}$$

Let $\mathbf{b}_{n,ap,l} = \{b_{l,1}^{n,ap}, ..., b_{l,T}^{n,ap}\}$ denote the vector of the resource reservation decision for slice l in cell n from access point ap within T time slots. To this end, we formulate the resource slicing problem in STIVN to minimize the long-term overall system cost while satisfying delay and reliability requirements in the presence of network dynamics, given by

$$\begin{aligned}
\mathbf{P0} \ \min_{\mathbf{b}_{n,ap,l}} \quad & \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{n\in\mathcal{N}} C_{n,t}^{\text{sys}}\right] && (4.11)\\
\text{s.t.} \quad & W_{n,ap,1,t}^{\text{dis}} \leq \varepsilon, && (4.11\text{a})\\
& 0 \leq \sum_{l} b_{l,t}^{n,ap} \leq a_{n,ap,t}, && (4.11\text{b})\\
& \sum_{n} \sum_{l} b_{l,t}^{n,ap} \leq K, \forall ap \in \mathcal{S}, && (4.11\text{c})\\
& (4.1), (4.7),
\end{aligned}$$

where

$$C_{n,t}^{\text{sys}} = \beta_1 C_{n,t}^{\text{res}} + \beta_2 C_{n,t}^{\text{dis}}, \tag{4.12}$$

with β_1 and β_2 representing the weights for two costs. Constraint (4.11a) represents that the delay dissatisfaction probability should be less than ε as the delay and reliability requirement of delay-sensitive services; constraint (4.11b) indicates that for each access point, the overall ratio of reserved resources for different slices in one cell should be between 0 and 1 if the connectivity is accessible, otherwise, the ratio should be 0; and constraint (4.11c) limits the maximum available beam resources from each satellite.

4.3.2 Problem Decomposition

Note that the formulated problem **P0** is a non-linear stochastic optimization problem with spatiotemporal dynamic service demands and time-varying available resources in STIVN. Therefore, conventional optimization methods face significant challenges in solving such long-term optimization problems with the space and time interactions among different cells. Furthermore, the continuous decision space for decisions from each access point

for all slices makes the direct application of model-free techniques impractical. Thus, we propose a distributed resource slicing (DRS) scheme to solve the problem. Specifically, to reduce the decision dimension, considering that the resource reservation decisions remain unchanged within each slicing window, we decouple the original problem into the following two sub-problems: 1) resource reservation problem with a given access point set in each slicing window obtained from the second subproblem, and 2) access point selection problem for the slicing window in each cell.

In the first subproblem, with a set of access points to serve the target cell, each controller determines the reservation ratio in each slicing window to minimize the system cost, given by

$$\mathbf{P1} \quad \min_{b_{l,t}^{n,ap}} \quad \sum_{t=t_{n,i}}^{t_{n,i}+w_{n,i}} C_{n,t}^{\text{sys}} \tag{4.13}$$

$$\text{s.t.} \quad ap \in \mathcal{AP}_{n,i}^{*}, \tag{4.13a}$$

$$(4.1), (4.7), (4.11a), (4.11b), (4.11c),$$

where $\mathcal{AP}_{n,i}^{*}$ is the selected access point set for i-th slicing window in cell n.

In the second subproblem, we find the optimal set of access points to serve each cell in each slicing window for a long-term optimal overall system cost, which is given by

$$\mathbf{P2} \quad \min_{\mathcal{AP}_{n,i}^{*}} \quad \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{n\in\mathcal{N}} C_{n,t}^{\text{sys}}\right] \tag{4.14}$$

$$\text{s.t.} \quad \mathcal{AP}_{n,i}^{*} \subseteq \mathcal{AP}_{n,i}, \tag{4.14a}$$

$$(4.1), (4.7), (4.11a), (4.11b), (4.11c).$$

4.4 Distributed Resource Slicing Scheme with Cross-Cell Coordination

In this section, we first derive the distributed optimization algorithm for the resource reservation algorithm in each cell given the selection of serving access points and then develop an RL-based access point selection algorithm.

4.4.1 Distributed Optimization Algorithm for Resource Reservation

In our considered STIVN, one satellite can cover multiple terrestrial cells, leading to potential resource competition among different cells. Since different controllers independently make decisions without direct communication, it is hard to design a centralized optimization approach to find the optimal decisions for all cells to solve **P1**. Therefore, we design a distributed optimization algorithm for resource reservation with centralized feedback from satellites, where the resource utilization of a satellite can be reported to its served cells to help the controller decisions. The designed algorithm is implemented in a two-step process involving local solution derivation and global decision execution.

In the first step, each controller finds the local solution $[\tilde{b}_{l,t}^{n,ap}]^{|\mathcal{AP}_{n,i}^*|\times w_{n,i}}$ at the beginning of each slicing window to solve **P1** by neglecting constraint (4.11c) for the potential resource competition. Since the objective function (4.13) and constraint (4.11a) includes a conditional expression related to the variable $\tilde{b}_{l,t}^{n,ap}$, the problem cannot be directly solved by the optimization technique. In this case, we add an additional binary variable $r_{l,t}^{n,ap}$ to constrain the resource reservation decision, equivalently transforming **P1** into

$$\mathbf{P1}' \quad \min_{\tilde{b}_{l,t}^{n,ap}, r_{l,t}^{n,ap}} \quad \sum_{t=t_{n,i}}^{t_{n,i}+w_{n,i}} C_{n,t}^{\mathrm{sys}'} \tag{4.15}$$

$$\text{s.t.} \quad \Pr(D_{1,t}^{n,ap} > D_{n,ap,1,t}^{\mathrm{B}}) + r_{l,t}^{n,ap} - 1 < \varepsilon, \tag{4.15a}$$

$$\tilde{b}_{l,t}^{n,ap} \leq r_{l,t}^{n,ap}, \tag{4.15b}$$

$$r_{l,t}^{n,ap} \in \{0,1\}, \tag{4.15c}$$

$$(4.1), (4.7), (4.11a), (4.11b), (4.13a).$$

Specifically,

$$C_{n,t}^{\mathrm{sys}'} = \beta_1 C_{n,t}^{\mathrm{res}} + \beta_2 C_{n,t}^{\mathrm{dis}'}, \tag{4.16}$$

where

$$C_{n,t}^{\mathrm{dis}'} = \max_{ap \in \mathcal{AP}_{n,t}} \Pr(D_{2,t}^{n,ap} > D_{n,ap,2,t}^{\mathrm{B}}) + r_{l,t}^{n,ap} - 1. \tag{4.17}$$

Since the formulated mixed integer programming problem with conic constraints is a convex problem, the local solution for each cell $[\tilde{b}_{l,t}^{n,ap}]^{|\mathcal{AP}_{n,i}^*|\times w_{n,i}}$ can therefore be directly obtained via convex optimization solvers.

Given local solutions, each controller then executes the decision for each access point. Considering the potential exceeded utilization of the beam resources from each satellite,

an alternating direction method of multipliers (ADMM)-based iteratively distributed optimization algorithm is designed to reach the final feasible decision execution. Specifically, ADMM is a highly efficient algorithm applicable to distributed convex optimization [108,109], which has been proved with a rapid convergence speed to a highly accurate global optimal solution [110]. In distributed ADMM, a global consensus problem is divided into several local subproblems, with the solutions to these local subproblems being coordinated to achieve a consensus solution for the original problem.

Taking the inequality in Eq. (4.11c) into account, the corresponding augmented Lagrangian expression is given by

$$\begin{aligned} L(\{\tilde{b}_{l,t}^{n,ap}\},\{r_{l,t}^{n,ap}\},\{y_{ap,i}\},\{z_{ap,i}\}) = \\ \sum_{t=t_{n,i}}^{t_{n,i}+w_{n,i}} \left(\sum_{n\in\mathcal{N}} C_{n,t}^{\text{sys}'} + \sum_{ap\in\mathcal{AP}_{n,i}^{*}} y_{ap,i} \left(\sum_{n\in\mathcal{N}} \sum_{l} \tilde{b}_{l,t}^{n,ap} + z_{ap,i} - K \right) + \right. \\ \left. \frac{\rho}{2} \sum_{ap\in\mathcal{AP}_{n,i}^{*}} \left(\sum_{n\in\mathcal{N}} \sum_{l} \tilde{b}_{l,t}^{n,ap} + z_{ap,i} - K \right)^2 \right), \end{aligned} \tag{4.18}$$

where $y_{ap,i}$ is the Lagrange multipliers for ap in the slicing window i for illustrating the impact from resource collision, $z_{ap,i}$ is the constraint of the overall available resources transformed from Eq. (4.11c), and ρ is the is a positive constant representing the penalty. Then, in each iteration, variables $\{\tilde{b}_{l,t}^{n,ap}\}$, $\{r_{l,t}^{n,ap}\}$, $\{y_{ap,i}\}$, and $\{z_{ap,i}\}$ should be updated sequentially for a global consensus. Specifically, with given $\{y_{ap,i}\}$ and $\{z_{ap,i}\}$, the optimal $\{\tilde{b}_{l,t}^{n,ap}\}$ and $\{r_{l,t}^{n,ap}\}$ in the current iteration can be obtained by solving the following optimization problem:

$$\begin{aligned} \mathbf{P1''} \quad \min_{\tilde{b}_{l,t}^{n,ap}, r_{l,t}^{n,ap}} \quad & L(\{\tilde{b}_{l,t}^{n,ap}\},\{r_{l,t}^{n,ap}\},\{y_{ap,i}\},\{z_{ap,i}\}) \\ \text{s.t.} \quad & (4.1), (4.7), (4.11a), (4.11b), (4.13a), (4.15a), (4.15b), (4.15c). \end{aligned} \tag{4.19}$$

Then, for each access point, with the updated $\{\tilde{b}_{l,t}^{n,ap}\}$ and $\{r_{l,t}^{n,ap}\}$, the Lagrange multipliers can be updated as

$$y_{ap,i} \leftarrow y_{ap,i} + \rho \sum_{t=t_{n,i}}^{t_{n,i}+w_{n,i}} \left(\sum_{n\in\mathcal{N}} \sum_{l} \tilde{b}_{l,t}^{n,ap} + z_{ap,i} - K \right), \tag{4.20}$$

and the constraint variable $z_{ap,i}$ can be updated by solving

$$\frac{\partial L(\{\tilde{b}_{l,t}^{n,ap}\},\{r_{l,t}^{n,ap}\},\{y_{ap,i}\},\{z_{ap,i}\})}{\partial z_{ap,i}} = 0. \tag{4.21}$$

Algorithm 3 Iteratively distributed optimization algorithm

1: **Initialization:** Local solution $\tilde{b}_{l,t}^{n,ap}$;
2: **while** $u < iter_{\max}$ and not converge **do**
3: **for** $ap \in \mathcal{S}_{n,t}$ **do**
4: **if** $\tilde{b}_{l,t}^{n,ap}$ not executed and $a_{l,t}^{n,ap} = 1$ **then**
5: Send the local solution to access points;
6: Each access point updates the Lagrange multiplier $y_{ap,i}$ and constraint value $z_{ap,i}$ according to Eqs. (4.20) and (4.21);
7: Send the updated variables to each controller;
8: **end if**
9: **end for**
10: **if** $\exists$ Eq. (4.11c) not satisfied **then**
11: Update $\tilde{b}_{l,t}^{n,ap}$ by solving **P1**$''$;
12: Go back to line 3;
13: **end if**
14: **end while**
15: If exceeding the capacity of each access point, proportionally reduced resource values;
16: Execute decisions to $b_{l,t}^{n,ap}$ and update $\tilde{b}_{l,t}^{n,ap}$;
17: **Output:** $\tilde{b}_{l,t}^{n,ap}$ and $b_{l,t}^{n,ap}$;

The main algorithm is given in Algorithm 3 for the decision execution. With the local solution of each controller, when the access point becomes accessible within the time window, the controller will send local solutions to the corresponding access point for execution. If the local solutions overuse the satellite resources, the Lagrange multiplier and constraint variables will update in the corresponding access points and send back to each controller as shown in lines 3-7. Otherwise, the reserved resource will remain the same within the slicing window while the satellite is accessible. Then, each controller needs to solve **P1**$''$ until reaching the maximum number of iterations or the objective function value converges, given by line 11 to 12. Finally, the controllers execute decisions and update the local solutions.

4.4.2 Reinforcement Learning-Based Access Point Selection

Next, we will investigate the optimal access point selection for each cell in each slicing window. Since the formulated access point selection problem is an integer nonlinear optimization problem with asynchronous decision-making from each controller, it is complex to obtain optimal decisions by deterministic optimization algorithms. Thus, we reformulate problem **P2** as a decentralized partially observable Semi-Markov decision process (Dec-

POSMDP) [111] to model the time-varying service demand and interaction among different cells.

We model the Dec-POSMDP as a tuple $(\mathcal{N}, \mathfrak{S}, \underline{\mathfrak{A}}, \mathcal{P}, \underline{\mathfrak{R}}^w, \mathcal{W}, \gamma, \underline{\mathcal{O}}, Z)$, including the agent space $\mathcal{N}$ represented by the set of controllers in each cell, the state space $\mathfrak{S}$, the action space of each controller $\underline{\mathfrak{A}}$, the state transition function $\mathcal{P} := \mathfrak{S} \times \underline{\mathfrak{A}} \times \mathfrak{S} \to [0,1]$, the reward function for each slicing window $\mathfrak{R}^w := \mathfrak{S} \times \underline{\mathfrak{A}} \times \mathcal{W} \to \mathbb{R}$, slicing window space $\mathcal{W}$, discounted factor γ, observation spaces from each controller $\underline{\mathcal{O}}$, and observation function $\mathcal{Z} := \mathfrak{S} \times \underline{\mathfrak{A}} \times \underline{\mathcal{O}} \to [0,1]$, where $\underline{\cdot}$ represents the joint parameters for all controllers. Let $\zeta_n^i = [\lambda_{n,l,t'}, POS_{ap,t'}, a_{n,ap,t'}, w_{n,i}, \tilde{b}_{l,t_{n,i}}^{n,ap}]_{ap \in \mathcal{AP}_{n,i}, t' \in \mathcal{W}_n^i} \in \underline{\mathcal{O}}$ denote the observation of the controller in cell n in i-th slicing window, where $POS_{ap,t'}$ is the position of access point ap. The state in time slot t is the joint observation of each controller, given by $\Xi_t = \bigcup_{\{\{n,i\}|t \in \mathcal{W}_n^i\}} \zeta_n^i$. The action in slicing window i for the controller in cell n is denoted by $\mathcal{A}_n^i = \mathcal{AP}_{n,i}^* \in \mathfrak{A}$ for the access point selection, which is determined at the beginning of each slicing window in each cell. To evaluate an action in a state, we define the obtained reward for each controller in each slicing window as

$$\mathcal{R}_{n,i}^w(\zeta_n^i, \mathcal{A}_n^i, \mathcal{W}_n^i) = \sum\nolimits_{t \in \mathcal{W}_n^i} \gamma^{t - t_{n,i}} (-C_{n,t}^{\text{sys}} - V_{n,t}^{\text{pen}}), \tag{4.22}$$

where $V_{n,t}$ is the penalty value indicating the violation of constraints (4.7) and (4.11a), given by

$$V_{n,t}^{\text{pen}} = p_1 \sum_l \mathbb{1}_{\tilde{\lambda}_{n,l,t}^{\text{tot}} < \lambda_{n,l,t}} + p_2 \max_{ap \in \mathcal{AP}} \{\mathbb{1}_{W_{n,ap,1,t}^{\text{dis}} > \varepsilon}\} \tag{4.23}$$

with the penalty weights p_1 and p_2, and $\gamma \in (0,1)$ reflects the importance of future reward compared with immediate reward. Let $\mathbf{\Pi}$ denote the set of all policies. To estimate the expected return of the current decision-making, a discounted cumulative reward function is considered and the objective function is transformed as

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{i'=0}^{\infty} \sum_{n \in \mathcal{N}} \gamma^{t_{n,i'}} \mathcal{R}_{n,i}^w(\zeta_n^i, \mathcal{A}_n^i, \mathcal{W}_n^i) \middle| \pi\right]. \tag{4.24}$$

To solve the problem, we propose an asynchronous multi-agent proximal policy optimization (AMAPPO)-based algorithm [112], where each agent can take actions asynchronously. A centralized training and decentralized execution (CTDE) paradigm is adopted, where each controller makes decisions based on individual observations and updates the policy from the global state in a centralized manner. For each controller, the actor network parameterized by ϕ is responsible for the selection of access points for each slicing window in its cell given the local observation. The critic network parameterized by ϑ is leveraged to approximate

Algorithm 4 AMAPPO-based access point selection algorithm

Initialization: Parameters ϕ, ϑ, and buffer $\mathcal{B} = \{\}$;
while *episode* $\leq$ *episode*$_{\max}$ **do**
  for $t \in \{1, 2, ..., T\}$ **do**
    Obtain the state Ξ_t;
    for $n \in \mathcal{N}$ **do**
      if $t = t_{n,i}$ **then**
        Obtain the observation ζ_n^i and action of access point selection $\mathcal{A}_n^i$ based on the policy π_ϕ;
        Solving $\mathbf{P1}'$ to obtain the initial local solution;
      end if
    end for
    Apply Algorithm 3 for reserving resources;
    for $n \in \mathcal{N}$ **do**
      if $t = t_{n,i} + w_{n,i} - 1$ **then**
        Compute reward $\mathcal{R}_{n,i}^w(\zeta_n^i, \mathcal{A}_n^i, \mathcal{W}_n^i)$ and insert transition data into buffer $\mathcal{B}$;
      end if
    end for
  end for
  Update the policy parameters $\phi \leftarrow \arg\max_\phi L^{\text{clip}}(\phi)$;
  Update critic network parameter
  $\vartheta \leftarrow \arg\min_\vartheta \frac{1}{|\mathcal{B}|} \sum_{(n,i)\in\mathcal{B}} (\mathbb{E}\big[\mathcal{R}_{n,i}^w + V_\vartheta(\Xi_{t_{n,i+1}}) - V_\vartheta(\Xi_{t_{n,i}})\big])^2$
  Empty buffer $\mathcal{B}$;
end while
Output: Parameter ϕ;

a state-value function by generating the value of the given global state for updating the actor networks. Due to multiple satellites that can serve a cell in a slicing window, a large action dimension is introduced for selecting multiple satellites from a controller. Moreover, considering the mobility of LEO satellites, the accessible satellites for each controller in different slicing windows may be different. In this case, the action can be presented as a multi-discrete action set [113], given by $\mathcal{A}_n^i = \{\mathcal{A}_{n,ap}^i \in \{0,1\} \mid ap \in \mathcal{AP}_{n,i}\}$, with the corresponding mask to avoid some inaccessible access points being selected, where $\mathcal{A}_{n,ap}^i = 1$ represents the access point ap will serve cell n in i-th slicing window.

