

Article

Assignment-Consistent Dynamic Multi-UAV Task Allocation: Communication-Efficient Dynamic Multi-Unmanned Aerial Vehicle Task Allocation Under Stale and Asymmetric Information

Chenxu Zhao, Meixuan Li * , Xianqiang Zhu and Cheng Zhu 

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; zhaochenxu@nudt.edu.cn (C.Z.); zhuxianqiang@nudt.edu.cn (X.Z.); zhucheng@nudt.edu.cn (C.Z.)

* Correspondence: li.meixuan1@nudt.edu.cn

Highlights

What are the main findings?

- The proposed framework demonstrates that distributed task allocation can be coordinated by testing whether stale information could alter an assignment, rather than by synchronizing every state change.
- The proposed assignment-consistency certificates use interval bids, winner and bundle stability checks, and recency-aware deconfliction to suppress assignment-irrelevant communication.

What are the implications of the main findings?

- The proposed framework reduces unnecessary communication by replacing broad state synchronization with assignment-relevant updates while maintaining competitive allocation quality in the simulated benchmark scenarios.
- The simulation results indicate that decision-triggered communication can improve the communication–allocation trade-off under bandwidth limits, delayed links, and packet loss without relying on broad full-neighbor resynchronization.

Abstract

Dynamic task allocation for unmanned aerial vehicles (UAVs) is often constrained by stale and asymmetric information. Variants of the Consensus-Based Bundle Algorithm (CBBA) can waste bandwidth by transmitting every state or bid update without distinguishing assignment-altering information from irrelevant perturbations. This paper proposes Assignment-Consistent Dynamic Multi-UAV Task Allocation (AC-DMTA), a decision-centric framework in which agents communicate only when local uncertainty may invalidate the current assignment. AC-DMTA represents stale records with uncertainty sets and converts point bids into interval bids, enabling task-winner and bundle-stability certificates to suppress assignment-irrelevant messages. These certificates provide local sufficient conditions for certifying the stability of task-winner and bundle decisions within the modeled uncertainty envelopes and candidate sets, allowing assignment-irrelevant communication to be skipped. Timestamp and recency records further resolve asynchronous conflicts without broad resynchronization. Across four benchmark environments and 20 paired seeds, AC-DMTA sends fewer bytes and directed messages than periodic CBBA, Event-Driven Consensus-Based Bundle Algorithm (ED-CBBA), Clustered CBBA, and Two-Level Clustered Consensus-Based Bundle Algorithm (TLC-CBBA), achieving byte reductions of 10.4–91.1% across baseline–scenario pairs while maintaining competitive reward and



Academic Editor: Constantinos Angelis

Received: 18 May 2026

Revised: 4 July 2026

Accepted: 6 July 2026

Published: 9 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

task-completion rates. Under the tested delay and packet-loss stress conditions, AC-DMTA also achieves the lowest number of bytes per completed task among the compared methods.

Keywords: multi-UAV task allocation; multi-robot task allocation; CBBA; event-triggered communication; assignment consistency; asynchronous deconfliction; graph-limited consensus

1. Introduction

A central challenge in dynamic task allocation for unmanned aerial vehicle (UAV) teams is maintaining consistent executable assignments among agents that operate with incomplete and temporally decoupled information. Each UAV decides which tasks to execute, in what order, and subject to capability, energy, and timing constraints. While the physical state of the team comprises positions, remaining energy, task status, link availability, and vehicle progress, these quantities matter for allocation only through the decisions they induce. A stale position that leaves every winner and insertion unchanged is inconsequential; conversely, even a small stale error at an allocation boundary can trigger duplicate assignments or missed deadlines. The problem is therefore not merely that of combinatorial optimization, but also of distributed decision-making under stale and asymmetric information.

Multi-Robot Task Allocation (MRTA) provides the standard language for this problem, including robot task cardinality, temporal extent, task coupling, and utility structure [1,2]. Recent UAV-oriented studies further emphasize dynamic task arrivals, heterogeneous capabilities, energy budgets, communication topology, and operational scenarios such as disaster response, delivery, surveillance, and persistent monitoring [3–7]. Market-based methods are especially useful in distributed settings because bids compress local utility into a form that can be compared and deconflicted [8–10]. Among these methods, the Consensus-Based Bundle Algorithm (CBBA) remains a central construction: agents build local bundles and then run consensus over task winners [11]. This construction has been extended to asynchronous communication, coupled constraints, local swaps, clustering, event-driven updates, and cooperative task structures [12–18].

Communication remains a limiting resource in many multi-UAV deployments. Repeated exchanges of broad state, bids, and bundle records consume bandwidth, delay convergence, and amplify instability under packet loss or intermittent links. Existing communication-efficient methods mitigate this burden by clustering agents, triggering on protocol events, optimizing under weak links, maintaining connectivity, or learning when to communicate [15,17,19–22]. Although these mechanisms are valuable, most remain state-centric or protocol-centric: they ask whether a state, bid, or protocol condition has changed, rather than whether that change can alter the assignment. These two questions are not equivalent.

This paper adopts a decision-centric hypothesis: communication should be triggered only when stale information can change the allocation decision. We operationalize this property through assignment consistency certificates. Assignment consistency means that stale information need not be updated when, under the maintained local decision model, it does not change task winners, bundles, or feasibility decisions. Each agent maintains local uncertainty sets over stale neighbor states, task statuses, and execution costs. Point bids are replaced with interval bids; winner, bundle, and feasibility stability are checked explicitly; and communication is requested only when a certificate cannot rule out an allocation change. When the certificate fails, communication is targeted to the agents and

state variables that most improve the assignment margin per byte. We refer to the resulting framework as Assignment-Consistent Dynamic Multi-UAV Task Allocation (AC-DMTA).

The contributions are as follows:

1. We formulate assignment-consistent communication as a decision-centric alternative to state-centric communication for dynamic multi-UAV task allocation, where communication is triggered when stale information may affect an allocation decision rather than by every state change.
2. We introduce an interval-bid certification mechanism that provides local sufficient conditions for task-winner and bundle stability under stale and asymmetric information, evaluated over modeled candidate sets and uncertainty envelopes, with a local-swap variant assessed in the ablation study.
3. We specify recency-aware asynchronous deconfliction using timestamp tables, recency matrices, version records, and deterministic identifier tie-breaking, thereby preserving auction consistency and preventing cyclic overwriting.

2. Related Work

We review MRTA foundations, CBBA-based allocation, communication-efficient variants, and dynamic MRTA under weak communication to clarify the assignment consistency gap addressed by AC-DMTA.

2.1. MRTA and Market-Based Allocation

MRTA provides the formal foundation for assigning tasks to autonomous teams. Classical studies defined the main dimensions of the problem, including robot task cardinality, instantaneous versus time-extended tasks, independent versus coupled task structures, and utility-based assignment models [1,2]. Assignment and auction theories further provide the underlying optimization framework: the Hungarian method gives a reference formulation for static one-to-one assignment, auction algorithms show how prices and bids can decentralize assignment, and early market-based robot coordination studies explain why compact bids are attractive for distributed teams [8–10,23]. Recent UAV-oriented reviews and surveys extend this foundation to aerial missions, where allocation must account for energy, payload, communication range, sensing footprint, path feasibility, deadlines, and operating domains such as surveillance, monitoring, delivery, and search-and-rescue [3–7,24]. Recent multi-UAV applications further show that practical allocation is often intertwined with persistent monitoring, cooperative target handling, heterogeneous resources, and learning-assisted task selection [25–31]. This body of work clarifies the problem addressed here. Dynamic UAV allocation is not only a combinatorial optimization problem over rewards and costs, but also an information-management problem, because the records used to compute bids and bundles may be stale, partial, or asymmetric.

2.2. CBBA and Communication-Efficient Variants

CBBA is especially relevant to distributed UAV allocation because it combines greedy bundle construction with consensus over task winners and bid values [11]. Its variants have improved asynchronous operation, coupled-task handling, and local task swaps while preserving the winner-table and bid-table structure that makes the algorithm suitable for decentralized execution [12–14]. Recent CBBA-related and CBBA-inspired methods further improve scalability through clustering, event-driven transmission, payload-aware reassignment, cluster bridges, and conflict-aware allocation [15–18,32,33]. Other recent UAV assignment studies use distributed optimization, greedy insertion, two-stage task decomposition, metaheuristic search, and dynamic particle-swarm or hierarchical planning to address larger and more realistic aerial workloads [34–38]. These methods re-

duce computation or communication through clustering, event triggers, or scheduling, but their communication rules are usually driven by protocol changes rather than by assignment stability.

2.3. Dynamic MRTA Under Uncertainty and Weak Communication

Recent MRTA studies increasingly consider online task arrivals, uncertain task outcomes, limited sensing, temporal constraints, and weak communication. In dynamic and uncertain environments, allocation must handle task completion uncertainty, incomplete information, reallocation after plan disturbances, probabilistic task visibility, and locally constrained welfare objectives [39–45]. UAV-specific work has also integrated task allocation with emergency rescue, mission replanning, simultaneous assignment and path planning, connectivity maintenance, range-aware planning, constrained communication, and hybrid dynamic scheduling [20–22,46–50]. Simulation and reinforcement learning studies further emphasize that weak communication, packet loss, changing topology, and event-triggered graph updates can substantially alter allocation quality and reproducibility [19,51]. These studies move beyond static full-information assignment and are closely related to the operating conditions considered here. However, their communication analysis is usually expressed through message counts, connectivity measures, learned triggers, or bandwidth budgets.

2.4. Decision Consistency and Research Gap

The closest conceptual idea to this work is decision consistency: agents do not always need identical information if their local records still induce the same decision. In decentralized belief-space planning and limited-communication multi-agent decision-making, recent studies have shown that communication can be triggered by action inconsistency or decision change rather than by every belief or state update [52–54]. However, transferring this idea to dynamic multi-UAV task allocation is nontrivial because the decision object is discrete, structured, and history-dependent. Unlike belief-space action selection, CBBA-type allocation maintains winner tables, bid tables, ordered bundles, task-status records, timestamps, path versions, and deterministic deconfliction rules [1,2,11,12]. A small stale error near a bid boundary can change the task winner, whereas a larger state update far from any assignment boundary may be allocation-irrelevant. The remaining gap is that existing methods rarely test whether a stale record is assignment-relevant for the current winner, bundle, or local neighborhood decision. AC-DMTA addresses this gap by constructing interval bids and assignment stability certificates that separate assignment-relevant stale information from assignment-irrelevant stale information.

3. Problem Formulation

This paper considers a distributed dynamic multi-UAV task allocation problem under stale and asymmetric information. A team of UAVs must allocate and execute dynamically arriving tasks while communicating over a bandwidth-limited and unreliable network. Each UAV makes allocation and communication decisions based on locally maintained records rather than on a synchronized system state. Consequently, different UAVs may hold different winner tables, bid tables, task status records, and bundle states at the same decision step.

