

Trustworthy Swarm Intelligence for Green IoT: Communication-Efficient and Safe Multi-Agent Synchronization across the Cloud-Edge Continuum

Bindiya Jain

Poornima University

Ravikant Gupta

`ravikant.gupta@jaipur.manipal.edu`

Manipal University Jaipur

Hemant K Upadhyay

Poornima University

Research Article

Keywords: Swarm Intelligence, Green IoT, Event-Triggered Control, Control Barrier Functions, Cloud-Edge Continuum, Multi-Agent Synchronization

Posted Date: October 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11241701/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Trustworthy Swarm Intelligence for Green IoT: Communication-Efficient and Safe Multi-Agent Synchronization across the Cloud-Edge Continuum

Bindiya Jain¹

¹Department of Computer Science and Engineering,
Poornima University, Jaipur, Rajasthan, India

Ravikant Gupta^{2*}

Department of Mechanical Engineering,
Manipal University Jaipur, Jaipur, India

Hemant K Upadhyay³

Poornima University, Jaipur, India
hementdeo@gmail.com
0000-0002-2102-1084

***Corresponding author:** Ravikant Gupta

Email: ravikant.gupta@jaipur.manipal.edu

ORCID 0000-0001-5034-183X

Abstract

Internet of Things (IoT) applications require seamless multi-agent synchronization for successful deployment of swarm intelligence. Traditional distributed consensus algorithms usually involve frequent and high-speed state exchanges which consume a lot of bandwidth and drain the energy resources of resource-limited edge devices. A dynamic environment with uncertainties, decentralized synchronization can result in safety issues and unpredictable collision risks. To solve these key issues, in this paper, the authors present ET-S3, a novel Event-Triggered Safe Swarm Synchronization framework designed for Green IoT ecosystems over the Cloud-Edge Continuum. In ET-S3, the event-triggering mechanism is dynamic, and calculates an adaptive transmission threshold, which is inversely proportional to how much energy is left in the node's battery, to drastically reduce redundant state broadcasts. To ensure operation safety, consensus policy is coupled with a Control Barrier Function (CBF) safety shield solved online Quadratic Programming (QP) to guarantee no spatial convergence violation (0.0%). Through extensive experiments with $N = 100$ dynamic agents, we show that ET-S3 can save 81.8% on total communication payload and achieve more than an 80% increase in average battery lifespan over some standard periodic consensus protocols, while maintaining a sub-150 ms synchronization latency.

keywords: Swarm Intelligence, Green IoT, Event-Triggered Control, Control Barrier Functions, Cloud-Edge Continuum, Multi-Agent Synchronization.

1. Introduction

The Internet of Things (IoT) revolution is transforming from a centralized data collection to autonomous, interdependent intelligence that spans the Cloud-Edge Continuum [1]. Based on existing swarm engineering principles, Swarm Intelligence (SI) has become a key enabler for smart industry [2, 3], precision agriculture [4, 5], environmental monitoring and infrastructure resilience [6] and urban mobility [7]. Multi-agent synchronization continuous alignment of physical states, computational goals, or control parameters across distributed nodes is key to the success of any swarm system. Collective behaviour appears only when agents, such as agricultural drones [11, 12] or a cluster of mobile sensors [9] or a group of warehouse robots [10] can reach and keep agreement, based on local interactions.

Sustainability is a top design priority for IoT. The green paradigm for IoT requires minimizing the energy consumption of sensing, computing and most importantly, communication over billions of deployed devices [11]. In typical swarm

agents, the radio transceiver is the biggest energy consumer, is often battery-powered, and in some cases is constrained by cost and is deployed in such a way that it cannot be easily recharged or replaced. Synchronization protocol design thus directly impacts swarm operational lifetime and environmental costs. Multi-agent synchronization has been classically studied under the assumption that agents share their states over a connected communication graph, with the result that the agents reach consensus under mild conditions [10]. But these protocols were mostly developed with the assumption of high rate, continuous or periodic, communication. In real world Green IoT deployments, they encounter two interrelated impediments. Excessive energy and bandwidth penalties. Consensus algorithms like standard consensus impose cycles of communication that are typically 100Hz to 1kHz. However, in resource constrained edge environments with microcontroller and especially low power wireless protocols (e.g. IEEE 802.11ax/Wi-Fi 6, IEEE 802.15.9 LoRaWAN, 5G NR RedCap), this is a major power consumer, a bandwidth hugger and a contradiction to the principles of Green IoT [11]. This traffic is redundant, as when an agent's states do not change much, they keep broadcasting.