For the AMAPPO-based algorithm, the goal is to find the optimal parameter ϕ for the policy that maximizes the following clipped surrogate objective [103]

$$L^{\text{clip}}(\phi) = \mathbb{E}\{\min(\xi_\phi A^{\pi_{\phi_{\text{old}}}}, \text{clip}(\xi_\phi, 1-\delta, 1+\delta) A^{\pi_{\phi_{\text{old}}}})\}, \tag{4.25}$$

where $\xi_\phi = \pi_\phi(\mathcal{A}|\zeta)/\pi_{\phi_{\text{old}}}(\mathcal{A}|\zeta)$ is the policy updating ratio based on current policy for ϕ.

Moreover, $A^{\pi_{\phi_{\text{old}}}}$ is the estimated advantage from policy $\pi_{\phi_{\text{old}}}$, which can be expressed as

$$A^{\pi_{\phi_{\text{old}}}}(\zeta_n^i, \mathcal{A}_n^i, \mathcal{W}_n^i) = \mathcal{R}_{n,i}^w(\zeta_n^i, \mathcal{A}_n^i, \mathcal{W}_n^i) + \gamma^{\Delta t_{n,i}^{i+1}} V_\vartheta(\Xi_{t_{n,i+1}}) - V_\vartheta(\Xi_{t_{n,i}}), \tag{4.26}$$

where $\Delta t_{n,i}^{i+1} = t_{n,i+1} - t_{n,i}$. The state-value function of the critic network is given by

$$V_\vartheta(\Xi_{t_{n,i}}) = \mathbb{E}\left\{\sum_{k=0}^{\infty} \gamma^{\Delta t_{n,i}^{i+k}} \mathcal{R}_{n,i}^w(\zeta_n^{i+1+k}, \mathcal{A}_n^{i+1+k}, \mathcal{W}_n^{i+1+k}) \,|\, \Xi_{t_{n,i}}, n\right\}. \tag{4.27}$$

The clipping function clips the ratio ξ_ϕ between $1-\delta$ and $1+\delta$ with a clipping hyperparameter δ, which removes incentives for the new policy to get far from the old policy to improve the learning stability.

The overall AMAPPO-based resource slicing algorithm is given in Algorithm 4. The buffer $\mathcal{B}$ is used to store the transitions of the interaction with the environment. As shown from lines 5-14, at the beginning of each slicing window, the controller of each cell obtains the observation from the environment and takes actions based on the current policy. At the end of each slicing window, the controller calculates the reward and records transition data into the buffer, including the state, observation, action, reward of a slicing window, and the next state. At the end of each episode, the policy and value function will be updated based on the clipped surrogate objective and the mean-square error based on lines 18-19.

4.5 Performance Evaluation

In this section, we present simulation results to demonstrate the performance of our proposed DRS scheme in STIVN. We adopt the Starlink Phase 1 for the satellite constellation parameters, including 72 orbits with 22 satellites on each orbit. The inclination angle is 53 degrees and the altitude of the satellite is 550 km. By default, we set the elevation angle to be 40 degrees. Each satellite has 3 beams to serve different cells simultaneously. The satellite transmit power is set to 10 dBW, with an antenna gain of 50 dBi and pathloss factor of 2.5. The transmit power of base stations is 32 dBm with an antenna gain of 10 dBi and pathloss factor of 3.5. In the simulation, we consider 7 cells in total, where each cell has different numbers of base stations and diversified service demand.

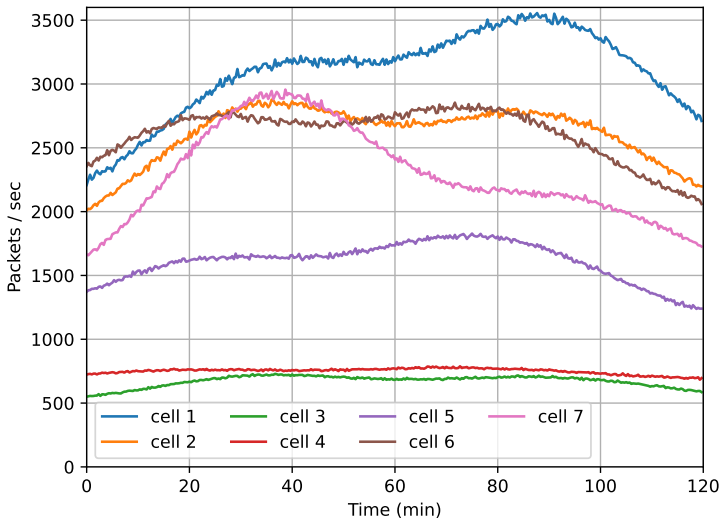


Figure 4.2: Service demand in different cells.

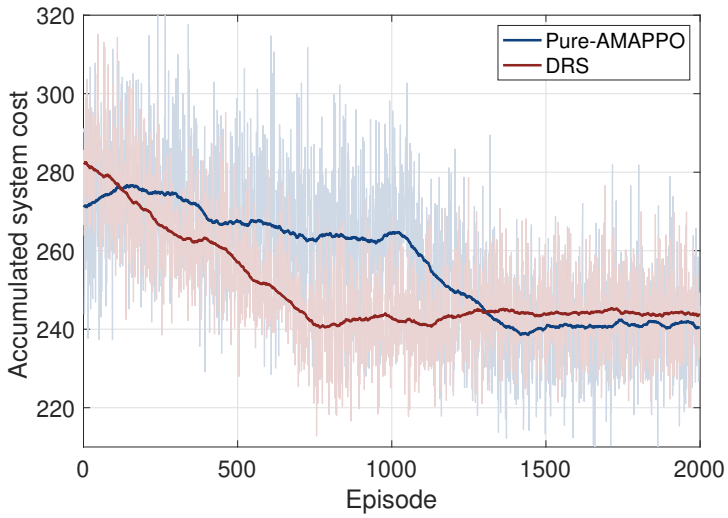
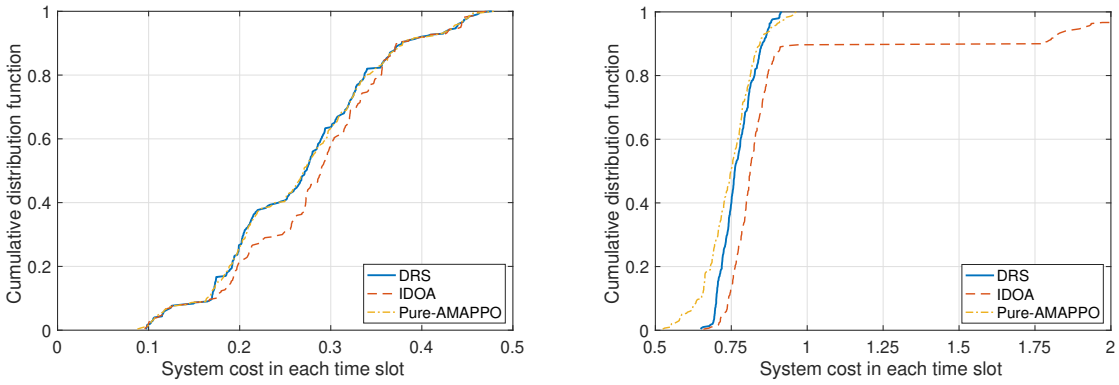


Figure 4.3: Convergence performance.

The overall time of the simulation includes 300 time slots where the duration of each time slot is set to 20 s. The maximum slicing window length is 300 s. The delay and



(a) Same service demand in all cells.

(b) Different service demand in each cell.

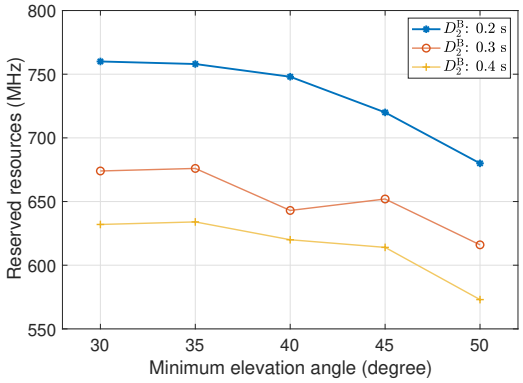
Figure 4.4: CDF of system cost in each time slot.

reliability requirement for delay-sensitive applications is set to 50 ms with $\varepsilon = 0.01$ and the packet size of 0.2 Mb, while the expected delay for delay-tolerant applications is 0.3 s with packet size of 2 Mb. The considered service demand is shown in Fig. 4.2.

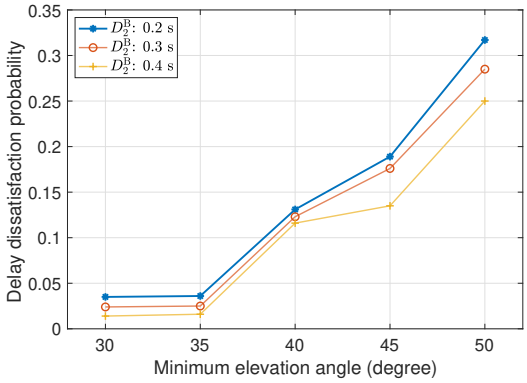
To evaluate the effectiveness of the proposed DRS scheme, we adopt the following two benchmark algorithms: 1) *Pure-AMAPPO*: the resource reservation decisions are directly made by AMAPPO-based algorithm by outputting the ratio of resources reserved for each slice from each access point, adopting the same learning rate of the proposed DRS scheme; and 2) *IDOA*: the resource reservation decisions are obtained from Algorithm 3, where resources from all satellites covering the cell within the slicing window can be reserved.

We first present the convergence performance of the overall system cost in each iteration of the proposed DRS scheme during the training period, compared with the *Pure-AMAPPO* algorithm, as shown in Fig. 4.3. We conducted the simulation with 2000 iterations. It can be seen that the accumulated system cost in each episode of our proposed DRS scheme converges after around 850 episodes, which has a faster convergence speed compared with the *Pure-AMAPPO* which converges after around 1500 episodes. This indicates that our proposed scheme in a hybrid data-model co-driven approach can achieve optimal results with a short training time.

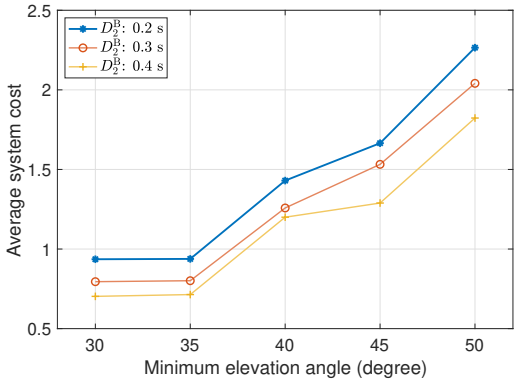
Next, as shown in Fig. 4.4, we compare the time slot-averaged system cost with benchmark algorithms in two scenarios. The first scenario assumes that all cells have identical service demands of cell 3 shown in Fig. 4.2, where the slicing window lengths are the same in every cell. The second scenario considers varying service demands with



(a) Overall reserved resources.



(b) Delay dissatisfaction rate.



(c) Average system cost.

Figure 4.5: Performance under different minimum elevation angles of satellites.

asynchronous resource reservation decisions in each cell. From Fig. 4.4(a), we can see that our proposed DRS scheme outperforms the *IDOA* and has similar performance to the *Pure-AMAPPO* since the reinforcement learning-based satellite selection can effectively reduce the competition among different cells. With the different service demands in each cell, as shown in Fig. 4.4(b), the performance of the DRS scheme can approach the *Pure-AMAPPO* and outperform the *IDOA*, indicating that our proposed scheme can effectively take the temporal and spatial interactions among different cells into account to reach a lower system cost.

Finally, we show the resource consumption performance and delay dissatisfaction probability of our proposed scheme in different expected delays of delay-tolerant services under

different minimum elevation angles in Fig. 4.5. From the figure, we can see that when the elevation angle is small, indicating that more satellites are able to serve users in each cell, a large amount of resources are reserved to ensure a small delay dissatisfaction probability, which enables a small overall system cost with the optimal tradeoff for delay-tolerant applications. With the increment of the elevation angle, the resource decreases and the delay dissatisfaction probability increases correspondingly. Specifically, some unavoidable satellite resource competition from different cells or even a service disruption from satellite networks with a large elevation angle may result in a large delay dissatisfaction probability. Combining these two factors, as shown in Fig. 4.5(c), the overall cost increases for a larger minimum elevation angle and small expected delays of delay-tolerant services results in a high cost due to the stringent requirements.

4.6 Summary

In this chapter, we have investigated resource slicing with cross-cell coordination in STIVN to minimize the system cost in terms of resource usage and delay performance for different services. To cope with the spatiotemporal dynamics of service demand and satellite mobility, which result in asynchronous slicing decisions and resource competition across different cells, we have proposed a DRS scheme by developing a hybrid data-model co-driven approach for resource slicing in STIVN, which can effectively reduce the training time compared with a fully data-driven method. Specifically, an AMAPPO-based algorithm is designed to find the optimal satellite set for each cell, and a distributed optimization-based algorithm is introduced for resource reservation to find the minimum cost in each slicing window.

Chapter 5

Operation-Stage Resource Management for Environment Sensing

In this chapter, we investigate a real-time multi-dimensional resource management problem in STIVN in the operation stage. Specifically, we study the infrastructure-assisted environment sensing in STIVN for CAVs, where satellites and RSUs cooperate to provide CAVs with fresh sensing data. To support satellite- and RSU-assisted environment sensing for CAVs, we formulate a resource scheduling problem in STIVN to satisfy sensing data freshness requirements with efficient resource usage. To deal with the challenges posed by the dynamic network environment as well as stringent data freshness requirements, we propose a cooperative satellite-terrestrial resource scheduling (CSTRS) scheme. CSTRS is a model-data co-driven approach that can adaptively optimize the sensing interval and resource allocation in STIVN. Specifically, benefiting from the multicast feature of the LEO satellite, coalition game and particle swarm optimization-based algorithms are designed to partition CAVs into groups and optimize sensing intervals. Then, a reinforcement learning-based algorithm is developed to make real-time computing and communication resource allocation decisions based on the CAV partition. Simulation results demonstrate that the proposed scheme outperforms benchmark methods in terms of resource usage and reliability performance.

5.1 Background and Motivations

As the number of CAVs on the road increases tremendously, the acquisition of real-time environment data becomes imperative for optimizing driving routes and enhancing the

overall travel experience [114]. Specifically, each CAV needs surrounding environment information within its range of interest (RoI), for improving transportation safety and efficiency. With the help of the built-in processors of CAVs, onboard sensors are able to sense the surrounding environment and process the sensing data to detect objects, such as non-connected vehicles, obstacles, etc. However, considering the limited sensing area as well as the potential blockage from a single CAV view, augmenting the sensing range and mitigating blocked areas are essential. Infrastructure-assisted environment sensing, using sensors deployed on different network infrastructures, can provide large-range traffic information within the RoI of CAVs to increase the trust level of the observation [8]. To maintain up-to-date environment information in CAVs, freshness and reliability of sensing results are crucial to avoid data expiration. However, the freshness of sensing results diminishes over time, underscoring the need to maintain a high freshness level for high safety and optimal CAV control.

Currently, most sensors including cameras and LIDAR sensors are deployed on terrestrial infrastructure such as roadside units (RSUs) to support environment sensing [115]. By deploying edge servers on RSUs, the sensing data, such as images of the covered area, can be processed to generate sensing results. These results are then transmitted to CAVs within the RSUs' service range to update surrounding traffic information. The significant computing and communication capabilities of RSUs for data processing and transmission enable environment sensing with low latency and high reliability, making terrestrial networks the primary choice for serving CAVs. However, RSUs cannot achieve global deployment due to high costs and geographical constraints, especially in rural or remote areas. In this case, the sensing data cannot always maintain a high level of freshness and may become outdated over time.

To enlarge the network coverage to support CAVs, Low Earth Orbit (LEO) satellites offer a promising solution [48, 116]. With their dense deployment, LEO satellites can augment terrestrial networks by providing ubiquitous connectivity and seamless service provision, especially in remote areas. Remote sensing LEO satellites have been studied for many years, where high-resolution satellite images allow the monitoring of road traffic conditions to improve traffic efficiency [117]. With edge computing capabilities equipped on LEO satellites, the sensing data can be processed directly, eliminating the need to transmit large volumes of raw data to ground servers and reducing transmission delay. Nevertheless, satellites move over time and have a higher unit cost of resources compared to terrestrial networks, where a sole satellite network is resource-inefficient in serving CAVs. To take full advantage and complementarity of both satellites and RSUs for environment sensing to satisfy the freshness requirement and optimize resource utilization, cooperative resource scheduling from both satellite and terrestrial networks in STIVN is critical.

However, cooperative resource scheduling for both satellite and terrestrial networks still encounters several challenges in supporting environment sensing. First, to maintain the freshness of sensing results in each CAV, two factors need to be jointly considered: sensing interval and infrastructure-related delays in data processing and result transmission, which leads to a highly complex resource scheduling strategy in a dynamic network environment. Second, the mobility of LEO satellites and CAVs causes the distance between them and CAVs to vary over time, introducing fluctuations in propagation delay, communication performance, and network accessibility over time, resulting in variable resource demands from both satellite and terrestrial networks. Third, LEO satellites and RSUs in STIVN have heterogeneous network features in terms of coverage areas and content delivery modes. Due to the distinct coverage areas of satellites and RSUs, connection durations for CAVs differ between satellite and terrestrial components. Meanwhile, each network component needs to schedule both computing and communication resources with varying resource costs for environment sensing. Since CAVs can be supported by satellites alone or by both satellite and terrestrial networks, cooperative resource scheduling between these two distinct networks becomes complicated. It requires managing varying connectivity patterns and resource differentiations, involving multiple decisions to ensure seamless service and data freshness for CAVs.

To address the abovementioned challenges, in this chapter, we propose a cooperative satellite-terrestrial resource scheduling (CSTRS) scheme. CSTRS scheme aims to cooperatively allocate communication and computing resources as well as determine sensing intervals in STIVN, ensuring stringent data freshness requirements with efficient resource utilization. Specifically, a hybrid data-model co-driven approach is developed to find the optimal solutions in two steps for simplifying the decision-making. Leveraging the multicast feature of LEO satellites to serve a group of CAVs simultaneously with less resource consumption, we design model-based sensing interval determination and CAV partitioning methods in a large timescale for reducing the computing and communication resource cost and the resource allocation complexity. Based on the CAV grouping and sensing interval, a reinforcement learning-based algorithm is then developed for the real-time computing and communication resource allocation decisions for different CAV groups to cope with time-varying network dynamics. The main contributions of this work are summarized as follows:

- We investigate the resource scheduling problem in STIVN to support infrastructure-assisted environment sensing for CAVs, where both LEO satellites in satellite networks and RSUs in terrestrial networks can provide network services for CAVs. Specifically, we formulate the cooperative resource scheduling problem for allocating computing and

communication resources as well as determining sensing interval and CAV partitioning to satisfy the data freshness requirements with efficient resource consumption.