Let $\mathcal{I} = \{1, \dots, N\}$ denote the UAV team and let $\mathcal{J}(t) \subseteq \mathcal{J}$ denote the active task set at decision step t . The state of UAV i is represented by

$$s_i(t) = (p_i(t), v_i, E_i(t), c_i, q_i(t)), \quad (1)$$

where $p_i(t)$ is position, v_i is nominal cruise speed, $E_i(t)$ is remaining energy, c_i is capability type, and $q_i(t)$ is execution state. Task j is described by

$$\tau_j(t) = (x_j(t), r_j, d_j, R_j, \tau_j^{\text{service}}, c_j, \sigma_j(t)), \quad (2)$$

where $x_j(t)$ is the task location, r_j is the release time, d_j is the deadline, R_j is the reward, τ_j^{service} is the service time, c_j is the required capability, and $\sigma_j(t)$ is the task status. The task set is dynamic: new tasks may arrive, completed tasks may be removed, and task statuses may change over time.

The communication network among UAVs is represented by a time-varying graph

$$G_t = (\mathcal{I}, \mathcal{E}_t), \quad (3)$$

where $(i, k) \in \mathcal{E}_t$ if UAVs i and k can exchange messages at step t . Links are affected by communication range, delay, and packet loss. A message can be delayed by δ decision steps and dropped with probability ρ .

3.1. Distributed Decisions and Local Information

At each decision step, UAV i maintains a local information state

$$\mathcal{L}_i(t) = (s_i(t), \widehat{\mathcal{R}}_i(t), \widehat{y}_i(t), \widehat{b}_i(t), \widehat{v}_i(t), P_i(t)), \quad (4)$$

where $\widehat{\mathcal{R}}_i(t)$ denotes the locally stored task and neighbor records, $\widehat{y}_i(t)$ is the locally maintained winner table, $\widehat{b}_i(t)$ is the bid table, $\widehat{v}_i(t)$ stores timestamps or version numbers, and $P_i(t)$ is the current ordered task bundle. Because communication is selective and may be delayed or lost, local information states are generally asymmetric:

$$\mathcal{L}_i(t) \neq \mathcal{L}_k(t), \quad i \neq k. \quad (5)$$

The newest source timestamp about UAV k available to UAV i is denoted by $\tau_{ik}(t)$, and the corresponding record age is

$$a_{ik}(t) = t - \tau_{ik}(t). \quad (6)$$

The local allocation decision of UAV i is an ordered executable bundle

$$P_i(t) = (j_{i,1}, j_{i,2}, \dots, j_{i,L_i}), \quad (7)$$

where L_i is the bundle length. Let $x_{ij}(t) \in \{0, 1\}$ indicate whether task j is assigned to UAV i at step t , and let $z_j(t) \in \{0, 1\}$ indicate whether task j is completed within its time window. In addition to bundle construction, UAV i also makes a communication decision. We define

$$\chi_{ik}(t) \in \{0, 1\} \quad (8)$$

as the indicator of whether UAV i sends an allocation-related message to UAV k at step t .

3.2. Distributed Mission Objective and Constraints

The distributed task allocation problem is to design local bundle-construction and communication-triggering policies,

$$P_i(t) = \pi_i(\mathcal{L}_i(t)), \quad \chi_{ik}(t) = \mu_{ik}(\mathcal{L}_i(t)), \quad (9)$$

that maximize mission utility while reducing unnecessary communication. Over a horizon H , the problem can be formulated as

$$\max_{\{\pi_i, \mu_{ik}\}} \mathbb{E} \left[\sum_{t=0}^H \left(\sum_{j \in \mathcal{J}(t)} R_j z_j(t) - \lambda_d \sum_{i \in \mathcal{I}} C_i^d(P_i(t)) - \lambda_e \sum_{i \in \mathcal{I}} C_i^e(P_i(t)) - \lambda_c \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{N}_i(t)} \chi_{ik}(t) B_{ik}(t) \right) \right], \quad (10)$$

where $C_i^d(P_i(t))$ and $C_i^e(P_i(t))$ are the travel and energy costs induced by the ordered bundle, $\mathcal{N}_i(t)$ is the communication neighborhood of UAV i , $B_{ik}(t)$ is the byte cost of the transmitted message from i to k , and λ_d , λ_e , and λ_c balance travel cost, energy cost, and communication cost. The expectation accounts for the key sources of uncertainty, including dynamic task arrivals, communication delay, packet loss, and stochastic status evolution.

The allocation variables are subject to the following constraints:

$$\sum_{i \in \mathcal{I}} x_{ij}(t) \leq 1, \quad \forall j \in \mathcal{J}(t), t, \quad (11)$$

$$x_{ij}(t) \leq \mathbf{1}\{c_i \succeq c_j\}, \quad \forall i, j, t, \quad (12)$$

$$T_{ij}(P_i(t)) \leq d_j + M(1 - x_{ij}(t)), \quad \forall i, j, t, \quad (13)$$

$$E_i(t) - C_i^e(P_i(t)) \geq E_i^{\text{res}}, \quad \forall i, t, \quad (14)$$

$$x_{ij}(t) = 1 \Rightarrow j \in P_i(t), \quad P_i(t) \text{ is order-feasible}, \quad \forall i, j, t, \quad (15)$$

$$\chi_{ik}(t) = 0, \quad \forall (i, k) \notin \mathcal{E}_t. \quad (16)$$

Constraint (11) avoids duplicate task winners, (12) enforces platform-task compatibility, (13) imposes deadline feasibility, (14) preserves a safety energy reserve, (15) links binary assignment variables to executable ordered bundles, and (16) restricts message transmission to available communication links.

This formulation defines the distributed decision problem considered here: each UAV must construct bundles and decide when to communicate using local records that may be stale or asymmetric. AC-DMTA addresses this problem by certifying whether the relevant local assignment decision remains stable before transmitting updates.

3.3. Local Uncertainty Sets

To evaluate whether stale records may change an allocation decision, each UAV does not need an exactly synchronized global state. Instead, it represents stale neighbor and task records using bounded local uncertainty envelopes. These envelopes expand with record age and provide the local uncertainty set on which the interval bids and assignment certificates are evaluated.

For receiver i and source UAV k , let $\hat{p}_{ik}(t)$, $\hat{E}_{ik}(t)$, and $\hat{q}_{ik}(t)$ denote the newest position, energy, and execution-state record about k available to i . The stale agent-state record is represented by

$$\mathcal{S}_{ik}(t) = \{(p, E, q) : \|p - \hat{p}_{ik}(t)\|_2 \leq \sigma_{ik}^p(a_{ik}), \\ E \in [\hat{E}_{ik}(t) - \sigma_{ik}^E(a_{ik}), \hat{E}_{ik}(t) + \sigma_{ik}^E(a_{ik})] \cap [0, E_k^{\text{max}}], \\ q \in \mathcal{Q}_{ik}(t)\}. \quad (17)$$

In the implementation, the position and energy envelopes grow with record age according to

$$\sigma_{ik}^p(a) = \sigma_{ik}^{p,0} + \bar{v}_k a, \quad \sigma_{ik}^E(a) = \sigma_{ik}^{E,0} + \bar{e}_k a.$$

Here, $\mathcal{Q}_{ik}(t)$ contains the execution states that remain consistent with the last received record and possible progress during age a_{ik} .

Similarly, for task j , agent i maintains a task uncertainty envelope

$$\mathcal{T}_{ij}(t) = \mathcal{X}_{ij}(t) \times \mathcal{R}_{ij}(t) \times \mathcal{D}_{ij}(t) \times \mathcal{Z}_{ij}(t), \quad (18)$$

where $\mathcal{X}_{ij}$ is the location set, $\mathcal{R}_{ij}$ is the reward interval, $\mathcal{D}_{ij}$ is the deadline interval, and $\mathcal{Z}_{ij} \subseteq \{\text{available, assigned, completed, expired}\}$ is the status ambiguity set. The local uncertainty product set used by agent i is then

$$\Omega_i(t) = \prod_{k \in \mathcal{I}} \mathcal{S}_{ik}(t) \times \prod_{j \in \mathcal{J}} \mathcal{T}_{ij}(t). \quad (19)$$

In the implementation, these envelopes are instantiated from bounded motion, energy-rate, and stale-status models.

4. Assignment Consistency Certification and Communication Triggering

Figure 1 illustrates the framework of AC-DMTA (Algorithm 1). At each decision step, an agent updates its uncertainty sets, computes interval bids, evaluates assignment certificates, and remains silent when all modeled assignment changes can be ruled out. If one or more certificates fail, the agent ranks missing records by their expected deficiency reduction per byte, requests a targeted update, fuses the returned records through recency-aware deconfliction, and repairs its local bundle. The central test is therefore applied before communication, whether stale information can change the current local winner or bundle decision. Under delayed or lossy links, unchanged failed certificates are retried only after a guard interval.

Algorithm 1 AC-DMTA execution at agent i during decision step t

- 1: Update local UAV state, observed task states, source timestamps Θ_i , and recency matrix Φ_i .
 - 2: Construct the local uncertainty product set $\Omega_i(t)$ from stale source ages, motion bounds, energy bounds, and task-status ambiguity sets.
 - 3: Repair the local bundle $P_i(t)$ after completed, expired, or infeasible tasks are removed.
 - 4: Enumerate feasible insertions and compute conservative interval bids $[b_{ij}(t), \bar{b}_{ij}(t)]$ for retained competitors and active tasks.
 - 5: Compute task-winner margins $m_{ij}^T(t)$, bundle margin $m_i^B(t)$, and, if enabled, local-swap margins $m_{ik}^S(t)$.
 - 6: Aggregate certificate violations into deficiency $D_i(t)$.
 - 7: **if** $D_i(t) = 0$ and all selected tasks remain locally feasible **then**
 - 8: Remain silent and execute the next feasible path action.
 - 9: **else**
 - 10: Rank candidate neighbors or records by the proxy message-value-per-byte score, $\widehat{\text{MPB}}_i(r)$.
 - 11: Transmit one targeted packet containing the highest-ranked state, task, winner/bid, path-version, timestamp, and recency records.
 - 12: Fuse returned records only if their precedence key is strictly newer or tie-broken deterministically.
 - 13: Release any bundle suffix invalidated by a fresher winner/path record, then rebuild the bundle and log the deconfliction case.
 - 14: **end if**
 - 15: Under delay or packet loss, retry unchanged failed certificates only after the guard interval.
-

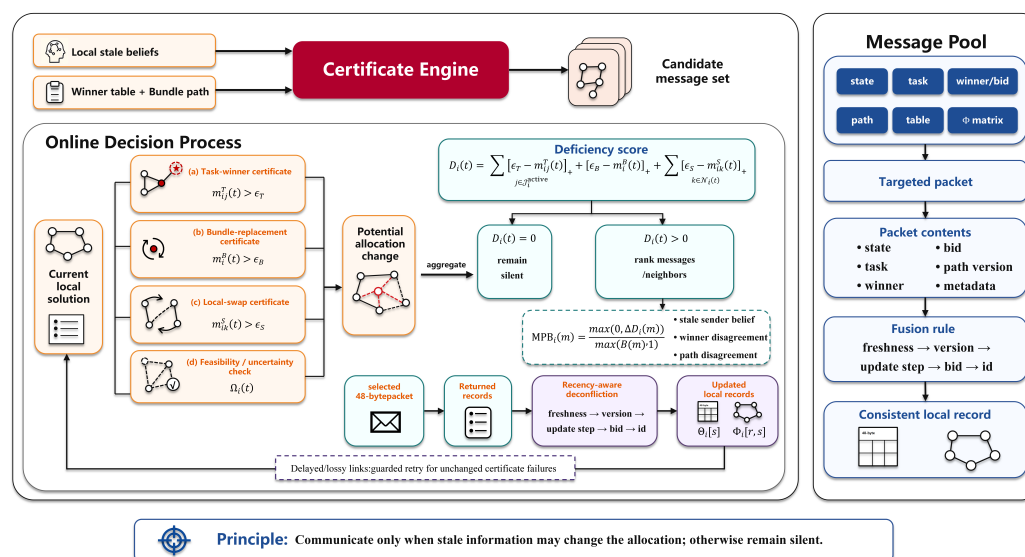


Figure 1. Overview of the AC-DMTA framework, including certificate evaluation, targeted message selection, recency-aware record fusion, and local bundle repair.

