

## Article

# Don't Fire Together: Desynchronised Communication Scheduling for Bandwidth-Limited Multi-Robot Exploration

Rong Zhao <sup>1,2,†</sup>, Haohua Que <sup>3,†</sup>, Jiajun Sun <sup>3</sup> , Jiayue Xie <sup>3</sup> , Qian Zhang <sup>3</sup> and Fei Qiao <sup>3,\*</sup>

<sup>1</sup> School of Computer Science and Technology, North University of China, Taiyuan 030051, China; 20240008@nuc.edu.cn

<sup>2</sup> Institute for Electronics and Information Technology in Tianjin, Tsinghua University, Tianjin 300467, China

<sup>3</sup> Sense Lab, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; haohuaque@outlook.com (H.Q.); jiajuns757@gmail.com (J.S.); xiejiayue2004@outlook.com (J.X.); zhangq22@mails.tsinghua.edu.cn (Q.Z.)

\* Correspondence: qiaofei@tsinghua.edu.cn

† These authors contributed equally to this work.

## Abstract

Cooperative multi-robot exploration relies on a shared belief over teammates' poses and maps, but bandwidth-limited radios refresh that belief only intermittently. We identify a failure mode beyond simply sending fewer messages: when robots broadcast on a common schedule, the shared belief goes stale *in phase*, the learned explorer acts on the same outdated information across the team, and robots herd toward the same frontiers. We call this the *synchrony tax* and show it is controllable at test time, without retraining. A scheduling layer over a frozen graph-attention policy fixes each robot's broadcast rate and changes only the *phase* of transmissions, cutting travel by 9.8% and sensing overlap by 13.2% (independent replications reach −12.9%). The primary contributing factor is a higher *peak* (not mean) team staleness, which a staleness-to-overlap bound links to redundant coverage; the benefit accordingly disappears on planners that barely use the shared belief. OW-Desync, a budget-constrained activation policy, carries even staggering to range-limited, heterogeneous channels, improving the freshness it controls while matching staggering on the task. Four-robot experiments over real WiFi reproduce the timing-to-staleness stage of the mechanism at a matched rate: across five repeated paired trials, staggering cuts the measured peak team staleness by 37% (all pairs concordant, within 1.5% of the predicted optimum) with the mean unchanged.

**Keywords:** multi-robot exploration; bandwidth-limited communication; event-triggered communication; desynchronised scheduling; communication synchronisation; LiDAR; ROS2



Academic Editor: Paolo Mercorelli

Received: 19 June 2026

Revised: 20 July 2026

Accepted: 24 July 2026

Published: 27 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](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Teams of mobile robots carrying onboard LiDAR and running real-time LiDAR-inertial odometry [1,2] can explore large unknown environments far faster than a single robot, in applications from search-and-rescue to inspection and subterranean mapping [3,4]. What limits such a team is rarely individual sensing. It is the freshness of the *shared belief* each robot holds about its teammates' poses and maps, which has to travel over a wireless link of limited and contended bandwidth [5–7]. How a team spends that scarce link, how often and *when* each robot shares its state, is a first-class sensing-system design question and the subject of this paper.

Learned explorers make the question acute. State-of-the-art policies such as ARI-ADNE [8] and its bandwidth-limited extension [9] exchange a learned message at *every* planning step [4,8,9], which onboard radios with finite bandwidth and packet loss cannot sustain. Throttling communication is the obvious fix, but it exposes a failure mode that has little to do with sending fewer messages. When a budgeted team broadcasts on a *common* schedule, every robot's view of its teammates goes stale at the same moment; the learned policy then acts on correlated, out-of-date beliefs, so the robots choose the same frontiers, cover ground twice, and travel further than they need to. We call this the *synchrony tax*: limited bandwidth forces synchronised updates, synchronised updates correlate belief staleness, and correlated staleness turns into duplicated frontier choices and wasted travel.

The remedy is to treat *phase* as a resource. At a fixed per-robot broadcast rate, *when* each robot transmits relative to the others is as controllable as how much it sends, and far cheaper to change. We show that communication timing is an overlooked control knob: at the same nominal broadcast rate, changing only the phase of transmissions removes a substantial efficiency penalty that synchronised schedules pay under scarce communication. This is naturally read in the language of information freshness. A robot acts on the last update it received, whose error grows with its age, so the team pays for coordination through how stale its shared belief is at decision time. Age-of-Information theory shows that on a shared or interference-limited medium this freshness is set by *when* sources transmit relative to one another, and that spreading correlated sources in time is close to optimal [10–12]. Because the scheduler runs on a frozen policy at test time, it needs no retraining and adds sub-millisecond overhead per step at  $N = 8$  (a few milliseconds at  $N = 20$ ), so it applies to an already-deployed explorer.

The novelty of this paper is that it identifies broadcast phase as a coordination resource that communication-limited multi-robot learning has left largely unexamined; explains its effect through a single supported mechanism (the peak, not the mean, of team staleness); shows that exploiting it is free (a task-level gain at a matched per-robot rate in simulation, whose timing-to-staleness stage is reproduced quantitatively on four robots over real WiFi); and charts its boundary: the benefit appears exactly where coordination runs through a shared belief, plain even staggering already attains the bound optimum on a symmetric medium, and OW-Desync carries the same principle to heterogeneous, interference-limited channels. Our contributions are as follows:

1. **The synchrony tax.** At a fixed per-robot broadcast rate, changing only the *phase* of transmissions cuts travel and sensing overlap by 9.8% and 13.2% on the learned explorer (independent replications of the same condition reach  $-12.9\%$  and  $-14.4\%$ ), largest at the scarcest budget tested. This is a test-time phenomenon that needs no retraining, and desynchronised schedules in fact send up to 16% more bytes, so the saving is not explained by the nominal broadcast rate or realised byte savings (Section 4).
2. **A staleness mechanism.** The cost is not a worse *average* age but a higher *peak* team staleness. Synchronised schedules make all beliefs stale together; even staggering lowers that peak while preserving its mean, which a staleness-to-overlap bound links to redundant sensing (Section 3.4). The benefit disappears on planners that barely deconflict through the shared belief (with a small apparent cost for the most myopic one), which is the disappearance the mechanism predicts. Four-robot WiFi experiments reproduce the timing-to-staleness stage of the mechanism on hardware: at a matched rate, staggering cuts the measured peak team staleness by 34–37% across six paired runs on two fleets (five repeated trials: all pairs concordant, paired  $t$   $p < 10^{-7}$ ), close to the no-jitter optimum, with the mean unchanged (Section 4.9).

3. **From even staggering to OW-Desync.** Even staggering is the natural schedule on a symmetric shared medium; on a range-limited, heterogeneous interference graph we derive **OW-Desync**, a budget-constrained activation policy that generalises it [11,13]. OW-Desync is a principled extension of the mechanism to lossy, heterogeneous channels; it provably improves the freshness objective it controls and, on the exploration task, *matches* even staggering rather than beating it, so the task-level gains come from desynchronisation itself rather than from OW-Desync specifically (Section 3).

Throughout, the scheduler is a test-time layer over a frozen explorer, not a new exploration policy.

## 2. Related Work

### 2.1. Communication-Constrained Multi-Robot Exploration and Real-Robot Systems

Autonomous exploration was classically driven by frontier selection [14], coordinated frontier assignment for teams [4], and sampling-based trees [15]; deep reinforcement learning later addressed cluttered single-robot exploration [3] on standardised benchmarks [16], and learned map prediction accelerates exploration further by inferring unseen structure from the partial map [17,18]; for teams, attention-based graph policies (ARiADNE [8]) and asynchronous multi-agent RL [19] now lead. A privileged-learning variant explicitly targets *bandwidth-limited* multi-robot exploration [9], and we use this as the frozen backbone for which we schedule communication; such systems share learned messages frequently and largely assume an abundant channel, which is the gap we address. Classical, non-learned pipelines have long treated communication as a constraint on coordination: coordinated frontier assignment [4], planning under periodic-connectivity requirements [20], information-theoretic control of distributed sensing [21], and hierarchical decentralised exploration robust to limited communication [22] constrain *where* robots may go so that the channel suffices, and the survey of [23] maps this landscape; we instead fix the budget and control *when* it is spent. Orthogonal per-robot constraints, such as energy-efficient legged locomotion and foothold planning [24], compound these coordination costs on physical platforms. Closely related communication-limited exploration work reduces what is sent (e.g., sharing only positions) or plans intermittent rendezvous [25]; we instead hold the per-robot broadcast rate fixed and control *timing*. The same communication pressure appears in real robotic stacks built on LiDAR odometry/mapping (LOAM [1], LeGO-LOAM [26], LIO-SAM [27], 2D SLAM [28], small-FoV LiDAR odometry [29], FAST-LIO2 [2]) and multi-robot SLAM (DOOR-SLAM [5], Kimera-Multi [6], and centralised collaborative dense TSDF mapping [7]), which motivates bandwidth-aware coordination when deployed on ROS 2 [30] with Nav2 [31].

### 2.2. Information Freshness and Communication Scheduling

Age-of-Information minimises staleness under an update budget [32–34]; on a symmetric shared medium, a greedy max-age-first/round-robin rule is age-optimal, while general networks call for Whittle-index restless-bandit schedulers [10,35], and value-of-information weights an update by its task impact [36,37]. Transmitting only on sufficient change is the event-/self-triggered-control idea [38–42], of which our drift-triggered broadcast gate is one self-desynchronising instance. Two structural results anchor our method. First, for *correlated* sources even simple staggering is *near-optimal*: round-robin is provably within a small constant of optimal [43] and correlation-aware scheduling cannot beat correlation-agnostic age scheduling by more than a constant factor [12,44]; this is why, in our symmetric regime, our Proposition 3 (evenly spaced offsets  $\phi_k = \lfloor (k-1)T/N \rfloor$  minimise peak staleness) is hard to beat. Second, on an *interference graph* where only an independent set may transmit per slot, a stationary randomised activation policy is exactly

peak-AoI-optimal [11]; OW-Desync instantiates this for the multi-robot-exploration overlap cost with heterogeneous weights and a per-slot budget, recovering even staggering as the symmetric special case. Decentralised self-organising desynchronisation (DESYNC [13], recently demonstrated on a drone swarm [45]) realises evenly spaced broadcasts from local overhearing with no coordinator. Our contribution is to carry this freshness-scheduling theory from a networking objective to a robotics *task* cost (sensing overlap and travel) under a frozen learned explorer.

### 2.3. Learned and Test-Time Communication on Frozen Policies

Cooperative multi-agent RL learns *what*, *whether*, and *with whom* to communicate. Representative frameworks include differentiable inter-agent learning [46], CommNet [47], BiCNet [48], ATOC [49], I2C [50], MAGIC [51], and TarMAC [52] atop centralised-training value factorisation [53,54] and attention [55]. The most relevant are budget-aware methods: SchedNet [56], IC3Net [57], When2com [58], decomposability/information-bottleneck compression [59,60], event-triggered MARL [61], value-of-information gating [62], model-based communication that skips predictable messages [63], and a recent survey [64]. These learn the communication policy end-to-end; we instead provide a lightweight, test-time scheduling *layer* for a *frozen* explorer and isolate *synchronisation* as the dominant cost. We separate two orthogonal axes of communication economy: *how much* to transmit (rate/volume, the target of the above) versus *when* at a fixed broadcast rate (phase/synchronisation, ours), and our baselines instantiate prior paradigms: periodic = time-triggered, distance-triggered = event-triggered [40], staleness-greedy = Age-of-Information [32], round-robin/random = structured/naive budgeted schedules [56].

## 3. Materials and Methods

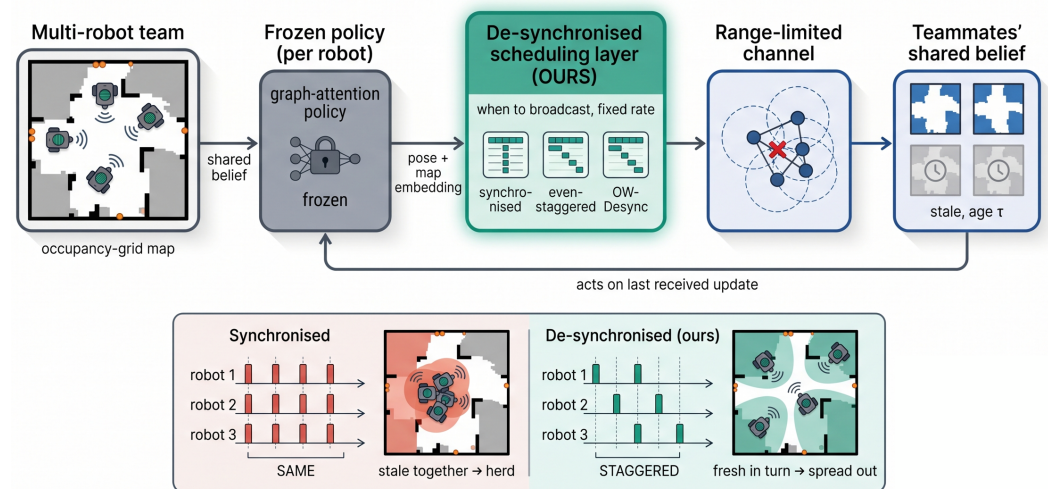
### 3.1. System Model and Problem

Figure 1 summarises the setup.  $N$  robots explore an unknown 2-D environment; robot  $i$  holds a belief occupancy map and a 64-dimensional learned state embedding produced by the frozen policy [9]. Coordination relies on each robot's pose and embedding, shared over a bandwidth-limited channel. Both the pose and the embedding travel on this gated channel, and there is no free side-channel that silently keeps poses fresh. For this reason the *schedule*, rather than the message *volume*, governs exploration efficiency. On a silent link the receiver must act on a stale pose, so the *timing* of refreshes changes *where* robots go, and not just *how many* bytes they send. We consider two budget models: (i) an average broadcast rate (a robot either broadcasts to all or stays silent), and (ii) a directed-link budget of  $B$  transmissions per step among the  $L = N(N - 1)$  ordered robot pairs (point-to-point). In the stationary OW-Desync analysis this appears as an average activation budget  $\sum_i f_i \leq B$ ; round-robin and greedy link schedulers use the corresponding hard per-step cap where stated. On a silent link the receiver retains the last received (stale) pose/embedding. Because coverage saturates, the objective is efficiency: min communication s.t. coverage  $\geq \tau$ ; equivalently min travel/overlap at a fixed budget.

### Communication Accounting

Three notions are easy to conflate, so we separate them. The *nominal rate* is how often a robot is scheduled to broadcast (once per period  $T$ , i.e.,  $N/T$  transmissions per step); this is what the phase-only experiments hold fixed. The *realised byte volume* is the payload actually sent, which can vary with map and update content, and which we report separately rather than constrain. The *broadcast model* (i) has one sender reach the whole team, whereas the *directed-link model* (ii) caps point-to-point messages and underlies the range-limited interference setting of OW-Desync (Section 3.4). Unless stated otherwise the

main experiments compare schedules at a matched *nominal broadcast rate* and vary only *phase*; we do not force every byte to match, and in fact desynchronisation sends up to 16% more bytes, which only sharpens the conclusion that the controllable cost is timing.



**Figure 1.** System overview. A frozen, pretrained multi-robot explorer (**left**) plans against a shared belief of teammates’ poses and map embeddings. We insert a test-time desynchronised scheduling layer that decides, at a fixed per-robot rate, *when* each robot broadcasts (synchronised, even-staggered, or OW-Desync) over a range-limited, lossy channel; this sets how stale each teammate’s belief is, and hence travel and sensing overlap. Bottom: Broadcasting on a common schedule (**left**) aligns the robots’ updates, so their beliefs go stale together and they herd onto the same frontiers, whereas desynchronising the phase at the *same* rate (**right**) staggers the updates, so the team stays fresh in turn and spreads out. This wasted-effort gap is the *synchrony tax*.

### 3.2. Communication Schedulers

We compare four broadcast schedulers, all operating on the *frozen* exploration policy at test time. Each robot  $i$  either broadcasts its current pose and embedding to the team or stays silent; on a silent step every receiver keeps the last copy it received and acts on that stale state. The schedulers differ only in *when* robots broadcast, and they are matched to the same nominal per-robot broadcast rate. The two periodic conditions transmit exactly once per period; the random and event-triggered conditions are calibrated to the same rate in expectation and over the aggregate evaluation. Thus, differences in exploration efficiency are attributable to broadcast *timing* rather than to systematic differences in how often robots transmit. Realised byte volume is reported and discussed in Section 4.

#### 3.2.1. Synchronised Periodic (Control)

Every robot broadcasts together every  $T$  steps, in lockstep. This is the textbook time-triggered schedule [40] and serves as the control: all robots refresh at the same instant, so the team’s beliefs go stale and are refreshed in phase.