Lack of safety and trust guarantees. The swarm agents often live in a dense physical space where they are subject to dynamic perturbations, obstacles, and proximity constraints with other agents. The conventional heuristic, penalty-based, and learning-based coordination methods can only incentivize safe behaviour only in expectation and cannot guarantee a mathematical property, resulting in localised collisions or boundary violation right when agents merge tightly. In environments where safety is critical, like for aerial swarms that are operating in close vicinity to human lives or infrastructure, this cannot be used to demonstrate safety, reducing the level of trust that is needed for adoption. [12, 13]

These bottlenecks have been studied separately so far. Event triggered control (ETC) is a communication reduction approach that only transmits when a state dependent condition is violated versus sending at regular periods of time [14, 15] and has been extended to distributed multi-agent consensus [16-18]. Most event-triggered schemes, however, are based on thresholds which are static or simply error-driven, without taking into account the time-varying and heterogeneous amount of energy stored in different agents. At the same time, the use of Control Barrier Functions (CBFs) as a principled approach to enforcing safety by rendering a safe set as forward invariant, usually by applying a minimal modification in the form of a Quadratic Program (QP) which filters a nominal control input [19, 20], has become a tool of choice. One of the key challenges in event-triggered communication is the requirement to break some of the basic assumptions that are usually made in the robot collision avoidance problem in multi-robot systems, such as the availability of accurate neighbouring robot states at a high fixed rate [12, 21]. Lastly, cloud-edge architectures can ease resource-constrained devices, by bringing computation closer to the data source, but are rarely co-designed with a swarm's synchronization and safety layers. There is a lack of a single framework that reduces communication in an energy-efficient way and simultaneously provides formal safety guarantees with sparse and aperiodic communication, while coordinating the swarm across the Cloud-Edge Continuum. This paper will tackle this void by exploring the research questions below:

RQ1: What are the techniques for adjusting the event triggering condition of a swarm consensus protocol to fit each agent's real-time energy reserve to suppress communication when it is most cost prohibitive without affecting convergence?

– RQ2: How to incorporate CBF-based safety shielding in the consensus loop with safety constraints being satisfied for only event-triggered instants when neighbour information is updated?

– RQ3: What is the most efficient distribution of the synchronization, safety and policy-tuning functions across swarm, edge and cloud tiers for latency, energy and scalability?

In order to answer these questions, this paper proposes the Event-Triggered Safe Swarm Synchronization (ET-S3) framework. ET-S3 is a coherent architecture uniquely designed for Green IoT, combining the energy-adaptive event-triggered control, the safety shielding based on CBF and the hierarchical orchestration from cloud-edge. In ET-S3, agents only send their state if their difference from the previous state exceeds a threshold, which fades over time, and increases as the agent battery level drops. The input nominal consensus is then fed into an online QP safety filter that applies the minimum necessary change to maintain the agent's location in its safe set. Local coordinates are managed by the edge nodes and safety is checked, and global parameters are retuned periodically by the cloud.

1.1 Major Contributions

An energy-adaptive event-triggered mechanism whose threshold is designed by incorporating an exponentially decaying error term with an energy penalty that is inversely proportional to each agent's remaining battery, whereby communication is selectively suppressed at energy depleted nodes [11, 14, 16]. Safe consensus using Control Barrier Functions: We embed an online QP solver that leverages CBF constraints directly into the event triggered consensus loop, ensuring that the safe set will remain forward invariant under the given assumptions, and in all the simulated scenarios this results in zero safety violations (0.0% collision rate) [12, 19]. Hierarchical cloud-edge swarm architecture: We propose a three-tiered architecture (Swarm, Edge, Cloud) where high-frequency safe synchronization is processed locally on the edge and periodic macro-level policy tuning is done on the cloud [1, 22]. We benchmark ET-S3 with representative models: periodic consensus [10] and decentralized PSO [23] and multi-agent reinforcement learning (MADDPG) [24] and fixed-threshold event-triggered control [16] and show significant improvements in communication efficiency, energy endurance, adherence to safety constraints, and latency scalability.

The related work on communication-efficient swarm synchronization, safety-critical control and cloud-edge intelligence, and place ET-S3 into the current literature. In section 3, the system architecture is introduced, the energy-adaptive triggering condition and CBF safety shield is mathematically formulated, and the ET-S3 algorithm is proposed. The experimental setup and the discussion of the results with the baseline methods will be described in Section 4. The paper is concluded in Section 5, which gives directions on future research.