4.1. Bundle Utility and Interval Bids

For a candidate ordered path $P_i = [j_1, \dots, j_K]$, agent i evaluates

$$U_i(P_i; \omega) = \sum_{j \in P_i} (R_j(\omega) - \lambda_d d_{ij}(P_i; \omega) - \lambda_e e_{ij}(P_i; \omega)), \quad (20)$$

subject to deadline, capability, and reserve-energy feasibility. That is, an ordered path is scored by the total reward of its tasks minus the weighted travel and energy costs required to execute them in order. Let $P_i \oplus_\ell j$ denote the insertion of task j into position ℓ of path P_i , and let $\mathcal{L}_{ij}^{\text{cand}}(P_i)$ be the candidate insertion positions allowed by the bundle size limit. Feasibility is evaluated inside U_i for each realization; infeasible paths or insertions are assigned utility $-\infty$. The insertion gain under realization ω is

$$\Delta U_{ij\ell}(P_i; \omega) = U_i(P_i \oplus_\ell j; \omega) - U_i(P_i; \omega). \quad (21)$$

This insertion gain plays the role of the marginal utility bid in CBBA-based bundle construction: it measures how much the current path improves when task j is inserted at position ℓ . Using the insertion gain above, AC-DMTA replaces the point bid computed from locally stored stale records with a conservative interval bid evaluated over the local uncertainty envelope $\Omega_i(t)$. The lower endpoint gives the largest insertion gain that can be guaranteed over the admissible realizations, whereas the upper endpoint gives the largest insertion gain attainable within the same envelope:

$$\underline{b}_{ij}(t) = \max_{\ell \in \mathcal{L}_{ij}^{\text{cand}}(P_i)} \inf_{\omega \in \Omega_i(t)} \Delta U_{ij\ell}(P_i; \omega), \quad (22)$$

$$\bar{b}_{ij}(t) = \max_{\ell \in \mathcal{L}_{ij}^{\text{cand}}(P_i)} \sup_{\omega \in \Omega_i(t)} \Delta U_{ij\ell}(P_i; \omega). \quad (23)$$

If no candidate insertion is feasible under any admissible local realization, both endpoints are set to $-\infty$. In the implementation, the extrema are evaluated from the distance, deadline,

reward, task status, and energy envelopes maintained in $\Omega_i(t)$. The resulting interval encloses the bid under the realized state ω^* :

$$\underline{b}_{ij}(t) \leq b_{ij}(\omega^*, t) \leq \bar{b}_{ij}(t). \quad (24)$$

Consequently, any winner or bundle comparison that holds at the interval endpoints also holds for the realized but unobserved bid. These interval bids are then used to construct the assignment margins and communication certificates in the next subsection.

4.2. Assignment Margins and Certificates

The following checks evaluate whether the current local assignment decision remains stable under the maintained uncertainty envelope and the modeled decision neighborhood. When a certificate passes, the corresponding update can be skipped at the current decision step. These certificates are local sufficient conditions: they rule out assignment changes only within the modeled candidate sets, swap neighborhoods, uncertainty envelopes, and local path-version records of a single agent, and they guarantee neither global optimality of the team allocation nor global consistency across all local records. Residual false-safe events can therefore still occur when the destabilizing change lies outside this local model, for example, through an unmodeled local swap or an asynchronous path-version disagreement. To address this issue, Section 6.4 evaluates these events using a replay-guarded diagnostic that forces communication and suffix repair whenever fresh replay detects a false-safe event. This diagnostic estimates the counterfactual mission-level benefit of eliminating residual false-safe events. In the tested no-delay diagnostic instances, eliminating these events does not materially improve reward or task completion, while it requires additional communication and repair. These results indicate that certificate-based local assignment decisions remain practically viable under the quantified residual false-safe risk.

Let $w_i(j, t)$ be agent i 's current locally known winner of task j . Let $\mathcal{C}_i(j, t)$ be the finite competitor set used by i for the task check. In the implementation, $\mathcal{C}_i(j, t)$ contains i , current neighbor records, and retained stale records such as previous winners. Each retained competitor enters the comparison through the optimistic upper bid below. Define the task assignment margin as

$$m_{ij}^T(t) = \underline{b}_{w_i(j,t),j}(t) - \max_{k \in \mathcal{C}_i(j,t) \setminus \{w_i(j,t)\}} \bar{b}_{kj}(t). \quad (25)$$

In other words, the margin compares the guaranteed worst-case bid of the currently stored winner with the most optimistic bid of its strongest modeled competitor. The task winner certificate passes if

$$m_{ij}^T(t) > \epsilon_T, \quad (26)$$

where $\epsilon_T \geq 0$ is the certificate margin.

Proposition 1 (Local task winner stability certificate). *If all interval bids are sound over the local operating envelope $\Omega_i(t)$ and (26) holds, then no competitor in $\mathcal{C}_i(j, t)$ can outbid $w_i(j, t)$ for task j under any realization in $\Omega_i(t)$. Hence the task winner is stable within the modeled candidate set.*

Proof. For any competitor $k \neq w_i(j, t)$, it holds that

$$b_{kj}(\omega, t) \leq \bar{b}_{kj}(t), \quad b_{w_i(j,t),j}(\omega, t) \geq \underline{b}_{w_i(j,t),j}(t)$$

for all $\omega \in \Omega_i(t)$, given the soundness of the interval-bid bounds. Equation (26) implies $b_{w_i(j,t),j}(\omega, t) > b_{kj}(\omega, t) + \epsilon_T$, so k cannot replace the current winner. $\square$

When (26) holds, the corresponding task winner update can be skipped at the current decision step.

The task certificate protects individual winners, but it does not by itself protect the order and composition of a local bundle. Let $\mathcal{R}_i(P_i)$ be all one-task-out/one-task-in paths formed by removing one task from P_i and inserting one external candidate task. Define the bundle margin as

$$m_i^B(t) = \inf_{\omega \in \Omega_i(t)} U_i(P_i; \omega) - \max_{P' \in \mathcal{R}_i(P_i)} \sup_{\omega \in \Omega_i(t)} U_i(P'; \omega). \quad (27)$$

A positive bundle margin certifies that the current bundle, even in its worst admissible case, outperforms every modeled one-task replacement in its best admissible case. The bundle replacement certificate passes if

$$m_i^B(t) > \epsilon_B. \quad (28)$$

For neighboring agents i and k , let $\mathcal{S}_{ik}^{\text{swap}}(P_i, P_k)$ be the set of paths formed by exchanging at most one task between P_i and P_k , with the corresponding feasibility checks. Define the swap assignment margin as

$$m_{ik}^S(t) = \inf_{\omega \in \Omega_i(t)} [U_i(P_i; \omega) + U_k(P_k; \omega)] - \max_{(P'_i, P'_k) \in \mathcal{S}_{ik}^{\text{swap}}} \sup_{\omega \in \Omega_i(t)} [U_i(P'_i; \omega) + U_k(P'_k; \omega)]. \quad (29)$$

The swap margin applies the same worst-case versus best-case comparison to single-task exchanges between two neighboring bundles. The optional local-swap certificate passes if

$$m_{ik}^S(t) > \epsilon_S. \quad (30)$$

Proposition 2 (Local bundle and optional swap stability certificate). *Under sound utility bounds over the local operating envelope $\Omega_i(t)$, Equation (28) rules out all modeled one-task replacement improvements for agent i , and Equation (30) rules out all modeled one-task exchanges between neighboring agents i and k .*

Proof. The result follows by comparing a conservative lower bound for the current path or path pair with an optimistic upper bound for every modeled alternative. If the former exceeds the latter by a positive margin, no alternative in the search neighborhood can improve utility under any realization in the local uncertainty set. $\square$

When these margins are positive, the corresponding bundle- or swap-related update can be skipped at the current decision step.

4.3. Timestamp Tables, Recency Matrices, and Deconfliction

In asynchronous dynamic task allocation, assignment records may arrive out of order and may be generated from different local views of the team. A large bid, including one embedded in a winner record, is not necessarily reliable if it was computed from stale information. To avoid this failure mode, AC-DMTA separates the numerical assignment value from the freshness of the information on which the record is based.

Each agent maintains a source clock $c_i(t)$, a timestamp table $\Theta_i[s]$ that stores the newest source timestamp about source s , and a recency matrix $\Phi_i[r, s]$ that stores the freshness of reporter r 's knowledge about source s . Messages carry Θ , Φ , winner sources, winner

versions, update steps, bids, and path versions. When receiver i compares a local winner record a with a sender record a' , it uses the lexicographic precedence key

$$\kappa(a) = (F(a), V(a), U(a), B(a), -\text{id}(a)), \quad (31)$$

where F is source-level freshness derived from Θ and Φ , V is winner version, U is update step, B is bid, and $-\text{id}$ is deterministic tie-breaking. A received assignment record a' replaces a local record a only when $\kappa(a') \succ \kappa(a)$ under this total lexicographic order. If the keys are equal, the records are identical for deconfliction purposes; if the numerical bids disagree under identical freshness and version fields, the deterministic identifier breaks the tie. Case labels such as `self-vs-sender`, `other-vs-sender`, and `other-vs-other` are logged to identify the decisive conflict resolution criterion, namely record metadata, such as freshness or version, the update step, or the bid value itself.

Auction consistency is preserved by separating record adoption from bundle repair. The receiver first accepts an incoming winner, bid, path version, or task status record only when its precedence key in (31) is strictly higher than the stored key. A larger bid alone is never sufficient to overwrite a fresher winner record. If a fresher accepted record invalidates a task in UAV i 's bundle, UAV i releases that task and the dependent suffix, then rebuilds the suffix using the current winner table.

Proposition 3 (Recency-aware deconfliction prevents cyclic overwriting). *Assume that each source clock is monotonically increasing and that a receiver replaces a local assignment record only when the incoming precedence key is strictly larger, with deterministic tie-breaking for equal freshness and version fields. Then, for any fixed record revision, repeated asynchronous exchanges cannot create an infinite overwrite cycle. Moreover, if a fresher accepted record invalidates a task in the local bundle, suffix release and rebidding keep the retained bundle prefix consistent with the currently accepted winner records.*

Proof. For a fixed record revision, the precedence key in (31) induces a finite total order over assignment records. Each accepted replacement strictly increases the stored key, so a record cannot return to a previously overwritten state, and infinitely repeated overwriting is impossible. Equal-key cases are resolved deterministically and therefore cannot oscillate. If an accepted fresher record invalidates a bundle element, suffix release removes the affected task and its dependent downstream bids; rebidding then rebuilds the suffix using the updated winner table. Thus, the retained bundle prefix remains consistent with the accepted assignment records. $\square$

4.4. Communication Trigger and Message Value per Byte

The task and bundle certificates, with an optional local-swap term, are combined into a nonnegative local deficiency score:

$$D_i(t) = \sum_{j \in \mathcal{J}_i^{\text{active}}} [\epsilon_T - m_{ij}^T(t)]_+ + [\epsilon_B - m_i^B(t)]_+ + \lambda_S \sum_{k \in \mathcal{N}_i(t)} [\epsilon_S - m_{ik}^S(t)]_+, \quad (32)$$

where $[x]_+ = \max(x, 0)$ and $\lambda_S \in \{0, 1\}$ toggles the optional swap term. Each bracketed term measures how far a certificate falls short of its required margin, so $D_i(t)$ vanishes exactly when every enabled check passes. Agent i remains silent when $D_i(t) = 0$ and all selected tasks are locally feasible. If $D_i(t) > 0$ or feasibility cannot be certified, the agent schedules a targeted neighbor message.