- We propose the CSTRS scheme to solve the mixed-integer sequential decision-making resource scheduling problem. In this scheme, the original problem is decoupled into two subproblems to reduce the complexity. Specifically, CAV partition and sensing interval decisions are adjusted in a large timescale, while resource allocation for each service group is determined in a small timescale.
- We design a particle swarm optimization (PSO)-based algorithm to optimize the long-term sensing interval and a coalition formation game-based approach to determine CAV partition, with reduced resource cost and decision-making complexity. The real-time communication and computing resource allocation subproblem is solved by the proposed proximal policy optimization (PPO)-based reinforcement learning algorithm to cope with dynamic environments.

The remainder of this chapter is organized as follows. The considered network scenario and system models are given in Section 5.2. The problem formulation and decoupling are presented in Section 5.3. Then, a two-step CSTRS scheme is proposed, where a PSO-based optimization algorithm and a coalition formation game are designed for determining sensing interval and CAV partitioning for long-term decisions in Section 5.4, and a reinforcement learning-based algorithm for real-time computing and communication resource allocation is provided in Section 5.5. Performance evaluation is given in Section 5.6 to demonstrate the performance of the proposed scheme, followed by the conclusions in Section 5.7.

5.2 Network Scenario and System Model

In this section, we first introduce the STIVN scenario for infrastructure-assisted environment sensing and then describe the system models. Some useful notations are listed in Table 5.1 for easy reference.

5.2.1 Scenario Description

The infrastructure-assisted environment sensing for CAVs in the STIVN is shown in Fig. 5.1. Specifically, terrestrial RSUs can support environment sensing for CAVs within their coverage areas. However, due to high deployment and maintenance costs or geographical

Table 5.1: Summary of Notations

Notation	**Definition**
$\mathbf{p}^{\mathrm{S}}, \mathbf{p}_n^{\mathrm{T}}, \mathbf{p}_{\mathbf{s}}^{\mathrm{SSG}}$	Coordination of the LEO satellite, RSU n, and the center of SSG $\mathbf{s}$
$a_u^{\mathrm{T}}(t)$	RSU accessibility for CAV u at time slot t
$\Omega^{\mathrm{S}}(t), \Omega^{\mathrm{T}}(t)$	Set of SSGs and TSGs at time slot t
$I_{\mathbf{g}}^{\mathrm{T}}(t)$, $I_{\mathbf{s}}^{\mathrm{S}}(t)$	Sensing intervals for SSG $\mathbf{s}$ and TSG $\mathbf{g}$
$\nu_{\mathbf{s}}^{\mathrm{S}}(t)$, $\beta_{\mathbf{s}}^{\mathrm{S}}(t)$	Sensing data size and sensing result data size for SSG $\mathbf{s}$
$Q_{\mathbf{s}}^{\mathrm{S}}(t)$, $W_{\mathbf{s}}^{\mathrm{S}}(t)$, $Z_{\mathbf{s}}^{\mathrm{S}}(t)$	Remaining processing data size, transmission data size, and propagation time for SSG $\mathbf{s}$
$c_{\mathbf{s}}^{\mathrm{S}}(t), b_{\mathbf{s}}^{\mathrm{S}}(t)$	Computing and communication resources allocated for SSG $\mathbf{s}$ at time slot t
χ_1^{S}, χ_2^{S}, χ_1^{T}, χ_2^{T}	Unit cost of computing and communication resource in the satellite and RSUs
T_u^{S}, T_u^{T}, $\hat{\psi}_u^{\mathrm{S}}$	Sensing interval from the satellite and RSUs, and offset set for CAV u
$q_u(t), q_{\mathbf{s}}^{\max}(t)$	Overall AoI for CAV u and maximum AoI value in SSG $\mathbf{s}$
$H_{\mathbf{s}}^{\max}(t)$	Maximal virtual reliability queue value of CAVs in SSG $\mathbf{s}$

constraints, RSUs can hardly guarantee full coverage, especially in remote areas. Therefore, LEO satellites can be utilized to complement terrestrial RSUs in supporting environment sensing. A central controller on the ground manages all RSUs and the LEO satellite covering the area. Here, we consider LEO satellites with steerable beams to cover the service area in a fixed time [118]. Consider a time-slotted system where each time slot $t \in \mathcal{T} = \{1, 2, ..., T\}$ has the same duration, denoted by τ, and T is the duration of LEO satellite service for a given area following the minimum elevation angle requirement. Let $(r^{\mathrm{S}}, \alpha, \omega(t), \rho(t))$ illustrate the time-varying position of the LEO satellite at time slot t, where r^{S} is the LEO satellite orbit radius, α is the orbital inclination angle, $\omega(t)$ is the longitude of the ascending node of the orbit, and $\rho(t)$ is the argument of periapsis, defining the angle measured from the ascending node of the orbit to the satellite. The Cartesian coordination of the satellite at time slot t can therefore be transformed and given by $\mathbf{p}^{\mathrm{S}}(t) = (x^{\mathrm{S}}(t), y^{\mathrm{S}}(t), z^{\mathrm{S}}(t))$ according to [48]. Let $\mathcal{N} = \{1, 2, ..., N\}$ denote the set of RSUs serving the target area, where each RSU has a fixed coverage radius of r^{T}. The Cartesian coordinate of RSU n is $\mathbf{p}_n^{\mathrm{T}} = (x_n^{\mathrm{T}}, y_n^{\mathrm{T}}, z_n^{\mathrm{T}})$. Let $\mathcal{U} = \{1, 2, ..., U\}$ denote the set of CAVs, where each CAV has the

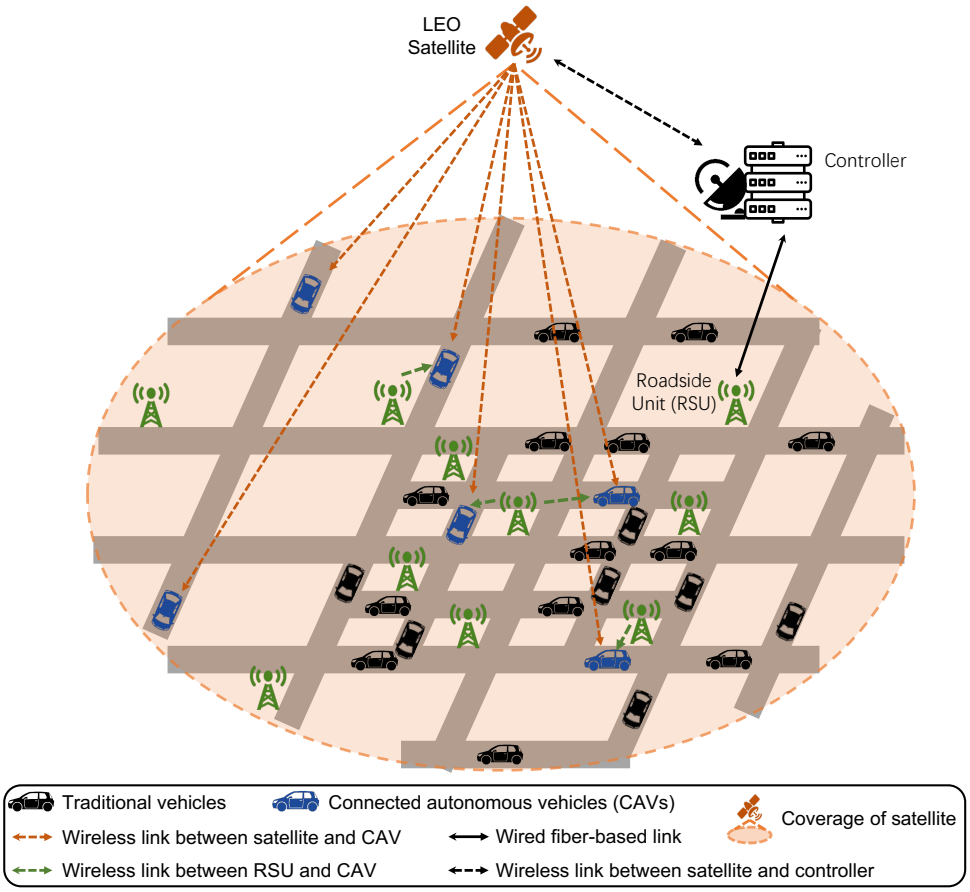


Figure 5.1: Considered environment sensing scenario in STIVN.

coordinate $\mathbf{p}_u^U(t) = (x_u^U(t), y_u^U(t), z_u^U(t))$ at time slot t. Considering the mobility of CAVs, the accessibility to RSUs may change over time. Let $a_u^{\mathrm{T}}(t)$ represent the RSU accessibility for CAV u at time slot t. If $a_u^{\mathrm{T}}(t) = 1$, CAV u can access at least one RSU at time slot t, while if $a_u^{\mathrm{T}}(t) = 0$, CAV u cannot access to any RSU.

Consider that both LEO satellites and RSUs have the capability of data processing with the deployment of edge servers. For infrastructure-assisted environment sensing, the LEO satellite and RSUs need to periodically sense the RoI of CAVs, process the sensing data, and deliver the sensing results to CAVs. Processing and transmitting the sensing data related to the RoI of each individual CAV may lead to unnecessarily excessive consumption of computing and communication resources due to the potential overlaps among different CAVs' RoIs. To mitigate this issue, we leverage the multicast feature of

LEO satellites to serve a bunch of CAVs simultaneously to enhance computing resource and bandwidth utilization. Specifically, we establish multiple satellite service groups (SSGs) for environment sensing from the satellite, where each group contains a subset of CAVs. Let $\Omega^{\mathrm{S}}(t) = \{\mathbf{s}_0(t), \mathbf{s}_1(t), ..., \mathbf{s}_m(t)\}$ denote the SSGs, where $\mathbf{s}_i(t), i \neq 0$ is the i-th SSGs served by the LEO satellite at time slot t and $\mathbf{s}_0(t)$ is the CAV group not served by the satellite. On the other hand, due to the small coverage of RSUs, we consider that each RSU serves one terrestrial service group (TSG) consisting of all CAVs within its coverage. Specifically, each RSU periodically senses the environment within its coverage, processes the sensing data, and transmits the result to all covering CAVs. In this case, TSGs can be denoted by $\Omega^{\mathrm{T}}(t) = \{\mathbf{g}_1(t), ..., \mathbf{g}_n(t)\}$, where $\mathbf{g}_i(t)$ is the set of CAVs in the i-th TSG served by RSU i. Considering the mobility of CAVs, CAVs in each TSG may change over time and SSGs are adjusted in a constant time interval, consisting of k time slots.

To satisfy the data freshness requirement, determining a proper sensing interval is essential for each service group (SG). Basically, a small sensing interval enables tracking real-time vehicle behavior but leads to stringent processing and transmission delay requirements, consuming a large amount of resources. On the other hand, a large sensing interval may violate the freshness requirements. Considering that sensing from the satellite and RSUs can both affect the freshness of the sensing results, the sensing interval from different sources should be jointly determined to minimize resource consumption. Let $I^{\mathrm{S}}_{\mathbf{s}}(t)$ and $I^{\mathrm{T}}_{\mathbf{g}}(t)$ denote sensing intervals for SSG $\mathbf{s}$ and TSG $\mathbf{g}$ at time slot t, respectively.

5.2.2 Age of Information Model for Environment Sensing

As up-to-date sensing results are crucial for CAVs to make timely and accurate decisions in dynamic environments, it is important to satisfy the freshness and reliability requirements of environment sensing. Age of information (AoI), which is a metric quantifying the time elapsed since the most recent sensing data was generated, can be utilized to describe the freshness and reliability of the sensing results received by CAVs [119]. Specifically, each sensing data is stamped with the time of the generation. The AoI of the sensing result for CAV u is defined as the time elapsed since the latest received sensing result was generated from the LEO satellite or RSU. The evolution of AoI for the sensing result of CAV u can be given by

$$q^{\mathbf{x}}_u(t) = \begin{cases} q^{\mathbf{x}}_u(t-1) + 1, & \text{if no packet received,} \\ t - \max_i\{t^{\mathbf{x}}_u(i) | t'^{\mathbf{x}}_u(i) \leq t\}, & \text{otherwise,} \end{cases} \tag{5.1}$$

where $t^{\mathbf{x}}_u(i)$ and $t'^{\mathbf{x}}_u(i)$ are the generation time and the delivered time of the i-th sensing data and result of CAV u, and $\mathbf{x} \in \{\mathrm{S}, \mathrm{T}\}$ represents that the sensing data is generated

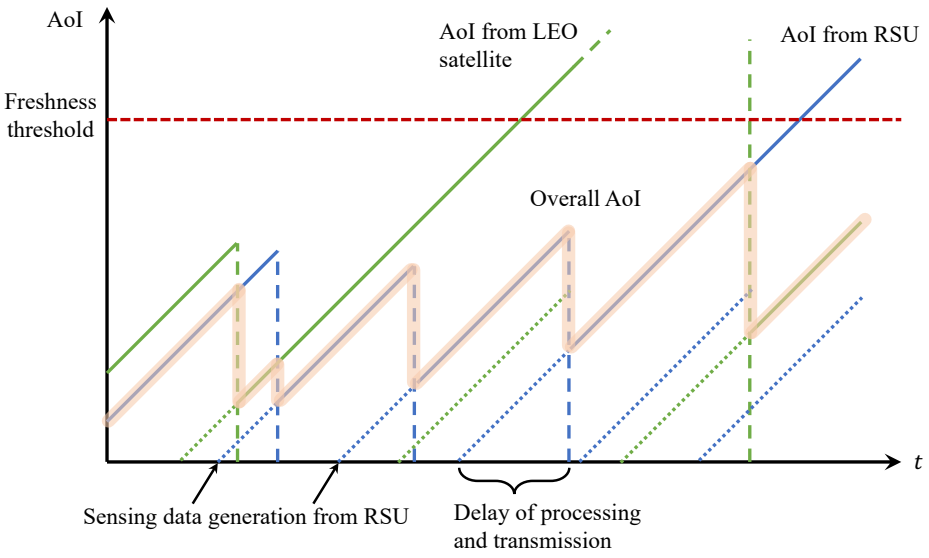


Figure 5.2: AoI illustration of CAV.

from satellite or RSUs. When CAV u receives new sensing results, the AoI is renewed to describe the freshness information of the latest data; otherwise, it increases linearly over time slots.

The AoI should be bounded to guarantee the freshness of the environment information. Moreover, to guarantee the reliability of environment sensing, the probability that the sensing data is correctly received within the freshness threshold T^{th} should be larger than $1-\varepsilon$ [120]. Since CAVs are able to receive sensing results from both RSUs and the LEO satellite, the overall AoI of a CAV depicts the AoI of the data from either source, which is determined by the minimum AoI from terrestrial or satellite networks, as shown in Fig. 5.2. Note that if a fresher sensing result is received by the CAV, the earlier generated data from either RSUs or LEO satellite for this CAV, which has not been received, will be discarded. In addition, if two sensing results are received at the same time, the formerly generated packet will be discarded. The overall AoI for CAV u can be given as

$$q_u(t) = \min\{q_u^{\text{S}}(t), q_u^{\text{T}}(t)\}. \tag{5.2}$$

From Fig. 5.2, we can find that both sensing interval and delay of data processing and transmission can affect the AoI value. Therefore, we need to jointly optimize the sensing interval for SGs and the allocation of computing and communication resources from both satellite and terrestrial networks to satisfy the freshness requirements of environment

sensing.

5.2.3 Computing and Communication Model

When the sensing data is generated for each SG, it needs to be processed and then the result needs to be transmitted to CAVs. For each SSG, the initial sensing data size to be processed is denoted by $\nu_{\mathbf{s}}^{\mathrm{S}}(t)$, which varies for different sensing areas. Let $K(t)+\Psi_{\mathbf{s}}^{\mathrm{S}}(t)+iI_{\mathbf{s}}^{\mathrm{S}}(t)$ represent the generation time of the i-th sensing data, where $K(t)=\max\{jk \,|\, jk \le t, \forall j\}$ is the initial time slot of the current interval for adjusting SSGs, and $\Psi_{\mathbf{s}}^{\mathrm{S}}(t)$ is a delayed time of the first sensing for SSG $\mathbf{s}$ from $K(t)$. The satellite computing resources allocated for SSG $\mathbf{s}$ at time slot t is denoted by $c_{\mathbf{s}}^{\mathrm{S}}(t)$. Let $Q_{\mathbf{s}}^{\mathrm{S}}(t)$ indicate the size of remaining data for processing at time slot t for SSG $\mathbf{s}$, given by

$$Q_{\mathbf{s}}^{\mathrm{S}}(t)=\begin{cases}\nu_{\mathbf{s}}^{\mathrm{S}}(t), & \text{if } t=K(t)+iI_{\mathbf{s}}^{\mathrm{S}}(t)+\Psi_{\mathbf{s}}^{\mathrm{S}}(t),\forall i\\ \max\left\{Q_{\mathbf{s}}^{\mathrm{S}}(t-1)-\Delta\nu_{\mathbf{s}}(t-1),0\right\}, & \text{otherwise.}\end{cases} \tag{5.3}$$

The amount of data processed at time slot t for SSG $\mathbf{s}$ is determined by $\Delta\nu_{\mathbf{s}}(t)=\min\{Q_{\mathbf{s}}^{\mathrm{S}}(t),\tau f^{\mathrm{S}}c_{\mathbf{s}}(t)/\gamma\}$, where f^{S} represents the CPU frequency unit (in CPU cycles per second) and γ is the computation workload (in CPU cycles per bit). Specifically, the following computing resource allocation constraints should be satisfied:

$$\sum_{\mathbf{s}\in\Omega^{\mathrm{S}}(t)} c_{\mathbf{s}}^{\mathrm{S}}(t)\le c_{\max}^{\mathrm{S}},\quad \forall t, \tag{5.4}$$

$$c_{\mathbf{s}}^{\mathrm{S}}(t)=c_{\mathbf{s}}^{\mathrm{S}}(t-1),\quad \text{if } 0<Q_{\mathbf{s}}^{\mathrm{S}}(t)<Q_{\mathbf{s}}^{\mathrm{S}}(t-1), \tag{5.5}$$

where (5.4) indicates the constraint on the maximum computing resources in the LEO satellite, and (5.5) represents that the allocated computing resources remain unchanged during the processing.

