Interpretable Multi-Agent Coordination Algorithms for Heterogeneous (Urban/Suburban/Rural) Autonomous Mobility Networks

Rivaldo Lumelino, Alexandr Voronovich[1]

CSC 36000: Modern Distributed Computing — Research Project Report
The City College of New York (CUNY)
Professor Bandyopadhyay
December 19, 2025.

Abstract

Multi-agent drone systems are increasingly deployed in real-world applications such as package delivery, traffic monitoring, disaster response, and autonomous navigation. In these scenarios, multiple drones must coordinate their movement and actions while operating under strict safety and reliability requirements. However, achieving reliable coordination remains challenging due to communication delays, packet loss, and inconsistent network connectivity, especially in large-scale distributed environments. Many existing approaches rely on black-box learning models that lack interpretability and often assume ideal communication conditions, limiting their applicability in real-world systems.

In this project, we present an interpretable distributed drone coordination framework designed to operate under realistic communication constraints. We explicitly model multiple network environments, including V2X, MQTT, urban, suburban, and rural settings, each characterized by different delay and packet loss profiles. Our system combines communication-aware coordination, sensor-based fallback behavior, and task-driven decision-making without centralized control. We evaluate our framework using a comprehensive set of metrics, including message delivery rate, average communication delay, collision count, sensor fallback usage, interpretability event frequency, and task completion. Our results show that although degraded network conditions significantly impact communication performance, local sensing and interpretable decision rules allow the system to remain safe, stable, and operational. We also provide visualizations of drone movement to support qualitative analysis. This work demonstrates that interpretable, rule-based coordination can serve as a strong and practical baseline for distributed multi-agent systems operating under unreliable communication conditions.

1. Introduction

Autonomous drone swarms have emerged as an important research area at the intersection of distributed computing, robotics, and artificial intelligence. Applications of such systems include package delivery in urban environments, aerial traffic monitoring, disaster response in hazardous zones, and large-scale environmental sensing. In these applications, multiple drones must coordinate their movement and actions while operating without centralized supervision.

A defining characteristic of these systems is their reliance on distributed decision-making. Each drone must act based on local information, partial observations, and messages received from other drones. This decentralized nature makes drone swarms inherently distributed systems, where communication delays, failures, and partial information are unavoidable.

One of the key challenges in real-world drone coordination is unreliable communication. Unlike controlled laboratory environments, real networks suffer from packet loss, variable delays, congestion, and interference. Urban environments introduce signal interference and congestion, while suburban and rural environments often experience long communication delays and limited coverage. Despite these realities, many existing multi-agent systems assume near-perfect communication, which limits their applicability outside simulation.

Another significant challenge is interpretability. Many modern approaches rely on reinforcement learning or deep neural networks to learn coordination policies. While these approaches can achieve strong performance, they produce black-box decision-making processes that are difficult to understand or debug. In safety-critical systems such as drone swarms, the inability to explain agent behavior is a serious limitation.

In this project, we address both challenges by designing an interpretable, rule-based, distributed drone coordination system that explicitly models unreliable communication. Rather than optimizing for maximum performance, our goal is to understand how communication quality affects coordination, safety, and task execution. By avoiding black-box learning and instead using explicit decision rules, we ensure that every agent action can be logged, inspected, and explained.

Our contributions establish a strong baseline for future work in distributed multi-agent coordination under realistic network conditions.

2. Literature Review

Research on multi-agent systems has explored a wide range of approaches to coordination, communication, and learning. One of the most prominent paradigms is Multi-Agent Reinforcement Learning (MARL), where agents learn coordination policies through interaction with the environment and other agents. MARL has demonstrated success in tasks such as cooperative navigation, resource allocation, and swarm control. However, MARL approaches often require extensive training, large datasets, and careful reward shaping. More importantly, the learned policies are typically difficult to interpret.

Recent work has also explored communication-based and language-grounded agents, where agents exchange messages to coordinate their behavior. These approaches study how communication protocols emerge and how message content influences coordination. While promising, many of these methods assume low-latency, reliable communication channels that do not reflect real-world network behavior.

From the perspective of distributed systems, classic research has studied coordination under asynchronous communication, message delays, and partial failures. Concepts such as eventual consistency, message ordering, and fault tolerance are well understood in theory. However, these ideas are often explored abstractly and are not commonly integrated into embodied multi-agent simulations such as drone swarms.

Compared to prior work, our project emphasizes realism and interpretability. We explicitly model network delay and packet loss, integrate local sensing as a fallback mechanism, and focus on rule-based decision-making that can be directly explained. This positions our work at the intersection of distributed computing and multi-agent robotics, with a focus on understanding system behavior rather than optimizing black-box performance.

3. System Model, Data, and Baselines

3.1. Simulation Environment

We simulate a two-dimensional continuous environment in which multiple drone agents move over discrete time steps. Each agent maintains a position and velocity and updates its behavior synchronously at each step. The environment supports varying swarm sizes, ranging from small groups of 8 drones to larger swarms of up to 100 drones, allowing us to evaluate scalability.