#### 3.2.2. Staggered Periodic (Desynchronised)

Each robot still broadcasts once per period  $T$ , but robot  $k$  ( $k = 1, \dots, N$ ) is given a balanced, as-even-as-possible phase offset  $\phi_k = \lfloor (k-1)T/N \rfloor \in \{0, \dots, T-1\}$ ; when  $T < N$ , this balances how many robots share each slot rather than giving distinct offsets. The per-robot rate (number of transmissions per robot) is identical to the synchronised control; only the broadcast *phase* differs. Realised byte volume is *not* held identical: desynchronisation transmits slightly *more* bytes (Section 4), so any efficiency gain cannot be bought with bandwidth.

### 3.2.3. Random (Desynchronised)

Each robot broadcasts independently, with a fixed probability chosen to match the periodic rate. Independent draws make the team's broadcasts unlikely to coincide, giving a desynchronised schedule without any imposed phase structure.

### 3.2.4. Event-Triggered

A robot broadcasts when its state, measured by the drift of its pose and embedding since its last broadcast, exceeds a threshold, otherwise it stays silent [38,40]. Because each robot's drift crosses the threshold at its own time, this is a principled, self-desynchronising trigger; the threshold is calibrated to the matched budget.

Algorithm 1 summarises one step for the periodic family; random and event-triggered schedulers replace the firing predicate (phase match) with a Bernoulli draw or a drift test, respectively. The exploration backbone is frozen throughout: only the schedule changes, never the policy.

---

#### Algorithm 1 Matched-budget broadcast scheduling (one step at time $t$ )

---

**Require:** robots  $\{1, \dots, N\}$ ; period  $T$ ; phase offsets  $\{\phi_k\}$  ( $\phi_k = 0 \ \forall k$  synchronised,  $\phi_k = \lfloor (k-1)T/N \rfloor$  staggered)

- 1:  $\mathcal{F} \leftarrow \{k : (t - \phi_k) \bmod T = 0\}$   $\triangleright$  robots whose broadcast slot is due
- 2: **for all**  $k \in \mathcal{F}$  **do** broadcast  $(p_k, \text{embedding}_k)$  to all receivers;  $\hat{p}_j^{(k)} \leftarrow p_k \ \forall j$
- 3: **end for**
- 4: **for all** receivers  $j$  and silent senders  $k \notin \mathcal{F}$  **do** keep stale  $\hat{p}_j^{(k)} \triangleright$  act on last received state
- 5: **end for**

---

### 3.2.5. Derived Closed-Loop Schedulers, and Why Even Staggering Is the Near-Optimal Default

The schedules above are open-loop. One can instead act on the staleness  $\rightarrow$  overlap bound of Section 3.4: since that bound is the contention-weighted team staleness  $\sum_{ij} w_{ij} \tau_{ij}$ , the greedy step refreshes the robots that are simultaneously *stale* and *contending*, broadcasting at the periodic budget the robots of largest index  $\tau_i \cdot c_i$ , where  $\tau_i$  is staleness and  $c_i = \sum_{j \neq i} \exp(-\|\hat{p}_i - \hat{p}_j\|^2 / 2\sigma^2)$  is contention from believed teammate positions, the Age-of-Information max-weight/Whittle-index policy [10,44] cast as a desynchronisation rule. We likewise evaluated a joint-submodular max-coverage scheduler and a value/loss-aware Whittle index. In the homogeneous limit all three reduce to refreshing the stalest robot in turn, i.e., evenly spaced phase staggering, which Proposition 3 proves bound-optimal. On our maps *none* of these sophisticated schedulers significantly improves on plain even staggering (Section 4), exactly as correlated-source Age-of-Information theory predicts [12,43]: on a symmetric single broadcast medium, uniform staggering is within a small constant of optimal, so added scheduler complexity cannot pay off there. We therefore adopt even staggering as the practical default in the symmetric regime, and treat this null as positive evidence for its near-optimality.

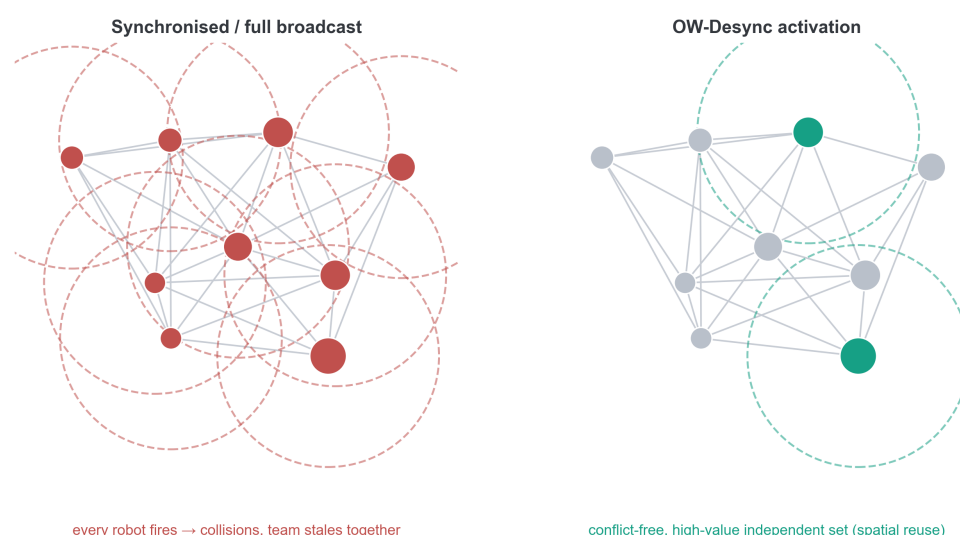
### 3.2.6. OW-Desync: The Optimal Schedule for a Range-Limited Interference Channel

Real radios have finite range: a broadcast reaches only receivers within  $R_{\text{comm}}$ , and two nearby transmitters that fire in the same slot collide. The feasible transmissions per slot are then an *independent set* of the interference graph  $G$  (edge  $(i, j)$  iff robots  $i, j$  interfere; Figure 2), and robots are *heterogeneous*: a robot with a larger sensor footprint or richer local frontier carries more team value, giving a per-robot weight  $w_i$  (overlap-coupling  $\times$

informativeness) and channel reliability  $\gamma_i$ . Writing  $f_i$  for robot  $i$ 's long-run activation frequency, the weighted peak Age-of-Information  $A_p$  and the OW-Desync schedule are

$$A_p(f) = \sum_i \frac{w_i}{\gamma_i f_i}, \quad f^* = \arg \min_{f \in \mathcal{P}(G, B)} A_p(f), \quad (1)$$

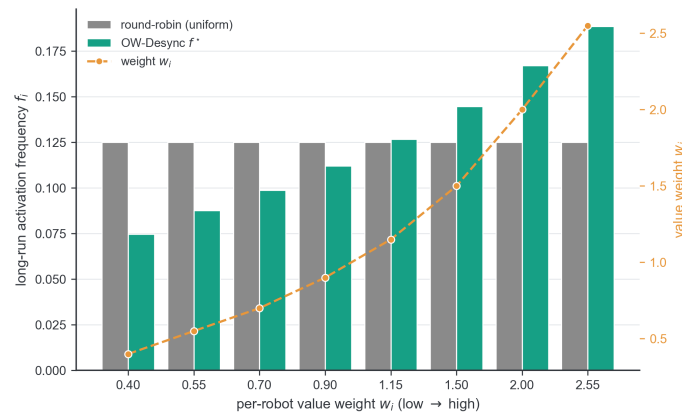
where  $\mathcal{P}(G, B)$  is the budget-scaled stable-set polytope of the interference graph  $G$ , so the program is convex. It is realised by a stationary policy that samples independent sets with marginals  $f^*$  (the policy of Theorem 1); online, we approximate it by transmitting, each slot, the conflict-free maximum-weight set of weight  $w_i \tau_i$ . When weights and reliabilities are symmetric and  $G$  is complete (a single shared medium),  $f^*$  is uniform and OW-Desync reduces exactly to even phase staggering; under heterogeneity it serves high-value robots more often (Figure 3) and is provably peak-age-optimal (Theorem 1, generalising the interference-graph optimality of [11]). It thus subsumes even staggering as its symmetric special case and extends it to the realistic interference-limited, heterogeneous setting. The activation frequencies can also be reached *decentrally*, from local overhearing with no coordinator, by a self-organising desynchronisation update [13,45].



**Figure 2.** The range-limited interference channel and OW-Desync activation. **(Left)** Synchronised or full-mesh broadcasting fires every robot in the same slot, so co-located transmitters (dashed range rings) collide and the whole team's beliefs go stale together. **(Right)** OW-Desync transmits, each slot, a conflict-free *independent set* of the interference graph, preferring high-value robots (marker area  $\propto$  weight  $w_i$ ), while non-interfering robots reuse the medium spatially. Red nodes denote simultaneous, mutually interfering transmissions; teal nodes form the activated independent set; grey nodes are inactive. Dashed circles show communication/interference range, and solid grey edges join interfering robot pairs. When the graph is complete and robots are symmetric, this reduces exactly to even staggering.

### 3.3. Complexity and Overhead

Per step, each scheduler makes  $N$  independent broadcast decisions, each  $O(1)$ : a phase test for the periodic schedules, a Bernoulli draw for random, or a state-drift threshold test for the event-triggered gate; memory is  $O(N^2)$  for the per-receiver beliefs and staleness counters. Measured in our CPU implementation, the scheduling step is sub-millisecond ( $\approx 0.6$  ms at  $N = 8$  and a few milliseconds at  $N = 20$ ), which is negligible relative to the team's per-step graph-attention policy inference and well below typical control periods. The overhead is negligible, and the scheduler runs comfortably onboard in real time. Because the backbone is frozen, no training-time cost is incurred.



**Figure 3.** How OW-Desync works in the frequency domain. On a shared medium the convex program assigns each robot a long-run activation frequency  $f_i^* \propto \sqrt{w_i/\gamma_i}$  (water-filling), giving more airtime to high-value robots (teal) than the uniform round-robin allocation (grey), which lowers the weighted peak Age-of-Information  $A_p$ . The reduction grows with weight heterogeneity as shown later in the heterogeneity sweep; when weights are symmetric,  $f^*$  is uniform and OW-Desync reduces to even staggering.

### 3.4. Theoretical Analysis

We now make the synchrony-tax mechanism precise: staleness bounds belief error (Lemma 1), correlated staleness inflates the overlap envelope (Proposition 2, which formalises a sufficient mechanism linking staleness to overlap rather than a tight numerical guarantee, as the envelope  $\Phi$  is only assumed non-decreasing;  $\Phi$  is deliberately abstract rather than a fitted functional form, the linear envelope  $\Phi(x) = \alpha x$  being its simplest admissible instance, with its induced empirical relation evaluated in the Results), and spreading broadcasts in time minimises that envelope (Proposition 3 on a shared medium, Theorem 1 on a general interference graph). The proofs are deferred to Appendix A.

**Notation 1.** Time is discrete,  $t = 0, 1, \dots$ . Robot  $i$  has true position  $p_i(t) \in \mathbb{R}^2$ . The directed communication links are  $\mathcal{L} = \{(i \rightarrow j) : i \neq j\}$ , with  $|\mathcal{L}| = L = N(N-1)$ ; at most  $B$  links deliver per step (a hard per-step cap, used by the periodic and round-robin schedules; the OW-Desync model of Theorem 1 instead constrains the average activation rate,  $\sum_i f_i \leq B$ , at the same nominal budget; there,  $f_i$  counts robot broadcasts, each delivering the robot's  $N-1$  outgoing links together). Receiver  $j$  stores a belief  $\hat{p}_j^{(i)}(t)$  of sender  $i$ 's position; on delivery of link  $i \rightarrow j$  at step  $t$ ,  $\hat{p}_j^{(i)}(t) \leftarrow p_i(t)$ , otherwise it is unchanged. The staleness  $\tau_{ij}(t)$  is the number of steps since link  $i \rightarrow j$  last delivered, and the belief error is  $e_{ij}(t) = \|p_i(t) - \hat{p}_j^{(i)}(t)\|$ . Let  $v_{\max} = \max_{i,t} \|p_i(t) - p_i(t-1)\|$  be the maximum single-step displacement and  $D$  the environment diameter (so,  $e_{ij} \leq D$ ).

**Lemma 1** (Belief error grows linearly with staleness). For all  $i, j, t$ ,  $e_{ij}(t) \leq \tau_{ij}(t) v_{\max}$ ; a stale belief drifts no faster than the robot moves.

**Proposition 1** (Staleness bounds). Round-robin scheduling (a fixed cyclic order,  $B$  deliveries/step) satisfies  $\tau_{ij}(t) \leq \lceil L/B \rceil - 1$  for all  $i, j$  and  $t \geq \lceil L/B \rceil$ ; in particular, any periodic schedule that serves every link once per cycle keeps staleness uniformly bounded.

**Assumption 1** (Deconfliction). Two robots sense redundantly only when  $\|p_i - p_j\| \leq 2R$  ( $R$  the sensing radius); the policy deconflicts using believed teammate positions, so a deconfliction failure for a near (“contending”) pair becomes more likely, and not less, as the receiver’s belief error  $e_{ij}$  grows. This holds to the extent that the planner actually coordinates on the shared teammate

belief: a planner that couples only weakly on that belief violates it, and the scheduling effect should then be weak or absent. This matches what we observe, with desynchronisation strong on the learned graph-attention backbone (which coordinates through the shared belief), weak on the utility frontier, and gone, indeed with a small apparent cost, on the myopic nearest-frontier planner (Section 4).

**Proposition 2** (Overlap bound). Under Assumption 1, the expected per-step team overlap satisfies

$$\mathbb{E}[\text{overlap}(t)] \leq \Phi\left(\sum_{(i,j)} w_{ij}(t) e_{ij}(t)\right) \leq \Phi\left(v_{\max} \sum_{(i,j)} w_{ij}(t) \tau_{ij}(t)\right), \quad (2)$$

where  $w_{ij}(t) = \mathbb{1}[\|p_i - p_j\| \leq 2R + \delta]$  marks contending pairs ( $\delta$  a margin) and  $\Phi$  is non-decreasing.

**Remark 1** (Why desynchronisation helps). From the bound  $\Phi(v_{\max} \sum_{ij} w_{ij} \tau_{ij})$ : at a fixed communication budget the total number of deliveries is fixed, so a synchronised schedule makes all  $\tau_{ij}$  peak simultaneously (correlated staleness), maximising the instantaneous  $\sum_{ij} w_{ij} \tau_{ij}$  during the team's silent phase and thus the overlap envelope of Proposition 2, whereas staggering spreads deliveries so this peak, and hence the overlap, is lower. The argument runs through Assumption 1, so it predicts a benefit only for planners that deconflict on the shared teammate belief: desynchronisation is a property of such learned, belief-coupled coordination rather than a universal law.

**Proposition 3** (Evenly spaced staggering is optimal in the symmetric case). On a single shared medium in the symmetric, fully contending case (equal, time-invariant link weights with all pairs contending, so every sender carries the same constant outgoing weight), consider periodic broadcast schedules with common period  $T$  in which each robot  $k$  broadcasts to all others exactly once per period, at the phase-shifted steps  $t \equiv \phi_k \pmod{T}$  with offset  $\phi_k \in \{0, \dots, T-1\}$ . Every such schedule has the same per-robot rate and the same broadcast count. Among all offset assignments  $\phi = (\phi_1, \dots, \phi_N)$ , the worst-case team-staleness peak  $\max_t \sum_{ij} w_{ij} \tau_{ij}(t)$ , and hence the overlap envelope of Proposition 2, is minimised by the balanced assignment  $\phi_k = \lfloor (k-1)T/N \rfloor$  (exactly optimal when  $N$  divides  $T$ , and within  $a(N-1)$  of a universal lower bound on the peak for general  $T \geq N$ , with  $a$  the common contention weight of the proof; distinct offsets when  $T \geq N$ , balanced slot loads when  $T < N$ ) and is maximised by the synchronised assignment  $\phi_1 = \dots = \phi_N$ .

**Theorem 1** (OW-Desync minimises the weighted expected peak age on the interference graph). On an interference graph  $G$  where each slot one feasible (independent) set transmits under an average per-slot budget  $B$  (the constraint  $\sum_i f_i \leq B$  below), with per-robot weights  $w_i$  and channel reliabilities  $\gamma_i$ , the stationary policy activating independent sets with the marginal frequencies  $f^* = \arg \min_{f \in \mathcal{P}} \sum_i w_i / (\gamma_i f_i)$  of Equation (1), where  $\mathcal{P} = \{f : f = \sum_S x_S \mathbf{1}_S, x \geq 0, \sum_S x_S \leq 1, \sum_i f_i \leq B, S \text{ independent in } G\}$  is the budget-scaled stable-set polytope, minimises the weighted expected (time-average) peak age  $A_p = \sum_i w_i / (\gamma_i f_i)$  over all stationary policies on  $G$  (and is within a constant factor of all policies [11]). Here,  $f_i$  is robot  $i$ 's long-run activation frequency; each activation succeeds independently with probability  $\gamma_i$ , so link  $i$  delivers at rate  $\gamma_i f_i$  and, under the stationary geometric-success age model of [11], has time-average peak age  $1/(\gamma_i f_i)$ .