After processing the sensing data, the sensing results need to be multicast to CAVs in each SSG, where the data size of the sensing results for SSG $\mathbf{s}$ is denoted by $\beta_s^{\mathrm{S}}(t)$. Let $W_{\mathbf{s}}^{\mathrm{S}}(t)$ denote the size of remaining transmission data for SSG $\mathbf{s}$ at time slot t, as expressed by

$$W_{\mathbf{s}}^{\mathrm{S}}(t)=\begin{cases}\beta_{\mathbf{s}}^{\mathrm{S}}(t), & \text{if } Q_{\mathbf{s}}^{\mathrm{S}}(t-1)>Q_s^{\mathrm{S}}(t)=0,\\ \max\left\{W_{\mathbf{s}}^{\mathrm{S}}(t-1)-\Delta\beta_{\mathbf{s}}(t-1),0\right\}, & \text{otherwise.}\end{cases} \tag{5.6}$$

In time slot t, the amount of transmitted data is $\Delta\beta_{\mathbf{s}}(t)=\min\{W_s^{\mathrm{S}}(t),\tau R_{\mathbf{s}}^{\mathrm{S}}(t)\}$, where $R_{\mathbf{s}}^{\mathrm{S}}(t)=b_{\mathbf{s}}^{\mathrm{S}}(t)\log_2(1+\frac{P^{\mathrm{S}}h_{\mathbf{s}}^{\mathrm{S}}(t)}{\sigma^2})$ is the data transmission rate. Here, $b_{\mathbf{s}}^{\mathrm{S}}(t)$ is the bandwidth

resources allocated to SSG $\mathbf{s}$ at time slot t, P^{S} denotes the transmit power of the LEO satellite, σ^2 is the noise power, and $h_{\mathbf{s}}^{\mathrm{S}}(t)$ is the channel gain. The communication resource allocation should satisfy

$$\sum_{\mathbf{s}\in\Omega^{\mathrm{S}}(t)} b_{\mathbf{s}}^{\mathrm{S}}(t) \leq b_{\max}^{\mathrm{S}}, \quad \forall t, \tag{5.7}$$

$$b_{\mathbf{s}}^{\mathrm{S}}(t) = b_{\mathbf{s}}^{\mathrm{S}}(t-1), \quad \text{if } 0 < W_{\mathbf{s}}^{\mathrm{S}}(t) < W_{\mathbf{s}}^{\mathrm{S}}(t-1), \tag{5.8}$$

where (5.7) constrains the available communication resources of the LEO satellite and (5.8) represents that the allocated communication resources remain unchanged during the transmission.

Considering the high LEO satellite altitude, the propagation delay cannot be neglected. At time slot t, the remaining propagation time is given by

$$Z_{\mathbf{s}}^{\mathrm{S}}(t) = \begin{cases} T_{\mathbf{s}}^{\mathrm{prop}}(t), & \text{if } W_{\mathbf{s}}^{\mathrm{S}}(t-1) > W_{\mathbf{s}}^{\mathrm{S}}(t) = 0, \\ \max\left\{Z_{\mathbf{s}}^{\mathrm{S}}(t-1) - \tau, 0\right\}, \text{otherwise}, \end{cases} \tag{5.9}$$

where $T_{\mathbf{s}}^{\mathrm{prop}}(t) = \sqrt{|\mathbf{p}^{\mathrm{S}}(t) - \mathbf{p}_{\mathbf{s}}^{\mathrm{SSG}}(t)|^2}/c$ is the propagation delay between SSG $\mathbf{s}$ and the satellite, with the coordination for the center of SSG $\mathbf{s}$ denoted by $\mathbf{p}_{\mathbf{s}}^{\mathrm{SSG}}(t)$ and the light speed being c.

For the TSG served by each RSU, the processing and transmission models are similar. However, the allocated computing and communication resources c^{T} and b^{T} to serve TSG $\mathbf{g}$ are fixed to process and transmit data, since the RSU resources are not shared by other TSGs, and no propagation delay is considered.

5.3 Problem Formulation and Decomposition

In this section, we formulate the resource scheduling problem to support environment sensing for CAVs, aiming to minimize the resource cost while satisfying freshness and reliability requirements in the presence of network dynamics.

Let $\Omega^{\mathrm{S}} = \{\Omega^{\mathrm{S}}(t), \forall t \in \mathcal{T}\}$ denote SSGs, and $\mathcal{I}^{\mathrm{S}} = \{I_{\mathbf{s}}^{\mathrm{T}}(t), \forall \mathbf{s} \in \Omega^{\mathrm{S}}(t), \forall t\}$ and $\mathcal{I}^{\mathrm{T}} = \{I_{\mathbf{g}}^{\mathrm{T}}(t), \forall \mathbf{g} \in \Omega^{\mathrm{T}}(t), \forall t\}$ denote sensing intervals of all SSGs and TSGs in T time slots, respectively. The resource cost for SSG $\mathbf{s}$ at time slot t from the LEO satellite is given by $\Upsilon_{\mathbf{s}}^{\mathrm{S}}(t) = \chi_1^{\mathrm{S}} c_{\mathbf{s}}^{\mathrm{S}}(t) + \chi_2^{\mathrm{S}} b_{\mathbf{s}}^{\mathrm{S}}(t)$, where $\chi_1^{\mathrm{S}}, \chi_2^{\mathrm{S}}$ are the unit cost of computing and communication resources from the satellite. Similarly, for TSG $\mathbf{g}$ in terrestrial networks, the resource consumption at time slot t is given by $\Upsilon_{\mathbf{g}}^{\mathrm{T}}(t) = \chi_1^{\mathrm{T}} c_{\mathbf{g}}^{\mathrm{T}}(t) + \chi_2^{\mathrm{T}} b_{\mathbf{g}}^{\mathrm{T}}(t)$, where χ_1^{T} and χ_2^{T} are the

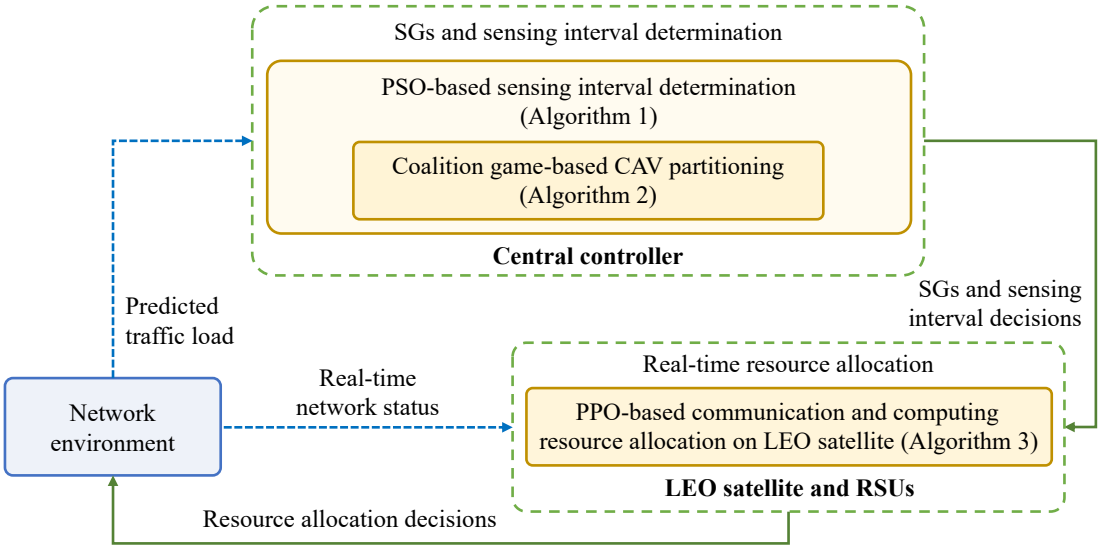


Figure 5.3: Working diagram of the proposed CSTRS scheme for environment sensing.

unit cost of computing and communication resources from the RSU. Let $\mathcal{C}^{\mathrm{S}} = \{c_{\mathbf{s}}^{\mathrm{S}}(t), \forall \mathbf{s} \in \Omega^{\mathrm{S}}, \forall t\}$ and $\mathcal{B}^{\mathrm{S}} = \{b_{\mathbf{s}}^{\mathrm{S}}(t), \forall \mathbf{s} \in \Omega^{\mathrm{S}}, \forall t\}$ denote the computing and communication resource allocation decisions for all time slots, respectively. Note that the unit costs of computing and communication resources vary across the heterogeneous satellite and terrestrial networks, necessitating a cooperative optimization of both terrestrial and satellite resource costs for environment sensing in STIVN. In this case, the goal is to minimize overall resource costs while guaranteeing the freshness and reliability of sensing data for CAVs, which can be formulated as

$$\mathbf{P1} \min_{\substack{\Omega^{\mathrm{S}}, \mathcal{C}^{\mathrm{S}}, \\ \mathcal{B}^{\mathrm{S}}, \mathcal{I}^{\mathrm{S}}, \mathcal{I}^{\mathrm{T}}}} \quad \sum_{t=1}^{\mathrm{T}} \left(\sum_{\mathbf{s} \in \Omega^{\mathrm{S}}} \Upsilon_{\mathbf{s}}^{\mathrm{S}}(t) + \sum_{\mathbf{g} \in \Omega^{\mathrm{T}}} \Upsilon_{\mathbf{g}}^{\mathrm{T}}(t) \right) \tag{5.10}$$

$$\text{s.t.} \quad \frac{\sum_{t'=1}^{t} \mathbb{1}_{q_u(t') \geq T^{\mathrm{th}}}}{t} < \varepsilon, \forall u \in \mathcal{U}, \forall t \in \mathcal{T} \tag{5.10a}$$

$$\mathbf{s}_i \cap \mathbf{s}_j = \emptyset, \forall \mathbf{s}_i, \mathbf{s}_j \in \Omega^{\mathrm{S}}(t), i \neq j, \tag{5.10b}$$

$$\cup_{\mathbf{s} \in \Omega^{\mathrm{S}}(t)} \mathbf{s} = \mathcal{U}, \tag{5.10c}$$

$$I_{\mathbf{s}}^{\mathrm{T}}(t), I_{\mathbf{g}}^{\mathrm{T}}(t) \in \{l\kappa | l \in \mathbb{N}\}, \tag{5.10d}$$

$$|\Omega^{\mathrm{S}}(t)| \leq M, \tag{5.10e}$$

$$(5.4), (5.5), (5.7), (5.8).$$

In problem **P1**, constraint (5.10a) represents the freshness and reliability requirement that the probability of task failure should be less than the threshold ε over any accumulated period of t time slots. Constraints (5.10b) and (5.10c) constrain the SSGs served by the LEO satellite, where CAVs cannot be simultaneously served in multiple SSGs and every CAV in the target area should be included in one SSG. Constraint (5.10d) constrains the available sensing interval for each SG, where κ is the minimum sensing interval limited by sensing devices. Considering the limited capacity of the LEO satellite with the edge server threads of M, indicating the maximum concurrent processing of sensing data [121], constraint (5.10e) regulates the maximum number of SSGs.

Solving **P1** encounters the following challenges. Determining the resource allocation in each time slot is a sequential decision problem, where the decision at a time slot will affect the subsequent decisions. Moreover, determining long-term SSGs and sensing interval decisions, along with the short-term resource allocation decisions, is a mixed-integer optimization problem, where two types of variables are mutually dependent. In this case, directly applying model-free techniques can exacerbate model convergence difficulties, especially considering the highly dynamic network environment and the high-dimensional action space. To address these challenges, we propose a cooperative satellite-terrestrial resource scheduling (CSTRS) scheme, which employs a data-model co-driven approach to solve the problem.

Considering the characteristics of the action, adjusting SSGs and the sensing interval is a long-term action for each time interval, and the resource allocation decisions are short-term actions determined in each time slot. Therefore, we decouple the original problem into two subproblems: 1) CAV partitioning and sensing interval decisions determined by the ground controller, and 2) real-time computing and communication resource allocation for SGs in the satellite and RSUs. Figure 5.3 shows the working diagram of the proposed CSTRS scheme for environment sensing in STIVN in two steps.

In the first subproblem, we aim to find the optimal CAV partitioning and sensing intervals for each SG to minimize resource consumption in each time interval consisting of k time slots. Specifically, considering the long-term performance in each time interval, the average resource cost of each SG can be estimated through predicted processing and transmission workload according to the predicted CAV trajectories[1]. Therefore, the first

[1]Note that the prediction method is not the primary focus of this work and several algorithms have been studied for this purpose, such as recurrent neural networks (RNN) [93], convolutional neural networks (CNN), and long-short term memory (LSTM) [122].

subproblem can be given by

$$\mathbf{P2} \quad \min_{\Omega^{\mathrm{S}},\mathbf{I}^{\mathrm{S}},\mathbf{I}^{\mathrm{T}}} \quad \sum_{\mathbf{s}\in\Omega^{\mathrm{S}}} C_{\mathbf{s}}^{\mathrm{S}} + \sum_{\mathbf{g}\in\Omega^{\mathrm{T}}} C_{\mathbf{g}}^{\mathrm{T}} \tag{5.11}$$
$$\text{s.t.} \quad (5.10\text{b}), (5.10\text{c}), (5.10\text{d}), (5.10\text{e}).$$

Specifically, $C_{\mathbf{s}}^{\mathrm{S}}$ is the average resource cost for SSG $\mathbf{s}$ in a time interval, calculated by the average communication and computing resource usage with their unit cost, given by

$$C_{\mathbf{s}}^{\mathrm{S}} = \chi_1^{\mathrm{S}} \frac{k\tau\bar{\nu}_{\mathbf{s}}^{\mathrm{S}}\gamma}{I_{\mathbf{s}}^{\mathrm{S}} f^{\mathrm{S}}} + \chi_2^{\mathrm{S}} \frac{k\tau\bar{\beta}_{\mathbf{s}}^{\mathrm{S}}}{I_{\mathbf{s}}^{\mathrm{S}} \bar{R}_{\mathbf{s}}^{\mathrm{S}}}, \tag{5.12}$$

where $I_{\mathbf{s}}^{\mathrm{S}}$ is the sensing interval for SSG $\mathbf{s}$ and $\bar{R}_{\mathbf{s}}^{\mathrm{S}}$ is the average transmission rate of unit spectrum bandwidth within the time interval. Here, $\bar{\nu}_{\mathbf{s}}^{\mathrm{S}}$ and $\bar{\beta}_{\mathbf{s}}^{\mathrm{S}}$ denote the predicted processing and transmission workloads for each sensing of SSG $\mathbf{s}$ in the time interval, respectively. Similarly, the average resource cost for TSG $\mathbf{g}$ in a time interval can be represented by

$$C_{\mathbf{g}}^{\mathrm{T}} = \chi_1^{\mathrm{T}} \frac{k\tau\bar{\nu}_{\mathbf{g}}^{\mathrm{T}}\gamma}{I_{\mathbf{g}}^{\mathrm{T}} f^{\mathrm{T}}} + \chi_2^{\mathrm{T}} \frac{k\tau\bar{\beta}_{\mathbf{g}}^{\mathrm{T}}}{I_{\mathbf{g}}^{\mathrm{T}} \bar{R}_{\mathbf{g}}^{\mathrm{T}}}. \tag{5.13}$$

Then, given the CAV partitioning and sensing intervals, in the second subproblem, we aim to find the optimal sequential decision of resource allocation to minimize the overall resource cost based on the real-time network state, given as

$$\mathbf{P3} \quad \min_{\mathcal{C}^{\mathrm{S}},\mathcal{B}^{\mathrm{S}}} \quad \sum_{t=1}^{\mathrm{T}} \left(\sum_{\mathbf{s}\in\Omega^{\mathrm{S}}(t)} \Upsilon_{\mathbf{s}}^{\mathrm{S}}(t) + \sum_{\mathbf{g}\in\Omega^{\mathrm{T}}(t)} \Upsilon_{\mathbf{g}}^{\mathrm{T}}(t) \right) \tag{5.14}$$
$$\text{s.t.} \quad (5.4), (5.5), (5.7), (5.8), (5.10\text{a}).$$

To solve these two subproblems, the proposed CSTRS scheme comprises two components: a CAV partitioning and sensing interval determination algorithm presented in Section 5.4, and a real-time resource allocation algorithm, detailed in Section 5.5.

5.4 CAV Partitioning and Sensing Interval Determination

To find the optimal SSGs and the sensing interval of each group, in this section, we design a two-step algorithm to solve subproblem **P2**. Specifically, the feasible sensing interval for

each CAV is first derived according to the freshness requirement. Then, to achieve the minimum estimated average resource cost in each time interval, we investigate a particle swarm optimization (PSO)-based method to find the optimal sensing intervals for CAVs and determine the SSGs by formulating a coalition formation game in each time interval.

5.4.1 PSO-based Sensing Interval Determination

For each CAV, let $T_u^{\mathrm{S}}(t)$ and $T_u^{\mathrm{T}}(t)$ denote the sensing interval from the LEO satellite and connected RSUs for CAV u at time slot t, respectively. Specifically, $T_u^{\mathrm{S}}(t)$ and $T_u^{\mathrm{T}}(t)$ mod $\kappa = 0$ follow the minimum interval limited by the sensing devices. For simplicity, we omit the notation t in this section, as decisions are adjusted in each time interval consisting of k time slots. To guarantee the sensing data freshness requirement for each CAV, some preliminary conditions on the sensing intervals should be satisfied according to the maximum allowable AoI requirement.

Proposition 1 (Feasible sensing interval pair for individual CAVs)**.** The sensing interval pair for CAV u is denoted by $(T_u^{\mathrm{S}}, T_u^{\mathrm{T}})$, with an offset of ψ_u^{S}. Here, the offset is the delayed time of the initial satellite sensing from the initial RSU sensing in a time interval, where the initial RSU sensing time is at the beginning of a time interval. To satisfy the freshness requirement of environment sensing, the sensing interval pair of each CAV must guarantee the following conditions:

- If CAV u cannot connect to RSUs either partially or completely during the time interval, i.e., $a_u^{\mathrm{T}}(t) = 0, \exists t$, the sensing interval from the satellite should guarantee $T_u^{\mathrm{S}} < T^{\mathrm{th}}$, with $\hat{\psi}_u^{\mathrm{S}} = [0, T_u^{\mathrm{S}})$;
- If CAV u can always connect to both the LEO satellite and RSUs during the time interval, the sensing interval pair should meet one of the following conditions:
  - $T_u^{\mathrm{S}} < T^{\mathrm{th}}$, with $\psi_u^{\mathrm{S}} \in \hat{\psi}_u^{\mathrm{S}} = [0, T_u^{\mathrm{S}})$;
  - $T_u^{\mathrm{T}} < T^{\mathrm{th}}$, with $\psi_u^{\mathrm{S}} \in \hat{\psi}_u^{\mathrm{S}} = [0, T_u^{\mathrm{T}})$;
  - $T_u^{\mathrm{S}} = T_u^{\mathrm{T}} \in [T^{\mathrm{th}}, 2T^{\mathrm{th}})$, with $\psi_u^{\mathrm{S}} \in \hat{\psi}_u^{\mathrm{S}} = \{l\kappa \,|\, l\kappa \in (T_u^{\mathrm{T}} - T^{\mathrm{th}}, T^{\mathrm{th}}), \forall l \in \mathbb{N}\}$,

where $\hat{\psi}_u^{\mathrm{S}}$ is the set of feasible offset for CAV u.

Proof. The proof is given in Appendix A. ■

With the feasible value ranges for individual CAVs' sensing interval pairs and satellite sensing time offset given in **Proposition 1**, we can then determine the sensing intervals of all SGs to achieve the optimum of problem **P2**. Note that determining the sensing interval for each SG is a combinatorial optimization problem and finding a globally optimal solution is challenging [123]. To address this issue, we employ PSO, an evolutionary heuristic mechanism, to solve problem **P2**. In this approach, a set of particles form a particle swarm. Each particle's position represents a solution for CAVs' sensing interval pairs, denoted by $\Lambda = [T_u^{\mathrm{S}}, T_u^{\mathrm{T}}]^{1\times|\mathcal{U}|}$. In each iteration, particles adjust their positions and speeds within the solution space to discover the corresponding values from the current positions [123]. After multiple iterations, the positions of all particles can converge to the same position indicating the selected sensing interval pair for all CAVs [124].

