Wireless Intelligent Distributed Consensus System for Autonomous Driving

Zongyao Li

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

October 2024

Declaration

University of Glasgow
College of Science & Engineering
Statement of Originality

Name: Zongyao Li
Registration Number: xxxxxxxxx

I certify that the thesis presented here for examination for a PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

The copyright of this thesis rests with the author. No quotation from it is permitted without full acknowledgement.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that if any issues are raised regarding good research practice based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature: ....
Date: ....2025.5.19..............

Abstract

The rapid advancements in embedded processing, sensing, artificial intelligence (AI), and communication technologies have accelerated the adoption of connected and autonomous systems (CAS). However, as devices in CAS become more intelligent and autonomous, their application scenarios—such as autonomous driving—are growing increasingly complex and dynamic. To enable intelligent, connected, and autonomous (ICA) nodes to engage in deeper deliberation and mutual understanding, joint decision-making has emerged as an effective solution. Joint decision-making is a process where multiple autonomous agents collectively analyze information, deliberate, and reach consensus to make unified decisions that align with common goals. However, traditional joint decision-making approaches face significant challenges when applied to the stringent demands of modern CAS. For instance, centralized decision-making (CDM) offers streamlined processes and high consistency but suffers from limitations like single points of failure (SPOF), scalability issues, and dependence on centralized infrastructure. By contrast, the decentralized nature of distributed decision-making (DDM) enhances scalability and system reliability, leveraging the intelligence of individual nodes to achieve collective intelligence, making it a promising alternative. In this context, distributed consensus (DC) protocols as a key element in distributed systems are essential to enabling DDM, with features such as data consistency and fault tolerance drawing significant research attention in recent years. This thesis focuses on the application, optimization, and development of wireless DC protocols to enable ICA nodes in CAS, with a particular focus on autonomous driving, to achieve robust and expressive joint decision-making.

First, the study proposes Intelligent Distributed Consensus (IDC) and introduces the first IDC protocol, Intelligent-Raft, which builds upon the traditional Raft algorithm. Additionally, to facilitate the deployment of IDC in practical CAS environments, the study introduces Wireless Intelligent Distributed Consensus System (WIDCS) which leverages distributed wireless communication combined with the Intelligent-Raft algorithm to enable ICA nodes to make collective joint-decisions and ensure fault tolerance. A practical

hardware module of WIDCS is implemented using microcontroller-based systems, which is named 'AIR-RAFT'. To validate the feasibility and effectiveness of WIDCS, we undertake research and evaluations within an autonomous driving scenario, specifically at uncontrolled intersections, utilizing both mathematical modeling and practical scenario testing. Numerical and experimental results, in good alignment, demonstrate that WIDCS substantially improves autonomous driving safety.

Second, this study enhances WIDCS by incorporating the functions of ad hoc network formation, management, and dismissal, improving its ability to provide better data consistency and joint decision-making services for ICA nodes. Additionally, we have developed the second-generation WIDCS module, RaBee, which enables distributed nodes to achieve Intelligent-Raft consensus via a ZigBee-based ad hoc network. In addition, we develope mathematical probability models to evaluate and compare the reliability of centralized decision-making systems and WIDCS. Employing autonomous driving in on-ramp merging as a case study, we further formulated a mathematical model to assess the safety of Autonomous Vehicles (AVs) under different decision-making frameworks. By integrating the RaBee module with AV, we conduct safety tests in practical on-ramp scenarios, and the results demonstrate that WIDCS notably enhances AV safety, indicating substantial potential for future CAS.

Third, this study proposes a novel IDC protocol, Converging-Raft, which leverages the collective intelligence of all nodes to make globally optimal joint decisions—a capability not present in Intelligent-Raft. To enhance the adaptability, we propose the Heterogeneous Intelligent Joint Decision System (HIntS), an architecture which integrates CDM, Intelligent-Raft, and Converging-Raft within a hybrid network combining ad hoc and cellular networks. Our self-developed hardware module at the core of HIntS, '5G-MInd', is designed to verify the system's feasibility and performance in practical experiments. We develop a mathematical model to analyze and compare the reliability and latency of HIntS under different working modes and validate these findings through joint-decision experiments using 5G-MInd modules. Our quantitative and qualitative results demonstrate the advantages and characteristics of different combinations of joint-decision mechanisms and network structures. These findings highlight HIntS's adaptability to complex, dynamic environments and provide critical guidance for the practical deployment of future wireless joint-decision mechanisms.

List of Publications

- **Z. Li**, H. Wu, Z. Dong, L. Zhang, and M. A. Imran, "Wireless intelligent distributed consensus enabled autonomous vehicles' cooperation at uncontrolled intersection," IEEE Transactions on Intelligent Vehicles, under review, 2025.
- **Z. Li**, L. Zhang, and M. A. Imran, "HetIJDS: Heterogeneous intelligent joint decision system for intelligent, connected and autonomous applications," IEEE Transactions on Systems, Man, and Cybernetics: Systems, under review, 2025.
- **Z. Li**, L. Zhang, and M. A. Imran, "Intelligent distributed consensus for connected vehicles: Models, implementation and testing," IEEE Internet of Things Journal, under review, 2025.
- **Z. Li**, L. Zhang, X. Zhang, and M. Imran, "Design and implementation of a Raft based wireless consensus system for autonomous driving," in GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, 2022, pp. 3736–3741.
- Y.Gou, R. Wang, **Z. Li**, M. A. Imran, and L. Zhang, "Clustered hierarchical distributed federated learning," in ICC 2022-IEEE International Conference on Communications. IEEE, 2022, pp. 177–182.
- H.Xu, Z. Li, **Z. Li**, X. Zhang, Y. Sun, and L. Zhang, "Metaverse native communication: A blockchain and spectrum prospective," in 2022 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2022, pp. 7–12.
- Z. Dong, H. Wu, **Z. Li**, D. Mi, O. Popoola, and L. Zhang, "Trustworthy VANET: Hierarchical dag-based blockchain solution with proof of reputation consensus algorithm," in 2023 IEEE International Conference on Blockchain (Blockchain). IEEE, 2023, pp. 127–132.

- Z. Dong, **Z. Li**, L. Zhang, and M. A. Imran, “Modeling of Probabilistic PBFT Consensus under Various Network Topologies and Its Applications to VANET,” in INFOCOM 2025-2025 IEEE International Conference on Computer Communications, under review. IEEE, 2025.

Contents

List of Tables

List of Figures

List of Acronyms

3GPP	3rd Generation Partnership Project
5G	Fifth-Generation Technology Standard for Broadband cellular networks
ACC	Adaptive Cruise Control
AD	Autonomous Driving
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
API	Application Programming Interface
AV	Autonomous Vehicle
BFT	Byzantine Fault Tolerance
CACC	Cooperative Adaptive Cruise Control
CAN	Controller Area Network
CAS	Connected and Autonomous System
CAV	Connected and Autonomous Vehicle
CB	Cooperative Broadcast
CDF	Cumulative Distribution Function
CDM	Centralized Decision-Making
CFT	Crash Fault Tolerance
CPS	Cyber-Physical Systems
DAG	Directed acyclic graph
DC	Distributed Consensus
DDM	Distributed Decision-Making
DL	Downlink
DNN	Deep Neural Network
EC	Electoral College
FSMC	Flexible Static Memory Controller
FSPL	Free Space Path Loss
GB	Gossip-Broadcasting

GNSS	Global Navigation Satellite System
HAL	Hardware Abstraction Layer
HCI	Human-Computer Interaction
HIntS	Heterogeneous Intelligent Joint Decision System
ICA	Intelligent, connected and autonomous
ID	IDentity
IDC	Intelligent Distributed Consensus
IIC	Inter-Integrated Circuit
IIoT	Industrial Internet of Things
IoT	Internet of Things
IoV	Internet of Vehicles
ISM	Industrial Scientific Medical
ITS	Intelligent Transportation Systems
MAC	Media Access Control
MCU	Micro-Controller Unit
ML	Machine Learning
mMTC	Massive Machine-Type Communications
MQTT	Message Queuing Telemetry Transport
MS	Merging Sequence
NSA	Non-Standalone
LoRa	Long Range Radio
LTE	Long Term Evolution
P2P	Peer to Peer
PBFT	Practical Byzantine Fault Tolerance
PCDA	Perception Collection Decision Action
PDF	probability density function
PF	Predecessor-Follower
PICA	Perception Initiative Consensus Action
PoE	Proof of Eligibility
PoS	Proof of Stake
PoW	Proof of Work
PPP	Poisson point process
PSO	particle swarm optimization
OBU	On-Board Unit
OLED	Organic Light-Emitting Diode

RAM	Random Access Memory
R2C	Random Representative Consensus
RC	Referendum Consensus
RF	Radio Frequency
ROM	Read-Only Memory
RSU	Roadside Unit
SA	Standalone
SD	Secure Digital
SED	Selective Edge Decision
SID	Sequence Identification Number
SINR	Signal-to-interference-plus-noise ratio
SPI	Serial Peripheral Interface
SPOF	Single Point of Failure
SQP	sequential quadratic programming
UART	Universal Asynchronous Receiver/Transmitter
UI	User Interface
UL	Uplink
URLLC	Ultra-Reliable Low Latency Communications
VANET	Vehicular Ad Hoc Network
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
WBN	Wireless Blockchain Networks
WDC	Wireless Distributed Consensus
WIC	Wait Insertion Count
WIDCS	Wireless Intelligent Distributed Consensus System
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Networks

Acknowledgements

On this special night, I have finally completed my doctoral dissertation, and my heart is filled with a mix of emotions—being deeply moved, excited, nostalgic, and content. First, I would like to express my heartfelt gratitude to my supervisor, Prof. Lei Zhang. His offer of a direct doctoral path was like a ray of light, dispelling doubts and made me determined to live up to this expectation. His meticulous guidance, encouragement, and belief in my potential gave me the confidence to find my own rhythm in research and gradually grow into a capable scholar. During moments of frustration or helplessness, his calls of encouragement lifted me from difficult times. I am incredibly fortunate to have had Prof. Lei Zhang as my supervisor, and I know his guidance and kindness will stay with me throughout my life. At the same time, I must also thank Prof. Muhammad Ali Imran and my colleagues for their supports on my research in the University of Glasgow.

Next, I want to thank my family - my parents, grandparents, and all my relatives, who gave me unconditional love and support and helped me step by step from a broken family to today's achievements. I am especially grateful to my uncle, who, like a godfather, has offered me comprehensive guidance and support in every aspect of life—spirit, economy, thought, emotion, and beyond. I am also thankful to my seniors, classmates, and juniors, whose camaraderie enriched my academic journey, bringing joy and warmth to my days of research. Additionally, I am grateful to the friends I've met in this foreign country over the past four years; each small affirmation and act of kindness has provided a sense of belonging and comfort far from home.

Finally, I wish to acknowledge those who have brought me pain and trials because these experiences have made me stronger, more confident, self-accepting, and optimistic. Recalling the lonely, cold, and challenging nights, I recognize that growth and comfort rarely coexist. Perhaps, at some point, I learned to accept the pain, knowing that it has shaped who I am today. I believe firmly that these hardships will eventually lead me to the bright redemption. As Her Majesty the Queen once said, "We should take comfort that while we may have more still to endure, better days will return."

Chapter 1

Introduction

1.1 Background of Connected and Autonomous System

Driven by the advancements of Industry 4.0, artificial intelligence (AI), and robotics, autonomous devices are becoming increasingly sophisticated, capable of independently executing and handling complex tasks with greater intelligence and precision. Autonomous driving is a typical representative of current automation and intelligent systems. Modern autonomous vehicles (AVs) are now equipped with an array of 20 to 40 specialized sensors, designed to gather comprehensive data about the environment and support either human drivers or fully autonomous systems in the decision-making process [3]. These sensors, which include radar, lidar, cameras, and ultrasonic detectors, monitor aspects such as vehicle speed, proximity to other objects, road conditions, and pedestrian activity, creating a rich data landscape for navigating complex environments.

However, current onboard systems primarily support localized and reactive decision-making, focusing on the individual vehicle's perspective rather than a collaborative, networked understanding of traffic flow. This limitation means that a vehicle's intentions remain hidden from surrounding traffic, lacking the kind of predictive, coordinated decision-making that could improve overall safety. Moreover, without a shared understanding, any malfunction in sensor algorithms or hardware can lead to catastrophic failures. For example, when sensor systems misinterpret or miss critical cues due to algorithmic errors or hardware faults, the AVs may fail to recognize potential hazards, posing severe risks to all road users.

A striking example of these limitations occurred recently when a Cruise autonomous vehicle failed to detect a pedestrian in its path, resulting in a tragic accident [4]. Due to an undetected object detection failure, the vehicle collided with and dragged the pedestrian for approximately 20 feet (6.1 meters), ultimately stopping with a tire positioned on the individual's leg [4]. Such incidents underscore the urgent need for enhanced communication between autonomous devices and their environments, enabling these systems to anticipate hazards collectively, reduce reaction times, and avoid accidents stemming from isolated sensor failures. This example highlights the importance of an integrated approach across all autonomous systems, where devices operate not just independently but in coordinated networks, ensuring safer and more resilient collaboration across various applications, from transportation to industrial automation.

To address the aforementioned challenges and elevate the operational efficiency of autonomous systems, the concept of connected and autonomous systems (CAS) is emerging as a powerful solution. CAS envisions a network of interconnected intelligent nodes, each capable of not only autonomous operation but also dynamic collaboration with other nodes. By facilitating information exchange, CAS enables these autonomous agents—whether vehicles, drones, or other robotic entities—to move beyond isolated, reactive decision-making and toward a model of proactive coordination. This shift holds immense potential for enhancing safety, resource efficiency, and cooperations across a wide array of applications, from transportation to industrial automation. Over the past two decades, rapid advancements in embedded processing, sensing, AI, and communication technology have propelled the widespread adoption of novel CAS, notably in applications such as ambient assisted living, entertainment, logistics optimization, energy management, and industrial automation [5–9]. For instance, smart grids and energy management systems equipped with sensors and Internet of Things (IoT) devices can dynamically optimize energy distribution in response to real-time load data, significantly enhancing energy efficiency and minimizing waste.

The advantages of CAS lie in its ability to create a shared, synchronized understanding of the environment. With constant data exchange among distributed nodes, CAS allows each device to "see" beyond its own sensory limitations, leveraging information from surrounding entities to create a richer, more accurate picture of the world. For example, in traffic scenarios, vehicles within a CAS network can anticipate each other's movements and intentions, enabling more precise navigation and reducing the likelihood of accidents caused by unexpected actions. Additionally, CAS can streamline operations in sectors like logistics, where autonomous robots coordinate in real-time to optimize storage, retrieval, and transportation of goods.

As CAS devices become increasingly interconnected and intelligent, their applications span highly complex and dynamic environments. Yet, while these systems are capable of exchanging large volumes of data, relying solely on basic data exchange often reveals significant limitations. In many scenarios, data sharing alone is not enough to meet the sophisticated, real-time demands of CAS applications. When devices in a network are limited to transmitting isolated data points—such as speed, position, or object detection—they often lack the contextual understanding required to anticipate and respond collectively to emergent conditions. This reliance on fragmented information can lead to suboptimal or conflicting decisions, especially in scenarios where coordinated action is critical for safety, efficiency, or operational success.

For instance, in an interconnected AVs network, vehicles can share information about their current speeds, trajectories, and nearby obstacles. However, without an additional layer of collective processing, each vehicle must still independently interpret these data, resulting in decisions that may overlook broader, cooperative objectives. In a multi-vehicle overtaking scenario, exchanging positional data alone may not provide sufficient information for optimal maneuvering. It is beneficial for each vehicle to not only be aware of its own position but also to infer the intentions and reactions of others to enhance the chances of smooth and safe coordination. While basic data exchange can help vehicles detect each other's presence, the absence of a joint decision-making mechanism may limit their ability to achieve effective and timely collective coordination during complex maneuvers.

This gap underscores the critical need for joint decision-making within CAS. Building on basic data exchange, joint decision-making enables CAS nodes to process shared information as a cohesive unit. This approach allows interconnected devices to synthesize data into a unified understanding, evaluate multiple perspectives, and reach consensus on actions in real time. For example, AVs can collectively deliberate on the best timing and path for an overtaking maneuver, weighing variables such as speed alignment, lane availability, and optimal spacing. Through this collaborative approach, vehicles not only react to each other's data but actively align their intentions, leading to safer, more efficient actions that individual data sharing alone could not achieve.

The importance of joint decision-making extends beyond autonomous vehicles to diverse CAS applications, such as industrial robotics, drone swarms, and intelligent energy grids. In each of these domains, joint decision-making transforms data exchange from a passive process into an active, coordinated strategy, allowing systems to adapt dynamically to complex environments. By facilitating proactive collaboration, joint decision-making builds on the foundation of data sharing to enhance the resilience, safety, and overall intelligence of CAS networks. This evolution from simple data exchange to collective decision-making marks a pivotal shift, positioning CAS to address increasingly challenging demands with a level of coordination previously unattainable.

With the increasing complexity, dynamism, and criticality of CAS application scenarios, the requirements for joint decision-making are becoming increasingly stringent. First, the synchronization of decisions across nodes in CAS is important. Robust synchronization on decisions and information ensures consistent, reliable and efficient operations, particularly in time-sensitive or coordination-intensive scenarios. However, in real-world operations, network delays and variations in data processing times can result in information asynchrony between nodes, which may lead to task incoordination. For instance, warehouses equipped with interconnected robotic systems and automated guided vehicles (AGVs) can optimize inventory management. These robots autonomously navigate aisles, pick and pack products, and restock inventory, all in synchronization with the warehouse centralized management system. This real-time synchronization not only enhances the movement consistency and safety of the swarm but also greatly improves efficiency. Asynchronous information will lead to incoordination between machines, which bring unnecessary safety threats and economic losses.

Moreover, the demand for high fault tolerance, reflecting robustness in CAS, is increasing rapidly due to the rising system fault rates driven by more dynamic and complex applications. Fault tolerance guarantee that even if some nodes experiences hardware limitations or software/algorithm errors such as incorrect AI judgments, the collective decision-making can still produce accurate global decisions. For example, in intelligent transportation systems (ITS), vehicles rely on accurate sensor data to prevent collisions and optimize traffic flow. If a sensor or algorithm fails—such as a radar misreading the distance or a camera malfunctioning due to adverse weather—the vehicle may transmit incorrect data to the system. Without a fault-tolerance mechanism, the direct exchange of faulty speed or position information could lead to unsafe driving decisions, such as premature acceleration or unnecessary emergency braking. This not only exacerbates traffic congestion but also increases the risk of chain-reaction accidents.

As mentioned above, CAS has an increasing demand for joint decision-making, and also requires support for synchronization, consistency, fault tolerance, etc. Therefore, it is essential to optimize and enhance the existing CAS system to enable intelligent and autonomous nodes to effectively handle increasingly complex and demanding application environments.

1.2 Methodologies and Motivations

1.2.1 Centralized versus Distributed Joint Decision-Making

Joint decision-making in CAS are typically structured around two main frameworks: centralized or distributed. In a centralized approach, nodes forward their collected data to a central control hub, which then makes decisions and issues commands for implementation. The centralized decision-making (CDM) system is widely used across various industries and fields due to its ability to simplify the decision-making and management process through a centralized structure. The single source of decision ensure unified control across all nodes in the network, which enhance system consistency by preventing individual nodes from making isolated decisions that could disrupt overall operations. In addition, CDM enables more efficient resource allocation and optimization. The central control unit can assess the needs and resource status of all nodes, making informed decisions to minimize resource waste. Furthermore, in rapidly changing environments, the central control can rapidly adjust strategies and issue instructions, allowing the system to adapt more quickly to external changes.

However, the scalability of CDM is increasingly strained by the growing number and diversity of connected devices and nodes, alongside escalating cybersecurity risks, pushing these systems towards their operational thresholds [10]. For example, centralized architectures are notably vulnerable to certain risks, such as the single point of failure (SPOF) and increased susceptibility to cybersecurity threats, especially when sensors operate in open and dynamic environments. In addition to security, privacy concerns emerge as a critical issue in centralized systems, given the potential aggregation of sensitive data at a central node. Furthermore, CDM restricts nodes to synchronizing information and decisions solely with the central node, potentially constraining the overall system performance by the weakest connection in the network. This can result in parameters such as latency and reliability falling below (or above, in case of latency) expected values, potentially resulting in accidents and the loss of human lives [11]. Moreover, the financial implications of establishing and maintaining a centralized infrastructure can be significant, potentially precluding its use or failing to meet the rigorous latency and reliability demands of certain applications. For example, the deployment of sensors for critical real-time decision-making processes necessitates connections with a central cloud/edge via 5G ultra-reliable and low latency communications (URLLC), a capability that might not be accessible in some rural areas.

As next-generation mobile applications evolve towards a distributed topology, distributed decision-making (DDM) leverages the distribution of decision authority among multiple intelligent nodes to overcome inherent limitations in centralized systems. The stability of DDM is not reliant on a single node, as data processing and decision-making are distributed across multiple nodes. This decentralized structure enhances system robustness and mitigates the risk of SPOF, ensuring the reliable completion of decisions. Additionally, each node processes only the information relevant to it, which helps safeguard user privacy and limits centralized access to sensitive data.

Moreover, DDM systems are not limited by the processing power or spectrum resources of a central node, as each node can independently process information and respond. This allows the system to flexibly scale by adjusting the number of nodes as needed. For example, in an IoT environment, as the number of devices increases, the system can seamlessly expand, with each node handling its own data and decisions without relying on a central hub, thereby enhancing both scalability and adaptability.

Another key advantage of DDM is its ability to achieve true swarm intelligence. In this architecture, each node actively contributes to the decision-making process by utilizing its own perception and computing resources, rather than relying solely on the intelligence of a central node. This approach often results in more precise and comprehensive decisions, as it integrates data and insights from multiple nodes, offering a broader and more diverse perspective.

Through these characteristics, DDM not only enhances system robustness and scalability, but also improves the transparency and intelligence of the decision-making process. This makes it particularly well-suited for CAS that require high reliability and support a large number of devices. In this context, Distributed Consensus (DC) plays a vital role in enabling DDM and has garnered significant attention in recent research [10, 12–16]. At its core, DC refers to the process by which multiple distributed entities achieve agreement on a shared state or decision, despite potential communication delays, faults, or conflicting objectives. Standardized distributed protocols, such as consensus algorithms, provide the foundation for these agreements, enabling robust and efficient operations even in dynamic and uncertain conditions [17]. Given its relevance, this research focuses on leveraging DC mechanisms to address key challenges in CAS. By exploring its application to DDM scenarios, this work aims to enhance the safety, coordination, and performance of Intelligent, Connected, and Autonomous (ICA) systems operating in complex environments.

1.2.2 Distributed Consensus: An Overview

Over the past few decades, DC technology has evolved from its early roots in computer science, including distributed databases and computational problems, to a critical component of modern distributed networks. Early research focused on achieving consensus in systems prone to failure, with notable developments such as Lamport's Paxos algorithm [18] (proposed in 1989, published in 1998) and Barbara Liskov's Viewstamped Replication [19]. As research advanced, consensus mechanisms were adapted to more complex networks and failure models, particularly addressing Byzantine faults, where nodes may act maliciously. The Practical Byzantine Fault Tolerance (PBFT) algorithm became a significant solution for these challenges. In 2008, the advent of Bitcoin ushered in a new era for distributed consensus. Bitcoin, along with its underlying blockchain technology, demonstrated the feasibility of secure, decentralized transactions via the Proof of Work (PoW) mechanism, without the need for a central trusted authority [20]. This innovation not only popularized cryptocurrencies but also spurred the application of blockchain technology across various fields.

The core principle of DC is to ensure consistency in distributed systems. Consistency refers to the ability of multiple nodes to agree on a data value or operation, ensuring that all non-faulty nodes maintain a uniform understanding of the system's state. This requires nodes to reach agreement before updating or operating on data, guaranteeing the accuracy and synchronization of information. While consistency is the primary focus, fault tolerance is a critical characteristic that enables the system to maintain functionality and performance despite node failures or abnormal behavior. Fault tolerance is typically achieved through redundancy, where multiple nodes store identical data or perform the same functions, allowing other nodes to take over in the event of failures, thereby preserving system continuity and data integrity.

The consensus process typically follows four key steps: Proposing, Negotiating, Agreeing, and Committing [1, 2, 21]. In the proposing phase, a node (often referred to as the leader or proposer) proposes a value that requires agreement. During the negotiation phase, other nodes review the proposal and provide feedback, which may include acceptance, requests for clarification, or rejection. In the agreeing phase, all non-faulty nodes work towards reaching consensus on the proposed value. Finally, the committing phase involves formally recording the agreed-upon value into the system.

DC technology is widely applied in key areas such as distributed databases, distributed computing, IoT, blockchain and cryptocurrency. These fields leverage DC to enhance data consistency, system reliability, and security. As the core technology of distributed systems, the application prospects of distributed consensus in modern distributed networks will become increasingly broad.

1.2.3 Distributed Consensus: Byzantine Fault Tolerance

Over the years, researchers have developed various algorithms and protocols for distributed consensus, including Paxos [21], Raft [2], and Practical Byzantine Fault Tolerance (PBFT) [1]. These have been applied across distributed systems, such as databases [22], blockchain [23], and cloud computing [24–26], to ensure system consistency and reliability. These protocols can also function as internal mechanisms, enabling nodes to make joint-decisions based on collected information. In DC protocols, each participant can transmit and receive commands to update the state of replicas, adhering to fault-tolerant protocols [27]. Most DC protocols focus on addressing two types of faults: Byzantine faults and crash faults.

Byzantine failures refer to malicious behaviors introduced by adversaries, such as contradictory commands, communication interruptions, or intentional delays of critical messages [28]. These failures may result from software bugs, hardware malfunctions, or malicious attacks, causing nodes to provide incorrect information or disrupt system operations. The concept originates from the Byzantine Generals Problem, where participants in a distributed system must agree on a strategy, even if some actors are unreliable or malicious. The term 'Byzantine' refers to this classic problem in distributed computing, where consensus must be achieved despite the presence of dishonest or faulty participants.

To address this, Byzantine fault tolerance (BFT) mechanisms ensure that the system continues to operate correctly and reliably despite the presence of Byzantine faulty nodes. BFT protocols like PBFT and HotStuff BFT [29, 30] maintain a consistent sequence of node states through quorum intersection, ensuring the correct propagation of information across the system even in the presence of Byzantine attack. These protocols often use redundancy, replicating and processing data across multiple nodes, and rely on consensus mechanisms that allow nodes to agree on the system's correct state.

PBFT is a consensus algorithm designed to function efficiently in asynchronous distributed systems and handle Byzantine faults. The workflow of PBFT is shown in Fig. 1.1. PBFT categorizes nodes into two roles: a primary node, which is responsible for proposing the order of client requests, and multiple backup nodes, which validate and reach agreement on this proposed order. The algorithm operates in three main phases: Pre-prepare, Prepare, and Commit. In the Pre-prepare phase, the primary node, upon receiving a client request, assigns a sequence number to the request and broadcasts a Pre-prepare message containing this sequence number and the request details to all backup nodes. The purpose of this phase is to ensure that all nodes receive the same initial proposal for processing the client request. In the Prepare phase, the backup nodes, after validating the Pre-prepare message, broadcast a Prepare message to all nodes, signaling their agreement with the proposed sequence. The Prepare message serves as confirmation that the node is ready to process the request. Each node must receive at least $2f+1$ Prepare messages (where f is the maximum number of faulty nodes the system can tolerate) before proceeding to the next phase, ensuring a majority of non-faulty nodes have agreed on the proposal. In the Commit phase, once a node receives $2f+1$ valid Prepare messages, it sends a Commit message to all other nodes. This phase confirms that at least two-thirds of the nodes have validated the request and are ready to commit the proposed order. Upon receiving $2f+1$ Commit messages, the nodes finalize the request, ensuring that the request has been accepted by the majority and can be safely executed. This multi-phase process guarantees that even in the presence of Byzantine nodes, the system can reach consensus and maintain consistency across all non-faulty nodes.

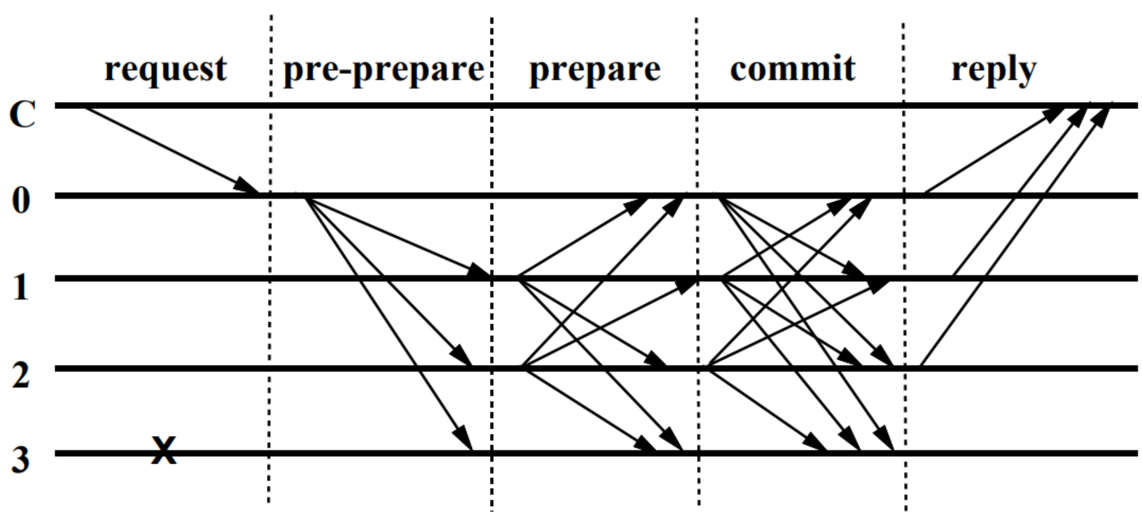

Figure 1.1: Communication scheme of PBFT [1]

An integral feature of PBFT is its view change mechanism, which ensures system continuity when the primary node is faulty or unresponsive. If backup nodes detect that the primary has failed to send expected messages within a defined timeout period, they initiate a view change by broadcasting View-change messages to all other nodes. When a quorum of $2f+1$ nodes agrees that the primary is faulty, a new primary is elected from the backup nodes following a predefined sequence. The new primary then resumes the consensus process, ensuring the system remains operational and no progress is lost. This mechanism is vital for maintaining liveness and reliability in PBFT, allowing the system to recover from primary failures efficiently.

PBFT's strength lies in its safety and robustness. Safety is ensured by requiring that transactions are only committed after validation by at least $2f+1$ nodes, preventing inconsistent decisions across the system. Robustness is achieved through its Byzantine fault tolerance, allowing the system to handle up to $(n-1)/3$ faulty nodes where n represents the total number of nodes, with the view change mechanism further ensuring reliable recovery from primary failures. These features make PBFT highly effective in permissioned blockchain platforms, providing fast transaction finality and robust fault tolerance. PBFT's practical design makes it highly suitable for real-world deployment, offering both efficiency and reliability in distributed systems.

1.2.4 Distributed Consensus: Crash Fault Tolerance

Although byzantine tolerance enhances security by addressing malicious faults in systems, its application is not always necessary, particularly in environments where nodes are authenticated and can be trusted. The complexity of BFT algorithms can significantly increase communication overhead, decrease system throughput, and hinder scalability, making it less practical for certain scenarios [17].

Crash faults refer to the sudden and unexpected failure of a node, often caused by hardware malfunctions, software bugs, or network issues, which disrupts the system's normal operation. In trusted systems that do not account for Byzantine faults, the risk posed by node crashes and link transmission failures is significant, as they directly threaten the system's reliability. To address this, crash fault tolerance (CFT) ensures that a distributed system can maintain functionality even when nodes or communication links fail. This is typically achieved through replication, where data and processes are duplicated across multiple nodes, allowing the system to continue operating seamlessly in the event of a failure. If a node crashes, the remaining nodes take over, ensuring the system remains functional while the failed node is replaced or restarted. CFT protocols, such as Raft and Paxos [31], are specifically designed to manage reliable state replication, thereby preventing system breakdowns caused by node crashes or communication failures.

Paxos is the first CFT consensus algorithm in the field of distributed systems, proposed by Leslie Lamport in 1998 [18]. The main goal of Paxos is to ensure that a network of distributed, potentially unreliable (in the sense of crashing or going offline), agents can agree on a single value (a consensus). This is particularly crucial for databases and other distributed storage systems, where they need to ensure data consistency across all nodes. Paxos operates through a series of proposals, where each proposal is issued by a proposer, voted on by acceptors, and disseminated by learners. The fundamental guarantee of Paxos is that if a value is chosen, then every future proposal will also suggest that value. The protocol is structured such that it ensures that only one value is selected and that all participants can eventually be aware of that value, despite the potential for failures and message losses [32]. However, Paxos can be quite complex to implement correctly due to the intricate interplay of its components. There are simpler versions like Multi-Paxos [33] and variations [34, 35] developed to handle more complex real-world scenarios, such as Google's Chubby [36] and Apache Zookeeper [37]. These derivatives seek to enhance the efficiency and simplicity of the base Paxos algorithm, yet they continue to exhibit considerable complexity.

Raft, as a crash fault tolerance (CFT) algorithm, is widely implemented in private, trusted distributed systems to handle replica failures and ensure system reliability [2]. Compared to the traditional CFT protocol Paxos [21], Raft is known for its simplicity and ease of implementation, making it more accessible for real-world applications [38–40]. A Raft-enabled distributed network consists of consensus replicas, which include a leader and multiple followers. The protocol operates in two primary stages: Leader election and log replication, as illustrated in Fig. 1.2 and Fig. 1.3 respectively.

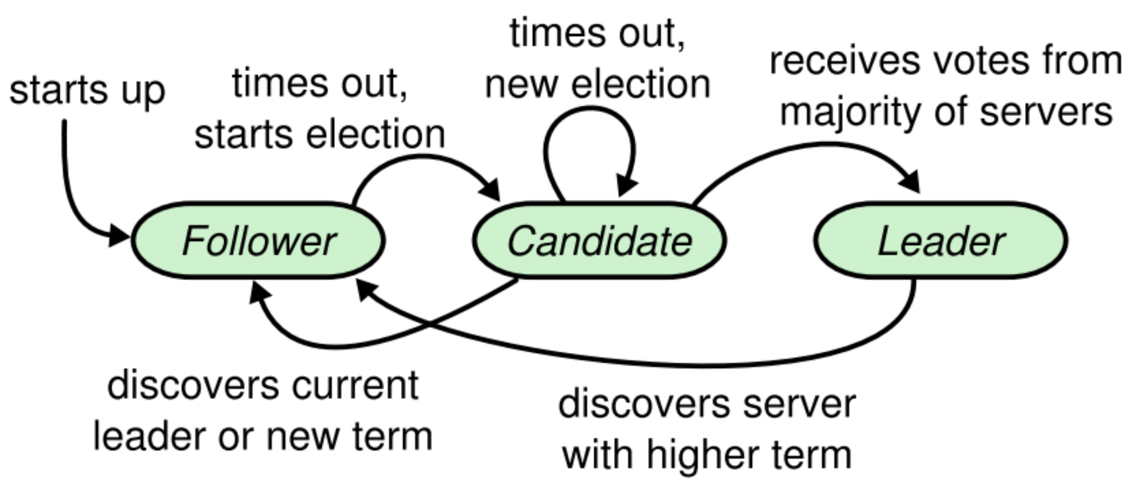

Figure 1.2: State change of Raft [2]

In Raft's consensus algorithm, the Leader election process is initiated when a Follower node detects the absence of heartbeat messages from the current leader within a predefined timeout interval. Upon this detection, the Follower transitions into the Candidate state and commences a new election term. The Candidate node votes for itself and broadcasts a RequestVote message to all other nodes in the system. Followers that receive this message

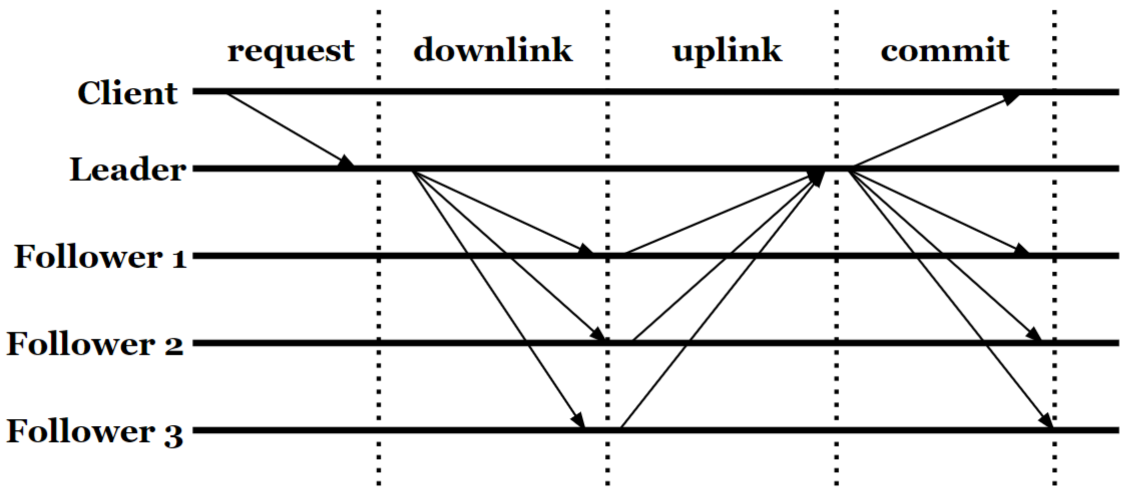

Figure 1.3: Log replication of Raft

determine whether to grant their vote based on two critical conditions: (1) whether they have already voted in the current election term, and (2) whether the Candidate's log is at least as up-to-date as their own. If both conditions are met, the Follower grants its vote to the Candidate. The Candidate is elected as the new Leader if it secures votes from a majority of the nodes in the system, representing more than 50% of the network. Once elected, the new Leader assumes control by broadcasting heartbeat messages to maintain its leadership role. If the Candidate fails to achieve a majority, it reverts to the Follower state, and the election process may repeat until a Leader is successfully chosen.

Following the Leader election, the log replication phase begins. The Leader is responsible for aggregating client requests into log entries and ensuring that these entries are replicated across all Followers through continuous downlink transmission. Once a follower receives the replicated log entries, it sends an acknowledgment to the Leader via an uplink unicast message, confirming the successful reception. When a majority of followers (over 50%) have acknowledged the receipt of log entries, the Leader commits the log and instructs the followers to either execute the confirmed commands or update their state in accordance with the leader's current term. This log replication ensures consistency across all nodes in the distributed system, safeguarding the system's correctness. In practice, the selection of the Leader may be influenced by the network performance of the nodes, as a node with more reliable connectivity may be more suited to assume the leadership role, thereby enhancing the overall system throughput and reducing latency, ensuring robust and efficient operation in real-time environments.

Raft's modular deployment, independent functions, low communication complexity, and high throughput have made it appealing to the practical system. It has been extensively studied for validation [41], enhancement [39], and application [42]. Therefore, this study will focus on the development of the Raft algorithm and its application in practical wireless scenarios, emphasizing its effectiveness in ensuring consensus and fault tolerance in CAS.

1.2.5 Distributed Consensus for CAS

As application environments become increasingly complex and dynamic, the need for joint decision-making in CAS continues to grow. As outlined in Section 1.2.1, DDM leverages its decentralized nature to eliminate reliance on the reliability of central nodes, enhancing system scalability and strengthening data privacy. Effective cooperative decision-making requires standardized information exchange between nodes and clear consensus reaching conditions. Additionally, all nodes should achieve a synchronized and consistent understanding of each joint-decision. Therefore, DC is recognized as an effective solution and has garnered increasing attention due to its features closely aligning with these requirements.

The DC mechanism provides ICA nodes with a structured framework to facilitate joint decision-making through well-defined processes. Each node first forms an initial opinion based on its own observations and local data, then exchanges information with other nodes via a preset protocol to reach joint-decisions. This structured approach ensures that even with a large number of nodes and complex data, the system can maintain orderly decision-making without falling into chaos. Additionally, DC can enable joint decision-making by integrating information from multiple nodes, providing a global perspective that mitigates the limitations of a single node's local view. As environments become more dynamic, DC allows for continuous adjustments through ongoing joint decision-making, ensuring the system can adapt flexibly to changing conditions.

In addition to enabling joint decision-making, DC allows devices to achieve consistent consensus on data or decisions, which is a critical function in CAS. DC employs a specific protocol to ensure that each node can compare, synchronize, and integrate the different data it receives through multiple rounds of communication, ultimately achieving system-wide data consistency. This consistency is essential for ensuring overall coordination and security within the system. Moreover, synchronization enhances the transparency and traceability of system operations, allowing any participating node to verify past actions and decisions—an important feature for scenarios that demand high levels of trust and auditability.

DC provides robust fault tolerance for both data consensus and joint decision-making.

Its decentralized nature ensures that system stability is not reliant on a single control point, thereby mitigating the risk of single points of failure and significantly enhancing the attack robustness of CAS. Additionally, even if some nodes in CAS fail, the consensus mechanism can still successfully reach a final agreement. Once failed nodes recover, they can restore missing data by synchronizing with other nodes via the same consensus process, thereby enhancing overall system resilience. Furthermore, in the event of a coordinator or leader node failure, DC protocols support dynamic leader election, allowing the system to promptly designate a new consensus coordinator and maintain operational continuity.

However, in wireless environments where CAS operates, communication links may suffer from failures due to channel fading or spectrum interference [43, 44]. These factors introduce significant challenges for the implementation of DC, as reliable message exchange is critical to ensuring timely and accurate joint decision-making. Unlike wired networks, wireless channels are inherently unstable and can lead to packet loss, variable latency, and degraded synchronization among nodes—all of which directly impact the convergence, reliability, latency and scalability of DC protocols. Therefore, to effectively realize joint decision-making in CAS, it is essential to conduct a detailed analysis and performance evaluation of DC mechanisms under realistic wireless conditions. This includes examining how wireless-specific characteristics influence key performance metrics such as consensus latency, fault tolerance, and communication overhead, thereby guiding the design of more reliable and adaptive consensus strategies for CAS.

In summary, DC mechanism provides CAS with a robust, efficient, and secure framework for joint decision-making and system operation, significantly improving overall functionality and efficiency. These advantages make DC an ideal solution for managing complex interactions and data synchronization challenges in CAS.

The exploration of DC in CAS spans a wide range of applications and poses numerous challenges. Autonomous driving, as a key example of CAS, relies on advanced artificial intelligence to process vast amounts of sensor data, enabling real-time individual decision-making and precise automatic control. With the continuous advancement of V2X (Vehicle-to-Everything) technology, wireless communication between AVs, pedestrians, and traffic management systems enhances situational awareness, contributing to safer and more efficient driving environments.

However, autonomous driving scenarios are highly dynamic, with the speed and relative positions of AVs constantly changing. As autonomous driving technology becomes more intelligent and automated, the complexity of these scenarios will also increase. To address these challenges, joint decision-making has emerged as a promising approach for

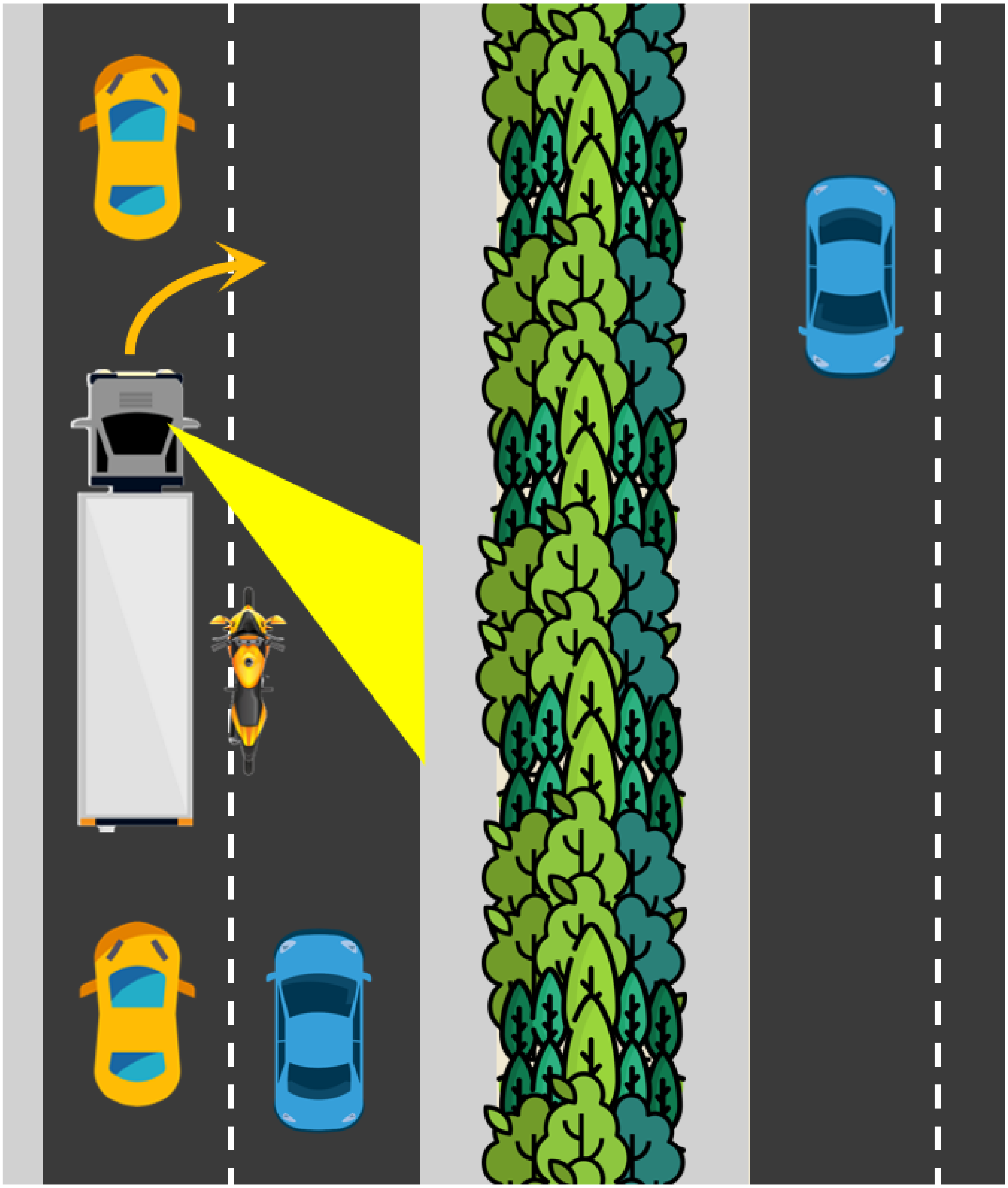

Figure 1.4: Truck lane change

enabling AVs to formulate more accurate, safer, and efficient traffic strategies in dynamic and evolving environments. By sharing real-time data and collaboratively assessing complex traffic conditions, AVs can leverage collective intelligence to formulate optimal action plans. Additionally, joint decision-making helps to prevent conflicts and enhances overall traffic flow efficiency.