Theorem 1 imports its optimisation core from the interference-graph peak-age framework of [11]; we claim no new optimisation theory. Its contribution is the transfer to exploration: task-derived overlap-coupling weights  $w_i$ , an explicit activation budget, and the reduction of Corollary 1, which shows that the exploration-motivated even staggering is exactly the symmetric case.

**Corollary 1** (Even staggering is the symmetric special case). If  $w_i \equiv w$ ,  $\gamma_i \equiv \gamma$  and  $G$  is complete (a single shared medium), then  $f_i^* \equiv \min(1, B)/N$  and OW-Desync coincides exactly

with the evenly spaced staggering of Proposition 3; Proposition 3 is thus the symmetric, complete-graph case of Theorem 1.

**Proposition 4** (Strict improvement under heterogeneity, shared-medium case). *In the shared-medium (simplex) case, where the only active constraint is the total budget  $\sum_i f_i \leq B$ , if the ratios  $w_i/\gamma_i$  are not all equal, then  $f^*$  is non-uniform and  $A_p(f^*) < A_p(f^{\text{RR}})$  strictly, where  $f^{\text{RR}}$  is the uniform (round-robin) allocation, and the gap grows with the dispersion of  $w_i/\gamma_i$ . On a general interference graph the optimum is still characterised by the convex program of Theorem 1, without a closed form.*

**Corollary 2** (Overlap envelope). *With  $w_i$  the overlap-coupling of robot  $i$ , OW-Desync's  $f^*$  minimises the overlap envelope of Proposition 2.*

### 3.5. Experimental Setup

We evaluate on the frozen pretrained checkpoint of [9] over 20 held-out maps, with parallel CPU evaluation. Simulation experiments use Python 3.8, PyTorch 2.3.1, Ray 2.10.0, and SciPy 1.10.1. The hardware software stack uses ROS 2 Humble, the Nav2 Humble distribution, and the FAST-LIO2 repository implementation. Metrics: final coverage ratio, total travel distance, sensing overlap ratio, and communication volume (bytes). Significance is assessed with paired Wilcoxon signed-rank tests and bootstrap 95% confidence intervals over matched (map, seed) pairs. Throughout the figures and tables we mark these paired tests as \* for  $p < 0.05$ , \*\* for  $p < 0.01$ , and “n.s.” (not significant) for  $p \geq 0.05$ ; a non-significant OW-Desync-versus-even result is the intended “does no harm” outcome. We treat the phase-only control and the matched-budget comparison as the primary tests; the team-size, planner-backbone, correlation, packet-loss, and sensor-range sweeps are exploratory, reported with nominal  $p$ -values. The two primary tests were specified in advance of the exploratory sweeps, and the headline conclusions rest on them (both remain significant under Holm correction); the individually marginal exploratory points (the 20% packet-loss dip,  $p = 0.057$ , and the 40 m sensor-range point,  $p = 0.051$ ) carry no conclusion. The decisive phase-only contrast (staggered versus synchronised periodic,  $N = 8$ , interval 12) recurs across several independently generated batches below (different map/seed draws and spawn layouts): the three-period sweep reported below ( $n = 160$  per period), the team-size sweep ( $n = 120$ ), the backbone run ( $n = 100$ ), the ordering check ( $n = 160$ ), and the correlation run ( $n = 160$ ). Its travel effect ranges from  $-9.8\%$  to  $-12.9\%$  across these replications; we quote the conservative three-period-sweep values as the headline. The baselines instantiate standard prior paradigms rather than proposed methods. *Original* (every-step broadcast) is the no-throttling reference condition; it is infeasible under the bandwidth budgets studied and is not evaluated. *Periodic* is the textbook synchronised time-triggered control [40], included as the control needed for the desynchronisation comparison and not as a method we advocate. *Staggered* (phase-offset periodic) is our de-sync control; *EventTrig* (distance/event-triggered [38,40]) and *Random* complete the naive fixed-rate schedules; *round-robin* appears as the uniform allocation in the OW-Desync freshness comparison. These fixed-volume schedules connect to Age-of-Information broadcasting, where a max-age-first/greedy-AoI rule is the natural age-based baseline [10,32]. The implementation's broadcast *interval* and the theory's *period*  $T$  denote the same quantity and are used interchangeably. Throughout, we hold the per-robot broadcast rate fixed (realised byte volume is reported separately) and vary only *when* robots broadcast, isolating schedule structure from bandwidth.

## 4. Results

### 4.1. Synchronisation Is the Controllable Cost

The benefit of desynchronisation first appears even in an approximately byte-matched comparison. At a matched, scarce broadcast budget ( $N = 8$ ,  $\approx 47$ – $50$  kB, identical coverage), merely changing the temporal pattern of broadcasts cuts travel by 10.6% and overlap by 10.8% relative to the synchronised periodic control (Table 1). This is the first sign of the synchrony tax: at essentially the same realised bytes, schedules that avoid lockstep updates waste less motion. The next control is the decisive one, isolating phase itself by comparing synchronised and staggered periodic schedules at the same per-robot broadcast rate.

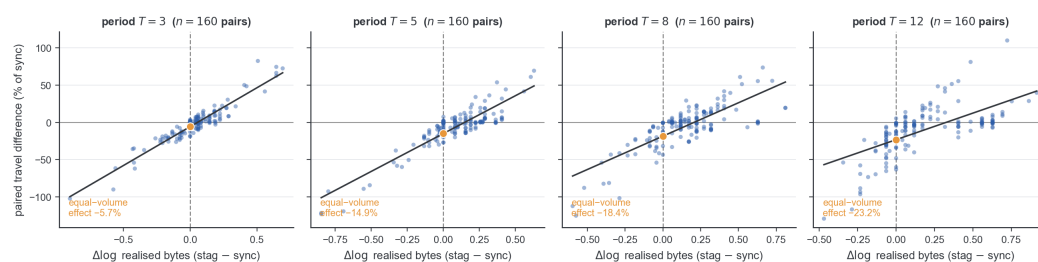
**Table 1.** Broadcast schedulers at a matched scarce budget ( $N = 8$ ,  $\approx 47$ – $50$  kB communication). Coverage and communication are matched; the desynchronised and event-triggered schedules cut travel and overlap. Lower travel/overlap/steps are better.

| Scheduler               | Coverage | Comm ( $10^3$ B) | Travel | Overlap | Steps |
|-------------------------|----------|------------------|--------|---------|-------|
| Periodic (synchronised) | 0.966    | 47               | 1231.2 | 0.378   | 17.7  |
| Random (desynchronised) | 0.966    | 48               | 1101.2 | 0.337   | 15.8  |
| Event-triggered         | 0.966    | 50               | 1124.1 | 0.340   | 16.2  |

To separate phase from broadcast count, content-adaptivity, and randomness, we run a decisive control. Comparing staggered against synchronised periodic broadcasting at the *same* interval (identical per-robot broadcast rate, the same number of transmissions per robot, differing only in phase) isolates synchronisation from broadcast count, content-adaptivity, and randomness. Staggered wins at every period, with the largest advantage at the scarcest budget tested (Table 2). The two arms are *not* byte-identical: because desynchronisation decorrelates beliefs, each staggered broadcast carries a larger map delta, so staggered in fact transmits *more* bytes (62.0 vs. 53.6 kB at period 12, a 15.7% surplus that itself grows as bandwidth tightens: +6.6%, +13.0%, +15.7% at periods 5, 8, 12). The gain is therefore not a bandwidth *saving*; desynchronisation spends more bytes yet travels less. The surplus could still mean more *delivered* information per broadcast; two controls address this. The byte-matched periodic-versus-random comparison of Table 1 ( $\approx 47$ – $48$  kB in both arms) reproduces the gain at matched realised volume, and the pooled ANCOVA of the team-size sweep below, which controls for log-communication volume, leaves the effect highly significant. More directly, adding the realised byte volume as a covariate to the paired contrasts (regressing each pair's travel difference on its log-volume difference) leaves the schedule effect negative and highly significant at every period: the equal-volume effect is  $-5.7\%$ ,  $-14.9\%$ ,  $-18.4\%$ , and  $-23.2\%$  of synchronised travel at periods 3, 5, 8, and 12 ( $n = 160$  pairs each, all  $p < 10^{-14}$ ), so the byte surplus does not account for the gain (Figure 4).

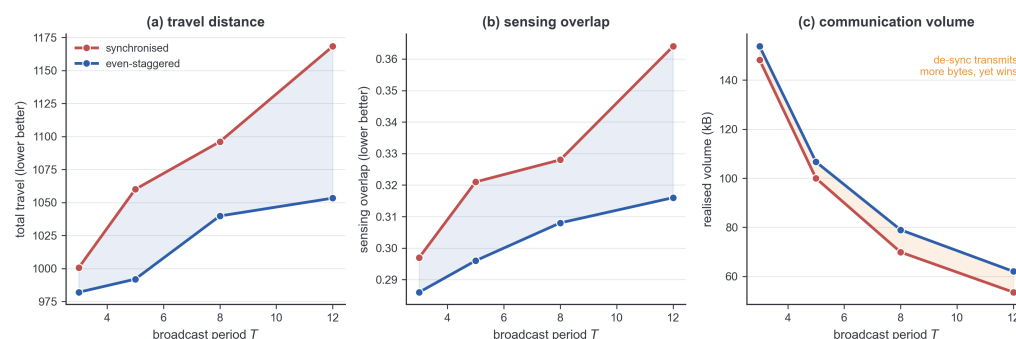
**Table 2.** Staggered vs. synchronised periodic at matched per-robot broadcast rate ( $N = 8$ ,  $n = 160$  paired episodes per period; negative = de-sync better; paired Wilcoxon). The last column gives the Holm-adjusted travel  $p$  across the three-period sweep; every period stays significant ( $p_{\text{Holm}} < 0.05$ ), including the marginal period 8. The realised byte volume is not held equal: staggered transmits up to 16% *more* (column 2), yet travels and overlaps less, so the gain is not a bandwidth saving.

| Period | Comm (kB) Sync $\rightarrow$ Staggered | Travel $\Delta$ | Overlap $\Delta$ | $p$ (Travel)         | $p_{\text{Holm}}$    |
|--------|----------------------------------------|-----------------|------------------|----------------------|----------------------|
| 5      | 100.0 $\rightarrow$ 106.6              | $-6.4\%$        | $-7.7\%$         | $1.5 \times 10^{-3}$ | $3.0 \times 10^{-3}$ |
| 8      | 69.9 $\rightarrow$ 79.0                | $-5.1\%$        | $-6.2\%$         | $4.5 \times 10^{-2}$ | $4.5 \times 10^{-2}$ |
| 12     | 53.6 $\rightarrow$ 62.0                | $-9.8\%$        | $-13.2\%$        | $1.4 \times 10^{-5}$ | $4.2 \times 10^{-5}$ |



**Figure 4.** The schedule effect is not a volume effect: each panel plots, for one broadcast period, the paired travel difference (staggered minus synchronised, as % of synchronised travel) against the paired difference in realised log byte volume over  $n = 160$  matched (map, seed) episodes. Realised bytes co-vary with travel (positive slopes), which is exactly the confound; the regression intercept at equal realised volume ( $\Delta \log \text{bytes} = 0$ , orange) remains negative and highly significant at every period ( $-5.7\%$  to  $-23.2\%$ , all  $p < 10^{-14}$ ), so the desynchronisation gain survives with byte volume held fixed. Grey dashed vertical lines and grey solid horizontal lines mark equal realised volume and zero travel difference, respectively; dark solid lines are least-squares fits, and orange points mark the fitted equal-volume intercepts.

Figure 5 summarises the phase-only sweep. At period 12, travel falls by 9.8% and overlap by 13.2% ( $p = 1.4 \times 10^{-5}$  and  $1.3 \times 10^{-5}$ ). Bootstrap 95% confidence intervals on the paired difference exclude zero at periods 5, 8, and 12 for both the travel and exploration steps. This is consistent with the byte-matched periodic-versus-random comparison of Table 1 ( $-10.6\%$  travel,  $p < 10^{-3}$ ). Desynchronisation also *accelerates* exploration. At a matched budget the team reaches target coverage in significantly fewer steps. Comparing staggered with synchronised, the bootstrap 95% CIs on the paired step difference exclude zero at periods 5, 8, and 12 (for instance,  $-1.6$  steps, CI  $[-2.4, -0.8]$  at period 12). The gain therefore lies in exploration time as well as travelled distance.



**Figure 5.** Synchronisation is the controllable cost at a matched per-robot broadcast rate ( $N = 8$ , periods 3–12). Desynchronised (blue) lies below synchronised (red) in both (a) total travel and (b) sensing overlap, with the largest gap at the scarcest budget; (c) desynchronisation transmits *more* bytes (up to 16% at period 12) yet travels less, so the gain is timing, not a bandwidth saving. The shaded regions show the between-schedule gap: the blue areas in (a,b) show the efficiency advantage of desynchronisation, and the orange area in (c) shows its additional realised byte volume.

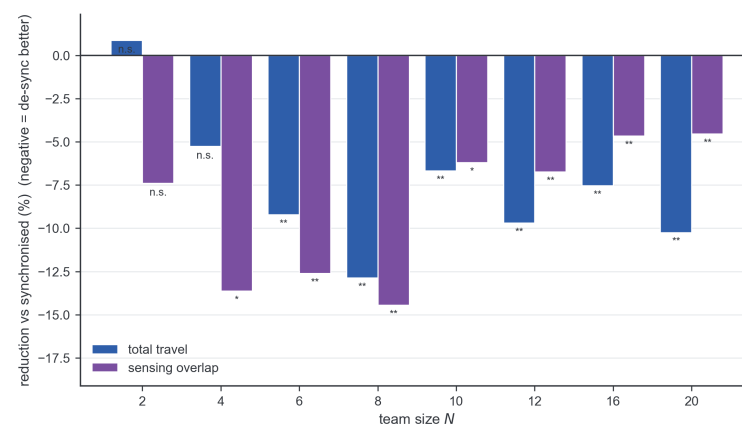
#### 4.2. Generalisation Across Team Size

Repeating the decisive phase-only control (staggered versus synchronised periodic at the same interval, hence matched per-robot broadcast rate) across team sizes from  $N = 2$  to  $N = 20$  confirms the advantage is not specific to  $N = 8$  (Table 3, Figure 6; interval 12,  $n \geq 102$  paired runs per  $N$ ). The desynchronisation travel reduction is significant ( $p < 0.05$ ) at every  $N \geq 6$ , reaching  $-12.9\%$  at  $N = 8$  ( $p = 1.6 \times 10^{-6}$ ); it is marginal at  $N = 4$  ( $-5.3\%$ ,  $p = 0.07$ ) and absent only for the  $N = 2$  dyad, where there is little teammate coordination to de-correlate. Sensing overlap falls at every  $N \geq 4$  (up to  $-14.4\%$ ). A pooled ANCOVA that additionally controls for log-communication volume and map difficulty

agrees (highly significant), and the effect emerges once team coupling is substantial (absent at  $N = 2$ , marginal at  $N = 4$ ) and is significant for every  $N \geq 6$  through  $N = 20$ . Among the desynchronised schedulers, the embedding-drift-triggered gate performs comparably to random staggering, consistent with synchronisation being the first-order effect, with the trigger choice a second-order one.

**Table 3.** Desynchronisation advantage across team size (phase-only control: staggered vs. synchronised periodic at interval 12, matched per-robot broadcast rate; paired Wilcoxon, negative is better). The travel effect is significant for every  $N \geq 6$ .

| $N$ | Travel $\Delta$ | Overlap $\Delta$ | $p$ (Travel)         | $n$ |
|-----|-----------------|------------------|----------------------|-----|
| 2   | +0.9%           | −7.4%            | 0.89                 | 120 |
| 4   | −5.3%           | −13.6%           | 0.07                 | 120 |
| 6   | −9.2%           | −12.6%           | $2 \times 10^{-4}$   | 120 |
| 8   | −12.9%          | −14.4%           | $1.6 \times 10^{-6}$ | 120 |
| 10  | −6.7%           | −6.2%            | $1 \times 10^{-3}$   | 120 |
| 12  | −9.7%           | −6.7%            | $1.8 \times 10^{-4}$ | 120 |
| 16  | −7.5%           | −4.7%            | $4 \times 10^{-4}$   | 111 |
| 20  | −10.2%          | −4.5%            | $2.5 \times 10^{-6}$ | 102 |



**Figure 6.** Desynchronisation advantage across team size ( $N = 2$ – $20$ ) from the phase-only control (staggered vs. synchronised periodic at matched budget, interval 12): total-travel (blue) and sensing-overlap (purple) reduction, negative is better. \*  $p < 0.05$ , \*\*  $p < 0.01$ ; n.s. = not significant.