Denote the set of particles and the position of particle x in y-th iteration by $\mathcal{X}$ and $\Lambda_x^{(y)}$, respectively. Let $\hat{\Lambda}_x$ and Λ^* denote the best position of particle x and the best position among all particles' positions up to the latest iteration, respectively, and $\hat{\Xi}_x$ and Ξ^* are the corresponding values calculated based on Eq. (5.11) at those positions. The speed of particle $x \in \mathcal{X}$ in the y-th iteration, denoted by $p_x^{(y)}$, is updated as

$$p_x^{(y)} = \sigma p_x^{(y-1)} + \sigma_1 \left(\hat{\Lambda}_x - \Lambda_x^{(y-1)}\right) + \sigma_2 \left(\Lambda^* - \Lambda_x^{(y-1)}\right) \tag{5.15}$$

where σ is the inertia weight for each particle to keep its speed from the previous iteration, σ_1 and σ_2 are the cognitive coefficient and the social coefficient to control how much weight should be given to refining the search result of the particle itself and recognizing the search result of the swarm consisting of the randomness for exploration [124]. The position of particle x in y-th iteration is given by

$$\Lambda_x^{(y)} = \Lambda_x^{(y-1)} + p_x^{(y-1)}. \tag{5.16}$$

For the PSO-based algorithm, each particle should explore the solution within the solution space $\Lambda \in \mathbb{F}$. Specifically, since TSGs are pre-defined according to the coverage of RSUs, CAVs in the same TSG should have the same sensing interval, given by

$$T_{u_1}^{\mathrm{T}} = T_{u_2}^{\mathrm{T}} = I_{\mathbf{g}}^{\mathrm{T}}, \forall u_1, u_2 \in \mathbf{g}. \tag{5.17}$$

Moreover, all sensing interval pairs in a possible solution of particle positions should fulfill **Proposition 1**. If a particle moves out of the solution space $\mathbb{F}$, it will be replaced by a new particle. In other words, all sensing interval pairs violating **Proposition 1** and Eq. (5.17) are replaced by new randomly selected sensing interval pairs, for the exploration in the solution space.

Algorithm 5 PSO-based algorithm sensing interval determination

1: **Input:** Ω^T, σ, σ_1, σ_2, $y^{\max}$;
2: **Initialization:** $y = 0$, $p_x^{(0)} = 0$, $\Lambda_x^{(0)} \in \mathbb{F}, \forall x \in \mathcal{X}$;
3: **while** $y \leq y^{\max}$ **do**
4: **for** $x \in \mathcal{X}$ **do**
5: **if** Proposition 1 and Eq. (5.17) are not satisfied **then**
6: Replace the particle by a new $\Lambda_x^{(y)} \in \mathbb{F}$;
7: **end if**
8: Obtain sensing intervals $\mathbf{I}_x^{T,(y)}$ of TSGs according to Eq. (5.17);
9: Determine $\Omega_x^{S,(y)}$ and $\mathbf{I}_x^{S,(y)}$ based on Algorithm 6 and calculate $\Xi_x^{(y)}$;
10: **if** $\Xi_x^{(y)} < \hat{\Xi}_x$ **then**
11: Update $\hat{\Lambda}_x \leftarrow \Lambda_x^{(y)}$, $\hat{\Xi}_x \leftarrow \Xi_x^{(y)}$;
12: **end if**
13: **if** $\Xi_x^{(y)} < \Xi^*$ **then**
14: Update $\Lambda^* \leftarrow \Lambda_x^{(y)}$, $\Xi^* \leftarrow \Xi_x^{(y)}$;
15: **end if**
16: Update $p_x^{(y+1)}$ based on (5.15);
17: Select $\Lambda_x^{(y+1)}$ based on (5.16);
18: **end for**
19: **end while**
20: **Output:** Ω^{S}, $\mathbf{I}^{\mathrm{S}}$, and $\mathbf{I}^T$

The details of the proposed PSO-based algorithm are given in **Algorithm 5**. Denote the maximum number of iterations by $y^{\max}$. The particles are initialized with the positions in the solution space $\mathbb{F}$ in Line 2. The position of each particle is updated in each iteration according to Lines 5 to 17. Specifically, with a given position, i.e., the sensing interval pair of each CAV, SSGs Ω^{S} and the corresponding sensing interval $\mathbf{I}^{\mathrm{S}}$ can be obtained through the coalition game-based partitioning algorithm, which will be presented in Subsection 5.4.2. Finally, the outputs of the PSO-based algorithm are the solution of problem **P2** for each time interval.

5.4.2 Coalition Game-Based CAV Partitioning Algorithm

Considering the coverage of RSUs, TSGs are pre-defined at the beginning of each time interval and the RSU sensing interval can be determined based on Eq. (5.17). On the other hand, the optimal SSGs and the corresponding sensing interval still remain to be

determined to solve problem **P2** with minimum resource costs. Recall that the satellite sensing interval for each CAV within an SSG can be calculated according to **Algorithm 5** and may vary among different CAVs. The sensing interval of SSG and the corresponding available offset can be given as follows.

Proposition 2 (Sensing interval and offset values of an SSG)**.** Given the sensing interval and feasible offset values of each CAV within an SSG, the sensing interval of SSG $\mathbf{s}$ $I_{\mathbf{s}}^{\mathrm{S}}$ and the offset $\Psi_{\mathbf{s}}^{\mathrm{S}}$ can be obtained according to the following two cases.

- If all CAVs in SSG $\mathbf{s}$ have the same sensing interval and common offset values, given by
$$\begin{cases} T_{u_1}^{\mathrm{S}} = T_{u_2}^{\mathrm{S}}, \forall u_1, u_2 \in \mathbf{s}, \\ \bigcap_{u \in \mathbf{s}} \hat{\psi}_u^{\mathrm{S}} \neq \emptyset, \end{cases} \tag{5.18}$$
then, the sensing interval of SSG $\mathbf{s}$ is $I_{\mathbf{s}}^{\mathrm{S}} = T_u^{\mathrm{S}}, \forall u \in \mathbf{s}$, and the offset of the SSG is $\Psi_{\mathbf{s}}^{\mathrm{S}} = \arg\max_{\varphi \in \hat{\Psi}_{\mathbf{s}}^{\mathrm{S}}} |T^{\mathrm{th}} - \varphi|$ to maximize the allowable delay for data processing and transmission of each satellite sensing. Here, $\hat{\Psi}_{\mathbf{s}}^{\mathrm{S}} = \bigcap_{u \in \mathbf{s}} \hat{\psi}_u^{\mathrm{S}}$ is the common feasible SSG sensing offset set;
- Otherwise, the sensing interval of SSG $\mathbf{s}$ is $I_{\mathbf{s}}^{\mathrm{S}} = \max\{l\kappa \mid l\kappa < T^{\mathrm{th}}, l \in \mathbb{N}^+\}$ with the offset $\Psi_{\mathbf{s}}^{\mathrm{S}} = 0$.

Note that employing a centralized approach to find the optimal SSGs through a complete search for all possible solutions encounters high complexity. Thus, the coalition game theory emerges as a suitable method to find the optimal partition of CAVs served by the LEO satellite [125].

Definition 3 (The coalition partition)**.** A coalition partition is defined as the set $\Gamma^{\mathrm{S}} = \{\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_m\}$, which partitions the CAV set $\mathcal{U}$ into m coalitions, where $0 \leq m \leq M$. Each coalition, defined as an SSG, is disjoint such that $\forall \mathbf{s}_i, \mathbf{s}_j \in \Gamma^{\mathrm{S}}, i \neq j, \mathbf{s}_i \cap \mathbf{s}_j = \emptyset$ and $\bigcup_i \mathbf{s}_i = \mathcal{U}$.

The CAV partition problem can be modeled as a $(\mathcal{U}, \phi)$ coalition game. Here, $\phi(\mathbf{s})$ is the utility function of SSG $\mathbf{s}$, which is the negative value of the estimated resource cost from the LEO satellite in the time interval, given by
$$\phi(\mathbf{s}) = -C_{\mathbf{s}}^{\mathrm{S}}. \tag{5.19}$$

To minimize the estimated resource cost from the LEO satellite for all SSGs, we aim to find the optimal partition $\tilde{\Gamma}^{\mathrm{S}}$ to maximize the total utility so that
$$\phi(\tilde{\Gamma}^{\mathrm{S}}) > \phi(\Gamma^{\mathrm{S}}), \forall\, \Gamma^{\mathrm{S}} \neq \tilde{\Gamma}^{\mathrm{S}}, \tag{5.20}$$

where $\phi(\Gamma^{\mathrm{S}}) = \sum_{\mathbf{s}\in\Gamma^{\mathrm{S}}} \phi(\mathbf{s})$ is defined as the total utility of coalitions. One key approach in coalition formation is enabling CAVs to join or leave a coalition based on their preferences, aiming to achieve the highest utility. To assess the preference, the concept of preference relation is introduced below.

Definition 4 (Preference relation)**.** For any CAV u, a preference relation is defined by $\succ_u$, where $\mathbf{s}_i \succ_u \mathbf{s}_j$ implies that CAV u strictly prefers to join coalition $\mathbf{s}_i$ over $\mathbf{s}_j$.

Specifically, the preference relation for CAV u is given as

$$\mathbf{s}_i \succ_u \mathbf{s}_j \Leftrightarrow \phi(\mathbf{s}_i) + \phi(\mathbf{s}_j \setminus \{u\}) > \phi(\mathbf{s}_j) + \phi(\mathbf{s}_i \setminus \{u\}), \tag{5.21}$$

where $\mathbf{s}_i$ and $\mathbf{s}_j$ are two potential coalitions containing CAV u. When CAV u finds a more preferable coalition, it can join the new coalition through the switch operation as defined below.

Definition 5 (Switch operation)**.** Considering a CAV partition Γ^{S}, if CAV u in coalition $\mathbf{s}_1$ can switch to coalition $\mathbf{s}_2$ to achieve a higher utility based on the preference relation, i.e., $\mathbf{s}_2 \cup \{u\} \succ_u \mathbf{s}_1$, a switch operation occurs. Correspondingly, the partition is modified as follows:

$$\Gamma^{\mathrm{S}} = (\Gamma^{\mathrm{S}} \setminus \{\mathbf{s}_1, \mathbf{s}_2\}) \cup \{\mathbf{s}_1 \setminus \{u\}\} \cup \{\mathbf{s}_2 \cup \{u\}\}. \tag{5.22}$$

With a sequence of switch operations, CAVs can detach from their current coalitions and join new coalitions until a stable partition is achieved, defined as follows.

Definition 6 (Stability)**.** A partition is Nash-stable if no CAV tends to change the belonging of the coalition in the current partition [126].

Based on these definitions, we design a coalition formation game-based algorithm for CAV partitioning, with the main workflow given in **Algorithm 6**. At the initial step, CAVs are partitioned into M coalitions, which is the maximum number of coalitions that can be supported by the satellite. Then, as shown in lines 4 to 10, it is determined whether a CAV should switch the coalition for a higher utility. $\mathcal{H}(u)$ indicates the history set of coalitions where CAV u has previously joined to avoid repetition. Based on **Definition 5**, each CAV can detach from the current coalition and join a new coalition, until a stable partition is obtained as per **Definition 6**. Finally, SSGs are determined by $\Omega^{\mathrm{S}} = \Gamma^{\mathrm{S}}$ with the sensing interval of all coalitions and their corresponding offset according to **Proposition 2**.

Algorithm 6 Coalition formation game-based algorithm for CAV partitioning

Input: $\mathcal{U}, \Lambda$;
Initialization: The initial partition of CAVs is $\Gamma^{\mathrm{S}}_{\mathrm{ini}}$, where CAVs are partitioned into M coalitions;
repeat
  Randomly select CAV u and choose another coalition $\mathbf{s} \in \Gamma^{\mathrm{S}}$;
  if CAV prefers to join a new coalition $\mathbf{s}$ **and** $\mathbf{s} \cup \{u\} \notin \mathcal{H}(u)$ **then**
    Add $\mathbf{s} \cup \{u\}$ to the history set $\mathcal{H}(u)$;
    CAV u detaches the current coalition and combines with the new coalition according to Definition 5;
    Update partition Γ^{S};
  end if
until Switch operation terminates with a stable partition
Output: Final SSGs $\Omega^{\mathrm{S}} = \Gamma^{\mathrm{S}}$, the sensing interval of all coalitions $\{I^{\mathrm{S}}_1, ..., I^{\mathrm{S}}_m\}$, and the offset $\{\Psi^{\mathrm{S}}_1, ..., \Psi^{\mathrm{S}}_m\}$ according to Proposition 2.

5.4.3 Computational Complexity Analysis

In **Algorithm 5**, a PSO is used to determine the sensing interval for each CAV. In each iteration of the proposed PSO-based algorithm, the positions of each particle are updated. The computational complexity of updating these particles in each iteration is $O(|\mathcal{X}|NU)$, where $|\mathcal{X}|$ is the number of particles and NU represents the dimension of each particle. Specifically, each particle's position encodes the sensing interval pairs for CAVs, which include the sensing interval from both the satellite and RSU. Since CAVs within the same TSG are required to have identical sensing intervals from RSU given in Eq. (5.17) and there are N RSUs in the target area, the dimensional of each particle can be given by NU. After determining the sensing interval of each CAV, a coalition game is used to partition CAVs into different SSGs in **Algorithm 6**. In the context of the coalition game, the maximum number of CAV selections is UM. By considering the worst case, each CAV is evaluated against $M-1$ coalitions in each iteration. This results in a computational complexity of $O(UM^2)$ for each iteration of the coalition game. Therefore, the computational complexity for determining CAV partitions and sensing intervals for each SG is $O(L_1|\mathcal{X}|(NU + L_2UM^2))$, where L_1 and L_2 are the finite number of iterations required for the PSO and coalition game to converge.

5.5 PPO-Based Real-Time Resource Allocation Algorithm

With the determined SSGs and sensing intervals, the LEO satellite needs to allocate computing and communication resources for processing sensing data and transmitting sensing results to CAVs in each SSG. In this section, we reformulate problem **P3** by leveraging the Markov decision process (MDP) and propose a reinforcement learning-based algorithm to solve the problem.

5.5.1 Markov Decision Process Formulation

Since constraint (5.10a) in **P3** is related to long-term performance in delay and reliability, it is challenging to solve the sequential decision problem directly. To decompose the long-term impact of the delay and reliability in each time slot, we construct a virtual reliability queue based on the Lyapunov optimization technique [127]:

$$H_u(t+1) = \max\{H_u(t) + \mathbb{1}_{q_u(t)>T^{\text{th}}} - \varepsilon, 0\}, \tag{5.23}$$

where $\mathbb{1}_{condition} = 1$ if *condition* is true and $\mathbb{1}_{condition} = 0$ otherwise, and $H_u(1) = 0$. When the AoI for CAV u violates the freshness requirement at time slot t, the virtual reliability queue $H_u(t+1)$ enlarges at time slot $t+1$. On the other hand, the virtual reliability queue will decrease if a new sensing result is received within the freshness requirement. As a historical measurement of freshness-related task failures, the virtual reliability queue reflects the deviation of the current failure rate from the reliability constraint. Therefore, the long-term delay and reliability constraint is transformed into guaranteeing the stability of the virtual reliability queue. Let $H_{\mathbf{s}}^{\max}(t)$ denote the group-based virtual queue value reflecting the reliability of SSG $\mathbf{s}$, which is the maximal virtual reliability queue value of CAVs in the SSG, given by $H_{\mathbf{s}}^{\max}(t) = \max_{u\in\mathbf{s}} H_u(t)$. Therefore, we can transform optimization problem **P3** into

$$\begin{aligned}\mathbf{P4}\ \min_{\mathcal{C}^{\mathrm{S}},\mathcal{B}^{\mathrm{S}}} & \sum_{t=1}^{\mathrm{T}} \left(\sum_{\mathbf{s}\in\Omega^{\mathrm{S}}(t)} \left(\Upsilon_{\mathbf{s}}^{\mathrm{S}}(t) + \eta H_{\mathbf{s}}^{\max}(t)\right) + \sum_{\mathbf{g}\in\Omega^{\mathrm{T}}(t)} \Upsilon_{\mathbf{g}}^{\mathrm{T}}(t) \right) \\ & \text{s.t. } (5.4), (5.5), (5.7), (5.8),\end{aligned} \tag{5.24}$$

where the weight value η is used to balance the resource cost and the reliability.

Since the formulated computing and communication resource allocation problem **P4** is an integer nonlinear optimization problem with random channel conditions and dynamic

mobility of CAVs, it cannot be solved by deterministic optimization algorithms. Thus, we reformulate problem **P4** by leveraging MDP to model the randomness of channels and CAV mobility.

Let $\zeta_{\mathbf{s}}(t) = [\mathbf{p}_{\mathbf{s}}^{\mathrm{SSG}}(t), q_{\mathbf{s}}^{\max}(t), H_{\mathbf{s}}^{\max}(t), Q_{\mathbf{s}}^{\mathrm{S}}(t), W_{\mathbf{s}}^{\mathrm{S}}(t), c_{\mathbf{s}}^{\mathrm{S}}(t-1), b_{\mathbf{s}}^{\mathrm{S}}(t-1)]$ denote the state of SSG $\mathbf{s}$ at time slot t, where $q_{\mathbf{s}}^{\max}(t) = \max_{u\in\mathbf{s}}\{q_u(t)\}$. Considering the varying SSG numbers in different time intervals, the variability in state and action spaces for all SSGs poses a challenge in solving the problem. Therefore, we use the maximum number M to regulate the overall state and action spaces for all SSGs. The overall state at time slot t is denoted by $\mathcal{S}(t) = [\zeta_{\mathbf{s}_1}(t), ..., \zeta_{\mathbf{s}_M}(t)] \in \mathfrak{S}$, where $\mathfrak{S}$ is the state space with all possible states. If the SSG number $|\Omega^{\mathrm{S}}(t)| < M$, $\zeta_{\mathbf{s}_i}(t) = [0]_{1\times|\zeta_{\mathbf{s}_i}|}$, $\forall i > |\Omega^{\mathrm{S}}(t)|$.

Let $\mathfrak{A} = \mathfrak{A}^c \otimes \mathfrak{A}^b$ denote the action space, formed by combining the action spaces for the computing and communication resource allocation, where $\otimes$ is the Cartesian product. Thus, the action at time slot t is denoted by $\mathcal{A}(t) = \{(c_{\mathbf{s}_1}^{\mathrm{S}}(t), b_{\mathbf{s}_1}^{\mathrm{S}}(t)), ..., (c_{\mathbf{s}_M}^{\mathrm{S}}(t), b_{\mathbf{s}_M}^{\mathrm{S}}(t))\} \in \mathfrak{A}$, where the action space is continuous with $\mathcal{A}(t) \in ([0,1]\times[0,1])^{1\times M}$.