The environment is designed to be simple yet expressive, enabling us to isolate the effects of communication and coordination without introducing unnecessary complexity.

3.2. Communication Model

We simulate five different network environments that reflect realistic communication conditions:

- V2X: Near real-time communication with minimal packet loss
- MQTT: IoT-style communication with moderate delay and packet loss
- Urban: Low to moderate delay and packet loss
- Suburban: Moderate delay and packet loss
- Rural: High delay and significant packet loss

Messages are broadcast between agents and may be delayed, dropped, or discarded if they become stale. This explicit modeling of communication failures allows us to study how coordination degrades under increasingly adverse conditions.

3.3. Baseline Behavior

As a baseline, agents initially select random velocities and broadcast their positions to other agents. No learning, centralized control, or global planning is used. This baseline isolates the impact of communication quality and coordination logic, allowing us to directly attribute performance differences to network conditions and decision rules.

4. Proposed Method and Innovations

4.1. Interpretable Agent Decision Pipeline

Each agent follows a deterministic, rule-based decision pipeline. Decisions such as slowing down, avoiding nearby drones, or steering toward a task goal are logged as human-readable events. This allows us to inspect agent behavior at any time step and understand why specific actions were taken.

4.2. Communication-Aware Coordination

Agents maintain a short history of received messages and compute smoothed estimates of neighbor positions. This mitigates noise caused by delayed or out-of-order messages and results in more stable coordination behavior.

4.3. Sensor-Based Fallback Mechanism

When communication fails, agents rely on local sensors within a fixed radius to detect nearby drones. This fallback mechanism ensures collision avoidance even under severe communication degradation.

4.4. Task-Based Distributed Behavior

Agents perform pickup and drop-off tasks by navigating to predefined locations. Task selection is decentralized, and agents transition between task states without global coordination.

4.5. Interpretability Metrics

We track interpretability events, such as large velocity changes or sensor-based decisions, to quantify how explainable the system behavior is.

5. Experimental Results

In this section, we present a detailed evaluation of our distributed drone coordination system under multiple network environments. The goal of these experiments is to understand how different communication conditions affect coordination quality, safety, interpretability, and task performance.:

5.1. Experimental Setup

All experiments were conducted in a two-dimensional environment of size 50×50 units. We varied the number of agents between 8 and 100 to observe both small-scale and more crowded scenarios. Each simulation was run for 300 discrete time steps.

For each run, all agents started at random positions and followed the same decision logic. The only factor that changed between experiments was the communication network profile. To ensure fair comparison, we used the same random seed across network modes so that initial positions and random velocity proposals were consistent.

We evaluated five network environments:

- V2X
- MQTT
- Urban
- Suburban
- Rural

Each environment differs in packet loss probability and message delay distribution, directly affecting how quickly and reliably agents receive information about their neighbors.

5.2. Evaluation Metrics

We evaluated system performance using the following metrics, all of which are directly computed by the simulation:

- Message delivery rate:the fraction of successfully delivered messages over total messages sent.
- Average Delivery Delay: the average number of time steps between message send and delivery.
- Collision Count: the number of times two agents occupied the same position (after rounding).
- Sensor Fallback Usage: how often agents relied on local sensors instead of communication.
- Interpretability Event Count: the number of logged explanatory events, such as slow-down decisions, avoidance maneuvers, or large turns.
- Tasks Completed: the total number of pickup-and-drop-off tasks completed by all agents.

Together, these metrics capture network quality, safety, behavioral transparency, and system usefulness.

5.3. Communication Performance Across Networks

Communication quality varied significantly across network environments.

In the V2X setting, message delivery rates were consistently high (above 95), and average delivery delay was close to one time step. This allowed agents to maintain up-to-date knowledge of their neighbors, resulting in smooth coordination and predictable motion patterns.

In contrast, the Rural network exhibited severe degradation. Delivery rates dropped substantially, and average delays increased to several time steps. Many messages were discarded as stale before delivery. This created highly asynchronous information flow, where agents often acted on outdated or missing data.

The MQTT, Urban, and Suburban networks fell between these extremes. MQTT exhibited higher delay variability, while suburban and urban networks showed moderate packet loss with manageable delays.

These results confirm that communication conditions strongly influence how much global information agents can rely on during decision-making.

5.4. Safety and Collision Analysis

Despite large differences in communication quality, collision counts remained very low across all environments. In many runs, zero collisions were observed, even under poor network conditions.

This outcome highlights the importance of the sensor-based fallback mechanism. When agents failed to receive messages, they relied on local sensing to detect nearby drones and adjust their velocity accordingly. This behavior prevented unsafe proximity even when global coordination broke down.