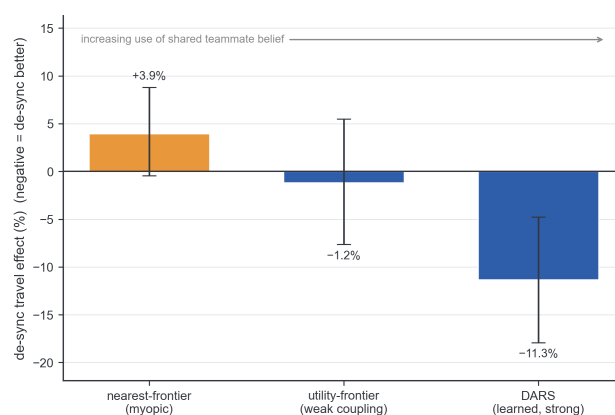
#### 4.3. The Synchrony Tax Appears Only When the Planner Uses the Shared Belief

If the synchrony tax is really about correlated stale *beliefs*, it should vanish on planners that barely use those beliefs. This gives a clean negative control. We ran the same phase-only control (staggered vs. synchronised periodic, interval 12,  $N = 8$ ) on two classical frontier explorers that reuse the identical environment, node graph, and communication layer: a *utility* frontier planner and a myopic *nearest*-frontier planner. The effect tracks how strongly each planner coordinates through the shared belief (Figure 7). On the learned graph-attention backbone (the privileged-learning explorer of [9], abbreviated DARS), staggering cuts travel by 11.3% ( $p = 2 \times 10^{-4}$ ); on the utility frontier it is only directional (−1.2%, not significant); and on the myopic nearest-frontier planner, which deconflicts only weakly on teammate state, the benefit disappears and a small apparent cost appears (+3.9% travel at interval 12,  $p = 0.011$  under the paired Wilcoxon; the bootstrap confidence intervals in Figure 7 are wider and exclude zero only for DARS). This negative control is informative rather than a failure to generalise: the staleness-to-overlap argument predicts the *disappearance* of the benefit when belief coupling is weak (it runs through the deconfliction assumption, Section 3.4); the small residual cost is not predicted by the bound, and we

read it as re-planning churn that asynchronous belief updates trigger in a myopic planner, a hypothesis the present experiments do not isolate. We therefore scope the finding to learned, belief-coupled coordination (Table 4).

**Table 4.** Desynchronisation effect by planner backbone (staggered vs. synchronised periodic,  $N = 8$ , interval 12; negative travel = de-sync better; paired Wilcoxon,  $n$  as shown; DARS and utility from the three-backbone run, nearest-frontier from the deconfliction pilot). The effect tracks how strongly each planner coordinates through the shared teammate belief. n.s. means not significant.

| Planner Backbone               | Belief Coupling | Travel $\Delta$ | $p$ (Travel)         | $n$ |
|--------------------------------|-----------------|-----------------|----------------------|-----|
| Nearest-frontier (myopic)      | Weak            | +3.9%           | 0.011                | 120 |
| Utility-frontier               | Moderate        | −1.2%           | 0.66 (n.s.)          | 100 |
| DARS (learned graph-attention) | Strong          | −11.3%          | $1.8 \times 10^{-4}$ | 100 |



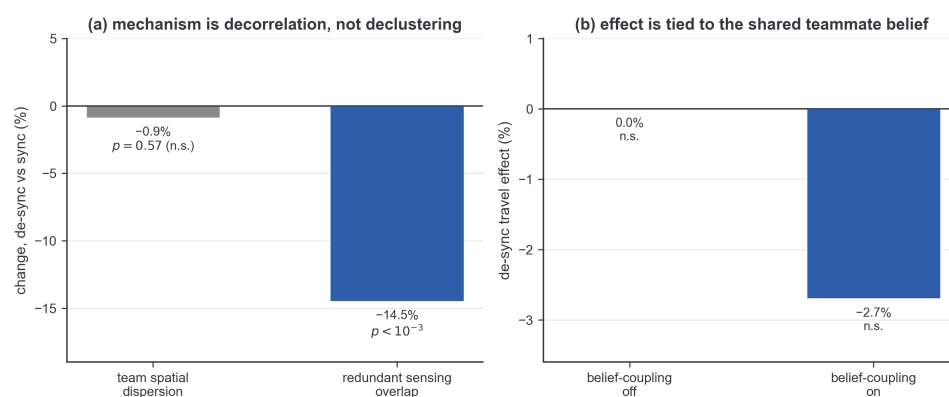
**Figure 7.** Desynchronisation effect by planner backbone (staggered vs. synchronised periodic, interval 12,  $N = 8$ ; negative means desynchronisation is better). The benefit grows with how much the planner coordinates through the shared teammate belief: harmful for the myopic nearest-frontier planner, weak for the utility frontier, and strong for the learned graph-attention policy. Error bars are 95% bootstrap confidence intervals (the DARS interval excludes zero; the others do not). The paired Wilcoxon test additionally detects a small but significant *harmful* effect for nearest-frontier (+3.9% travel,  $p = 0.011$ ); the bootstrap intervals shown here are wider and are reported separately from those exact  $p$ -values, which are given in the text.

#### 4.4. Mechanism: Desynchronisation Decorrelates Staleness

The staleness-to-overlap bound (Proposition 2) attributes overlap to the *peak* team staleness, not its mean. For a period- $T$  schedule, each link's staleness is a sawtooth, so the team-staleness sum has the *same* time mean, and the schedules issue the *same number of broadcasts*, under synchronised and staggered broadcasting; only the simultaneity differs. Evenly spaced staggering leaves the mean unchanged but roughly halves the peak (a 41–50% reduction across  $N = 4$ –12 at  $T = 12$ ); the measured sensing overlap falls accordingly (Figure 6). Desynchronisation therefore works by *decorrelating* staleness across robots, not by reducing it. Two further checks separate the peak from the mean. Within each period the mean team staleness is identical across the two schedules by construction, yet overlap falls (Table 2); and across the eight (schedule, period) conditions, jointly regressing mean overlap on the peak and the mean of the team-staleness sum attributes the variation to the peak ( $p = 0.001$ ), with no additional contribution from the mean ( $p = 0.16$ ), as reported below.

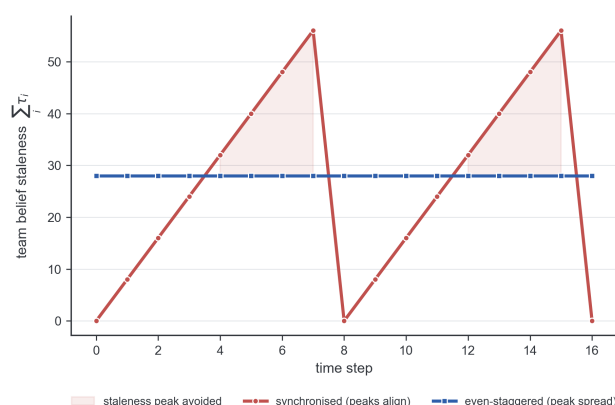
Two controls support this causal reading. First, when the planner is made to ignore the shared teammate belief entirely (the utility frontier with its deconfliction weight set to zero), the desynchronisation effect vanishes *exactly* (synchronised and staggered runs

are identical, 0.0%); restoring belief coupling restores the benefit only directionally ( $-2.7\%$  travel at weight 1, stable near  $-2.8\%$  up to weight 6,  $n = 160$  per weight,  $p = 0.56$ , not significant, consistent with this planner's weak coupling in Table 4). The causal force of this control rests on the exact null at zero coupling, with the dose direction as corroboration. Second, the benefit is not simple spatial declustering: team spatial dispersion is statistically unchanged between synchronised and staggered runs ( $p = 0.57$ ), while redundant *sensing* overlap falls sharply ( $-14.5\%$ ,  $p < 10^{-3}$ ), consistent with decorrelated stale beliefs reducing simultaneous redundant coverage rather than physically spreading the robots (Figure 8).



**Figure 8.** Two controls isolating the mechanism. (a) Desynchronisation leaves team spatial dispersion statistically unchanged ( $-0.9\%$ ,  $p = 0.57$ ) while sharply cutting redundant sensing overlap ( $-14.5\%$ ,  $p < 10^{-3}$ ): the gain comes from decorrelating stale beliefs, not from physically spreading the robots. (b) When the planner is made to ignore the shared teammate belief (deconfliction weight set to zero) the desynchronisation effect vanishes exactly ( $0.0\%$ ); restoring belief coupling restores it only directionally ( $-2.7\%$ , n.s.), so the causal force rests on the exact null at zero coupling. n.s. means not significant.

Figure 9 gives the time-domain view of this decorrelation.

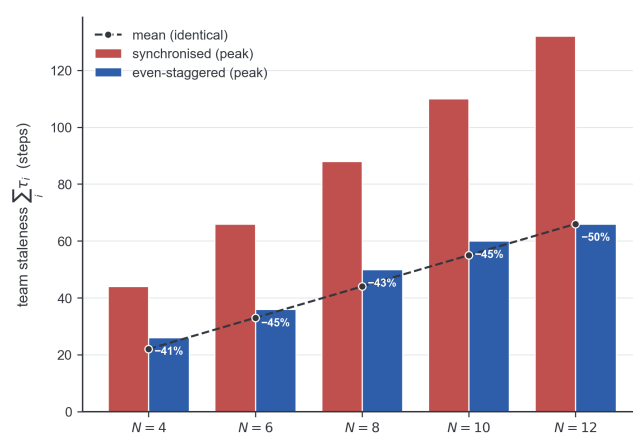


**Figure 9.** Time-domain view of the mechanism ( $N = 8$ , period  $T = 8$ ). Each link's staleness is a sawtooth; under synchronised broadcasting the sawtooths peak together so the team-staleness sum  $\sum_i \tau_i$  swings to a large peak (red), whereas even staggering spreads the offsets and flattens the sum toward its (identical) time mean (blue). The shaded area is the peak the staleness-to-overlap bound penalises and that Proposition 3 minimises.

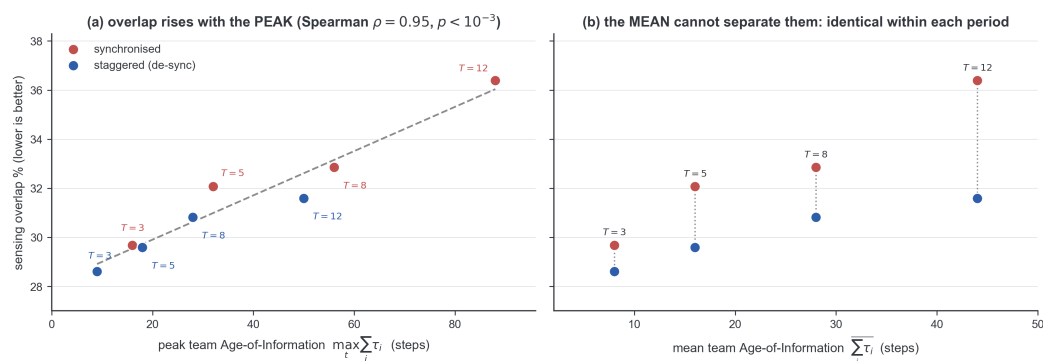
#### Empirical Support for Proposition 2 in the Age-of-Information Metric

The staleness-to-overlap bound is naturally read in the language of Age-of-Information, where the relevant cost is the *peak* of the team-staleness sum rather than its mean. We test the bound empirically in this metric. Figure 10 first shows that staggering reduces this peak while preserving the mean. Across eight broadcast conditions (synchronised

and staggered periodic at intervals 3, 5, 8, 12,  $N = 8$ ), the measured sensing overlap rises with the team's peak Age-of-Information, the peak of the team-staleness sum (Spearman  $\rho = 0.95$ ,  $p < 10^{-3}$ ; Figure 11). Synchronised schedules sit at the high-AoI end of this relation and staggered schedules at the low end, consistent with Proposition 2: the quantity the bound penalises is the peak team staleness, and overlap tracks it with measured data in the central metric of the Age-of-Information literature. Even staggering is, by Proposition 3, the peak-AoI-optimal fixed-rate periodic broadcast schedule; for symmetric broadcast this coincides with the AoI-optimal round-robin policy [10].



**Figure 10.** Desynchronisation decorrelates staleness at constant mean. For broadcast period  $T = 12$  the team-staleness sum  $\sum_i \tau_i$  has identical time mean (dashed) under synchronised and staggered schedules, but even staggering roughly halves its peak (percentages shown), the quantity the staleness-to-overlap bound penalises.



**Figure 11.** Empirical support for Proposition 2 in the Age-of-Information metric. Each point is one of eight  $N = 8$  broadcast conditions (synchronised and staggered periodic, intervals 3, 5, 8, 12). (a) Measured sensing overlap rises with the team's peak Age-of-Information, the peak of the team-staleness sum (Spearman  $\rho = 0.95$ ,  $p < 10^{-3}$ ); synchronised schedules occupy the high-AoI end and staggered schedules the low end. (b) The same overlap values plotted against the *mean* of the team-staleness sum: within each period the two schedules share one mean by construction (dotted verticals), yet their overlaps differ, so the mean cannot explain the separation of the peak.

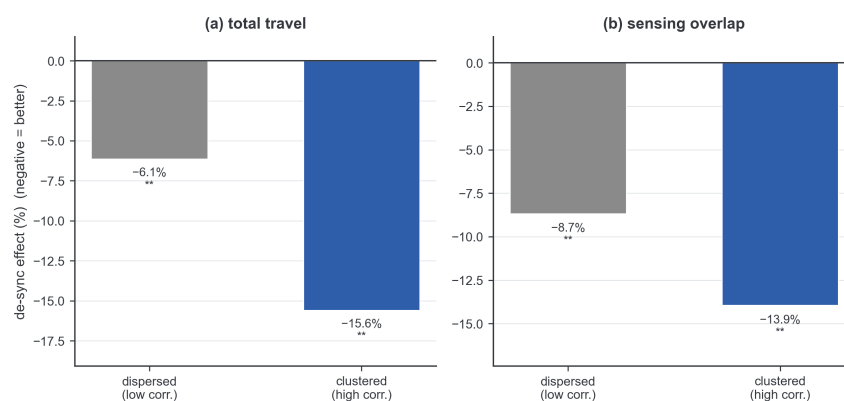
We also check Proposition 3 directly. Comparing synchronised broadcasting against a suboptimal half-split stagger and the evenly spaced stagger ( $N = 8$ , interval 12), the desynchronisation travel changes are ordered exactly as the theorem predicts: even staggering ( $-11.7\%$ ,  $p < 10^{-3}$ )  $<$  half-split staggering ( $-9.8\%$ ,  $p = 2 \times 10^{-4}$ )  $<$  synchronised broadcasting ( $0\%$ , baseline). Even staggering yields the largest benefit, confirming that the evenly spaced offsets are the bound-optimal schedule.

#### 4.5. The Benefit Tracks Source Correlation

If desynchronisation works by decorrelating staleness, its benefit should scale with how correlated the robots' information is to begin with. The correlated-source Age-of-Information theorem of [44] predicts exactly this: staggering correlated sources in phase helps more when the sources are more correlated. We instantiate that prediction for multi-robot exploration, with redundant sensing overlap as the downstream cost, by varying only the initial correlation of the team. Robots spawned as *clustered* start with highly correlated beliefs and sensing footprints; robots spawned as *dispersed* start with weakly correlated ones. At  $N = 8$  and interval 12, the desynchronisation travel benefit is  $-15.6\%$  ( $p < 10^{-3}$ ) for the clustered, high-correlation team but only  $-6.1\%$  ( $p = 3 \times 10^{-5}$ ) for the dispersed, low-correlation team (Figure 12). The ordering and its direction match [44]: the more correlated the sources, the more there is to gain from not firing them together (Table 5).