To evaluate an action with a given state, the set of rewards is denoted by $\mathfrak{R} := \mathfrak{S}\times\mathfrak{A} \to \mathbb{R}$. We define the reward function obtained from taking action $\mathcal{A}(t)$ in state $\mathcal{S}(t)$ as

$$\mathcal{R}(\mathcal{S}(t), \mathcal{A}(t)) = -\sum_{\mathbf{s}\in\Omega^{\mathrm{S}}(t)} \left(\Upsilon_{\mathbf{s}}^{\mathrm{S}}(t) + \eta H_{\mathbf{s}}^{\max}(t)\right) - \sum_{\mathbf{g}\in\Omega^{\mathrm{T}}(t)} \Upsilon_{\mathbf{g}}^{\mathrm{T}}(t). \tag{5.25}$$

Let $\mathbf{\Pi}$ denote the set of all policies, which is the function mapping from state space $\mathfrak{S}$ to action space $\mathfrak{A}$. To estimate the expected return of the current decision-making, a discounted cumulative reward function is considered and the objective function is transformed as

$$\max_{\pi\in\mathbf{\Pi}} \mathbb{E}\left[\sum_{t=1}^{\mathrm{T}} \iota^t \mathcal{R}(\mathcal{S}(t), \mathcal{A}(t))\middle|\pi\right], \tag{5.26}$$

where $\iota \in (0,1)$ is a discounted factor to reflect the importance of future rewards compared with immediate rewards.

5.5.2 PPO-Based Algorithm for Resource Allocation

Considering the lack of information on state transition probabilities and the continuous action space, policy gradient-based reinforcement learning is a potential method to solve the MDP-based problem. This method directly updates the policy to maximize the total reward without requiring state transition probabilities. However, the vanilla policy gradient

approach often suffers from high variance in parameter updates, leading to substantial policy fluctuation during learning and potential instability. Therefore, we propose to adopt a PPO-based algorithm [103], which can reuse the historical samples and handle a large-scale problem to find the optimal policy. Specifically, the PPO-based algorithm aims to find the optimal parameter θ for policy π_θ that maximizes the clipped surrogate objective

$$L^{\text{clip}}(\theta) = \mathbb{E}\left\{\min(\xi_\theta A^{\pi_{\theta_{\text{old}}}}, \text{clip}(\xi_\theta, 1-\delta, 1+\delta)A^{\pi_{\theta_{\text{old}}}})\right\}. \tag{5.27}$$

Specifically,

$$\xi_\theta = \frac{\pi_\theta(\mathcal{A}|\mathcal{S})}{\pi_{\theta_{\text{old}}}(\mathcal{A}|\mathcal{S})} \tag{5.28}$$

is the policy updating ratio based on the current policy and $A^{\pi_{\theta_{\text{old}}}}$ is the estimated advantage from policy $\pi_{\theta_{\text{old}}}$, which can be expressed as

$$\begin{aligned} &A^{\pi_{\theta_{\text{old}}}}(\mathcal{S}(t), \mathcal{A}(t)) = \\ &\quad \mathcal{R}(\mathcal{S}(t), \mathcal{A}(t)) + \iota V_\vartheta(\mathcal{S}(t+1)) - V_\vartheta(\mathcal{S}(t)), \end{aligned} \tag{5.29}$$

where $V_\vartheta(\mathcal{S}(t))$ is a state-value function parameterized by ϑ, given by

$$V_\vartheta(\mathcal{S}(t)) = \mathbb{E}\left\{\sum_{i=0}^{\infty} \iota^i \mathcal{R}(\mathcal{S}(t+i+1), \mathcal{A}(t+i+1)) \,|\, \mathcal{S}(t)\right\}. \tag{5.30}$$

The clipping function clips the ratio ξ_θ between $1-\delta$ and $1+\delta$ with a clipping hyperparameter δ, which removes incentives for the new policy to get far from the old policy to improve the learning stability.

The overall PPO-based resource allocation algorithm is given in **Algorithm 7**, where E is the epoch number for training in each episode to update the network parameters through different minibatches. Note that given the diverse scales of components in the state, we perform normalization on each component to improve the stability of the training. First, the policy and the value functions are initialized with parameters θ and ϑ, respectively. The buffer $\mathcal{B}$ is used to store transitions resulting from interactions with the environment. Then, at each time slot, the LEO satellite collects the state from the environment, takes actions based on the current policy, computes the reward, observes the new state, and records transitions into buffer based on Lines 5 to 7. After T time slots, the policy and value function will be updated based on the clipped surrogate objective and the mean-square error based on Lines 13 to 15.

Algorithm 7 PPO-based resource allocation algorithm

Initialization: Parameters θ and ϑ for policy and state-value function, and buffer $\mathcal{B}$;
for $j \in \{1, 2, ...\}$ **do**
  Obtain initial state $\mathcal{S}(1)$;
  for $t \in \{1, 2, ..., T\}$ **do**
    Execute action $\mathcal{A}(t)$ based on the policy π_θ;
    Compute reward $\mathcal{R}(\mathcal{S}(t), \mathcal{A}(t))$ and observe new state $\mathcal{S}(t+1)$;
    Record transition $(\mathcal{S}(t), \mathcal{A}(t), \mathcal{R}(t), \mathcal{S}(t+1))$ into buffer $\mathcal{B}$;
    if $t \mod k = 0$ **then**
      Run **Algorithm 5** to adjust sensing interval and CAV partition;
    end if
  end for
  for $e \in \{1, 2, ..., E\}$ **do**
    Sample a minibatch of transitions $\mathcal{B}_N$ from buffer $\mathcal{B}$;
    Update the policy parameter $\theta \leftarrow \arg\max_\theta \frac{1}{|\mathcal{B}_N|} \sum_{t_N \in \mathcal{B}_N} L^{\text{clip}}(\theta)$;
    Update state-value function parameter
    $\vartheta \leftarrow \arg\min_\vartheta \frac{1}{|\mathcal{B}_N|} \sum_{t_N \in \mathcal{B}_N} \left(\mathcal{R}(t_N) + V_\vartheta(\mathcal{S}(t_N + 1)) - V_\vartheta(\mathcal{S}(t_N))\right)^2$
  end for
  Empty buffer $\mathcal{B}$ for a new training episode;
end for
Output: Parameter θ;

5.6 Performance Evaluation

In this section, extensive simulations are carried out to evaluate the proposed CSTRS scheme. Specifically, we first elaborate on the simulation settings. Afterward, simulation results are presented to demonstrate the effectiveness of our proposed scheme.

5.6.1 Simulation Setting

In our simulation scenario, the target area is a beam coverage of a steerable LEO satellite. The orbit altitude of the LEO satellite is 550 km, and the satellite velocity is 7.56 km/s. CAVs and RSUs are randomly distributed in the area, with the radius of the coverage area of each RSU being 500 m. Specifically, the target area consists of both urban and rural areas, where different areas have different densities of RSUs and CAVs. By default, urban areas have RSU densities of 10 per km^2 and CAV densities of 40 per km^2, while rural areas

Table 5.2: Simulation Parameters

LEO satellite transmission power P^{S}	40 dBm
LEO satellite antenna gains	32 dBi
LEO satellite bandwidth	250 MHz
LEO satellite path-loss exponent	2.5
CPU frequency in LEO satellite	1 Gcycles/s
RSU transmission power	32 dBm
RSU antenna gains	10 dBi
RSU bandwidth	100 MHz
RSU path-loss exponent	3
CPU frequency in RSU	1 Gcycles/s
Noise power σ^2	-174 dBm/Hz
CAV antenna gains	10 dBi
Unit cost of resources $\chi_1^{\mathrm{S}}, \chi_2^{\mathrm{S}}, \chi_1^{\mathrm{T}}, \chi_2^{\mathrm{T}}$	0.5, 2.5, 0.1, 0.5
Workload of sensing data processing	25 cycles/bit

have RSU densities of 2 per km^2 and CAV densities of 20 per km^2, respectively. In the simulated scenario, CAVs move with stochastic direction and speed in each time slot. The maximum freshness requirement of CAVs is set as 100 ms. The time duration for each time slot is set to be $\tau = 5$ ms and the time interval for adjusting SSGs is 10 s. The minimum sensing interval κ is 20 ms. The default reliability requirement in this simulation is 0.99. More simulation parameters, which are related to the computing and communication of satellites and RSUs [128–130], are listed in Table 5.2.

For the training of the PPO-based reinforcement learning model, the deep neural network utilized in our proposed algorithm includes three fully connected hidden layers with (128, 128, 64) neurons. The ReLU activation function is applied after each hidden layer to realize a nonlinear approximation. Each training episode spans 8000 time slots with CAVs and RSUs randomly reset at the beginning of each episode.

5.6.2 Simulation Results

We first present the convergence performance of the proposed CSTRS scheme in terms of resource cost and reliability in each episode of the training period. The simulation comprises

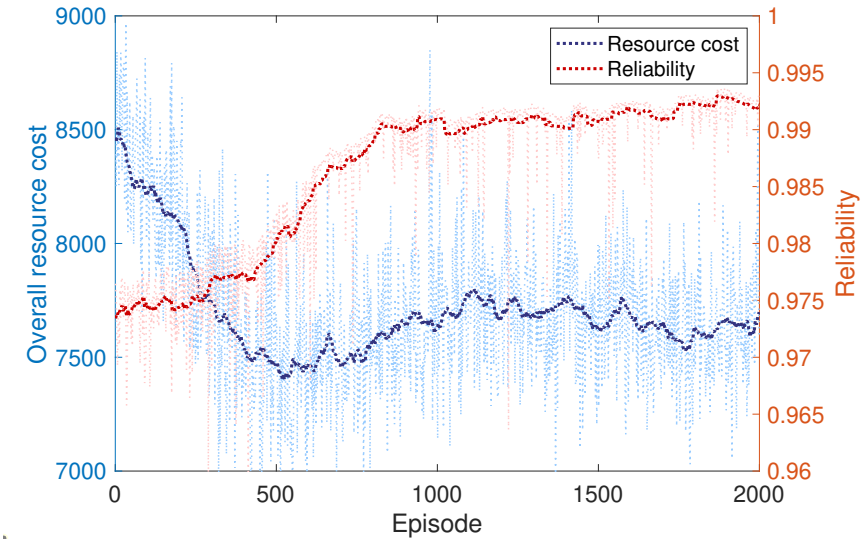
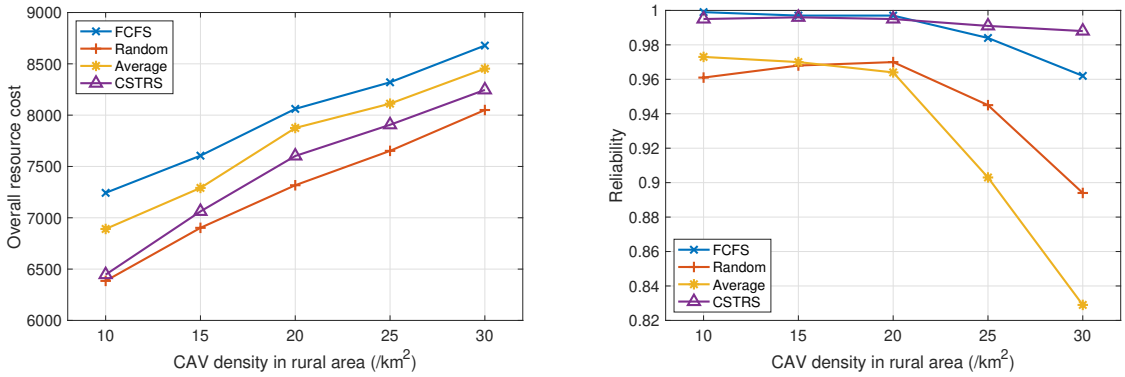


Figure 5.4: Convergence performance of training process.

1800 episodes with a learning rate of 3×10^{-4} and a clipping ratio of 0.3. Figure 5.4 shows the overall accumulative resource cost and reliability in each episode, which converges after around 1100 episodes.

Next, we perform several experiments to validate the efficacy of each component in CSTRS. We first evaluate the effectiveness of the PPO-based resource allocation algorithm. Specifically, we adopt the following three benchmark algorithms for resource allocation [131], namely 1) *FCFS*-based allocation: the LEO satellite adopts the first-come-first-serve policy and allocates all available resources to the first requesting SSG; 2) *Random* allocation: the LEO satellite allocates computing and communication resources randomly, only considering the resource capacity constraints; and 3) *Average* allocation: computing and communications resources are allocated equally to each SSG. For *FCFS*-based resource allocation algorithm, if multiple SSGs start to process or transmit sensing data at the same time, all available resources will be allocated equally to serve SSGs.

Considering that the LEO satellite is essential to providing environment sensing services for CAVs in rural areas, the performances of different resource allocation algorithms under different CAV densities in rural areas are shown in Fig. 5.5. We can find that our proposed CSTRS scheme can well satisfy the reliability requirements with low resource cost by efficiently allocating resources to adapt to dynamic network environments. When CAV density is small, the *FCFS* algorithm and the CSTRS scheme can both satisfy the reliability



(a) Resource cost performance.

(b) Reliability performance.

Figure 5.5: Performance of different resource allocation algorithms.

requirement of 0.99, whereas our proposed scheme has lower resource cost. With a large CAV density, *FCFS* can hardly guarantee reliability due to the improper resource scheduling policy for different SSGs, while the CSTRS scheme consistently satisfies the reliability requirements.

In Fig. 5.6, we compare the resource cost in the LEO satellite with different reliability requirements. The results show the adaptivity of our proposed CSTRS scheme, which can achieve efficient resource costs with any given reliability requirements by allocating resources for different SSGs.

Next, we investigate the effectiveness of the PSO-assisted sensing interval and CAV partition algorithm in our proposed CSTRS scheme. For comparison, we introduce a benchmark algorithm named *fixed-number SSG*, where the SSG number in the serving area is pre-determined and fixed during the simulation. The sensing interval of each SG is determined through **Algorithm 5**, while SSGs are determined by k-means algorithm based solely on CAV positions at the beginning of each time interval. The communication and computing resources are allocated by adopting the PPO-based algorithm.

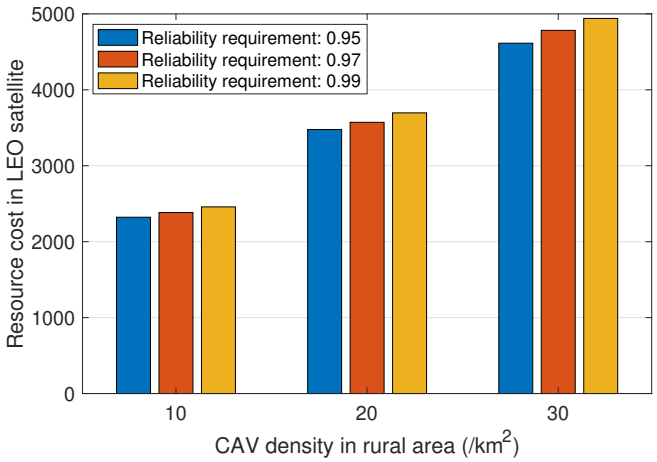


Figure 5.6: Resource cost in LEO satellite with different reliability requirements.

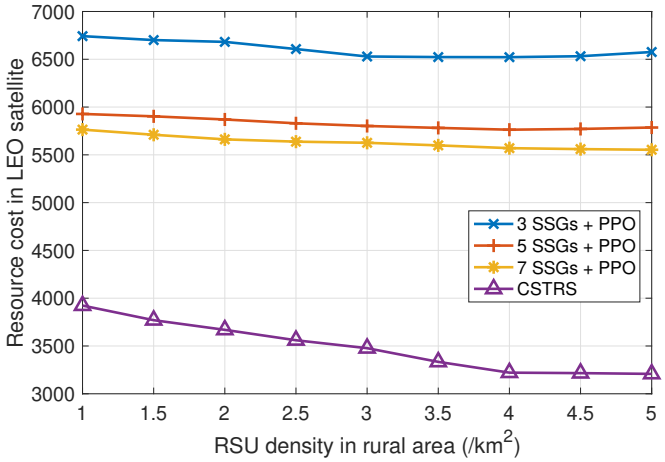
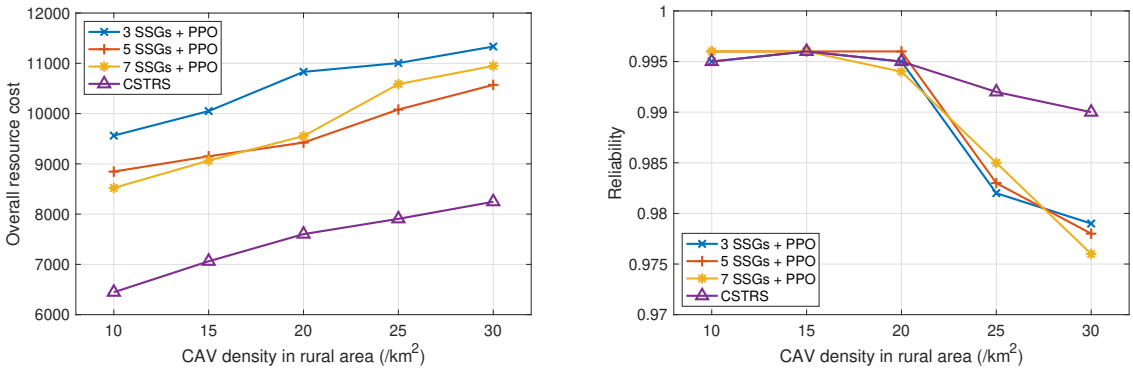


Figure 5.8: Resource cost in LEO satellite under different RSU densities in rural area.

In Fig. 5.7, we compare the resource cost and reliability performances of different CAV partition methods with different CAV densities in rural areas. Specifically, our proposed CSTRS scheme has a low resource cost compared with *fixed-number SSG* strategies. This advantage stems from the adaptive adjustment of CAV partitions and sensing intervals of SGs in the CSTRS scheme based on dynamic environments by jointly considering both LEO satellite and terrestrial networks, enhancing the overall efficiency. With the increment of CAV densities, resource cost increases accordingly, and the CSTRS scheme consistently



(a) Resource cost performance.

(b) Reliability performance.

Figure 5.7: Performance of different CAV partition policies.

maintains a low level of resource cost while satisfying reliability requirements. On the other hand, *fixed-number SSG* strategies face challenges in effectively utilizing terrestrial and satellite resources due to improper CAV partitioning and sensing intervals. Particularly, the reliability of *fixed-number SSG* strategies cannot be guaranteed for environment sensing with a high CAV density due to insufficient resources.