In addition, data consistency among vehicles is equally critical in autonomous driving. Inconsistent data between vehicles could lead to conflicting decisions made simultaneously within the same scenario, thereby increasing the risk of accidents. To enhance data consistency, autonomous driving systems should incorporate fault tolerance and recovery mechanisms. Fault tolerance ensures that when a vehicle experiences hardware sensor failures, software malfunctions, or communication breakdowns, the system can quickly identify and isolate the affected vehicle. Meanwhile, decision-making and operations continue through functioning vehicles to maintain system safety and coordination. Faulty nodes can be restored and synchronized with accurate data to prevent decision errors caused by data inconsistencies.

The characteristics and demands of autonomous driving underscore the significance and potential of DC mechanisms in this field. The strengths of DC can effectively address the requirements of autonomous driving, including joint decision-making, coordinated consistency, and system robustness, making it a suitable solution for ensuring safety and efficiency in dynamic driving environments.

Fig. 1.4 illustrates a specific scenario in which a truck attempts to change lanes, posing a significant risk due to the presence of an undetected motorcycle in its blind spot. In this situation, relying solely on the perception and decision-making capabilities of a single vehicle may result in an incomplete or inaccurate understanding of the surrounding environment, potentially leading to safety risks.

Joint decision-making, supported by DC mechanisms, is crucial in such scenarios. In the illustrated case, if the truck can communicate with other vehicles and infrastructure, its lane change proposal would be rejected by vehicles aware of the motorcycle's presence. The truck would reach a consensus with the surrounding vehicles before executing the lane change, thereby preventing a potential accident. Specifically, when the truck submits a lane change request, the system broadcasts the request to nearby vehicles and roadside units (RSUs). These nodes evaluate the request by checking for vehicles in the truck's blind spot, assessing the motorcycle's speed and position, and considering other relevant information. Using DC protocols like Raft or Paxos, the nodes quickly agree on whether the lane change is safe. The truck proceeds with the lane change only if a majority of nodes confirm it is risk-free. This collaborative approach and information-sharing mechanism significantly enhance the accuracy and safety of decision-making.

In summary, DC mechanisms offer a promising approach to enhancing the ability of AVs to navigate complex traffic scenarios more effectively through communication and collaboration. By supporting joint decision-making and information consistency, DC can optimize the decision-making capabilities of individual vehicles and improve the overall coordination and safety of the traffic system, showcasing significant potential for future intelligent transportation systems (ITS).This research aims to explore the practical implementation of DC protocols as an optimization strategy within autonomous driving scenarios.

1.3 Objectives and Original Contributions

To fully harness the potential of DC mechanisms in CAS, further research is needed to develop efficient algorithms and protocols that enhance joint decision-making among increasingly intelligent and autonomous devices. Moreover, rigorous mathematical analysis is critical for evaluating key performance metrics—such as reliability, latency, and computational complexity—under wireless communication constraints, which significantly impact the performance of DC. Such analysis enables iterative refinement and optimization of the proposed solutions.

In addition, this thesis aims to enhance the efficiency and safety of autonomous driving by applying both traditional and self-developed DC protocols to autonomous driving scenarios. The primary objective is to enhance reliability and coordination in autonomous driving cooperation through joint decision-making, thereby mitigating misjudgments arising from the limitations of single-node perspectives. This approach also improve the collective vehicle intelligence, enabling more effective responses to complex and dynamic environments. Through an in-depth analysis of DC applications in autonomous driving, this thesis seeks to offer practical insights and forward-looking guidance for the future deployment of these technologies, while establishing a theoretical foundation for safe joint decision-making and system optimization in challenging scenarios.

This thesis also aims to design and implement a comprehensive hardware module to validate the effectiveness and performance of the wireless DC. Another core goal is to deploy this hardware system on an experimental autonomous driving platform, testing the potential of wireless DC in enhancing safety and efficiency through real-world application. The ultimate goal is to establish a robust theoretical and experimental foundation for the practical implementation of wireless DC technology, supporting its future deployment in large-scale autonomous driving systems.

The original contributions in this thesis can be summarized as follows:

- This research introduces the Intelligent Distributed Consensus (IDC), developed from traditional DC protocols, and presents the first IDC protocol, Intelligent-Raft. IDC allows nodes within a network to perform more intelligent evaluations and improve the quality of joint decision-making processes. Furthermore, the study proposes an innovative system architecture called the Wireless-Intelligent-Distributed-Consensus System (WIDCS), which integrates wireless communication with the Intelligent-Raft algorithm. The research also involves the design and implementation of the first WIDCS hardware module, AIR-RAFT, on an embedded platform. Additionally, a novel and secure traffic management scheme for AVs at uncontrolled intersections is proposed, utilizing joint decision-making facilitated by WIDCS. Simulation analysis and experimental results confirm the feasibility of WIDCS, demonstrating significant improvements in the safety of AVs navigating uncontrolled intersections.

- This thesis addresses the limitations encountered with the first-generation WIDCS hardware system, AIR-RAFT, and introduces an evolved second-generation system named RaBee. RaBee incorporates more advanced network management capabilities and richer hardware resources, offering significant improvements in consensus reliability, scalability, and latency compared to AIR-RAFT. Additionally, this research examines the safety of autonomous driving in on-ramp merging scenarios and proposes an efficient traffic coordination solution based on WIDCS. By employing mathematical models, we analyze the probability of AVs safely navigating on-ramp merging scenarios under three conditions: without a communication system, with a centralized decision-making system, and with WIDCS. The results demonstrate that WIDCS significantly enhances safety. This finding is further validated through practical tests using the RaBee system and experimental AVs, with optimization results closely aligning with theoretical predictions.

- This thesis introduces a second IDC protocol, Converging-Raft, which enables intelligent nodes not only to engage in the joint decision-making process but also to reach a globally optimal solution through convergence and discussion. Recognizing that different application scenarios have varying requirements for joint decision-making and information consensus, this research proposes a unified system that integrates multiple consensus mechanisms to adapt to specific needs. To achieve this, we develop the Heterogeneous Intelligent Joint Decision System (HIntS), an architecture that combines Centralized Decision-Making (CDM), Intelligent-Raft, and Converging-Raft within a hybrid ad hoc and cellular network structure. Furthermore, the thesis designs a complete hardware implementation of HIntS, named 5G-MInd. The performance of HIntS is analyzed in five operational modes using both quantitative and qualitative methods, assessing key factors such as reliability, latency, global optimality, scalability, transmission coverage, and fault tolerance. This analysis provides essential insights for the future deployment of wireless joint decision-making systems.

1.4 Thesis Outline

The remainder of this thesis is structured as follows. Chapter 2 provides a comprehensive literature review, beginning with key studies on Cyber-Physical Systems, followed by an overview of its subclass, CAV. It then offers a systematic review of DC in wireless environments, followed by a discussion of state-of-the-art DC applications in autonomous driving. Finally, Chapter 2 identifies current research gaps, establishing the foundation for the contributions presented in this thesis.

Chapter 3 builds upon the works "Wireless intelligent distributed consensus enabled autonomous vehicles' cooperation at uncontrolled intersection" (first publication in the **List of Publications**) and "Design and implementation of a Raft-based wireless consensus system for autonomous driving" (third publication in the **List of Publications**). The chapter begins by introducing the new IDC protocol, Intelligent-Raft, detailing its workflow and underlying principles. It further presents WIDCS, which leverages Intelligent-Raft to provide consistent joint decision-making services within CAS. The design and implementation of AIR-RAFT, a practical hardware module for WIDCS, are also discussed. A secure passage scheme utilizing WIDCS for autonomous driving at uncontrolled intersections is then proposed. The chapter also derives a detailed mathematical model for AV behavior at uncontrolled intersections, including collision risk assessments for autonomous driving, both with and without WIDCS. For the proposed scheme, Chapter 3 showcases AVs consensus experiments using AIR-RAFT, along with a thorough analysis of the experimental data. Finally, numerical results are presented and conclusions are drawn.

Chapter 4 builds on the work "Intelligent distributed consensus for connected vehicles:

Models, implementation and testing” (fourth publication in the **List of Publications**). It begins by introducing the enhanced ad hoc networking functions added to the original WIDCS, encompassing network formation, management, and dissolution. Following this, the chapter details the design, architecture, usage, and workflow of RaBee, the second-generation WIDCS hardware module. A secure passage scheme based on WIDCS with networking capabilities is then proposed for autonomous driving in on-ramp merging scenarios. Mathematical models for AV safe passage in on-ramp merging—considering scenarios without communication, with CDM, and with WIDCS—are developed, and simulation results are analyzed. Chapter 4 further includes practical AV experiments using RaBee, accompanied by an in-depth analysis of the experimental data. Finally, comparisons are made, and key conclusions are presented.

Chapter 5 builds on the work “HetIJDS: Heterogeneous intelligent joint decision system for intelligent, connected, and autonomous applications” (second publication in the **List of Publications**). It begins with a detailed overview of Converging-Raft and HIntS, concluding with results that highlight the advantages and characteristics of HIntS under various working modes. Following this, the chapter outlines the design of the practical HIntS module, 5G-MInd. A mathematical model for assessing the reliability and latency of HIntS across five distinct working modes is then introduced, with subsequent sections providing simulations and data analysis based on this model. Chapter 5 further presents practical experiments conducted with 5G-MInd modules to validate the performance of HIntS in each of its five modes. Finally, the conclusion is given in the end.

Finally, Chapter 6 concludes the thesis and discusses the future research trends associated with this topic.

Chapter 2

Literature Review

CAS represent a specialized and rapidly evolving subclass of Cyber-Physical Systems (CPS), characterized by their ability to perceive complex environments, communicate with other agents, and make autonomous decisions in real time. CPS, as the broader foundational concept, refers to the integration of computational elements with physical processes, typically through a tightly coupled loop of sensing, computation, and actuation. The CPS paradigm has enabled the development of intelligent, adaptive, and resilient systems across domains such as industrial automation, healthcare, transportation, and robotics.

CAS builds upon this foundation by incorporating advanced features such as distributed intelligence, inter-agent communication, and real-time cooperative behavior. In contexts like autonomous driving, these systems are expected not only to process local sensor data and make self-contained decisions but also to collaborate with surrounding agents and infrastructure through wireless communication to achieve safe and efficient outcomes. The transition from traditional CPS to CAS introduces new challenges, particularly in ensuring coordination, robustness, and scalability in highly dynamic and uncertain environments.

Given that CAS inherits its core principles, system architectures, and many enabling technologies from CPS, it is essential to first review the existing body of research on CPS to establish a comprehensive understanding of its underlying methodologies. This includes an examination of its definitions and architectural frameworks, the significance and impact of CPS in modern engineered systems, its defining characteristics, and the key applied domains. Therefore, before addressing literature specific to CAS—particularly in the context of DC and autonomous driving—this review begins with a focused overview of relevant CPS research to ground the subsequent discussion in a broader theoretical and technological context.

2.1 Overview of Cyber-Physical Systems

Many recent review papers have explored the utilization of Cyber-Physical Systems (CPS) across various emerging application domains. [45] conducted a comprehensive survey of 77 relevant studies, categorizing them into ten application areas, including agriculture, education, energy management, environmental monitoring, medical systems, process control, security, smart cities and homes, smart manufacturing, and transportation. [46] provided a concise overview of CPS, highlighting its applications and associated challenges, and emphasized its potential to enhance convenience, comfort, and safety in everyday life. [47] examined existing definitions and application domains of CPS, underlining the critical role of human factors in system design. They also proposed a domain-independent definition and metamodel, offering a theoretical foundation for CPS research. In more domain-specific studies, [48] reviewed CPS applications in healthcare, classifying key components and methods necessary for implementation, while [49] focused on the integration of CPS in the chemical industry.

In addition to application-oriented studies, CPS architecture and system characteristics are widely addressed in the literature. [50] provided an in-depth analysis of CPS architecture, proposing a general framework based on service-oriented architecture (SOA). This approach emphasizes flexibility in integrating heterogeneous devices, supporting real-time control, and ensuring system security. Similarly, [51] conducted a systematic review of human-cyber-physical systems (HCPS), clarifying fundamental concepts and identifying key challenges in establishing a rigorous systems engineering foundation. These challenges include complex heterogeneity, the absence of suitable abstraction methods, dynamic black-box integration, and multifaceted functional demands. The authors proposed four targeted research directions to address these issues: abstraction and computational theory, architectural modeling methods, model specification and verification, and software-defined HCPS frameworks. [52] further surveyed the state-of-the-art in CPS, detailing intrinsic features such as autonomy, stability, robustness, efficiency, scalability, trustworthiness, consistency, and high precision, alongside methodologies and design challenges. [53] introduced the core definition of CPS, highlighting its integration of computing, communication, and control technologies, and its strengths in real-time performance, safety, and reliability. Their work also identified critical research issues related to system modeling, information processing, and software architecture. Additionally, [49] examined the intersection of artificial intelligence (AI) and CPS in the chemical industry, underscoring AI's role in enabling cognitive capabilities that allow CPS to interact with the physical environment in a more autonomous and adaptive manner.

Topics such as control and optimization architectures, as well as Digital Twins (DT), are increasingly recognized as essential components of CPS. [54] proposed a new classification framework and analytical matrix for identifying contemporary CPS security threats, using quantitative methods to categorize them by attack type, impact, intent, and event category. [55] conducted a comprehensive review of the role of collective intelligence in industrial CPS, examining its key characteristics, enabling technologies, and primary application areas, with illustrative examples drawn from automobile assembly lines. The study also identified and summarized the major challenges involved in implementing collective intelligence in industrial contexts. [from cite Carolina Villarreal Lozano] proposes an intelligent CPS framework that incorporates mechanisms for autonomy and self-adaptation, while also improving bandwidth efficiency and reducing energy consumption. The system is designed to offer fault prediction, enhanced autonomy, and adaptability to dynamic conditions. [56] investigated smart manufacturing from the perspective of self-organization, presenting a systematic literature review that summarizes current enabling technologies, implementation strategies, and future research directions. [57] explored the application of multi-agent system (MAS) technologies within cyber-physical production systems (CPPS), employing a SWOT analysis and expert validation to assess the strengths and limitations of MAS-based architectures. [58] conducted a comprehensive survey spanning 14 critical CPS application areas, offering a holistic overview of the current research landscape, identifying key challenges—particularly in security and data privacy—and outlining promising future directions. In parallel, DT-based research has gained momentum, with [59] highlighting the use of DTs in automotive CPS for enhanced adaptability, shorter development cycles, and improved scalability. [60] discussed the interconnection between CPS and DT components in smart manufacturing, while [61] emphasized the role of DTs in strengthening the resilience and security of CPS and advancing the vision of Industry 4.0.

In summary, the CPS literature provides a solid foundation in system architecture, key technologies, and application domains. These insights are essential for understanding how computation and control integrate with physical systems. Building on this foundation, the next section focuses on CAS, which extend CPS by emphasizing autonomy, connectivity, and cooperative decision-making in dynamic environments.

2.2 Overview of CAS

Among the various applications of CAS, Connected and Autonomous Vehicles (CAVs) represent one of the most prominent and actively studied domains. As a natural evolution of CPS, CAVs integrate on-board sensing, computation, and communication to enable real-time perception, information exchange, and vehicles cooperation in complex and dynamic environment.

Specifically, the United States Department of Transportation (USDOT) defines three primary types of connected vehicle communication for CAVs: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) [62]. The term "V2X" broadly encompasses communication with various entities, including passengers, other vehicles, onboard devices, cloud services [63], wireless sensors, and navigation systems. As illustrated in Fig. 2.1, these communication modes enable CAVs to interact through V2V, V2I, and infrastructure-to-infrastructure (I2I) channels. In this context, a roadside unit (RSU) may represent traffic signals or other active traffic management sensors. I2I refers to communication between infrastructure elements, such as consecutive signalized intersections, to support functions like green wave coordination [64].

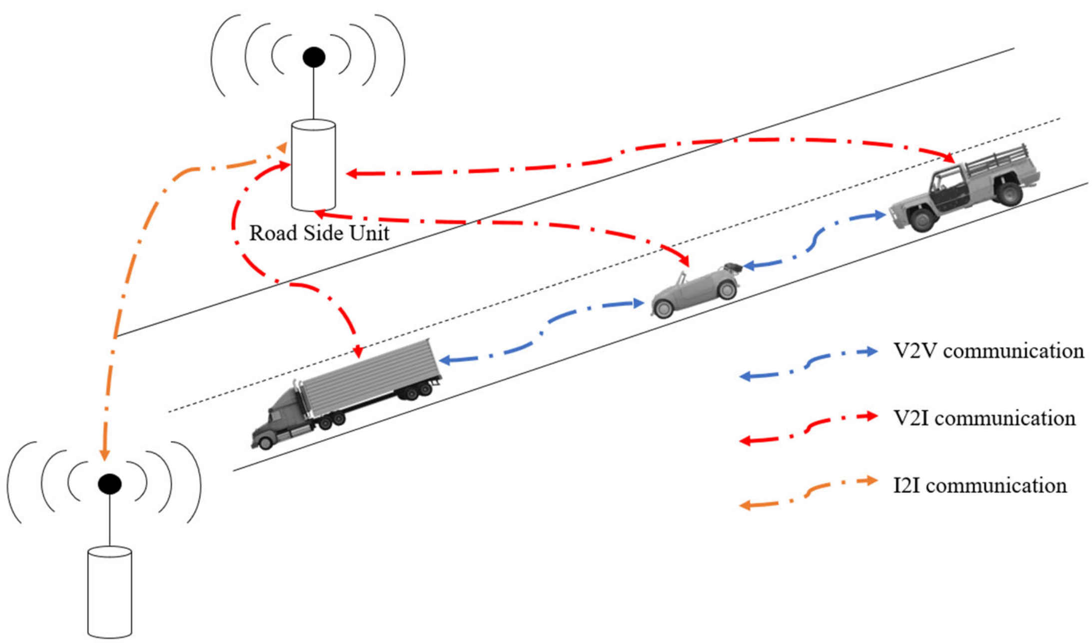

Figure 2.1: Connected vehicle communications

Several ongoing research initiatives in the UK aim to explore the application of V2X communication in the context of autonomous vehicles. For example, the UKCITE project focuses on establishing a real-time testing environment for connected autonomous vehicles by deploying V2X technologies along more than 40 miles of urban roads, dual carriageways, and motorways across Coventry and Warwickshire [65]. The i-MOTORS project is dedicated to developing a vehicular cloud computing platform that integrates vehicle-generated data with environmental information to produce dynamic maps and real-time hazard alerts [66]. The anticipated outcomes include (i) reduced fuel consumption and travel time through real-time traffic-aware route planning, and (ii) enhanced road safety via car platooning. Similarly, the G-ACTIVE project targets fuel efficiency improvements for a range of drivetrain architectures—conventional, electric, and hybrid—by utilizing off-board data such as traffic conditions and signal timings to jointly optimize energy management and vehicle speed [67]. Lastly, the CARMA project seeks to develop and evaluate cooperative automated driving technologies built upon distributed control systems and supported by ultra-low latency, high-reliability cloud infrastructure [68].

There are also many studies using various technologies to improve the efficiency of CAVs in junctions, roundabouts, and interchanges. Various studies have explored strategies to improve intersection performance, focusing on objectives such as reducing traffic delays, enhancing junction throughput, and mitigating congestion. For instance, [69] proposed a time-independent trajectory optimization method for connected and autonomous vehicles under reservation-based junction control, aiming to minimize group evacuation time and maximize intersection efficiency. Similarly, [70] applied dynamic programming and a heuristic approach known as the least extra time (SET) method. Their work compared the effectiveness of several algorithms—including genetic, branch-and-bound, heuristic, and dynamic programming—against adaptive control and traditional fixed-cycle signal systems. The results indicated that their proposed strategy significantly reduced evacuation time, queue length, and vehicle waiting time. In another study, [71] introduced a temporal delay Petri net (TdPN)-based control strategy within a cooperative vehicle-infrastructure system. Simulation results demonstrated that, under high traffic flow conditions (exceeding 1200 vehicles per hour), the TdPN approach outperformed conventional signal control in terms of delay, average queue length, stop time, and vehicle speed.

Applications of advanced intersection control strategies have been proposed to tackle real-time coordination challenges involving multiple vehicles and lanes. One approach integrates ant colony optimization and discrete methods to develop an Autonomous Intersection Management system that accounts for individual vehicle behaviors and junction conditions, showing improvements in throughput, queue length, evacuation time, and delay [72]. Game-theoretic applications, inspired by the chicken game, have also been used to manage CACC-enabled vehicles at unsignalized intersections, enabling real-time communication with a centralized agent and offering a viable alternative to traditional control mechanisms [73]. Reservation-based scheduling strategies, such as PriorFIFO and its extension csPrior-FIFO, model heterogeneous vehicles and centralized schedulers within a unified framework, achieving better scheduling performance and lower latency than first-come-first-served methods [74, 75]. Furthermore, time-sensitive programming has been applied to address real-time data transmission (RTD) challenges, demonstrating superior efficiency under high traffic flow compared to conventional Autonomous Intersection Management (AIM) systems [76].

One of the primary objectives of AIM is to enhance intersection safety. To achieve this, various strategies have been proposed, focusing on subgoals such as collision avoidance and conflict resolution. Model Predictive Control (MPC) has been employed to ensure collision-free traffic flow through intersections, with simulations conducted using VISSIM and CarSim platforms for validation [77]. In a similar effort, a cooperative driving strategy was developed to reduce intersection-related crashes by enabling vehicles to navigate through decentralized local optimizations, ensuring safe passage without centralized coordination [78]. Another approach introduced a rule-based control method that determines optimal vehicle sequencing and braking actions based on real-time collision detection [79]. This speed control framework helps prevent accidents, clarify vehicle priority, and facilitate safe traversal of unsignalized intersections.

To mitigate conflicts among AVs at unsignalized intersections, a cooperative strategy was proposed that employs a cost function to determine optimal vehicle actions upon detecting potential conflicts [77]. In a related effort, a centralized Model Predictive Control (MPC) framework was introduced to manage AV trajectories and prevent collisions at intersections [80]. The problem was formulated as a convex quadratic program in spatial coordinates, enabling optimal path planning while incorporating penalized time gaps to enhance safety under sensor uncertainties. To further improve intersection safety, a real-time intersection supervisor based on mixed-integer quadratic programming (MIQP) was developed, capable of overriding vehicle commands when necessary to ensure safe operation [79]. Addressing the coordination challenge from a distributed perspective, a parallelizable optimization method—the augmented Lagrangian-based alternating direction inexact Newton (ALADIN) algorithm—was proposed to achieve efficient multi-vehicle coordination at intersections [81].

Despite recent advancements, CAV systems still face critical challenges such as decision conflicts among agents, uncertainty in communication, and safety risks in complex traffic scenarios. These limitations reveal the need for more advanced joint decision-making mechanisms that enable robust, coordinated behavior among distributed CAVs. Addressing these challenges requires moving beyond conventional control strategies toward decentralized, consensus-driven approaches capable of supporting real-time, cooperative autonomy.

2.3 Oveview of Wireless DC Analysis

With the continued development of wireless technology and CAS, scholars have begun exploring the application of DC protocols, originally designed for wired networks, in wireless environments, conducting significant research in this area. For instance, [82] examines the implementation of the PBFT consensus mechanism in wireless networks, introducing the concept of a "feasible area" to ensure the minimum number of replica nodes required for maintaining protocol security and liveness. By analyzing factors such as coverage range, transmission power, and receiving sensitivity, it proposes a method to optimize network parameters for energy efficiency and performance improvements. Furthermore, the influence of node count and link transmission reliability on Raft consensus has been explored in research efforts such as those by [83] and [13]. Introduced by [83] is the concept of "reliability gain," which reveals a linear relationship between consensus reliability and transmission link reliability, while also addressing the trade-off between consensus delay and reliability. Additionally, reliability gain and tolerance gain are used by [13] to demonstrate a linear correlation between consensus reliability, failure rates, and the maximum number of tolerable faulty nodes. Collectively, these studies offer valuable insights into the performance of DC in wireless environments, providing crucial guidance for future wireless DC protocol deployments.

In addition, several studies have conducted comparative analyses of the performance of different DC protocols under wireless conditions. A framework for wireless blockchain networks (WBN) under various commonly used consensus mechanisms, such as PoW, PBFT, and Raft, was proposed by [17]. The relationship between communication resource availability and the performance of consensus mechanisms—covering factors such as scalability, throughput, latency, and security—was analyzed, highlighting the critical importance of adequate communication resources for ensuring security and performance in wireless blockchain networks. Also examined in [84] are the advantages and disadvantages of PBFT and Raft, studying performance metrics like success rate, latency, throughput, and energy consumption in wireless networks, with simulations conducted using various signal types such as terahertz and millimeter waves. Moreover, the PICA paradigm in wireless networks, combining Raft and Hotstuff BFT, was proposed by [85]. The study focused on the protocols' reliability and latency when facing communication failures, quantifying the strengths and weaknesses of both approaches. These comparative analyses effectively highlight the distinct advantages and limitations of various DC protocols while providing theoretical insights for future critical CAS applications in wireless networks. However, these studies have only explored certain wireless conditions, and further research is needed to address the impact of factors like network architecture, bandwidth, and channel conditions on the performance of DC protocols.

In addition to examining the impact of wireless communication on DC, some studies also explore how DC affects wireless network. The communication resource allocation problem for the Raft protocol in wireless networks is addressed by [86], with the goal of improving decision-making reliability and reducing latency. In this study, communication resource standards necessary for consensus—such as transmission power, bandwidth, and the number of nodes—are defined. Several optimization methods are proposed, including optimizing transmission power through sequential quadratic programming (SQP), optimizing bandwidth allocation using particle swarm optimization (PSO), and determining the optimal number of nodes under fixed resources. Additionally, [87] investigates the security performance of wireless blockchain networks that utilize the RAFT mechanism in environments with malicious interference. This research models blockchain transactions as a wireless network with both uplink and downlink transmissions and employs the Poisson point process (PPP) assumption to analyze node location and the probability of successful transactions. These studies offer fresh insights into using DC to impact wireless environments.

However, traditional DC protocols, which were designed for wired environments, face significant challenges when applied directly to wireless networks and often cannot fully adapt to the needs of many modern applications. As a result, scholars have developed optimizations and extensions to traditional DC protocols. For instance, an adaptive Raft consensus protocol tailored to wireless environments is introduced by [88]. The protocol includes several key phases, such as node counting, leader election, log replication, state synchronization, and mechanisms for node joining and exiting, all aimed at improving protocol robustness and efficiency.Additionally, [89] explores the challenge of multi-valued fault-tolerant DC, particularly focusing on achieving exact output in voting validity. The study defines voting validity as the requirement that the consensus output of non-faulty nodes must precisely match the majority of non-faulty node inputs. To ensure this accuracy, even with arbitrary input values, the authors design a synchronous Byzantine fault-tolerant protocol, particularly suited for safety-critical scenarios. Meanwhile, [90] presents a scalable multi-layer PBFT mechanism, which reduces communication complexity by organizing nodes into layers and restricting intra-group communication, effectively addressing the scalability limitations of traditional PBFT in large-scale networks. These adaptations underscore the ongoing efforts to enhance DC protocols for better performance in wireless applications.

In addition to enhancing existing DC algorithms, some studies have proposed new consensus mechanisms specifically designed to address critical wireless scenarios. For instance, a referendum consensus based on gossip-broadcasting (GB-RC) is introduced by [91], improving communication efficiency and consensus performance in large-scale networks by combining multi-hop gossiping and single-hop broadcasting. To further address scalability issues, [91] also proposes a cooperative broadcast-based electoral college consensus (CB-EC), which reduces consensus latency by sacrificing some robustness. Moreover, [92] introduces Random Representative Consensus (R2C), a novel communication-efficient scheme, and compares its effectiveness against the baseline referendum consensus (RC), while deriving the minimum number of validators needed to ensure resilience against failed nodes and robustness to missing validators. Additionally, [93] presents a fully distributed algorithm for achieving average consensus in wireless sensor networks (WSNs), allowing nodes to converge efficiently to the average of initial measurements using only local node information. [94] offers an algorithm designed to achieve global optimal decision-testing consensus in WSNs without the need for a fusion center. This algorithm is notable for its ability to adapt to multipath and frequency-selective channels, effectively handle propagation delays, and converge to optimal decisions at an exponential rate under bounded delay conditions.

Although the aforementioned studies have provided significant insights into the performance of DC protocols in wireless communication, there remains a gap in the analysis of their application in specific real-world wireless scenarios. Moreover, different scenarios impose varying requirements on consensus algorithms, necessitating tailored analysis and further development to address these unique challenges.

2.4 Oveview of DC Applications in CAV

DC mechanisms have garnered significant attention in autonomous driving applications, including Intelligent Transportation Systems, Cooperative Adaptive Cruise Control (CACC), and Vehicular Ad-hoc Networks (VANETs) [95]. Initial consensus algorithms, primarily focused on managing inter-vehicle spacing within platoons, have been extensively researched in recent years. For instance, [96] introduces a consensus algorithm for second-order dynamic systems, demonstrating how appropriate selection of information states can solve the formation control problem. Additionally, it unifies existing approaches, such as leader-follower, behavioral, and virtual structure methods, can be unified within the general framework of consensus building. Further research, such as [97], proposes a four-layer framework combining longitudinal and lateral controllers to enhance cooperation and coordination between vehicles across different platoons, enabling more complex group consensus. Specifically addressing the continuous value range problem in vehicle platooning, [98] introduces the BFT-ARM (Byzantine Fault Tolerant and Asynchronous Real-Valued Consensus Protocol), which employs the median validity principle to ensure that the consensus decision aligns closely with the median of all normal nodes, improving protocol robustness. Meanwhile, a novel DC control algorithm, designed to coordinate vehicle spacing while considering communication topology and latency, is developed in [99], which validates its effectiveness in real highway conditions using the Veins simulation platform. Moreover, a distributed control protocol is presented in [100], integrating local vehicle state actions with data from neighboring vehicles to ensure the stability of the platoon, with its effectiveness proven using the Lyapunov–Razumikhin theorem.

In addition to vehicles platooning, DC has been applied to other autonomous driving scenarios as well. Blockchain consensus mechanisms have been integrated into the Internet of Vehicles (IoV) architecture, as demonstrated in [14], [101], [102], and [103], where technologies like block verification mechanisms and decentralized data management ensure the credibility of participating vehicles, prevent data tampering, and promote secure data exchange. A decentralized trust management system, proposed in [15], allows vehicles to verify messages from neighboring vehicles using a Bayesian inference model, addressing the issue of information credibility assessment in vehicle networks. Furthermore, a distributed consensus algorithm based on "Proof-of-Eligibility" (PoE) is introduced in [104], which limits the number of vehicles participating in the consensus process, thereby reducing the impact of malicious vehicles on information dissemination outside the event area. Additionally, [105] proposes a parallel consensus mechanism using a directed acyclic graph (DAG)-lattice structure to improve the efficiency and adaptability of blockchain in IoV environments. The R-PBFT consensus algorithm, introduced in [106], reduces consensus delay by 50%, increases transaction frequency by 24% and throughput by 50%, and decreases block congestion by 57% through the use of a reputation-based mechanism, while also providing resistance to attacks such as Finney, Sybil, eclipse, routing, and impersonation attacks.

The structure of the literature review is illustrated in Fig. 2.2, providing a clear overview of the key research domains and their interrelations. Although these studies apply consensus mechanisms to autonomous driving scenarios, several gaps remain in existing research on wireless DC:

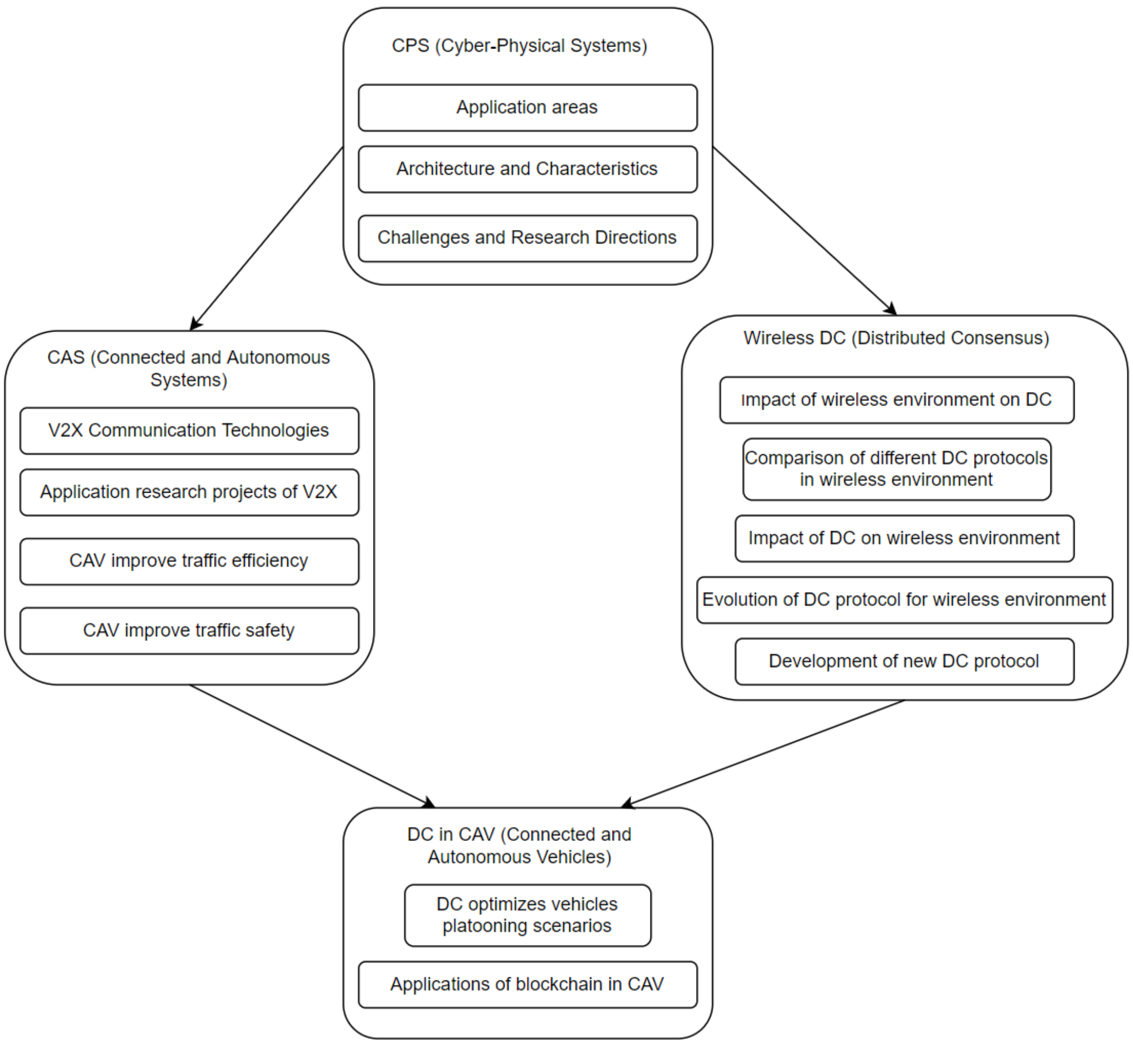

Figure 2.2: Outline of the literature review

- **Limited methods on joint decision-making:** Most studies emphasize privacy, data security, or group control, while joint decision-making and its related approach in autonomous driving scenarios is under explored.
- **Insufficient adaptation to intelligent nodes:** Current wireless DC research has not

adequately evolved to leverage the growing intelligence and automation of terminal devices.

- **Lack of practical hardware validation:** Most research is confined to simulations and theoretical analysis, with limited validation on actual hardware systems.

These gaps present challenges in providing constructive guidance for the real-world deployment of wireless DC in future ICA systems.

Chapter 3

Wireless Intelligent Distributed Consensus enabled Autonomous Vehicles' Cooperation at Uncontrolled Intersection

3.1 Introduction

Chapter 1 highlighted the importance of applying DC in CAS, explaining that traditional DC was originally developed to ensure consistent storage in distributed systems. As multiple nodes (e.g., computers or servers) must replicate data or status, but uncertainties like network delays and node failures make maintaining consistency challenging. To address this, DC protocols were designed to enable nodes to reach consensus, even amid partial failures or network disruptions, thereby ensuring reliable system operation. Early DC protocols, such as Paxos and PBFT, were primarily developed to address this consistency challenge, particularly in distributed databases, computing, and systems applications.

Nevertheless, in applications involving ICA nodes, the focus shifts beyond merely maintaining consistent data storage to enabling joint decision-making. Joint decision-making requires each node to communicate, interpret, and evaluate the data to facilitate final decisions. However, traditional DC protocols do not evaluate the content of the transmitted data itself. Using Raft as an example, as outlined in Chapter 1: after the Leader is elected, Raft facilitates log replication under the Leader's management. The Leader replicates client requests and data to all Followers through downlink transmission and waits for confirmation from them. When a Follower node receives a log entry, it performs several checks to ensure the log's order and consistency. First, each log entry contains a term number, a critical element of the Raft protocol. The Follower verifies whether the Leader's term number is greater than or equal to its own. Additionally, to ensure log consistency, the Follower checks if the log index in the leader's message conflicts with its own logs. These checks are essential to achieving Raft consensus, ensuring data consistency among distributed nodes through strict adherence to the protocol. If the checks pass, the Follower appends the new log entry to its own log and returns a confirmation to the Leader, signifying successful replication. Once the leader receives confirmation from more than 50% of the followers, the consensus is considered achieved.

It can be seen that in the traditional Raft protocol, when the Follower confirms the consensus, it does not interpret or evaluate the data content itself. The checks performed, such as verifying term consistency or checking log index alignment with the Leader, are insufficient for supporting joint decision-making processes. As a result, while traditional DC offers many advantages, it cannot be directly applied to meet the joint decision-making requirements in CAS without further adaptation.

In addition, traditional DC, originally developed for distributed storage systems, does not account for the intelligence of nodes. In such systems, each node follows a preset protocol and fixed verification processes to achieve consistent data storage without requiring high levels of intelligence. However, with the rapid advancement of AI, devices are becoming increasingly intelligent, autonomous, and capable, such as in the case of self-driving cars or robots. These ICA nodes often require specialized intelligent algorithms or neural networks to evaluate data and make joint-decisions. Therefore, applying traditional DC directly to scenarios involving ICA nodes limits their potential. To fully harness the capabilities of these intelligent nodes, DC protocols must be adapted to accommodate their unique characteristics and requirements, enabling more effective joint decision-making. Thus, it is necessary to modify traditional DC protocols to better suit ICA scenarios.

Based on traditional DC, we proposed Intelligent Distributed Consensus (IDC) which refers to consensus mechanisms within ICA nodes that incorporate advanced inference and decision methods, such as deep learning models to improve joint decision-making performance. Unlike traditional DC mechanisms that rely on pre-defined rules only for data duplication, IDC enables nodes within a network to make more intelligent evaluations and judgments on information and enhances the ability to reach collective joint-decisions and consensus. In addition, IDC leverages the benefits of customized consensus mechanisms designed for intelligent and automated environments, enhancing system performance. As

IDC continues to evolve, future ICA nodes will be able to achieve globally optimal joint decisions through iterative discussions and deliberations, much like human joint decision-making processes.

This chapter introduces the first IDC protocol, Intelligent-Raft, which builds upon the traditional Raft algorithm and is specifically optimized for more effective application in intelligent environments. Additionally, to facilitate the deployment of IDC in practical CAS environments, this chapter introduces an innovative system architecture called the Wireless Intelligent Distributed Consensus System (WIDCS) which leverages distributed wireless communication combined with the Intelligent-Raft algorithm. In addition, this chapter also realized WIDCS through a prototype named AIR-RAFT, which integrates extensive hardware (i.e., Radio-Frequency (RF) module) and software architectures (i.e., Intelligent-Raft), enabling practical nodes to achieve wireless joint-decisions. This study represents the first implementation and verification of wireless DC functionality using an practical hardware system. In addition, this chapter proposes a novel and secure traffic passage scheme for AVs passing uncontrolled intersections safely by leveraging WIDCS. AVs can achieve very reliable and efficient collision avoidance with the assistance of WIDCS, which guarantee the security. We establish a comprehensive mathematical model to accurately capture the position, speed, and arrival time distribution of AVs at uncontrolled intersections. We also derive and analyse the collision risk of AVs under conditions without and with WIDCS based on the established mathematical model. Through the utilisation of the AIR-RAFT hardware platform and integration with AVs, we recreate the experimental uncontrolled intersection environment and successfully demonstrate collision avoidance. From the perspective of numerical results, the congruence between numerical outcomes and experimental data robustly substantiates the precision of our theoretical framework and the viability of our proposed scheme, unequivocally demonstrating that WIDCS substantially improves autonomous driving safety.

3.2 Methods

3.2.1 Intelligent-Raft

The Intelligent-Raft protocol, which we propose as the first IDC method, builds upon the traditional Raft algorithm with enhancements tailored to the intelligent demands of CAS. As in Raft, nodes are categorised into three distinct roles: Follower, Candidate, and Leader. In addition, Intelligent-Raft also operates through two key stages: leader election and log replication. The leader election stage remains consistent with the original Raft algorithm, where the elected Leader manages the consensus process for each term. However, log replication diverges from traditional Raft, incorporating joint decision-making mechanisms, which we will explain through the specific consensus process.

The foundational workflow of Intelligent-Raft is illustrated in Fig. 3.1. Within each consensus process, a node initiates by submitting a decision proposal. Subsequent to the acquisition of a majority vote and the successful reception of these vote affirmations by the leader, the proposal is allowed for action or incorporation. The Intelligent-Raft is intricately subdivided into four sequential steps: Initiative, Distribution, Intelligent Evaluation, and Commitment.

In traditional Raft, neither Followers nor Leaders can actively initiate consensus; each consensus process is triggered by a client request to the Leader. Specifically, the client sends a consensus request to the Leader, which then manages and completes the consensus process within the cluster. Once the process is finalized, the Leader returns the result to the client. In contrast, Intelligent-Raft eliminates the distinction between client and server, meaning no external client is available to initiate consensus. As independent ICA nodes, Intelligent-Raft nodes must actively initiate consensus based on their own requirements. Thus, each node must possess both the ability to initiate consensus requests, like a client, and the capability to handle the consensus process, like a server.

Since each Intelligent-Raft node can actively initiate consensus, there are two situations for consensus initialization. In the first, a Follower node initiates a consensus proposal to the Leader, which then broadcasts the proposed data according to the Intelligent-Raft mechanism followed by the standard process. In the second scenario, the Leader directly initiates the consensus by broadcasting its own proposal, bypassing the step where Followers submit proposals to the Leader.

The specific workflow of Intelligent-Raft is outlined in detail as follows:

Initiative

Within the ICA scenario, specific triggering conditions enable a Follower to initiate a decision proposal packet to the Leader. Decision proposals are classified into two categories: real-time and reserved. Real-time proposals, requiring immediate action, take the highest priority in the intelligent consensus mechanism to prevent conflicts, ensuring only one is processed at a time. In contrast, multiple reserved proposals can be buffered at the leader and executed concurrently, enhancing system throughput.

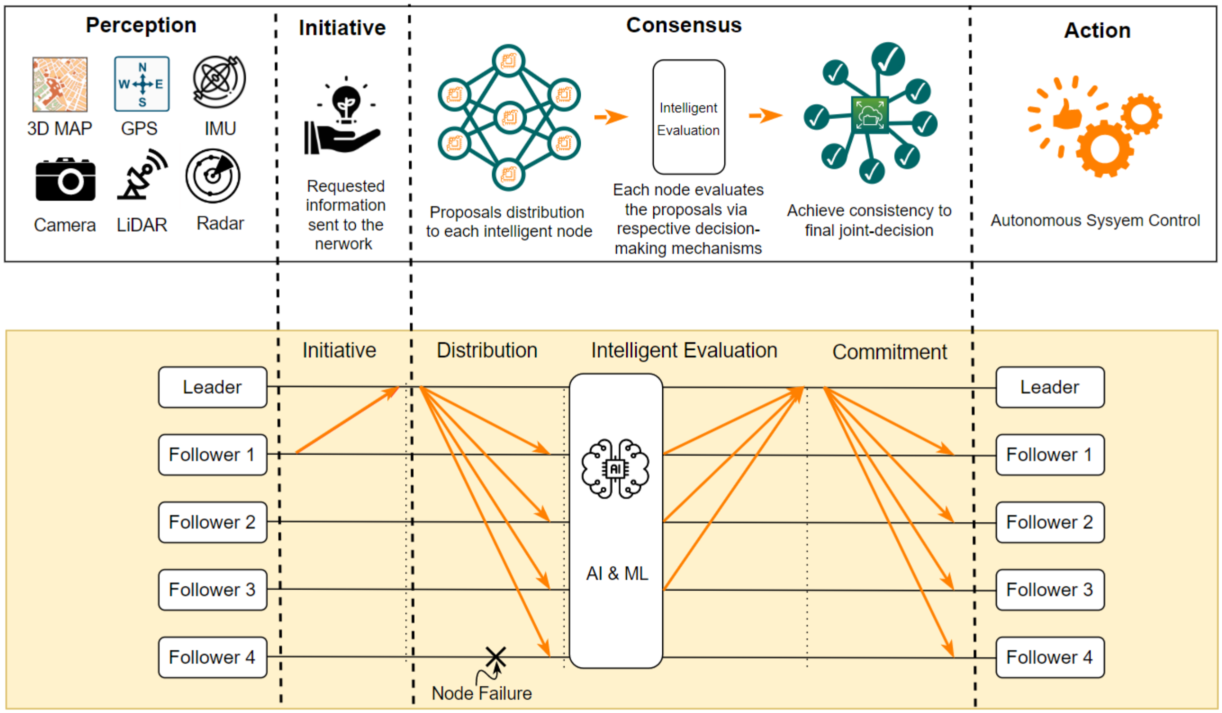

Figure 3.1: The workflow of Intelligent Raft in PICA architecture of WIDCS

Distribution

In this phase, the Leader first receives the decision proposal from the Follower. Unlike traditional Raft, Intelligent-Raft does not send replication requests to followers. Instead, it broadcasts a decision proposal requiring intelligent evaluation by all participating nodes to assess its feasibility. After broadcasting the proposal, the Leader awaits evaluation feedback from all Followers, facilitating joint decision-making.

Intelligent Evaluation

During this phase, participating Followers that successfully receive the Leader's broadcasted proposal begin the validation process. Unlike traditional Raft, which focuses on log term and index alignment for simple storage operations, Intelligent-Raft performs an intelligent feasibility assessment of the proposal. Followers assess both real-time and reserved proposals by evaluating their safety, feasibility, and potential conflicts with their own intended maneuvers, leveraging individual reasoning and decision-making capabilities. For instance, in autonomous driving, Followers assess whether a proposed action could lead to an accident undetected by the proposer's perceptual systems. After completing the validation, each Follower uploads its approval or disapproval to the Leader, without detailing the underlying assessment. Real-time proposals are prioritized during the validation and response process.

Commitment

In the commitment phase, Intelligent-Raft concludes the current consensus and joint decision-making cycle, regardless of the outcome. Upon receiving the response messages, the Leader tallies the positive feedback from Followers. If more than 50% of the Followers consent to the proposal, indicating that sufficient validation has been achieved, the proposal is approved for execution. The Leader then notifies all Followers that decision consensus has been reached. However, if the Leader fails to gather enough approvals or receives a majority of rejections, the proposal is denied. In such cases, the reasons for rejection, along with individual evaluations (if required), are communicated to all participants. This allows the originating node to reconsider and potentially resubmit the proposal with adjustments based on the feedback from the validation process.