For a candidate message m , the byte model is

$$B(m) = B_0 + b_s|S_m| + b_\tau|\mathcal{T}_m| + b_w|W_m| + b_p|P_m| + b_\Theta|\Theta_m| + b_\Phi|\Phi_m|, \quad (33)$$

where S_m is the set of state records, $\mathcal{T}_m$ is the set of task records, W_m is the set of winner/bid records, P_m is the set of path records, Θ_m is the timestamp table, and Φ_m is the recency matrix. The cost of a message therefore grows with the number and type of records it carries rather than being a fixed per-packet constant. The ideal message value per byte is defined as the expected reduction in certificate deficiency per transmitted byte:

$$\Delta D_i(m) = D_i(\Omega_i(t)) - \mathbb{E}_{\omega \sim q_i}[D_i(\Omega_i^m(t))], \quad (34)$$

$$\text{MPB}_i(m) = \frac{\max(0, \Delta D_i(m))}{\max(B(m), 1)}. \quad (35)$$

Here, q_i is the receiver's local predictive distribution over uncertain records and $\Omega_i^m(t)$ is the uncertainty set after incorporating message m . Under Equation (35), a message is considered valuable when it reduces certificate deficiency at low byte cost. In practice, AC-DMTA uses a deterministic surrogate to avoid online counterfactual sampling.

To make the online rule deterministic and reproducible, AC-DMTA replaces the intractable numerator $\Delta D_i(m)$ with an observable surrogate for each candidate sender r . Let

$$\widehat{\Delta D}_i(r) = a_{ir} + 2\chi_{ir}^P + 0.5n_{ir}^W, \quad \chi_{ir}^P = \mathbf{1}\{P_r \neq P_i^r\}, \quad (36)$$

where a_{ir} is the age of receiver i 's record about sender r , P_r is the path record carried by sender r , P_i^r is receiver i 's stored path copy for that sender, and n_{ir}^W counts the tasks for which the winner/bid record carried by r disagrees with i 's local winner/bid record. The three terms approximate three mechanisms by which a message can reduce the ideal deficiency in (34): older records are more likely to shrink the uncertainty envelope, a path version disagreement may alter bundle feasibility or suffix costs, and winner/bid disagreements directly affect task-level certificate margins.

The proxy message value per byte is then computed as

$$\widehat{\text{MPB}}_i(r) = \frac{\widehat{\Delta D}_i(r)}{\max(B_i(r), 1)}, \quad (37)$$

where $B_i(r)$ is the byte cost of the candidate update from sender r under the same accounting model as (33). The constants in (36) are fixed priority weights: the path version disagreement term is emphasized because it can invalidate a bundle suffix, while winner disagreements are counted per affected task. The proxy in (37) ranks requested neighbors or records after communication has already been triggered by a certificate deficiency or feasibility failure. Each failed certificate triggers at most one highest-ranked request before the next retry interval, and guarded retry avoids retransmitting unchanged failures at every decision step.

5. Experimental Setup

5.1. Benchmark Scenarios

Experiments are conducted on four environments drawn from three scenario families: synthetic boundary stress, disaster response, and urban delivery. The first two scenarios are controlled synthetic configurations, whereas the latter two represent application-oriented mission families:

- Near-boundary, where small changes in stale information can flip task winners;
- Large-margin, where the same communication perturbations rarely change the allocation;
- Disaster response, with clustered incidents, urgent tasks, and tighter deadlines;
- Urban delivery, with multi-hub cross-zone routes and denser background traffic.

The experiments are conducted with $N = 6$ UAVs, $M = 18$ tasks, 40 decision steps, bundle size 4, communication range 35, certificate margin 1.0, and interval-penalty scale 0.5. The final AC-DMTA configuration uses task winner and bundle replacement certificates with freshness-aware deconfliction, corresponding to $\lambda_S = 0$ in (32). The swap-enabled variant sets $\lambda_S = 1$ for the ablation study. The main benchmark, delay/loss stress test, and ablation study use 20 fixed seeds; the scaling stress uses 10 fixed seeds and extends to a maximum 80-UAV/200-task case. Parameter-tuning sweeps use a fixed five-seed sensitivity set.

5.2. Simulator and Multi-UAV Environment Emulation

The paired-seed benchmark runs are conducted in a deterministic, time-stepped Python 3.12.3 numerical simulator. The simulator proceeds in fixed decision epochs, with discrete-event updates applied at the boundaries of these epochs. This design makes the allocation, communication, and certificate-evaluation traces reproducible from the scenario file and random seed while ensuring that AC-DMTA and all baselines are evaluated under the same execution model.

At the beginning of each run, the simulator initializes UAV start positions, task locations, task rewards, deadlines, service requirements, communication range, delay/loss parameters, and random-number streams from a fixed seed. For a paired seed comparison, all algorithms receive the same initialized scenario instance and are executed on the same simulator loop. The only difference across algorithms lies in the allocation and communication policy used during the auction/update phase. Therefore, differences in reward, completion, bytes, and message counts come from the algorithms rather than from different task instances or simulator settings.

The simulator uses a numerical kinematic abstraction for UAV motion. Each UAV travels between task locations at a fixed nominal speed, consumes energy proportional to path length and service cost, and can execute only tasks that satisfy capability, deadline, energy reserve, and path order constraints. This abstraction is consistent with the assignment layer focus of the compared methods, which operate on task allocation and communication rather than continuous control. The distributed multi-UAV information structure is simulated by maintaining a separate local record for each UAV: UAV i knows its own position, energy, current bundle, and generated bids exactly, but stores only timestamped or versioned records for other UAVs, task states, winners, bids, and paths. These records become stale when neighboring UAVs are out of range, when messages are delayed, or when packets are dropped. Consequently, two UAVs can hold different winner tables, bid tables, task status records, and path versions at the same decision epoch. This local record mechanism is the source of the stale and asymmetric information studied in the paper.

Communication is simulated at the record level. When a method triggers communication, the simulator creates one or more directed messages containing the corresponding state, task, winner/bid, path, timestamp, or recency records. The byte cost of each message is computed using the same byte-accounting model for AC-DMTA and all baselines. Periodic CBBA broadcasts full-neighbor synchronization messages at its fixed interval, Event-Driven Consensus-Based Bundle Algorithm (ED-CBBA) transmits when its event condition is satisfied, clustered baselines communicate through their cluster structure, and AC-DMTA sends targeted records only after a local certificate indicates that a specific as-

signment decision may be affected by stale information. Messages can be queued, delayed, delivered, or dropped according to the same link schedule for all methods.

For reproducibility, benchmark runs use fixed paired seeds, and per-seed outputs record the scenario name, seed, algorithm configuration, completed tasks, reward, bytes, directed messages, send events, certificate counts, and diagnostic counters.

5.3. Baselines and Tuning

We compare AC-DMTA against periodic CBBA, ED-CBBA, Clustered CBBA, and Two-Level Clustered Consensus-Based Bundle Algorithm (TLC-CBBA) as the primary baselines. Table 1 lists the trigger variables and fixed settings used for the main comparison. ED-CBBA follows the winner/bid-list event condition reported in [17]; TLC-CBBA uses a two-level clustered bridge that periodically propagates task-list information through cluster heads.

Table 1. Main baseline families, sweep ranges, and selected settings used in the communication-intensity benchmark.

Family	Trigger Variable	Sweep/Main Setting
Periodic CBBA	full-neighbor synchronization	interval $\in \{1, 2, 4, 8\}$; main = 1
ED-CBBA	winner/bid change	prefix $\in \{1, 2, 3, 4\}$; main = 2
Clustered CBBA	cluster exchange	clusters $\in \{2, 3, 4\}$; main = 2
TLC-CBBA	two-level cluster bridge	bridge period main = 4
AC-DMTA	certificate deficiency	$\epsilon_T = \epsilon_B \in \{0.5, 1, 2, 3\}$; main = 1; message cap = 1; retry interval = 3

Each baseline was tuned within the same five-seed sensitivity budget and then frozen before the 20 paired-seed benchmark. The selection criterion was reward-first: for each baseline family, the sweep setting with the highest mean reward over the five tuning seeds was selected, and mean bytes were used only to break ties among settings whose mean rewards differed by less than one 95% CI half-width. Under this rule, no selected baseline setting sacrificed reward for communication savings, so the comparison does not depend on an explicit weighting between reward and bytes. Periodic CBBA with interval 1 is intentionally retained as a strong full-synchronization reference. AC-DMTA uses the same simulator calls, candidate-task limits, bundle size, execution model, and byte-cost accounting as the baselines.

5.4. Metrics and Statistical Treatment

We track nine metrics: completed tasks, reward, bytes, directed messages, send events, bytes per completed task, certificate statistics, deconfliction cases, and runtime. The primary communication measures are bytes and send events. The main benchmark reports means and 95% confidence-interval half-widths over 20 paired seeds. Reward, bytes, messages, and send events are tested with paired seed-wise *t*-tests and Wilcoxon signed-rank tests.

6. Results

The experiments evaluate five aspects: communication efficiency, case-level allocation behavior, robustness under delay and packet loss, certificate diagnostics, and scalability.

6.1. Experiment 1: Main Communication Benchmark

To evaluate the effectiveness of the proposed method in improving communication efficiency, we run AC-DMTA and the CBBA-style baselines in four environments (near-boundary, large-margin, disaster response, and urban delivery) using the same task in-

stances and simulator loop for all methods. Figure 2 summarizes completed-task counts and bytes per completed task, and Table 2 reports the corresponding numerical statistics.

The results show that AC-DMTA achieves consistent communication savings while preserving competitive allocation quality. First, AC-DMTA sends the fewest bytes and directed messages in every scenario. Relative to periodic CBBA, byte reductions are 84.5%, 77.3%, 91.1%, and 86.9% in the near-boundary, large-margin, disaster-response, and urban-delivery scenarios, respectively. Second, the savings persist against the communication-efficient baselines, with byte reductions across all baseline–scenario pairs ranging from 10.4% to 91.1%. Third, the communication savings coexist with competitive allocation quality. AC-DMTA achieves the best or tied-best task completion in the near-boundary, disaster-response, and urban-delivery scenarios; in the large-margin scenario, periodic CBBA completes 0.15 more tasks on average but sends 4.4 times more bytes.