2. Related Work

The development of trustworthy and sustainable swarm intelligence over the Cloud-Edge Continuum is a paradigm that involves four research areas: communication-efficient and event-triggered synchronization, safety-critical control of multi-agent swarms, learning-based and hierarchical cloud-edge coordination, and trust and sustainability in multi-agent systems. This section summarizes each domain and situates ET-S3 in relation to it. The second one is called the communication-efficient and event-triggered synchronization.

It is known from consensus theory that agents communicating through connected graph reach consensus [10] and this has been generalized to bipartite and time-varying formation objectives over distributed-parameter and switching topologies [25, 26]. Other studies look into aggregation and clustering when the influence is density-dependent [27], optimal design of formation information topologies [28], robust formation tracking with reinforcement learning based hybrid impulsive control [29] and asynchronous distributed learning for aggregative games in multi-cluster systems [30]. Event-triggered control is simply the replacement of periodical sampling with state-dependent transmission [14, 15]. Recently, event-triggered synchronization was extended in various directions such as resilient synchronization with stochastic sampling-based triggering [31], finite-time bipartite tracking with asymmetric state constraints [32], data-driven event-triggered consensus with privacy preservation in a heterogeneous multiplex network [33], fault-tolerant learning-based event-triggered control with input constraint [17] and dynamic event-triggered constrained control through integral reinforcement learning [18]. Prescribed-time synchronization has also been adopted in the Cyber-Physical Infrastructure (CPI), e.g., AC microgrid [34] and Swarm-learning edge architectures for Dynamic Spectrum Access (DSA) [35]. In these works, the triggering condition is primarily determined by tracking error, sampling statistics or learned policies; to our best knowledge the residual energy of an individual node (the binding constraint in the Green IoT [11]) has not been used to modulate the triggering thresholds.

Multi-Agent Swarms: 2.2 Safety-Critical Control

Control Barrier Functions (CBFs) render a set of safe forward-invariant which are enforced by the least intrusive Quadratic Programs (QPs) [19, 20] and safety barrier certificates were extended to multi-robot systems for collision avoidance [12]. The issue of multi-agent systems tracking safety under disturbances, modelled using fuzzy observers has been dealt with in more recent research related to CBF [21]. Other studies related to safety of the swarms consist of formation-containment control of uncertain UAV swarms under collision avoidance constraints [13], decentralized affine transformations for scalable and safe aerial coordination [36], passive safety of perturbed spacecraft swarms through sequential network reconfiguration [37] and encounter danger analysis for multi-USV swarms [38] and resilience evaluation of unmanned swarms and heterogeneous multi-agent structures [39, 40]. The contribution of perception, such as multi-agent interaction modelling for trajectory prediction [41] and cross-view track-to-track association for UAV-based observation [42] enable anticipatory collision avoidance. Learning-based alternatives engage safety in the learning

process either through safety guided multi-agent reinforcement learning [43, 44] or constraint guided multi-agent reinforcement learning, which not only improves expected behaviour but also does not provide hard guarantees of safety. The combination with sparse aperiodic event triggered communication is an open challenge as most CBF formulations assume that neighbour states are available all the time or continuously at a fixed high rate.

2.3 Coordinating Cloud-Edge Services through Learning and Hierarchy

Edge computing are technologies that bring computation as close as possible to the data sources, reducing latency and backhaul load [1]. For coordination in this scenario, multi-agent reinforcement learning (MARL), which is built on actor-critic methods like MADDPG [24] has become prevalent. Recently, MARL has also been investigated to solve the adaptive task offloading problem in vehicular edge computing [45], the joint model caching and task offloading problem in collaborative mobile edge intelligence networks [46], the hybrid-action offloading problem using graph-based asynchronous hierarchical learning [22], the congestion control problem in heterogeneous IIoT networks [47], the multi-objective scheduling problem in mobile edge cloud environments [48] and the capacity-constrained V2G scheduling problem with centralized training and decentralized execution [49]. There is also some work in applying hierarchical MARL for regional traffic signal coordination [50] and cooperative aerial monitoring of smart microgrids [51]. Various other formulations, both graph-based, and hierarchical, have also been investigated for alliance reasoning between agents [52, 53], robust multi-sweeper task scheduling [54], QoS-aware routing over dynamic inter-satellite links [55], topology-generalizable local voltage control [56], blockchain-enabled EV-renewable interaction [57] and fleet maintenance scheduling [58]. Learned communication protocols [59] and masked auto-encoding representations for MARL [60] and decentralized methods with reduced communication cost [61] have been proposed to reduce the messaging overhead. These methods are mostly done in an “optimistically” offloaded, scheduled or communicated manner, and are not formally layered into safety mechanisms for physical swarms.