**Table 5.** The desynchronisation benefit grows with the team's initial information correlation ( $N = 8$ , interval 12, staggered vs. synchronised periodic; paired Wilcoxon,  $n = 160$ ). Clustered spawns start with highly correlated beliefs and sensing footprints, dispersed spawns with weakly correlated ones.

| Initial Correlation | Travel $\Delta$ | Overlap $\Delta$ | $p$ (Travel)       |
|---------------------|-----------------|------------------|--------------------|
| Dispersed (low)     | $-6.1\%$        | $-8.7\%$         | $3 \times 10^{-5}$ |
| Clustered (high)    | $-15.6\%$       | $-13.9\%$        | $<10^{-3}$         |



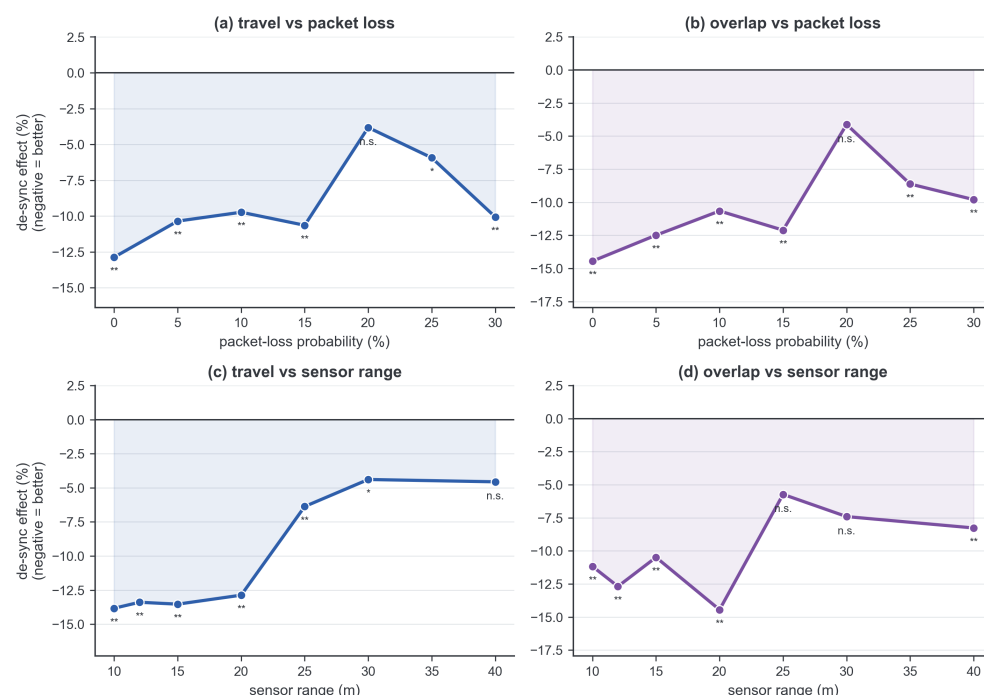
**Figure 12.** The desynchronisation benefit grows with the team's initial information correlation, as predicted by [44]. Both (a) total travel and (b) sensing overlap improve more for clustered, high-correlation spawns ( $-15.6\%$ ,  $-13.9\%$ ) than for dispersed, low-correlation ones ( $-6.1\%$ ,  $-8.7\%$ ) ( $N = 8$ , interval 12, staggered vs. synchronised periodic; \*\*  $p < 0.01$ ). More correlated sources gain more from desynchronisation.

This invites a correlation- or contention-aware scheduler that refreshes the more-correlated robots more often. We tested two and report honestly that neither significantly beat plain even staggering at matched communication. A static rule that separates spatially near robots in phase was, if anything, slightly worse ( $+3.6\%$  travel,  $p = 0.43$ ). The contention-weighted Whittle-index scheduler of Section 3 was directionally better on heterogeneous (dispersed) teams ( $-3.2\%$  travel at  $N = 8$ ,  $p = 0.45$ ) but coincided with even staggering for larger teams ( $+0.7\%$  at  $N = 16$ ,  $p = 0.08$ ) and even for teams with heterogeneous sensing (mixed 10/30 m ranges;  $-0.6\%$ ,  $p = 0.92$ ). Across homogeneous, dispersed, large, and heterogeneous sensing teams, the weighted scheduler never significantly beat even staggering. The reason is structural: for a homogeneously correlated team on a single shared medium, the bound-optimal even spacing of Proposition 3 is already (near-)optimal, leaving little for a weighted rule to exploit. The regime where heterogeneity genuinely pays off is instead the range-limited *interference* channel of Theorem 1: there, OW-Desync

provably and measurably improves communication freshness over round-robin (Section 4), while still reducing to even staggering when the team is symmetric. We therefore adopt even staggering as the simple, bound-optimal default for the symmetric medium and OW-Desync as its optimal generalisation under interference and heterogeneity.

#### 4.6. Robustness and Breadth

Two further experiments probe whether the desynchronisation benefit is an artefact of an idealised channel or a single sensing regime. Both use  $N = 8$ , interval 12, and contrast staggered against synchronised periodic scheduling under paired Wilcoxon tests ( $n \geq 116$  per cell); both are summarised in Figure 13.



**Figure 13.** Desynchronisation benefit across two stressors at  $N = 8$ , interval 12 (staggered vs. synchronised periodic;  $n \geq 116$  per point). **(a,b)** Travel and sensing overlap across seven packet-loss levels (0–30%): the gain persists, the lone shallow point at 20% being consistent with sampling variation flanked by significant neighbours. **(c,d)** Travel and overlap across seven sensor ranges (10–40 m): the gain holds across the sweep (the 40 m point marginal,  $p = 0.051$ ) and grows at shorter range, where robots rely more on the shared belief. Shaded areas between each curve and zero visualise the magnitude and direction of the schedule effect. \*  $p < 0.05$ , \*\*  $p < 0.01$ ; n.s. means not significant.

First, we inject independent per-packet erasure as a deliberately simple model of a lossy link; full WiFi contention, retransmission and fading are not modelled here and are deferred to the planned hardware study. We sweep seven loss levels from 0 to 30%. The travel benefit survives; it is  $-5.9$  to  $-12.9\%$  and significant at every level except 20% ( $-3.8\%$ ,  $p = 0.057$ ), where a shallow dip is flanked by significant reductions at 15% ( $-10.6\%$ ,  $p < 10^{-3}$ ) and 25% ( $-5.9\%$ ,  $p = 0.02$ ); the dip is therefore consistent with sampling variation rather than a genuine reversal. Sensing overlap improves in lockstep ( $-8.6$  to  $-14.5\%$ ). The free benefit is thus robust to independent packet loss rather than dependent on a perfect channel.

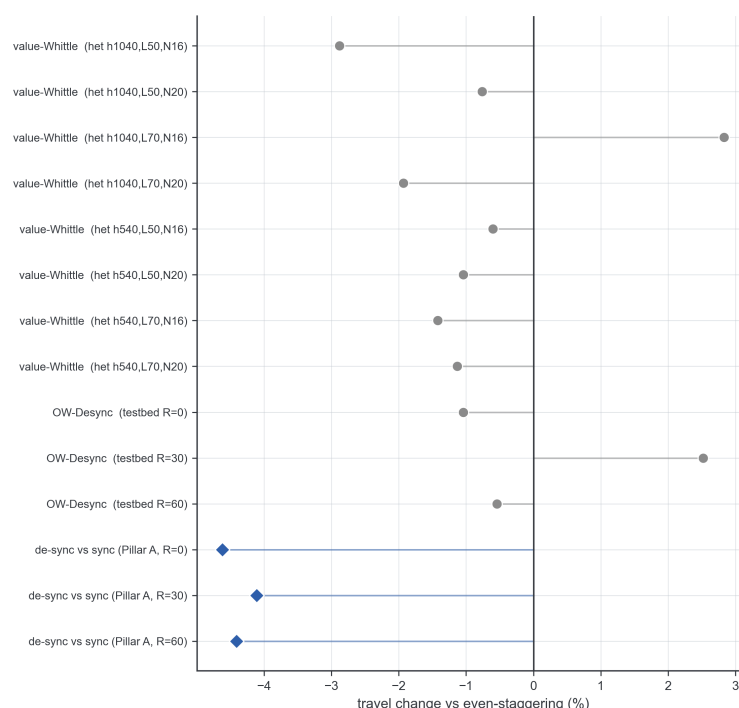
Second, we sweep seven sensor ranges from 10 to 40 m. The benefit holds across the sweep and grows at shorter range, where the team leans more on the shared belief: travel falls from  $-13.8\%$  at 10 m ( $p < 10^{-4}$ ) to  $-4.4$  to  $-4.6\%$  at 30–40 m (the 40 m point

marginal at  $p = 0.051$ ), and overlap by  $-7$  to  $-14\%$ . When each robot sees less on its own, decorrelating teammates' stale views matters more.

#### 4.7. From Even Staggering to OW-Desync

##### 4.7.1. Even Staggering Is Near-Optimal in the Symmetric Regime

We first ask whether a smarter schedule can beat plain even staggering on the exploration task. Across the contention-weighted Whittle, joint-submodular max-coverage, and value/loss-aware Whittle schedulers of Section 3, and a large sweep over broadcast rate ( $k = N/T \in [0.5, 4]$ ), team size ( $N = 4\text{--}24$ ), spatial overlap, per-robot value heterogeneity (up to  $64\times$ ), and packet loss (up to 0.7), *no* scheduler significantly beats even staggering on travel or overlap (e.g., the once-promising joint-submodular overlap effect,  $-2.9\%$  at  $n = 186$ ,  $p = 0.015$ , regresses to  $+0.09\%$  at  $n = 1262$ ,  $p = 0.73$ ). This null is exactly what correlated-source Age-of-Information near-optimality predicts [12,43]: in a symmetric single medium, uniform staggering is hard to beat, so we read the result as positive evidence for its near-optimality rather than a failure (Figure 14).

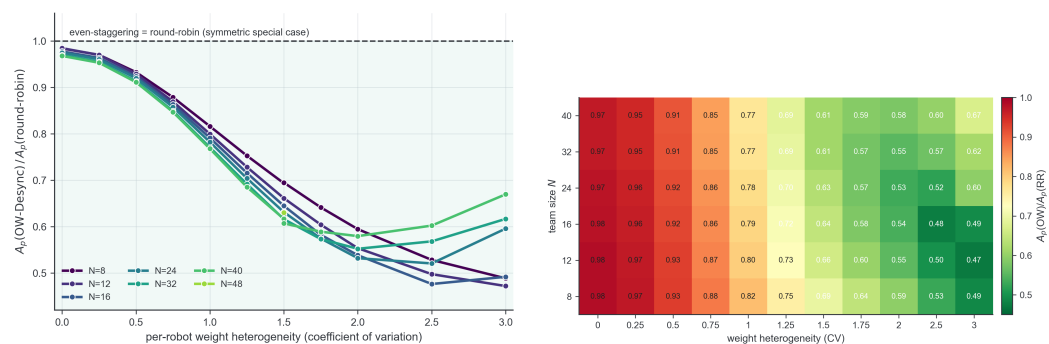


**Figure 14.** Effect sizes on the exploration objective, measured as travel change relative to even staggering (negative is better). The value/loss-aware Whittle scheduler across heterogeneity and loss regimes, and OW-Desync on the testbed, all cluster around zero (grey; none significantly beats even staggering), whereas the desynchronised-vs-synchronised contrast (blue diamonds) moves travel by  $-4$  to  $-4.6\%$ . On this objective, scheduling identity is near-optimal; only whether the team is synchronised matters.

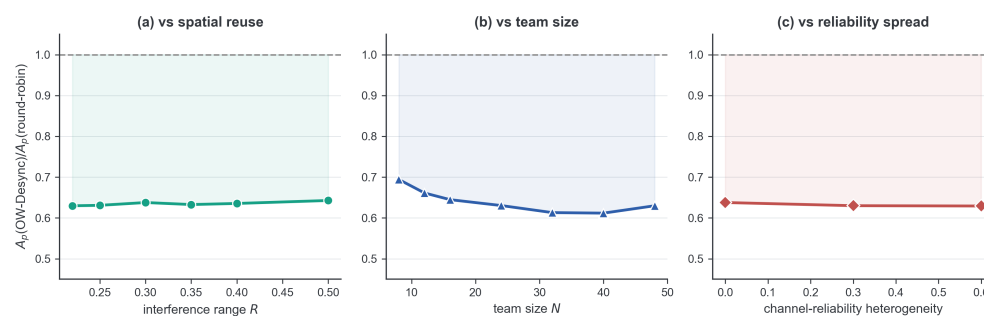
##### 4.7.2. OW-Desync Is Provably and Empirically Optimal on the Interference Channel

The picture changes on a range-limited interference graph with heterogeneous robots, where even staggering is no longer optimal (Theorem 1). Measuring the weighted peak Age-of-Information that OW-Desync optimises, the ratio  $A_p(\text{OW-Desync}) / A_p(\text{round-robin})$  equals 1.000 exactly in the fully symmetric complete-graph configuration of Corollary 1; stays at 0.97–0.98 at zero value heterogeneity on the range-limited graph (residual spatial-coupling differences already yield a small gain); and falls with heterogeneity: across  $N$  mean ratios 0.92, 0.79, 0.64, 0.56 as the weight coefficient of variation grows (0.5 to 2.0; every  $(N, CV)$  cell with  $CV > 0$  improves on average, with per-instance win fractions 0.80–1.00

over 500 seeds per cell,  $N = 8$ –48; Figure 15), with a deterministic, realisable activation schedule attaining the convex lower bound. The advantage is robust across team size ( $N = 8$ –48), spatial reuse, and heterogeneous channel reliability (Figure 16). OW-Desync is therefore Pareto-optimal in the precise sense that it is *optimal* on the communication-freshness axis it controls while *matching* the near-optimal even-staggering baseline on exploration. Practically, OW-Desync earns its coordination machinery once heterogeneity is substantial: in the sweeps of Figure 15, the freshness gain is 8% at a weight CV of 0.5 and 21–44% for a CV of 1–2, while near-homogeneous teams ( $CV \approx 0$ ) gain little and should simply stagger.



**Figure 15.** OW-Desync weighted peak-AoI relative to round-robin at matched budget on the interference graph. **(Left)** The ratio is close to 1 (0.97–0.98) at zero weight heterogeneity on this range-limited graph (Corollary 1 predicts exact coincidence on a symmetric shared medium) and falls as per-robot weight heterogeneity grows up to  $CV \approx 2$ , with a partial rebound for the largest teams beyond, for team sizes  $N = 8$ –48. **(Right)** The same gain as a heatmap over team size and heterogeneity, where every  $(N, CV)$  cell with  $CV > 0$  improves on average (greener is a larger OW-Desync advantage). The gain is on the communication-freshness objective OW-Desync provably optimises.

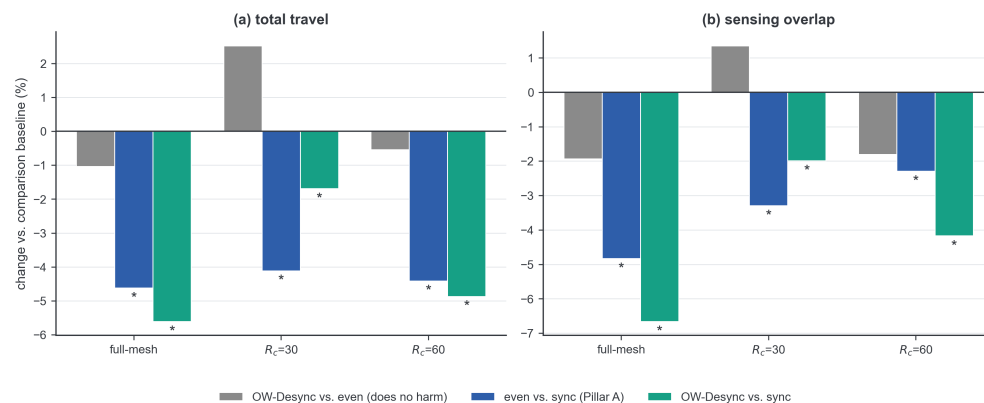


**Figure 16.** The OW-Desync weighted peak-AoI gain (ratio to round-robin at matched budget; lower is better, dashed line is parity) stays well below 1 across (a) interference range/spatial reuse, (b) team size  $N = 8$ –48, and (c) heterogeneous channel reliability, so the advantage is not an artefact of any single operating point. The shaded area between each curve and the parity line visualises the magnitude of the OW-Desync advantage.

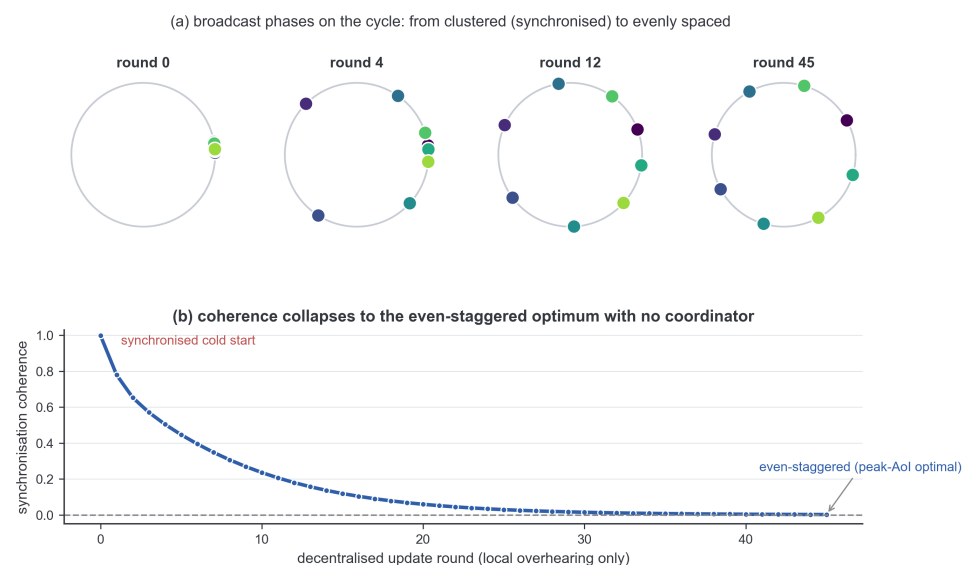
#### 4.7.3. On the Frozen Explorer, OW-Desync Preserves Exploration Efficiency

Running OW-Desync, even staggering, and synchronised periodic scheduling on the frozen explorer under the same range-limited channel ( $N = 16$ , heterogeneous sensing, paired over 231–240 map-seed pairs per cell), OW-Desync ties even staggering on exploration efficiency at every communication range tested (travel  $-1.0\%$  to  $+2.5\%$ , all  $p > 0.05$ ; coverage essentially unchanged, largest change  $-0.26\%$ ), i.e., it preserves exploration efficiency (Figure 17, Table 6). Both desynchronised schedules retain the synchrony-tax advantage over synchronised broadcasting even on the range-limited channel (even staggering vs. sync  $-4.1\%$  to  $-4.6\%$  travel,  $p < 0.005$ ; OW-Desync vs. sync up to  $-5.6\%$ ,

$p < 10^{-3}$ ). OW-Desync thus inherits the free desynchronisation benefit and adds provably optimal freshness under interference, without trading away exploration efficiency. Its activation frequencies are also reachable decentrally: a self-organising desynchronisation rule recovers a significant fraction of the benefit from a synchronised cold start with no coordinator (overlap  $-3.0\%$ ,  $p = 0.003$  at  $N = 16$ ; Figure 18 illustrates the convergence dynamics at  $N = 8$ ).