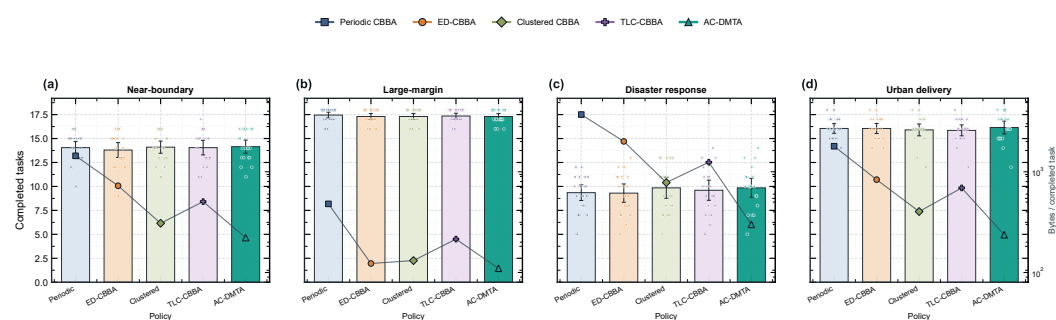


Figure 2. Main benchmark results over 20 paired seeds across four environments, reporting completed tasks and bytes per completed task with 95% confidence intervals. The colored dots are the individual per-seed completed-task values behind each bar, colored by policy as in the legend.

Table 2. Benchmark results across the four environments. Entries report mean $\pm$ 95% confidence interval (CI) half-width over 20 paired seeds. B/task denotes bytes per completed task.

Scenario	Method	Completed	Reward	Bytes	Messages	Send Events	B/Task
Near-boundary	Periodic CBBA	14.1 $\pm$ 0.6	1401.1 $\pm$ 66.4	20,030.4 $\pm$ 2273.3	417.3 $\pm$ 47.4	240.0 $\pm$ 0.0	1440.9 $\pm$ 184.8
	ED-CBBA	13.8 $\pm$ 0.8	1379.1 $\pm$ 81.1	10,058.4 $\pm$ 1711.8	209.6 $\pm$ 35.7	95.0 $\pm$ 6.3	726.7 $\pm$ 115.2
	Clustered CBBA	14.1 $\pm$ 0.6	1409.0 $\pm$ 66.3	4365.6 $\pm$ 892.0	91.0 $\pm$ 18.6	65.5 $\pm$ 9.4	308.3 $\pm$ 60.5
	TLC-CBBA	14.1 $\pm$ 0.8	1410.0 $\pm$ 81.8	7065.6 $\pm$ 964.6	147.2 $\pm$ 20.1	96.2 $\pm$ 9.0	505.5 $\pm$ 67.8
	AC-DMTA	14.2 $\pm$ 0.7	1410.7 $\pm$ 71.7	3108.0 $\pm$ 287.5	64.8 $\pm$ 6.0	64.8 $\pm$ 6.0	221.6 $\pm$ 21.5
Large-margin	Periodic CBBA	17.4 $\pm$ 0.3	1731.4 $\pm$ 31.1	8347.2 $\pm$ 460.6	173.9 $\pm$ 9.6	240.0 $\pm$ 0.0	480.1 $\pm$ 32.0
	ED-CBBA	17.3 $\pm$ 0.3	1716.7 $\pm$ 32.7	2109.6 $\pm$ 237.5	44.0 $\pm$ 4.9	55.2 $\pm$ 3.7	122.8 $\pm$ 15.5
	Clustered CBBA	17.3 $\pm$ 0.3	1716.7 $\pm$ 32.7	2256.0 $\pm$ 246.9	47.0 $\pm$ 5.1	44.2 $\pm$ 3.5	131.3 $\pm$ 16.1
	TLC-CBBA	17.4 $\pm$ 0.3	1720.9 $\pm$ 31.8	3708.0 $\pm$ 243.5	77.2 $\pm$ 5.1	73.1 $\pm$ 3.0	214.5 $\pm$ 16.2
	AC-DMTA	17.3 $\pm$ 0.3	1717.8 $\pm$ 32.5	1891.2 $\pm$ 263.0	39.4 $\pm$ 5.5	39.4 $\pm$ 5.5	109.8 $\pm$ 15.9
Disaster response	Periodic CBBA	9.3 $\pm$ 0.8	1086.6 $\pm$ 96.0	33,105.6 $\pm$ 2182.7	689.7 $\pm$ 45.5	240.0 $\pm$ 0.0	3704.1 $\pm$ 457.5
	ED-CBBA	9.3 $\pm$ 1.0	1076.8 $\pm$ 113.4	17,954.4 $\pm$ 1789.1	374.1 $\pm$ 37.3	112.2 $\pm$ 8.6	2004.4 $\pm$ 224.9
	Clustered CBBA	9.8 $\pm$ 1.1	1147.8 $\pm$ 130.3	7468.8 $\pm$ 589.7	155.6 $\pm$ 12.3	87.8 $\pm$ 6.0	781.8 $\pm$ 59.7
	TLC-CBBA	9.6 $\pm$ 1.0	1119.4 $\pm$ 127.9	11,426.4 $\pm$ 729.6	238.1 $\pm$ 15.2	121.5 $\pm$ 6.1	1244.1 $\pm$ 117.5
	AC-DMTA	9.8 $\pm$ 1.0	1145.8 $\pm$ 118.6	2961.6 $\pm$ 410.1	61.7 $\pm$ 8.5	61.7 $\pm$ 8.5	299.9 $\pm$ 25.0
Urban delivery	Periodic CBBA	16.1 $\pm$ 0.5	1675.1 $\pm$ 53.2	28,761.6 $\pm$ 1877.5	599.2 $\pm$ 39.1	240.0 $\pm$ 0.0	1794.2 $\pm$ 109.7
	ED-CBBA	16.1 $\pm$ 0.5	1673.4 $\pm$ 57.6	13,327.2 $\pm$ 1005.5	277.6 $\pm$ 20.9	111.0 $\pm$ 3.9	836.0 $\pm$ 73.5
	Clustered CBBA	15.9 $\pm$ 0.6	1658.4 $\pm$ 67.1	6345.6 $\pm$ 442.3	132.2 $\pm$ 9.2	84.7 $\pm$ 5.5	403.1 $\pm$ 35.6
	TLC-CBBA	15.8 $\pm$ 0.6	1655.5 $\pm$ 65.7	10,886.4 $\pm$ 577.6	226.8 $\pm$ 12.0	125.8 $\pm$ 5.1	692.0 $\pm$ 45.9
	AC-DMTA	16.1 $\pm$ 0.7	1687.4 $\pm$ 74.5	3780.0 $\pm$ 257.5	78.8 $\pm$ 5.4	78.8 $\pm$ 5.4	236.6 $\pm$ 20.4

The scenario-wise pattern is consistent with the intended role of the assignment-consistency certificates. In near-boundary cases, stale records are more likely to affect winner margins, so AC-DMTA retains communication around uncertain assignment boundaries. In large-margin cases, many state changes remain assignment-irrelevant, allowing more updates to be suppressed. The disaster-response and urban-delivery results further suggest that this reduction is not limited to synthetic margin-controlled settings, but can

also appear in application-oriented mission scenarios with tighter deadlines, clustered incidents, cross-zone routes, and denser communication demands.

6.2. Experiment 2: Task Scene Allocation

Experiment 2 complements the aggregate benchmark by visualizing the main communication efficiency claim in a concrete urban-delivery scene. This visualization examines whether AC-DMTA reduces communication cost while maintaining task allocation performance, rather than reducing communication by executing fewer tasks. All methods are evaluated on the same task instance, so differences in completed tasks, route structure, and transmitted bytes primarily reflect their allocation and communication mechanisms rather than scenario variation. Figure 3 shows UAV start positions, completed and unfinished tasks, execution paths, and communication cost. In this scene, AC-DMTA completes all 18 tasks with substantially fewer transmitted bytes than the baselines, illustrating that its communication reduction can be achieved while maintaining effective task-scene allocation.

Figure 4 provides a qualitative Robot Operating System (ROS) visualization of one AC-DMTA urban-delivery trace. The visualization illustrates how an AC-DMTA allocation unfolds in a three-dimensional urban scene: the scene view displays buildings, delivery points, UAV positions, and route traces, while the ROS view and post-run log show the corresponding allocation process and communication events. The reported 18/18 tasks, 7920 bytes, 165 send events, and 165 directed messages pertain only to this single displayed run and are not part of the quantitative benchmark evaluation.

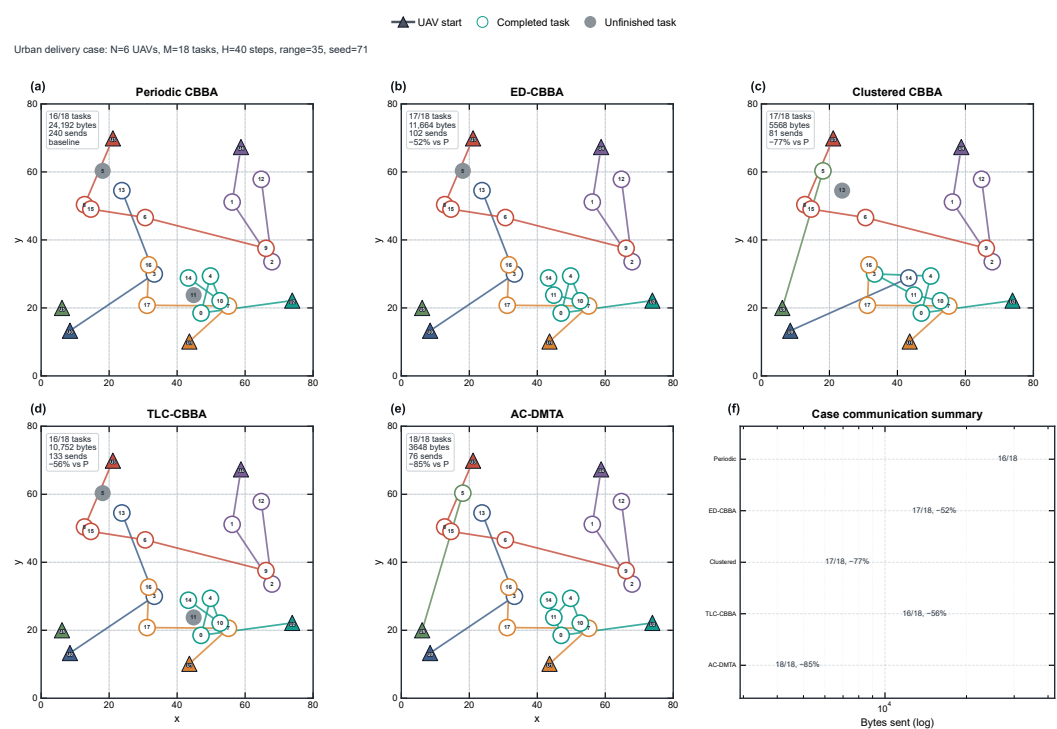


Figure 3. Urban-delivery allocation case under one paired seed. Panels (a–e) show method-specific task allocations and paths; panel (f) summarizes communication cost. Colors distinguish the individual UAVs: each UAV's start marker, route, and completed tasks are drawn in that UAV's color, while unfinished tasks are shown in gray as indicated in the legend.