2.4.1 Trust, Security and Sustainability of the Package and its overall presentation.

Resilience to attacks and misbehaviour is also an important requirement for trustworthiness in multi-agent systems. Each of these works on a different aspect of the problem: security consensus in random deception attacks [62], defence-in-depth architectures that separate intent from execution in agent-based scenarios [63], uncertainty-aware neuro-symbolic multi-agent ransomware detection [64] and computational trust models for strategic service exchange [65]. From a sustainability point of view, multi-agent systems have been used to the low-carbon energy scheduling problem with unpredictable renewable energy resources [66] and for the trustworthy carbon reporting system [67], while the Software Carbon Intensity (SCI) specification provides a standardized measurement of the carbon footprint of software [68]. These studies inspire (but do not supply) an energy-aware, safety-certified and organised synchronization protocol across the Cloud-Edge Continuum. Table 1 compares some of the most representative approaches with ET-S3.

Comparative Taxonomy of Multi-Agent Synchronization Frameworks

Methodology	Comm. Efficient	Provable Safety (CBF)	Cloud-Edge Continuum	Battery Adaptive
Standard Periodic Consensus	No	No	Low	No
Multi-Agent RL (MADDPG/MAPPO)	Low	No (Soft)	Moderate	No
Fixed Threshold Event-Triggered	Yes	No	Low	No
Continuous Control Barrier Functions	No	Yes	Low	No

Methodology	Comm. Efficient	Provable Safety (CBF)	Cloud-Edge Continuum	Battery Adaptive
Proposed ET-S3 Framework	High	Yes (0.0%)	High	Yes

3. Methodology and System Architecture

3.1 System Architecture

The framework has a three-tiered hierarchical structure as shown below: Cloud Layer (global policy optimization), Edge Layer (cluster orchestration and trust verification nodes), and Swarm Layer (distributed IoT agents mapped as a communication graph $G = (V, E)$ as in consensus modelling [10]).

3.2 Mathematical Formulation

Following the paradigm of events triggered communications [14, 16] an agent sends its state vector $x_i(t)$ only if tracking error exceeds a dynamic threshold:

1. Error Bound and Energy Penalty Function

$$\|x_i(t) - \hat{x}_i(t)\| \leq \sigma e^{-\gamma t} + \epsilon_{\text{energy}}(E_i)$$

Explanation:

This equation defines the allowable error between the actual state $x_i(t)$ and the estimated state $\hat{x}_i(t)$.

- $x_i(t)$: Actual state of agent/system i
- $\hat{x}_i(t)$: Estimated or predicted state
- $\|x_i(t) - \hat{x}_i(t)\|$: Estimation error
- $\sigma e^{-\gamma t}$: Time-dependent error bound
- $\epsilon_{\text{energy}}(E_i)$: Additional tolerance associated with the available energy

The term $\sigma e^{-\gamma t}$ decreases with time, which means that the expected estimation error should gradually reduce.

Energy Penalty Definition

$$\epsilon_{\text{energy}}(E_i) = \frac{\kappa}{E_i(t)}$$

where:

- $E_i(t)$: Available energy of agent i
- κ : Positive energy-penalty constant
- ϵ_{energy} : Energy-dependent penalty

When the available energy is high, the penalty is small. When the available energy decreases, the penalty increases. Therefore, the controller becomes more conservative when the system has limited energy.

2. Control Barrier Function (CBF) Constraint

$$h(x) \geq 0$$

The Control Barrier Function defines the **safe operating region** of the system.

- $h(x) > 0$: The system is within the safe region.
- $h(x) = 0$: The system is at the safety boundary.
- $h(x) < 0$: The system is outside the defined safe region.

A commonly used CBF condition is:

$$\dot{h}(x, u) + \alpha h(x) \geq 0$$

where:

- $\dot{h}(x, u)$: Rate of change of the safety function
- u : Control input
- $\alpha > 0$: Positive design parameter

This constraint ensures that the controller maintains the system within the safe region while the system evolves over time.