3.2.2 Wireless Intelligent Distributed Consensus System

Building on IDC, we propose WIDCS to enable the practical deployment of IDC in CAS. WIDCS incorporates wireless communication for data interaction and joint decision-making based on the Intelligent-Raft protocol. All ICA nodes equipped with WIDCS are capable of utilizing distributed data consensus and joint decision-making services.

Its workflow is structured through the Perception-Initiative-Consensus-Action (PICA) framework, as described in [107]. As shown in Fig. 3.1, the process begins with a node making an initial request. Intelligent-Raft consensus is then achieved through a network of relevant nodes working jointly, before executing the actions that reflect the outcome of the joint decisions. Specifically, the preliminary decision-making (i.e., Perception) is driven by inputs from local sensors, possibly enhanced by various advanced algorithms. A request (i.e., Proposal) is then generated based on the node's local decision and sent to the consensus network for further joint decision-making (i.e., Consensus). Only decisions that obtain collective consent are recorded and executed on a global scale (i.e., Action). Additionally, the consensus is securely logged, preventing any single node from altering it, ensuring accountability and enabling oversight by authorities or insurance entities in the event of an incident.

3.3 Design of Practical WIDCS

3.3.1 AIR-RAFT Framework

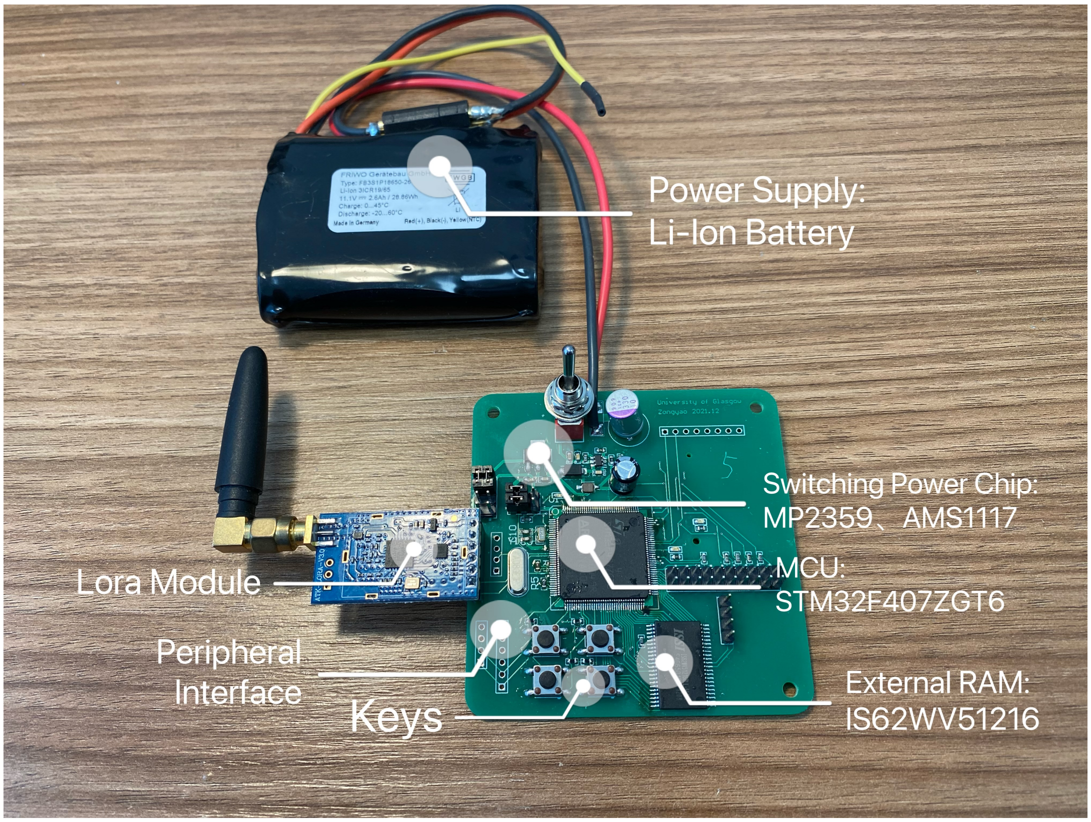

Figure 3.2: This picture shows the hardware composition of AIR-RAFT, the conceptualized entity of WIDCS

To address the issue that many studies on wireless DC lack validation with practical hardware systems, we designed and implemented a fully functional hardware module based on the principles of WIDCS called AIR-RAFT which is schematically illustrated in Fig. 3.2.

The hardware composition of the AIR-RAFT system is shown in Table 3.1, which is mainly divided into five parts: power supply unit, main control unit, memory expansion unit, communication interface and other peripherals. MP2359 and AMS1117 switching power supply chips are selected as the power supply unit, and the gradient step-down mode is used to supply power to the MCU and RF modules. STM32F407ZGT6 chip is selected as the main control unit (MCU). It integrates the powerful Cortex-M4 core with the main frequency up to 168MHz, which can efficiently operate the Intelligent-Raft consensus algorithm. Because the Intelligent-Raft algorithm needs to invoke a large

Table 3.1: AIR-RAFT System Architecture

<table>
<tr><td rowspan="8">AIR-RAFT</td><td rowspan="5">Hardware</td><td>Power Supply</td></tr>
<tr><td>MCU</td></tr>
<tr><td>External RAM</td></tr>
<tr><td>Communication Interface</td></tr>
<tr><td>Other Peripherals</td></tr>
<tr><td rowspan="3">Firmware</td><td>Raft Mechanism</td></tr>
<tr><td>Selective Edge Decision Layer</td></tr>
<tr><td>Transceiver</td></tr>
</table>

number of dynamic memory management operations, we add a memory expansion chip IS62WV51216 with 1 Megabytes capacity for the MCU through the FSMC transmission protocol. In this way, all dynamic data can be stored in the dynamic memory of the MCU. The communication interface mainly adopts the UART interface connected to the Lora RF module, which sends and receives information from the physical layer. Lora is chosen due to its wide coverage and low power consumption, making it suitable for embedded systems. Moreover, the signal transmission is stable, and the topology of the network node connection can be customized flexibly.

As shown in Table 3.1, the frimware of AIR-RAFT module is equipped with FreeRTOS which is a real time operational system to manage various task threads of MCU. The threads in the system are mainly composed of three parts. The most important part is the Raft consensus algorithm. The second part is called the Selective Edge Decision Layer (SED) layer which has a certain privilege to process or execute part of the consensus data as a edge computing unit. The third part manages the communication peripherals, mainly used to receive or transmit data in the wireless environment.

3.3.2 AIR-RAFT Usage

AIR-RAFT can support different distributed IoT application scenarios. For example, as shown in the Fig. 3.3, it can be installed on AVs to work cooperatively with in-vehicle systems and build up a distributed, wireless, reliable and fault-tolerant network. AV system is generally divided into three layers: awareness, local decision and execution. The awareness layer uses a variety of sensors installed in the vehicle (such as millimeter-wave radar, lidar, depth camera, GNSS, etc.) to perceive the surrounding environment. The local decision-making layer uses the pretrained or precompiled algorithm to process and analyse these sensor data, which eventually outputs execution instructions for the underlying mechanical hardware to execute.

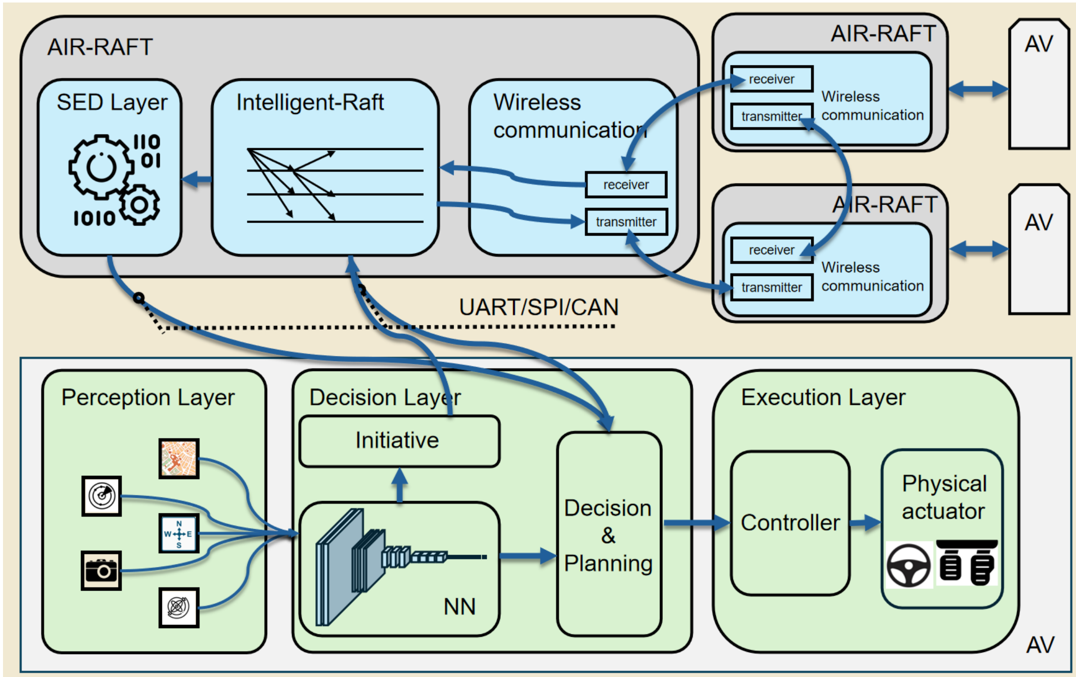

Figure 3.3: This structure shows that how AIR-RAFT is equipped in autonomous driving system

AIR-RAFT can interact with the in-vehicle system via UART or CAN ports like a sensor. The input to the AIR-RAFT module comes from two primary sources. First, since Intelligent-Raft supports data consistency consensus, the data can originate from the environmental information sensed by radar or camera from each independent AV. Through the consensus network, AVs can share environmental data that may be unknown to other nodes. With Intelligent-Raft's consistency guarantees, each node gains a more comprehensive perception of the environment, enabling more accurate and safer actions. Additionally, AIR-RAFT input can also come from the AV's local decisions. When multi-vehicle collaboration is required, an AV can initiate a constructive proposal based on its local decision via AIR-RAFT. Intelligent-Raft then facilitates joint decision-making among AVs, allowing them to coordinate and cooperate based on the agreed decisions. This also highlights the two primary applications of AIR-RAFT or WIDCS: achieving information-sharing consensus and enabling joint decision-making.

The output of the AIR-RAFT module follows two distinct data paths. Information from the consensus network formed by AIR-RAFT can be incorporated into the local decision layer of an AV, which aggregates other sensors data and makes integrated predictions and decisions. Another option is to use the AIR-RAFT as a SED layer [108]. It allows AIR-RAFT to process the committed data directly under certain conditions, and transfer the processed data or commands back to in-vehicle system.

3.4 Uncontrolled Intersection and Secure Passage Scheme

In urban and rural driving scenarios, intersections, particularly those uncontrolled with blind spots, are identified as significant areas of risk on city roads. An uncontrolled intersection is a road intersection where no traffic lights, road markings or signs are used to indicate the right of way [109]. These intersections lack a direct line-of-sight due to physical obstructions such as buildings, trees, or terrain, making it difficult for drivers—and, by extension, AVs—to perceive approaching traffic and potential hazards. For AVs, which rely heavily on sensor input and line-of-sight data to navigate and make decisions, the absence of visual cues at uncontrolled intersections severely hampers their ability to operate safely and efficiently. According to the Federal Highway Administration, which is a division of the U.S. Department of Transportation in Washington, D.C., about 65.7% of fatalities at U.S. intersections in 2021 occurred at unsignalized intersections [110]. Therefore, the important issue in order to achieve safety at the uncontrolled blind intersection is to manage the orderly passage of AVs.

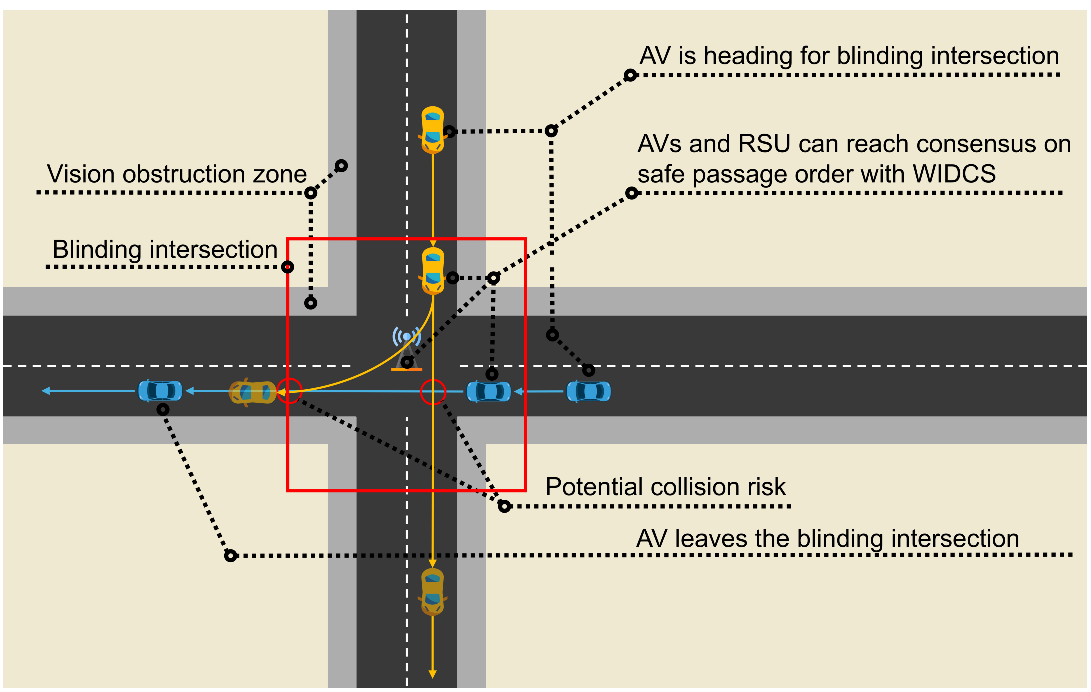

Figure 3.4: The simulation scenario for autonomous driving at uncontrolled intersection

To mitigate collision risks at uncontrolled intersections, where perception systems like cameras and radars in AVs have limited efficacy, a WIDCS-based solution is proposed to enhance AV safety in these environments. Near the intersection center, AVs and Road-Side Units (RSUs) form a local wireless ad hoc network (see Fig. 3.4). The RSU, positioned at the intersection, coordinates network formation and logs consensus-completed data. Once the network is established, the RSU initiates an Intelligent-Raft consensus request to manage the sequential passage of AVs across various lanes. In the absence of an RSU, AVs independently initiate an Intelligent-Raft request upon detecting uncontrolled intersections which can be characterized by obstructed views and the absence of traffic signals. This joint decision-making ensures that a majority of AVs recognize and follow the agreed-upon passage sequence, maintaining consistency. Nodes not initially part of the consensus are updated through network heartbeat messages and quickly align with the joint-decisions. AVs then proceed in the established order, ensuring traffic sequence consistency for safe navigation.

3.5 Mathematical Model Analysis

To evaluate the accident risk associated with AVs at uncontrolled intersections necessitates, we first conducted a theoretical analysis to obtain the spatial positioning and velocity distribution of AVs in proximity to the intersection. We then derived the probability density function (PDF) for the time a vehicle reaches the intersection center using a two-dimensional random variable edge distribution model. Subsequent calculations were made for the wireless link success rate based on the PDF of the signal-to-interference-plus-noise ratio (SINR). Additionally, we developed a probability model for the consensus agreement among AVs facilitated by WIDCS at these intersections. Finally, we integrated the vehicle arrival time distribution model with the wireless consensus model to analyze the variation in accident rates with and without the implementation of WIDCS.

As shown in Fig. 3.4, we assume that an observer stands at the center point of an uncontrolled intersection. We consider the vehicles' distribution of the four lanes of the intersection are the same but are independent to each other, so we only consider one lane situation. For convenience, we use capital letters to denote random variables, and the corresponding lowercase to the value of random variables. The frequently used notations are summarized in Table 3.2.

Notation	Definition
D_n	Distance between the n^{th} vehicle and RSU at intersection center
D_{Sn}	Distance between the nth vehicle which is signal source and RSU at intersection center
$\mathbf{D}_I$	Distance vector of all interference vehicles
F_L	CDF of distances between each vehicles
f_V	Probability density function of vehicles speed
L	Distance between each vehicle
N	Number of vehicles
N^I	Number of vehicles in the first lane
V	Vehicle speed
C	The number of lanes
T	Time when will the first vehicle reach RSU
G	Time gap between each two head vehicles in each lane
P_{RISK1}	Collision risk of vehicles without consensus system assistance
P_{RISK2}	Collision risk of vehicles with consensus system assistance
P_C	Probability of successful consensus
P_l	Probability of successful link transmission in consensus
$P_{SenseFail}$	Probability of autonomous driving sensor perception failure
P	Node transmit power
$g(d)$	Channel path loss model

Table 3.2: Frequently Used Notations

3.5.1 Evaluations of AV Accidents Risk without WIDCS

First we need to know the distribution of vehicle spacing on each lane. We assume that the number of vehicles passing the observer per unit of time is a Poisson process with mean λ which means the traffic flow is λ (in vehicles per hour) [111]. There are K discrete levels of constant speed $v_i(i=1,...,K_{vel})$ on each lane where the speeds are independent identically distributed (i.i.d.). Denote the rate of arrivals of vehicles at each level of speed as $\lambda_i(i=1,...,K_{vel})$ where $\sum_{i=1}^{K_{vel}} \lambda_i = \lambda$, thus, the occurrence probability of each speed level is $P_i = \lambda_i/\lambda$. Thus, the distance between AVs on each lane with observer as the origin obeys the exponential distribution with parameter $\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i}$. So the cumulative distribution function (CDF) of intervehicle distance is:

$$F_L(l) = \begin{cases} 0, & \text{if } l < 0 \\ 1 - e^{-\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i} l}, & \text{if } l \geq 0 \end{cases} \tag{3.1}$$

$$\begin{aligned} P(L > l) = 1 - F_L(l) &= e^{-\sum_{i=1}^{K_{vel}} \frac{\lambda_i}{v_i} l} \\ &= e^{-\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i} l} \end{aligned} \tag{3.2}$$

Where L_n is the distance between the n_{th} closest car to the observer and the $(n-1)_{th}$ closest car to the observer. The distances between each vehicle are independent.

Given the spatial distribution of vehicles, it becomes imperative to acquire the speed distribution of each independent AV. It has been widely accepted that the vehicles' velocity in the free-flow traffic state shows normal distribution and the speeds of vehicles on different lanes have same distribution. Thus, velocity is distributed according to the following probability density function (PDF):

$$f_V(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(v-u)^2}{2\sigma^2}} \tag{3.3}$$

We can combine the distance distribution between the AVs and the velocity distribution to obtain the time distribution of the leading vehicle on each lane arriving at the intersection center. We utilize the edge distribution model of the quotient of two-dimensional random variables, that is, $T = \frac{L}{V}$. Therefore, the PDF and CDF for arriving time T are:

$$f_T(t) = \int_{-\infty}^{\infty} |v| f_L(vt) f_V(v) dv \tag{3.4}$$

$$F_T(t) = \int_{-\infty}^{t} \int_{-\infty}^{\infty} |v| f_L(vu) f_V(v) dv du \tag{3.5}$$

After integral, we can obtain that the specific distribution function of $F_T(t)$:

$$f_T(t) = \begin{cases} \dfrac{\left(\sqrt{2\pi}\left(R\sigma^2 t - u\right)\exp\left(\frac{\left(R\sigma^2 t - u\right)^2}{2\sigma^2}\right)\left(\operatorname{erf}\left(\frac{R\sigma^2 t - u}{\sqrt{2}\sigma}\right)+1\right)+2\sigma\right)}{2^{\frac{3}{2}}\sqrt{\pi}} * & \\ \dfrac{R\exp\left(-\frac{u^2}{2\sigma^2}\right)}{2^{\frac{3}{2}}\sqrt{\pi}}, & t < 0 \\ \dfrac{\left(\sqrt{2\pi}\left(R\sigma^2 t - u\right)\exp\left(\frac{\left(R\sigma^2 t - u\right)^2}{2\sigma^2}\right)\left(\operatorname{erf}\left(\frac{R\sigma^2 t - u}{\sqrt{2}\sigma}\right)-1\right)+2\sigma\right)}{2^{\frac{3}{2}}\sqrt{\pi}} * & \\ \dfrac{R\exp\left(-\frac{u^2}{2\sigma^2}\right)}{2^{\frac{3}{2}}\sqrt{\pi}}, & t > 0 \end{cases} \tag{3.6}$$

$$F_T(t) = \begin{cases} e^{\frac{R^2\sigma^2 t^2}{2} - R\mu t}(\mathrm{erf}(\frac{\sqrt{2}R\sigma^2 t - \sqrt{2}\mu}{2\sigma}) + 1)/2, & t < 0 \\ \frac{e^{-R\mu t}(e^{\frac{R^2\sigma^2 t^2}{2}}\mathrm{erf}(\frac{\sqrt{f2}R\sigma^2 t - \sqrt{2}\mu}{2\sigma}))}{2} & \\ + \frac{e^{-R\mu t}(-e^{\frac{R^2\sigma^2 t^2}{2}} + (\mathrm{erf}(\frac{\mu}{\sqrt{2}\sigma}) + 1)e^{R\mu t})}{2} & \\ - \frac{\mathrm{erf}(\frac{\mu}{\sqrt{2}\sigma}) - 1}{2}, & t > 0 \end{cases} \tag{3.7}$$

Where the $R = \lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i}$, and the σ, μ are the mean and variance of the velocity's PDF, respectively.

Then, we investigate the probability that vehicles on two perpendicular lanes collide. We assume that the leading vehicles on perpendicular lanes reaching the intersection center will collide if the time difference between them is within a certain threshold. The collision probability is:

$$\begin{aligned} P_{RISK1} &= Pr((TimeGap < Threshold) \cap SenseFail) \\ &= Pr(|t_x - t_y| < m) \times P_{SenseFail} \end{aligned} \tag{3.8}$$

Where t_x and t_y are the times when the leading vehicle in two lanes arrives at the intersection and the m represents the threshold of time gap. When the $|t_x - t_y| < m$, we consider there is a certain risk of traffic accidents.

Thus, we investigate the CDF of the time gap between two vehicles, and we use the distribution formula of the two-dimensional random variable difference in our model. We set:

$$G = T_x - T_y \tag{3.9}$$

Where the G represents time gap. Therefore, the PDF and CDF for time gap G are:

$$f_G(g) = \int_{-\infty}^{\infty} f_T(g + t_y) f_T(t_y) dt_y \tag{3.10}$$

$$\begin{aligned} F_G(g) &= \int_{-\infty}^{g} \int_{-\infty}^{\infty} f_T(u + t_y) f_T(t_y) dt_y du \\ &= \int_{-\infty}^{\infty} F_T(g + t_y) f_T(t_y) dt_y \end{aligned} \tag{3.11}$$

Consequently, we can obtain the risk of collision between two AVs as follows:

$$\begin{aligned} P(|t_x - t_y| < m) P_{SenseFail} &= (F_G(m) - F_G(-m)) P_{SenseFail} \\ &= P_{SenseFail} \int_{-\infty}^{\infty} (F_T(m + t_y) - F_T(-m + t_y)) f_T(t_y) dt_y \end{aligned} \tag{3.12}$$

3.5.2 Evaluations of AV Accidents Risk with WIDCS

Wireless Transmission Success Rate

In our framework, the stability of WIDCS is mainly affected by the consensus success rate, which, in turn, is significantly influenced by the reliability of the wireless link. Consequently, our initial step involves analysing wireless link reliability through the modeling of SINR. Spatially, we assume that only nodes within a predetermined range, centered around the intersection, contribute to interference. Furthermore, it is hypothesized that communication among nodes is unscheduled, with signal transmission occurring sporadically. The path loss experienced by desired signals is denoted as $g(d)$, where d represents the distance between the vehicles and the Leader. We assume a uniform transmit power P for all nodes, and the received SINR can be expressed as:

$$SINR(D_{Sn}, N_I, \mathbf{D}_I) = \frac{Pg(D_{Sn})}{\sum_{i=1}^{N_I} Pg(D_{Ii}) + \alpha} \tag{3.13}$$

where D_{Sn} is the distance between the desired vehicle and RSU, $\mathbf{D}_I = [D_{I1}, D_{I2}, ..., D_{In}]$ is the distance vector for all interference nodes, N_I is the total number of interference nodes in four lanes, and α is the noise power. Denote β as the SINR threshold that nodes can successfully receive information bits.

We can see from Eq. (3.13) that the wireless link reliability of each vehicle varies depending on the number of AVs and the distribution of vehicle positions. Then, we analyse the desired signal power and interference separately. For a specific node, the desired signal power S is a random variable written as $S = Pg(D_{Sn})$, where D_{Sn} is the distance between the signal source node and the Leader. Because the distance between each vehicle is independent $D_{Sn} = L_1 + L_2 + L_3 + ... + L_n$ and L_n satisfies the exponential distribution, D_{Sn} obeys Gamma distribution:

$$D_{Sn} \sim \Gamma(n, \lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i}) \tag{3.14}$$

As transmit power P is fixed in this chapter, S is only related to D_{Sn}. Therefore, we obtain that the PDF of desired signal power as:

$$\begin{aligned} f_{Sn}(S = Pg(d_{Sn})) &= f_{D_{Sn}}(d_{Sn}) \\ &= \frac{(\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i})^n d_{Sn}^{n-1} e^{-(\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i}) d_{Sn}}}{(n-1)!} \end{aligned} \tag{3.15}$$

Next we investigate the distribution of the received interference and start from the number of interference nodes N_I. As the average number of AVs on each lane at the intersection has the same distribution, we can obtain the total AVs number distribution:

$$N_I \sim Poisson(\sum_{i=1}^{C} \lambda^i) \tag{3.16}$$

where the C is the maximum value of lanes. In the scenario of uncontrolled intersection, $C = 4$ because there are 4 lanes extending from the center.

Then, we investigate the distance D_{In} between an interference node n and the Leader. The interference nodes' distance has the same distribution with signal source node. Then, we can express the PDF of interference I_n generated by node n as

$$\begin{aligned} & f_{I_n}(I_n = Pg(d_{In})) = f_{D_{In}}(d_{In}) \\ & = \frac{(\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i})^n d_{In}^{n-1} e^{-(\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i}) d_{In}}}{(n-1)!} \end{aligned} \tag{3.17}$$

The total interference, denoted by $I(N_I, \mathbf{D}_I)$, is related to the number of interference nodes N_I and the distance $\mathbf{D}_I$ of these interference nodes. From Eq. (3.16) to Eq. (3.17), we have the PDF of $I(N_I, \mathbf{D}_I)$

$$f_I(N_I = n_I, \mathbf{D}_I = \mathbf{d}_I) = f_{N_I}(n_I) Pr(\mathbf{D}_I = \mathbf{d}_I | N_I = n_I) \tag{3.18}$$

$$Pr(\mathbf{D}_I = \mathbf{d}_I | N_I = n_I) = \prod_{i=I}^{IV} Pr(\mathbf{D}_I^i = \mathbf{d}_I^i | N_I^i = n_I^i) \tag{3.19}$$

$$Pr(\mathbf{D}_I^i = \mathbf{d}_I^i | N_I^i = n_I^i) = \prod_{j=1}^{n} f_{D_{Ij}^i}(d_{Ij}^i) \tag{3.20}$$

Where each roman numerals represent the situation on each lane. We accumulate the node interference for the four lanes centered on the RSU as we need to consider the interference from all the nodes.

The distribution of SINR is obtained from the joint distribution of signal strength and noise contribution. As SINR expressed in Eq. (3.13), by applying the marginalization process to this joint distribution, the PDF of SINR can be expressed as

$$\begin{aligned} & f_{SINR}(D_{Sn} = d_{Sn}, N_I = n_I, \mathbf{D}_I = \mathbf{d}_I) \\ & = \int_{-\infty}^{\infty} f_{D_{Sn}}(SINR \cdot w) f_I(w) \cdot w \, dw \end{aligned} \tag{3.21}$$

Then, we consider the probability that the SINR is greater than a certain threshold as the success rate of the communication link:

$$P_l(d_{Sn}, n_I, \mathbf{d}_I) = Pr(SINR > \beta) = \int\underset{\Omega}{\int}\int f_{SINR} d\Omega \tag{3.22}$$

where Ω is the area of $(D_{Sn}, N_I, \mathbf{D}_I)$ that satisfies $SINR(D_{Sn}, N_I, \mathbf{D}_I) > \beta$. As f_{SINR} is obtained in Eq. (3.21), we only need to find the satisfied area.

Reliability of WIDCS

The probability of achieving an Intelligent-Raft consensus in the WIDCS system can be calculated by considering the transmission success rates at different locations and with varying numbers of autonomous vehicles. Since each vehicle's transmission success rate can differ based on the specific circumstances, we employ the exhaustive method to sequentially calculate and accumulate the consensus success rate using Eq. (3.23). We use conditional judgment to identify events that meet the success criteria, calculate their probability of success, and then accumulate them to obtain the final consensus success rate.

Section 3.2 outlines the description of the communication process and success criteria for completing a Raft consensus. To analyze the probability of consensus under varying transmission success rates mathematically, we introduce random events A_m^i and B_n^j. Event A_m^i represents RSU successfully communicate with m random downlink vehicles out of $N-1$, and B_n^j represents the successful uplink response of n vehicles out of m to the RSU. i has a total of C_{N-1}^m cases and j has a total of C_m^n cases, where $\frac{N-1}{2} <= n <= m$. Employing the full probability formula, we derive the probability of completing a consensus, accounting for different transmission success rates and considering all possible combinations:

$$P_C = \sum_{m=\frac{N-1}{2}}^{N-1} \sum_{i=1}^{C_{N-1}^m} P(A_m^i) P(\sum_{n=\frac{N-1}{2}}^{m} \sum_{j=1}^{C_m^n} B_n^j | A_m^i) \tag{3.23}$$

$$P(A_m) = \prod_{i_0=1}^{m} P_{li_0} \prod_{j_0=m+1}^{N-1-m} (1 - P_{lj_0}) \tag{3.24}$$

$$P(B_n) = \prod_{i_0=1}^{n} P_{li_0} \prod_{j_0=n+1}^{m} (1 - P_{lj_0}) \tag{3.25}$$

AV accident risk with WIDCS

Here, we consider that as long as a consensus can be reached, all active vehicles can reach final consistency on the passage order of the uncontrolled intersection, and traffic safety can be guaranteed. Therefore, the probability of accidents occurring at uncontrolled intersections is:

$$P_{RISK2} = Pr(|t_x - t_y| < m) \times (1 - P_C) \times P_{SenseFail} \tag{3.26}$$

3.6 Experiments

To validate the above-illustrated concepts and designs, we conducted tests to evaluate the performance of AIR-RAFT. Through the integration of the AIR-RAFT and AVs, we recreate the experimental uncontrolled intersection scenario and demonstrate collision avoidance to verify the feasibility of WIDCS and examine the performance. The following paragraphs introduce the setup, parameters and conditions of the experiment. Then, we introduce the steps and the process of the experiments. Finally, we analyze the experimental data.

3.6.1 Hardware setup

JetRacer Pro, as shown in Fig. 3.5, is a high performance AI Racing Car with fast speed. JetRacer Pro is powered by NVIDIA Jetson Nano as its main computing unit which is a small, powerful AI computer. According to the uncontrolled intersection traffic scheme, we need not only to identify road path and uncontrolled intersection, but also to visually detect whether there are vehicles passing by. For visual road following in autonomous driving, we use the in-depth learning framework and tools provided by Snow Official, which provides high frame rate processing through Torch2trt (PyTorch to TensorRT Translator) optimising. For object detection, we use the DetectNet of Jetson Inference which is a deep neural network (DNN) library developed for hardware computing resources of NVIDIA Jetson. We collected classification datasets of cars, roads and traffic signs, and then trained our own target recognition neural network model.

In the experiment, AIR-RAFT is installed on each JetRacer cars, facilitating communication through UART. The Lora module carried by the AIR-RAFT supports half-duplex mode, with a space speed of 2.4 KB/s, a transmission power of 20 dBm, a transmission frequency of 410 MHz and a bandwidth of 1MHz. In addition, the MCU (STM32F407ZGT6)

Figure 3.5: JetRacer Pro autonomous vehicles

has an operating principal frequency of 168MHz, and its baud rate for data exchange to Lora is 115200.

3.6.2 Experimental Setup

Experiment 1

In the first experiment, we evaluate the feasibility of WIDCS by investigating the relationship between consensus latency and different numbers of AIR-RAFT. Under certain trigger conditions, one node can propose a packet of data with a fixed length to the leader node to initialize a Raft consensus. The AIR-RAFT cluster then deals with the consensus process based on the Raft consensus mechanism. When a consensus is complete, AIR-RAFT nodes callback the data and execute. The performance of AIR-RAFT is evaluated by measuring the time required for varying numbers of nodes to achieve consensus.

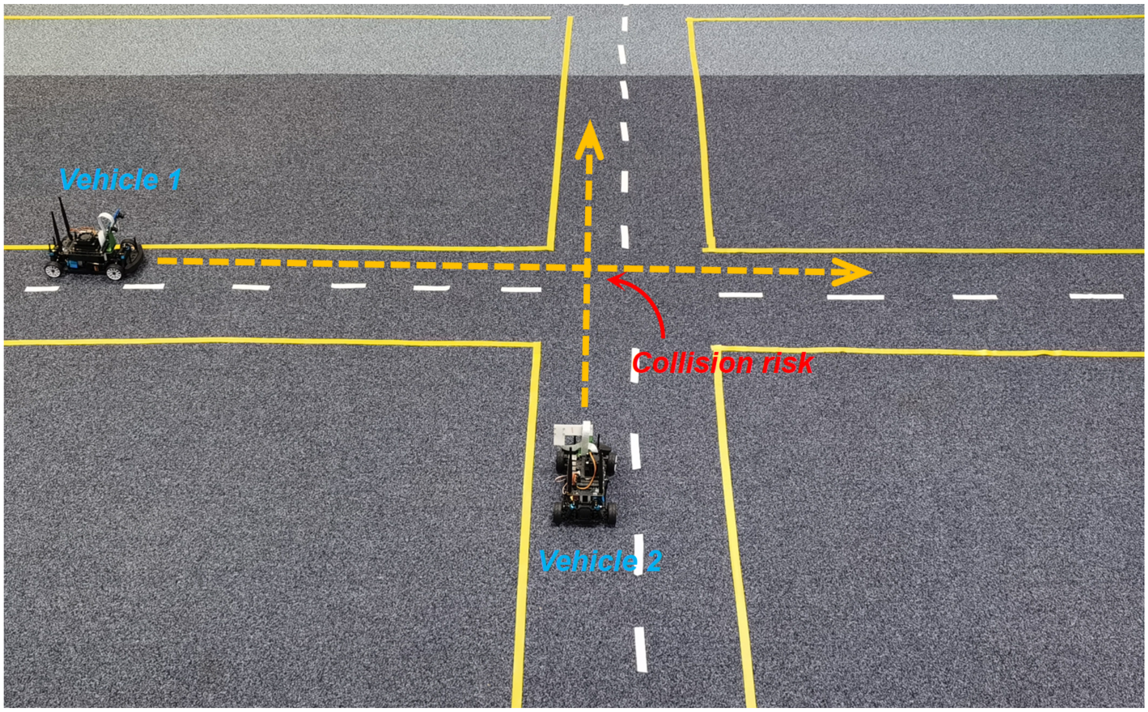

Figure 3.6: 2 autonomous vehicles try to pass blind intersection without WIDCS

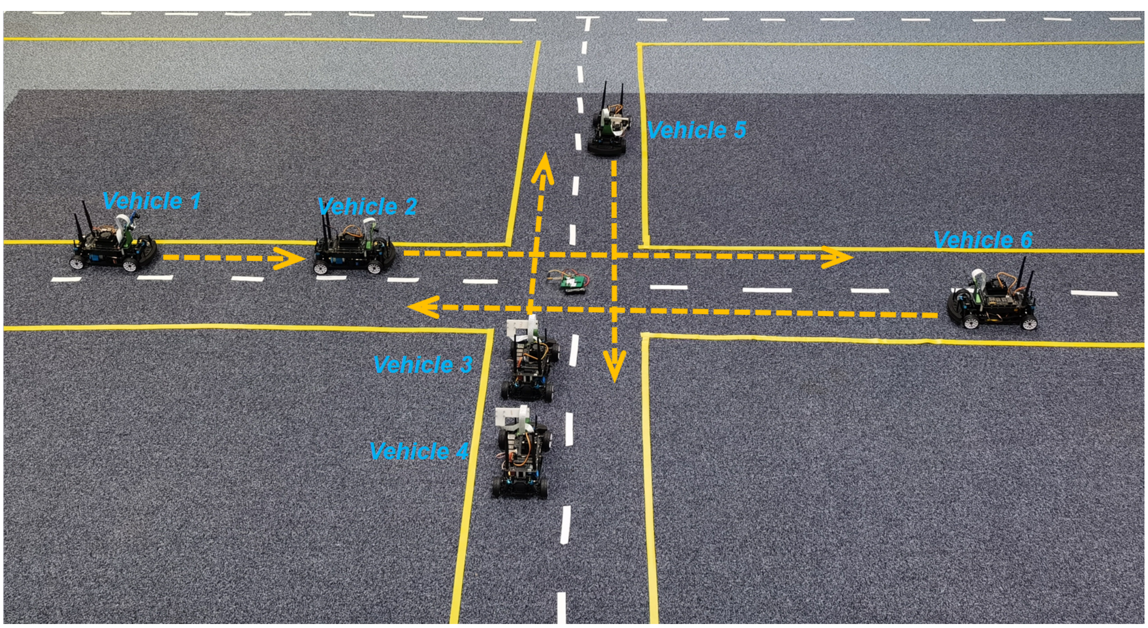

Figure 3.7: 6 autinomous vehicles assisted by WIDCS at blind intersection

Experiment 2

In research on the accident risk of AVs at uncontrolled intersections, we post a scene of an intersection without traffic signal to regulate traffic order, as shown in Fig. 3.6 and 3.7. In the absence of WIDCS, we let two JetRacer cars drive in two perpendicular lanes at uncontrolled intersections, and they both drive toward the center. Consequently, we defined the event of two cars passing through an intersection as a random occurrence, dividing the sample space into two outcomes: collision and non-collision. For each random event, the departure distances and velocities of the two cars are randomly initialized based on their corresponding distributions. A collision event is defined as occurring when two vehicles reach the center of the intersection simultaneously.

To maximize the simulation of the randomness of a vehicle's initialized parameters, a software random number process is used to generate the speed and departure distances of vehicles at each random event. The randomly generated speed satisfies the normal distribution, and we set the maximum speed to 2.6m/s and the minimum speed to 0.2m/s. We also set the maximum distance to 10 m. In this experiment, we repeated 100 times and counted the frequency of collision events. We repeated the experiment 4 times to estimate the collision probability in this simulated uncontrolled intersection.

Experiment 3

For AIR-RAFT-assisted scenarios, We research the effect of different numbers of vehicles on the collision probability. We formed distributed clusters consisting of a RSU and two to six vehicles, and investigated the probability of collisions within each cluster at uncontrolled intersections. As with previous settings, we also use the software random number process to generate the speed and departure distance for each car and treat whether they collide at the center of the intersection as a random event. When vehicles drive towards the center of the uncontrolled intersection, one of the leading vehicle who recognises the intersection through cameras can trigger a consensus within the ad hoc network. When a consensus is reached, AVs pass in an orderly and safe manner in accordance with the agreed passage order. We calculate the frequency of collisions by repeating 100 random events and repeate the above process twice for each cluster to estimate the AIR-RAFT-assisted accident probability.

3.6.3 Experimental Results

Experiment 1

Table 3.3: Relationship between Nodes Quantity and Consensus Duration

Quantity of Nodes	Latency (ms) Start from Followers	Latency (ms) Start from the Leader
3	928.8	688.2
4	1145.6	900.1
5	1149.1	903.8
6	1370.0	1120.2
7	1373.8	1125.5

Table 3.3 shows the average latency of 20 consensus requests for different numbers of nodes within a cluster. The second column shows the average latency of a complete PICA process that followers initiate a consensus in the AIR-RAFT system. As the cluster expands with more nodes, the duration necessary to achieve consensus correspondingly increases. This increase in timeis attributed to the leader's requirement to await additional responses from the followers, thereby extending the period needed to receive and determine the attainment of consensus. In addition, Table 3.3 shows that the consensus time of two adjacent numbers of nodes is basically the same. For example, the consensus duration of four and five AIR-RAFT nodes is around 1145 ms, and increases by about 210 ms for each additional level, e.g., from four nodes to six nodes. This is because leader confirm the success of the consensus immediately after receiving more than half positive feedbacks from followers in Commitment stage. At this time, the suggestions from the rest of the nodes do not affect the consensus of this round.

The third column shows the consensus time when the leader initiates a consensus directly, that is, there is no Initiative stage. It can be seen that when the follower's initialization of the proposal to the leader is not required, the time to reach consensus is reduced by about 250 ms.

Experiment 2

The findings from Experiment 2 are summarized in Table 3.4 which indicate that the probability of collision between vehicles traveling along two vertical lanes in the absence of WIDCS assistance is approximately 0.215. This observation underscores a high risk of accidents for autonomous driving at uncontrolled intersection.

Table 3.4: Accident Probability without WIDCS

nth test	**1**	**2**	**3**	**4**
Collision	20	19	23	24
No collision	80	81	77	76
Collision frequency	0.20	0.19	0.23	0.24

Experiment 3

Table 3.5: Accident Probability with WIDCS

Number of nodes(with RSU)	**3**	**4**	**5**	**6**	**7**
1_{th} collision frequency	0.06	0.05	0.03	0.03	0.01
2_{th}collision frequency	0.05	0.05	0.04	0.03	0.01
Average collision frequency	0.055	0.05	0.035	0.03	0.01

The findings listed in Table 3.5 revealed that in the presence of two vehicles and an RSU (three nodes), the collision probability was determined to be 0.055. Compared to the case without AIR-RAFT in Experiment 2, the AIR-RAFT system reduces the AV collision probability by 74.4% with the same number of vehicles. These findings provide compelling evidence that AIR-RAFT effectively facilitates the establishment of a coherent passage order at uncontrolled intersections, thereby substantially enhancing the safety of autonomous driving.

In addition, with an increase to seven nodes (including RSU), the collision probability reduced significantly to 0.01. This observed trend indicates that augmenting the number of nodes within the network can lead to a diminished collision probability for autonomous vehicles.

This reduction in collision probability can be attributed to the achievement of a successful consensus among the nodes, and enabling the establishment of a consistent traffic order for autonomous driving. Nonetheless, it is imperative to acknowledge that there exists an upper limit on the number of nodes in the network due to inherent constraints in latency and bandwidth. As the number of nodes continues to increase, the demands on time and bandwidth resources for consensus attainment also escalate, potentially resulting in prolonged consensus times that could compromise safety and lead to accidents. Hence, it is crucial to undertake further investigations into the potential risks associated with higher numbers of nodes in future research endeavors.

While the current experiments are conducted in a controlled laboratory environment, the results nonetheless reveal key insights with real-world implications. Specifically, the findings demonstrate the potential of WIDCS to facilitate both reliable information sharing and coordinated joint decision-making among autonomous agents. In future real-world deployments, such a system would enable vehicles to exchange real-time environmental information and collaboratively make context-aware decisions. This enhanced level of interaction is expected to significantly improve the responsiveness, adaptability, and overall safety of autonomous driving in dynamic and complex traffic scenarios.

3.7 Simulation Numerical Results And Discussion

This chapter focuses on simulated accident probabilities analysis and comparison for autonomous driving with and without the assistance of WIDCS at blind intersections. Firstly, we evaluate the time distribution for the lead vehicle to reach the center of the intersection under different traffic flows, average vehicle speeds, and speed standard deviations. By considering the time difference between two vehicles, we can analyze the collision probability in the absence of WIDCS. Additionally, we assess the probability distribution of the vehicle communication SINR exceeding a specific threshold at different distances from the RSU and with varying numbers of vehicles. Furthermore, we conduct simulations to estimate the probability of achieving a successful consensus within the cluster. Finally, we evaluate the collision probability in autonomous driving when WIDCS is utilized.

3.7.1 Simulation Settings

Table 3.6: Summary of Parameters

Parameter	**Value**
Average vehicle speed σ	70 km/h
Speed standard deviation μ	30 km/h
Traffic flow each lane λ	250 veh/h
The radius of considered area	0.5 km
Poisson distribution $\lambda(min)$	17 veh/min
Transmit power P	20 dBm
Path model loss $g(d)$	$g(d) = d^{-2.5}$
Noise power α	-104 dBm

To accurately simulate the real-world scenarios of autonomous driving at uncontrolled intersections, it is crucial to ensure that the preset data closely resembles practical conditions. Therefore, we adopt specific values to capture the realistic aspects of the simulation.

For instance, we assume an average vehicle speed of 70 km/h, a standard deviation of 30 km/h, and a traffic flow of 250 vehicles per hour. Subsequently, we analyze and compare the influence of individual parameters by controlling variable method. In terms of intersections, we define a maximum distance of 0.5 km. Accordingly, our model focuses on AVs within a circular range centered on the RSU with a radius of 0.5 km. The distribution of AVs across the entire intersection area is controlled by a parameter, λ, which we set to 17 veh/min. To facilitate ease of reference, we summarize all the relevant parameters in Table 3.6.

3.7.2 Accidents Risk Evaluations without WIDCS

To evaluate the risk of accidents involving autonomous vehicles without WIDCS at blind intersections, our analysis focus on examining the time it takes for the leading vehicle on each lane to reach the center of the intersection. This analysis involves studying the distribution of arrival times, which is computed from Eq. (3.7). Specifically, we derive the distance distribution F_L between vehicles and the probability density function f_V of vehicle speed. Using the edge distribution model of two-dimensional random variables, we obtain the time distribution function F_T for the leading vehicle based on the relationship $T = \frac{L}{V}$.

Fig. 3.8 illustrates the time distribution function F_T of AV arrival with different speed means. A higher speed mean leads to a higher CDF value at the same time, indicating that AVs with a higher mean speed are more likely to reach the intersection center in a shorter time.

Fig. 3.9 presents the time distribution function F_T of AV arrival under different speed standard deviations, while keeping the mean and traffic flow fixed. We observe that a smaller standard deviation of vehicle speed indicates less dispersion, results in a higher probability of vehicles reaching the center of the intersection simultaneously.

Fig. 3.10 explores the impact of different traffic flows on the distribution of vehicle arrival time. As the traffic flow decreases, the probability of vehicles arriving at the center decreases simultaneously. This is due to the influence of traffic volume on the distance distribution between autonomous vehicles. According to Yousefi [111]'s model analysis, lower traffic flow leads to a more uniform distribution of vehicle spacing, reducing the occurrence of very small spacing. Consequently, vehicles are more likely to arrive over a relatively longer time period.