In environments with reliable communication, sensor fallback was rarely triggered. In rural and MQTT environments, fallback usage increased significantly, demonstrating that agents adaptively switched strategies based on network conditions.

These results show that local autonomy is essential for safety in distributed systems with unreliable communication.

5.5. Task Completion Performance

Task-based experiments introduced pickup and drop-off objectives that required agents to navigate toward fixed locations while avoiding collisions and coordinating with others.

Under V2X and Urban networks, task completion rates were high. Agents quickly reached pickup locations, transitioned to drop-off states, and completed multiple tasks within the simulation window.

In Suburban and MQTT networks, task completion slowed but remained feasible. Delayed communication caused agents to take less optimal paths, but task objectives helped prevent excessive clustering and encouraged spatial exploration.

In Rural networks, task completion rates were lower but still non-zero. Even with minimal communication, agents were able to complete tasks using local sensing and goal-following rules. This demonstrates that the system remains functional even when communication is severely degraded.

5.6. Interpretability and Behavioral Transparency

One of the main goals of this project is interpretability. Across all experiments, agents generated large numbers of human-readable logs explaining their decisions.

Interpretability event counts increased in poorer network environments. This is expected, as agents more frequently slowed down, relied on sensors, or executed large turns due to missing information.

By examining these logs, we can directly explain why performance changes across networks. For example:

- Increased delay leads to outdated neighbor positions
- Outdated positions cause abrupt corrections
- Abrupt corrections trigger large-turn events

Such causal chains are clearly visible in the logs, which would not be possible in black-box learning-based systems.

5.7. Summary of Results

Overall, the experiments demonstrate that:

- Communication quality strongly affects coordination efficiency
- Safety can be preserved using local sensing even under severe network degradation
- Task-based objectives improve spatial diversity and system usefulness
- Interpretability provides clear explanations for performance differences

These findings validate our design choices and confirm that interpretable, rule-based coordination is a strong and realistic baseline for distributed multi-agent systems.

6. Discussion

Our experimental results reveal several important insights about distributed multi-agent coordination under unreliable communication. When communication is reliable, such as in the V2X environment, drones are able to coordinate smoothly and maintain structured movement patterns. Agents receive timely neighbor information, resulting in gradual steering, stable formations, and efficient task execution.

As communication quality degrades, coordination becomes less efficient. Delayed or missing messages lead to outdated neighbor positions, which causes agents to react more conservatively. However, despite this degradation in coordination efficiency, overall system safety is preserved. The sensor-based fallback mechanism plays a critical role in this outcome by allowing agents to detect nearby drones and adjust their behavior even when communication fails completely.

We also observe clustering behavior in several network settings. This occurs because agents apply cohesion rules that pull them toward the center of nearby neighbors. While clustering can be useful for maintaining group coherence, excessive clustering reduces spatial coverage and task efficiency. The introduction of task-based objectives helps mitigate this effect by encouraging agents to move toward pickup and drop-off locations, thereby breaking large clusters and improving spatial diversity.

Another important observation is the value of interpretability. Because agent decisions are rule-based and logged in human-readable form, we can directly trace the causes of degraded performance. For example, we can observe that increased message delay leads to more abrupt steering corrections, which in turn results in large-turn events. This level of transparency would be difficult to achieve using black-box learning models.

Overall, these results highlight the trade-offs between communication quality, coordination efficiency, and safety, and demonstrate the practical benefits of interpretable decision-making in distributed multi-agent systems.

The interpretability of the system allows us to directly trace the causes of degraded performance, highlighting the value of explainable decision-making in distributed multi-agent systems.

7. Conclusion and Future Work

In this project, we presented an interpretable distributed drone coordination system designed to operate under realistic and unreliable communication conditions. Our system combines communication-aware coordination, sensor-based fallback behavior, and decentralized task-based decision-making without relying on centralized control or black-box learning models.

Through extensive simulations across multiple network environments, we showed that while poor communication degrades coordination efficiency, the system remains safe and functional. Local sensing allows agents to avoid collisions, and task-driven behavior ensures that meaningful objectives can still be completed even under adverse network conditions. Interpretability further strengthens the system by making agent behavior transparent and explainable.

This work demonstrates that simple, rule-based coordination can serve as a strong baseline for studying distributed multi-agent systems under realistic assumptions. Rather than optimizing for maximum performance, our approach emphasizes robustness, safety, and explainability.

Future work includes integrating learning-based policies on top of the interpretable baseline, using more realistic physics-based simulators, supporting heterogeneous agents with different sensing and communication capabilities, and evaluating the system in real-world or hardware-in-the-loop deployments. These extensions would allow us to explore how learning and interpretability can be combined in safety-critical distributed systems.

References

Li, H. Language-grounded multi-agent reinforcement learning with human-interpretable communication. Technical report, Honda Research Institute USA, 2024.

Liu, W. Autonomous agents for collaborative tasks under information asymmetry. Vancouver, Canada, 2025. NeurIPS.