6.3. Experiment 3: Delay and Packet Loss

This experiment evaluates the communication trigger under delayed and dropped messages. We use the disaster-response environment, sweep communication delay and packet-loss probability over 20 paired seeds. Figures 5 and 6 report reward behavior, certificate behavior, and completion robustness across the delay-loss settings. In the

delay sweep, AC-DMTA's mean reward varies only slightly as the delay increases from 0 to 4 steps, changing from 1145.8 to 1147.3. The diagnostics show that certificate pass rates decline and false-safe rates increase as uncertainty grows, indicating that the trigger responds to assignment uncertainty under stressed communication.

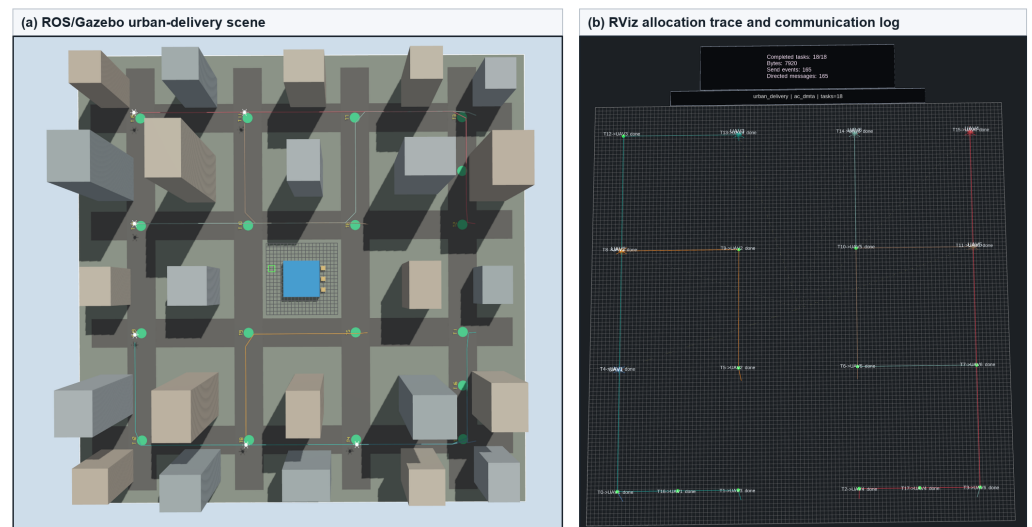


Figure 4. Qualitative ROS visualization of one AC-DMTA urban-delivery run. (a) Simulation scene; (b) ROS allocation trace and communication log. In both panels, the differently colored lines are the execution routes of the individual UAVs, and the green circles mark the delivery task locations.

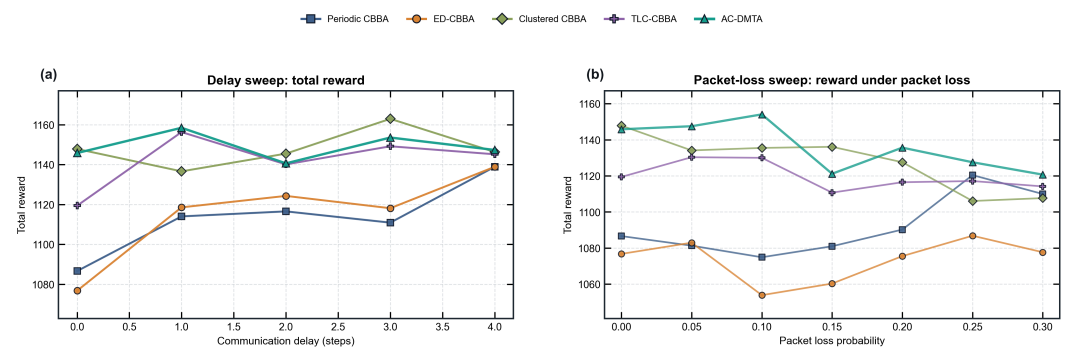


Figure 5. Delay and packet-loss stress test over 20 paired seeds. (a) Delay sweep for total reward; (b) packet-loss sweep for total reward.

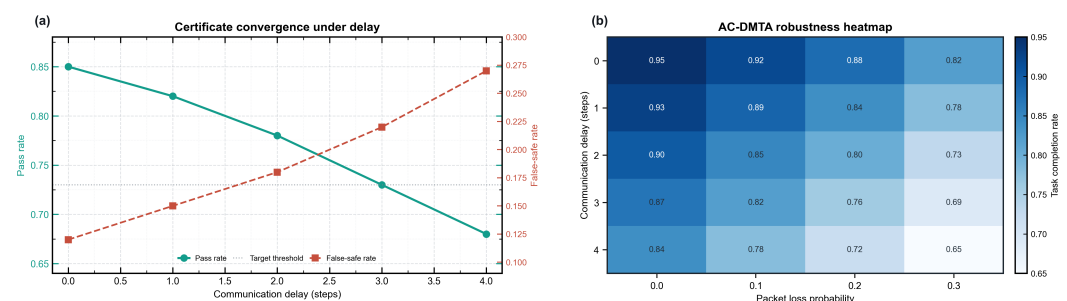


Figure 6. Certificate behavior and task completion robustness under delay and packet loss. (a) Certificate pass and false-safe rates under increasing delay; (b) completion rate heatmap across delay–loss settings.

The delay–loss results in Figure 6 show how certificate pass rates and false-safe rates vary under communication stress. The next experiment isolates the certificate stack and analyzes residual false-safe events by false-safe source and mission-level consequence.

6.4. Experiment 4: Certificate Ablation and Residual False-Safe Analysis

The ablation evaluates how each certificate component affects certificate passes, false-safe events, reward, and bytes. A false-safe event is recorded if a certificate passes but a fresh-information replay changes the corresponding winner or bundle. The false-safe rate is normalized by the number of certificate passes; therefore, it measures how often passed certificates become false-safe under replay. In the 20-seed no-delay diagnostic replay, adding the bundle certificate increases the aggregate certificate pass rate from 0.361 to 0.727 and reduces the false-safe rate from 0.322 to 0.273 compared with the task-only variant. Enabling the swap certificate further reduces the false-safe rate to 0.241, but it also introduces additional communication. The selected full configuration, which combines task and bundle certificates with recency-aware deconfliction, obtains a 0.730 certificate pass rate and a 0.279 false-safe rate with lower communication than the swap-enabled variant.

To interpret residual false-safe events, Table 3 reports their sources and mission-level consequences in the selected full configuration. We further compare this configuration with a replay-guarded diagnostic variant (ReplayGuard) using the same seeds, task instances, link schedules, and algorithmic settings. The replay-guarded variant forces communication and suffix repair whenever fresh replay detects a false-safe event, thereby providing a paired diagnostic for estimating how residual false-safe events affect reward and task completion.

Table 3. Diagnostic interpretation of residual false-safe events in the selected full configuration.

False-Safe Source	Event Share (%)	Winner Flip (%)	Bundle Change (%)	Consequential Runs (%)	Δ Reward, Full — ReplayGuard	Δ Completed, Full — ReplayGuard
Unmodeled local swap	99.2	80.5	100.0	37.5	6.3	0.0
Path-version disagreement	0.8	100.0	0.0	33.3	14.9	0.0
Overall	100.0	80.6	99.2	37.5	6.3	0.025

Note: Positive Δ values indicate that the selected full configuration achieves a higher average metric than the replay-guarded diagnostic variant. A consequential run denotes an observable change in reward or completion in either direction. The false-safe rate is normalized by certificate passes.

Table 3 shows that the residual false-safe events are concentrated in a small number of mechanisms. In the selected full configuration, 99.2% of the detected false-safe events are associated with unmodeled local swaps, while the remaining 0.8% involve path version disagreement. ReplayGuard changes at least one final metric in 37.5% of the paired runs, yet the average completed-task difference remains close to zero in the no-delay diagnostic setting. These results suggest that, in this diagnostic setting, the 0.279 false-safe rate mainly reflects winner or bundle changes exposed by fresh-information replay rather than systematic degradation in completed-task performance.

This distinction also clarifies how the false-safe rate should be read in practice. Because the rate is normalized by certificate passes, it captures how often a locally passed certificate is overturned by fresh-information replay. It is therefore a measure of decision consistency, not of mission failure or completed-task loss. In the tested no-delay diagnostic instances, the replay-detected changes had a limited average effect on completed tasks, suggesting that the selected configuration can offer a useful communication–reliability trade-off for delay-tolerant delivery or monitoring missions. For safety-critical or real-time UAV missions, however, the same rate would motivate more conservative deployment choices, such as larger certificate margins, swap-aware checks, shorter retry intervals, or periodic full-synchronization safeguards. These measures should then be validated under the corresponding mission requirements.

Figure 7 summarizes the resulting trade-off across certificate stacks. Task-only checks minimize bytes but leave more residual false-safe events; bundle checks improve stability with modest communication overhead; and swap-enabled checks further reduce false-safe events at the cost of additional bytes.

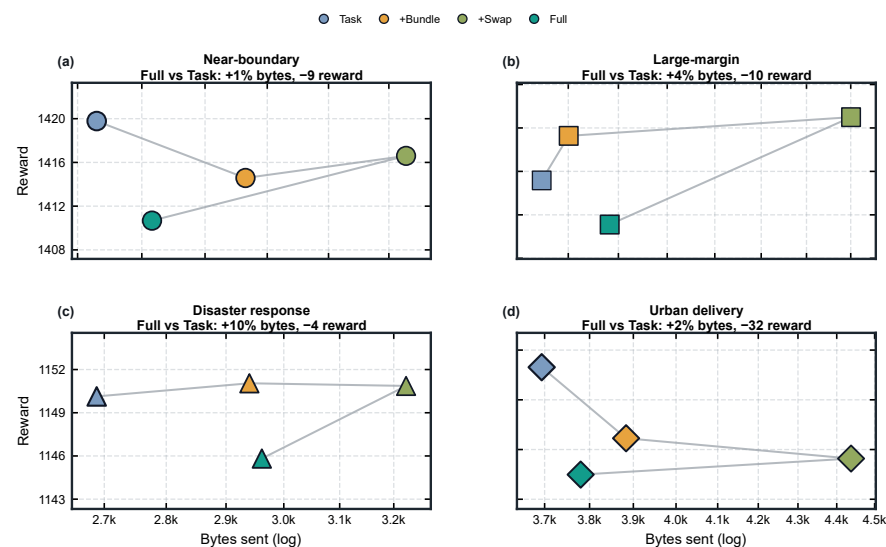


Figure 7. Certificate-stack ablation over 20 paired seeds, showing the reward–byte trade-off across four scenarios. Marker shape denotes the scenario: circles for near-boundary, squares for large-margin, triangles for disaster response, and diamonds for urban delivery.

6.5. Experiment 5: Scalability and Communication–Quality Trade-Off

This experiment evaluates how communication cost and allocation quality change as the numbers of UAVs and tasks increase. We use the near-boundary generator and vary only N and M , while keeping the world size (100), communication range (35), bundle size (4), horizon (40), and reward/deadline distributions fixed. Table 4 reports bytes per completed task and relative reward, while Figures 8 and 9 summarize the corresponding scaling and reward–byte trade-off profiles.

Table 4. Paired-seed scalability stress through the 80-UAV/200-task endpoint (10 seeds). Entries are bytes per completed task; Full and Cons. denote AC-DMTA-Full and AC-DMTA-Conservative. Rel. R_F and Rel. R_C give the rewards of Full and Cons., respectively, divided by the best reward among the listed methods.