Fig. 3.11 illustrates the impact of the time gap threshold on the probability of collisions. The horizontal axis represents the threshold, indicating the range within which the time gap between two vehicles would result in a collision. As the time gap threshold

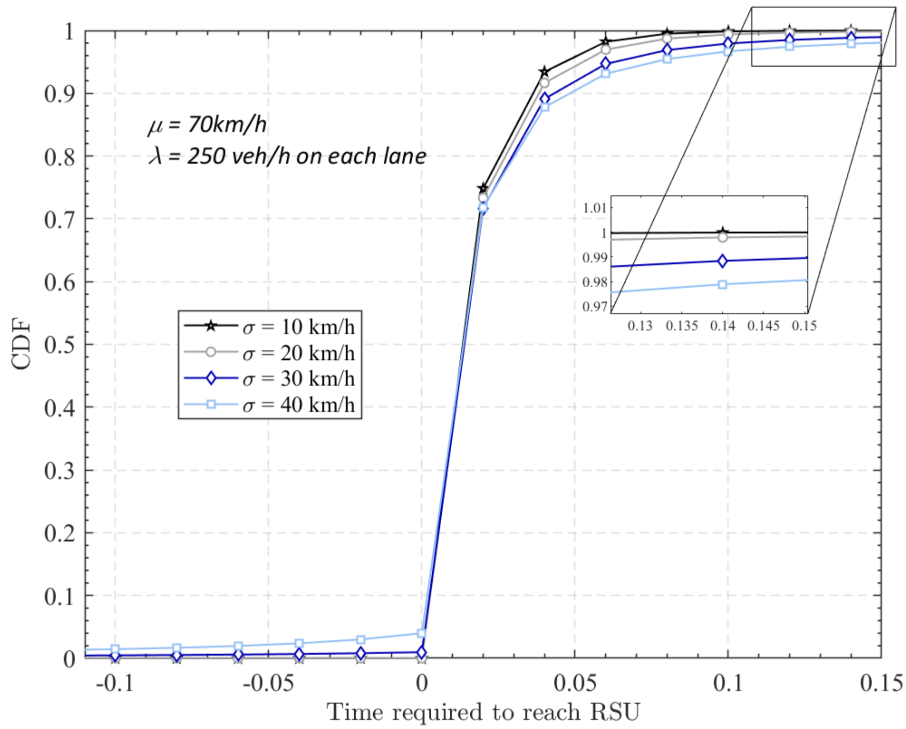

Figure 3.8: Time consuming distribution function with different σ

increases, the probability of collisions also increases. Furthermore, Fig. 3.11 compares the influence of different traffic flows on the probability of accidents. As the flow of AVs increases, the probability of accidents under the same time gap threshold also increases.

Notably, for our model and specified parameters (mean speed: 70 km/h, standard deviation: 30 km/h, traffic flow: 250 veh/h), the probability of the leading vehicle arriving within 0.02 hours (1.2 min) is 0.72. In other words, there is a 0.72 probability that the leading vehicle will reach the intersection center within 1.2 min at any given time. If two AVs arrive at the intersection within a time interval of less than 5 seconds, the scenario is classified as a potential collision, resulting in an accident probability of 0.3.

3.7.3 Accidents Risk Evaluations with WIDCS

Here, we investigate the probability of collision accidents with the assistance of WIDCS. In our simulation, we examine the transmission success rate, denoted as $Pr(SINR > \beta)$, with a single RSU positioned as the leader at the center of the intersection. The analytical results are computed based on Eq. (3.21). In details, we obtain the PDF of the distance between the signal source vehicle and the RSU, utilizing Eq. (3.1). Additionally,

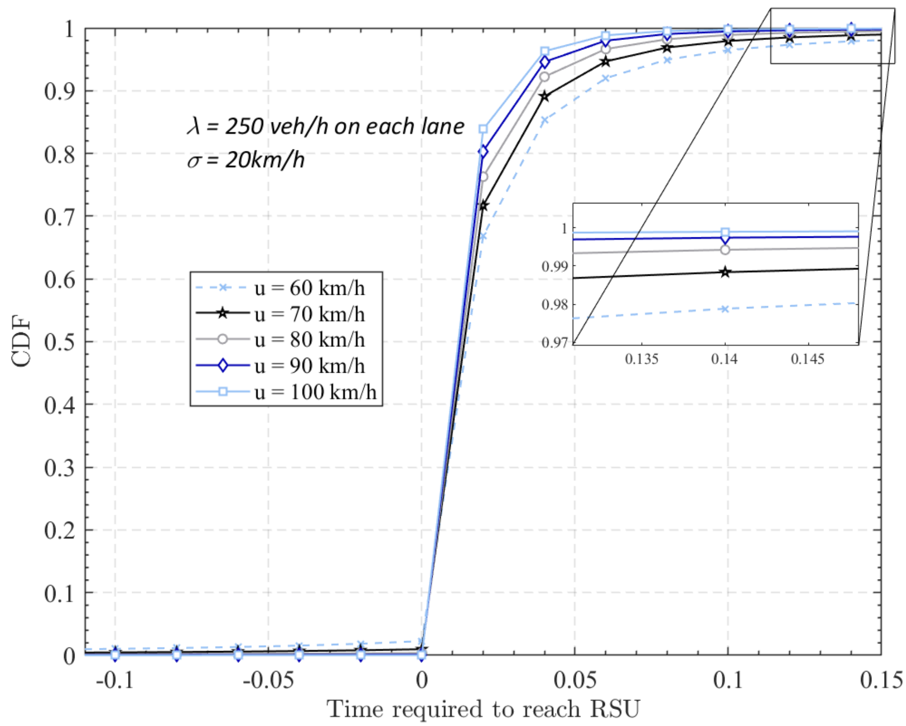

Figure 3.9: Time consuming distribution function with different μ

we employ the Poisson distribution model to describe the number of vehicles within the considered area. By combining Eq. (3.15) and Eq. (3.18), we derive the distribution of $Pr(SINR > \beta)$, which represents the successful rate for each transmission.

Our investigation delves into the impact of varying distances and vehicle numbers on the transmission success rate. We consider all other vehicles within the considered area as potential interference. For simulation purposes, if the received SINR for a transmission surpasses the threshold value β, it is regarded as a successful transmission; otherwise, it is counted as a failure.

Fig. 3.12 illustrates the probability $Pr(SINR > \beta)$ under different distances to the RSU and varying vehicle numbers. Notably, as the distance increases, the transmission success rate experiences a significant decline. This phenomenon can be attributed to the compounded effects of path energy loss and susceptibility to complex electromagnetic environments. For instance, at a distance of 2 km, the transmission success rate is approximately 0.1, indicating a considerable low probability of successful transmissions. As a result, our simulation solely considers vehicles within a radius of 0.5 km from the RSU as the central area of operation.

The probability of achieving a Intelligent-Raft consensus in the WIDCS system can

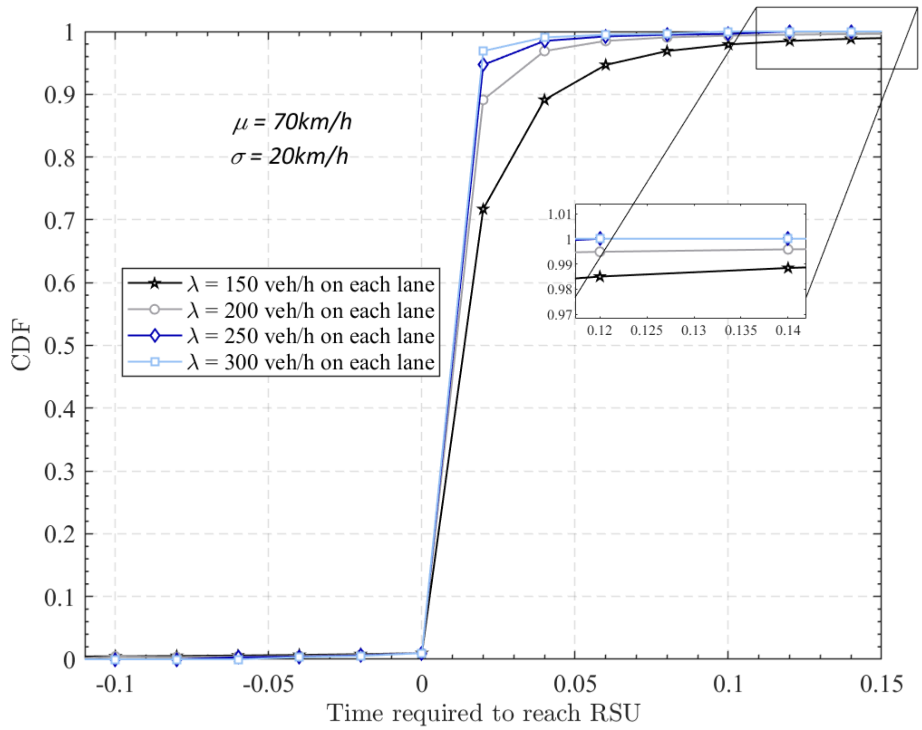

Figure 3.10: Time consuming distribution function with different λ

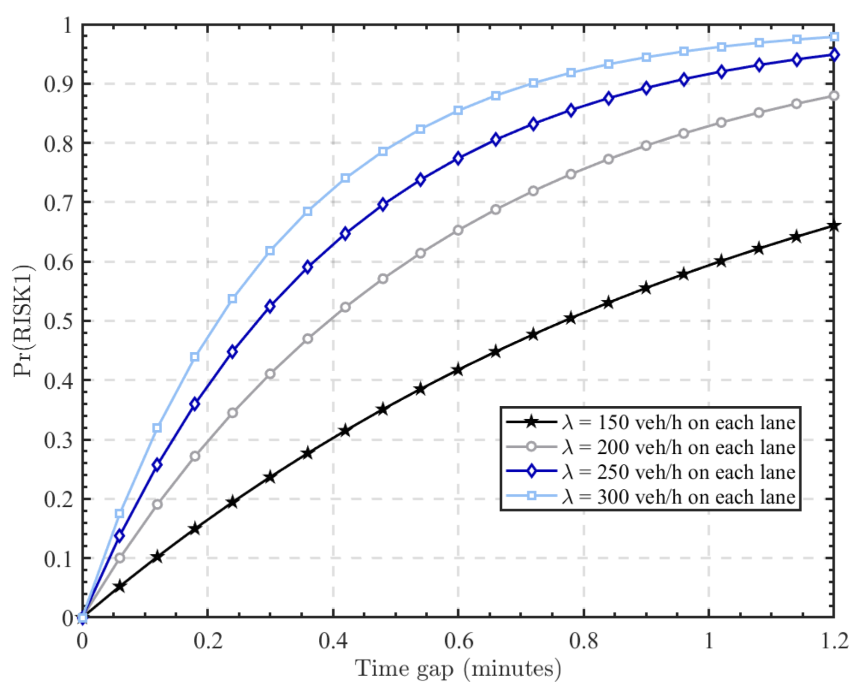

Figure 3.11: Relationship between the time gap threshold and collision risk of leading cars in two vertical lanes

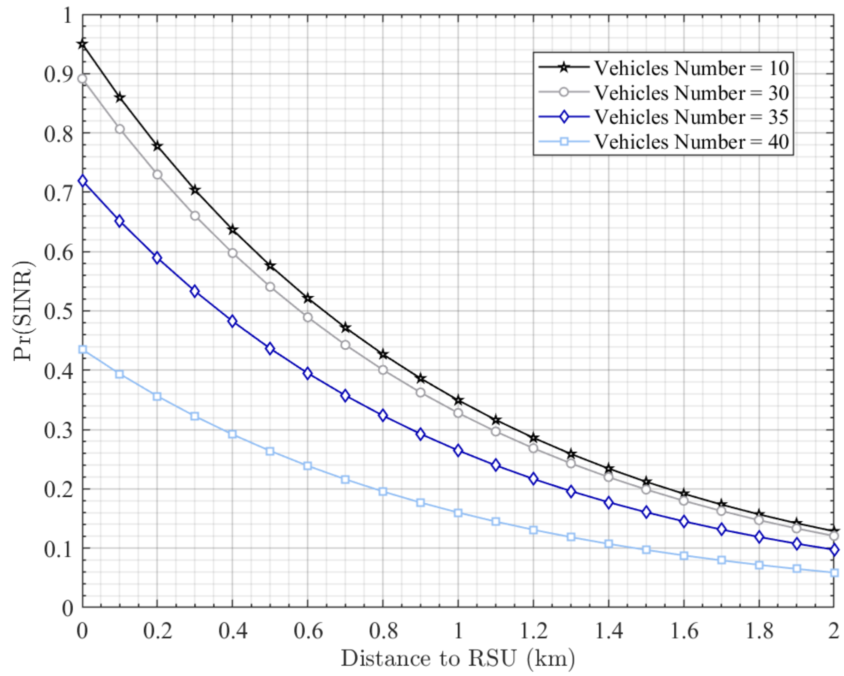

Figure 3.12: The different number of AVs in the network and their distance will affect the transmission success rate of each node

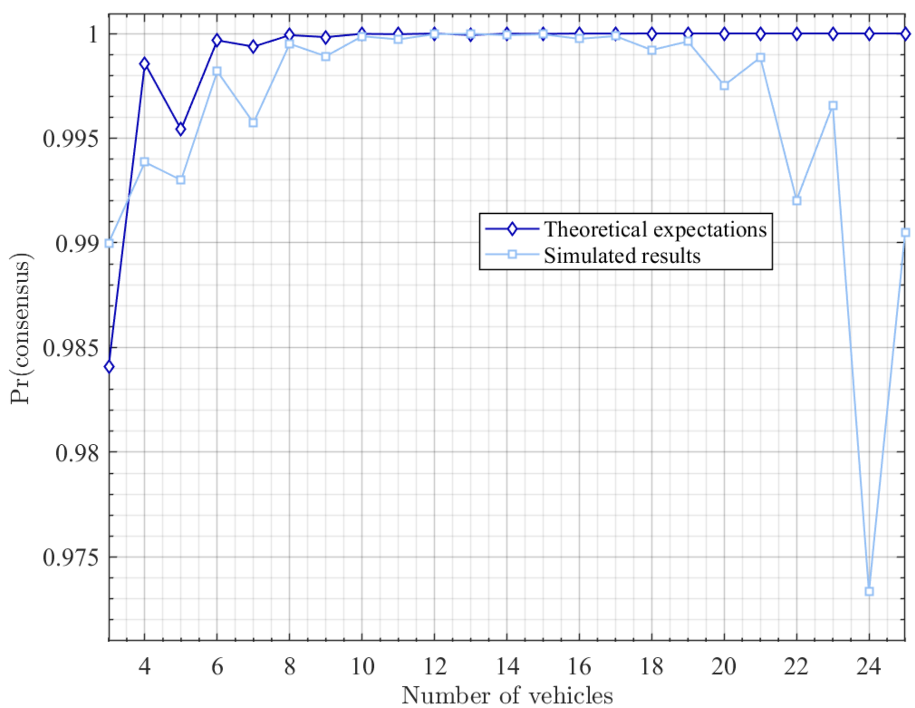

Figure 3.13: Comparision of $P_{Consensus}$ versus number of autonomous vehicles

be calculated by considering the transmission success rates at different locations and with varying numbers of autonomous vehicles. Since each vehicle's transmission success rate can differ based on the specific circumstances, we employ the exhaustive method to sequentially calculate and accumulate the consensus success rate using Eq. (3.23). Fig. 3.13 illustrates the consensus probability with both analytical and simulation results.

In the simulation, we randomly generate the position of each vehicle and vehicle's number following the distance distribution and the Poisson distribution. We repeat the calculation 200 times to obtain an average consensus success rate in the simulation environment. For the theoretical case, we calculate the mathematical expectations of the distance for each vehicle and the quantity, and use them to compute the corresponding consensus success rate.

Fig. 3.13 demonstrates a close match between the analytical and simulation results when nodes number is less than 15. Both analytical and simulated consensus success rate increase with the nodes number. When the number of vehicles is 10, the consensus success rate is approximately 0.998 for the simulated results and 0.999 for the theoretical expectation. This observation aligns with the trend observed in experiments. However, when the number of nodes exceeds 15, the simulated results begin to exhibit great fluctuations and a decline. This phenomenon arises from the increased interference among nodes as their density grows in practical scenarios. The resulting reduction in SINR diminishes the success rate of communication links, leading to a sharp drop in the overall consensus success rate. Hence, it is necessary to limit the number of nodes participating in joint-decisions.

Once a round of Intelligent-Raft consensus is successfully completed, vehicles on different lanes are made aware of each other's presence and their respective passage order, effectively preventing collisions at uncontrolled intersections. The collision probability with WIDCS assistance, denoted as P_{RISK2}, can be derived using Eq. (3.26). For P_{RISK1}, we assume that there is a risk of collision when the time gap between two lead cars is less than 5s. Therefore, the probability of P_{RISK1} is 0.05, which has no relations with the number of vehicles at this moment from Eq. (3.8). We can see from Fig. 3.14 that with the help of WIDCS, the probability of accidents is significantly reduced, which is consistent with our experimental conclusion. In addition, as the number of nodes increases, the accident probability of AVs at uncontrolled intersections decreases to a certain extent. These results indicate that by facilitating effective information exchange and intelligent joint decision-making among vehicles, WIDCS proves to be a valuable asset in ensuring safer navigation and collision avoidance at uncontrolled intersections. These results reaffirm the notion that the integration of WIDCS technology holds great potential for advancing

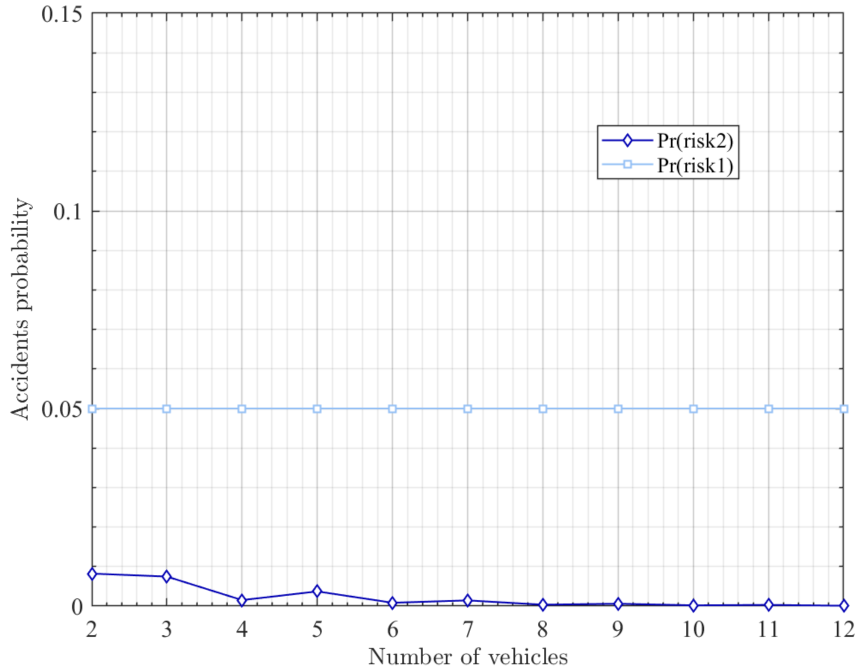

Figure 3.14: Accident probability comparison of autonomous vehicles with and without WIDCS at uncontrolled intersection

the field of autonomous driving and promoting safer mobility on our roads.

3.8 Conclusion

In summary, this chapter presents significant contributions to the field of autonomous driving perception and wireless distributed consensus systems. We successfully implemented the Intelligent-Raft consensus protocol on the embedded hardware platform, AIR-RAFT, making it the first wireless intelligent distributed consensus system of its kind. Furthermore, we introduced a novel and effective solution for ensuring safe passage at uncontrolled intersections through the WIDCS. Our comprehensive mathematical model accurately captures the position distribution, speed distribution, and average time distribution of vehicles at uncontrolled intersections, enabling analysis of the relationship between SINR and transmission success rate, as well as the consensus success rate. We also developed a collision risk model for autonomous driving with and without WIDCS.

To validate our approach, we conducted experiments using the AIR-RAFT hardware platform, recreating the uncontrolled intersection scenario with JetRacer cars and successfully achieving collision avoidance through AIR-RAFT. The consistency between experimental and simulation results reinforces the effectiveness of WIDCS in significantly reducing accident rates in autonomous driving scenarios.

Despite these promising results, several limitations remain. In real-world deployments, wireless communication environments are highly dynamic and unpredictable, which may lead to performance fluctuations of WIDCS. Although this study considers communication reliability, external factors such as node mobility may further impact system robustness. In addition, while WIDCS improves decision-making efficiency, scalability challenges may arise as the number of nodes increases, potentially causing consensus delays and network congestion—especially in dense or rapidly changing traffic environments.

Despite these limitations, the findings provide a solid foundation for future research aimed at advancing the deployment and refinement of WIDCS in real-world autonomous driving scenarios.

Chapter 4

Intelligent Distributed Consensus for Connected Vehicles: Models, Implementation and Testing

4.1 Introduction

Increasingly intelligent and autonomous systems place critical demands on distributed information synchronization and joint decision-making, underscoring the potential of DC, and more advanced IDC, for ensuring joint-decision consistency and providing fault tolerance. Chapter 2 successfully verified the effectiveness of WIDCS in enhancing safety for autonomous vehicles at uncontrolled intersections, demonstrating its potential to optimize traffic flow and reduce collision risks. However, the current WIDCS design focuses primarily on wireless data transmission and has not considered the importance of the wireless ad hoc network. The formation and management of an ad hoc network is not merely a technical detail but a fundamental requirement for implementing IDC. IDC's smooth operation relies on continuous, efficient data exchange between nodes, which in turn depends on a stable wireless network. Thus, the establishment of the wireless network, along with any variations in network conditions, will directly impact IDC's performance.

Wireless ad hoc networks are essential to the functioning of WIDCS. From the engineering feasibility standpoint, these networks enable nodes to identify the communication addresses of other nodes for data exchange. For example, in autonomous driving scenarios, as depicted in Fig. 4.1, two AVs may need to share environmental perception data but initially lack knowledge of each other's communication addresses or frequency bands, making it difficult to establish effective communication. With the support of the ad hoc

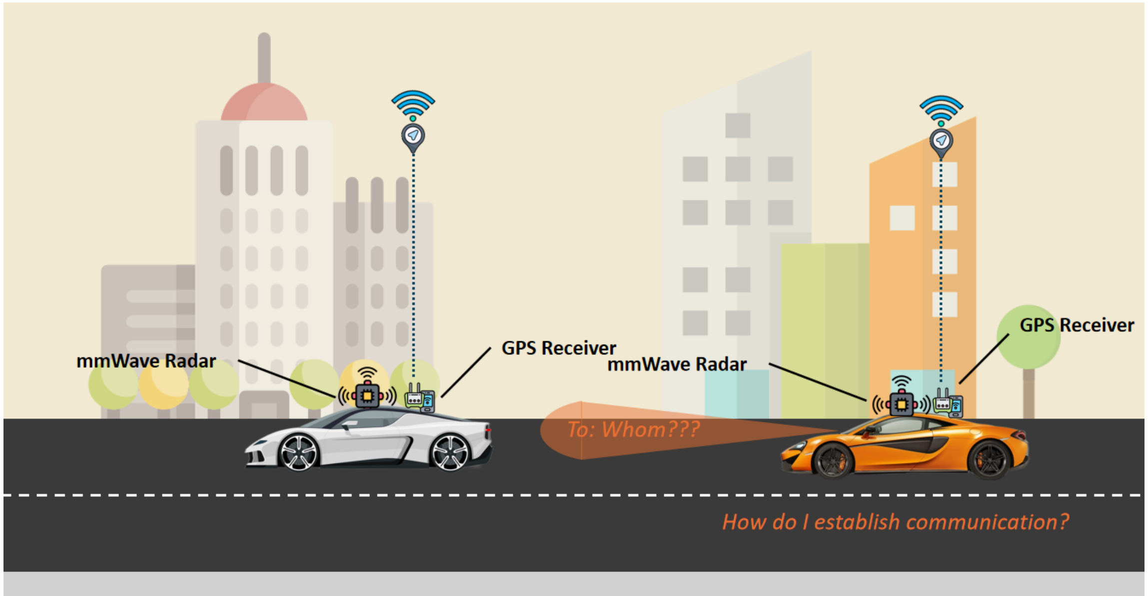

Figure 4.1: 2 AVs try to establish communication

network, all nodes within the network can identify each other's communication addresses and IDs, facilitating seamless data exchange. Furthermore, ad hoc networks significantly improve the communication link success rates. Once an ICA node joins the network, part of the communication resources are allocated by protocol to maintaining network activity, monitoring node status, dynamically adjusting the topology, and managing node entries and exits. This ensures that data transmission between nodes within the network remains stable, reducing link failure rates and enhancing the reliability of IDC by minimizing packet loss. Additionally, the wireless ad hoc network simplifies node management and coordination, particularly in dynamic environments like autonomous driving, where AVs frequently join and leave the network. By automatically identifying and managing nodes involved in the IDC process, the network can quickly determine the number of AVs participating in the consensus, ensuring the efficiency and accuracy of joint decision-making. Moreover, ad hoc networks offer greater scalability and flexibility, adapting to CAS applications of varying scales while optimizing system performance.

However, the current WIDCS has not yet taken into account the function of ad hoc networks, which may face certain challenges in real-world deployment. Specifically, factors such as network establishment time, stability, and the complexity of node management will directly affect the system's performance in practical applications. If wireless ad hoc networks cannot be efficiently established and maintained, IDC performance will be significantly degraded, compromising the reliability of CAS in dynamic and complex environments. Few studies [13, 83, 112–114] have explored how transmission modes, such as unicast, broadcast, and varying wireless communication link conditions, influence the reliability of DC. However, limited attention has been given to the impact of wireless networks on DC performance, and there is a lack of mathematical models developed for analysis based on specific scenarios. Therefore, future research must expand the WIDCS design to encompass the processes of network formation, maintenance, routing management, and dissolution to improve its feasibility. Additionally, thorough analysis and verification in real-world CAS scenarios will be necessary to ensure its practical applicability and performance. Furthermore, the first-generation WIDCS hardware module, AIR-RAFT, also lacks a networking protocol capable of supporting ad hoc networks for joint decision-making. Consequently, iterations of WIDCS hardware modules with integrated wireless network functions will also be necessary.

This chapter advances the Intelligent-Raft-based WIDCS by incorporating the capability to establish and manage wireless ad hoc networks, expanding upon its original functionality. We have also developed and implemented the second-generation WIDCS module, called RaBee, which integrates both hardware and software within an embedded framework. RaBee allows distributed nodes to achieve Intelligent-Raft consensus through a ZigBee ad hoc network. Additionally, Chapter 2 demonstrated that AVs can use WIDCS to reach consistent joint decisions, improving safety at uncontrolled intersections. However, further scenario testing is necessary to verify WIDCS's feasibility and optimization, as well as to devise tailored traffic solutions. This chapter focuses on utilizing the enhanced WIDCS to tackle the challenges posed by autonomous driving in on-ramp merging scenarios. We propose the 'Waiting for Insert Count' method, supported by the WIDCS, to enhance on-ramp merging safety for AVs. This approach facilitates reliable and efficient collaborative control, ensuring safer and more coordinated merging processes. We establish an analytical reliability model for both centralized and distributed joint-decision systems operating under a wireless ad hoc network. We also derive and analyze the safety of AVs under conditions with different joint decision-making schemes and different on-ramp merging guidance schemes based on the established mathematical model. Through the utilization of the RaBee platform and integration with practical AVs, we recreate the experimental on-ramp merging scenario and successfully demonstrate AVs collaborative merging. The experimental data and simulation results substantiate the accuracy of our theoretical analysis, confirming that WIDCS significantly enhances the safety of autonomous driving in on-ramp merging scenario.

4.2 Literature Review of CAV in On-ramp Merging

Researchers around the world have been developing various CAV applications to address traffic-related issues and improve efficiency and safety in specific traffic scenarios, such as highway on-ramp merging. On-ramp merging is a frequently encountered traffic scenario whose improper handling might cause heavy traffic congestion even accidents [115, 116]. It could form a traffic bottleneck since the merging vehicles may have to slow down or even stop at the ramp to wait for a proper opportunity to merge. The AVs on the highway should also carefully accommodate vehicle speeds and positions to avoid collision with the merging vehicles from the on-ramp, which has high traffic safety hazards. In addition, frequent decelerations and stops may also increase the fuel consumption and travel time of on-ramp vehicles, and reduce merging efficiency [117].

Research on on-ramp merging strategies generally follows two fundamental procedures: establishing the merging sequence (MS) and coordinating vehicle trajectories, typically in that order. The MS defines the specific order in which vehicles from the main road and the ramp should merge. The primary objective of the second step, vehicle trajectory coordination, is to adjust vehicle speeds to ensure a smooth merging process, thereby avoiding disruptions to traffic flow. While strategies in the literature share common features, the specific algorithms used to implement these procedures often differ.

Conventional MS is governed by predefined rules, often referred to as ad hoc negotiation-based strategies. As vehicles approach a junction, they generate short-term schedules and engage in bilateral negotiations with the Roadside Agent. Initial plans are frequently rejected due to timing conflicts, making rescheduling necessary. This strategy follows a first-in, first-out approach, which allows adjustments but typically results in locally optimal solutions. Its advantage lies in quick response times, making it well-suited for complex traffic convergence scenarios. For instance, in [118], a rule-based decision system was used to manage merging control effectively in congested traffic conditions. Wang et al. [119] proposed three basic scenarios to determine the merging sequence for on-ramp vehicles.

The MS can also be determined through a specific function that considers multiple criteria, such as safety and priority. In this case, the MS is generated by an upper-level controller, while a lower-level controller manages the merging maneuvers for each vehicle. As early as 1969, Athans [120] formulated the merging problem as an optimal control problem for a given sequence, evaluating all possible sequences and selecting the optimal one. Since then, several approaches have been developed to create the MS. Li et al. [121] described the solution space of all feasible driving schedules using a spanning tree, based on vehicle safe-passing order. Cao et al. [122] generated the MS based on prescribed merging points, while Jing et al. [123] sought the optimal MS by formulating a cost function using Game Theory.

The objective of vehicle motion control during merging is to create an adequate gap between two consecutive vehicles on the highway's main road and to accurately guide the merging vehicle into this gap. Several factors influence the merging process, including safety, passenger comfort, traffic flow efficiency, energy consumption, and time delay. Improved coordination between both mainline and on-ramp vehicles leads to a more efficient and seamless merging process.

To achieve efficient merging, both centralized and decentralized control methods can be employed. In a centralized approach, vehicle trajectories are uniformly generated by the Roadside Agent. Awal et al. [124] focused on minimizing merging time, with particular emphasis on optimizing the average merging speed. Rios Torres et al. [117] developed an optimization framework and a closed-form analytical solution for real-time coordination of CAVs at on-ramp merging zones. Xie et al. and Cao et al. [125] applied the C/GMRES method to solve the optimal merging path problem, with their system considering the states of both ramp and main road vehicles, making it a centralized approach. Additionally, Ito et al. [126] designed a global controller to facilitate a smooth merging process under mixed traffic conditions.

Decentralized control approaches operate locally among vehicles using vehicle-to-vehicle (V2V) communication, thereby reducing the communication burden compared to centralized methods. The distributed control protocol proposed by Dao et al. [127] focused on assigning vehicles into platoons to improve traffic safety and increase lane capacity. Lu et al. [128] introduced a vehicle longitudinal control algorithm based on predecessor-follower (PF) information flow topology and formulated the merging problem differently depending on the road's geometric layout (i.e., with or without a parallel lane). Xu et al. [129] investigated the impact of controlled vehicle rates on highway traffic and found that vehicles equipped with Adaptive Cruise Control (ACC) or Cooperative Adaptive Cruise Control (CACC) systems show significant potential for improving coordination during highway merging.

Solving these issues necessitates the exchange of information, and advanced communication technologies, such as V2V and V2I communication, are essential for enabling cooperative decision-making and motion control in AVs. However, many researches focus on path planning or control algorithms based on MS, and there are very few and superficial optimization studies on communication protocol, networking, consensus consistency, etc.

in on-ramp merging scenario.

4.3 WIDCS with Ad Hoc Network

The ad hoc network function of WIDCS ensures effective consensus and joint decision-making in complex, dynamic environments. The following detail each key function point and the process is shown in Fig. 4.2:

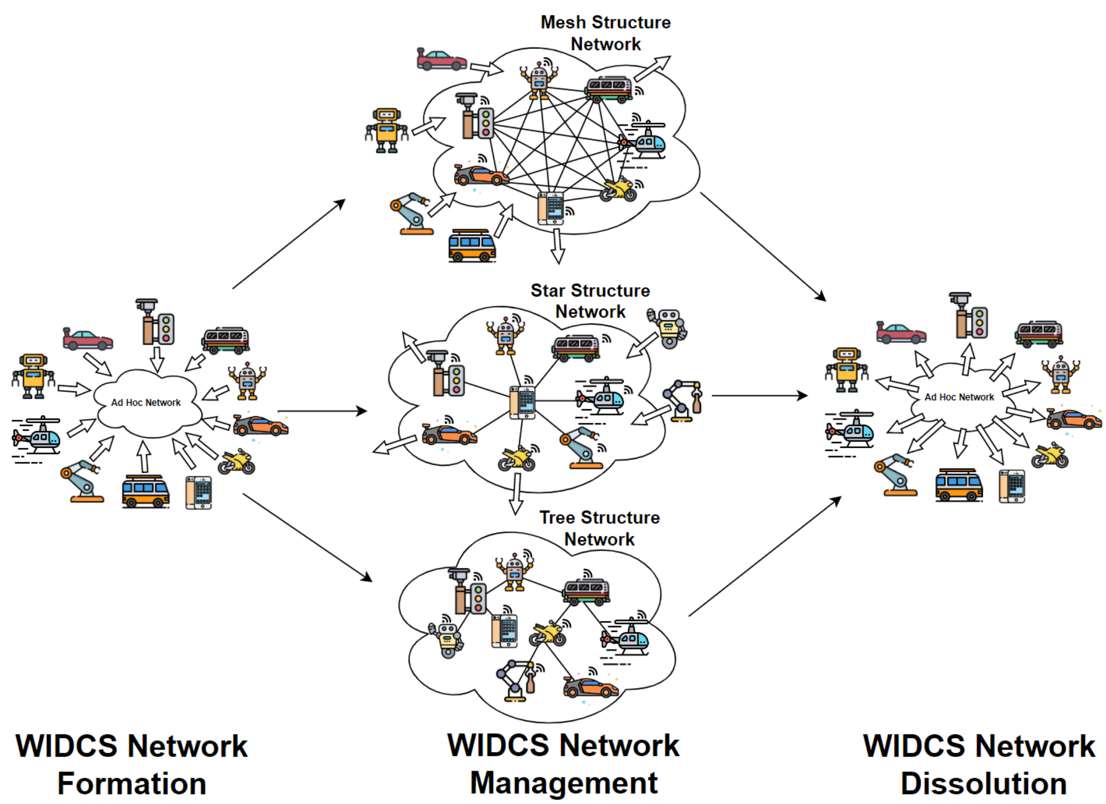

Figure 4.2: The process of WIDCS ad hoc network formation, different topologies, and network dissolution under task-driven.

4.3.1 Formation of WIDCS Ad Hoc Network

The initiation and formation of the WIDCS ad hoc network form the foundation for the entire distributed consensus process. When a WIDCS node needs to initiate consensus (such as AVs entering an intersection), the system first detects nearby nodes via wireless communication and initiates the ad hoc network's formation. The initiating node broadcasts a "network establishment request" to determine if surrounding nodes are available and capable of joining the network. Nodes receiving the request will respond, providing their current status and resource information. All nodes that respond positively become part of the initial network and establish a connection through a networking protocol, such as ZigBee or Wi-Fi. At this point, the ad hoc network is formed, and nodes can exchange their IDs, addresses, and resource information, laying the groundwork for the subsequent consensus process.

4.3.2 Management of WIDCS Ad Hoc Network

WIDCS network management is composed of two key components: node management and structure management. In node management, new nodes must first support WIDCS functions before joining the ad hoc network. A joining node broadcasts a request to the existing network, and the network management node verifies this request through a security authentication protocol to ensure compliance with network requirements. When a node leaves, its exit must be confirmed by the network management node to prevent disruption to the consensus process. The network topology is automatically updated to reflect this change, reallocating communication resources and adjusting routing tables accordingly. Additionally, WIDCS includes a node status detection function that monitors each node's operational status in real time, such as signal strength, battery level, and activity state. If a node's condition deteriorates, the network activates a recovery mechanism to prevent node failure during the consensus process.

Network management plays a critical role in ensuring the efficient and reliable operation of the WIDCS ad hoc network. After the network is established, communication paths and topologies between nodes dynamically evolve. WIDCS employs an adaptive routing protocol that updates routing paths in response to changes in the network environment, such as the addition or departure of nodes or changes in their status. Each node regularly sends status updates according to the network management protocol to ensure all nodes maintain the latest routing tables. When a path becomes unavailable due to node failure or environmental interference, the WIDCS network management function rapidly identifies an alternative route and re-plans data packet transmission to maintain continuous and efficient communication. Additionally, WIDCS supports various topologies, including star, tree, and mesh structures, providing the flexibility needed to adapt to a dynamic environment. For instance, in traffic scenarios, the changing speed and position of vehicles require real-time updates to the network structure to ensure the consensus process proceeds smoothly.

4.3.3 Dissolution of WIDCS Ad Hoc Network

The dissolution of the ad hoc network must be orderly and secure. After consensus or joint decision-making is completed, nodes will exit the network gradually based on their current status and requirements. The dissolution process begins when the network management node sends a dissolution signal to notify all participating nodes to prepare for network exit. Subsequently, each node sequentially disconnects its communication links with other nodes and updates its internal state to mark the network's termination. In cases where dissolution is necessary due to a deteriorating communication environment or node failure, WIDCS includes an emergency dissolution mechanism. In such situations, nodes quickly disconnect and enter a low-power state to conserve resources. Ensuring a smooth network dissolution optimizes resource usage and avoids unnecessary communication overhead.

4.4 Design and Implementation of RaBee

4.4.1 RaBee Hardware Architecture

We leveraged embedded technology to design and implement 'RaBee', as shown in Fig. 4.3, a fully functional WIDCS utilizing the Intelligent-Raft algorithm and ZigBee, enabling us to practically evaluate IDC's performance. The hardware architecture of RaBee, as depicted in Fig. 4.4, consists of six main components: the power unit, Micro-Controller Unit (MCU), RAM expansion unit, ROM expansion unit, human-computer interaction (HCI) unit, wireless module unit, and communication interface. The STM32F407ZGT6 chip, featuring a powerful Cortex-M4 core with a main frequency of up to 168MHz, is selected as the MCU. To accommodate the substantial RAM requirements for processing node information, status parameters, and large network data, an additional 1MB of RAM chip has been added. An SD card serves as a ROM expansion to provide ample space for recording processed consensus data. The OLED screen and buttons function as HCI units for displaying or setting parameters and monitoring consensus progress. The DIGI XBee3 series model XB3-24Z8UM ZigBee module, operating in the 2.4GHz ISM band, is used for wireless communication. Various interfaces are reserved to support different applications.

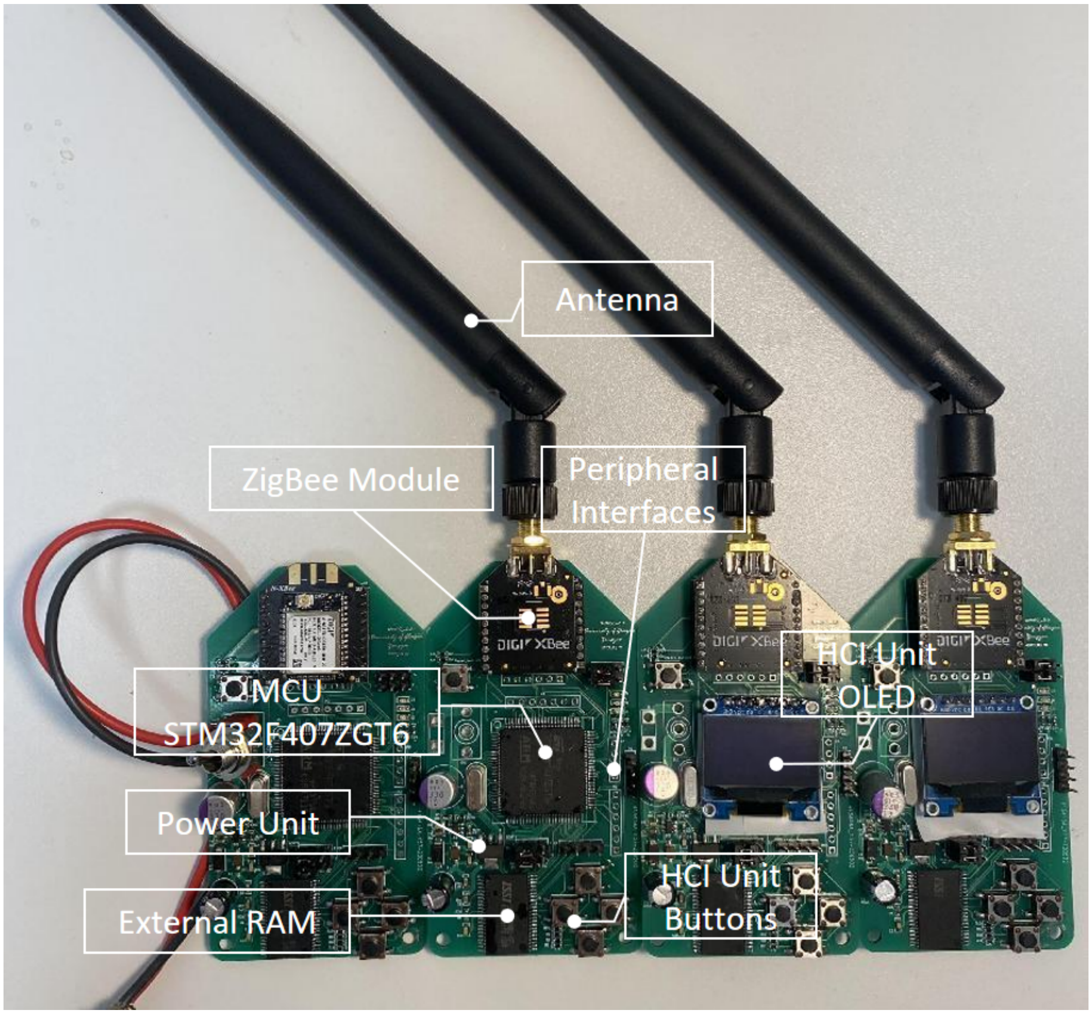

Figure 4.3: The hardware module of RaBee (4 copies)

4.4.2 RaBee Software Architecture

Fig. 4.5 illustrates the software system framework of RaBee. The Hardware Abstraction Layer (HAL) provides drivers for basic MCU hardware resources, such as the UART used by the ZigBee module. The Scheduling Service acts as the hardware manager, handling upper-layer drivers for hardware peripherals like the SD card, and processing hardware interrupts. The application layer comprises three main threads: UI interface management, communication packet management, and consensus status management.

4.4.3 RaBee Network Architecture

ZigBee is an advanced communication protocol based on the IEEE 802.15.4 standard, specially designed for Wireless Personal Area Networks (WPAN) in low-power IoT sce-

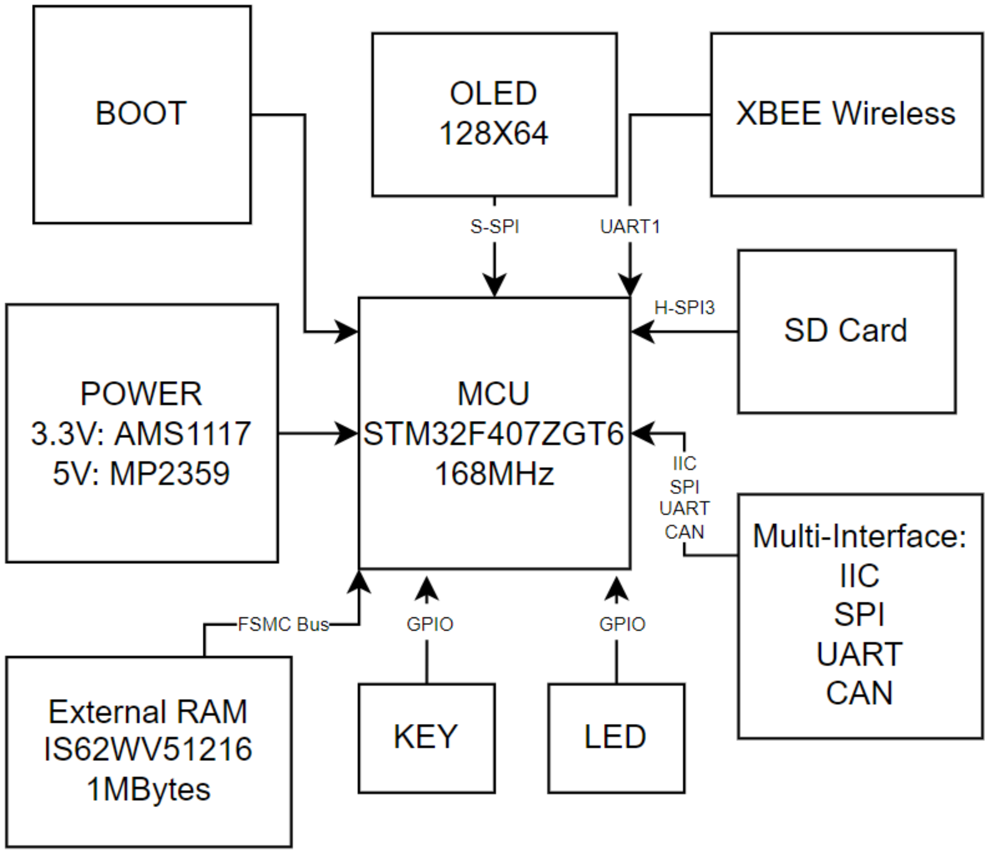

Figure 4.4: Hardware structure of RaBee

narios. The self-organizing capability of the ZigBee protocol stands out as a significant advantage, supporting diverse network topologies, and operating effectively in dynamically changing environments and situations necessitating adaptable deployment. Consequently, ZigBee can not only ensure a robust network environment for the upper-layer consensus mechanisms of WIDCS, but also exhibit substantial engineering viability.

The hierarchical architecture of the distributed consensus ad hoc network of RaBee is depicted in Fig. 4.6. Powered by ZigBee module, the IEEE 802.15.4 protocol defines the physical and MAC layers. The network layer utilizes the ZigBee protocol to establish the wireless distributed ad hoc network, configuring the network ID and allocating addresses to each node. At the application layer, the Intelligent-Raft manages the consensus process and status.