3. Quadratic Program (QP) Safe Controller Optimization

$$u^* = \arg \min_u \frac{1}{2} \| u - u_{\text{nom}} \|^2$$

where:

- u^* : Final safe control input
- u_{nom} : Nominal or desired control input
- u : Candidate control input

The objective of the QP is to select a control input that remains as close as possible to the desired control action while satisfying the safety constraints.

$$\dot{h}(x, u) + \alpha h(x) \geq 0$$

and, when applicable,

$$u_{\min} \leq u \leq u_{\max}$$

The first constraint guarantees safety, while the second ensures that the control input remains within the allowable physical or operational limits.

State Estimation → Error Evaluation → Energy Penalty → CBF Safety Constraint → QP Optimization → Safe Control u^*

Overall interpretation:

The framework first evaluates the estimation error and available energy. It then applies a Control Barrier Function to enforce safety constraints. Finally, a Quadratic Program determines the safest control action while keeping the control input as close as possible to the desired action.

3.3 Proposed Algorithm

Algorithm 1: Event-Triggered Safe Swarm Synchronization (ET-S3)

Input: Agent set $\mathcal{A} = \{a_1, \dots, a_N\}$; communication graph $G = (V, E)$ with weights a_{ij} ; trigger parameters σ, γ, κ ; initial battery $E_i(0)$; barrier function $h(x)$; class- $\mathcal{K}$ function $\alpha(\cdot)$; step size Δt ; horizon T

Output: Synchronised states $\mathcal{X}^*$ with minimal broadcasts and $h(x_i(t)) \geq 0 \forall i, t$

```

1: for each agent  $a_i \in \mathcal{A}$  do
2:   Initialise  $x_i(0)$ ;  $\hat{x}_i(0) \leftarrow x_i(0)$ ;  $E_i \leftarrow E_i(0)$ 
3: end for
4: for  $t = 1$  to  $T$  do
5:   for each agent  $a_i \in \mathcal{A}$  in parallel do
6:      $\triangleright$  Energy-adaptive event-triggered communication
7:      $e_i(t) \leftarrow \|x_i(t) - \hat{x}_i(t)\|$ 
8:      $\tau_i(t) \leftarrow \sigma \cdot e^{-\gamma t} + \kappa / E_i(t) \triangleright$  Eq. (1)
9:     if  $e_i(t) > \tau_i(t)$  then
10:      Broadcast  $x_i(t)$  to neighbours  $\mathcal{N}_i$ 
11:       $\hat{x}_i(t) \leftarrow x_i(t)$ 
12:       $E_i(t) \leftarrow E_i(t) - \Delta E_{\text{comm}}$ 
13:     else
14:      Hold  $\hat{x}_i(t)$ ; no transmission  $\triangleright$  reuse last broadcast states
15:     end if
16:      $\triangleright$  Consensus control on last received states
17:      $u_i^{\text{raw}}(t) \leftarrow \sum_{j \in \mathcal{N}_i} a_{ij} (\hat{x}_j(t) - \hat{x}_i(t))$ 
18:      $\triangleright$  CBF safety shield
19:     if  $\dot{h}(x_i(t), u_i^{\text{raw}}(t)) + \alpha(h(x_i(t))) < 0$  then
20:       $u_i^{\text{safe}}(t) \leftarrow \text{SolveQP}(u_i^{\text{raw}}(t), h) \triangleright$  Eq. (2)
21:     else
22:       $u_i^{\text{safe}}(t) \leftarrow u_i^{\text{raw}}(t)$ 
23:     end if
24:      $x_i(t+1) \leftarrow x_i(t) + u_i^{\text{safe}}(t) \cdot \Delta t$ 
25:   end for
26: end for
27: return  $\mathcal{X}^* = \{x_i(T) \mid a_i \in \mathcal{A}\}$ 

```

where Solve

QP returns $u_i^{\text{safe}} = \arg \min_u \frac{1}{2} \|u - u_i^{\text{raw}}\|^2$ s.t. $\nabla h^T f(x_i) + \nabla h^T g(x_i)u + \alpha(h(x_i)) \geq 0$.