In Fig. 5.8, we compare the LEO satellite resource cost of different CAV partition strategies with different RSU densities in the rural area. From the figure, we observe that the CSTRS scheme consumes less resources in the LEO satellite compared to *fixed-number SSG* policies, due to the joint decision of CAV partition and sensing interval between satellite and terrestrial networks. Moreover, when RSU density increases, the LEO satellite resource cost of the CSTRS scheme reduces significantly, since more CAVs can be served by RSUs by adaptively adjusting SSGs. However, *fixed-number SSG* strategies struggle to effectively manage resources across satellite and terrestrial networks in dynamic environments. In this case, despite increased RSU densities, the sensing interval remains small for SSGs, resulting in a high resource cost.

Last, we investigate the resource cost for CAVs in rural areas from the LEO satellite and terrestrial RSUs as shown in Fig. 5.9. We can observe that the resource cost from the LEO satellite decreases for CAVs with increasing RSU density in rural areas, while the resource cost from RSUs increases to serve more CAVs. Specifically, the decline in satellite resource cost is more pronounced than the increase in RSU resource cost since the unit cost of satellite resources is higher than that of RSU resources. In this case, cooperative resource scheduling from satellite and terrestrial networks can effectively reduce the overall

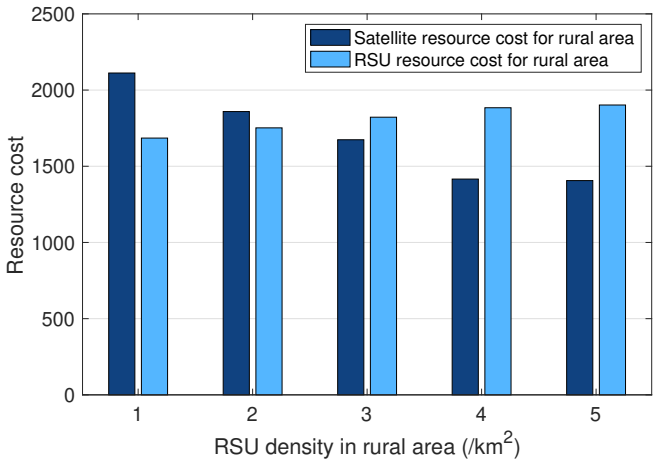


Figure 5.9: Resource cost in rural areas from terrestrial and LEO satellite networks.

resource cost with more network support from RSUs.

5.7 Summary

In this chapter, we have investigated a joint communication and computing resource scheduling problem for infrastructure-assisted environment sensing in STIVN. To cope with the dynamic network environment and satisfy the stringent freshness and reliability requirements for environment sensing, we have proposed the CSTRS scheme, which is a hybrid data-model co-driven approach for resource scheduling in STIVN to jointly find the optimal CAV partition, sensing interval, and resource allocation decisions. The proposed scheme can cooperatively manage resources from both satellite and terrestrial networks, capable of leveraging the advantages of different networks, thus reducing resource costs while satisfying service requirements in STIVN. Simulation results demonstrate that the proposed CSTRS scheme can significantly reduce resource usage with high reliability compared to the benchmark schemes.

Chapter 6

Conclusions and Future Works

In this chapter, we summarize the main results and contributions of this thesis and present our future research directions.

6.1 Main Research Contributions

In this thesis, we have investigated slicing-based resource management in STIVN. Specifically, three resource management schemes have been designed in the reservation and operation stage for low-complexity, flexible, and adaptive resource management in STIVN, i.e., TLRS scheme in the reservation stage, supporting both multicast and unicast communication modes with joint communication and caching resource reservation with low complexity; DRS scheme for flexibly coordinating different networks across multiple cells in the reservation stage, which jointly determine the access satellites and communication resource reservation; and the CSTRS scheme in the operation stage which adaptively determine the sensing interval, CAV grouping, and resource allocation in STIVN. The proposed slicing-based resource management schemes enable low-complexity, flexible, and adaptive resource management for STIVN. The main contributions of this thesis are summarized as follows:

1. A reservation-stage multi-dimensional resource management has been investigated, where heterogeneous network segments can cooperate to support V2X applications. Considering the different requirements of multicast and unicast communication modes, a resource slicing architecture with two subslices has been developed. To find the optimal solution to minimize the overall cost while maximizing the service demand that

can be accommodated to meet the service requirements, we have proposed a TLRS scheme to jointly optimize the slicing window length, multicast, and communication and caching resource reservation decisions. TLRS scheme is a data-model co-driven approach, which can effectively reduce the complexity in decision-making. Specifically, the slicing window length and communication resource reservation decisions are made through an HPPO-based algorithm, while the multicast and caching decisions are determined using a matching-based algorithm. The problem formulation of resource slicing and the designed scheme in this work can provide a theoretical basis for future studies for effectively managing multi-dimensional decisions in STIVN in the reservation stage with low complexity.

2. We have investigated the joint optimization of communication resources and satellite selection designs in the reservation stage in STIVN with cross-cell coordination. For resource management across multiple cells, the impact of factors includes diversified service demand, as well as time-varying and distinct available resources in each cell. We have formulated a long-term resource slicing problem and proposed a DRS scheme for flexible resource management. With the DRS scheme, we develop an AMAPPO-based algorithm to select satellites serving each cell by learning the state transition probability with partial observation and then derive a distributed optimization algorithm to optimize the resource reservation decisions. This work provides insight into providing a decentralized resource slicing solution in the reservation stage to facilitate cross-cell coordination with a flexible resource management scheme in STIVN.

3. We have proposed a CSTRS scheme for efficient cooperative resource slicing in STIVN in the operation stage, which jointly optimizes the sensing interval and resource allocation for CAVs. Benefiting from the multicast feature of the LEO satellite, we group CAVs based on their mobility patterns and enable group-based resource allocation to minimize resource costs with satisfied service requirements. With the CSTRS scheme, we design a coalition game, and PSO-based algorithms are designed to partition CAVs and develop a reinforcement learning-based algorithm to make real-time computing and communication resource allocation decisions. This work provides an adaptive resource management solution for real-time resource operation in STIVN to cope with highly dynamic environments.

6.2 Future Works

For the future research, there are some interesting and promising research directions listed as follows:

1. **User-centric resource slicing in STIVN -** In STIVN, users may have different preferences for connecting to satellite or terrestrial networks due to factors such as latency, coverage, or service type. For instance, some users prefer satellite connections for extensive coverage in remote areas, while others might favor terrestrial networks for their lower latency in urban environments. Additionally, even for the same application, users may have varying Quality of Experience (QoE) needs, such as differing tolerances for delay, jitter, or data loss. This variability presents a challenge, as the service-oriented performance of a network slice may not adequately meet the needs of each individual user. In this case, user-centric resource slicing in STIVN is essential, where the network must cater to the tradeoff between managing resources for individual users and optimizing for overall application performance. A potential solution involves leveraging user data, such as position, SLA, historical behavior, etc, to capture the performance of slices and understand user preferences more accurately. By collecting and analyzing this data, resource management policies can be timely adjusted to better satisfy both QoS and QoE requirements with efficient resource utilization. However, this approach introduces challenges in managing the additional complexity and potential costs associated with more granular resource management with the collection and processing of data. While taking benefits of data for both users and networks, the design of user-centric resource slicing needs further investigation.

2. **Resource management for AI Services -** With the rapid development of AI in different research and industrial areas, AI can function as a service, namely AI services. Designing network management schemes to facilitate AI services, also known as "Networking for AI", is becoming increasingly crucial, particularly as AI services such as real-time data analytics, distributed training, federated learning, generative AI models, and large language models require substantial 3C resources. Achieving a better AI service performance requires not only high bandwidth and low latency but also robust coordination across diverse network nodes, whereas current networks do not focus on improving AI service performance by efficiently managing resources. AI services have unique characteristics, such as precise synchronization, efficient data exchange, and adaptive resource allocation to ensure that the training process remains consistent and timely across the network, which makes resource management challenging. Heterogeneous networks, such as STIVN, provide a promising environment

for such AI services by allowing different network nodes to take on specialized roles, such as data preprocessing, model training, or inference. However, challenges still exist including adaptive resource scheduling in satisfying varying demands of AI tasks, balancing the computational load across heterogeneous nodes, and ensuring that AI services maintain their performance under fluctuating network conditions. These challenges highlight the need for advanced resource management strategies specifically designed to enhance the operation of AI services.

References

[1] H. Wang, T. Liu, B. Kim, C.-W. Lin, S. Shiraishi, J. Xie, and Z. Han, "Architectural design alternatives based on cloud/edge/fog computing for connected vehicles," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2349–2377, 2020.

[2] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," *NHTSA, US Department of Transportation*, vol. 2, pp. 1–2, 2015.

[3] World Health Organization, *Global status report on road safety*. World Health Organization, 2023.

[4] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, 2018.

[5] U. Montanaro, S. Dixit, S. Fallah, M. Dianati, A. Stevens, D. Oxtoby, and A. Mouzakitis, "Towards connected autonomous driving: review of use-cases," *Veh. Syst. dynamics*, vol. 57, no. 6, pp. 779–814, 2019.

[6] Z. Wang, H. Zhu, M. He, Y. Zhou, X. Luo, and N. Zhang, "GAN and multi-agent DRL based decentralized traffic light signal control," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1333–1348, 2021.

[7] M. He, X. Luo, Z. Wang, F. Yang, H. Qian, and C. Hua, "Global traffic state recovery via local observations with generative adversarial networks," in *IEEE ICASSP*, 2020, pp. 3767–3771.

[8] 5GAA, "C-V2X use cases volume II: Examples and service level requirements," *White Paper*, Aug. 2020.

[9] C. Zhou, J. Gao, M. Li, N. Cheng, X. Shen, and W. Zhuang, "Digital twin-based 3D map management for edge-assisted device pose tracking in mobile AR," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17 812–17 826, 2024.

[10] ETSI, "Intelligent transport systems (ITS); vehicular communications; basic set of applications; definitions," TR 102 638, ETSI, Tech. Rep., 2009.

[11] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1243–1274, 2018.

[12] Z. Amjad, A. Sikora, B. Hilt, and J.-P. Lauffenburger, "Low latency V2X applications and network requirements: Performance evaluation," in *Proc. IEEE IV*. IEEE, 2018, pp. 220–225.

[13] 3GPP, "Service requirements for enhanced V2X scenarios," TS 22.186, 3GPP, Tech. Rep., 2019.

[14] ——, "NR and NG-RAN overall description, version 17.2.0, release 17," 3GPP, Tech. Rep., Standard TS 38.300, 3GPP, Sept. 2022.

[15] X. Shen, J. Gao, M. Li, C. Zhou, S. Hu, M. He, and W. Zhuang, "Toward immersive communications in 6G," *Front. Comput. Sci.*, vol. 4, p. 1068478, 2023.

[16] N. Cheng, J. He, Z. Yin, C. Zhou, H. Wu, F. Lyu, H. Zhou, and X. Shen, "6G service-oriented space-air-ground integrated network: A survey," *Chinese J. Aeronaut.*, 2021.

[17] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband LEO satellite communications: Architectures and key technologies," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 55–61, 2019.

[18] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62–69, Apr. 2019.

[19] Telesat, https://www.telesat.com/leo-satellites/.

[20] Starlink, https://www.starlink.com/.

[21] OneWeb, https://oneweb.net/.

[22] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, 2017.

[23] T. Hong, W. Zhao, R. Liu, and M. Kadoch, "Space-air-ground IoT network and related key technologies," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 96–104, 2020.

[24] 3GPP, "Solutions for NR to support non-terrestrial networks (NTN)," 3GPP, Technical Report (TR) 38.821, 2021, version 16.1.0, Release 16.

[25] M. He, C. Hua, W. Xu, P. Gu, and X. Shen, "Delay optimal concurrent transmissions with raptor codes in dual connectivity networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1478–1491, 2021.

[26] M. He, C. Hua, and P. Gu, "Design and analysis for dual connectivity and raptor codes assisted handover in vehicular networks," in *IEEE WCNC*. IEEE, 2020, pp. 1–7.

[27] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of things realization in 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 957–994, 2021.

[28] C. Zhou, J. Gao, M. Li, X. Shen, W. Zhuang, X. Li, and W. Shi, "AI-assisted slicing-based resource management for two-tier radio access networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 6, pp. 1691–1706, 2023.

[29] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 45–66, 2020.

[30] C. Zhou, J. Gao, M. Li, X. Shen, and W. Zhuang, "Digital twin-empowered network planning for multi-tier computing," *J. Commun. Inf. Netw.*, vol. 7, no. 3, pp. 221–238, 2022.

[31] 3GPP, "Study on new radio (NR) to support non-terrestrial networks, version 15.4.0, release 15," 3GPP, Tech. Rep., Standard TR 38.811, 3GPP, Oct. 2020.

[32] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff *et al.*, "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, 2020.

[33] I. Del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta astronautica*, vol. 159, pp. 123–135, 2019.

[34] 3GPP, "Service requirements for the 5G system; stage 1, version 19.0.0, release 19," 3GPP, Tech. Rep., Standard TS 22.261, 3GPP, Sept. 2022.

[35] ——, "Study on scenarios and requirements for next generation access technologies, version 17.0.0, release 17," 3GPP, Tech. Rep., Standard TR 38.913, Apr. 2022.

[36] ——, "Study on using satellite access in 5G; stage 1. version 16.0.0. release 16," 3GPP, Tech. Rep., Standard TR 22.822, 3GPP, Jul. 2018.

[37] ETSI, Satellite Earth Stations and Systems (SES), "Combined satellite and terrestrial networks scenarios, document v1.1.1," ETSI, Tech. Rep., ETSI TR 103 124, Jul. 2013.

[38] ——, "Overview of present satellite emergency communications resources, v1.2.2," ETSI, Tech. Rep., ETSI TR 102 641, Aug. 2013.

[39] ETSI, System Reference document (SRdoc), "Cognitive radio techniques for satellite communications operating in Ka band. v1.2.1," ETSI, Tech. Rep., ETSI TR 103 263, Feb. 2016.

[40] ETSI, Satellite Earth Stations and Systems (SES), "Multi-link routing scheme in hybrid access network with heterogeneous links, v1.1.1," ETSI, Tech. Rep., ETSI TR 103 351, Jul. 2017.

[41] ETSI, Broadband Radio Access Networks (BRAN), "Broadband wireless access and backhauling for remote rural communities. v1.1.1," ETSI, Tech. Rep., ETSI TR 103 293, Jul. 2015.

[42] ETSI, Satellite Earth Stations and Systems (SES), "Broadband satellite multimedia (BSM); common air interface specification; satellite independent service access point (SISAP) interface: Primitives. v1.2.1," ETSI, Tech. Rep., ETSI TS 102 357, May 2015.

[43] D.-H. Jung, J.-G. Ryu, W.-J. Byun, and J. Choi, "Performance analysis of satellite communication system under the shadowed-rician fading: A stochastic geometry approach," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2707–2721, 2022.

[44] B. Al Homssi and A. Al-Hourani, "Modeling uplink coverage performance in hybrid satellite-terrestrial networks," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3239–3243, 2021.

[45] H. Lin, C. Zhang, Y. Huang, R. Zhao, and L. Yang, "Fine-grained analysis on downlink LEO satellite-terrestrial mmwave relay networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1871–1875, 2021.

[46] J. Ye, G. Pan, and M.-S. Alouini, "Earth rotation-aware non-stationary satellite communication systems: Modeling and analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5942–5956, 2021.

[47] P. K. Sharma, B. Yogesh, D. Gupta, and D. I. Kim, "Performance analysis of IoT-based overlay satellite-terrestrial networks under interference," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 985–1001, 2021.

[48] H. Wu, M. He, X. Shen, W. Zhuang, N. Dao, and W. Shi, "Network performance analysis of satellite-terrestrial vehicular network," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16 829–16 844, 2024.

[49] C. Zhou, H. Wu, M. He, W. Wu, N. Cheng, and X. Shen, "Adaptive access mode selection in space-ground integrated vehicular networks," in *Proc. IEEE GLOBECOM*, 2021, pp. 1–6.

[50] L. Yan, X. Fang, L. Hao, and Y. Fang, "Safety-oriented resource allocation for space-ground integrated cloud networks of high-speed railways," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2747–2759, 2020.

[51] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: Integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 47–62, 2018.

[52] H. A. Shah, L. Zhao, and I.-M. Kim, "Joint network control and resource allocation for space-terrestrial integrated network through hierarchical deep actor-critic reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4943–4954, 2021.

[53] X. Zhang, D. Guo, K. An, G. Zheng, S. Chatzinotas, and B. Zhang, "Auction-based multichannel cooperative spectrum sharing in hybrid satellite-terrestrial IoT networks," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 7009–7023, 2020.

[54] J. He, N. Cheng, Z. Yin, H. Zhou, C. Zhou, K. Aldubaikhy, A. Alqasir, and X. Shen, "Load-aware network resource orchestration in LEO satellite network: A gat-based approach," *IEEE Internet Things J.*, 2024.

[55] X. Zhu, C. Jiang, L. Yin, L. Kuang, N. Ge, and J. Lu, "Cooperative multigroup multicast transmission in integrated terrestrial-satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 981–992, 2018.

[56] J. Chu and X. Chen, "Robust design for integrated satellite–terrestrial internet of things," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9072–9083, Jun. 2021.

[57] L. Wang, Y. Wu, H. Zhang, S. Choi, and V. C. Leung, "Resource allocation for NOMA based space-terrestrial satellite networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1065–1075, 2020.

[58] Y.-C. Liang, J. Tan, H. Jia, J. Zhang, and L. Zhao, "Realizing intelligent spectrum management for integrated satellite and terrestrial networks," *J. Commun. Inf. Nets.*, vol. 6, no. 1, pp. 32–43, 2021.

[59] H. Wu, J. Chen, C. Zhou, F. Lyu, N. Zhang, L. Wang, and X. Shen, "Load- and mobility-aware cooperative content delivery in sag integrated vehicular networks," in *Proc. IEEE ICC*, Montreal, Quebec, Canada, Jun. 2021.

[60] Y. Zhu, D. Zhou, M. Sheng, J. Li, and Z. Han, "Stochastic delay analysis for satellite data relay networks with heterogeneous traffic and transmission links," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 156–170, 2021.

[61] Z. Gao, A. Liu, C. Han, and X. Liang, "Non-orthogonal multiple access based average age of information minimization in LEO satellite-terrestrial integrated networks," *IEEE Trans. Green Commun. Netw.*, 2022.

[62] L. Chen, F. Tang, J. Liu, X. Li, Y. Zhu, J. Yu, L. T. Yang, Z. Li, B. Yao, and Y. Yu, "Time-varying resource graph based processing on the way for space-terrestrial integrated vehicle networks," *IEEE Trans. Mobile Comput.*, 2023.

[63] X. Zhang, B. Qian, X. Qin, T. Ma, J. Chen, H. Zhou, and X. Shen, "Cybertwin-assisted mode selection in ultra-dense LEO integrated satellite-terrestrial network," *J. Commun. Inf. Netw.*, vol. 7, no. 4, pp. 360–374, 2022.