**Figure 17.** On the frozen explorer under a range-limited channel ( $N = 16$ , heterogeneous sensing, paired over 240 map–seed pairs per cell). OW-Desync ties even staggering on travel at every communication range (grey, all not significant), while both desynchronised schedules retain a significant advantage over synchronised broadcasting (blue, green; \* marks  $p < 0.05$ ), confirming that the synchrony-tax benefit persists on the interference-limited channel.



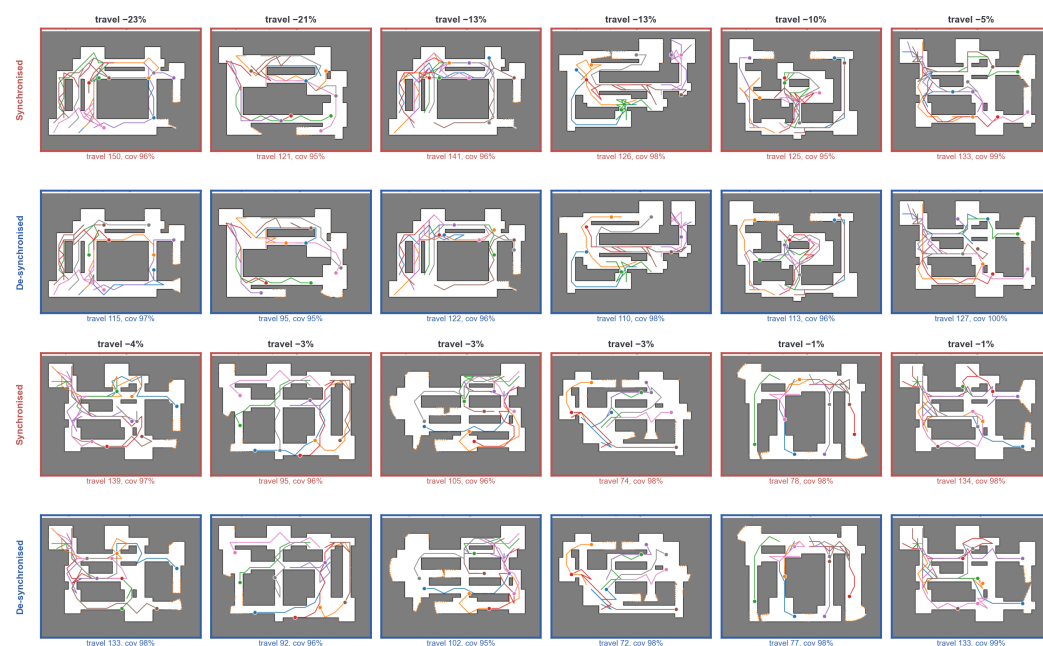
**Figure 18.** The even-staggered schedule is reachable with no coordinator. Starting from a fully synchronised cold start, a decentralised desynchronisation primitive in which each robot nudges its broadcast phase away from the neighbours it overhears (a) drives the phases to even spacing on the broadcast cycle and (b) collapses the synchronisation coherence to the even-staggered, peak-AoI-optimal configuration, using only local overhearing ( $N = 8$ ). In (a), the eight colours identify the eight robots; in (b), the blue curve is synchronisation coherence and the grey dashed line marks the even-staggered optimum.

**Table 6.** OW-Desync on the frozen explorer under a range-limited channel ( $N = 16$ , heterogeneous sensing, paired over 231–240 map–seed pairs per cell; travel change, paired Wilcoxon; \*  $p < 0.05$ , \*\*  $p < 0.01$ ); n.s. means not significant. OW-Desync ties even staggering (preserving exploration efficiency) while both desynchronised schedules keep the synchrony-tax advantage over synchronisation.

| Comm. Range | OW-Desync vs. Even | Even vs. Sync | OW-Desync vs. Sync |
|-------------|--------------------|---------------|--------------------|
| Full mesh   | −1.0% (n.s.)       | −4.6% **      | −5.6% **           |
| $R_c = 30$  | +2.5% (n.s.)       | −4.1% **      | −1.7% *            |
| $R_c = 60$  | −0.5% (n.s.)       | −4.4% **      | −4.9% **           |

#### 4.8. Qualitative Visualisation

Figure 19 contrasts synchronised and desynchronised broadcasting at *matched* bandwidth ( $N = 8$ , period  $T = 12$ ) across twelve episodes. Under synchronised broadcasting the robots refresh their shared belief simultaneously, briefly act on identical stale information, and herd toward the same frontiers, producing crossing and back-tracking trajectories. Even staggering spreads the belief updates in time, so the team fans out and covers the map with less redundant travel (−1 to −23% here, at essentially unchanged coverage). These episodes are selected illustrations of the mechanism, not a random sample; the aggregate effect over all maps and seeds is quantified by the statistics above.

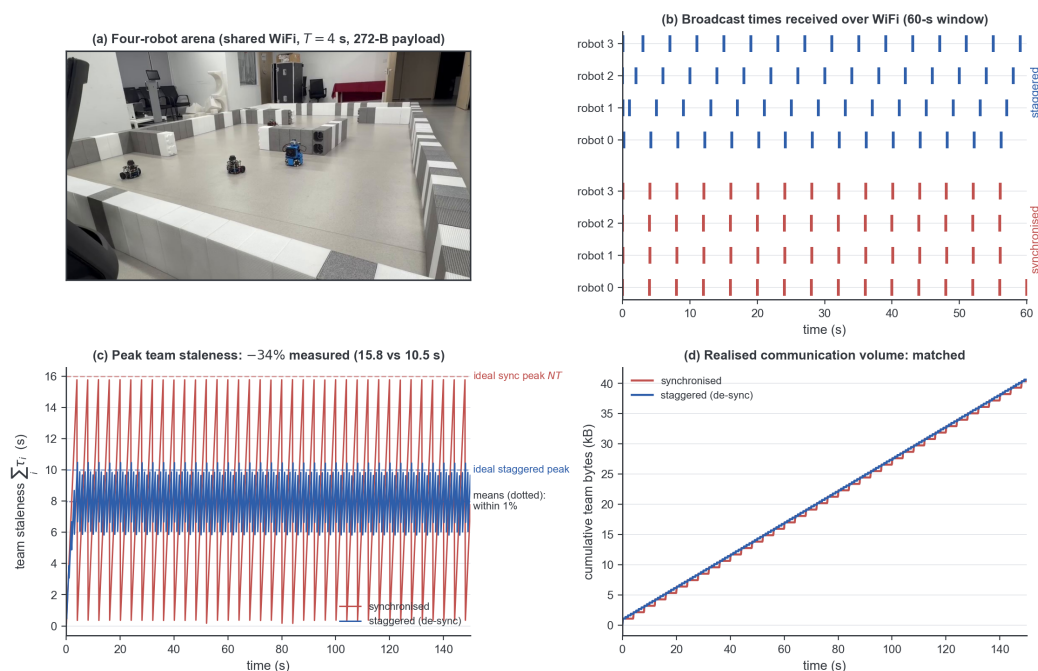


**Figure 19.** Synchronised (rows 1 and 3, red) vs. desynchronised/even-staggered (rows 2 and 4, blue) exploration at matched bandwidth ( $N = 8$ , broadcast period  $T = 12$ ); coloured curves are the eight robot trajectories, and the grey/white background is the occupancy map. Desynchronisation cuts herding and redundant travel; each column is one episode (a map and start-layout seed), ordered by the travel reduction printed above each column (−1% to −23%); all reduce travel at essentially unchanged coverage (within 0.3 percentage points, equal or higher in 8 of 12), selected from a 20-map  $\times$  3-seed sweep to illustrate the mechanism; aggregate statistics over all maps and seeds are reported above.

#### 4.9. Hardware Demonstration: The Timing-to-Staleness Stage on Real Robots

We deployed the scheduling layer on four differential-drive robots in a walled indoor arena (Figure 20a): a heterogeneous team of two LiDAR-inertial platforms (Mid-360 LiDAR; Livox Technology Company Limited, Hong Kong, China; FAST-LIO2 odometry) and two wheel-odometry Nav2 platforms, all running ROS2 Humble and sharing one WiFi network.

The scheduling layer of Section 3, implemented as a ROS2 communication node, runs onboard each robot and gates a 272-byte state broadcast (identifier, planar pose, and the 64-dimensional policy embedding) at a fixed period of  $T = 4.0$  s (20 ticks of the 0.2 s onboard control loop); a central monitor on the same network records every broadcast it receives, so all timing is measured on one clock after traversing the real channel. We ran one paired trial per condition in the same arena: *synchronised* (all robots on a common phase) versus *evenly staggered* (offsets  $T/N = 1$  s apart, i.e.,  $\{0, 1, 2, 3\}$  s for robots 0–3), with everything else identical.



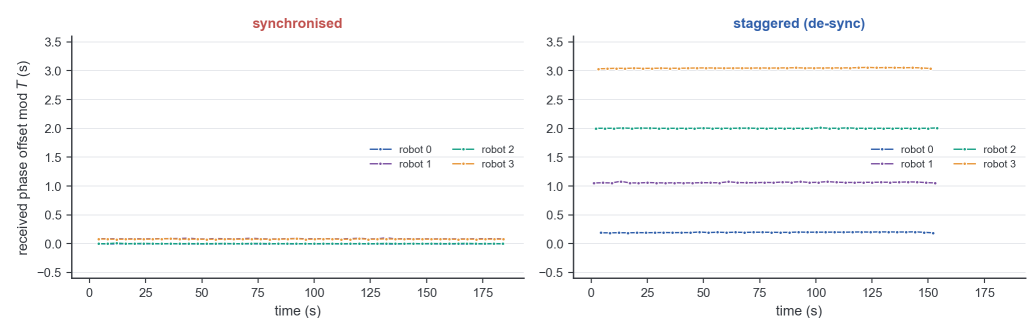
**Figure 20.** Hardware demonstration of the synchrony-tax mechanism (four robots, real WiFi, one paired trial per condition). **(a)** The walled arena with the four-robot team. **(b)** Broadcast times as received at the central monitor: synchronised (red) robots fire in lockstep; staggered (blue) robots realise even offsets one second apart. **(c)** Team staleness  $\sum_i \tau_i(t)$ : synchronisation drives the peak to the worst case  $NT$  while staggering holds it near the no-jitter staggered optimum  $T(N+1)/2$  (dashed), the worst-case-versus-optimum ordering that Proposition 3 formalises; the means (dotted) agree to within 1%, so the difference is purely in the peak. **(d)** Cumulative team communication volume: the two schedules transmit at the same rate with identical payloads, so the curves coincide and the staleness gap is attributable to timing alone.

Three things transfer from simulation to hardware (Figure 20, Table 7). First, the phase-only control is exact on a physical network: both conditions realise the same per-robot rate (0.0505 vs. 0.0506–0.0519 fires per tick, matched to within 3%) and the identical 272-byte payload, and the cumulative team volume curves coincide (Figure 20d). Second, the schedules survive the real channel: the received broadcast times show the synchronised robots aligned to within about 0.1 s and the staggered robots at offsets  $\{0.20, 1.06, 2.00, 3.05\}$  s, close to the ideal  $\{0, 1, 2, 3\}$  s, WiFi jitter included (Figure 20b). The schedules also hold over the run: each staggered offset stays within a  $\pm 15$  ms band around its median with drift below 5 ms per minute across the 150 s recording, so no long-run re-alignment was observed; if drift did accumulate on longer missions, the decentralised primitive of Figure 18 re-establishes even spacing online (Figure 21). Third, and centrally, the mechanism appears with the predicted magnitude: the measured peak team staleness is 15.8 s when synchronised versus 10.5 s when staggered, a 34% reduction. Both values sit close to the no-jitter ideals for the per-robot age sum,  $NT = 16$  s for synchronised and  $T(N+1)/2 = 10$  s for even staggering (deviations  $-1\%$  and  $+5\%$ ), reproducing on hardware the worst-case-versus-optimum

ordering that Proposition 3 formalises, while the mean staleness is essentially unchanged (8.0 vs. 7.9 s; unrounded 8.00 and 7.94 s, both within 1% of the schedule-invariant value  $NT/2 = 8$  s). On hardware, as in theory, synchronisation costs peak freshness and buys nothing in the mean. We emphasise the scope of this demonstration. It is a single paired run, so its numbers are measurements of one realisation rather than statistics, and it establishes only the first stage of the causal chain, from communication timing to peak team staleness, on a physical channel. The downstream stages (staleness to redundant sensing to travel) are quantified in simulation above; the five repeated paired trials reported below add statistical support for exactly this stage.

**Table 7.** Hardware demonstration: four robots, one trial per condition, all timing measured at a central monitor over real WiFi. Rates and payloads are matched; only the phase differs. Broadcast counts differ because the staggered recording is 30 s shorter. Ideals are the no-jitter values for  $N = 4$ ,  $T = 4$  s.

|                                   | Synchronised             | Staggered (De-Sync)            | Ideal              |
|-----------------------------------|--------------------------|--------------------------------|--------------------|
| Per-robot period (median)         | 4.00 s                   | 4.00 s                         | $T = 4$ s          |
| Per-robot fire rate (per tick)    | 0.0505                   | 0.0506–0.0519                  | 0.05               |
| Payload per broadcast             | 272 B                    | 272 B                          | Equal              |
| Broadcasts per robot              | 47                       | 39–40                          | Rate-matched       |
| Received phase offsets (mod $T$ ) | $\{0, 0.08, 0, 0.08\}$ s | $\{0.20, 1.06, 2.00, 3.05\}$ s | $\{0, 1, 2, 3\}$ s |
| Peak team staleness               | 15.8 s                   | 10.5 s (−34%)                  | 16 vs. 10 s        |
| Mean team staleness               | 8.0 s                    | 7.9 s                          | $NT/2 = 8$ s       |



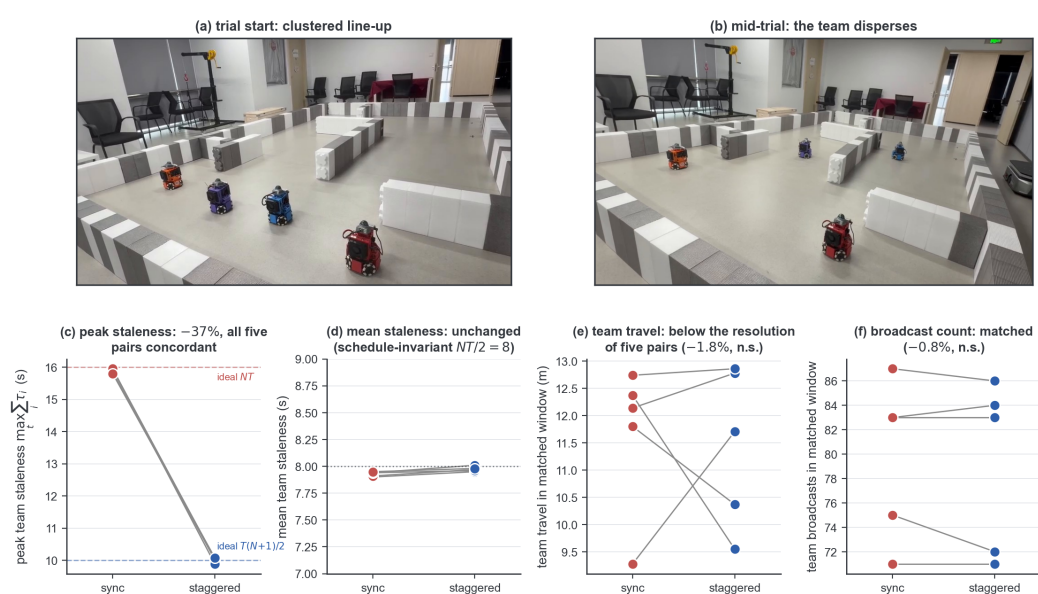
**Figure 21.** Long-run stability of the hardware schedules: received phase offsets (mod  $T$ ) of every broadcast over the full recordings. The synchronised robots stay aligned (**left**) and the staggered robots hold their assigned offsets (**right**) throughout, with per-robot drift below 5 ms per minute and offset bands within  $\pm 15$  ms, so asynchronous execution and WiFi jitter did not re-align the schedules over the mission; longer deployments can re-apply the decentralised primitive of Figure 18 online.