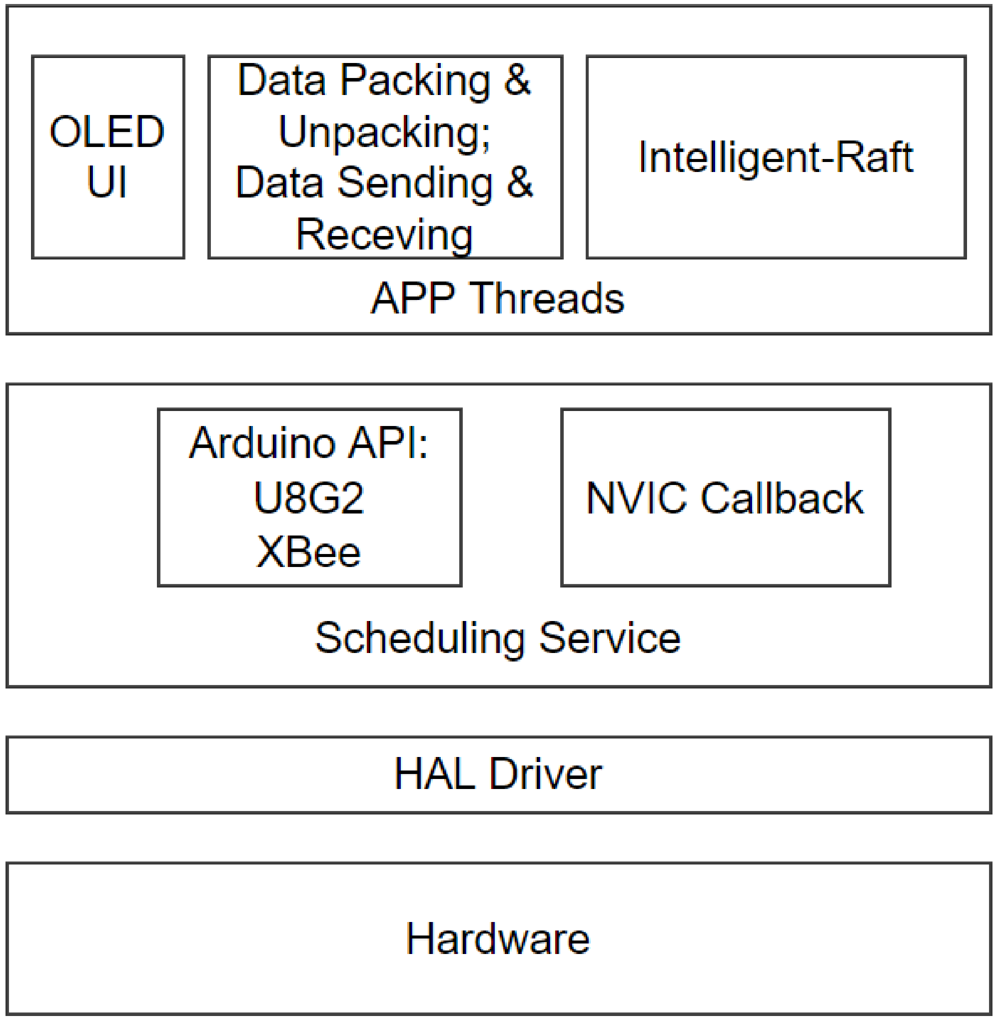

Figure 4.5: Software structure of RaBee

4.4.4 RaBee Workflow

The workflow of Intelligent-Raft is depicted in Fig. 4.7. Following the traditional Raft protocol, nodes are categorized into followers, candidates, and leaders [2]. Intelligent-Raft works by electing a leader in a cluster of nodes, where the elected leader receives all followers initiatives, and broadcasts the initiatives to all network nodes. The followers then assess the proposed data utilizing intelligent evaluation mechanism and inform the leader of feedback. Consensus is achieved when the number of feedbacks signaling agreement surpasses the 50% of the cluster nodes. This mechanism ensures fault tolerance, allowing the system to continue functioning normally as long as the majority of nodes remain operational. Moreover, each consensus require a collaborative assessment by mul-

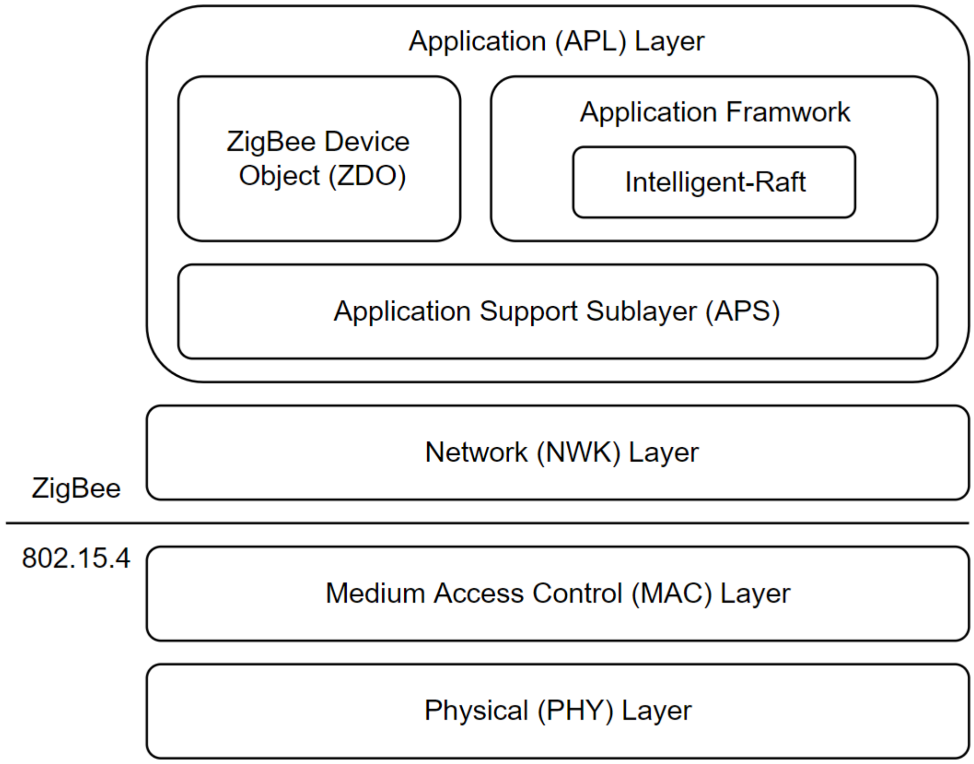

Figure 4.6: The hierarchical ad hoc network architecture of RaBee

tiple nodes on identical information, thereby ensuring synchronization and consistency across the network.

4.4.5 Potential Applications of RaBee

RaBee supports various distributed IoT application scenarios, and its extensive and diverse communication interfaces facilitate seamless integration with AV in-vehicle systems. Installing RaBee on an AV facilitates consensus within a wireless, distributed, reliable, and fault-tolerant network, allowing the synchronized data to integrate with the AV's local sensor data for comprehensive predictions and decisions before execution.

4.5 Secure On-Ramp Merging Scheme

In the realm of autonomous driving, WIDCS is capable of enhancing AV safety across various traffic scenarios through consistent distributed decision-making, with this section specifically examining the on-ramp merging scenario. On-ramp merging, a crucial highway scenario, poses significant risks for AVs as they transition from ramps to main roads.

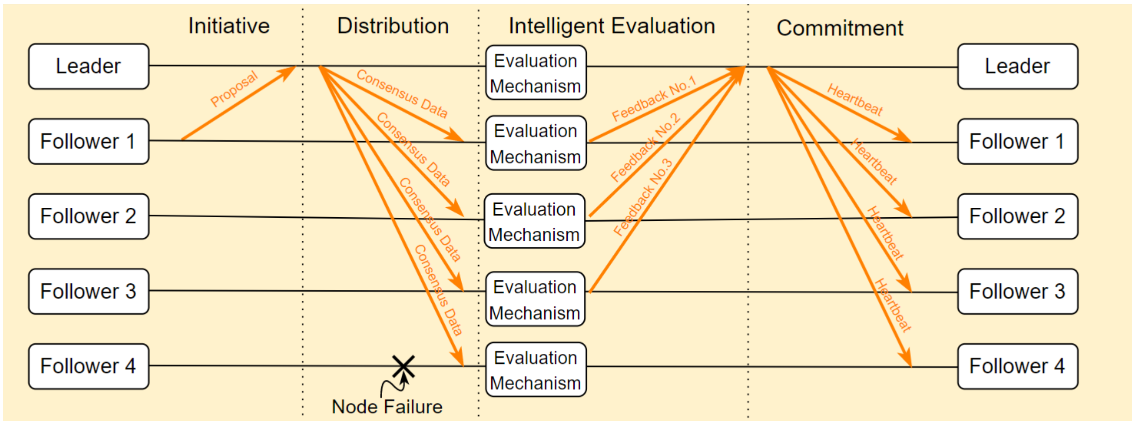

Figure 4.7: The workflow of Intelligent-Raft

This situation involves challenges like speed mismatches and blind spots that can obscure a vehicle's view, increasing the risk of accidents due to informational asymmetries. Moreover, uncertainty about right-of-way can cause congestion or collisions. Thus, ensuring safety in on-ramp merging scenarios depends crucially on guiding AVs to merge in an organized and systematic manner at a macro level.

As listed in Section 4.2, traditional methods like the MS have been developed to ensure safe passage in on-ramp merging scenarios. MS prescribes an Sequence Identification Numbers (SIDs) order for vehicles on both main and ramp roads, determined by a network proposer. This approach not only clarifies merging priorities but also improves the efficiency of vehicle passage in situations where visibility is compromised. However, traditional MS solutions encounter significant development challenges, notably the difficulty in ensuring consistency across all nodes. In addition, it is difficult for the vehicles behind to know the progress of the current merge, which limits the effectiveness of SID. Moreover, MS requires real-time updates and dissemination of SIDs, placing vehicle safety heavily on the reliability of the central node, increasing communication and computational demands, and have low fault tolerance.

To address these limitations, we develop a new method called Wait Insertion Count (WIC) based on the traditional MS approach. WIC quantifies the number of vehicles in adjacent lanes that an AV must wait for after the preceding vehicle in its own lane has passed. For instance, a WIC value of 0 allows an AV to follow the vehicle directly ahead through the merging area, while a non-zero WIC requires waiting for a specified number of vehicles in another lane to pass. Since WIC is static, the merging process for each cluster requires only a single allocation by the communication system in the cases without overtaking, thus reducing dependency on it.

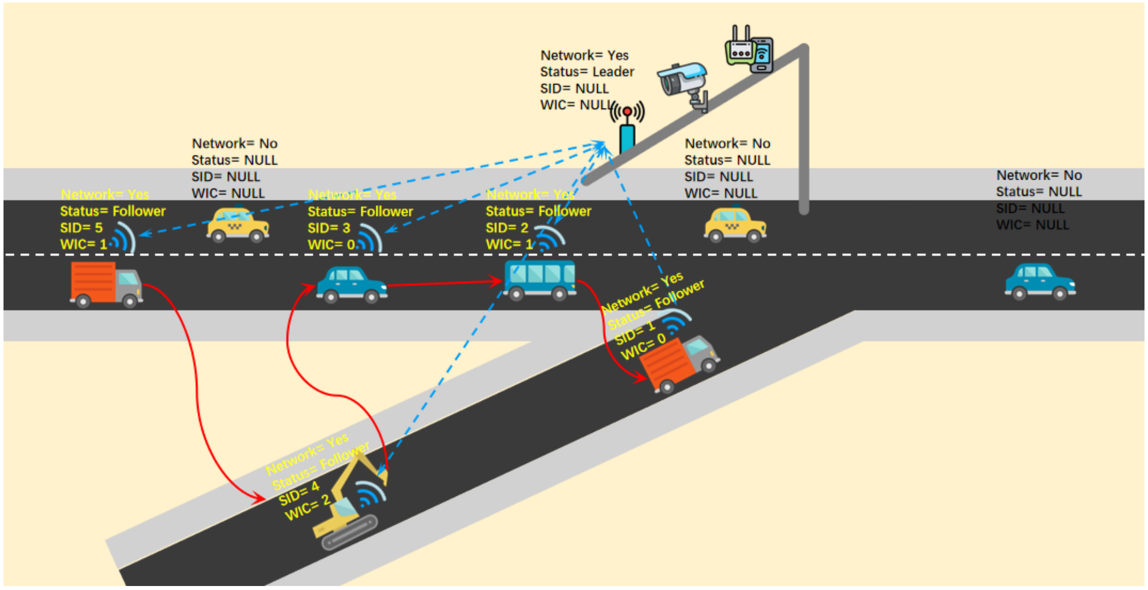

Figure 4.8: An illustration of the cooperative highway on-ramp merging scenario

We have also developed a comprehensive passage solution for on-ramp merging using WIC combined with WIDCS, as depicted in Fig. 4.8. As part of the network integration process, each vehicle uploads critical data such as speed, position, and the distance to the preceding vehicle to the network coordinator that could be either Follower or Leader. The coordinator calculates each vehicle's estimated time of arrival at the merging point, then determines the WIC and optimal speed to ensure safe merging. Following this, the coordinator compiles the speeds, positions, distances, and the recommended WIC and merging speed for each vehicle into the consensus data, initiating the consensus process. During the intelligent evaluation stage, each network node employs its own evaluation criteria to assess the accuracy of the proposed WIC and votes to confirm whether a consensus has been reached.

Once consensus is achieved, all AVs proceed efficiently through the merging area based on their WIC number of vehicles in another merging road they need to wait for. The red line in Fig. 4.8 indicates the order in which vehicles merge. For example, the vehicle on the ramp with a WIC of 2 must allow two vehicles from the main road to pass after its preceding vehicle has merged. Through the use of WIDCS, each vehicle verifies the global traffic order via an ad hoc network, diminishing reliance on central nodes.

4.6 Mathematical Models

In this section, we establish mathematical models for the reliability of joint decision-making by a centralized system and WIDCS in ad hoc networks. We first obtain the probability distribution model of SINR by analyzing the free space path loss (FSPL) of AV's signal transmission power in the on-ramp merging scenario, and use it to analyze the communication link success rate. On this basis, we respectively analyze the probability of AVs safely merging into the on-ramp area under two schemes: one assisted by SIDs within the centralized decision system, and the other supported by WIC within the WIDCS. **In our model, the probability values derived from the model serve as indicators to represent the reliability of different communication modes and the safety of AVs.** Several assumptions are established to guide the development of the model.

1. **Assumption 1:** Since the method of determining traffic order significantly affects the safety of each AV, the likelihood of an AV safely navigating the on-ramp merging area varies across different joint decision-making systems.

2. **Assumption 2:** The scenario is only considered safe when all AVs navigate through the on-ramp merging area without incident.

3. **Assumption 3:** Overtaking behavior during the merging process is not considered.

4. **Assumption 4:** Given that the Roadside Unit (RSU) can be reliably deployed in the on-ramp area, it is responsible for establishing and managing the entire network.

5. **Assumption 5:** Joint decision-making require a minimum of three nodes in the network.

Based upon the assumptions, the frequently used notations are summarized in Table 4.1.

4.6.1 Communication Link Success Rate Model

SINR is a key parameter in wireless communications, which is used to measure the ratio of signal strength to interference and background noise strength. The higher the SINR, the better the quality of the communication link is usually, and the lower the error rate of data transmission. So we can analyze the communication link success rate through SINR, which can be expressed as:

Table 4.1: Frequently Used Notations

Notation	Definition
P_{S_Nor}	Probability of each AV passing the on-ramp merging area safely without V2X communication
P_{S_Net}	Probability of each AV passing the on-ramp merging area safely with the assistance of wireless network
P_{S_SID}	Probability of each AV passing safely under the auxiliary conditions of centralized decision-making and SID
P_{S_WIDCS}	Probability of each AV passing safely under the auxiliary conditions of WIDCS and WIC
N_{MAX}	Number of vehicles that pass through the on-ramp merging area
n_{net}	Number of nodes that successfully joined the network
n_{on}	Number of nodes that have successfully joined the network and are online
p_{signal}	Signal power at the receiver
p_{noise}	Environmental noise power
$p_{interfe}$	Environmental interference power
p_{trans}	Signal power at the transmitter
p_{link}	Probability of successful link transmission
P_{NF}	Probability of exactly n_{net} nodes among N_{MAX} nodes successfully join the network.
P_{CenSta}	Reliability of the centralized network
P_{DisSta}	Reliability of the distributed network
P_{CenDec}	Probability of successfully completing a centralized decision
P_{DisDec}	Probability of successfully completing an Intelligent-Raft consensus

$$\mathrm{SINR}_{dB} = 10\log_{10}(\frac{P_{signal}}{P_{noise} + P_{interference}}) \tag{4.1}$$

Based on the communication scheme we proposed, all nodes will try to access the Zigbee network and complete Raft consensus collaboration within the network. So we assume that $P_{noise} + P_{interference}$ is a constant, and we introduce a FSPL model for P_{signal}.

$$FSPL_{dB} = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}(\frac{4\pi}{c}) \tag{4.2}$$

where $FSPL(dB)$ represents free space path loss in decibels (dB), d is the propagation distance, f is the signal frequency, c is the speed of light.

Next, we can derive the relationship between propagation distance and SINR through the SINR and FSPL models:

$$P_{signal}(dB) = P_{trans}(dB) - FSPL_{dB} \tag{4.3}$$

$$\begin{aligned}&\mathrm{SINR}_{dB}\\&= 10\log_{10}(P_{trans}) - FSPL_{dB} - 10\log_{10}(P_{noise} + P_{interfe})\\&= 10\log_{10}(\frac{P_{trans}}{d^2 f^2 (\frac{4\pi}{c})^2 (P_{noise} + P_{interfe})})\end{aligned} \tag{4.4}$$

Next, we only need to know the communication distance distribution model between vehicles to derive the SINR distribution model. The number of vehicles passing the observer per unit of time is a Poisson process with mean λ which means the traffic flow is λ (in vehicles per hour) [111]. There are K discrete levels of constant speed $v_i (i = 1, ..., K_{vel})$ on each lane where the speeds are independent identically distributed (i.i.d.). Denote the rate of arrivals of vehicles at each level of speed as $\lambda_i (i = 1, ..., K_{vel})$ where $\sum_{i=1}^{K_{vel}} \lambda_i = \lambda$, thus, the occurrence probability of each speed level is $P_i = \lambda_i / \lambda$. Thus, the distance between AVs on each lane with observer as the origin obeys the exponential distribution with parameter $\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i}$. So the cumulative distribution function (CDF) of intervehicle distance is:

$$F_L(D) = \begin{cases} 0, & \text{if } l < 0 \\ 1 - e^{-\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i} d}, & \text{if } l \geq 0 \end{cases} \tag{4.5}$$

Where D is the distance between vehicles. The distances between each vehicle are independent.

So we can get the distribution of SINR_{dB} through the distribution of the derived relational function of random variable d. According to the definition, we assume that when SINR_{dB} is greater than a certain threshold, its distribution function value is used as the communication link success rate of the system:

$$
\begin{aligned}
p_{link} &= P\{\mathrm{SINR}_{dB} \geq \mathrm{SINR}_{threshold}\} = \\
&P\{10\log_{10}(\frac{P_{trans}}{d^2 f^2 (\frac{4\pi}{c})^2 (P_{noise}+P_{interfe})}) \geq \mathrm{SINR}_{threshold}\} \\
&= P\{d \leq \frac{P_{trans}^{\frac{1}{2}} 10^{-\frac{\mathrm{SINR}_{threshold}}{20}}}{\frac{4\pi f}{c}(P_{noise}+P_{interfe})^{\frac{1}{2}}}\} \\
&= 1-\exp(-\lambda \sum_{i=1}^{K_{vel}} \frac{P_i}{v_i} \frac{P_{trans}^{\frac{1}{2}} 10^{-\frac{\mathrm{SINR}_{threshold}}{20}}}{\frac{4\pi f}{c}(P_{noise}+P_{interfe})^{\frac{1}{2}}})
\end{aligned}
\tag{4.6}
$$

4.6.2 Safety model without Communication Support

In scenarios lacking communication, all AVs must depend solely on their own sensors and environment exploration to navigate. Under these conditions, with a total of N_{MAX} vehicles on both the main and ramp roads passing through the on-ramp area, the system's overall safety probability is defined as follows:

$$
P_{Pass\_Safe\_NoV2X} = P_{S\_Nor}^{N_{MAX}} \tag{4.7}
$$

where safety is assessed by the successful merging of all AVs through the area.

4.6.3 Safety Model with SID in Centralized Decision System

We first analyze the reliability of the central decision-making system within an ad hoc network, noting that centralized decisions are feasible only when the network is stable. The reliability of such a network relies heavily on a single central node and is significantly affected by the number of nodes it serves. Therefore, we assume that the reliability of the centralized network conforms to the exponential decay model:

$$
P_{CenSta} = P_{CenSta\_min} + (1-P_{CenSta\_min}) * e^{-k*n_{net}} \tag{4.8}
$$

where k is the decay constant and P_{CenSta_min} is the minimum value for central network reliability.

In centralized decision-making system, all nodes receive and unconditionally execute the decisions issued by the central node. Therefore, the reliability of centralized decision-making is:

$$P(CenDec|Network) = \frac{P(Network \cap CenDec)}{P(Network)} \tag{4.9}$$

$P(Network)$ in (4.9) represents the probability of successfully establishing the network. In the dynamic environments, AVs' joint decision-making necessitates the formation of a wireless ad hoc network, with a designated coordinator node responsible for network establishment and maintenance. When AVs enter the coordinator's range, they attempt to respond to networking requests based on **Assumption 4**. The coordinator makes up to X attempts per AV, with one successful response ensuring network entry. Our model assumes a minimum of three nodes for decision-making, necessitating at least three nodes in the network.

$$P(Network) = \sum_{n_{net}=3}^{N_{MAX}} (P_{NF}(N = n_{net})) \tag{4.10}$$

$$\begin{aligned} &P_{NF}(N = n_{net}) \\ &= \binom{N_{MAX}}{n_{net}} P_{forming}^{n_{net}} (1 - P_{forming})^{N_{MAX} - n_{net}} \end{aligned} \tag{4.11}$$

$$P_{forming} = \sum_{i=1}^{X} ((1 - p_{link})^{i-1} p_{link}) \tag{4.12}$$

where $P_{forming}$ represents the probability of a node successfully joining the network within X networking attempts. Based on **Assumption 5**, $P(Network)$ can be calculated by combining (4.10), (4.11) and (4.12).

$P(Network \cap CenDec)$ indicates the probability of successfully establishing a stable centralized network where centralized decisions are made concurrently, as calculated by (4.13) and (4.14).

$$\begin{aligned} &P(Network \cap CenDec) \\ &= \sum_{n_{net}=3}^{N_{MAX}} (P_{NF}(N = n_{net}) \times P_{CenSta} \times P_{CenDec}) \end{aligned} \tag{4.13}$$

$$P_{CenDec} = \sum_{n_{on}=0}^{n_{net}} (\binom{n_{net}}{n_{on}} p_{link}^{n_{on}} (1 - p_{link})^{n_{net} - n_{on}}) \tag{4.14}$$

Next, we determine the safety probability for all AVs navigating under a centralized network with SID using the conditional probability formula:

$$\begin{aligned}&P(Pass\_Safe\_All|(Network \cap SID\_UD))\\&= \frac{P(Pass\_Safe\_All \cap Network \cap SID\_UD)}{P(Network \cap SID\_UD)}\end{aligned} \tag{4.15}$$

$P(SID\_UD)$ denotes the aggregate probability of successful SID updates by the communication system for each vehicle merge. In the traditional MS method, a single update per merge is essential to ensure all vehicles are apprised of the current merging progress. $P(Network \cap SID\_UD)$ denotes the probability of both successfully establishing a centralized network and updating the SID on each merge, as derived from (4.16) and (4.17).

$$\begin{aligned}&P(Network \cap SID\_UD)\\&= \prod_{n_{max}=3}^{N_{MAX}} \sum_{n_{net}=0}^{n_{max}} (P_{NF}(N = n_{net}) \times P_{CenSta} \times P_{SID\_UD})\end{aligned} \tag{4.16}$$

$$P_{SID\_UD} = \sum_{n_{on}=0}^{n_{net}} (\binom{n_{net}}{n_{on}} p_{link}^{n_{on}} (1 - p_{link})^{n_{net}-n_{on}}) \tag{4.17}$$

$P(Pass\_Safe\_All \cap Network \cap SID\_UD)$ represents the likelihood that successful network formation and centralized SID updates coincide with the safe passage of all AVs. For the AVs within the network, the safety probability is denoted as P_{S_Net}, while for those unable to join, it is P_{S_Nor}. Here we accumulate all possible cases by combining (4.18), (4.19), and (4.20):

$$\begin{aligned}&P(Pass\_Safe\_All \cap Network \cap SID\_UD)\\&= \prod_{n_{max}=3}^{N_{MAX}} \sum_{n_{net}=0}^{n_{max}} (P_{NF}(N = n_{net}) \times P_{CenSta} \times P_{Pass\_Safe\_SID})\end{aligned} \tag{4.18}$$

$$\begin{aligned}P_{Pass\_Safe\_SID} = &\sum_{n_{on}=0}^{n_{net}} (\binom{n_{net}}{n_{on}} p_{link}^{n_{on}} (1 - p_{link})^{n_{net}-n_{on}}\\&* P_{Pass\_Safe1})\end{aligned} \tag{4.19}$$

$$\begin{aligned}P_{Pass\_Safe1} = &\frac{P_{S\_SID} * (n_{on})}{n_{max}} + \frac{P_{S\_Net} * (n_{net} - n_{on})}{n_{max}}\\&+ \frac{P_{S\_Nor} * (n_{max} - n_{net})}{n_{max}}\end{aligned} \tag{4.20}$$

The probability of AVs passing safely under the condition of centralized SID updates with network can be determined by combining (4.16) and (4.18).

4.6.4 Safety model with WIC in WIDCS

In a distributed network, the reliability does not limited to a single node but rather on the collective reliability of all nodes involved. We assume that $P_{DisNode}$ is the reliability of each node, therefore:

$$P_{DisSta} = 1 - (1 - P_{DisNode})^{n_{net}} \tag{4.21}$$

We assume that achieving an Intelligent-Raft consensus must be based upon the establishment of the network, the reliability of distributed decision-making is:

$$P(DisDec|Network) = \frac{P(Network \cap DisDec)}{P(Network)} \tag{4.22}$$

$P(Network \cap DisDec)$ denotes the likelihood of successfully establishing a stable distributed network and achieving Intelligent-Raft consensus, as calculated by (4.23).

$$\begin{aligned} &P(Network \cap DisDec) \\ &= \sum_{n_{net}=3}^{N_{MAX}} (P_{NF}(N = n_{net}) \times P_{DisSta} \times P_{DisDec}) \end{aligned} \tag{4.23}$$

P_{DisDec} in (4.23) represents the probability of reaching an Intelligent-Raft consensus, a process that entails the leader sending messages via downlink, as defined in (4.24), and receiving responses through uplink, as specified in (4.25).

$$P_{Downlink} = \binom{n_{net}-1}{m_1} p_{link}^{m_1} (1 - p_{link})^{n_{net}-1-m_1} \tag{4.24}$$

$$P_{Uplink} = \binom{m_1}{m_2} p_{link}^{m_2} (1 - p_{link})^{m_1-m_2} \tag{4.25}$$

The probability of achieving a successful Intelligent-Raft is defined by the leader's ability to receive more than $(N_{MAX}-1)/2$ uplink messages from followers. Therefore, m_2 in (4.25) and (4.26) must not be less than $(N_{MAX}-1)/2$.

$$\begin{aligned} P_{DisDec} = &\sum_{m_1=\lceil \frac{n_{net}-1}{2} \rceil}^{n_{net}-1} (\binom{n_{net}-1}{m_1} (1 - p_{link})^{n_{net}-1-m_1} \\ &p_{link}^{m_1} \sum_{m_2=\lceil \frac{n_{net}-1}{2} \rceil}^{m_1} (\binom{m_1}{m_2} p_{link}^{m_2} (1 - p_{link})^{m_1-m_2})) \end{aligned} \tag{4.26}$$

where we consider all possible situations in which Intelligent-Raft consensus can be completed when there are N_{MAX} AVs that need to pass.

Then, we determine the probability of AVs safely navigating the on-ramp merging area with assistance from WIDCS and WIC, employing a conditional probability method:

$$\begin{aligned} &P(Pass\_Safe\_All|(Network \cap DisDec)) \\ &= \frac{P(Pass\_Safe\_All \cap Network \cap DisDec)}{P(Network \cap DisDec)} \end{aligned} \tag{4.27}$$

$P(Pass\_Safe\_All \cap Network \cap DisDec)$ quantifies the probability of simultaneous successful network formation, IDC regarding WIC allocation, and safe passage for all AVs. Thus, we get:

$$\begin{aligned} &P(Pass\_Safe\_All \cap Network \cap DisDec) \\ &= \sum_{n_{net}=3}^{N_{MAX}} (P_{NF}(N = n_{net}) \times P_{DisSta} \times P_{Pass\_Safe\_DisDec}) \end{aligned} \tag{4.28}$$

$P_{Pass\_Safe\_DisDec}$ consider all potential AV passage cases across different numbers of network nodes and their consensus outcomes. Therefore:

$$\begin{aligned} &P_{Pass\_Safe\_DisDec} \\ &= \sum_{m_1=\lceil \frac{n_{net}-1}{2} \rceil}^{n_{net}-1} (\binom{n_{net}-1}{m_1} p_{link}^{m_1} (1-p_{link})^{n_{net}-1-m_1} \\ &\sum_{m_2=\lceil \frac{n_{net}-1}{2} \rceil}^{m_1} (\binom{m_1}{m_2} p_{link}^{m_2} (1-p_{link})^{m_1-m_2} P_{Pass\_Safe2})) \end{aligned} \tag{4.29}$$

$$P_{Pass\_Safe2} = P_{S\_WIDCS}^{m_2} P_{S\_Net}^{n_{net}-m_2} P_{S\_Nor}^{N_{MAX}-n_{net}} \tag{4.30}$$

The probability of AVs passing safely under the condition of IDC with distributed network can be determined by combining (4.23) and (4.28).

4.7 Simulations

In this section, we conduct simulations on the safety model of on-ramp merging under three distinct scenarios: no communication, SID scheme within centralized decision system, and WIC scheme within WIDCS. Moreover, we compared the reliability of two different decision-making systems in ad hoc networks through simulation.

4.7.1 Wireless Link Success Rate

According to Eq. (4.6), we derive the link success rate when the SINR strength at the receiving node exceeds a specified threshold, using this value as the link success rate parameter p_{link} for subsequent simulations. In the FSPL model adhering to the IEEE 802.15.4 protocol, the physical layer operates in the 2.4 GHz frequency band with f set to 2.4 GHz. For the SINR models in Eq. (4.1) and Eq. (4.3), we set the transmission power P_{trans} at 50 dBm and the combined noise and interference power $P_{noise} + P_{interfe}$ at 10 dBm. We then evaluate the vehicle distance distribution model in Eq. (4.5), assuming a vehicle flow rate of 30 to 60 veh/min and an average vehicle speed of 40 km/h with a standard deviation of 25 km/h and the number of speed level K_{vel} is 7.

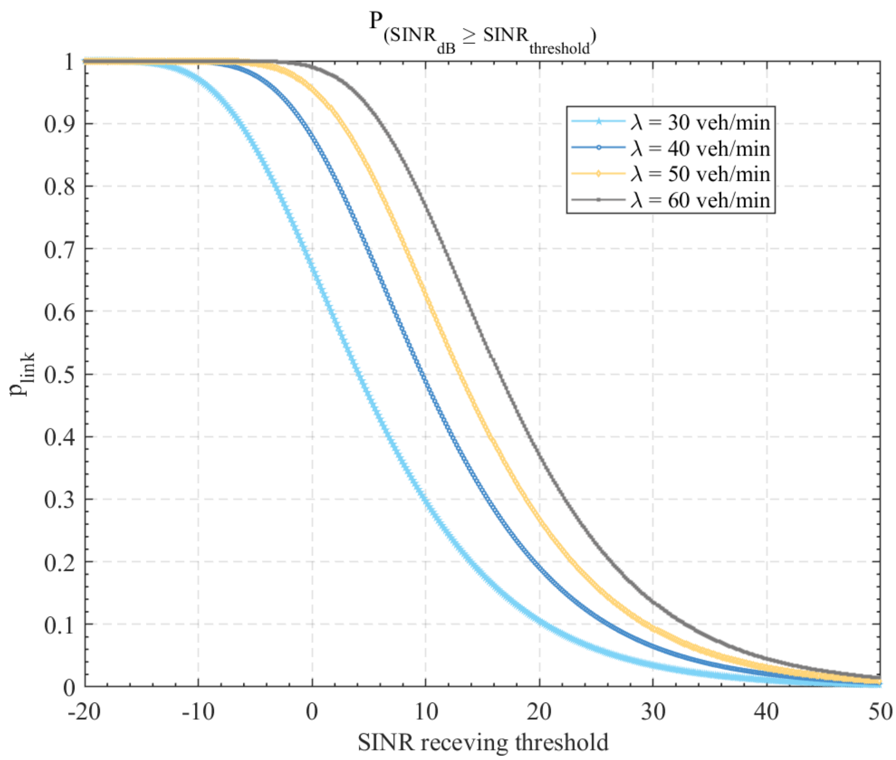

Figure 4.9: Relationship between the preset SINR threshold and wireless link success rate

Fig. 4.9 shows the relationship curve between the communication link success rate and the preset SINR threshold. It can be seen that as the SINR threshold is set higher, it will become more and more difficult to reach such SINR for each data signal, and the success rate of the communication link will naturally decrease. Here we assume that the receiving SINR threshold is -10 dbm, and the communication link success rate is 0.99. We can use this value as a reference for subsequent Simulations.

4.7.2 Decision-Making Reliability

Table 4.2: Summary of Parameters

Parameter	**Value**
Probability of successful link transmission p_{link}	0.99
Centralized network reliability P_{CenSta}	0.99
Node reliability in distributed network $P_{DisNode}$	0.99
Safety without V2X P_{S_Nor}	0.990
Safety with network P_{S_Net}	0.995
Safety with centralized decision and SID P_{S_SID}	0.995
Safety with WIDCS and WIC P_{S_WIDCS}	0.999

We assess the reliability of IDC across varying node counts and network complexities, detailed in (4.22), alongside the reliability of centralized decision-making systems, outlined in (4.9). For ease of reference, all relevant parameters are summarized in Table 4.2, and adjustments are made based on specific conditions.

Fig. 4.10 illustrates that as the number of nodes requiring passage through the on-ramp merge area increases, the probability of achieving distributed consensus exhibits fluctuations with a general upward trend, in contrast to the reliability of centralized decision-making, which declines. This divergence stems from the inherent fault tolerance of distributed consensus mechanisms. Specifically, Intelligent-Raft reliability is less likely to decrease with the addition of nodes due to its distributed nature. Conversely, centralized decision-making's reliability is heavily reliant on the central node's reliability, which diminishes as the node count and consequent load increase. Furthermore, Fig. 4.10 also shows the reliability of the distributed Intelligent-Raft consensus system consistently surpasses that of centralized decision-making. Thus, compared to centralized systems, distributed consensus significantly enhances the reliability and stability of intelligent, connected, and autonomous systems.

Additionally, the difficulty of establishing or joining the network significantly influences the reliability of subsequent decisions. As depicted in Fig. 4.10, both centralized decision-making and distributed Intelligent-Raft systems demonstrate that higher $P_{forming}$ increases the likelihood of successful decision outcomes, thereby boosting the overall reliability of the decision-making process. This effect is more pronounced and stable in scenarios with a higher total number of nodes. The findings indicate that the reliability of the wireless ad hoc network improves the reliability of both centralized and distributed decision-making systems.

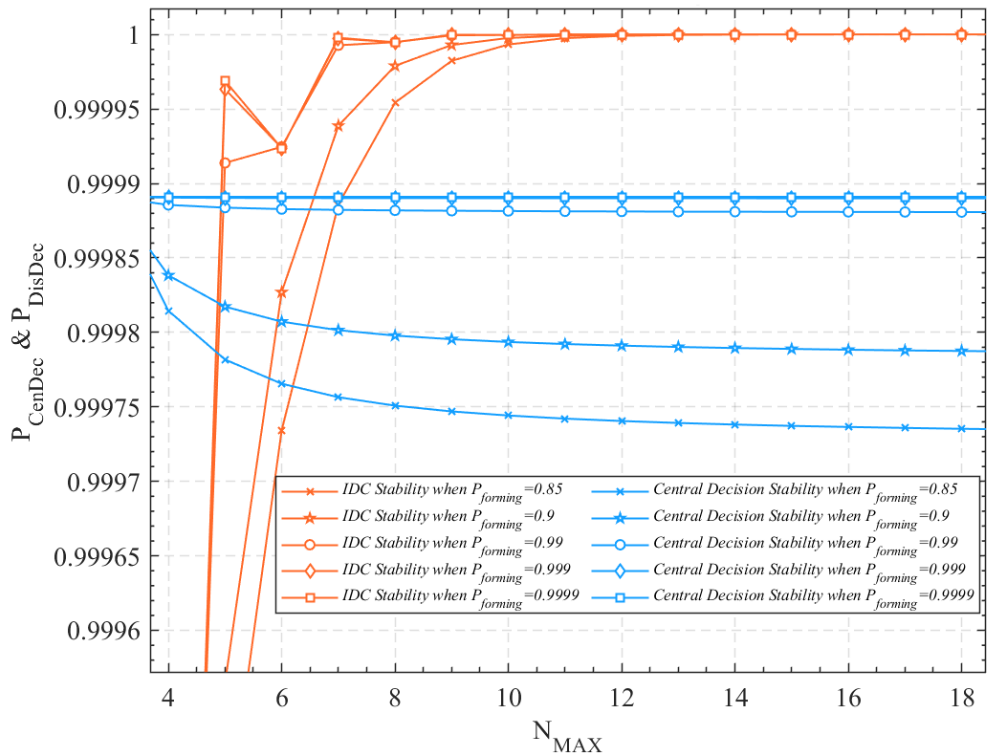

Figure 4.10: Reliability of centralized and distributed decision making system under wireless ad hoc network

4.7.3 AVs Safety in Different Decision-Making Systems and Schemes

The probabilities of safe passage with the WIC scheme within WIDCS, with the SID scheme in a centralized decision-making system, and without Internet of Vehicles support are calculated using (4.27), (4.15), and (4.7), respectively.

It can be seen intuitively from Fig. 4.11 that the safety probabilities under varying conditions exhibit a declining trend as the total number of nodes increases. In addition, with the help of the WIC solution based on WIDCS, the probability of AVs passing the on-ramp merging area safely is always much higher than the other two cases. The probability of AVs passing safely with the assistance of the SID solution based on centralized decision-making is also higher than without any communication interaction. Setting P_{S_SID} and P_{S_WIDCS} at 0.999 still shows WIDCS as safer than centralized decision systems. Furthermore, even when both P_{S_Nor} and P_{S_SID} are set at 0.995, centralized systems outperform non-communicative scenarios. This demonstrats that the substantial safety benefits of both WIDCS and centralized decision-making in enhancing on-ramp merging safety for AVs, with WIDCS offering superior improvements.

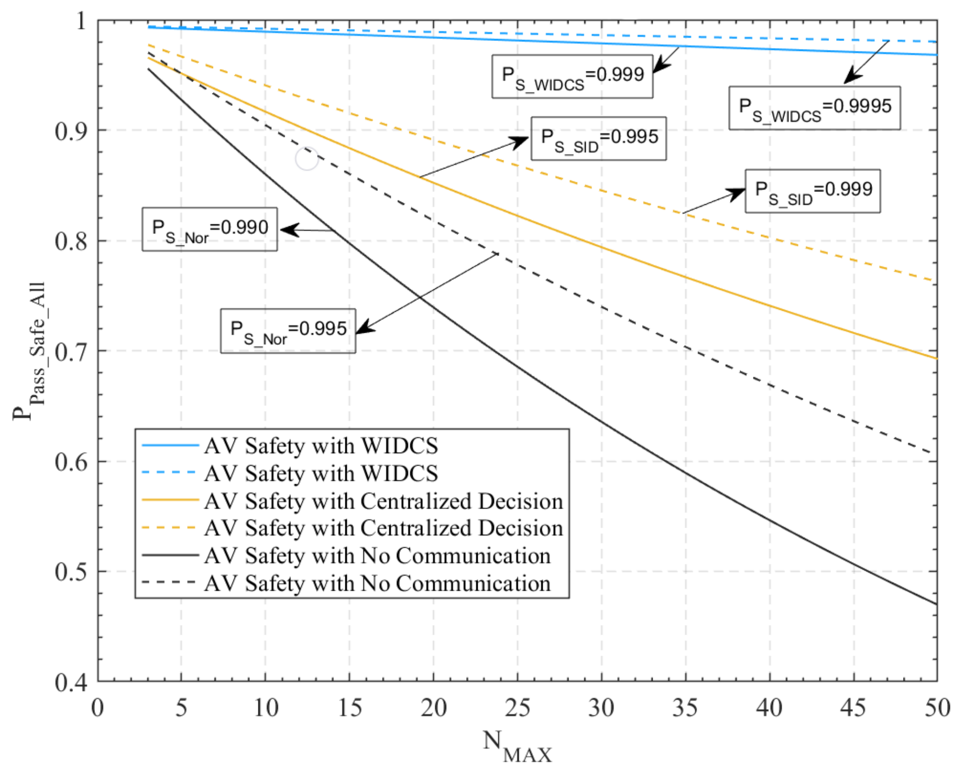

Figure 4.11: Safety of AVs in on-ramp merging scenarios without communication systems, with centralized decision-making system, and with WIDCS

Regarding the foundational principles, the reliability of centralized decision-making is primarily dependent on the central node, as illustrated in (4.14). As node numbers increase, the central node's reliability diminishes, adversely impacting the safety of autonomous driving reliant on this system. Conversely, distributed decision-making demonstrates improved reliability with an increasing number of nodes, as defined in (4.26). Because this reliability derives from the collective reliability of all participating nodes, rather than a single one, underscoring the enhanced safety provided by WIDCS. Vehicles lacking communication systems are forced to depend solely on their internal perception and judgment, lacking crucial collaborative data. This shortcoming makes them less safe in certain situations compared to systems that utilize communication-based decision-making.

4.8 Experiments

In this section, we verify the feasibility of WIDCS by practically testing the throughput and consensus success rate of the RaBee. In addition, through the integration of the RaBee and AVs, we recreate the experimental on-ramp merging scenario to examine the impact of different decision-making methods and different traffic schemes on AV safety. The following paragraphs introduce the setup and the process of the experiments. Finally, we analyze the experimental data.