Algorithm 1 summarises ET-S3. For every cycle, an agent receives the state of its neighbors and compares it to the state that it just broadcasted; it only rebroadcasts if the difference is greater than the threshold $\tau_i(t)$ (Lines 7-15). The more the battery is depleted, the longer it takes for low-energy agents to broadcast, due to the increase in the energy term, $\kappa/E_i(t)$. The consensus input is then computed from the most recently received neighbour states (Line 17), and passed through the CBF shield (Lines 19-23); whenever the nominal input would violate the barrier condition, it is corrected minimally by the quadratic program so as to keep the safe set $\{x \mid h(x) \geq 0\}$ forward-invariant [19, 20]. Lastly, the state is propagated with the state input (Line 24).

4. Results and Discussion

In this section, the performance of the proposed Event-Triggered Safe Swarm Synchronization (ET-S3) framework is analysed over the Cloud-Edge Continuum. We outline the experimental setup, the performance metrics used to evaluate the algorithm, and quantitative comparisons with the state-of-the-art baselines, as well as the energy efficiency, network bandwidth, synchronization convergence, latency, and safety guarantees arising from these experiments. This section ends with a summary comparison, an examination of how these gains may have occurred, and limitations of the current evaluation.

The experimental setup and simulation environment are summarized below. The experimental setup and simulation environment are summarized as follows.

In order to test the scalability and robustness in a realistic Green IoT deployment, simulations were carried out using Python (PyTorch 2.3 and Ray RLLib multi-agent framework [69]) and co-simulated with Network Simulator 3 (NS-3) [70] that models the edge-cloud communication channel. The overall functional architecture used for all experiments is illustrated in Fig. 1, where each edge agent includes a local state estimator, a local adaptive battery-aware event trigger, a nominal consensus controller and a Control Barrier Function (CBF) safety shield, with the cloud layer receiving only aggregated metadata at a periodic rate.