We then repeated the comparison as *five* paired trials with a second team: four identical mecanum-drive robots with onboard LiDAR-inertial odometry and Nav2 navigation, each broadcasting gated occupancy-map updates (tens of kilobytes per broadcast) at the same period  $T = 4.0$  s and logging every signal locally. Each pair ran synchronised (common phase) and evenly staggered (offsets  $\{0, 1, 2, 3\}$  s) in the same arena, and the two runs of a pair are analysed over a common matched window. The mechanism is replicated, now with statistics (Table 8, Figure 22): the measured peak team staleness falls from 15.9 to 10.0 s (−37%), with all five pairs concordant (one-sided exact Wilcoxon  $p = 0.031$ , the smallest value attainable at  $n = 5$ ; paired  $t$ -test  $p < 10^{-7}$ ) and every run within 1.5% of the no-jitter ideals  $NT = 16$  s and  $T(N + 1)/2 = 10$  s; the mean staleness stays within 0.1 s of the schedule-invariant  $NT/2 = 8$  s in all ten runs; and the broadcast counts are matched (−0.8%, n.s.). The timing-to-staleness stage of the mechanism therefore reproduces across two fleets and two payload regimes (272-byte state vectors and multi-kilobyte map updates) with statistical support. The trials also recorded task-level travel: five pairs resolve the

37% staleness effect decisively but are underpowered for a simulation-scale ( $\approx 10\%$ ) travel effect against between-run variability (matched-window team travel  $-1.8\%$ , n.s.); a power calculation indicates roughly twenty pairs would be needed, which we leave to future work.

**Table 8.** Five repeated paired hardware trials on the second fleet (matched analysis window  $W$  per pair; peak and mean of the team-staleness sum; broadcasts counted within  $W$ ). The peak falls by 37% in every pair while the mean and the broadcast count are unchanged. n.s. means not significant.

| Pair | $W$ (s) | Peak $\tau$ Sync (s) | Peak $\tau$ Stag. (s) | Mean $\tau$ Sync/Stag. (s) | Broadcasts Sync $\rightarrow$ Stag. |
|------|---------|----------------------|-----------------------|----------------------------|-------------------------------------|
| 1    | 85.6    | 15.84                | 10.02                 | 7.94/8.01                  | 83 $\rightarrow$ 84                 |
| 2    | 73.5    | 15.93                | 10.03                 | 7.90/7.95                  | 75 $\rightarrow$ 72                 |
| 3    | 86.7    | 15.94                | 10.06                 | 7.91/7.97                  | 87 $\rightarrow$ 86                 |
| 4    | 71.7    | 15.95                | 9.89                  | 7.94/7.96                  | 71 $\rightarrow$ 71                 |
| 5    | 84.2    | 15.80                | 10.08                 | 7.95/7.98                  | 83 $\rightarrow$ 83                 |



**Figure 22.** Five repeated paired hardware trials on the second fleet (matched analysis windows). (a) Trial start: the four robots begin in a clustered line-up. (b) Mid-trial: the team disperses through the arena. (c) Peak team staleness: all five pairs fall from the synchronised worst case to the staggered optimum ( $-37\%$ ; dashed lines are the no-jitter ideals  $NT$  and  $T(N+1)/2$ ). (d) Mean team staleness: within  $0.1$  s of the schedule-invariant  $NT/2 = 8$  s in every run. (e) Team travel over the matched window: below the resolution of five pairs ( $-1.8\%$ , n.s.; the simulation-scale  $10\%$  effect needs roughly twenty pairs). (f) Broadcast counts: the matched-rate control holds ( $-0.8\%$ , n.s.).

## 5. Discussion

The main takeaway is that under scarce communication a team pays a *synchrony tax*. When robots can update only intermittently, broadcasting them on a common schedule is the costly choice: their shared beliefs then go stale together, and the explorer acts on correlated, out-of-date information. The lever that removes this cost is timing, not volume. At a fixed nominal broadcast rate, and with desynchronised schedules in fact sending more bytes (up to  $16\%$  at the scarcest budget), simply staggering the phase removes a  $10$ – $13\%$  travel and overlap penalty paid by synchronised schedules, largest where bandwidth is scarcest. The mechanism is specific. Synchronisation does not worsen the *average* freshness of the shared belief; it worsens its *peak*, and our staleness-to-overlap bound ties that peak to redundant sensing. Even phase staggering provably minimises this peak on a symmetric medium, which is why it, and desynchronised scheduling in general, is hard to beat there.

The synchrony tax is a property of learned, belief-coupled coordination rather than a universal law. A cross-backbone control makes this concrete: the gain is strong on the learned graph-attention explorer, weak on a utility-frontier planner, and gone on a myopic nearest-frontier planner that barely deconflicts through teammate state (with a small apparent cost there that the bound does not predict). Read as a negative control, this disappearance supports the mechanism, a planner that does not act on the shared belief has no synchrony tax to remove, and it also bounds the claim: expect the benefit wherever a policy plans against a shared, communicated belief, and not where robots act independently.

OW-Desync turns this observation into a principled scheduling layer. On a symmetric shared medium, even staggering is already optimal, so there is nothing to add. The interesting regime is the realistic one, where robots have unequal informativeness and the channel is range-limited, so symmetry breaks and the question becomes how to *allocate* freshness across heterogeneous robots under interference. OW-Desync answers this by minimising the weighted expected peak age over the feasible activation polytope: it recovers even staggering as the symmetric special case and improves the freshness objective it is designed to optimise while preserving exploration efficiency on the frozen explorer.

## 6. Limitations and Future Work

The backbone is frozen, so what we contribute is a test-time scheduling layer rather than an end-to-end co-trained system. Because exploration coverage saturates under the budgets we study, the gains we measure are in efficiency (travel and sensing overlap) rather than in final coverage. The synchrony tax is also backbone-dependent: it is a property of learned, belief-coupled coordination, and a planner that does not act on the shared belief sees little or none of it. Finally, OW-Desync's advantage is on the communication-freshness (peak-age) objective it is built to optimise; on the exploration task it preserves efficiency rather than improving it further, because the frozen explorer is largely insensitive to which teammate is freshest. The analysis also presumes reasonably accurate self-localisation, an essentially static environment, and communication delays that are short relative to the mission: large localisation error corrupts the shared belief irrespective of its age, moving objects add a staleness source that no broadcast schedule can control, and delays comparable to the task duration would leave the stationary age model of Section 3.4 inapplicable. Whether coordination architectures beyond the three planners tested, for example, explicit task-allocation or market-based teams, pay a similar synchrony tax also remains untested.

Several extensions follow naturally. OW-Desync's activation frequencies are reachable decentrally from local overhearing, which we verify only in simulation here (the hardware demonstration assigns its offsets centrally), so a fully on-robot realisation is the obvious next step. Beyond deciding *when* to broadcast, choosing *which* teammates to refresh (targeting) is a complementary lever, and co-training the policy together with the scheduler could let the explorer exploit freshness it currently ignores. The four-robot WiFi experiments of Section 4.9 reproduce the timing-to-staleness stage on hardware with statistical support (five repeated paired trials on a second fleet, all concordant); task-level travel and overlap statistics on hardware need a larger campaign, roughly twenty pairs for a ten-percent travel effect by our power calculation, under genuinely range-limited, lossy links, which is the next step.

## 7. Conclusions

We studied communication scheduling for bandwidth-limited multi-robot exploration on a frozen learned belief-coupled policy. Empirically, through a decisive phase-only con-

trol that holds the per-robot broadcast rate fixed (desynchronisation in fact transmits up to 16% more bytes) and across team-size sweeps, and theoretically via a staleness  $\rightarrow$  overlap bound, we showed that for such a policy communication *synchronisation* is the controllable efficiency cost under scarce bandwidth, a *synchrony tax*: desynchronising broadcast timing recovers efficiency without relying on byte savings, and even simple phase staggering is provably bound-optimal in the symmetric single-medium regime, where an exhaustive scheduler sweep confirms no sophisticated alternative beats it. On a realistic range-limited interference channel with heterogeneous robots, we generalise even staggering to *OW-Desync*, a provably peak-age-optimal overlap-weighted activation policy that subsumes even staggering as its symmetric special case (Theorem 1) and strictly improves communication freshness under heterogeneity while preserving exploration efficiency on the frozen explorer; its schedule is also reachable decentrally with no coordinator. The effect is a property of learned belief-coupled coordination rather than a universal law, disappearing (with at most a small apparent cost) on planners that barely deconflict through the shared belief. Four-robot experiments over real WiFi show that the first stage of the mechanism, from communication timing to peak team staleness, survives a physical channel: at matched rate, even staggering cuts the measured peak by 34–37% across six paired runs on two robot fleets, with five repeated trials giving the stage statistical support (all pairs concordant, paired  $t$   $p < 10^{-7}$ ), close to the predicted optimum, with the mean essentially unchanged. Repeated paired hardware trials for task-level efficiency are the planned next step.

In one sentence: under scarce bandwidth, broadcast phase is a free coordination resource for belief-coupled multi-robot learning; its benefit has a single staleness mechanism, whose timing stage survives a real channel, appears exactly where coordination runs through the shared belief, and on a symmetric medium is already captured optimally by plain even staggering, with OW-Desync carrying it to the heterogeneous, interference-limited setting.

**Author Contributions:** Conceptualisation, R.Z. and H.Q.; methodology, R.Z. and H.Q.; software, H.Q.; validation, R.Z., H.Q., J.S., J.X. and Q.Z.; formal analysis, R.Z. and H.Q.; investigation—hardware experiments, J.S., J.X., Q.Z. and H.Q.; data curation, J.S., J.X. and Q.Z.; writing—original draft preparation, R.Z. and H.Q.; writing—review and editing, Q.Z. and F.Q.; visualisation, H.Q.; supervision, F.Q.; project administration, F.Q. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (U25A20489 and 62334006), the Beijing Natural Science Foundation (L253009), the National Science and Technology Major Project Fund of China (2025ZD0215600), the National Key Technologies R&D Program of China (2025YFF1500600), and the Fundamental Research Program of Shanxi Province (grant Nos. 202403021212166 and 202303021211153).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The simulation and hardware data and the analysis code supporting this study, together with our scheduling layer (the desynchronised schedulers and evaluation runners), are available from the corresponding author on reasonable request and will be released publicly upon publication. The work builds on the publicly available bandwidth-limited multi-robot exploration framework of [9]; the pretrained policy is used frozen and unmodified.

**Acknowledgments:** During the preparation of this manuscript, the authors used Claude (Anthropic) to assist with generating figure-plotting code. The authors have reviewed and edited all output and take full responsibility for the content of this publication.

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

## Appendix A. Proofs of the Theoretical Results

**Proof of Lemma 1.** Let  $s = t - \tau_{ij}(t)$  be the last delivery step, so  $\hat{p}_j^{(i)}(t) = p_i(s)$ . By the triangle inequality and the per-step bound,

$$\begin{aligned} e_{ij}(t) &= \left\| \sum_{u=s+1}^t (p_i(u) - p_i(u-1)) \right\| \leq \sum_{u=s+1}^t \|p_i(u) - p_i(u-1)\| \\ &\leq (t-s) v_{\max} = \tau_{ij}(t) v_{\max}. \end{aligned} \quad (\text{A1})$$

□

**Proof of Proposition 1.** Round-robin partitions  $\mathcal{L}$  into  $\lceil L/B \rceil$  groups of at most  $B$  links; group  $k$  is served at steps  $\equiv k \pmod{\lceil L/B \rceil}$ . Each link lies in exactly one group, so it is served once every  $\lceil L/B \rceil$  steps and its staleness never exceeds  $\lceil L/B \rceil - 1$ . □

**Proof of Proposition 2.** Only contending pairs ( $w_{ij} = 1$ ) can co-sense, so non-contending pairs contribute nothing. For a contending pair, by Assumption 1 the expected redundant-sensing contribution is non-decreasing in  $e_{ij}$ ; summing over pairs and bounding by a non-decreasing envelope  $\Phi$  of the total weighted belief error  $\sum_{ij} w_{ij} e_{ij}$  yields the first inequality. Applying Lemma 1 term-by-term ( $e_{ij} \leq \tau_{ij} v_{\max}$ ) and monotonicity of  $\Phi$  yields the second. □

**Proof of Proposition 3.** Fix a sender  $i$ . Under a period- $T$  schedule with offset  $\phi_i$ , link  $i \rightarrow j$  delivers exactly at the steps  $t \equiv \phi_i \pmod{T}$  (delivery depends only on when the sender broadcasts), so its staleness is the period- $T$  sawtooth  $\tau_{ij}(t) = (t - \phi_i) \bmod T \in \{0, 1, \dots, T-1\}$ , identical for every receiver  $j$ . Hence the team-staleness sum is the superposition of  $N$  unit-rate sawtooths,  $S(t) = \sum_{ij} w_{ij} \tau_{ij}(t) = \sum_k a_k ((t - \phi_k) \bmod T)$ , where  $a_k = \sum_{j \neq k} w_{kj} \geq 0$  is the (nonnegative) outgoing contending weight of sender  $k$ .

*Synchronised case.* If all offsets coincide,  $\phi_k \equiv \phi$ , then every sawtooth reaches its maximum  $T-1$  at the same step  $t \equiv \phi - 1 \pmod{T}$ , so  $\max_t S(t) = (T-1) \sum_k a_k$ , the largest value  $\max_t S(t)$  can ever take since each summand is bounded by  $a_k(T-1)$ . Thus synchronisation maximises the peak.

*Even spacing.* Take the symmetric, fully contending case  $a_k \equiv a$  (the worst case for overlap, all pairs near). Then  $S(t) = a \sum_k ((t - \phi_k) \bmod T)$ . As  $t$  advances one step every non-wrapping term increases by 1 and a term whose sawtooth wraps drops by  $T-1$ , so between wraps  $S$  climbs at rate  $N$  per step and its peak is attained immediately before some wrap, at an instant  $t \equiv \phi_m - 1 \pmod{T}$ . Evaluate  $S$  at all  $N$  pre-wrap instants and average. For distinct offsets, pairing the contributions of robots  $m$  and  $j$  across the two instants gives  $((\phi_m - \phi_j - 1) \bmod T) + ((\phi_j - \phi_m - 1) \bmod T) = T-2$ , so the average pre-wrap value equals  $a(N(T-1)/2 + (T-N)/2)$  for every assignment with distinct offsets, while coincident offsets only raise it (a coincident pair contributes  $2(T-1) > T-2$ ). The maximum is at least the average, so every schedule has  $\max_t S(t) \geq a(N(T-1)/2 + (T-N)/2)$ . When  $N$  divides  $T$  the evenly spaced assignment makes all  $N$  pre-wrap values equal, attaining this lower bound with equality, so it is exactly optimal; the synchronised assignment attains  $aN(T-1)$ , the largest value any schedule can reach. For general  $T \geq N$ , write  $\phi_k = \lfloor (k-1)T/N \rfloor$ . For  $m \neq k$  the offset difference  $\phi_m - \phi_k$  differs from  $(m-k)T/N$  by less than 1, so each pairwise term  $((\phi_m - \phi_k - 1) \bmod T)$  differs from its evenly spaced value by less than 1, and each pre-wrap value of  $S/a$  differs from the assignment-independent average  $N(T-1)/2 + (T-N)/2$  by less than  $N-1$ . The balanced peak therefore exceeds the universal lower bound by less than  $a(N-1)$ , a guarantee no schedule can beat by more than that margin; an exhaustive search over all offset assignments for  $N \leq 6$ ,  $T \leq 16$  in fact finds the balanced assignment exactly optimal in every case. When  $T < N$  distinct offsets are infeasible; the peak is then minimised

by balancing the senders across the  $T$  slots ( $\lfloor N/T \rfloor$  or  $\lceil N/T \rceil$  per slot), which the same assignment achieves, while synchronisation (all robots in one slot) remains the maximiser. Indeed an unbalanced assignment places at least one extra simultaneous reset in some slot and leaves a longer unopposed climb elsewhere, so balancing the slot loads minimises the maximum of the sawtooth superposition. The general weighted and interference-limited case does *not* reduce to even phase-spacing at a fixed rate (a heavy sender may need a different share of the medium), and is handled instead by Theorem 1, which optimises activation *frequencies* rather than phases.