UAVs	Tasks	Periodic	ED-CBBA	Clustered	TLC-CBBA	Full	Rel. R_F	Cons.	Rel. R_C
6	18	1420.6	728.9	309.6	501.1	218.9	99.9%	263.7	100.0%
10	30	2312.4	1412.0	797.5	1026.6	698.5	100.0%	698.5	100.0%
20	60	4914.7	3549.1	2201.9	2454.6	1023.5	98.9%	1023.5	98.9%
40	100	11,405.7	8287.0	5608.9	6369.8	2053.1	100.0%	2053.1	100.0%
60	150	18,544.4	14,388.9	9340.9	10,120.4	1882.3	98.7%	2359.1	99.2%
80	200	25,194.2	20,356.2	13,015.8	13,940.4	2080.2	99.2%	2555.0	99.4%

Figure 8 reports the scalability profile up to the 80-UAV/200-task setting, with the corresponding runtime trend shown as an inset. The results show that AC-DMTA’s byte cost grows substantially more slowly than those of the baselines as the task count increases: AC-DMTA-Full rises from 218.9 to 2080.2 bytes per completed task between the 6-UAV/18-task and 80-UAV/200-task settings, whereas periodic CBBA increases from 1420.6 to 25,194.2 over the same range. Figure 9 further plots normalized reward against normalized bytes using paired-seed means, thereby separating communication reduction

from allocation-quality degradation. Across all tested scales, AC-DMTA-Full maintains 98.7–100.0% relative reward. At the 80-UAV/200-task endpoint, it reduces bytes per completed task by 91.7%, 89.8%, 84.0%, and 85.1% relative to periodic CBBA, ED-CBBA, Clustered CBBA, and TLC-CBBA, respectively, while achieving 99.2% relative reward. In this comparison, AC-DMTA-Full uses $\lambda_5 = 0$, certificate margin 1.0, and interval-penalty scale 0.5, whereas AC-DMTA-Conservative uses the same certificate stack but lowers the certificate margin to 0.8. At the largest scale, the conservative variant reaches 2555.0 bytes per completed task and 99.4% relative reward. Together, these results suggest that the decision-centric trigger prevents byte cost from growing at the baseline rate while incurring only small reward reductions at the largest tested scales.

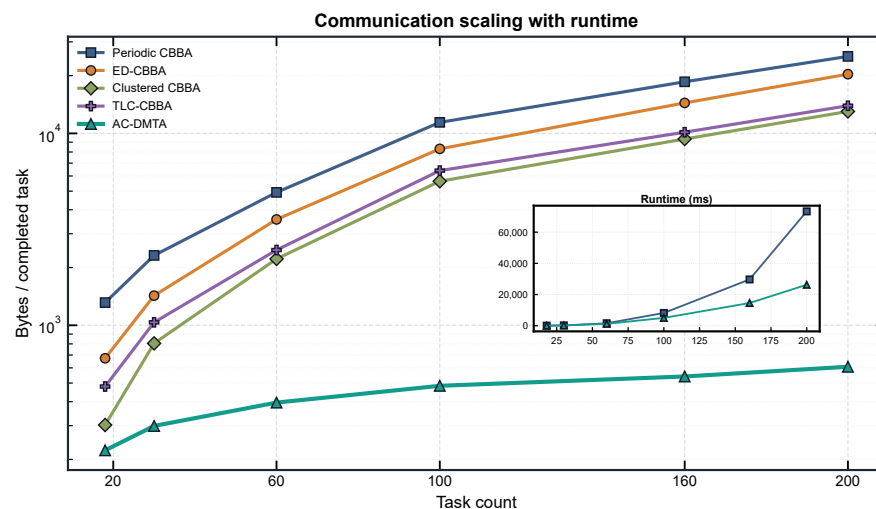


Figure 8. Scalability profile up to the 80-UAV/200-task setting. The main curve reports bytes per completed task across task counts, and the inset reports runtime.

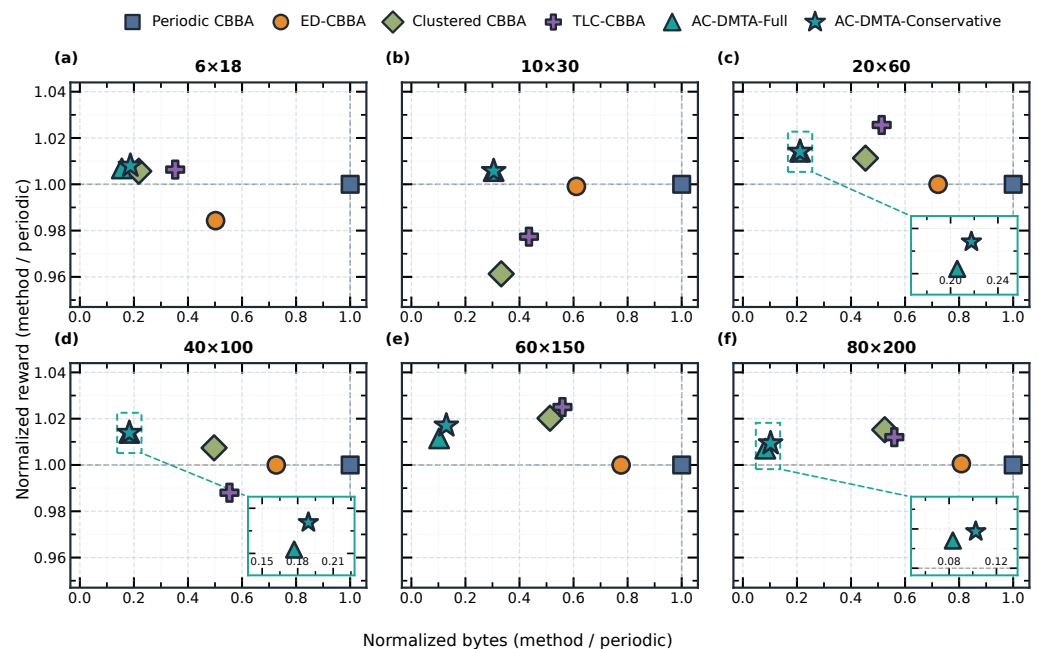


Figure 9. Paired-seed scalability benchmark normalized by periodic CBBA. Markers denote paired-seed means, and the insets in panels (c,d,f) magnify the low-byte region around the AC-DMTA configurations.

7. Discussion

The experiments show that AC-DMTA consistently reduces bytes and directed messages across the tested task allocation environments. These savings do not result from indiscriminate communication suppression. Instead, AC-DMTA uses assignment consistency certificates to skip records whose uncertainty cannot change the current winner or bundle decision, and sends targeted updates only when stale records may affect the allocation decision.

The results also show that AC-DMTA achieves substantial communication savings with limited allocation quality loss. In the main 6-UAV/18-task benchmark, AC-DMTA is best or tied in task completion for 3 of 4 scenarios and remains competitive in the 4th while using substantially fewer bytes. In the scaling stress test, the relative reward remains within 98.7–100.0% across the tested scales.

The replay and ablation results further clarify the sources of residual false-safe events. Most such events are associated with unmodeled local swaps, while a small fraction is related to path-version disagreement. In the diagnostic replay, these events mainly appear as winner or bundle changes, and their average effect on completed tasks remains limited. This indicates that the proposed AC-DMTA method provides a practical balance between communication savings and certificate reliability. For more time-critical or safety-sensitive missions, more conservative certificate margins may be adopted to further improve reliability.

8. Conclusions

This paper introduces AC-DMTA, a decision-centric communication framework for dynamic multi-UAV task allocation under stale and asymmetric information. The underlying idea is that the useful unit of communication is not every state change, but the subset of state information capable of altering the allocation decision. AC-DMTA operationalizes this idea through interval bids, local assignment-consistency certificates, recency-aware deconfliction, and targeted message-value-per-byte ranking.

Across the simulated benchmarks, AC-DMTA reduces bytes and directed messages relative to periodic CBBA, ED-CBBA, Clustered CBBA, and TLC-CBBA while maintaining competitive task completion and reward performance. Within the tested delay, packet loss, and scale ranges, decision-triggered communication provides substantial savings with limited allocation quality loss, and its communication cost grows more slowly than that of the broad-synchronization baselines.

The certificates remain local sufficient conditions, and the residual false-safe events traced to unmodeled swaps and path-version disagreement mark the limits of their guarantees. Future work will examine adaptive certificate margins, richer swap checks under available communication budgets, and stronger path-version handling, as well as evaluation on physical UAV platforms beyond the present simulation study.