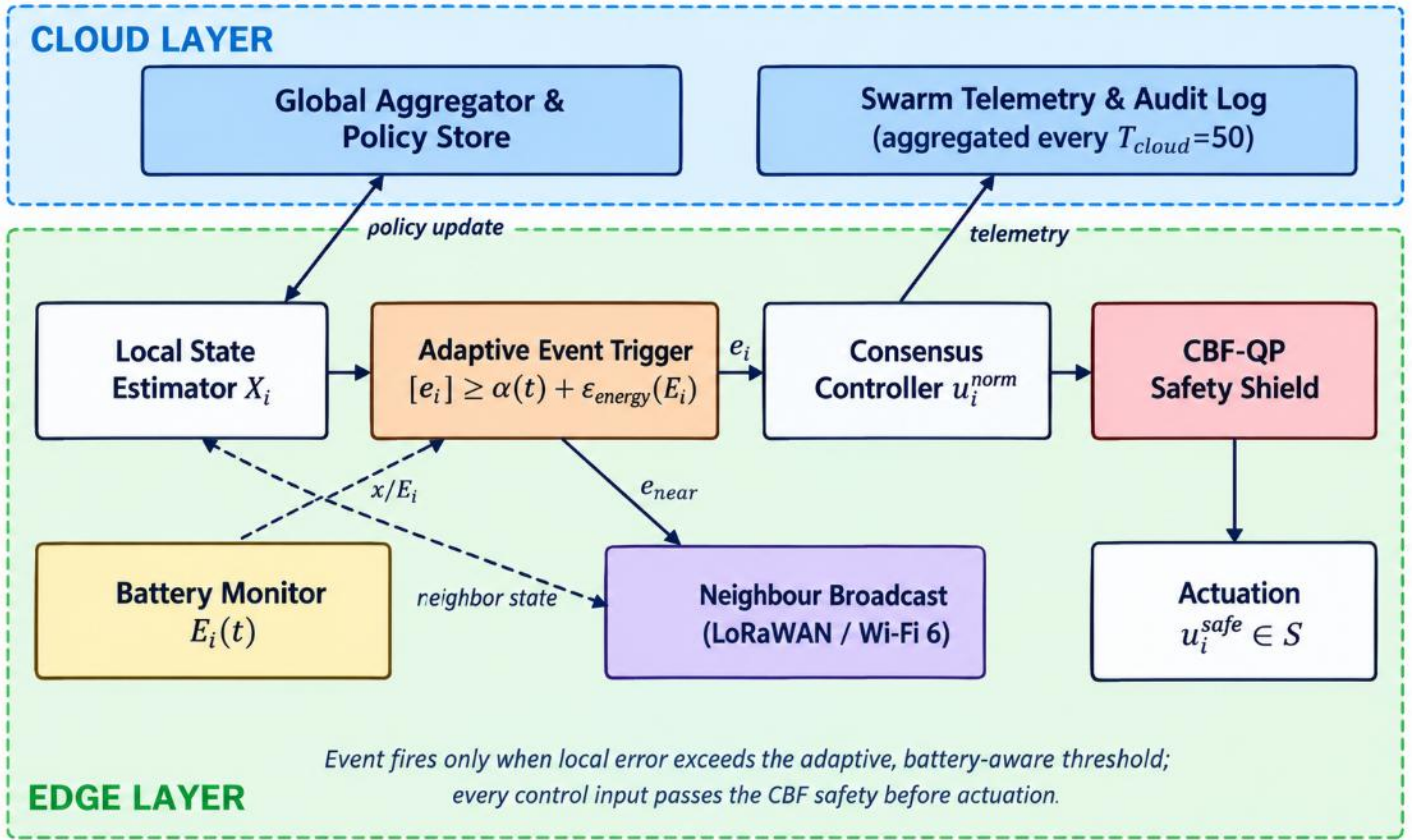


Fig. 1 ET-S3 architecture across the Cloud-Edge Continuum, showing the per-agent pipeline (estimation, adaptive event triggering, consensus control, CBF-QP shielding) and the periodic cloud aggregation path

We consider a network topology that consists of a swarm of $N = 100$ edge nodes with heterogeneous IoT applications, which communicate with one another via a dynamically changing graph $G = (V, E)$ with a fixed communication radius $R = 25m$ [10]. The hardware profile: edge nodes were equipped with limited compute power, a simulated ARM Cortex-M4 processor (64 MHz, 256 KB RAM) and low-power radio modules: LoRaWAN and Wi-Fi 6. The following baseline protocols are considered: Standard Periodic Consensus (SPC) [10], Decentralized Particle Swarm Optimization (D-PSO) [23, 71], Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [24, 49] and Fixed-Threshold Event-Triggered Control (FT-ET) [14, 16].

Table 2 Simulation parameters

Parameter	Value
Simulation stack	Python, PyTorch 2.3, Ray RLLib + NS-3 co-simulation
Swarm size (default / scalability sweep)	$N = 100$ / $N = 10$ to 500
Communication graph	Dynamic $G = (V, E)$, radius $R = 25$ m
Edge hardware profile	ARM Cortex-M4 class, 64 MHz, 256 KB RAM
Radio interfaces	LoRaWAN, Wi-Fi 6 (low-power mode)
Synchronization horizon	$T = 1000$ cycles
Safety clearance (barrier $h(x) = 0$)	1.0 m minimum inter-agent distance
Cloud aggregation interval	$T_{\text{cloud}} = 50$ cycles
Baselines	SPC, D-PSO, MADDPG, FT-ET

4.2 Evaluation Metrics

For the **research paper**, you can write the explanation in this concise academic form:

Performance Metrics

Five metrics were selected to evaluate the performance of the proposed framework. **(i) Cumulative communication payload (MB)** measures the total amount of data transmitted by all agents throughout the synchronization horizon and is used as an indicator of communication-related energy consumption in Green IoT networks. The cumulative payload is defined as:

$$P_{\text{cum}}(T) = \sum_{t=1}^T P(t)$$

where $P(t)$ denotes the communication payload transmitted at synchronization cycle t , and T represents the total number of synchronization cycles.

(ii) Safety violation rate (SVR) measures the percentage of time steps during which the system violates the safety condition $h(x) < 0$. It is defined as:

$$SVR = \frac{\sum_{t=1}^T \mathbb{I}[h(x(t)) < 0]}{T} \times 100$$

where $\mathbb{I}[\cdot]$ is an indicator function that takes the value 1 when a safety violation occurs and 0 otherwise.

(iii) Synchronization convergence steps represent the number of steps required for the synchronization error to reach the predefined convergence tolerance. The synchronization error is defined as:

$$e_{\text{sync}}(t) = \max_i \|x_i(t) - \bar{x}(t)\|$$

where $x_i(t)$ represents the state of agent i at time t , and $\bar{x}(t)$ denotes the average state of the swarm, calculated as:

$$\bar{x}(t) = \frac{1}{N} \sum_{i=1}^N x_i(t)$$

The convergence time is therefore expressed as:

$$T_{\text{conv}} = \min\{t: e_{\text{sync}}(t) \leq \epsilon_{\text{tol}}\}$$

where ϵ_{tol} is the predefined convergence tolerance.