Therefore the balanced assignment  $\phi_k = \lfloor (k-1)T/N \rfloor$  minimises  $\max_t \sum_{ij} w_{ij} \tau_{ij}(t)$ , and by the monotonicity of  $\Phi(\cdot)$  in Proposition 2 it minimises the overlap envelope. The evenly spaced staggering used throughout is thus the bound-optimal open-loop, zero-side-information schedule, and the synchronised periodic baseline is its worst case.  $\square$

**Proof of Theorem 1.** For any policy, let  $f_i$  be source  $i$ 's long-run *activation* (transmission-attempt) frequency; the achievable set of  $f$  is exactly  $\mathcal{P}$ , since each slot selects one independent set and successful delivery on link  $i$  occurs with probability  $\gamma_i$  when  $i$  transmits, while any point of  $\mathcal{P}$  is realised by a stationary policy that samples  $S$  with probability  $x_S$  (Birkhoff-type decomposition of  $f$  into independent-set indicators). The *expected* (time-average) peak age of source  $i$  under a stationary policy with activation frequency  $f_i$  and per-attempt success probability  $\gamma_i$  is, by a geometric renewal,  $1/(\gamma_i f_i)$ , so  $A_p = \sum_i w_i / (\gamma_i f_i)$ , a convex function of  $f$  on  $\mathcal{P}$ ; minimising it gives  $f^*$  and the optimal stationary policy. This instantiates the peak-age optimality of stationary policies on interference networks [11] with task weights  $w_i$  and an average per-slot budget.  $\square$

**Proof of Corollary 1.** For complete  $G$  only one robot transmits per slot, so  $\mathcal{P} = \{f \geq 0 : \sum_i f_i \leq \min(1, B)\}$ . The objective  $(w/\gamma) \sum_i 1/f_i$  is symmetric and strictly convex; by Jensen (or the KKT conditions) its minimiser over the simplex is uniform,  $f_i^* = \min(1, B)/N$ , which is round-robin with even offsets  $\phi_k = \lfloor (k-1)T/N \rfloor$ . Uniform marginals fix only the *expected* peak age; among the schedules that realise them, the even offsets additionally minimise the worst-case peak by Proposition 3.  $\square$

**Proof of Proposition 4.** On the simplex  $\sum_i f_i = B$  the stationarity condition  $\partial_{f_i} \sum_i (w_i/\gamma_i)/f_i = \lambda$  gives  $f_i^* \propto \sqrt{w_i/\gamma_i}$ , which is non-uniform exactly when the  $w_i/\gamma_i$  differ. Since  $\sum_i c_i/f_i$  (with  $c_i = w_i/\gamma_i$ ) is strictly convex and  $f^{\text{RR}} = B/N \mathbf{1}$  is feasible but not stationary,  $A_p(f^*) < A_p(f^{\text{RR}})$ ; substituting the two allocations,  $A_p(f^{\text{RR}})/A_p(f^*) = (\sum_i c_i)/(\sum_i \sqrt{c_i})^2 \cdot N$ , which increases with the spread of  $\{c_i\}$  (equality iff all  $c_i$  equal, by Cauchy–Schwarz). When graph constraints beyond the total budget bind, this closed form no longer applies and the optimum is the convex program of Theorem 1.  $\square$

**Proof of Corollary 2.** Aggregate the pair weights per sender,  $w_i = \sum_{j \neq i} w_{ij}$ , and bound each  $\tau_{ij}(t)$  by sender  $i$ 's stationary peak age  $\tau_i = 1/(\gamma_i f_i)$ ; then  $\sum_{ij} w_{ij} \tau_{ij} \leq \sum_i w_i \tau_i = A_p$  and, by Proposition 2 and the monotonicity of  $\Phi$ ,  $\mathbb{E}[\text{overlap}] \leq \Phi(v_{\max} A_p)$  is minimised exactly when  $A_p$  is, i.e., at  $f^*$ .  $\square$

## References

1. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In *Proceedings of the Robotics: Science and Systems (RSS)*; MIT Press: Cambridge, MA, USA, 2014. [CrossRef]
2. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [CrossRef]
3. Niroui, F.; Zhang, K.; Kashino, Z.; Nejat, G. Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments. *IEEE Robot. Autom. Lett.* **2019**, *4*, 610–617. [CrossRef]

4. Burgard, W.; Moors, M.; Stachniss, C.; Schneider, F.E. Coordinated Multi-Robot Exploration. *IEEE Trans. Robot.* **2005**, *21*, 376–386. [\[CrossRef\]](#)
5. Lajoie, P.Y.; Ramtoula, B.; Chang, Y.; Carlone, L.; Beltrame, G. DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1656–1663. [\[CrossRef\]](#)
6. Tian, Y.; Chang, Y.; Herrera Arias, F.; Nieto-Granda, C.; How, J.P.; Carlone, L. Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems. *IEEE Trans. Robot.* **2022**, *38*, 2022–2038. [\[CrossRef\]](#)
7. Que, H.; Gao, H.; Shan, W.; Yang, X.; Zhao, R. A Lightweight, Centralized, Collaborative, Truncated Signed Distance Function-Based Dense Simultaneous Localization and Mapping System for Multiple Mobile Vehicles. *Sensors* **2024**, *24*, 7297. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Cao, Y.; Hou, T.; Wang, Y.; Yi, X.; Sartoretti, G. ARiADNE: A Reinforcement Learning Approach Using Attention-Based Deep Networks for Exploration. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2023; pp. 10219–10225. [\[CrossRef\]](#)
9. Ma, Y.; Liang, J.; Cao, Y.; Tan, D.M.S.; Sartoretti, G. Privileged Reinforcement and Communication Learning for Distributed, Bandwidth-Limited Multi-Robot Exploration. In *Proceedings of the 17th International Symposium on Distributed Autonomous Robotic Systems (DARS 2024)*; Springer: Cham, Switzerland, 2026; Volume 34, pp. 337–350. [\[CrossRef\]](#)
10. Kadota, I.; Sinha, A.; Uysal-Biyikoglu, E.; Singh, R.; Modiano, E. Scheduling Policies for Minimizing Age of Information in Broadcast Wireless Networks. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2637–2650. [\[CrossRef\]](#)
11. Talak, R.; Karaman, S.; Modiano, E. Optimizing Information Freshness in Wireless Networks under General Interference Constraints. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*; ACM: New York, NY, USA, 2018. [\[CrossRef\]](#)
12. Ramakanth, R.V.; Tripathi, V.; Modiano, E. Monitoring Correlated Sources: AoI-based Scheduling is Nearly Optimal. *arXiv* **2023**, arXiv:2312.16813.
13. Degesys, J.; Rose, I.; Patel, A.; Nagpal, R. DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*; ACM: New York, NY, USA, 2007; pp. 11–20. [\[CrossRef\]](#)
14. Yamauchi, B. A Frontier-Based Approach for Autonomous Exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*; IEEE: Piscataway, NJ, USA, 1997; pp. 146–151. [\[CrossRef\]](#)
15. Umari, H.; Mukhopadhyay, S. Autonomous Robotic Exploration Based on Multiple Rapidly-exploring Randomized Trees. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1396–1402. [\[CrossRef\]](#)
16. Xu, Y.; Yu, J.; Tang, J.; Qiu, J.; Wang, J.; Shen, Y.; Wang, Y.; Yang, H. Explore-Bench: Data Sets, Metrics and Evaluations for Frontier-based and Deep-reinforcement-learning-based Autonomous Exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2022; pp. 6225–6231. [\[CrossRef\]](#)
17. Gao, H.; Que, H.; Li, K.; Shan, W.; Liu, M.; Zhao, R.; Mu, L.; Yang, X.; Wei, Q.; Qiao, F. Mapping at First Sense: A Lightweight Neural Network-Based Indoor Structures Prediction Method for Robot Autonomous Exploration. In *Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–10.
18. Gao, H.; Que, H.; Au, H.; Shan, W.; Liu, M.; Qin, Y.; Mu, L.; Zhao, R.; Yang, X.; Wei, Q.; et al. SenseExpo: Efficient Autonomous Exploration with Prediction Information from Lightweight Neural Networks. *arXiv* **2025**, arXiv:2503.16000.
19. Yu, C.; Yang, X.; Gao, J.; Chen, J.; Li, Y.; Liu, J.; Xiang, Y.; Huang, R.; Yang, H.; Wu, Y.; et al. Asynchronous Multi-Agent Reinforcement Learning for Efficient Real-Time Multi-Robot Cooperative Exploration. In *Proceedings of the 22nd International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*; IFAAMAS: Richland, SC, USA, 2023; pp. 1107–1115.
20. Hollinger, G.A.; Singh, S. Multirobot Coordination with Periodic Connectivity: Theory and Experiments. *IEEE Trans. Robot.* **2012**, *28*, 967–973. [\[CrossRef\]](#)
21. Julian, B.J.; Angermann, M.; Schwager, M.; Rus, D. Distributed Robotic Sensor Networks: An Information-Theoretic Approach. *Int. J. Robot. Res.* **2012**, *31*, 1134–1154. [\[CrossRef\]](#)
22. Zhou, B.; Xu, H.; Shen, S. RACER: Rapid Collaborative Exploration with a Decentralized Multi-UAV System. *IEEE Trans. Robot.* **2023**, *39*, 1816–1835. [\[CrossRef\]](#)
23. Amigoni, F.; Banfi, J.; Basilico, N. Multirobot Exploration of Communication-Restricted Environments: A Survey. *IEEE Intell. Syst.* **2017**, *32*, 48–57. [\[CrossRef\]](#)
24. Chen, L.; Ye, S.; Sun, C.; Zhang, A.; Deng, G.; Liao, T. Optimized Foothold Planning and Posture Searching for Energy-Efficient Quadruped Locomotion over Challenging Terrains. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2020; pp. 399–405. [\[CrossRef\]](#)
25. da Silva, A.R.; Chaimowicz, L. Intermittent Rendezvous Plans with Mixed Integer Linear Program for Large-Scale Multi-Robot Exploration. *arXiv* **2025**, arXiv:2511.12237.

26. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 4758–4765. [\[CrossRef\]](#)
27. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 5135–5142. [\[CrossRef\]](#)
28. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2016; pp. 1271–1278. [\[CrossRef\]](#)
29. Lin, J.; Zhang, F. Loam Livox: A Fast, Robust, High-Precision LiDAR Odometry and Mapping Package for LiDARs of Small FoV. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2020; pp. 3126–3131. [\[CrossRef\]](#)
30. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot Operating System 2: Design, Architecture, and Uses in the Wild. *Sci. Robot.* **2022**, *7*, eabm6074. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Macenski, S.; Martín, F.; White, R.; Clavero, J.G. The Marathon 2: A Navigation System. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2020; pp. 2718–2725. [\[CrossRef\]](#)
32. Kaul, S.; Yates, R.; Gruteser, M. Real-Time Status: How Often Should One Update? In *Proceedings of the 2012 Proceedings IEEE INFOCOM*; IEEE: Piscataway, NJ, USA, 2012; pp. 2731–2735. [\[CrossRef\]](#)
33. Sun, Y.; Uysal-Biyikoglu, E.; Yates, R.D.; Koksai, C.E.; Shroff, N.B. Update or Wait: How to Keep Your Data Fresh. *IEEE Trans. Inf. Theory* **2017**, *63*, 7492–7508. [\[CrossRef\]](#)
34. Yates, R.D.; Sun, Y.; Brown, D.R., III; Kaul, S.K.; Modiano, E.; Ulukus, S. Age of Information: An Introduction and Survey. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 1183–1210. [\[CrossRef\]](#)
35. Hsu, Y.P. Age of Information: Whittle Index for Scheduling Stochastic Arrivals. In *Proceedings of the 2018 IEEE International Symposium on Information Theory (ISIT)*; IEEE: Piscataway, NJ, USA, 2018; pp. 2634–2638. [\[CrossRef\]](#)
36. Howard, R.A. Information Value Theory. *IEEE Trans. Syst. Sci. Cybern.* **1966**, *2*, 22–26. [\[CrossRef\]](#)
37. Ayan, O.; Vilgelm, M.; Klügel, M.; Hirche, S.; Kellerer, W. Age-of-Information vs. Value-of-Information Scheduling for Cellular Networked Control Systems. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs)*; IEEE: Piscataway, NJ, USA, 2019; pp. 109–117. [\[CrossRef\]](#)
38. Åström, K.J.; Bernhardsson, B.M. Comparison of Riemann and Lebesgue Sampling for First Order Stochastic Systems. In *Proceedings of the 41st IEEE Conference on Decision and Control (CDC)*; IEEE: Piscataway, NJ, USA, 2002; Volume 2, pp. 2011–2016. [\[CrossRef\]](#)
39. Tabuada, P. Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks. *IEEE Trans. Autom. Control* **2007**, *52*, 1680–1685. [\[CrossRef\]](#)
40. Heemels, W.P.M.H.; Johansson, K.H.; Tabuada, P. An Introduction to Event-Triggered and Self-Triggered Control. In *Proceedings of the 2012 IEEE 51st Conference on Decision and Control (CDC)*; IEEE: Piscataway, NJ, USA, 2012; pp. 3270–3285. [\[CrossRef\]](#)
41. Dimarogonas, D.V.; Frazzoli, E.; Johansson, K.H. Distributed Event-Triggered Control for Multi-Agent Systems. *IEEE Trans. Autom. Control* **2012**, *57*, 1291–1297. [\[CrossRef\]](#)
42. Girard, A. Dynamic Triggering Mechanisms for Event-Triggered Control. *IEEE Trans. Autom. Control* **2015**, *60*, 1992–1997. [\[CrossRef\]](#)
43. Jiang, Z. Round-Robin is Provably Near-Optimal for Minimizing Age with HARQ over Heterogeneous Unreliable Multiaccess Channels. In *2020 32nd International Teletraffic Congress (ITC 32)*; IEEE: Piscataway, NJ, USA, 2020; pp. 46–54. [\[CrossRef\]](#)
44. Shisher, M.K.C.; Tripathi, V.; Chiang, M.; Brinton, C.G. AoI-based Scheduling of Correlated Sources for Timely Inference. *arXiv* **2025**, arXiv:2509.01926.
45. Quinn, S.; Molloy, S.; Siljak, H. Decentralised, Self-Organising Drone Swarms using Coupled Oscillators. *arXiv* **2025**, arXiv:2505.00442.
46. Foerster, J.N.; Assael, Y.M.; de Freitas, N.; Whiteson, S. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29, pp. 2137–2145.
47. Sukhbaatar, S.; Szlam, A.; Fergus, R. Learning Multiagent Communication with Backpropagation. In *Proceedings of the Advances in Neural Information Processing Systems 29 (NIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 2244–2252.
48. Peng, P.; Wen, Y.; Yang, Y.; Yuan, Q.; Tang, Z.; Long, H.; Wang, J. Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games. *arXiv* **2017**, arXiv:1703.10069.
49. Jiang, J.; Lu, Z. Learning Attentional Communication for Multi-Agent Cooperation. In *Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2018; pp. 7265–7275.

50. Ding, Z.; Huang, T.; Lu, Z. Learning Individually Inferred Communication for Multi-Agent Cooperation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 22069–22079.
51. Niu, Y.; Paleja, R.; Gombolay, M. Multi-Agent Graph-Attention Communication and Teaming. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*; IFAAMAS: Richland, SC, USA, 2021; pp. 964–973.
52. Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; Pineau, J. TarMAC: Targeted Multi-Agent Communication. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*; JMLR: New York, NY, USA, 2019; Volume 97, pp. 1538–1546.
53. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 6379–6390.
54. Rashid, T.; Samvelyan, M.; Schroeder de Witt, C.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*; JMLR: New York, NY, USA, 2018; Volume 80, pp. 4295–4304.
55. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 5998–6008.
56. Kim, D.; Moon, S.; Hostallero, D.; Kang, W.J.; Lee, T.; Son, K.; Yi, Y. Learning to Schedule Communication in Multi-agent Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
57. Singh, A.; Jain, T.; Sukhbaatar, S. Learning when to Communicate at Scale in Multiagent Cooperative and Competitive Tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
58. Liu, Y.C.; Tian, J.; Glaser, N.; Kira, Z. When2com: Multi-Agent Perception via Communication Graph Grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2020; pp. 4105–4114. [[CrossRef](#)]
59. Wang, T.; Wang, J.; Zheng, C.; Zhang, C. Learning Nearly Decomposable Value Functions Via Communication Minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*; Curran Associates, Inc.: Red Hook, NY, USA, 2020.
60. Wang, R.; He, X.; Yu, R.; Qiu, W.; An, B.; Rabinovich, Z. Learning Efficient Multi-agent Communication: An Information Bottleneck Approach. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*; JMLR: New York, NY, USA, 2020; pp. 9908–9918.
61. Hu, G.; Zhu, Y.; Zhao, D.; Zhao, M.; Hao, J. Event-Triggered Multi-Agent Reinforcement Learning with Communication under Limited-Bandwidth Constraint. *arXiv* **2020**, arXiv:2010.04978.
62. Zhang, Q.; Sun, Z.; Zhang, Y.; Yu, Z.; Guo, B.; Zhang, J. VIL2C: Value-of-Information Aware Low-Latency Communication for Multi-Agent Reinforcement Learning. *arXiv* **2025**, arXiv:2511.19146.
63. Han, S.; Dastani, M.; Wang, S. Model-Based Sparse Communication in Multi-Agent Reinforcement Learning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*; IFAAMAS: Richland, SC, USA, 2023; pp. 439–447.
64. Zhu, C.; Dastani, M.; Wang, S. A Survey of Multi-Agent Deep Reinforcement Learning with Communication. *Auton. Agents Multi-Agent Syst.* **2024**, *38*, 4. [[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.