Author Contributions: C.Z. (Chenxu Zhao): Writing—original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. M.L.: Conceptualization, Supervision, Writing—review and editing. X.Z.: Writing—review and editing, Investigation. C.Z. (Cheng Zhu): Writing—review and editing, Investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 72501288.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Gerkey, B.P.; Mataric, M.J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [\[CrossRef\]](#)
- Korsah, G.A.; Stentz, A.; Dias, M.B. A Comprehensive Taxonomy for Multi-Robot Task Allocation. *Int. J. Robot. Res.* **2013**, *32*, 1495–1512. [\[CrossRef\]](#)
- Alqefari, S.; Menai, M.E.B. Multi-UAV Task Assignment in Dynamic Environments: Current Trends and Future Directions. *Drones* **2025**, *9*, 75. [\[CrossRef\]](#)
- Ghauri, S.A.; Sarfraz, M.; Qamar, R.A.; Sohail, M.F.; Khan, S.A. A Review of Multi-UAV Task Allocation Algorithms for a Search and Rescue Scenario. *J. Sens. Actuator Netw.* **2024**, *13*, 47. [\[CrossRef\]](#)
- Rinaldi, M.; Wang, S.; Geronel, R.S.; Primates, S. Application of Task Allocation Algorithms in Multi-UAV Intelligent Transportation Systems: A Critical Review. *Big Data Cogn. Comput.* **2024**, *8*, 177. [\[CrossRef\]](#)
- Li, M.; Li, N.; Shao, X.; Wang, J.; Xu, D. Survey on Collaborative Task Assignment for Heterogeneous UAVs Based on Artificial Intelligence Methods. *CAAI Artif. Intell. Res.* **2024**, *3*, 9150033. [\[CrossRef\]](#)
- Song, Y.; Ma, Z.; Chen, N.; Zhou, S.; Srigrarom, S. Comparative Analysis of Centralized and Distributed Multi-UAV Task Allocation Algorithms: A Unified Evaluation Framework. *Drones* **2025**, *9*, 530. [\[CrossRef\]](#)
- Gerkey, B.P.; Mataric, M.J. Sold!: Auction Methods for Multirobot Coordination. *IEEE Trans. Robot. Autom.* **2002**, *18*, 758–768. [\[CrossRef\]](#)
- Dias, M.B.; Zlot, R.; Kalra, N.; Stentz, A. Market-Based Multirobot Coordination: A Survey and Analysis. *Proc. IEEE* **2006**, *94*, 1257–1270. [\[CrossRef\]](#)
- Bertsekas, D.P. The Auction Algorithm: A Distributed Relaxation Method for the Assignment Problem. *Ann. Oper. Res.* **1988**, *14*, 105–123. [\[CrossRef\]](#)
- Choi, H.L.; Brunet, L.; How, J.P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. [\[CrossRef\]](#)
- Johnson, L.B.; Ponda, S.S.; Choi, H.L.; How, J.P. Improving the Efficiency of a Decentralized Tasking Algorithm for UAV Teams with Asynchronous Communications. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010. [\[CrossRef\]](#)
- Whitten, A.K.; Choi, H.L.; Johnson, L.B.; How, J.P. Decentralized Task Allocation with Coupled Constraints in Complex Missions. In Proceedings of the American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 1642–1649. [\[CrossRef\]](#)
- Liu, L.; Michael, N.; Shell, D.A. Communication Constrained Task Allocation with Optimized Local Task Swaps. *Auton. Robot.* **2015**, *39*, 429–444. [\[CrossRef\]](#)
- Dong, N.; Liu, S.; Mai, X. Communication-Efficient Heterogeneous Multi-UAV Task Allocation Based on Clustering. *Commun. Commun.* **2025**, *229*, 107986. [\[CrossRef\]](#)
- Wang, Y.; Wang, C.; Ren, S. A Two-Level Clustered Consensus-Based Bundle Algorithm for Dynamic Heterogeneous Multi-UAV Multi-Task Allocation. *Sensors* **2025**, *25*, 6738. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sao, V.; Ho, T.D.; Bhore, S.; Sujit, P.B. Event Driven CBBA with Reduced Communication. In Proceedings of the 2025 International Conference on Unmanned Aircraft Systems (ICUAS), Charlotte, NC, USA, 14–17 May 2025; pp. 504–510. [\[CrossRef\]](#)
- Gan, W.; Xu, H.; Bai, Y.; Zhou, X.; Wu, W.; Du, X. Large-Scale Multi-UAV Task Allocation via a Centrality-Driven Load-Aware Adaptive Consensus Bundle Algorithm for Biomimetic Swarm Coordination. *Biomimetics* **2026**, *11*, 69. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yu, J.; Zhang, Y.; Sun, C. Multi-UAV Dynamic Task Assignment Based on Event-Triggered Graph Reinforcement Learning Under Weak Communication. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 19106–19121. [\[CrossRef\]](#)
- Dou, W.; Yang, P.; Zhang, Z.; Wang, Z. Cooperative Multi-UAV Search for Prioritized Targets Under Constrained Communications. *Drones* **2025**, *9*, 855. [\[CrossRef\]](#)
- Qazzaz, M.M.H.; Zaidi, S.A.R.; McLernon, D.C.; Salama, A.; Al-Hameed, A.A. Optimizing Search and Rescue UAV Connectivity in Challenging Terrain Through Multi Q-Learning. *arXiv* **2024**, arXiv:2405.10042. [\[CrossRef\]](#)
- Osburn, M.D.; Peterson, C.K.; Salmon, J.L. Task Allocation and Motion Planning in Dynamic, Cluttered Environments via CBBA and Graphs of Convex Sets. *arXiv* **2026**, arXiv:2606.18516. [\[CrossRef\]](#)
- Kuhn, H.W. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [\[CrossRef\]](#)
- Rahman, M.; Sarkar, N.I.; Lutui, R. A Survey on Multi-UAV Path Planning: Classification, Algorithms, Open Research Problems, and Future Directions. *Drones* **2025**, *9*, 263. [\[CrossRef\]](#)
- Han, D.; Jiang, H.; Wang, L.; Zhu, X.; Chen, Y.; Yu, Q. Collaborative Task Allocation and Optimization Solution for Unmanned Aerial Vehicles in Search and Rescue. *Drones* **2024**, *8*, 138. [\[CrossRef\]](#)
- Yan, S.; Xia, Y. A Distributed Task Allocation Method for Heterogeneous UAVs in Dynamic and Communication-Constrained Environments. *J. Supercomput.* **2025**, *81*, 11. [\[CrossRef\]](#)
- Chen, Y.; Chen, R.; Huang, Y.; Xiong, Z.; Li, J. Distributed Task Allocation for Multiple UAVs Based on Swarm Benefit Optimization. *Drones* **2024**, *8*, 766. [\[CrossRef\]](#)

28. Fan, B.; Bo, Y.; Wu, X. Learning Improvement Heuristics for Multi-Unmanned Aerial Vehicle Task Allocation. *Drones* **2024**, *8*, 636. [\[CrossRef\]](#)
29. Jeon, H.M.; Lim, J.W.; Ryoo, C. Task Assignment for Multiple Multi-Purpose Unmanned Aerial Vehicles Using Greedy Algorithm. *Int. J. Aeronaut. Space Sci.* **2024**, *25*, 1380–1394. [\[CrossRef\]](#)
30. Liu, H.; Shao, Z.; Zhou, Q.; Tu, J.; Zhu, S. Task Allocation Algorithm for Heterogeneous UAV Swarm with Temporal Task Chains. *Drones* **2025**, *9*, 574. [\[CrossRef\]](#)
31. Li, C.; Li, G.; Liu, Y.; Zheng, Q.; Yang, G.; Liu, K.; Diao, X. Cooperative Task Allocation for Unmanned Aerial Vehicle Swarm Using Multi-Objective Multi-Population Self-Adaptive Ant Lion Optimizer. *Drones* **2025**, *9*, 733. [\[CrossRef\]](#)
32. Qiu, X.; Zhu, P.; Lu, H.; Zeng, Z. Consensus-Based Dynamic Task Allocation for Multi-Robot System Considering Payloads Consumption. *arXiv* **2024**, arXiv:2412.10087. [\[CrossRef\]](#)
33. Zeng, Y.; Wu, L.; Li, J.; Zhuang, X.; Wu, C. Resilient Task Allocation for UAV Swarms: A Bilevel PSO-ILP Optimization Approach. *Drones* **2025**, *9*, 623. [\[CrossRef\]](#)
34. Yan, S.; Feng, J.; Pan, F. A Distributed Task Allocation Method for Multi-UAV Systems in Communication-Constrained Environments. *Drones* **2024**, *8*, 342. [\[CrossRef\]](#)
35. Zhang, C.; Xu, C.; Li, G.; He, B. A Distributed Task Allocation Approach for Multi-UAV Persistent Monitoring in Dynamic Environments. *Sci. Rep.* **2025**, *15*, 6437. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Zhang, S.; Hu, C.; Zhao, D.; Yang, K.; Xu, Z.; Li, M. A Two-Stage Multi-UAV Task Allocation Approach Based on Graph Theory and a Learning-Inspired Immune Algorithm. *Drones* **2025**, *9*, 599. [\[CrossRef\]](#)
37. Wang, S.; Qiao, P.; Yue, Q.; Xu, Z.; Shang, Q. Research on Dynamic Particle Swarm Optimization for Multi-Objective Reconnaissance Task Allocation of UAVs. *Drones* **2025**, *9*, 556. [\[CrossRef\]](#)
38. Li, Y.; Chen, W.; Fu, B.; Wu, Z.; Hao, L. A Hierarchical Decoupling Task Planning Method for Multi-UAV Collaborative Multi-Region Coverage with Task Priority Awareness. *Drones* **2025**, *9*, 575. [\[CrossRef\]](#)
39. Street, C.; Lacerda, B.; Mühlig, M.; Hawes, N. Right Place, Right Time: Proactive Multi-Robot Task Allocation Under Spatiotemporal Uncertainty. *J. Artif. Intell. Res.* **2024**, *79*, 137–171. [\[CrossRef\]](#)
40. Calvo, A.; Capitan, J. Heterogeneous Multi-Robot Task Allocation for Long-Endurance Missions in Dynamic Scenarios. *IEEE Trans. Robot.* **2025**, *41*, 6494–6513. [\[CrossRef\]](#)
41. Mendoza, M.G.; Su, P.Y.; Ferguson, B.L.; Sastry, S.S. Dynamic Multi-Robot Task Allocation Under Uncertainty and Communication Constraints: A Game-Theoretic Approach. *arXiv* **2026**, arXiv:2604.11954. [\[CrossRef\]](#)
42. Silva, T.C.; Yu, X.; Hsieh, M.A. Probabilistic Multi-Robot Planning with Temporal Tasks and Communication Constraints. *Auton. Robot.* **2026**, *50*, 2. [\[CrossRef\]](#)
43. Beaver, L.E. A Global Games-Inspired Approach to Multi-Robot Task Allocation for Heterogeneous Teams. *arXiv* **2025**, arXiv:2501.01531. [\[CrossRef\]](#)
44. Bezerra, L.C.D.; dos Santos, A.M.G.; Park, S. Learning Policies for Dynamic Coalition Formation in Multi-Robot Task Allocation. *IEEE Robot. Autom. Lett.* **2025**. [\[CrossRef\]](#)
45. Rossano, B.; Lim, J.; How, J.P. Uncertainty-Aware Multi-Robot Task Allocation With Strongly Coupled Inter-Robot Rewards. *arXiv* **2025**, arXiv:2509.22469. [\[CrossRef\]](#)
46. Cao, M.; Peng, J.; Yang, L.; Yang, Y. Multi-UAV Task Allocation and Path Planning in Emergency Rescue Scenarios with Uncertain Requirements. *J. Ind. Manag. Optim.* **2025**, *21*, 6107–6133. [\[CrossRef\]](#)
47. Alqefari, S.; Menai, M.E.B. A Hybrid Method to Solve the Multi-UAV Dynamic Task Assignment Problem. *Sensors* **2025**, *25*, 2502. [\[CrossRef\]](#) [\[PubMed\]](#)
48. Liu, C.; Lu, Y.; Xie, F.; Ji, T.; Zheng, Y. Dynamic Real-Time Multi-UAV Cooperative Mission Planning Method Under Multiple Constraints. *Drones* **2026**, *10*, 132. [\[CrossRef\]](#)
49. Li, G.; Liu, Y.; Zheng, Q.; Liu, K. A Multi-Swarm Dynamic Crow Search Algorithm for Multi-UAV Dynamic Task Allocation. *Drones* **2026**, *10*, 467. [\[CrossRef\]](#)
50. Xu, G.; Wu, Y.; Tao, S.; Yang, Y.; Liu, T.; Huang, T.; Wu, H.; Liu, Y. Multi-Robot Task Allocation and Path Planning with Maximum Range Constraints. *arXiv* **2024**, arXiv:2409.06531. [\[CrossRef\]](#)
51. Jang, I. SPACE: A Python-Based Simulator for Evaluating Decentralized Multi-Robot Task Allocation Algorithms. *arXiv* **2024**, arXiv:2409.04230. [\[CrossRef\]](#)
52. Omidshafiei, S.; Pazis, J.; Amato, C.; How, J.P.; Vian, J. Deep Decentralized Multi-Task Multi-Agent Reinforcement Learning Under Partial Observability. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 2681–2690.

53. Kundu, T.; Rafaei, M.; Gulyaev, A.; Indelman, V. Action-Consistent Decentralized Belief Space Planning with Inconsistent Beliefs and Limited Data Sharing: Framework and Simplification Algorithms with Formal Guarantees. *arXiv* **2025**, arXiv:2403.05962. [[CrossRef](#)]
54. Rafaei Shimron, M.; Indelman, V. Towards Optimal Performance and Action Consistency Guarantees in Dec-POMDPs with Inconsistent Beliefs and Limited Communication. *arXiv* **2025**, arXiv:2512.20778. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.