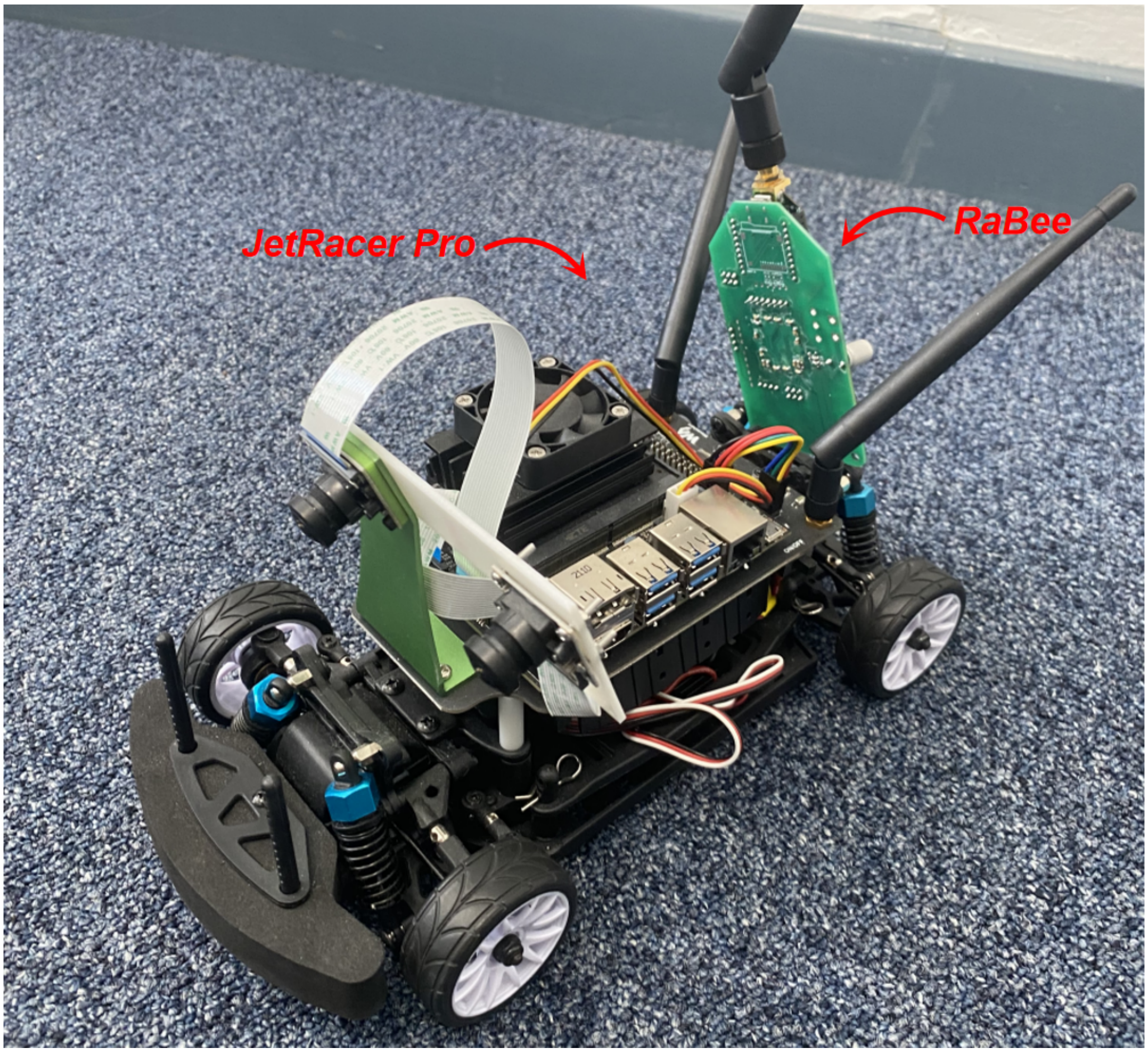

Figure 4.12: JetRacer Pro autonomous vehicle equipped with a RaBee module

4.8.1 Hardware Setup

As shown in Fig. 4.12, the JetRacer Pro is a high-performance AI racing car noted for its rapid speed. It is equipped with the NVIDIA Jetson Nano, a small but powerful AI computer that provides the computing power needed for tracking roads and detecting objects. In the experiment, each Jetracer AV is equipped with a RaBee module that communicates

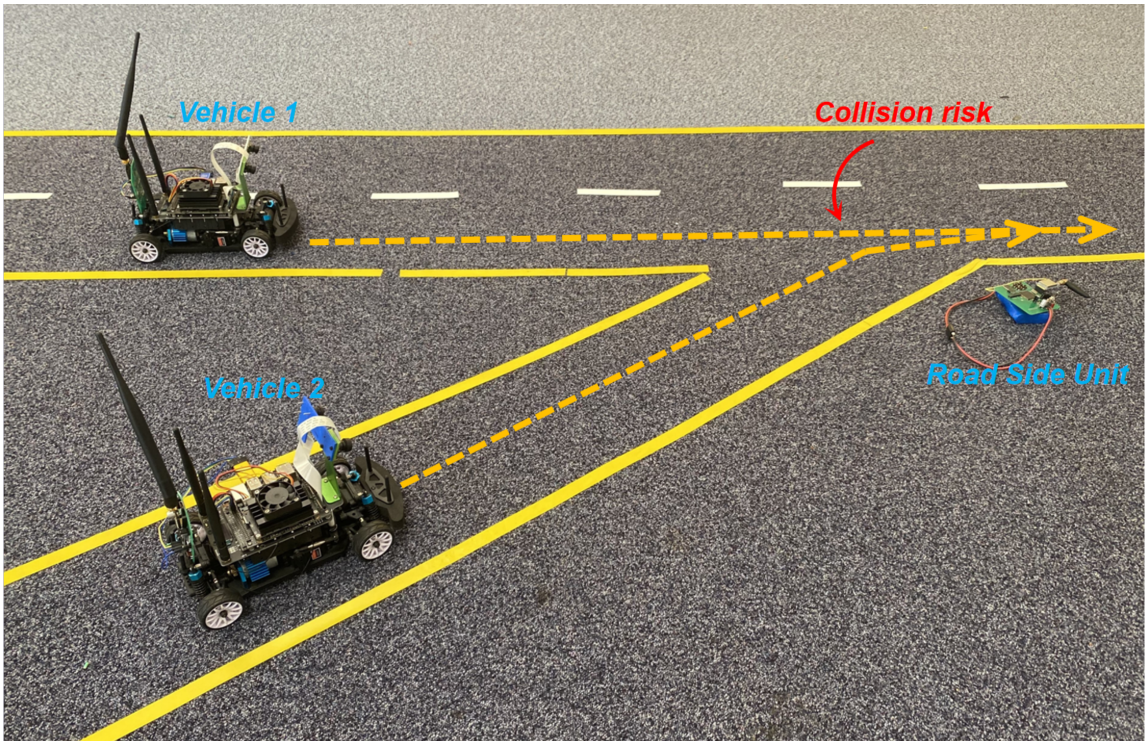

Figure 4.13: 2 JetRacers assisted by RaBee pass on-ramp merging area

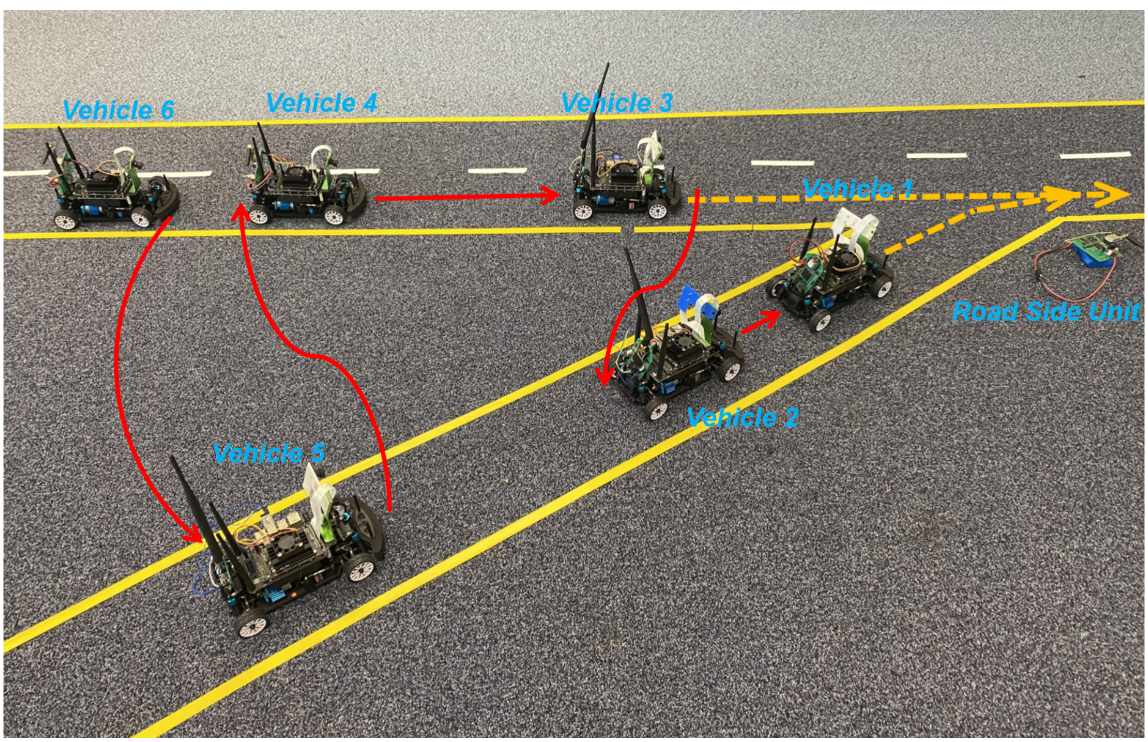

Figure 4.14: 6 JetRacers assisted by RaBee pass on-ramp merging area

via UART. The ZigBee module manufactured by DIGI carried by the RaBee supports half-duplex mode, with a data speed of 250 Kb/s, a transmission power of 20 dBm, a transmission frequency of ISM 2.4GHz. In addition, the MCU (STM32F407ZGT6) has an operating principal frequency of 168MHz, external expansion heap memory size is 1MB, and the SD card size is 4GB for storing history consensus log and data. Moreover, during the data transmission stage (same for reception), the data needs to be transferred from the MCU to the ZigBee module first, and then transmitted to the free space wirelessly. The slowest link in each transmission will affect the performance of the entire system. So we increase the baud rate of the MCU's UART protocol as much as possible, and set the baud rate for interaction with the ZigBee module to 921600.

4.8.2 Experimental Setup

Experiment 1

In research of the consensus throughput of RaBee, various numbers of integrated modules are tested to evaluate system performance. Initially, upon activation, the RaBee nodes automatically establish an ad hoc network via their ZigBee modules and prepare for the consensus process. Following consensus, RaBees store the data on SD cards before execution. Throughput is assessed by measuring the time required for different node counts to achieve consensus, averaging results across 30 trials.

We also investigates how the consensus success rate varied from different number of RaBee nodes. We tested three rounds of experiments in total, and took the average of the three experiments as the final consensus success rate. For each round of experiments, we test the consensus success rate when there are 3 to 9 RaBees in the network. For each test, we repeatedly triggered the consensus initialization 100 times and recorded the number that the consensus was finally completed.

Experiment 2

To evaluate the enhancement of on-ramp merging safety by the RaBee system with WIC schemes, we conducted an experimental scenario, depicted in Fig. 4.13 and 4.14. We established a distributed cluster consisting of one RSU and two to eight JetRacers, each equipped with a RaBee module that autonomously established a ZigBee network at the start of the experiment. The RSU initiates a round of consensus to determine the merging order, with all AVs merging according to their assigned WIC. The safety probability for all vehicles during merging is denoted as P_{S_WIDCS} if the consensus is successful, and P_{S_Net} if not.

For comparison, we also assessed the safety of AVs under a centralized decision-making system, which relies on real-time SID updates by the central node for vehicle merging. The safety probability for each JetRacer under successful SID updates is P_{S_SID}, and P_{S_Net} if updates fail. We evaluate the system safety by monitoring the collision rate among JetRacers in each cluster.

To clearly demonstrate the influence of the communication decision system on AV safety, we preset the parameters as follows: $P_{S\_Nor} = 0$, $P_{S\_Net} = 0.5$, $P_{S\_WIDCS} = 0.95$, and $P_{S\_SID} = 0.9$. In the experiment, each vehicle simulates its accident rate using a software random process based on its specific conditions and preset parameters.

4.8.3 Experimental Results

Experiment 1

Table 4.3: RaBee Quantity versus Consensus Performance

Quantity of Nodes	*Latency (ms) Start from Followers*	*Consensus Success Rate*
3	328.8	0.980
4	355.6	0.980
5	359.1	0.990
6	400.3	0.997
7	405.8	0.997
8	452.1	1.00
9	455.0	1.00

Table 4.3 presents the average latency for RaBee consensus under varying node counts. As node numbers increase, consensus time clearly rises, since the leader must coordinate more followers, lengthening the confirmation process. Notably, consensus times for adjacent node counts are similar; for example, clusters of four and five nodes both take around 350 ms. This is because consensus is achieved as soon as the leader receives successful acknowledgments from over half the nodes. Once this threshold is met, the leader updates and broadcasts the heartbeat, rendering additional acknowledgments from remaining nodes irrelevant to the current round's consensus. Thus, three successful acknowledgments are needed regardless of whether there are four or five nodes.

The third column of Table 4.3 presents the consensus success rates for varying numbers of RaBee nodes. With three nodes in the network, the success rate is approximately 0.98, increasing to 0.997 when the node count reaches six. Beyond six nodes, the probability of achieving successful consensus nears 1.0. Experimental results show a consistent upward trend in the success rate as node numbers increase, suggesting enhanced system fault tolerance in line with theoretical simulation. This progression highlights the increasing robustness of WIDCS with additional nodes.

Experiment 2

Table 4.4: AV Safety under Different Conditions

Quantity of Nodes	*AV Safety with WIC and RaBee*	*AV Safety with SID and Central Decision System*
3	0.84	0.76
4	0.82	0.64
5	0.78	0.52
6	0.72	0.40
7	0.70	0.34
8	0.68	0.26
9	0.64	0.22

Table 4.4 shows the frequencies of AVs safely navigating the on-ramp merging area under various conditions. Both in IDC frameworks and centralized decision-making systems, the safety of AVs declines as the number of nodes increases, aligning with simulation results. The data also reveal that AVs assisted by the distributed Intelligent-Raft generally exhibit higher safety levels compared to those managed by centralized decision-making systems. These experimental results conclusively demonstrate that the WIDCS enhances the safety of autonomous driving in on-ramp merging scenarios.

Although the experiment is conducted in a laboratory setting, the results provide meaningful implications for real-world on-ramp merging scenarios. In practice, AVs equipped with WIDCS would be able to exchange real-time information—such as speed, position, and intent—with surrounding vehicles, enabling further coordinated joint decision-making. This level of interaction allows AVs to negotiate merging sequences more smoothly, reduce uncertainty, and respond to dynamic traffic conditions, thereby enhancing overall safety and traffic flow in real highway environments.

4.9 Conclusion

In this chapter, we successfully implemented the Intelligent-Raft consensus protocol on the embedded hardware platform RaBee, pioneering practical IDC systems with ad hoc network capabilities. The IDC-based method, known as WIC, is introduced to enhance the safety of autonomous driving during on-ramp merging. Our developed mathematical model demonstrates the reliability of both centralized and distributed decision systems and analyzes the safety of AVs navigating on-ramp merging scenarios under different decision-making frameworks. Simulation results highlight the advantages of applying DC and confirm that AV safety with WIDCS significantly outperforms traditional centralized decision-making approaches. Experimental validation was also conducted using the RaBee module in recreated on-ramp merging scenarios with JetRacer AVs. The consistency between experimental and simulation outcomes underscores the efficacy of WIDCS in reducing accident rates, laying a solid foundation for further exploration of its practical deployment.

Nevertheless, several limitations should be acknowledged. The RaBee system utilizes the ZigBee communication protocol, which may not meet the stringent requirements of automotive-grade applications, potentially limiting its applicability under high-demand conditions. Furthermore, while the Intelligent-Raft protocol shows promise, additional empirical studies are needed to verify its performance across dynamic and heterogeneous environments and to assess its adaptability to varying decision-making processes. Even with these limitations, the present work provides a strong basis for future research aimed at advancing wireless IDC systems for real-world autonomous driving scenarios.

Chapter 5

HIntS: Heterogeneous Intelligent Joint Decision for Connected and Autonomous System

5.1 Introduction

Joint decision-making enables CAS to execute advanced automation tasks and enhance cooperation, becoming more essential as the individual intelligence encounter more challenges in increasingly complex and dynamic scenarios. Joint decision-making mechanisms are primarily categorized into centralized and distributed approaches, each offering distinct advantages [130–132]. CDM enables nodes to upload data to a central control hub, which makes decisions and issues commands for execution [133]. By centralizing decision power, CDM has simplified joint-decision process and can lead to more efficient and straightforward service, which is ideal for situations requiring rapid decision response. In addition, the single source of decision ensure unified control across all nodes in the network, which enhance system consistency by preventing individual nodes from making isolated decisions that could disrupt overall operations [134]. The efficiency and uniformity of CDM have long established it as the primary solution for joint-decision [135–137].

However, due to performance limitations such as reliability, privacy, scalability, and the risk of single points of failure, CDM faces challenges in certain practical applications, particularly in scenarios with a large number of nodes or stringent robustness requirements [138–140]. Distributed decision-making has recently gained widespread attention for addressing limitations inherent in centralized systems. Decentralizing decision power reduces system reliance on a central control node, thereby enhancing both robustness and scalability while mitigating the risk of single points of failure.

IDC, as a novel distributed decision-making mechanism, has recently gained substantial attention in academia. This mechanism enables ICA nodes to reach joint-decisions through mutual agreement facilitated by standardized distributed protocols. In addition to the inherent advantages of distributed decision-making, IDC mechanism also provides fault tolerance and data consistency, a feature lacking in CDM.

Despite the various advantages of IDC and CDM, they still encounter challenges in practical applications. First, current IDC mechanisms struggle to achieve globally optimal solutions. Moreover, relying on a single type of joint-decision mechanism is insufficient to meet the diverse requirements of various applications. This chapter addresses these two challenges and proposes solutions, which are subsequently verified and analyzed.

5.1.1 Challenge 1

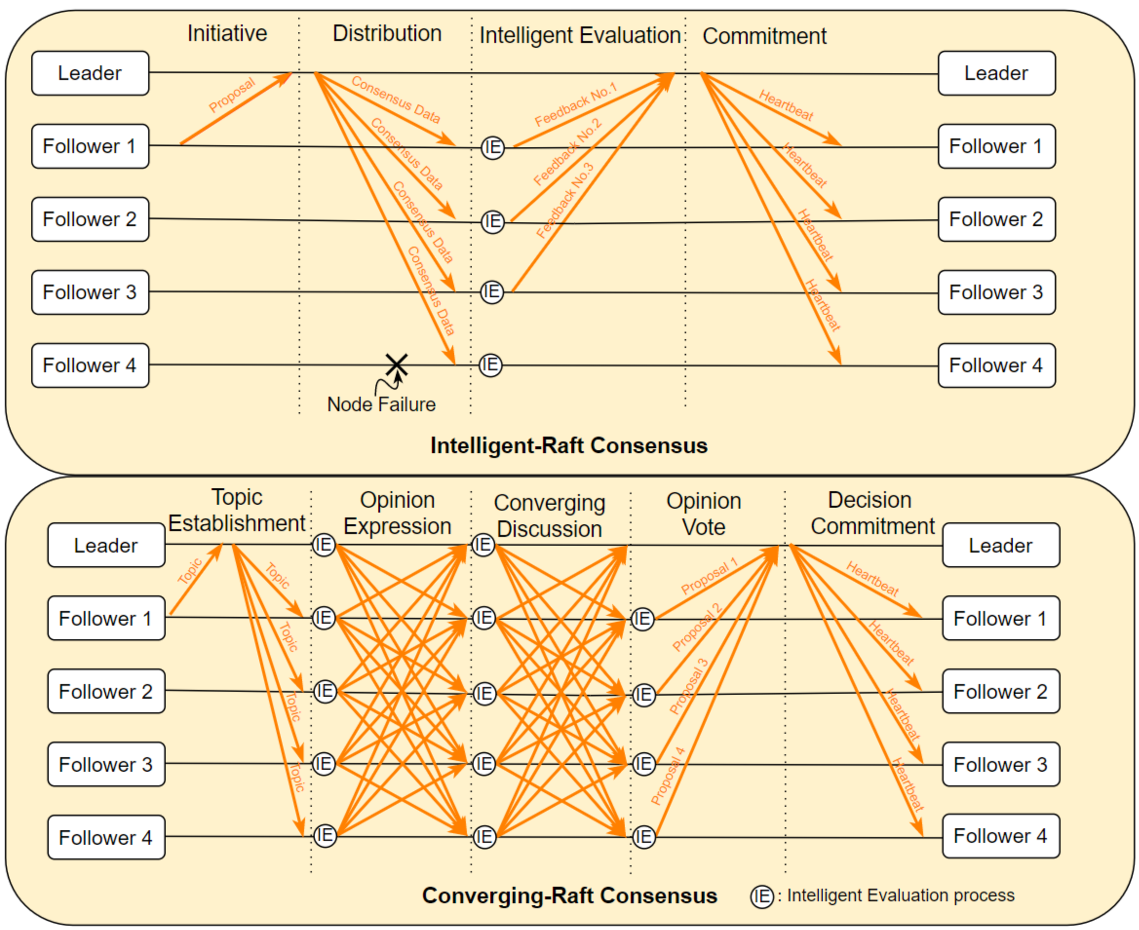

Figure 5.1: The workflow of Intelligent-Raft and proposed Converging-Raft

IDC is not always able to achieve a globally optimal decision. We use Intelligent-Raft, an IDC protocol adapted from the Raft crash fault tolerance algorithm and designed for increasingly intelligent and autonomous nodes, as an example. Intelligent-Raft implements the joint-decision process based on a decision proposal made by one node, as shown in the workflow in Fig. 5.1. All participating nodes independently vote to either accept or reject the proposal based on their intelligent evaluation, without the option to modify or improve it [2] [12]. This limitation often results in a suboptimal joint decision, which can negatively impact overall system performance. For instance, in an vehicles platooning, if one AV proposes a suboptimal route due to incomplete traffic data, other vehicles may follow this decision, leading to delays or inefficient navigation.

It is inspiring that if nodes are able to discuss and improve the proposal during the consensus process, a globally optimal joint-decision could be achieved for specific scenarios. However, no existing IDC mechanism currently takes this capability into account. Thus, there is a need for an enhanced IDC protocol that is thoroughly analyzed and verified, capable of leveraging the collective intelligence of all nodes to facilitate convergence toward a globally optimal joint-decision.

5.1.2 Challenge 2

Centralized and distributed decision-making mechanisms each have their strengths and weaknesses. For example, CDM enables fast decision-making but relies heavily on trusted public infrastructure. On the other hand, IDC excels in scalability and fault tolerance, but consensus-based protocols often bring more latency. Due to their unique characteristics, a single type of mechanism is insufficient to meet the diverse requirements of all scenarios in a complex application. We use autonomous driving as an example. In vehicle platooning scenario, AVs demands real-time traffic information updates. CDM efficiently meets these requirements, whereas the complexity of IDC may introduce unnecessary latency. On the other hand, complex urban scenarios often include a large number of nodes and need high robustness for reaching joint-decision, while CDM faces challenges.

Additionally, the implementation of the joint-decision mechanism must rely on the underlying wireless network structure. Each network structure possesses unique characteristics that directly affect the joint-decision performance. For instance, while ad hoc network improve scalability of cellular networks, their limited transmission range restricts joint-decision to smaller areas.

Consequently, heterogeneous joint-decision mechanisms and network structure may be needed depending on the situation especially in dynamic and complex scenarios. To more accurately tailor the solution, it is worth to analyze the performance of various combinations of joint-decision mechanisms (e.g., CDM, Intelligent-Raft, Converging-Raft) and network architectures (e.g., cellular networks, distributed ad hoc networks), and integrates them into one system to better meet the diverse requirements of different applications.

5.1.3 Literature Review

In recent works, there are few related studies on proposing and analyzing DC protocols in wireless communication scenarios. [141] introduced a belief model based on Dempster-Shafer theory to address uncertainty in multi-agent systems and proposed a method to generate random networks that ensure consensus decision-making under trust boundary constraints. Enhancing decision-making performance in dynamic environments, [142] proposed a cooperative algorithm called "Restarted Bayesian Online Change Point Detection" (RBO-Coop-UCB), which combines change point detection with multi-agent information-sharing mechanisms. Similarly, [143] developed a distributed consensus algorithm that improves efficiency, robustness, and credibility in service-oriented IoT by optimizing multi-parameter matching values and leveraging cluster-based local consensus calculations. The reliability of Raft under low communication link reliability was analyzed by [83], providing deployment guidelines for its application in the Industrial Internet of Things. Moreover, [17] examined the communication resources required for various DCs in wireless networks, highlighting their respective challenges and advantages. These studies primarily focus on either designing novel consensus algorithms to achieve efficient data consensus in uncertain wireless environments or analyzing the performance of traditional DC mechanisms in these settings. However, the influence of node intelligence on distributed consensus mechanisms remains underexplored. With advancements in AI, wireless devices are becoming increasingly intelligent, requiring joint decision-making processes that allow nodes to deeply deliberate and seamlessly exchange opinions to reach consensus. Additionally, many studies neglect the global optimality of the decisions made, a critical aspect as different joint decisions can have varying impacts on intelligent systems, including potentially negative ones. Further research is needed to develop DC protocols that not only address performance in wireless environments but also ensure globally optimal joint decisions that positively impact smart devices.

Other studies have explored DC algorithms to address challenges across diverse application scenarios. For instance, a novel DC algorithm tailored for blockchain systems in multi-hop wireless IoT networks was introduced by [144], targeting the complexities of wireless environments while achieving asymptotically optimal time complexity and energy efficiency. To enhance decision-making in underwater passive target detection, [145] proposed an autonomous distributed consensus network that employs relative entropy for effective information fusion, enabling multiple sensor nodes to achieve consistent detection outcomes in complex environments. A general DC algorithm enabling each sensor node in a wireless sensor network to compute the average log-likelihood ratio (LLR) of local observations and obtain a complete information vector—or its estimate—was proposed by [146], significantly improving decision-making capabilities. Addressing energy constraints in wireless sensor networks, [147] presented a multi-period scheduling approach based on a distributed consensus algorithm to optimize sensor working modes and maximize network utility. [148] combines a two-hop Raft-based consensus mechanism with dynamic negotiation to improve the coordination and reliability of driving decisions in delay-sensitive applications. However, these studies typically focus on a single type of DC protocol and do not explore combining different consensus protocols to optimize system performance across various applications. Relying solely on one joint decision protocol may fail to address the diverse requirements of different applications, such as balancing latency, scalability, or achieving optimality. This limitation underscores the need for further research into hybrid approaches that leverage the strengths of multiple consensus mechanisms.

In addition, most research in this area remains theoretical or simulation-based, with few tangible hardware implementations or verification. It's more inspiring to design a hardware platform to practically verify the feasibility and improvement of wireless joint-decision protocol. Furthermore, the effect of wireless network structure on the performance of joint-decision is often underestimated or neglected.

5.1.4 Contributions

In this chapter, we propose Converging-Raft, a novel IDC protocol designed to leverage the integrated intelligence of all nodes to achieve a globally optimal solution, addressing Challenge 1. To address Challenge 2, We propose HIntS, a versatile joint-decision architecture that integrates CDM, Intelligent-Raft, and Converging-Raft mechanisms within a hybrid wireless ad hoc and cellular network framework, designed to meet diverse application requirements in ICA scenarios. We design and implement the practical HIntS, designated as 5G-MInd, which is a fully functinoal hardware module that integrates both hardware and software within an embedded framework. We develop an analytical reliability model of HIntS and derived the latency under its different working modes using the established Markov chain model. The simulation results combined with qualitative analysis demonstrate the different advantages of HIntS working modes and offer valuable guidance for future joint-decision mechanism deployment. Utilizing the 5G-MInd platform, we experimentally tested the reliability and latency performance across different working modes. The experimental data validate the simulation results and theoretical analysis.

5.2 Methods and Results

This section presents the principles of Converging-Raft and HIntS, and **summarizes the main results, providing a comparative analysis of fault tolerance reliability, and latency performance across different modes**. Detailed mathematical models, simulation analyses, and experimental results will be presented in subsequent sections.

5.2.1 Converging-Raft

While Intelligent-Raft falls short in certain applications due to limitations imposed by the initiator's intelligence, we introduce Converging-Raft which is an IDC mechanism that derived from traditional Raft protocol with the workflow depicted in Fig. 5.1. Following the Raft protocol, nodes are categorized as followers, candidates, and a leader [2]. Each Converging-Raft process comprises five stages: Topic Establishment, Opinion Expression, Converging Discussion, Opinion Vote, Decision Commitment.

Converging-Raft operates by electing a Leader from within a cluster of candidates prior to initiating the consensus process, with the Leader managing the consensus converging procedure. Once the Leader is elected, the remaining Candidates revert to Follower status. During the *Topic Establishment* stage, the Leader broadcasts a joint-decision topic requiring further discussion to all nodes. In the *Opinion Expression* stage, participating ICA nodes broadcast their proposals to all other nodes based on their individual perspectives about the topic in order to allow them to think about all ideas, although these proposals may not represent optimal solutions. In the *Converging Discussion* stage, nodes evaluate and analyze all received proposals. Regarding this joint-decision topic, since nodes have seen all the solutions proposed by others, they can carefully compare these solutions and select the one they believe is best based on their own analysis. Alternatively, after comparing the proposals, nodes may recognize potential for optimization and calculate an improved proposal. The Converging Discussion stage plays a critical role in providing

Table 5.1: Performance Comparison of Different Working Modes of HIntS

HIntS Modes	***Fault Tolerance***	***Global Optimal Solution***	***Transmission Coverage***	***Scalability***	***Reliability***	***Latency***
Intelligent-Raft within Ad Hoc Network	Strong	No	Medium	High	High	Low
Converging-Raft within Ad Hoc Network	Strong	Yes	Medium	High	Lowest	High
CDM within Ad Hoc Network	＼	＼	＼	＼	＼	＼
Intelligent-Raft within Cellular Network	Strong	No	Wide	Medium	Highest	Medium
Converging-Raft within Cellular Network	Strong	Yes	Wide	Medium	Low	Highest
CDM within Cellular Network	Weak	No	Wide	Medium	Dynamic	Lowest

crucial optimization process of joint-decision. In the *Opinion Vote* stage, nodes vote for their preferred solution, submitting their choice to the Leader based on prior discussions and analysis. In the *Decision Commitment* stage, a consensus is reached when more than 50% of the nodes evaluate the same one proposal as the optimal solution. The Leader then updates the final optimal proposal in the cluster network for the following execution.

Through comparison, iteration, and convergence among ICA nodes, the final joint-decision reflects the collective intelligence of the cluster, ensuring the safest and most efficient performance for the system. However, this protocol also bring new challenges such as leading to higher latency owing to increased interacting procedures. Thus, it's necessary to analyze and compare the performance of Converging-Raft with Intelligent-Raft and CDM in Section 5.5, with the results summarized in Section 5.2.3.

5.2.2 Heterogeneous Intelligent Joint-Decision System

To enable ICA nodes to utilize various joint-decision mechanisms and network structures, we propose HIntS, a unified system that integrates heterogeneous joint-decision mechanisms with diverse network architectures, as illustrated in Fig. 5.2. The heterogeneous joint-decision mechanisms of HIntS include CDM, Intelligent-Raft and Converging-Raft. The heterogeneous network structures include ad hoc network and cellular network. This integration results in five possible working mode combinations: Intelligent-Raft within an ad hoc network, Converging-Raft within an ad hoc network, Intelligent-Raft within a cellular network, Converging-Raft within a cellular network, and CDM within a cellular network. Each mode has its own strengths and limitations, making it a tangled choice in real-world applications in order to balance the needs of different prospects.

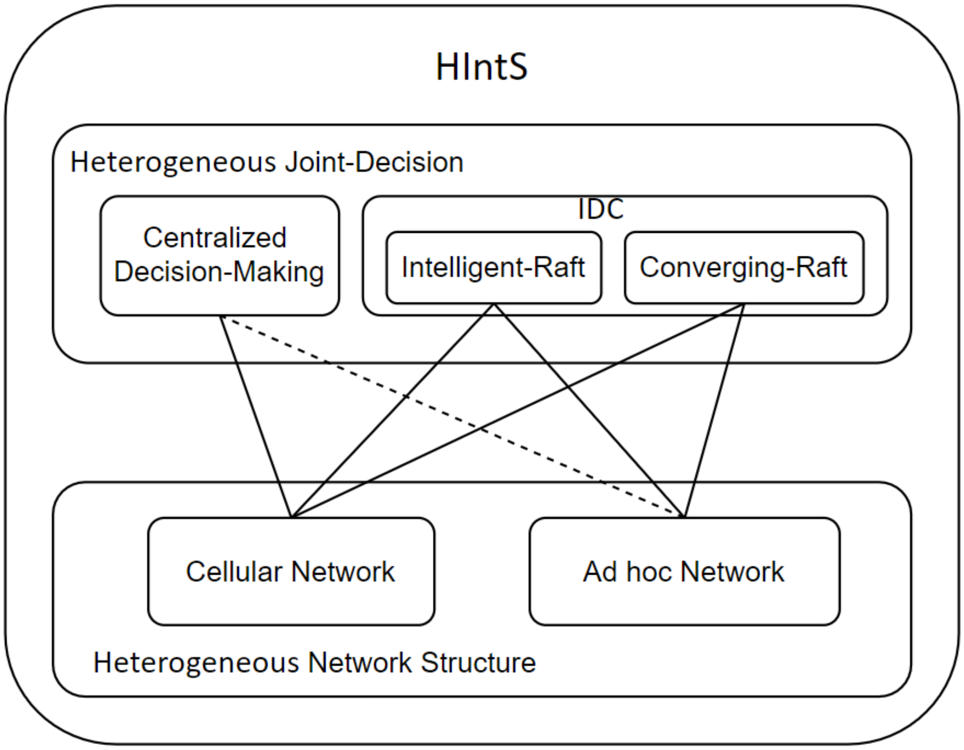

Figure 5.2: HIntS Architecture

5.2.3 Main Results

In this section, we would like to first present **the main results in Table 5.1** to identify the strength of different combination of joint-decision mechanisms and network architectures, providing guidance for future deployment strategies. We will **give the detailed mathematical analysis in Section 5.4 and simulated results in Section 5.5.** We exclude the case of CDM within ad hoc network since CDM heavily depends on a central control server, which is designed for high capacity and flexibility. In contrast, ad hoc network nodes are less reliable as the central nodes compared to a dedicated central control server. Therefore, this article focuses on the remaining five modes, with an initial analysis of each indicator.

Thanks to the fault tolerance of the IDC mechanism, both Intelligent-Raft and Converging-Raft can withstand up to 50% node failure, offering greater robustness compared to CDM. Additionally, the distributed nature of IDC supports leader reelection in the event of the Leader crash, ensuring system robustness and continuity. Converging-Raft's key advantage is its ability to achieve a globally optimal joint decision, making it ideal for applications that prioritize optimal performance. In addition, the choice of network architecture also impacts transmission coverage and scalability. While ad hoc networks have a smaller geographical coverage than cellular networks, they can support a larger number of nodes to reach consensus.

Next, we discuss the reliability performance. As shown in Table 5.1, Intelligent-Raft within a cellular network offers the highest system reliability due to its relatively simple process and the mature, stable architecture of cellular networks. Conversely, Converging-Raft within an ad hoc network exhibits the lowest reliability. CDM reliability is dynamic, decreasing as the number of connected nodes grows, due to the central control node's heavy dependence on the total number of nodes.

In terms of latency, CDM has the fastest decision rate due to its simplicity, whereas Converging-Raft within a cellular network experiences the longest latency due to its complex protocol and the relatively slower network transmission in cellular networks. It is also clear that the complexity of the joint-decision mechanism has a greater impact on final latency than the network structure.

The results demonstrate that each working mode of HIntS presents distinct advantages. Intelligent-Raft within an ad hoc network excels in reliability and latency, providing a well-rounded solution. Converging-Raft within an ad hoc network enables global optimal decision-making across a large number of devices. Intelligent-Raft within a cellular network offers the highest reliability, while Converging-Raft within a cellular network extends optimal decision-making over a wide area. CDM, with the fastest processing, supports rapid decision-making over long distances.

5.3 Design of 5G-MInd Module

5.3.1 5G-MInd Architecture

5G-MInd, short for 5G Mind Induction, is a fully functional HIntS module, as shown in Fig. 5.3. For the hardware, a high-performance embedded MCU, specifically the STM32F407ZGT6 chip with a powerful Cortex-M4 core running at up to 168MHz, provides the necessary computing power for joint-decision mechanisms. To support the mechanism's memory requirements to manage a large number of parameters, data and runtime status, an additional 1MB RAM chip (IS62WV51216) is connected to the MCU via a Flexible Static Memory Controller (FSMC) parallel interface. For ease of debugging, we integrated an OLED display and buttons as a human-computer interface (HCI) for status monitoring, parameter setting, etc. Additionally, the power unit supplies stable power to the system, and an SD card is used for ROM expansion, offering ample space to store processed consensus data.

Another key feature of 5G-MInd is the heterogeneous network architecture. For the wireless ad hoc network in HIntS, we implemented the ZigBee network using the DIGI XBee3 module, XB3-24Z8UM, operating in the 2.4GHz ISM band. The 5G architecture, utilizing Quectel's RM500Q-GL module, provides cloud connectivity for 5G-MInd, enabling seamless integration with cellular networks. The stackable design, illustrated in Fig. 5.3, provides a compact structure for 5G-MInd, with the MCU system board and RM500Q-GL dongle exchanging data via the UART interface using AT commands.

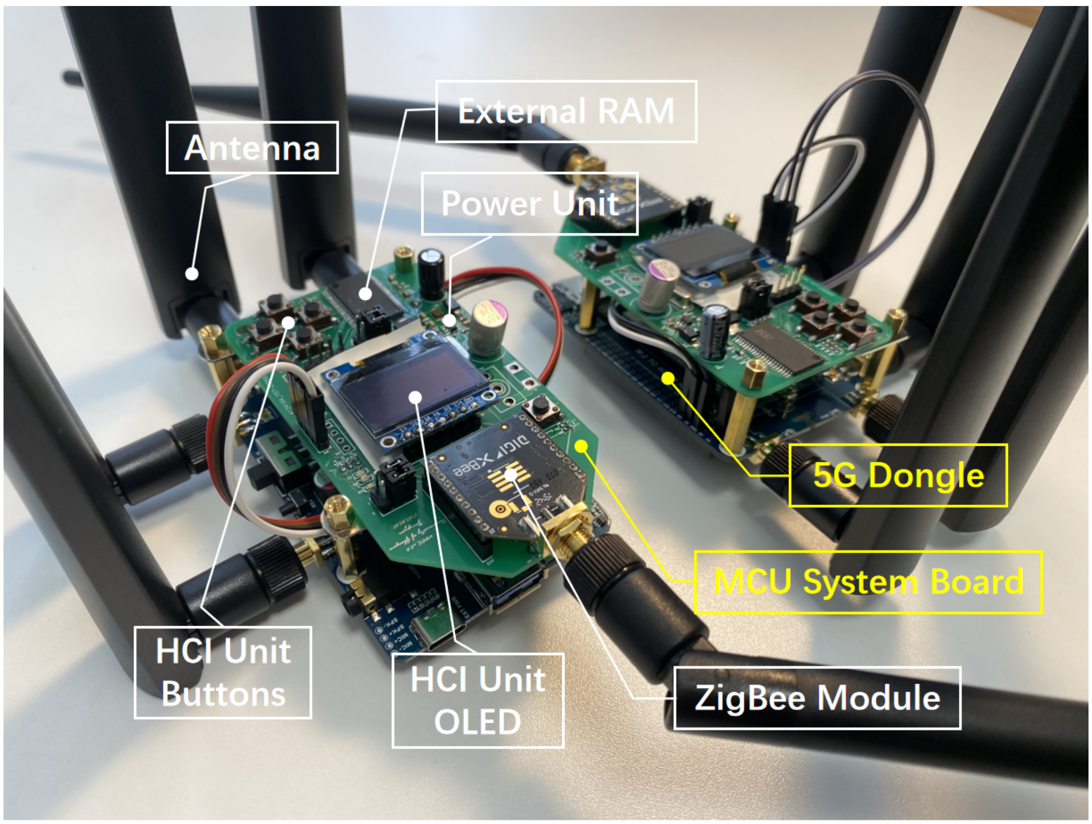

Figure 5.3: The hardware of 5G-MInd (2 copies)

The 5G-MInd software framework is designed to implement joint-decision mechanisms and support network communication. The software application comprises five primary threads: three dedicated to CDM, Intelligent-Raft, and Converging-Raft, respectively; one for HCI management; and one for managing the communication transceiver. The 5G and ZigBee modules are connected to the MCU via UART interfaces, providing wireless communication resources for the MCU tasks. Additionally, data agreed upon through different joint-decision mechanisms are stored in a shared data pool, tagged according to different working modes for traceability. 5G-MInd provides a comprehensive set of peripheral interfaces, allowing seamless integration into various ICA nodes, such as AVs and industrial robots, to provide joint-decision services.

5.3.2 5G-MInd Network

Nodes equipped with 5G-MInd modules can perform joint-decision through the ZigBee network. ZigBee, an advanced communication protocol based on the IEEE 802.15.4 standard, is specifically designed for low-power IoT scenarios. Its self-organizing capability is a key advantage, supporting diverse network topologies—including star, tree, and mesh—and operating effectively in dynamically changing environments. In cases of node failure or environmental changes that block communication paths, the ZigBee network can automatically reconfigure routes to maintain stable communication. This ensures a robust network environment for the upper-layer joint-decision mechanisms of HIntS.

In addition, 5G-MInd nodes can provide joint-decision service through the cellular network. In CDM, all participating nodes are connected to a central server which aggregates the information uploaded by the nodes, generates a central decision, and distributes it to all ICA nodes for execution. For Intelligent-Raft and Converging-Raft, the MQTT protocol is required for network transmission.

MQTT (Message Queuing Telemetry Transport) is a lightweight protocol designed to simplify communication between small devices and enhance data transmission efficiency in bandwidth-limited environments. It involves three main roles: Publishers, who publish messages to specific topics; Subscribers, who receive messages from topics they subscribe to; and Broker, a third-party server that manage message delivery between publishers and subscribers. In MQTT, a node can function as either a Publisher or a Subscriber. If data needs to be transferred between two nodes, the sending node must first publish the data to the Broker Server, which then forwards it to the receiving node. As the Broker continually distributes messages across these nodes, the joint-decision will eventually be completed.

The following is an example of implementing Intelligent-Raft within a 5G cellular network. All ICA nodes first establish a connection with the MQTT Broker, which manages message forwarding and delivery essential for Intelligent-Raft using its publish-subscribe model. In MQTT, a node can function as either a Publisher to send messages or a Subscriber to receive them. The Leader publishes its messages to the topic <Intelligent-Raft: LEADER>, to which all Followers subscribe in order to receive updates. Conversely, Followers publish their messages to the topic <Intelligent-Raft: FOLLOWER>, which the Leader subscribes to for receiving follower responses. As the Broker continually distributes messages across these topics, the Intelligent-Raft will eventually be completed.

5.4 Mathematical Models

In this section, we establish mathematical models to analyze the reliability and latency of the five working modes of HIntS to support the main results in Section 5.2. The frequently used notations are summarized in Table 5.2.

5.4.1 Joint-Decision Reliability Analysis of HIntS

Intelligent-Raft Reliability within Ad Hoc Network

In ICA application, both nodes and links may fail in a wireless network at any stage of the uplink or downlink in the Intelligent-Raft. We define consensus reliability as the probability of successfully completing a consensus and use a fault tolerance model to assess the impact of node and link reliability on Intelligent-Raft in wireless environments.

We assume that each node's reliability and each communication link's reliability are random variables. Let the threshold for the number of faulty nodes be defined as $f = \left\lfloor \frac{n}{2} \right\rfloor$, given that Intelligent-Raft can tolerate up to half of the followers failing. We consider the stage where the leader sends messages to followers as one downlink (DL) communication, and the stage where a follower responds to the leader as one uplink (UL) communication. Let $\Omega = \{N_1, N_2, N_3, \ldots, N_n\}$ represent the set of n followers connected to the leader. Additionally, let the reliability of node i be a random variable P_i^N, the reliability of the downlink between node i and the leader be P_i^{DL}, and the reliability of the uplink between node i and the leader be P_i^{UL}. During the consensus, let $S_{1,x}, S_{2,y}, S_{3,z} \subseteq \Omega$ represent the set of non-faulty nodes, the set of followers that successfully receive the leader's message via DL, and the set of followers whose messages are successfully received by the leader via UL, respectively. The sizes of these sets are given by $|S_{1,x}| = x$, $|S_{2,y}| = y$, and $|S_{3,z}| = z$. The probability $P(S_{1,x}, S_{2,y}, S_{3,z})$ represents the probability that all followers in $S_{1,x}$ are non-faulty, all followers in $S_{2,y}$ have successful DL communication, and all followers in $S_{3,z}$ have successful UL communication.

In the Intelligent-Raft consensus protocol, only the non-faulty nodes can receive messages from the leader by DL communication, and only the nodes that receives the leader's message can send the response back to the leader by UL communication. Therefore, $S_{3,z} \subseteq S_{2,y} \subseteq S_{1,x} \subseteq \Omega$ and $z \leq y \leq x \leq n$. In addition, for the next state, the last state contains all the information of previous states for the state transitions, e.g. the conditional probability $P(S_{3,z}|S_{1,x}, S_{2,y})$ is equal to $P(S_{3,z}|S_{2,y})$. Thus $P(S_{1,x}, S_{2,y}, S_{3,z})$ can be calculated as:

Table 5.2: Frequently Used Notations

Notation	Definition
f	Maximum number of faulty nodes in IDC
n	Number of followers in IDC
N	Number of total nodes in HIntS
P^N	Node reliability
P^{UL}	Probability of successful uplink communication in ad hoc network
P^{DL}	Probability of successful downlink communication in ad hoc network
P^{MTDL}	Probability of successful downlink communication in cellular network through MQTT
P^{MTUL}	Probability of successful uplink communication in cellular network through MQTT
P_{IR_AH}	Probability of successful Intelligent-Raft in ad hoc network
P_{CR_AH}	Probability of successful Converging-Raft in ad hoc network
P_{IR_Cel}	Probability of successful Intelligent-Raft in cellular network
P_{CR_Cel}	Probability of successful Converging-Raft in cellular network
P_{CDM_Cel}	Probability of successful CDM in cellular network
D_{Node}	Node latency expectation
D_{Link}	Communication link latency expectation in ad hoc network
D_{Broker}	Broker expected state transfer Latency in cellular network
D_{IR_AH}	Latency of successful Intelligent-Raft in ad hoc network
D_{CR_AH}	Latency of successful Converging-Raft in ad hoc network
D_{IR_Cel}	Latency of successful Intelligent-Raft in cellular network
D_{CR_Cel}	Latency of successful Converging-Raft in cellular network
D_{CDM_Cel}	Latency of successful CDM in cellular network

$$P(S_{1,x}, S_{2,y}, S_{3,z}) = P(S_{1,x})P(S_{2,y} \mid S_{1,x})P(S_{3,z} \mid S_{2,y}) \tag{5.1}$$

where $P(S_{1,x})$ represents the joint probability that the nodes in $S_{1,x}$ are non-faulty while in $\mathscr{C}_\Omega S_{1,x}$ are faulty; $P(S_{2,y}|S_{1,x})$ represents the joint probability that downlink with nodes in $S_{2,y}$ are successful while that in $\mathscr{C}_{S_{1,x}} S_{2,y}$ are faulty; $P(S_{3,z}|S_{2,y})$ represents the joint probability that uplink with nodes in $S_{3,z}$ are successful while that in $\mathscr{C}_{S_{2,y}} S_{3,z}$ are faulty.

To facilitate the derivation, assuming node reliability, UL reliability and DL reliability of different followers are independent, we have:

$$P(S_{1,x}) = \prod_{u \in S_{1,x}} P_u^N \prod_{v \in \mathscr{C}_\Omega S_{1,x}} (1 - P_v^N) \tag{5.2}$$

$$P(S_{2,y} \mid S_{1,x}) = \prod_{u \in S_{2,y}} P_u^{DL} \prod_{v \in \mathscr{C}_{S_{1,x}} S_{2,y}} (1 - P_v^{DL}) \tag{5.3}$$

$$P(S_{3,z} \mid S_{2,y}) = \prod_{u \in S_{3,z}} P_u^{UL} \prod_{v \in \mathscr{C}_{S_{2,y}} S_{3,z}} (1 - P_v^{UL}) \tag{5.4}$$

According to the Intelligent-Raft protocol, when the number of messages the leader receives from the followers z is no less than $n - f$, where $f = \lfloor \frac{n}{2} \rfloor$, the cluster will reach consensus. Therefore, the probability that the cluster successfully reaches an Intelligent-Raft consensus within ad hoc network, P_{IR_AH}, is the sum of probabilities of all the $(S_{1,x}, S_{2,y}, S_{3,z})$ satisfying $n \geq x \geq y \geq z \geq n - f$ and $\Omega \supseteq S_{1,x} \supseteq S_{2,y} \supseteq S_{3,z}$

$$\begin{aligned} P_{IR\_AH} = & \sum_{n \geq z \geq n-f, \Omega \supseteq S_{3,z}} \sum_{n \geq y \geq z, \Omega \supseteq S_{2,y} \supseteq S_{3,z}} \\ & \sum_{n \geq x \geq y, \Omega \supseteq S_{1,x} \supseteq S_{2,y}} P(S_{1,x}, S_{2,y}, S_{3,z}) \end{aligned} \tag{5.5}$$

The three summations is to traverse all possible combinations of possible node sets for given x, y and z. Through mathematical derivations, we transform Eq. (5.5) as:

$$P_{IR\_AH} = \sum_{n \geq k \geq n-f, \Omega \supseteq S_{J,k}} \prod_{u \in S_{J,k}} P_u^{IRsub1} \prod_{v \in \mathscr{C}_\Omega S_{J,k}} (1 - P_v^{IRsub1}) \tag{5.6}$$

where $S_{j,k}$ is a running variable of subset of nodes with $|S_{J,k}|$ equal to k and $P_i^{IRsub1} = P_i^N P_i^{DL} P_i^{UL}$.

According to Eq. (5.1)–(5.6), we can analyze the influence of the reliability of each node/link on the final consensus reliability which can be universally applied to arbitrary practical situation as long as the reliability of node, UL and DL are given.

Converging-Raft Reliability within Ad Hoc Network

Although Converging-Raft and Intelligent-Raft are both Raft extensions designed for joint-decision applications, Converging-Raft includes additional stages for decision convergence, which can impact system reliability. As outlined in Section 5.2, the Converging-Raft consists of five stages: Topic Establishment, Opinion Expression, Converging Discussion, Opinion Vote and Decision Commitment. With the exception of Opinion Expression and Converging Discussion, which utilize fully connected communication, all other stages depend on either UL or DL communication with the leader. We use the parameters defined in the ad hoc network model, where the reliability of node i is P_i^N, the reliability of the DL between node i and the leader is P_i^{DL}, and the reliability of the UL is P_i^{UL}. For the fully connected Converging Discussion and Opinion Vote stage, we assume that their reliability follow a binomial distribution with the number of non-faulty nodes.