[64] J. Li, K. Xue, D. S. Wei, J. Liu, and Y. Zhang, "Energy efficiency and traffic offloading optimization in integrated satellite/terrestrial radio access networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2367–2381, 2020.

[65] M. He, C. Zhou, H. Wu, and X. Shen, "Learning-based cache placement and content delivery for satellite-terrestrial integrated networks," in *Proc. IEEE GLOBECOM*, 2021, pp. 1–6.

[66] C. Jiang and Z. Li, "Decreasing big data application latency in satellite link by caching and peer selection," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2555–2565, 2020.

[67] D. Han, H. Peng, H. Wu, and W. Liao, "Joint cache placement and content delivery in satellite-terrestrial integrated C-RANs," in *Proc. IEEE ICC*. IEEE, 2021, pp. 1–6.

[68] D. Jiang, F. Wang, Z. Lv, S. Mumtaz, S. Al-Rubaye, A. Tsourdos, and O. Dobre, "QoE-aware efficient content distribution scheme for satellite-terrestrial networks," *IEEE Trans. Mobile Comput.*, 2021.

[69] M. He, H. Wu, X. Shen, and W. Zhuang, "Cooperative resource scheduling for environment sensing in satellite-terrestrial vehicular networks," *IEEE Internet Things J.*, 2024.

[70] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, 2019.

[71] X. Zhang, J. Liu, Z. Xiong, Y. Huang, R. Zhang, S. Mao, and Z. Han, "Cost-effective hybrid computation offloading in satellite-terrestrial integrated networks," *IEEE Internet Things J.*, 2024.

[72] D. Han, Q. Ye, H. Peng, W. Wu, H. Wu, W. Liao, and X. Shen, "Two-timescale learning-based task offloading for remote iot in integrated satellite-terrestrial networks," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10 131–10 145, 2023.

[73] C. Zhou, W. Wu, H. He, P. Yang, F. Lyu, N. Cheng, and X. Shen, "Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 2, pp. 911–925, 2020.

[74] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic SFC mapping and scheduling in SDN/NFV-enabled space-air-ground-integrated networks for internet of vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5824–5838, 2021.

[75] D. Chen, C. Yang, P. Gong, L. Chang, J. Shao, Q. Ni, A. Anpalagan, and M. Guizani, "Resource cube: Multi-virtual resource management for integrated satellite-terrestrial

industrial IoT networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11 963–11 974, 2020.

[76] S. Fu, J. Gao, and L. Zhao, "Integrated resource management for terrestrial-satellite systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3256–3266, 2020.

[77] S. Fu, B. Wu, S. Wu, and F. Fang, "Multi-resources management in 6G-oriented terrestrial-satellite network," *China Commun.*, vol. 18, no. 9, pp. 24–36, 2021.

[78] Y. Drif, E. Chaput, E. Lavinal, P. Berthou, B. Tiomela Jou, O. Gremillet, and F. Arnal, "An extensible network slicing framework for satellite integration into 5G," *Int. J. Satellite Commun. Netw.*, vol. 39, no. 4, pp. 339–357, 2021.

[79] H. Wu, J. Chen, C. Zhou, W. Shi, N. Cheng, W. Xu, W. Zhuang, and X. Shen, "Resource management in space-air-ground integrated vehicular networks: SDN control and AI algorithm design," *IEEE Wireless Commun.*, vol. 27, no. 6, pp. 52–60, 2020.

[80] I. Kovacevic, A. S. Shafigh, S. Glisic, B. Lorenzo, and E. Hossain, "Multi-domain network slicing with latency equalization," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2182–2196, 2020.

[81] H. H. Esmat, B. Lorenzo, and W. Shi, "Toward resilient network slicing for satellite-terrestrial edge computing iot," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14 621–14 645, 2023.

[82] T. De Cola and I. Bisio, "QoS optimisation of eMBB services in converged 5G-satellite networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12 098–12 110, 2020.

[83] F. Lyu, P. Yang, H. Wu, C. Zhou, J. Ren, Y. Zhang, and X. Shen, "Service-oriented dynamic resource slicing and optimization for space-air-ground integrated vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[84] M. He, H. Wu, C. Zhou, S. Hu, Z. Tang, and W. Zhuang, "Digital twin-assisted robust and adaptive resource slicing in LEO satellite networks," *arXiv:2404.03025*, 2024.

[85] H. Wu, J. Chen, C. Zhou, J. Li, and X. Shen, "Learning-based joint resource slicing and scheduling in space-terrestrial integrated vehicular networks," *J. Commun. Inf. Nets.*, vol. 6, no. 3, pp. 208–223, 2021.

[86] M. He, H. Wu, C. Zhou, and X. Shen, "Resource slicing with cross-cell coordination in satellite-terrestrial integrated networks," in *Proc. IEEE ICC*, 2024, pp. 2501–2506.

[87] G. Zhou, L. Zhao, G. Zheng, S. Song, J. Zhang, and L. Hanzo, "Multiobjective optimization of space–air–ground-integrated network slicing relying on a pair of central and distributed learning algorithms," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8327–8344, 2024.

[88] L. Chen, Y. Li, C. Huang, Y. Xing, D. Tian, L. Li, Z. Hu, S. Teng, C. Lv, J. Wang *et al.*, "Milestones in autonomous driving and intelligent vehicles—Part I: Control, computing system design, communication, hd map, testing, and human behaviors," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 9, pp. 5831–5847, 2023.

[89] F. Wu, W. Yang, J. Lu, F. Lyu, J. Ren, and Y. Zhang, "RLSS: A reinforcement learning scheme for HD map data source selection in vehicular NDN," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10 777–10 791, 2021.

[90] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 2021.

[91] S. Zhang, H. Luo, J. Li, W. Shi, and X. Shen, "Hierarchical soft slicing to meet multi-dimensional QoS demand in cache-enabled vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2150–2162, 2020.

[92] Y. Tang, N. Cheng, W. Wu, M. Wang, Y. Dai, and X. Shen, "Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3967–3979, 2019.

[93] N. Ramakrishnan and T. Soni, "Network traffic prediction using recurrent neural networks," in *Proc. IEEE ICMLA*, 2018, pp. 187–193.

[94] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent traffic steering in beyond 5g open ran based on lstm traffic prediction," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7727–7742, 2023.

[95] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, 2003.

[96] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014.

[97] J. Zhao, Y. Liu, K. K. Chai, Y. Chen, and M. Elkashlan, "Many-to-many matching with externalities for device-to-device communications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 1, pp. 138–141, 2016.

[98] B. Di, L. Song, and Y. Li, "Sub-channel assignment, power allocation, and user scheduling for non-orthogonal multiple access networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7686–7698, 2016.

[99] Y. Tang and S. Agrawal, "Discretizing continuous action space for on-policy optimization," in *Proc. AAAI Artificial Intell.*, vol. 34, no. 04, 2020, pp. 5981–5988.

[100] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proc. AAAI Artificial Intell.*, vol. 30, no. 1, 2016.

[101] E. Wei, D. Wicke, and S. Luke, "Hierarchical approaches for reinforcement learning in parameterized action space," in *Proc. AAAI Spring Symp.*, 2018, pp. 358–364.

[102] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," in *Proc. IJCAI*, 2019, pp. 2279–2285.

[103] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[104] Z. Lin, Z. Ni, L. Kuang, C. Jiang, and Z. Huang, "Multi-satellite beam hopping based on load balancing and interference avoidance for NGSO satellite communication systems," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 282–295, 2022.

[105] D. Peng, D. He, Y. Li, and Z. Wang, "Integrating terrestrial and satellite multibeam systems toward 6G: Techniques and challenges for interference mitigation," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 24–31, 2022.

[106] A. A. Khalek, C. Caramanis, and R. W. Heath, "Delay-constrained video transmission: Quality-driven resource allocation and scheduling," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 60–75, 2015.

[107] C.-S. Chang, *Performance Guarantees in Communication Networks.* London: Springer-Verlag, 2012.

[108] J. Eckstein and W. Yao, "Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," *RUTCOR Research Reports*, vol. 32, no. 3, p. 44, 2012.

[109] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed joint resource and routing optimization in wireless sensor networks via alternating direction method of multipliers," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5454–5467, 2013.

[110] J. Giesen and S. Laue, "Distributed convex optimization with many convex constraints," *arXiv:1610.02967*, 2018.

[111] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1226–1252, 2021.

[112] C. Yu, X. Yang, J. Gao, J. Chen, Y. Li, J. Liu, Y. Xiang, R. Huang *et al.*, "Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration," in *Proc. AAMAS*, 2023, pp. 1107–1115.

[113] A. Kanervisto, C. Scheller, and V. Hautamäki, "Action space shaping in deep reinforcement learning," in *Proc. IEEE CoG*, 2020, pp. 479–486.

[114] Q. Chen, Y. Xie, S. Guo, J. Bai, and Q. Shu, "Sensing system of environmental perception technologies for driverless vehicle: A review of state of the art and challenges," *Sensors and Actuators A: Phys.*, vol. 319, p. 112566, 2021.

[115] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1852–1864, 2020.

[116] X. Liu, T. Ma, Z. Tang, X. Qin, H. Zhou, and X. Shen, "Ultrastar: A lightweight simulator of ultra-dense LEO satellite constellation networking for 6G," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 632–645, 2023.

[117] A. Van Etten, "Satellite imagery multiscale rapid detection with windowed networks," in *Proc. IEEE WACV*, 2019, pp. 735–743.

[118] 3GPP, "Study on architecture aspects for using satellite access in 5G," 3GPP, Technical Report (TR) 23.737, 2021, version 17.2.0, Release 17.

[119] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.

[120] ETSI, "Intelligent transport systems (ITS); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service," TR 302 637-2, ETSI, Tech. Rep., 2019.

[121] H. Teng, Z. Li, K. Cao, S. Long, S. Guo, and A. Liu, "Game theoretical task offloading for profit maximization in mobile edge computing," *IEEE Trans. Mobile Comput.*, 2022.

[122] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "Trafficgan: Network-scale deep traffic prediction with generative adversarial nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 219–230, 2019.

[123] M. Cygan, Ł. Kowalik, and M. Wykurz, "Exponential-time approximation of weighted set cover," *Inf. Process. Lett.*, vol. 109, no. 16, pp. 957–961, 2009.

[124] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, 2002.

[125] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 77–97, 2009.

[126] A. Bogomolnaia and M. O. Jackson, "The stability of hedonic coalition structures," *Games Econ. Behav.*, vol. 38, no. 2, pp. 201–230, 2002.

[127] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synth. Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[128] I. Leyva-Mayorga, B. Soret, B. Matthiesen, M. Röper, D. Wübben, A. Dekorsy, and P. Popovski, "Non-geostationary orbit constellation design for global connectivity," in *Non-Geostationary Satellite Communications Systems*. Institution of Engineering and Technology, 2023, pp. 237–267.

[129] T. T. de Almeida, L. de Carvalho Gomes, F. M. Ortiz, J. G. R. Júnior, and L. H. M. Costa, "Comparative analysis of a vehicular safety application in NS-3 and veins," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 620–629, 2020.

[130] K. Tan, L. Feng, G. Dán, and M. Törngren, "Decentralized convex optimization for joint task offloading and resource allocation of vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 13 226–13 241, 2022.

[131] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, 2017.

[132] G. A. Jones and J. M. Jones, *Elementary number theory*. London: Springer-Verlag, 1998.

APPENDICES

Appendix A

Proof of Proposition 1

For each CAV, to satisfy the sensing result freshness requirement T^{th}, consecutive sensing data, either from the LEO satellite or connected RSUs, should be generated with an interval less than T^{th}. For the case when CAV u cannot consistently connect to RSUs throughout the entire time interval, ensuring the reliability requirement of environment sensing necessitates a satellite sensing interval of $T_u^{\text{S}} < T^{\text{th}}$. This guarantees the continuous reception of sensing results for the CAV. The sensing offset can be any time less than the sensing interval T_u^{S}. If CAV u can consistently connect to both the LEO satellite and RSUs during the time interval, there are three cases. The first two cases indicate that the freshness requirement can be satisfied through sensing from one source, either the LEO satellite or RSUs for an individual CAV. In these cases, either the sensing interval from the LEO satellite or RSUs should be less than T^{th}. For the third case, if the environment sensing requirement cannot be met with a single sensing source, then both the satellite and RSUs should be involved for environment sensing, and the following relation should be satisfied

$$\frac{1}{T_u^{\text{S}}} + \frac{1}{T_u^{\text{T}}} > \frac{1}{T^{\text{th}}} \text{ and } T_u^{\text{S}}, T_u^{\text{T}} \in [T^{\text{th}}, 2T^{\text{th}}). \tag{A.1}$$

However, Eq. (A.1) serves as a necessary but not sufficient condition for ensuring the sensing freshness requirement. To avoid potentially outdated sensing results, a proper satellite sensing time offset ψ_u^{S} should be determined. Let jT_u^{T} and $kT_u^{\text{S}} + \psi_u^{\text{S}}$ denote the time stamps of j-th RSU sensing and k-th satellite sensing, respectively. To satisfy the freshness requirement, the sensing time stamp of the satellite should not fall within any of the following ranges:

- $kT_u^{\mathrm{S}} + \psi_u^{\mathrm{S}} \in [jT_u^{\mathrm{T}}, (j+1)T_u^{\mathrm{T}} - T^{\mathrm{th}}]$: This indicates that after receiving the k-th satellite sensing result, CAV u's sensing result become outdated before the $(j+1)$-th RSU sensing starts;
- $kT_u^{\mathrm{S}} + \psi_u^{\mathrm{S}} \in [jT_u^{\mathrm{T}} + T^{\mathrm{th}}, (j+1)T_u^{\mathrm{T}}]$: This represents that after receiving j-th RSU sensing result, CAV u's sensing result will be outdated before the k-th satellite sensing starts;
- $kT_u^{\mathrm{S}} + \psi_u^{\mathrm{S}} \in [(j+1)T_u^{\mathrm{T}} - T_u^{\mathrm{S}}, jT_u^{\mathrm{T}}]$: This implies that the k-th satellite sensing begins before the j-th RSU sensing and the $k+1$-th satellite sensing will begin after the $j+1$-th RSU sensing. In other words, the j-th RSU sensing result will be outdated since $T_u^{\mathrm{T}} \geq T^{\mathrm{th}}$.

Combining the above three cases, we can find that the freshness requirement cannot be satisfied if

$$\begin{aligned} &kT_u^{\mathrm{S}} + \psi_u^{\mathrm{S}} = jT_u^{\mathrm{T}} + g, \ \exists j, k \in \mathbb{N}, \\ &g \in \mathcal{L} = [0, T_u^{\mathrm{T}} - T^{\mathrm{th}}] \cup [T^{\mathrm{th}}, T_u^{\mathrm{T}}] \cup [T_u^{\mathrm{T}} - T_u^{\mathrm{S}}, 0], \end{aligned} \tag{A.2}$$

which can be rewritten as

$$jT_u^{\mathrm{T}} - kT_u^{\mathrm{S}} = \psi_u^{\mathrm{S}} - g, \ \exists j, k \in \mathbb{N}. \tag{A.3}$$

According to Bézout's identity [132], there exists values $j, k \in \mathbb{N}$ such that

$$|jT_u^{\mathrm{T}} - kT_u^{\mathrm{S}}| = G, \tag{A.4}$$

where $G = \mathbf{gcd}(T_u^{\mathrm{T}}, T_u^{\mathrm{S}})$ is the greatest common divisor of two values. In this case, Eq. (A.2) is satisfied indicating the violation of the freshness requirement if

$$\exists (\psi_u^{\mathrm{S}} - g) \bmod G = 0. \tag{A.5}$$

In addition, the relation of two sensing intervals has the following proposition.

Proposition 3. If two different sensing interval $T_u^{\mathrm{S}} \neq T_u^{\mathrm{T}}$, $|T_u^{\mathrm{S}} - T_u^{\mathrm{T}}| \geq G$.

Proof. According to the definition of the greatest common divisor, there exist integers a and b, so that

$$T_u^{\mathrm{S}} = aG, T_u^{\mathrm{T}} = bG, \tag{A.6}$$

where a and b are coprime integers. In this case, we have

$$|T_u^{\mathrm{S}} - T_u^{\mathrm{T}}| = |a - b||G|. \tag{A.7}$$

Since $a \neq b$, $|a - b| \geq 1$. Therefore, $|T_u^{\mathrm{S}} - T_u^{\mathrm{T}}| \geq G$. ■

To this end, we can divide the values of the sensing interval pair into the following three cases:

- $T_u^{\mathrm{S}} > T_u^{\mathrm{T}}$: consider one interval $[T_u^{\mathrm{T}} - T_u^{\mathrm{S}}, 0] \subset \mathcal{L}$ where the range is given by $T_u^{\mathrm{S}} - T_u^{\mathrm{T}}$. Since $T_u^{\mathrm{S}} - T_u^{\mathrm{T}} \geq G$ according to **Proposition 3**, regardless of the value of ψ_u^{S}, there exists $g \in \mathcal{L}$ to satisfy Eq. (A.5);

- $T_u^{\mathrm{S}} < T_u^{\mathrm{T}}$: consider the interval $[0, T_u^{\mathrm{T}} - T^{\mathrm{th}}] \cup [T^{\mathrm{th}}, T_u^{\mathrm{T}}] = \mathcal{L}$ where the range is given by $(T_u^{\mathrm{T}} - T^{\mathrm{th}} - 0) + (T_u^{\mathrm{T}} - T^{\mathrm{th}}) = 2(T_u^{\mathrm{T}} - T^{\mathrm{th}})$. Since $T_u^{\mathrm{T}} - T_u^{\mathrm{S}} \geq G$ and $T_u^{\mathrm{T}} > T_u^{\mathrm{S}} \geq T^{\mathrm{th}}$, we can have $2(T_u^{\mathrm{T}} - T^{\mathrm{th}}) \geq G$. In this case, there exists $g \in \mathcal{L}$ to satisfy Eq. (A.5);

- $T_u^{\mathrm{S}} = T_u^{\mathrm{T}}$: consider that $[0, T_u^{\mathrm{T}} - T^{\mathrm{th}}] \cup [T^{\mathrm{th}}, T_u^{\mathrm{T}}] = \mathcal{L}$, similarly, the range of the intervals is given by $2(T_u^{\mathrm{T}} - T^{\mathrm{th}})$. Since $2T^{\mathrm{th}} > T_u^{\mathrm{T}} \geq T^{\mathrm{th}}$, we can have $2(T_u^{\mathrm{T}} - T^{\mathrm{th}}) < T_u^{\mathrm{T}} = T_u^{\mathrm{S}} = G$. Therefore, there exist feasible offset values such that Eq. (A.5) cannot be satisfied, given by

$$\psi_u^{\mathrm{S}} \in \hat{\psi}_u^{\mathrm{S}} = \{l\kappa \,|\, l\kappa \in (T_u^{\mathrm{T}} - T^{\mathrm{th}}, T^{\mathrm{th}}), \forall l \in \mathbb{N}\}. \tag{A.8}$$