In Converging-Raft, let $S_{1,x} \subseteq \Omega$ represents the set of non-faulty nodes, $S_{2,y}, S_{5,q} \subseteq \Omega$ represent the sets of followers that successfully communicate with the leader and $S_{3,z}, S_{4,p} \subseteq \Omega$ represent the sets of nodes that successfully survive the Converging Discussion and Opinion Vote stages. As only nodes that survived the previous stage can execute the next stage, we derive the following:

$$\begin{aligned}
&P(S_{1,x}, S_{2,y}, S_{3,z}, S_{4,p}, S_{5,q}) \\
&= P(S_{1,x})P(S_{2,y} \mid S_{1,x})P(S_{3,z} \mid S_{2,y})P(S_{4,p} \mid S_{3,z}) \\
&P(S_{5,q} \mid S_{4,p})
\end{aligned} \tag{5.7}$$

We assume that the survival criterion of a node during the Converging Discussion and Opinion Vote stages is that it remains active and receives at least one message in each stage. The success rate of each node, $P_S(a,b)$, in these two stages is:

$$P_S(a,b) = P_a^N(1 - \prod_{v \in b-1} (1 - P_v^{DL})) \tag{5.8}$$

Where the a indicates which node and the b represents the number of nodes from the previous stage. From this, we can derive the reliability for the Converging Discussion and Opinion Vote stages:

$$\begin{aligned}
&P(S_{3,z} \mid S_{2,y}) = P_{CD_AH} \\
&= \binom{S_{2,y}}{S_{3,z}} \prod_{u \in S_{3,z}} P_S(u, S_{2,y}) \prod_{v \in \mathscr{C}_{S_{2,y}} S_{3,z}} (1 - P_S(v, S_{2,y}))
\end{aligned} \tag{5.9}$$

$$
\begin{aligned}
P(S_{4,p} \mid S_{3,z}) &= P_{OV\_AH} \\
&= \binom{S_{3,z}}{S_{4,p}} \prod_{u \in S_{4,p}} P_S(u, S_{3,z}) \prod_{v \in \mathscr{C}_{S_{3,z}} S_{4,p}} (1 - P_S(v, S_{3,z}))
\end{aligned}
\tag{5.10}
$$

Where P_{CD_AH} denotes the reliability of the Opinion Vote stage, while P_{OV_AH} represents the reliability of the Converging Discussion stage.

According to the Converging-Raft protocol, when the number of messages the leader receives from the followers in Opinion Vote q is no less than $n-f$, where $f = \lfloor \frac{n}{2} \rfloor$, the cluster will reach consensus. Therefore, the probability that the cluster successfully reaches a Converging-Raft consensus in ad hoc network, P_{CR_AH}, is the sum of probabilities of all the $(S_{1,x}, S_{2,y}, S_{3,z}, S_{4,p}, S_{5,q})$ satisfying $n \geq x \geq y \geq z \geq p \geq q \geq n-f$ and $\Omega \supseteq S_{1,x} \supseteq S_{2,y} \supseteq S_{3,z} \supseteq S_{4,p} \supseteq S_{5,q}$.

$$
\begin{aligned}
P_{CR\_AH} = P_{CD\_AH} P_{OV\_AH} \times \\
\sum_{n \geq k \geq n-f, \Omega \supseteq S_{J,k}} \prod_{u \in S_{J,k}} P_u^{CRsub1} \prod_{v \in \mathscr{C}_\Omega S_{J,k}} (1 - P_v^{CRsub1})
\end{aligned}
\tag{5.11}
$$

where $S_{J,k}$ is a running variable of subset of nodes with $|S_{J,k}|$ equal to k and $P_i^{CRsub1} = P_i^N P_i^{DL} P_i^{UL}$.

Intelligent-Raft Reliability within Cellular Network

To implement Intelligent-Raft in cellular networks with the MQTT protocol, as described in Section 5.3, the key difference from ad hoc networks is the Broker's role in forwarding. Each message transfer is split into two segments: Follower to Broker and Broker to Leader. The Broker's failure rate at each sub-stage of the process is denoted as $P_{BrokerS}$.

Additionally, we assume the link success rate of node i accessing the MQTT Broker is a random variable $P_i^{MTlinkS}$, the reliability of the DL between the leader and node i is P_i^{MTDL}, and the reliability of the UL between node i and the leader is P_i^{MTUL}. To simplify the differentiation, we assume that the random variables for the MQTT's DL and UL of each node are identical. Therefore, we can obtain:

$$
P_i^{MTDL} = P_i^{MTUL} = P_i^{MTlinkS} \times P_{BrokerS} \times P_i^{MTlinkS} \tag{5.12}
$$

To facilitate the derivation, assuming node reliability, UL reliability and DL reliability of different followers are independent. We can still apply Eq. (5.1) to derive the reliability of Intelligent-Raft in a cellular network. Additionally, by replacing the communication link model in Eqs. (5.3) and (5.4) with the MQTT link model, we obtain the following:

$$P(S_{2,y} \mid S_{1,x}) = \prod_{u \in S_{2,y}} P_u^{MTDL} \prod_{v \in \mathscr{C}_{S_{1,x}} S_{2,y}} (1 - P_v^{MTDL}) \tag{5.13}$$

$$P(S_{3,z} \mid S_{2,y}) = \prod_{u \in S_{3,z}} P_u^{MTUL} \prod_{v \in \mathscr{C}_{S_{2,y}} S_{3,z}} (1 - P_v^{MTUL}) \tag{5.14}$$

As mentioned above, the fault tolerance threshold of Intelligent-Raft is $f = \lfloor \frac{n}{2} \rfloor$. Therefore, the probability that the cluster successfully reaches an Intelligent-Raft consensus with ad hoc network P_{IR_Cel} is the sum of probabilities of all the $(S_{1,x}, S_{2,y}, S_{3,z})$ satisfying $n \geq x \geq y \geq z \geq n - f$ and $\Omega \supseteq S_{1,x} \supseteq S_{2,y} \supseteq S_{3,z}$

$$P_{IR\_Cel} = \sum_{n \geq k \geq n-f, \Omega \supseteq S_{J,k}} \prod_{u \in S_{J,k}} P_u^{IRsub2} \prod_{v \in \mathscr{C}_\Omega S_{J,k}} (1 - P_v^{IRsub2}) \tag{5.15}$$

where $S_{j,k}$ is a running variable of subset of nodes with $|S_{J,k}|$ equal to k and $P_i^{IRsub2} = P_i^N P_i^{MTDL} P_i^{MTUL}$.

Converging-Raft Reliability within Cellular Network

For Converging-Raft within cellular network, we still consider reliability by analyzing each consensus stages. let $S_{1,x} \subseteq \Omega$ represents the set of non-faulty nodes, $S_{2,y}, S_{3,z}, S_{4,p}, S_{5,q} \subseteq \Omega$ represent the sets of nodes that successfully survive the Topic Establishment, Opinion Expression, Converging Discussion and Opinion Vote stages, respectively. We can still apply Eq. (5.7) to derive the reliability of Converging-Raft in cellular network.

In addition, we use the parameters defined in previous cellular network model, where the reliability of node i is P_i^N, the reliability of the DL between node i and the leader is P_i^{MTDL}, and the reliability of the UL is P_i^{MTUL}. We replace the communication link model used in Converging-Raft within ad hoc network with the MQTT link model which is illustrated in Eq. (5.12). So the success rate of each node in Converging Discussion and Opinion Vote stage is:

$$P_{S\_MT}(a,b) = P_a^N (1 - \prod_{v \in b-1} (1 - P_v^{MTDL})) \tag{5.16}$$

By substituting P_S in Eq. 5.9 and Eq. 5.10 with P_{S_MT}, we can derive the reliability for the Converging Discussion and Opinion Vote stages as P_{CD_Cel} and P_{OV_Cel}, respectively.

As the fault tolerance of Converging-Raft is also $f = \lfloor \frac{n}{2} \rfloor$, the probability that the cluster successfully reaches a Converging-Raft consensus within cellular network P_{CR_Cel}

is the sum of probabilities of all the $(S_{1,x}, S_{2,y}, S_{3,z}, S_{4,p}, S_{5,q})$ satisfying $n \geq x \geq y \geq z \geq p \geq q \geq n-f$ and $\Omega \supseteq S_{1,x} \supseteq S_{2,y} \supseteq S_{3,z} \supseteq S_{4,p} \supseteq S_{5,q}$.

$$\begin{aligned} P_{CR\_Cel} &= P_{CD\_Cel} P_{OV\_Cel} \times \\ &\sum_{n \geq k \geq n-f, \Omega \supseteq S_{J,k}} \prod_{u \in S_{J,k}} P_u^{CRsub2} \prod_{v \in \mathscr{C}_\Omega S_{J,k}} (1 - P_v^{CRsub2}) \end{aligned} \tag{5.17}$$

where $S_{J,k}$ is a running variable of subset of nodes with $|S_{J,k}|$ equal to k and $P_i^{CRsub2} = P_i^N P_i^{MTDL} P_i^{MTUL}$.

CDM Reliability within Cellular Network

The reliability of CDM depends on the reliability of the central server and the number of nodes it connects to. Given that its stability diminishes as the number of nodes increases, we assume it has the following distribution:

$$P_{CDM\_Cel} = P_{CenSta\_max} - \lambda * e^{-k*10^{-N/10}} \tag{5.18}$$

where k is the decay constant and P_{CenSta_max} is the maximum value for central server reliability.

5.4.2 Joint-Decision Latency Analysis of HIntS

Intelligent-Raft Latency within Ad Hoc Network

In ad hoc networks, node and link latency vary with their state changes, affecting overall latency. Since each state transition depends only on the current state, the consensus process, considering node and link latency, satisfies the Markov property. Thus, Markov chains can be used to model the state transition matrix of nodes and links, deriving the steady state to estimate latency expectations. The overall latency is then derived based on the operation of Intelligent-Raft in the ad hoc network. We begin with the state transition matrix of ICA nodes.

We divide nodes into four states, namely $Idle$, $Working$, $Busy$, $Crash$. There is a certain probability that these states can transition between one another. For example, when the node is $Idle$, there is a probability of $p_{IW}^{node} = P(Working|Idle)$ to transfer to the $Working$ state, that is, the ICA node processing consensus data. Similarly, the Idle state also has a probability of $p_{II}^{node} = P(Idle|Idle)$ to maintain its state. When the node is in the $Working$ state, there is a probability of $p_{WI}^{node} = P(Idle|Working)$ that it will finish processing the data and return to the $Idle$ state.

We assume that the status changes between $Idle$, *Working*, and *Busy* must be continuous and cannot jump. For example, *Busy* can only be transferred to *Working* first before it can continue to be transferred to $Idle$. In addition, each node status has a certain probability of $Crash$. In the $Crash$ state, there exists a probability $p_{CI}^{node} = P(Idle|Crash)$ that the system will return to the $Idle$ state. So the state transition probabilities are summarized:

$$\mathbf{P}^{node} = \begin{bmatrix} p_{II}^{node} & p_{IW}^{node} & p_{IB}^{node} & p_{IC}^{node} \\ p_{WI}^{node} & p_{WW}^{node} & p_{WB}^{node} & p_{WC}^{node} \\ p_{BI}^{node} & p_{BW}^{node} & p_{BB}^{node} & p_{BC}^{node} \\ p_{CI}^{node} & p_{CW}^{node} & p_{CB}^{node} & p_{CC}^{node} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & 0 & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & p_{32} & p_{33} & p_{34} \\ p_{41} & 0 & 0 & p_{44} \end{bmatrix} \tag{5.19}$$

$$\sum_{j=1}^{4} p_{ij} = 1, \quad i = 1,2,3,4. \tag{5.20}$$

Since the *Crash* probability of a node depends solely on its current state, we can assume a fixed value for $p_{IC}^{node}, p_{WC}^{node}, p_{BC}^{node}, p_{CC}^{node}$ during the simulation. As the number of nodes increases, the frequency of communication requests rises, making the transition from the $Idle$ state to the *Working* state more likely, and from the *Working* state to the *Busy* state more frequent. We model these transitions using an exponential distribution, as described by $p_{IW}^{node} = (1 - p_{IC}^{node})(1 - e^{-\lambda_W N})$, $p_{WB}^{node} = (1 - p_{WC}^{node})(1 - e^{-\lambda_B N})$. However, both the *Working* state returning to the $Idle$ state and the *Busy* state returning to the *Working* state can be modeled by $p_{WI}^{node} = (1 - p_{WC}^{node})e^{-\lambda_R N}$ and $p_{BW}^{node} = (1 - p_{BC}^{node})e^{-\lambda_R N}$, respectively. The remaining probability is allocated to the likelihood of remaining in its current state.

Next, we can find the steady-state distribution column

$$\pi^{node} = \begin{bmatrix} \pi_{Idle}^{node} & \pi_{Working}^{node} & \pi_{Busy}^{node} & \pi_{Crash}^{node} \end{bmatrix} \tag{5.21}$$

by solving the Markov chain, which satisfies:

$$\pi^{node}\mathbf{P}^{node} = \pi^{node} \tag{5.22}$$

As the latency of a node in each state is different, we can use the node's steady-state probability to calculate the latency expectation. We assume a latency space $\mathbf{D_N} = \begin{bmatrix} D_{Idle} & D_{Working} & D_{Busy} & D_{Crash} \end{bmatrix}^T$, where each value corresponds to the average latency of a node in this state. The latency expectation of each node is:

$$D_{Node} = \pi^{node}\mathbf{D_N} \tag{5.23}$$

We use the same method to analyze link delay expectations. First, we define the state space of communication link in distributed network. $Idle$ means that no node occupies the channel bandwidth resource; $Engaged$ means that there are nodes that are using the link resources; $Congested$ means that there are many nodes occupying resources at the same time, causing communication congestion. So the state transition probabilities are summarized:

$$\mathbf{P}^{link} = \begin{bmatrix} p_{II}^{link} & p_{IE}^{link} & p_{IC}^{link} \\ p_{EI}^{link} & p_{EE}^{link} & p_{EC}^{link} \\ p_{CI}^{link} & p_{CE}^{link} & p_{CC}^{link} \end{bmatrix} \tag{5.24}$$

We also assume that the status changes between each state must be continuous and cannot jump, so p_{IC}^{link} and p_{CI}^{link} equals to zero. In addition, we model the transition from the $Idle$ state to the $Engaged$ state and from the $Engaged$ state to the $Congested$ state using an exponential distribution, represented by $p_{IE}^{link} = 1 - e^{-\lambda_E N}$ and $p_{EC}^{link} = 1 - e^{-\lambda_C N}$, respectively. Recovery probabilities are similarly modeled as $p_{EI}^{link} = e^{-\lambda_R N}$, $p_{CE}^{link} = e^{-\lambda_R N}$.

Then we define the steady-state probability distribution column of communication link as $\pi^{link} = \begin{bmatrix} \pi_{Idle}^{link} & \pi_{Engaged}^{link} & \pi_{Congested}^{link} \end{bmatrix}$ and satisfy:

$$\pi^{link}\mathbf{P}^{link} = \pi^{link} \tag{5.25}$$

Similarly, we assume a latency space $\mathbf{D_L} = \begin{bmatrix} D_{Idle} & D_{Engaged} & D_{Congested} \end{bmatrix}^T$, where each value corresponds to the average latency of the link in this state. We can derive the latency expectation of each link is:

$$D_{Link} = \pi^{link}\mathbf{D_L} \tag{5.26}$$

Next, we analyze the system latency of Intelligent-Raft in a distributed ad hoc network. According to the consensus process shown in Fig. 5.1, when the proposal is initiated by a follower, the system delay is mainly divided into three link delays, three node delays and an intelligent evaluation delay. When the proposal is initiated directly by the Leader, the system delay is mainly divided into two link delays, two node delays and an intelligent evaluation delay. Then we can conclude that the total system latency D_{IDC_AH} is:

$$D_{IR\_AH\_LeaIni} = 2D_{Link} + 2D_{Node} + D_{Evaluation} \tag{5.27}$$

$$D_{IR\_AH\_FolIni} = 3D_{Link} + 3D_{Node} + D_{Evaluation} \tag{5.28}$$

$$D_{IR\_AH} = \frac{1}{n} D_{IR\_AH\_LeaIni} + \frac{n-1}{n} D_{IR\_AH\_FolIni} \tag{5.29}$$

Converging-Raft Latency within Ad Hoc Network

Based on Eq. (5.23) and (5.26), we can construct the Converging-Raft latency model in ad hoc network. Similar to Intelligent-Raft, a consensus may be initiated by either a follower or a leader. During the Opinion Expression, Converging Discussion, and Opinion Vote stages, nodes intelligently evaluate the content received in the previous stage, such as topics and proposals. The latency of these intelligent evaluations is also considered separately. Therefore, the total Converging-Raft latency within the ad hoc network, D_{CR_AH}, is:

$$D_{CR\_AH\_LeaIni} = 4D_{Link} + 4D_{Node} + 3D_{Evaluation} \tag{5.30}$$

$$D_{CR\_AH\_FolIni} = 5D_{Link} + 5D_{Node} + 3D_{Evaluation} \tag{5.31}$$

$$D_{CR\_AH} = \frac{1}{n} D_{CR\_AH\_LeaIni} + \frac{n-1}{n} D_{CR\_AH\_FolIni} \tag{5.32}$$

Intelligent-Raft Latency within Cellular Network

For the latency characteristics of Intelligent-Raft based on cellular network, we build a Markov chain model for the server Broker in the MQTT protocol to describe latency expectations through its steady-state matrix. First, we assume the Broker's state space. We divide the Broker's status into 5 states, namely S1: the message has not reached the Broker; S2: the message has arrived at the Broker and is waiting to be processed; S3: the message is being processed; S4: the message is processed and is waiting to be sent to client; S5: Message sent. The state transition probabilities are summarized:

$$\mathbf{P} = \begin{bmatrix} p_{11}^B & p_{12}^B & 0 & 0 & 0 \\ 0 & p_{22}^B & p_{23}^B & 0 & 0 \\ 0 & 0 & p_{33}^B & p_{34}^B & 0 \\ 0 & 0 & 0 & p_{44}^B & p_{45}^B \\ p_{51}^B & 0 & 0 & 0 & p_{55}^B \end{bmatrix} \tag{5.33}$$

Different from the Markov chain model in ad hoc network, this model includes not only the processing and waiting delays of the Broker itself, but also the delays of message arrival and sending. For example, the transfer from S1 to S2 depends on network traffic and the Broker's ability to receive messages. The transfer from S2 to S3 depends on the current workload and processing capabilities of the Broker. The transfer from S3 to S4 is determined by message processing time. The transfer from S4 to S5 depends on the broker's ability to send messages and network traffic. So we do not define the delay in each state here, but focus on the delay of each state transition. That is, for each non-zero element p_{ij} in the state transition matrix $\mathbf{P}$, a corresponding transition delay τ_{ij} is determined, which represents the average time it takes to transition from state i to state j. Based on this, we create a transfer delay matrix $\mathbf{T}$ corresponding to the state transition matrix $\mathbf{P}$.

$$\mathbf{T} = \begin{bmatrix} \tau_{11} & \tau_{12} & 0 & 0 & 0 \\ 0 & \tau_{22} & \tau_{23} & 0 & 0 \\ 0 & 0 & \tau_{33} & \tau_{34} & 0 \\ 0 & 0 & 0 & \tau_{44} & \tau_{45} \\ \tau_{51} & 0 & 0 & 0 & \tau_{55} \end{bmatrix} \tag{5.34}$$

Since all nodes must connect to the Broker server and wait for message distribution, the time required for the Broker to buffer or process messages increases as the number of nodes grows. Thus, we can model this as $p_{12} = e^{-\lambda_{12}N}, p_{23} = e^{-\lambda_{23}N}, p_{34} = e^{-\lambda_{34}N}, p_{45} = e^{-\lambda_{45}N}$. Additionally, we assume that p_{51} remains constant.

We define the steady-state probability distribution column of communication link as $\pi = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 & \pi_5 \end{bmatrix}$ and satisfy:

$$\pi \mathbf{P} = \pi \tag{5.35}$$

In this case, the calculation of the overall delay expectation will no longer be a simple weighted sum of state delays, but a weighted sum of state transition delays. Then the overall delay expectation $E[T]$ can be expressed as:

$$D_{Broker} = E[T] = \sum_{i=1}^{n} \sum_{j=1}^{n} \pi_i p_{ij} \mathbf{T}_{ij} \tag{5.36}$$

Where n is the total number of states, π_i is the probability of being in state i for a long time, p_{ij} is the probability of transitioning from state i to state j, and $\mathbf{T}_{ij}$ is the corresponding transition delay.

Next, we analyze the system throughput of Intelligent-Raft within cellular network. Similarly, a consensus may be initiated by either a follower or a leader. Then, we can conclude that the total system latency D_{IR_Cel} is:

$$D_{IR\_Cel\_LeaIni} = 2D_{Broker} + 2D_{Node} + D_{Evaluation} \tag{5.37}$$

$$D_{IR\_Cel\_FolIni} = 3D_{Broker} + 3D_{Node} + D_{Evaluation} \tag{5.38}$$

$$D_{IR\_Cel} = \frac{1}{n} D_{IR\_Cel\_LeaIni} + \frac{n-1}{n} D_{IR\_Cel\_FolIni} \tag{5.39}$$

Converging-Raft Latency within Cellular Network

Based on (5.23) and (5.36), we can construct the latency model in a cellular network following the Converging-Raft process. Since the Broker latency model encompasses both the Broker's inherent latency and the data transmission latency, we can get the total system latency D_{CR_Cel}:

$$D_{CR\_Cel\_LeaIni} = 4D_{Broker} + 4D_{Node} + 3D_{Evaluation} \tag{5.40}$$

$$D_{CR\_Cel\_FolIni} = 5D_{Broker} + 5D_{Node} + 3D_{Evaluation} \tag{5.41}$$

$$D_{CR\_Cel} = \frac{1}{n} D_{CR\_Cel\_LeaIni} + \frac{n-1}{n} D_{CR\_Cel\_FolIni} \tag{5.42}$$

CDM Latency within Cellular Network

Here, we use the latency model of Broker server as the central control server. According to the CDM working process, the latency mainly depends on the central server and the decision-making. We can derive the total latency of CDM within cellular network:

$$D_{CDM\_Cel} = D_{Broker} + D_{Evaluation} \tag{5.43}$$

5.5 Simulation Numerical Results And Discussion

This section presents a performance analysis of the reliability and latency of HIntS's five working modes, with the key results summarized in Section 5.2.

5.5.1 HIntS Reliability

First, we analyze the simulated reliability results of CDM, Intelligent-Raft, and Converging-Raft within two network structures: ad hoc network and cellular network. In our model, we decouple node failure probability from link failure probability and use the control variable method to individually assess their impact on system reliability. Although the reliability of nodes, uplink, and downlink are modeled as random variables, we assume their mathematical expectation to be $P^N = P^{UL} = P^{DL} = P_{BrokerS} = 0.95$ and $P^{MTlinkS} = 0.995$ by default.

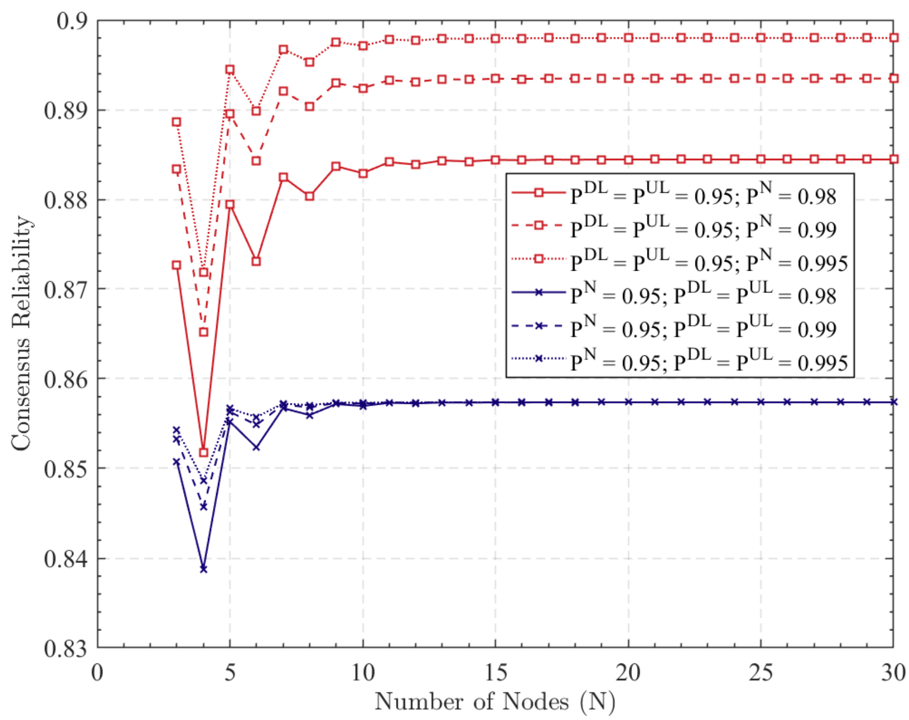

Figure 5.4: Intelligent-Raft reliability in ad hoc network

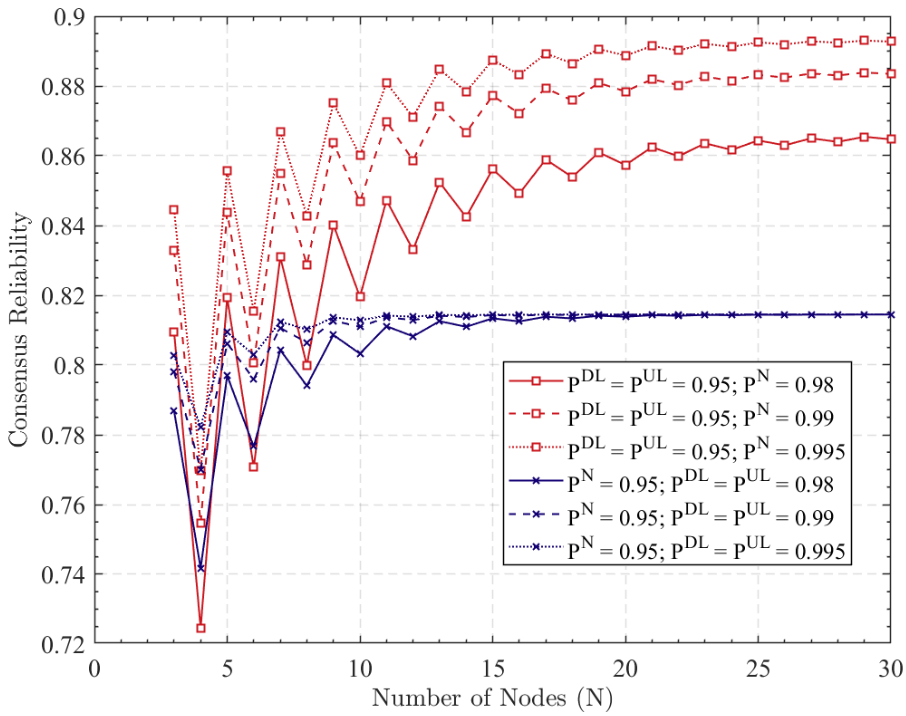

Figure 5.5: Converging-Raft reliability in ad hoc network

Figs. 5.4, 5.5, 5.6, and 5.7 illustrate that as the number of nodes participating in the joint-decision process increases, the reliability of both Intelligent-Raft and Converging-Raft improves, though with some fluctuations across different networks. The simulation results show that all reliability factors—node reliability, link reliability, MQTT link reliability, and broker reliability—contribute positively to the overall reliability of the joint-decision mechanisms. As these factors improve, the overall reliability of HIntS exhibits a consistent upward trend.

In Figs. 5.4 and 5.5, the red and blue curves represent the impact of varying node reliability and communication link success rates on IDC mechanisms, respectively, while holding the other factor constant. In ad hoc networks, whether using Intelligent-Raft or Converging-Raft, node reliability directly influences the maximum value to which joint-decision reliability converges. In contrast, link reliability has minimal impact on this maximum value, primarily affecting the amplitude of reliability fluctuations. Figs. 5.6 and 5.7 show that broker reliability has a greater impact on the final converged value of IDC mechanisms compared to MQTT link reliability in a cellular network, while keeping node reliability constant.

Fig. 5.8 compares the reliability of HIntS's five working modes under the conditions

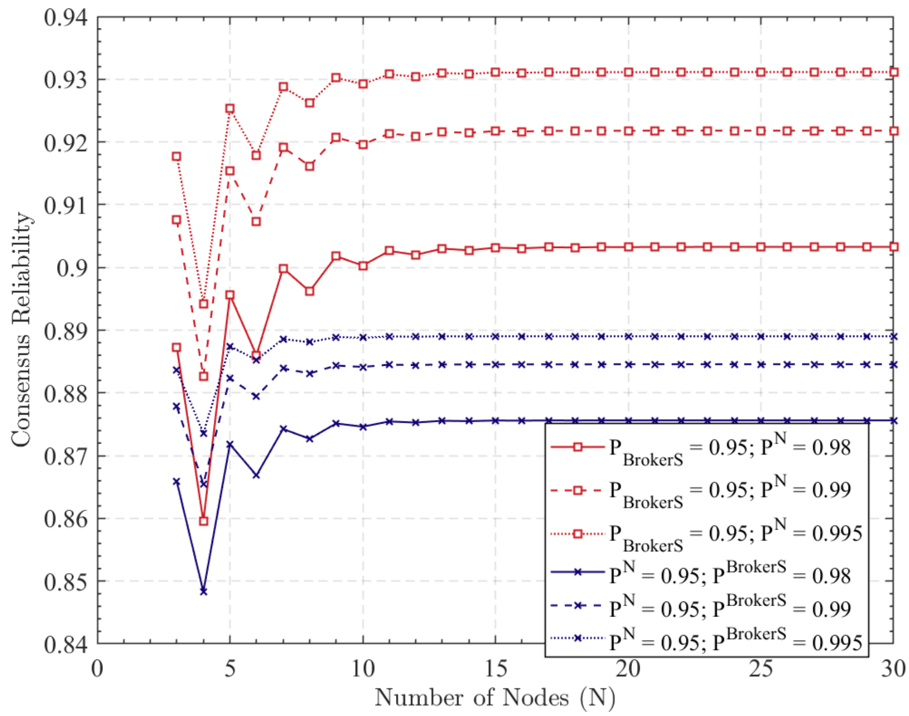

Figure 5.6: Intelligent-Raft reliability in cellular network

$P^{UL} = P^{MTlinkS} = P^{DL} = 0.98$, and $P^N = P_{BrokerS} = 0.99$. As the number of nodes increases, the reliability of joint-decision under various conditions also increases where their order of magnitude remains unchanged, with the exception of CDM. Since the reliability of CDM is limited by the stability of the central server, its stability gradually decreases as the number of connected nodes increases. In other cases, the reliability follows this order: Intelligent-Raft within cellular network $>$ Intelligent-Raft within ad hoc network $>$ Converging-Raft within cellular network $>$ Converging-Raft within ad hoc network. This comparison also shows that Converging-Raft is less stable and reliable than Intelligent-Raft under the same conditions, highlighting the need for heterogeneous joint-decision to leverage different strengths for various requirements. Additionally, the same joint-decision mechanism is more stable in a cellular network compared to an ad hoc network.

5.5.2 HIntS Latency

Then, we analyze the simulated latency results of HIntS's five working modes. In ad hoc network, latency is primarily composed of three components: node delay, communication link delay, and intelligent evaluation delay. We simulate the Markov chain models for

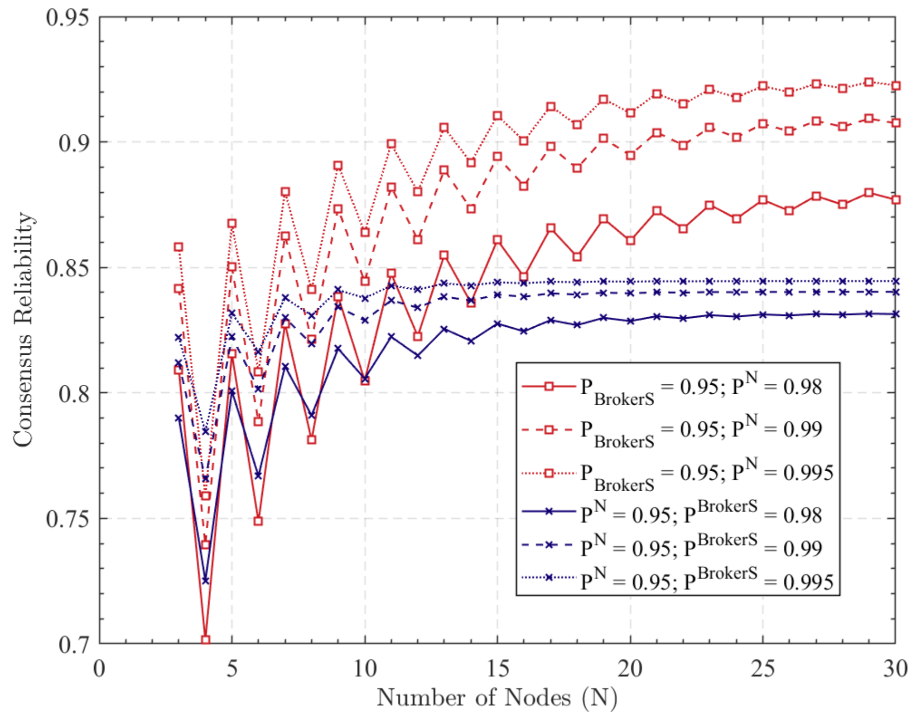

Figure 5.7: Converging-Raft reliability in cellular network

nodes and links separately, analyzing the convergence of their state transition matrices to obtain delay expectations, which are then used to calculate the overall joint-decision latency. We assume the latency space of the nodes as:

$$\mathbf{D_N} = \begin{bmatrix} D_{Idle} & D_{Working} & D_{Busy} & D_{Crash} \end{bmatrix}^T = \begin{bmatrix} 10 & 15 & 25 & 1000 \end{bmatrix}^T \tag{5.44}$$

and the latency space of the communication links as:

$$\mathbf{D_L} = \begin{bmatrix} D_{Idle} & D_{Engaged} & D_{Congested} \end{bmatrix}^T = \begin{bmatrix} 10 & 15 & 25 \end{bmatrix}^T \tag{5.45}$$

with all values in milliseconds (ms). We also assume the exponential distribution parameters for node model to be $\lambda_W = 0.1, \lambda_B = 0.15, \lambda_R = 0.2$, and $p_{IC}^{node} = 0.01, p_{WC}^{node} = 0.03, p_{BC}^{node} = 0.05, p_{CC}^{node} = 0.7$. For link model, we assume that $\lambda_E = 0.1, \lambda_C = 0.15, \lambda_R = 0.2$, and $p_{II}^{link} = p_{EE}^{link} = p_{CC}^{link} = 0.4$.

In the cellular network with MQTT protocol, latency is primarily composed of three components: node delay, Broker state delay, and intelligent evaluation delay. Since our model of the Broker accounts for the delays in the MQTT uplink, Broker forwarding,

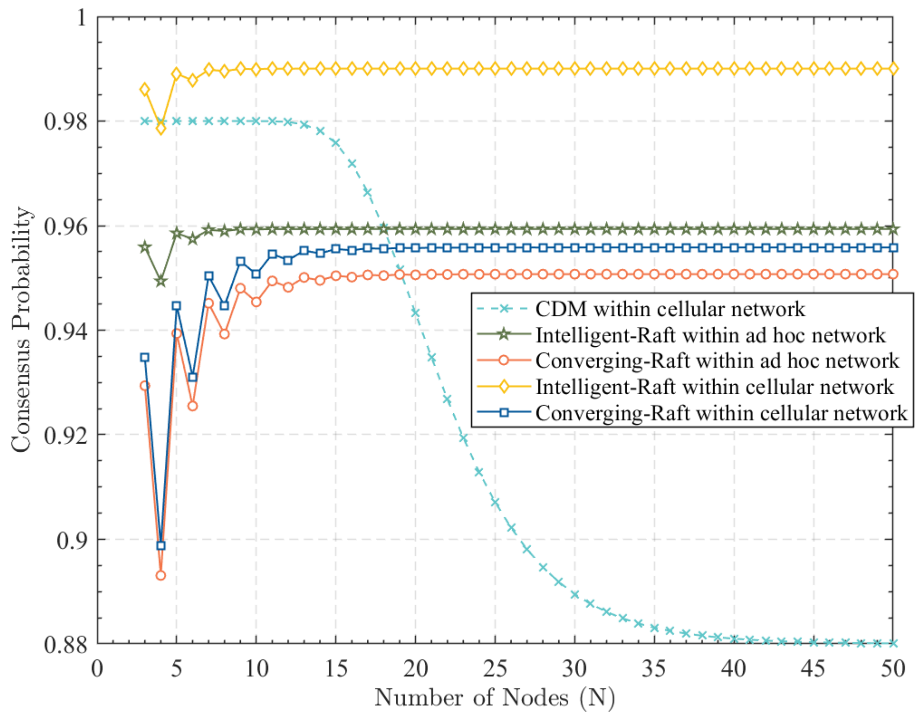

Figure 5.8: Different joint-decision modes reliability within different network

and MQTT downlink processes, the steady-state delay of the Broker can be considered equivalent to link delay in an ad hoc network. We assume the distribution parameters to be $\lambda_{12} = \lambda_{23} = \lambda_{34} = \lambda_{45} = 0.1$, and $p_{51}^{B} = 0.95$. The corresponding transfer delay matrix of the Broker is then assumed to be:

$$\mathbf{T} = \begin{bmatrix} 20 & 10 & 0 & 0 & 0 \\ 0 & 15 & 10 & 0 & 0 \\ 0 & 0 & 15 & 10 & 0 \\ 0 & 0 & 0 & 20 & 10 \\ 10 & 0 & 0 & 0 & 5 \end{bmatrix} \tag{5.46}$$

By calculating the convergence delay expectation of nodes, links and the Broker from the steady state, we can calculate the total latency across different HIntS working modes. Fig. 5.9 shows the convergence process of the five working modes. The delay is the highest for Converging-Raft within cellular network, followed by Converging-Raft within ad hoc network, Intelligent-Raft within cellular network, Intelligent-Raft within ad hoc network, CDM within cellular network. The comparison reveals that, for the same joint-decision mechanism, total latency is higher in the cellular network than in the ad hoc network. This

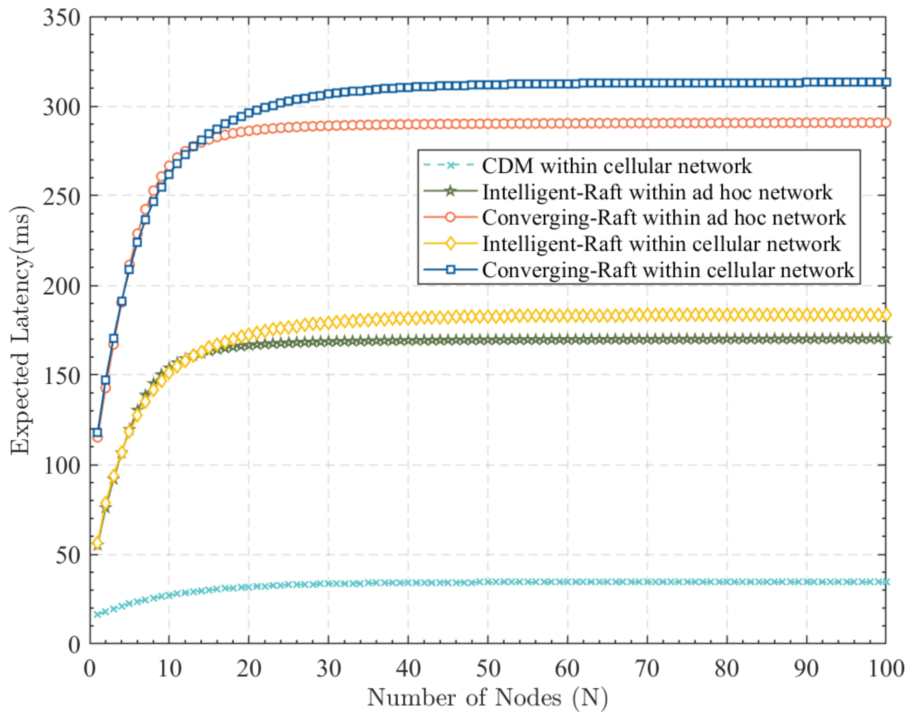

Figure 5.9: Overall Latency

is because the greater routing path between nodes in cellular network leads to longer data forwarding times, while ad hoc networks typically use direct communication over shorter distances, resulting in lower latency. Additionally, within the same network, Converging-Raft has higher latency than Intelligent-Raft due to its more complex process. CDM within cellular network has the lowest latency, given its simplicity.

The results demonstrate that different HIntS working modes have distinct advantages and drawbacks. For example, Converging-Raft within a cellular network supports a broad network range and can achieve a globally optimal joint decision, but it also exhibits the highest latency. CDM within cellular network is the easiest to deploy and has the lowest latency, but it lacks fault tolerance, with the system's performance being heavily reliant on the central server. These findings underscore the importance of HIntS, which integrates heterogeneous joint-decision mechanisms and network structures to ensure optimal system performance in various applications.

5.6 Experiments

In this section, We practically test the reliability, latency of practical HIntS, 5G-MInd. The following paragraphs introduce the setup and the process of the experiments. Finally, we analyze the experimental data.

5.6.1 Experimental Setup

Hardware Setup

In the experiment, the ZigBee module manufactured by DIGI carried by the 5G-MInd supports half-duplex mode, with a data speed of 250 Kb/s , a transmission power of 20 dBm, a transmission frequency of ISM 2.4GHz. We set the MCU's UART baud rate to a maximum value, 921600, for communication with the ZigBee module. In addition, the MCU (STM32F407ZGT6) has an operating principal frequency of 168MHz, external expansion heap memory size is 1MB, and the SD card size is 4GB for storing history consensus log and data.

For the 5G module, we use the 5G module RM500U manufactured by Quectel. Its operating frequency is Sub-6G Hz and is compatible with LTE mode. Under the 5G-SA architecture, the maximum DL is 2Gbps and the maximum UL is 1Gbps; under the 5G-NSA architecture, the maximum DL is 2Gbps and the maximum UL is 575Mbps. Its interactive mode uses 3GPP TS27.007 and Quectel Enhanced AT commands.

Experiment 1

In the first experiment, we examine how varying the number of 5G-MInd nodes and their distances impact the joint-decision success rate. We tested the success rates of Intelligent-Raft and Converging-Raft under ad hoc network based on ZigBee and 5G networks. CDM is not tested due to the insufficient number of modules required to reach its scalability limit. In both protocols, consensus is reached when the Leader receives successful feedback from more than $\frac{N-1}{2}$ nodes at the final stage. For this experiment, we first fixed the Leader's position within the 5G-MInd nodes and evenly placed follower nodes on a ring with a radius of 300 to 400 meters around the Leader. To minimize signal interference among followers, a certain distance was maintained between each node. Once all 5G-MInd nodes were powered on, their wireless modules automatically established either the ad hoc network or the 5G cellular network based on preset parameters, readying the system for joint-decision processes. We conducted three rounds of experiments in each network environment, averaging the results to determine the final consensus success rate. In each round, we tested the joint-decision success rate with 3 to 9 nodes, triggering consensus initialization 200 times per test and recording the frequency of successful joint-decisions.

Experiment 2

In research of the throughput of 5G-MInd, we used varying numbers of 5G-MInd modules to evaluate performance. We measured the latency of Intelligent-Raft and Converging-Raft in both ZigBee network and 5G network. We positioned the leader node at the center and evenly placed follower nodes on a ring with a radius of 300 meters. Once the network was established, a node could initiate a round of joint-decision with the leader to start the Intelligent-Raft process (excluding cases where the leader directly initiates a consensus). We assessed 5G-MInd's performance by testing the time required for different numbers of nodes to reach consensus across various network environments. Each test was repeated 200 times, with only the time taken to successfully reach a joint-decision averaged as the result.

5.6.2 Experimental Results

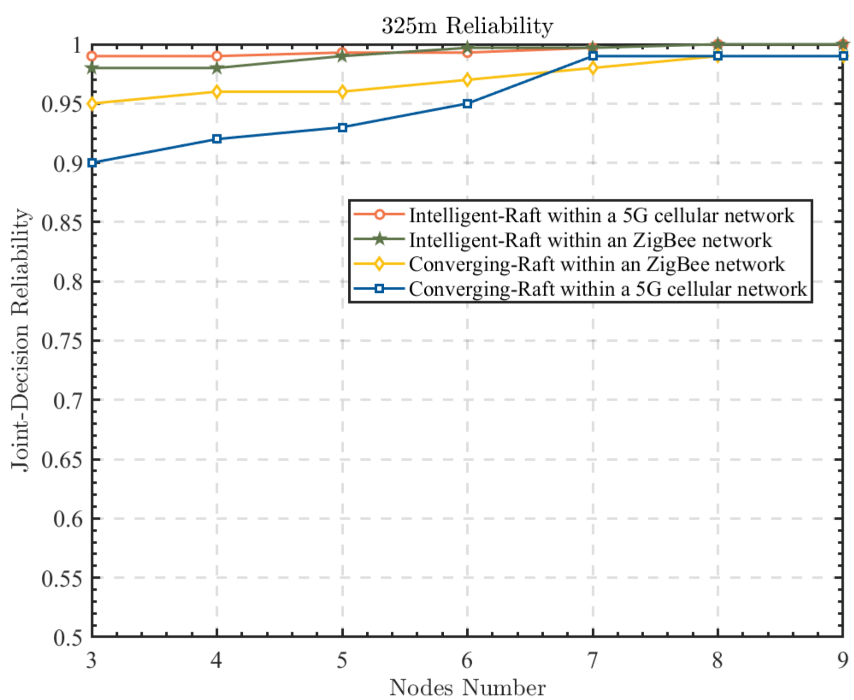

Figure 5.10: 5G-MInd reliability within 325m

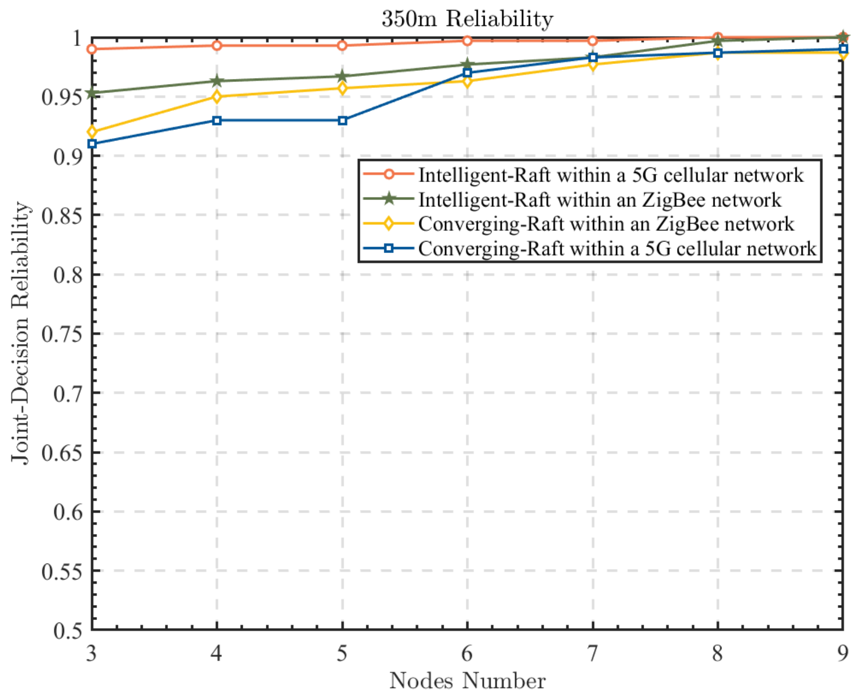

Figure 5.11: 5G-MInd reliability within 350m

Fig. 5.10, 5.11, 5.12 and 5.13 shows the success rate of joint-decision with varying numbers of 5G-MInd modules at different distances. The success rate increases as the number of nodes grows. At a distance of 325 meters from the leader, the reliability across different network conditions and joint-decision modes aligns with theoretical predictions: Intelligent-Raft within a 5G cellular network > Intelligent-Raft within a ZigBee ad hoc network > Converging-Raft within a 5G cellular network > Converging-Raft within a ZigBee ad hoc network. At 350 meters, the reliability of Intelligent-Raft and Converging-Raft within the ZigBee network is slightly lower than at 325 meters but still consistent with simulation results. However, at 375 meters and 400 meters, the reliability of joint-decision in the ZigBee network significantly drops, as power dissipation, noise interference hinder effective data transmission over such distances. This highlights that joint-decision completion in ad hoc networks is heavily distance-dependent. In contrast, the success rate in 5G cellular networks remains nearly unaffected by distance.

Fig. 5.14 shows the average latency results for 5G-MInd nodes with varying numbers of participants. Consensus time increases across all modes, except CDM within the cellular network, as the number of nodes grows. This is due to the leader needing to coordinate more followers and taking longer to receive and tally responses. The experimental results confirm that the average consensus time follows the order: Converging-Raft in 5G cellular

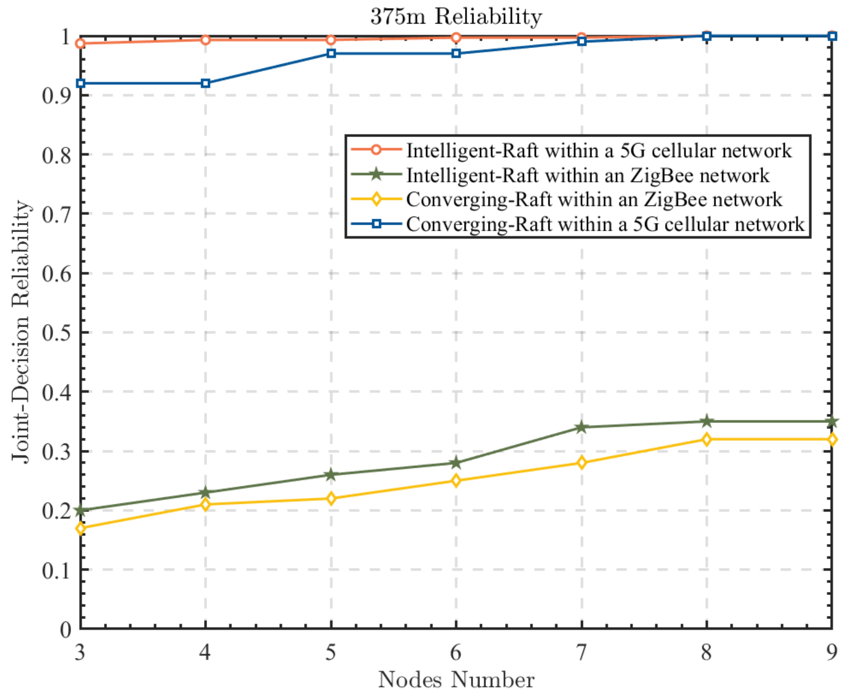

Figure 5.12: 5G-MInd reliability within 375m

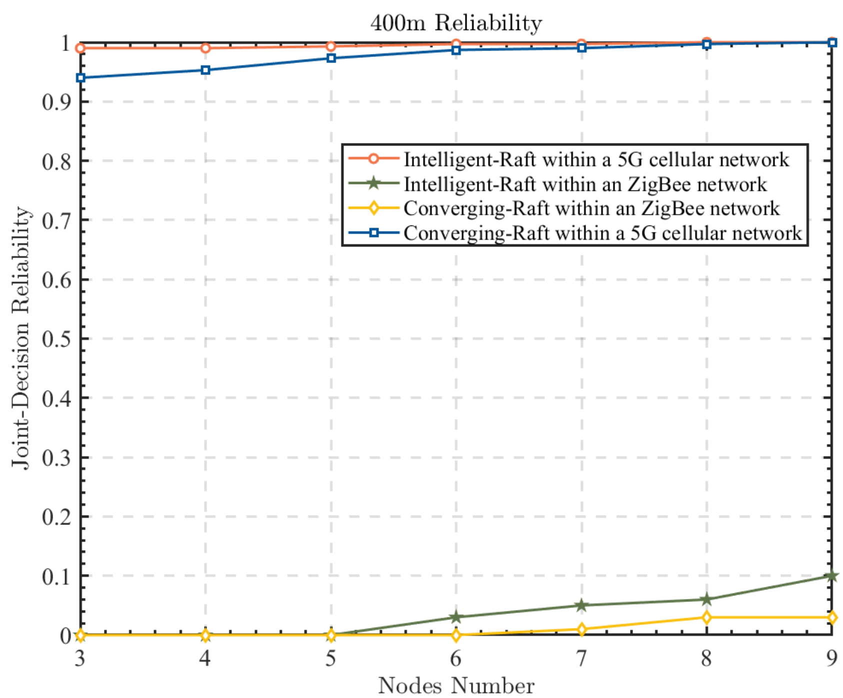

Figure 5.13: 5G-MInd reliability within 400m

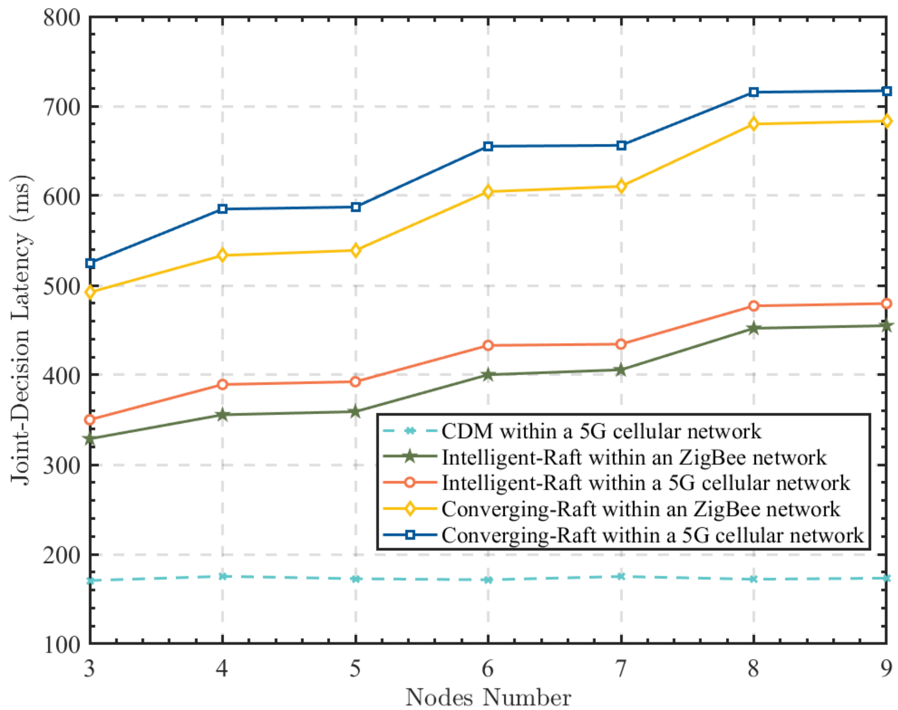

Figure 5.14: 5G-MInd latency

network > Converging-Raft in ZigBee ad hoc network > Intelligent-Raft in 5G cellular network > Intelligent-Raft in ZigBee ad hoc network > CDM in 5G cellular network, aligning with theoretical simulations.

This experiment demonstrates that different networks and joint-decision protocols offer distinct advantages in fault tolerance reliability, latency and coverage. The experimental data validates our theoretical findings and offers valuable insights and data for the practical deployment of joint-decision mechanisms.

5.7 Conclusion

In this chapter, we first proposed Converging-Raft, an IDC protocol that enables ICA nodes to achieve globally optimal joint decisions through discussions. Additionally, we introduced HIntS, a heterogeneous system that integrates three joint-decision mechanisms (CDM, Intelligent-Raft, and Converging-Raft) with two network structures (ad hoc and cellular networks) to enhance adaptability for ICA nodes. We also designed and implemented 5G-MInd, a fully functional HIntS physical module built on an embedded platform. Our mathematical models provided a detailed analysis of the reliability and latency performance across HIntS's five working modes. Simulation results demonstrated that Intelligent-Raft in cellular networks achieves the highest reliability, while CDM in cellular networks offers the lowest latency. Experimental validation using 5G-MInd further confirmed the alignment between the simulated and experimental results. By combining quantitative and qualitative analyses, we summarized the advantages and characteristics of different joint-decision mechanisms across various network structures, providing valuable guidance for future practical deployments and demonstrating the significant potential of HIntS for industrial CAS applications.

Despite these contributions, several limitations warrant attention. While Converging-Raft aims to achieve globally optimal decision-making, its practical implementation remains challenging due to potential information asymmetry and coordination difficulties among distributed nodes. Furthermore, although HIntS embraces multiple decision mechanisms to improve adaptability, the integration and dynamic switching between heterogeneous mechanisms in complex real-world scenarios introduce additional system complexity that requires further investigation. Nonetheless, this work provides a strong foundation for future research on scalable, adaptive, and high-reliability joint decision-making in industrial CAS environments.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis provides a comprehensive study on the characteristics of wireless IDC, its performance across various wireless networks, and the requirements and optimizations needed for diverse application scenarios, particularly autonomous driving. First, two novel IDC algorithms, Intelligent-Raft and Converging-Raft, are introduced in Chapters 3 and 4, respectively, to address limitations of traditional DC algorithms within CAS systems. Intelligent-Raft addresses the traditional Raft algorithm's inability to enable nodes to intelligently assess data requiring consensus by introducing an Intelligent Evaluation process. Converging-Raft expands on traditional Raft by adding steps such as Converging Discussion, allowing ICA nodes to achieve a globally optimal solution on specific topics through sharing, comparison, updates, and voting. The achievement of this global optimal solution integrates the wisdom of all ICA nodes in the cluster. These two IDC protocols offer distinct advantages and can be adapted to different CAS scenario.

In addition, this thesis also introduces two system architectures to support the practical deployment of IDC: WIDCS and HIntS. WIDCS implements the PICA architecture within CAS, leveraging wireless communication and the Intelligent-Raft protocol to provide ICA nodes with data consistency storage and joint decision-making capabilities. Chapter 4 enhances WIDCS by incorporating ad hoc network functions such as formation, management, and dissolution, thereby improving its engineering feasibility and expanding its utility for CAS applications. To address the complex requirements that cannot be met by a single IDC protocol or network architecture, Chapter 5 presents HIntS, which integrates three joint decision-making protocols—CDM, Intelligent-Raft, and Converging-Raft—and two network architectures: ad hoc and cellular networks. HIntS can adaptively select the appropriate protocol and network structure to optimize performance based on specific application scenarios. Chapter 5 also provides a detailed mathematical analysis of HIntS performance across various operating modes, offering valuable guidance for future deployment of joint decision-making systems.

Moreover, this thesis also introduces the design and iteration of three generations of hardware modules to enable wireless joint decision-making in practical applications. The first-generation system, AIR-RAFT, implements the basic functions of WIDCS via P2P communication based on the Lora protocol, as detailed in Chapter 3. The second-generation system, RaBee, extends AIR-RAFT with an added ZigBee module to enable ad hoc networking, enhancing both the stability and throughput of joint decision-making, as discussed in Chapter 4. The third-generation system, 5G-MInd, is developed based on the new HIntS architecture and described in Chapter 5. This system integrates ZigBee and Quectel 5G modules to support both ad hoc and cellular networking functions, respectively. By incorporating the three protocols within HIntS, 5G-MInd can facilitate various joint decision-making processes. Chapter 5 also presents experimental tests conducted using 5G-MInd, with results aligning closely with theoretical predictions.

To verify the feasibility and performance of the proposed IDC protocol and system architecture, this paper analyzes two AD scenarios and presents safe passage solutions enabled by WIDCS. First, at uncontrolled intersections, AVs can achieve consistency in passage order using WIDCS. Mathematical analysis and comparison with scenarios lacking a communication system confirm that this ordered approach enhances AV safety. Similarly, in on-ramp merging scenarios, AVs can reach a joint decision on merging order through WIDCS, allowing for sequential passage. Comparative mathematical analyses across various conditions demonstrate that the probability of AVs safely navigating on-ramp merges is higher with WIDCS than with CDM or without network support. This thesis implements AVs' safe passage at uncontrolled intersections and on-ramp merging scenarios based on joint decisions in a laboratory environment, using AIR-RAFT and RaBee hardware modules. Experimental data further validate the feasibility and optimization achieved by WIDCS.

However, this programme also presents several inherent limitations. In particular, the cross-layer and interdisciplinary nature of WIDCS requires strong integration across diverse fields such as wireless communication, distributed consensus, networked systems, and autonomous driving. Achieving effective collaboration across these domains demands substantial cross-disciplinary expertise and often results in long and complex research cycles, making it difficult to address all aspects comprehensively. Moreover, due to the limited real-world deployment of highly intelligent autonomous driving systems, it remains challenging to validate our theoretical and hardware research—such as WIDCS protocols and embedded consensus modules—in practical, large-scale scenarios. The lack of industrial-scale platforms also makes collaboration with industry challenging, increasing the risk that the research may diverge from practical application needs.

6.2 Future Trends

6.2.1 Evolution of IDC Protocol and Adaptive Algorithm

Although this thesis has developed the Intelligent-Raft and Converging-Raft protocols, and has explored the application of Intelligent-Raft in two autonomous driving scenarios—uncontrolled intersections and on-ramp merging—many other CAS scenarios remain to be optimized. These diverse contexts highlight the need for future research to develop more flexible and application-aware IDC protocols capable of addressing the varied demands of intelligent agents in complex and dynamic environments. Beyond the AD scenarios analyzed in this paper, IDC holds significant potential for other applications, such as roundabouts. In roundabouts, AVs encounter intricate traffic flows and yielding rules, requiring real-time decision-making in multi-vehicle interactions. Vehicle intentions (e.g., entering, navigating, or exiting the roundabout) directly impact safety. Using the IDC protocol, vehicles approaching the roundabout can share their intentions, speeds, positions, and projected paths in real-time. This collective data enables each vehicle to anticipate and interpret the potential actions of others, reducing the reliance on individual sensor data and providing a comprehensive understanding of the surrounding dynamic environment. This shared awareness minimizes misjudgments arising from uncertainty. Additionally, the complex pathways within roundabouts necessitate precise avoidance and safe passage within limited space. By sharing and optimizing path decisions, IDC protocols enable each vehicle to choose a safer, more efficient path. This coordination allows vehicles to adjust paths relative to nearby traffic before entering the roundabout, maximizing traffic flow and minimizing collision risks associated with excessive speed or improper lane selection.

Beyond autonomous driving, IDC demonstrates strong application potential across various CAS scenarios, such as drone swarms and distributed logistics and warehousing. For drone swarms performing tasks like disaster search and rescue, environmental monitoring, and logistics transport, real-time collaboration and data sharing are essential within the mission area. IDC enables drones to coordinate route planning, task allocation, and obstacle avoidance in dynamic environments. Through IDC, drones can rapidly reach consensus on optimal paths or task distribution, reducing delays and collision risks. In cases of drone malfunction, IDC allows for swift reallocation of tasks among other drones, ensuring mission continuity.

Similarly, in large-scale logistics and warehousing, multiple automated sorting robots require coordination for sorting, packaging, and transport. IDC facilitates rapid task alignment among robots, preventing redundant actions and path conflicts. During peak periods, IDC can dynamically adjust resource and task allocations based on real-time data, improving overall logistics efficiency. Furthermore, IDC supports real-time monitoring of device status, ensuring that tasks can be immediately reassigned to functioning robots when others fail, thereby maintaining operational stability and efficiency.

As space exploration missions grow increasingly ambitious, the complexity of managing distributed systems in harsh and unpredictable environments becomes a significant challenge. Tasks such as spacecraft formation control, resource coordination, and real-time autonomous decision-making are difficult to achieve through traditional centralized methods, which often struggle with scalability, responsiveness, and resilience. IDC offers a promising framework to address these challenges by enabling decentralized coordination and collaboration among spaceborne agents. In spacecraft formations and constellations—critical for applications such as Earth observation, deep space exploration, and communication—IDC allows satellites to autonomously negotiate trajectory adjustments, share resources, and recover from faults without relying on constant ground control. Similarly, for deep space probe networks operating under extreme latency and bandwidth constraints, IDC enables localized, autonomous decision-making, real-time data fusion, and adaptive task allocation. By reducing dependence on Earth-based intervention, IDC significantly enhances the robustness, adaptability, and efficiency of distributed space systems, demonstrating strong potential for future space missions.

Beyond the development of customized IDC protocols for specific scenarios, future research will focus on heterogeneous IDC systems and adaptive algorithms. The goal is for future smart devices to progress toward human-like decision-making, capable of independently and flexibly reaching joint-decisions without reliance on a single protocol. Adaptive algorithms will enable devices to autonomously assess scenario requirements and adjust the joint decision-making process, achieving "adaptive protocol selection" and reaching consensus without the need for centralized control. This advancement promises a more responsive and intelligent framework for distributed decision-making across diverse applications.

6.2.2 Intelligent Wireless Network Management Protocol

In Chapter 4, we introduced a network management protocol integrated within WIDCS to improve its operational reliability and adaptability under dynamic conditions. This integration enhanced the system's ability to handle node fluctuations and communication uncertainty. However, this general-purpose protocol, such as ZigBee, still falls short in fully supporting the stringent demands of consensus execution and joint decision-making in large-scale ICA networks. Given the tight coupling between network behavior and distributed coordination performance, there is a pressing need for a dedicated network management protocol tailored specifically to the needs of WIDCS—one that facilitates flexible and efficient node management, resource adaptability, robust security, rapid consensus formation, and ensures robust node participation throughout the decision-making process.

In highly dynamic scenarios, such as when AVs enter or leave a designated area, the network must quickly identify, verify, and integrate new nodes, ensuring that exiting nodes do not disrupt ongoing consensus processes. To achieve this, the protocol must support automatic discovery and rapid verification. Upon a new node's entry, the protocol should automatically detect and authenticate it to confirm compliance with security and resource requirements. Following authentication, network addresses and resources are swiftly allocated to enable seamless integration into network activities. Additionally, a distributed node status monitoring system is essential for real-time tracking of node availability, resource status, and task execution. As nodes enter or exit the network, the system should automatically update the topology and reorganize resource allocations and communication paths. To prevent disruptions during consensus processes, the protocol should also implement a locking mechanism: nodes engaged in consensus are temporarily locked, restricting exit or disconnection until consensus is reached. This ensures decision consistency and integrity.

Secondly, in complex and dynamic environments, network nodes often experience imbalanced load and resource demands. A new network management protocol should therefore support dynamic resource management to meet varying node requirements. The protocol should allocate network bandwidth, storage, and computing resources based on each node's current status, data traffic, and computing tasks. Through resource monitoring and forecasting, it can adjust allocations according to task priority and real-time demand, optimizing efficiency under high load conditions. Additionally, with dynamic path selection and load balancing, the protocol can intelligently distribute communication tasks to avoid overloading or resource depletion in certain nodes. Furthermore, a hierarchical resource control strategy is needed to dynamically adapt node resource usage according to network scale and evolving scenario demands.

Furthermore, the new network management protocol should offer enhanced security and fault tolerance. As ICA nodes operate in increasingly open and potentially hostile environments, the protocol must be resilient against malicious intrusions, data tampering, and signal interference. Additionally, it should include dynamic fault detection and repair mechanisms to promptly isolate faults and reconfigure the network in response to anomalies in any device or node, thereby ensuring system continuity and stability.

6.2.3 AI-enhanced ICA Nodes Interaction and Cooperation

At the current stage, the IDC protocols developed in this work primarily utilize AI inference capabilities as auxiliary tools for intelligent evaluation. However, the integration between IDC mechanisms and AI models remains relatively superficial. In future research, IDC should move toward a deeper fusion with AI—enabling intelligent reasoning not merely as a supporting component, but as a core driver of distributed consensus processes. Such integration has the potential to enhance decision quality, adaptivity, and autonomy, particularly in complex and uncertain environments. AI can support IDC protocols in achieving real-time optimization and adapting automatically to changes in network topology and node status. For instance, in complex scenarios with node failures or communication disruptions, AI-enhanced IDC can employ anomaly detection and deep learning to continuously monitor node health and automatically identify faulty or Byzantine nodes. Upon fault detection, the protocol can use ML models to predict optimal repair methods and paths, quickly reconfiguring the network to ensure system stability and security.

Moreover, while IDC protocols rely on fixed consensus processes, AI-enhanced IDC can learn node behaviors and interaction patterns, generating adaptive consensus strategies for the current environment. Through reinforcement or federated learning, AI enables IDC protocols to make intelligent joint-decisions autonomously, allowing each node to operate with incomplete information and continuously optimize strategies through distributed collaboration.

Future IDC protocols will also need to support cross-platform collaboration, facilitating seamless interaction among various ICA systems (e.g., drones, smart transportation, and smart grids). AI-enhanced IDC can achieve cross-platform compatibility by using reinforcement learning and multi-objective optimization to adjust communication protocols and data formats intelligently. Each system maintains independent consensus processes and optimization strategies while enhancing the ecosystem's connectivity and intelligence through selective sharing of strategies and information.

Looking forward, we envision intelligent nodes evolving beyond the limitations of predefined protocols to engage in highly flexible, human-like deliberations. These nodes could autonomously exchange data, discuss critical tasks, and achieve joint decisions dynamically, adapting seamlessly to diverse scenarios. Leveraging semantic communication, these systems could go beyond mere data transmission to exchange contextually rich information, enabling deeper mutual understanding and more nuanced decision-making processes. By interpreting intent and meaning behind the data, nodes could achieve higher levels of efficiency and safety in their interactions, surpassing human capabilities in both speed, reliability and accuracy.

Bibliography

[1] M. Castro, B. Liskov *et al.*, “Practical byzantine fault tolerance,” in *OsDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[2] D. Ongaro and J. Ousterhout, “In search of an understandable consensus algorithm,” in *2014 USENIX Annual Technical Conference (Usenix ATC 14)*, 2014, pp. 305–319.

[3] J. Wang, J. Liu, and N. Kato, “Networking and communications in autonomous driving: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2018.

[4] S. F. Chronicle. (2023) Driver hits woman in s.f., then cruise driverless car runs her over; photo shows victim trapped. [Online]. Available: https://www.sfchronicle.com/bayarea/article/woman-run-autonomous-vehicle-san-francisco-18403044.php

[5] I. Sarrigiannis, L. M. Contreras, K. Ramantas, A. Antonopoulos, and C. Verikoukis, “Fog-enabled scalable C-V2X architecture for distributed 5g and beyond applications,” *IEEE Network*, vol. 34, no. 5, pp. 120–126, 2020.

[6] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, and G. Fortino, “Agent-based Internet of Things: State-of-the-art and research challenges,” *Future Generation Computer Systems*, vol. 102, pp. 1038–1053, 2020.

[7] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, “Green industrial internet of things architecture: An energy-efficient perspective,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48–54, 2016.

[8] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, “Agent-oriented cooperative smart objects: From IoT system design to implementation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 11, pp. 1939–1956, 2017.

[9] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao, "Energy-efficient industrial Internet of Things: Overview and open issues," *IEEE Transactions on Industrial informatics*, vol. 17, no. 11, pp. 7225–7237, 2021.

[10] B. Balamurugan, T. Poongodi, M. Manu, S. Karthikeyan, and Y. Sharma, *Convergence of Blockchain, AI and IoT: A Digital Platform*. CRC Press, 2023.

[11] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2x access technologies: Regulation, research, and remaining challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1858–1877, 2018.

[12] Z. Li, H. Wu, Z. Dong, L. Zhang, and M. A. Imran, "Wireless intelligent distributed consensus enabled autonomous vehicles' cooperation at uncontrolled intersection," *IEEE Transactions on Communications, under review*, 2024.

[13] Y. Li, Y. Fan, L. Zhang, and J. Crowcroft, "Raft consensus reliability in wireless networks: Probabilistic analysis," *IEEE Internet of Things Journal*, 2023.

[14] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[15] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[16] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *Ieee Access*, vol. 7, pp. 22 328–22 370, 2019.

[17] L. Zhang, H. Xu, O. Onireti, M. A. Imran, and B. Cao, "How much communication resource is needed to run a wireless blockchain network?" *IEEE Network*, vol. 36, no. 1, pp. 128–135, 2021.

[18] L. Lamport, "The part-time parliament," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 277–317.

[19] B. M. Oki and B. H. Liskov, "Viewstamped replication: A new primary copy method to support highly-available distributed systems," in *Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*, 1988, pp. 8–17.

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[21] L. Lamport, "Paxos made simple," *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pp. 51–58, 2001.

[22] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, "Healthblock: A secure blockchain-based healthcare data management system," *Computer Networks*, vol. 200, p. 108500, 2021.

[23] C. Cachin *et al.*, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, no. 4. Chicago, IL, 2016, pp. 1–4.

[24] D. Tosh, S. Shetty, P. Foytik, C. Kamhoua, and L. Njilla, "Cloudpos: A proof-of-stake consensus design for blockchain integrated cloud," in *2018 IEEE 11Th international conference on cloud computing (CLOUD)*. IEEE, 2018, pp. 302–309.

[25] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Blockchain-based database to ensure data integrity in cloud computing environments," 2017.

[26] A. Ailijiang, A. Charapko, and M. Demirbas, "Consensus in the cloud: Paxos systems demystified," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2016, pp. 1–10.

[27] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 27, no. 2, pp. 228–234, 1980.

[28] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the attack surface of blockchain: A comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1977–2008, 2020.

[29] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, 2019, pp. 347–356.

[30] M. Baudet, A. Ching, A. Chursin, G. Danezis, F. Garillot, Z. Li, D. Malkhi, O. Naor, D. Perelman, and A. Sonnino, "State machine replication in the libra blockchain," *The Libra Assn., Tech. Rep*, 2019.

[31] L. Lamport and P. M. Melliar-Smith, "Synchronizing clocks in the presence of faults," *Journal of the ACM (JACM)*, vol. 32, no. 1, pp. 52–78, 1985.

[32] L. Lamport, "Generalized consensus and paxos," 2005.

[33] R. Van Renesse and D. Altinbuken, "Paxos made moderately complex," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, pp. 1–36, 2015.

[34] L. Lamport, "Fast paxos," *Distributed Computing*, vol. 19, pp. 79–103, 2006.

[35] L. Lamport and M. Massa, "Cheap paxos," in *International Conference on Dependable Systems and Networks, 2004*. IEEE, 2004, pp. 307–314.

[36] M. Burrows, "The chubby lock service for loosely-coupled distributed systems," in *Proceedings of the 7th symposium on Operating systems design and implementation*, 2006, pp. 335–350.

[37] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "Zookeeper: wait-free coordination for internet-scale systems." in *USENIX annual technical conference*, vol. 8, no. 9, 2010.

[38] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee, "Performance evaluation of the quorum blockchain platform," *arXiv preprint arXiv:1809.03421*, 2018.

[39] S. Pedersen, H. Meling, and L. Jehl, "An analysis of quorum-based abstractions: A case study using gorums to implement raft," in *Proceedings of the 2018 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating Algorithms for Distributed systems*, 2018, pp. 29–35.

[40] J. Polge, J. Robert, and Y. Le Traon, "Permissioned blockchain frameworks in the industry: A comparison," *Ict Express*, vol. 7, no. 2, pp. 229–233, 2021.

[41] H. Howard, M. Schwarzkopf, A. Madhavapeddy, and J. Crowcroft, "Raft refloated: Do we have consensus?" *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 12–21, 2015.

[42] E. Sakic and W. Kellerer, "Response time and availability study of raft consensus in distributed sdn control plane," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 304–318, 2017.

[43] B. R. Vojcic and R. L. Pickholtz, "Performance of direct sequence spread spectrum in a fading dispersive channel with jamming," *IEEE journal on selected areas in communications*, vol. 7, no. 4, pp. 561–568, 1989.

[44] K. C. Teh, A. C. Kot, and K. H. Li, "Multitone jamming rejection of ffh/bfsk spread-spectrum system over fading channels," *IEEE transactions on communications*, vol. 46, no. 8, pp. 1050–1057, 1998.

[45] H. Chen, "Applications of cyber-physical system: a literature review," *Journal of Industrial Integration and Management*, vol. 2, no. 03, p. 1750012, 2017.

[46] M. Sadiku, Y. Wang, S. Cui, and S. M. Musa, "Cyber-physical systems: A literature review," *European Scientific Journal*, vol. 13, no. 36, pp. 52–58, 2017.

[47] B. A. Yilma, H. Panetto, and Y. Naudet, "Systemic formalisation of cyber-physical-social system (cpss): A systematic literature review," *Computers in Industry*, vol. 129, p. 103458, 2021.

[48] S. A. Haque, S. M. Aziz, and M. Rahman, "Review of cyber-physical system in healthcare," *international journal of distributed sensor networks*, vol. 10, no. 4, p. 217415, 2014.

[49] L. M. Oliveira, R. Dias, C. M. Rebello, M. A. Martins, A. E. Rodrigues, A. M. Ribeiro, and I. B. Nogueira, "Artificial intelligence and cyber-physical systems: A review and perspectives for the future in the chemical industry," *AI*, vol. 2, no. 3, p. 27, 2021.

[50] L. Hu, N. Xie, Z. Kuang, and K. Zhao, "Review of cyber-physical system architecture," in *2012 IEEE 15th international symposium on object/component/service-oriented real-time distributed computing workshops*. IEEE, 2012, pp. 25–30.

[51] Z. Liu and J. Wang, “Human-cyber-physical systems: concepts, challenges, and research opportunities,” *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 11, pp. 1535–1553, 2020.

[52] C. V. Lozano and K. K. Vijayan, “Literature review on cyber physical systems design,” *Procedia manufacturing*, vol. 45, pp. 295–300, 2020.

[53] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, “Review on cyber-physical systems,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27–40, 2017.

[54] M. N. Al-Mhiqani, R. Ahmad, W. Yassin, A. Hassan, Z. Z. Abidin, N. S. Ali, and K. H. Abdulkareem, “Cyber-security incidents: a review cases in cyber-physical systems,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, 2018.

[55] P. Leitão, J. Queiroz, and L. Sakurada, “Collective intelligence in self-organized industrial cyber-physical systems,” *Electronics*, vol. 11, no. 19, p. 3213, 2022.

[56] L. A. Estrada-Jimenez, T. Pulikottil, S. Nikghadam-Hojjati, and J. Barata, “Self-organization in smart manufacturing—background, systematic review, challenges and outlook,” *Ieee Access*, vol. 11, pp. 10 107–10 136, 2023.

[57] T. Pulikottil, L. A. Estrada-Jimenez, H. Ur Rehman, F. Mo, S. Nikghadam-Hojjati, and J. Barata, “Agent-based manufacturing—review and expert evaluation,” *The International Journal of Advanced Manufacturing Technology*, vol. 127, no. 5, pp. 2151–2180, 2023.

[58] M. Hamzah, M. M. Islam, S. Hassan, M. N. Akhtar, M. J. Ferdous, M. B. Jasser, and A. W. Mohamed, “Distributed control of cyber physical system on various domains: a critical review,” *Systems*, vol. 11, no. 4, p. 208, 2023.

[59] G. Xie, K. Yang, C. Xu, R. Li, and S. Hu, “Digital twinning based adaptive development environment for automotive cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1387–1396, 2021.

[60] F. Tao, Q. Qi, L. Wang, and A. Nee, “Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison,” *Engineering*, vol. 5, no. 4, pp. 653–661, 2019.

[61] G. Lampropoulos and K. Siakas, "Enhancing and securing cyber-physical systems and industry 4.0 through digital twins: A critical review," *Journal of software: evolution and process*, vol. 35, no. 7, p. e2494, 2023.

[62] C. M. Richard, J. F. Morgan, L. P. Bacon, J. Graving, G. Divekar, M. G. Lichty *et al.*, "Multiple sources of safety information from v2v and v2i: Redundancy, decision making, and trust—safety message design report," United States. Federal Highway Administration. Office of Research . . . , Tech. Rep., 2015.

[63] M. R. Fyfe, "Safety evaluation of connected vehicle applications using micro-simulation," Ph.D. dissertation, University of British Columbia, 2016.

[64] F. Zhu, "Modeling, control, and impact analysis of the next generation transportation system," Ph.D. dissertation, Purdue University, 2016.

[65] UKCITE, "UK Connected Intelligent Transport Environment," https://www.ukcite.co.uk, 2016.

[66] I-MOTORS, "i-Motors Driving Innovation," http://www.i-motors.cloud/, 2016.

[67] G-ACTIVE, "Green Adaptive Control for Future Interconnected Vehicles," http://g-active.uk/.

[68] A. Stevens, M. Dianati, K. Katsaros *et al.*, "Cooperative automation through the cloud: The carma project," in *ITS European Congress*, 2017.

[69] M. Ma and Z. Li, "A time-independent trajectory optimization approach for connected and autonomous vehicles under reservation-based intersection control," *Transportation Research Interdisciplinary Perspectives*, vol. 9, p. 100312, 2021.

[70] F. Yan, M. Dridi, and A. El Moudni, "An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 183–200, 2013.

[71] W. ShangGuan, J. Yu, B. Cai, and J. Wang, "Research on unsigned intersection control method based on cooperative vehicle infrastructure system," in *2017 Chinese Automation Congress (CAC)*. IEEE, 2017, pp. 6436–6441.

[72] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," *Applied Intelligence*, vol. 37, pp. 207–222, 2012.

[73] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *2015 IEEE 18th international conference on intelligent transportation systems*. IEEE, 2015, pp. 343–347.

[74] K. Zhang, A. De La Fortelle, D. Zhang, and X. Wu, "Analysis and modeled design of one state-driven autonomous passing-through algorithm for driverless vehicles at intersections," in *2013 IEEE 16th International Conference on Computational Science and Engineering*. IEEE, 2013, pp. 751–757.

[75] K. Zhang, A. Yang, H. Su, A. de La Fortelle, K. Miao, and Y. Yao, "Service-oriented cooperation models and mechanisms for heterogeneous driverless vehicles at continuous static critical sections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1867–1881, 2016.

[76] E. Andert, M. Khayatian, and A. Shrivastava, "Crossroads: Time-sensitive autonomous intersection management technique," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 1–6.

[77] G. R. De Campos, P. Falcone, and J. Sjöberg, "Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1456–1461.

[78] G. Lu, L. Li, Y. Wang, R. Zhang, Z. Bao, and H. Chen, "A rule based control algorithm of connected vehicles in uncontrolled intersection," in *17th international IEEE conference on intelligent transportation systems (itsc)*. IEEE, 2014, pp. 115–120.

[79] F. Belkhouche, "Control of autonomous vehicles at an unsignalized intersection," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 1340–1345.

[80] F. Altché, X. Qian, and A. de La Fortelle, "Least restrictive and minimally deviating supervisor for safe semi-autonomous driving at an intersection: An miqp approach," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2520–2526.

[81] Y. Jiang, M. Zanon, R. Hult, and B. Houska, "Distributed algorithm for optimal vehicle coordination at traffic intersections," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 577–11 582, 2017.

[82] O. Onireti, L. Zhang, and M. A. Imran, "On the viable area of wireless practical byzantine fault tolerance (PBFT) blockchain networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[83] D. Yu, W. Li, H. Xu, and L. Zhang, "Low reliable and low latency communications for mission critical distributed industrial internet of things," *IEEE Communications Letters*, 2020.

[84] H. Luo, X. Yang, H. Yu, G. Sun, B. Lei, and M. Guizani, "Performance analysis and comparison of non-ideal wireless pbft and raft consensus networks in 6g communications," *IEEE Internet of Things Journal*, 2023.

[85] D. Yu and L. Zhang, "Centralized and distributed consensus in wireless network: An analytical comparison," in *2022 IEEE 20th International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE, 2022, pp. 81–89.

[86] D. Yu, Y. Sun, Y. Li, L. Zhang, and M. A. Imran, "Communication resource allocation of raft in wireless network," *IEEE Sensors Journal*, vol. 23, no. 17, pp. 19 398–19 408, 2023.

[87] H. Xu, L. Zhang, Y. Liu, and B. Cao, "Raft based wireless blockchain networks in the presence of malicious jamming," *IEEE Wireless Communication Letters*, 2020.

[88] D. Yu, H. Wu, Y. Sun, L. Zhang, and M. Imran, "Adaptive protocol of raft in wireless network," *Ad Hoc Networks*, vol. 154, p. 103377, 2024.

[89] Z. Xu, Y. Li, C. Feng, and L. Zhang, "Exact fault-tolerant consensus with voting validity," in *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2023, pp. 842–852.

[90] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.

[91] H. Lee, H. Seo, and W. Choi, "Fast and scalable distributed consensus over wireless large-scale internet of things network," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 7916–7930, 2021.

[92] H. Seo, J. Park, M. Bennis, and W. Choi, "Communication and consensus co-design for distributed, low-latency, and reliable wireless systems," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 129–143, 2020.

[93] C. Guyeux, M. Haddad, M. Hakem, and M. Lagacherie, "Efficient distributed average consensus in wireless sensor networks," *Computer Communications*, vol. 150, pp. 115–121, 2020.

[94] G. Scutari and S. Barbarossa, "Distributed consensus over wireless sensor networks affected by multipath fading," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 4100–4106, 2008.

[95] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (cacc) systems: Architectures, controls, and applications," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2884–2891.

[96] W. Ren, "Consensus based formation control strategies for multi-vehicle systems," in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.

[97] Y. Li, C. Tang, K. Li, X. He, S. Peeta, and Y. Wang, "Consensus-based cooperative control for multi-platoon under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2220–2229, 2018.

[98] M. Wegner, W. Xu, R. Kapitza, and L. Wolf, "Byzantine consensus in vehicle platooningvia inter-vehicle communication," 2016.

[99] S. Santini, A. Salvi, A. S. Valente, A. Pescapé, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1985–1999, 2016.

[100] M. Di Bernardo, A. Salvi, and S. Santini, "Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 102–112, 2014.

[101] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[102] W. Hu, Y. Hu, W. Yao, and H. Li, "A blockchain-based byzantine consensus algorithm for information authentication of the internet of vehicles," *IEEE Access*, vol. 7, pp. 139 703–139 711, 2019.

[103] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2018.

[104] H. Liu, C.-W. Lin, E. Kang, S. Shiraishi, and D. M. Blough, "A byzantine-tolerant distributed consensus algorithm for connected vehicles using proof-of-eligibility," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2019, pp. 225–234.

[105] X. Zhang, R. Li, and H. Zhao, "A parallel consensus mechanism using pbft based on dag-lattice structure in the internet of vehicles," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5418–5433, 2023.

[106] A. Kumar, L. Vishwakarma, and D. Das, "R-pbft: A secure and intelligent consensus algorithm for internet of vehicles," *Vehicular Communications*, vol. 41, p. 100609, 2023.

[107] H. Xu, Y. Fan, W. Li, and L. Zhang, "Wireless distributed consensus for connected autonomous systems," *IEEE Internet of Things Journal*, 2022.

[108] Z. Li, L. Zhang, X. Zhang, and M. Imran, "Design and implementation of a Raft based wireless consensus system for autonomous driving," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 3736–3741.

[109] M. G. Karlaftis and I. Golias, "Effects of road geometry and traffic volumes on rural roadway accident rates," *Accident Analysis & Prevention*, vol. 34, no. 3, pp. 357–365, 2002.

[110] (2023) United states department of transportation. about intersection safety. [Online]. Available: https://highways.dot.gov/safety/intersection-safety/about

[111] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3341–3356, 2008.

[112] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless internet of things: Performance analysis and optimal communication node deployment," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5791–5802, 2019.

[113] Y. Zou, M. Xu, J. Yu, F. Zhao, and X. Cheng, "A fast consensus for permissioned wireless blockchains," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 102–12 111, 2021.

[114] V. Poirot, B. Al Nahas, and O. Landsiedel, "Paxos made wireless: Consensus in the air." in *EWSN*, 2019, pp. 1–12.

[115] J. Zhu and I. Tasic, "Safety analysis of freeway on-ramp merging with the presence of autonomous vehicles," *Accident Analysis & Prevention*, vol. 152, p. 105966, 2021.

[116] M. S. Sheikh and Y. Peng, "Improved collision risk assessment for autonomous vehicles at on-ramp merging areas," *IEEE Access*, vol. 11, pp. 130 974–130 989, 2023.

[117] J. Rios-Torres and A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2016.

[118] V. Milanés, J. Godoy, J. Villagrá, and J. Pérez, "Automated on-ramp merging system for congested traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 500–508, 2010.

[119] Y. Wang, W. E, W. Tang, D. Tian, G. Lu, and G. Yu, "Automated on-ramp merging control algorithm based on internet-connected vehicles," *IET Intelligent Transport Systems*, vol. 7, no. 4, pp. 371–379, 2013.

[120] M. Athans, "A unified approach to the vehicle-merging problem," 1968.

[121] L. Li, F.-Y. Wang, and Y. Zhang, "Cooperative driving at lane closures," in *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007, pp. 1156–1161.

[122] W. Cao, M. Muka, T. Kawabe, H. Nishira, and N. Fujiki, "Merging trajectory generation for vehicle on a motor way using receding horizon control framework consideration of its applications," in *2014 IEEE Conference on Control Applications (CCA)*. IEEE, 2014, pp. 2127–2134.

[123] S. Jing, F. Hui, X. Zhao, J. Rios-Torres, and A. J. Khattak, "Cooperative game approach to optimal merging sequence and on-ramp merging control of connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4234–4244, 2019.

[124] T. Awal, L. Kulik, and K. Ramamohanrao, "Optimal traffic merging strategy for communication-and sensor-enabled vehicles," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1468–1474.

[125] W. Cao, M. Mukai, and T. Kawabe, "Two-dimensional merging path generation using model predictive control," *Artificial Life and Robotics*, vol. 17, pp. 350–356, 2013.

[126] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S.-i. Azuma, "Coordination of connected vehicles on merging roads using pseudo-perturbation-based broadcast control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3496–3512, 2018.

[127] T.-S. Dao, C. M. Clark, and J. P. Huissoon, "Distributed platoon assignment and lane selection for traffic flow optimization," in *2008 IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 739–744.

[128] X.-Y. Lu and J. K. Hedrick, "Longitudinal control algorithm for automated vehicle merging," *International Journal of Control*, vol. 76, no. 2, pp. 193–202, 2003.

[129] Q. Xu and R. Sengupta, "Simulation, analysis, and comparison of acc and cacc in highway merging control," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*. IEEE, 2003, pp. 237–242.

[130] M.-O. Hongler, O. Gallay, M. Hülsmann, P. Cordes, and R. Colmorn, "Centralized versus decentralized control—a solvable stylized model in transportation," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 19, pp. 4162–4171, 2010.

[131] M. Shahraeini, M. H. Javidi, and M. S. Ghazizadeh, "Comparison between communication infrastructures of centralized and decentralized wide area measurement systems," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 206–211, 2010.

[132] Y. Zhao, X. Wang, C. Wang, Y. Cong, and L. Shen, "Systemic design of distributed multi-uav cooperative decision-making for multi-target tracking," *Autonomous Agents and Multi-Agent Systems*, vol. 33, pp. 132–158, 2019.

[133] C. Feng, Z. Xu, X. Zhu, P. V. Klaine, and L. Zhang, "Wireless distributed consensus in vehicle to vehicle networks for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 8061–8073, 2023.

[134] S. Ahmad and A. H. Mir, "Scalability, consistency, reliability and security in sdn controllers: a survey of diverse sdn controllers," *Journal of Network and Systems Management*, vol. 29, pp. 1–59, 2021.

[135] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[136] J. L. Zambrano-Martinez, C. T. Calafate, D. Soler, L.-G. Lemus-Zúñiga, J.-C. Cano, P. Manzoni, and T. Gayraud, "A centralized route-management solution for autonomous vehicles in urban areas," *Electronics*, vol. 8, no. 7, p. 722, 2019.

[137] M. Z. Li, W. R. Tam, S. M. Prakash, J. F. Kennedy, M. S. Ryerson, D. Lee, and Y. V. Pant, "Design and implementation of a centralized system for autonomous unmanned aerial vehicle trajectory conflict resolution," in *NAECON 2018-IEEE National Aerospace and Electronics Conference*. IEEE, 2018, pp. 43–51.

[138] T. S. Lee, S. Ghosh, and A. Neerode, "Asynchronous, distributed, decision-making systems with semi-autonomous entities: A mathematical framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 1, pp. 229–239, 2000.

[139] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 514–529, 2018.

[140] R. Zhang, Y. Yu, M. El Chamie, B. Açikmese, and D. H. Ballard, "Decision-making policies for heterogeneous autonomous multi-agent systems with safety constraints." in *IJCAI*, 2016, pp. 546–553.

[141] R. Dabarera, T. L. Wickramarathne, K. Premaratne, and M. N. Murthi, "Achieving consensus under bounded confidence in multi-agent distributed decision-making," in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.

[142] X. Cheng and S. Maghsudi, "Distributed consensus algorithm for decision-making in multi-agent multi-armed bandit," *IEEE Transactions on Control of Network Systems*, 2024.

[143] S. Li, G. Oikonomou, T. Tryfonas, T. M. Chen, and L. Da Xu, "A distributed consensus algorithm for decision making in service-oriented internet of things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1461–1468, 2014.

[144] L. Yang, Y. Zou, M. Xu, Y. Xu, D. Yu, and X. Cheng, "Distributed consensus for blockchains in internet-of-things networks," *Tsinghua Science and Technology*, vol. 27, no. 5, pp. 817–831, 2022.

[145] P. C. Mignerey, L. E. Emokpae, J. A. Schindall, G. F. Edelmann, and S. Finette, "Experimental demonstration of an autonomous distributed-consensus network for underwater passive-acoustic detection," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 1155–1171, 2022.

[146] J. Choi, S. Li, X. Wang, and J. Ha, "A general distributed consensus algorithm for wireless sensor networks," in *2012 Wireless Advanced (WiAd)*. IEEE, 2012, pp. 16–21.

[147] J. He, L. Duan, F. Hou, P. Cheng, and J. Chen, "Multiperiod scheduling for wireless sensor networks: A distributed consensus approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1651–1663, 2015.

[148] J. Cao, S. Leng, L. Zhang, M. Imran, and H. Chai, "A v2v empowered consensus framework for cooperative autonomous driving," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 5729–5734.