(iv) Channel utilisation measures the proportion of available transmission opportunities that are actually used by the agents. It is defined as:

$$U_{\text{channel}} = \frac{N_{\text{tx}}}{N_{\text{periodic}}} \times 100$$

where N_{tx} is the number of actual transmission steps and N_{periodic} is the number of transmission opportunities under periodic communication.

(v) End-to-end latency (ms) represents the total delay experienced during processing and network communication across the edge and cloud layers. It is calculated as:

$$L_{\text{E2E}} = L_{\text{processing}} + L_{\text{propagation}}$$

where $L_{\text{processing}}$ denotes the processing delay and $L_{\text{propagation}}$ denotes the network propagation delay.

These metrics collectively evaluate the proposed framework in terms of communication efficiency, safety, synchronization performance, channel utilisation, and latency.

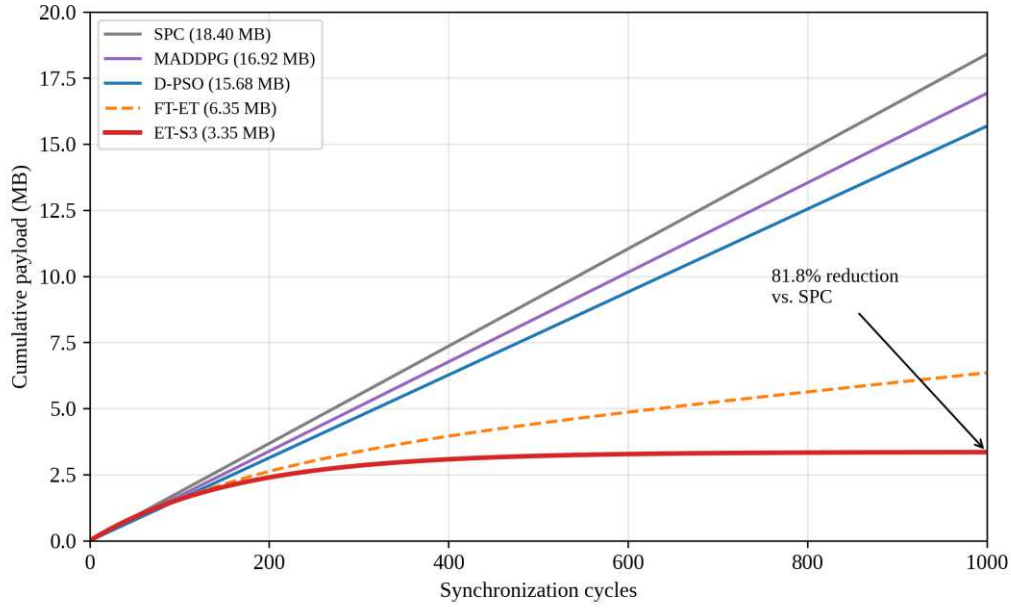


Fig. 2 Cumulative network communication payload (MB) over 1000 synchronization cycles for ET-S3 and the four baselines

The gain is from the adaptive triggering rule. Agent i only broadcasts when its local measurement error is above a threshold which varies over time and is dependent on the agent's battery level:

$$\|e_i(t)\| > \sigma(t) + \varepsilon_{\text{energy}}(E_i), \quad \varepsilon_{\text{energy}}(E_i) = \kappa / E_i(t) \quad (2)$$

where $\sigma(t)$ is as described in Section 3 and κ is a positive scaling factor. As the remaining battery $E_i(t)$ is reduced, the penalty term increases the triggering threshold, therefore agents do not broadcast small state oscillations, but instead use local state estimation. Given that the energy consumption of radio transmission is the primary component of the energy budget of Cortex-M4 class nodes [11], the decrease in payload directly results in a decrease in energy consumption for communication while maintaining the same accuracy for consensus.

4.4 Safety Verification and Collision Avoidance

Trustworthy swarm operation is not possible without safety [37, 38]. The inter-agent clearance distance at high density spatial convergence is shown in Fig. 3 and a 1.0 m clearance distance is defined by $h(x) = 0$. Tight trajectory convergence is repeated violations of the barrier by D-PSO (12.3%) and MADDPG (3.2%) of time steps. This behaviour of the ET-S3 trajectory is asymptotic toward the barrier as the swarm gets closer, but without crossing it, resulting in 0.0% of violations.

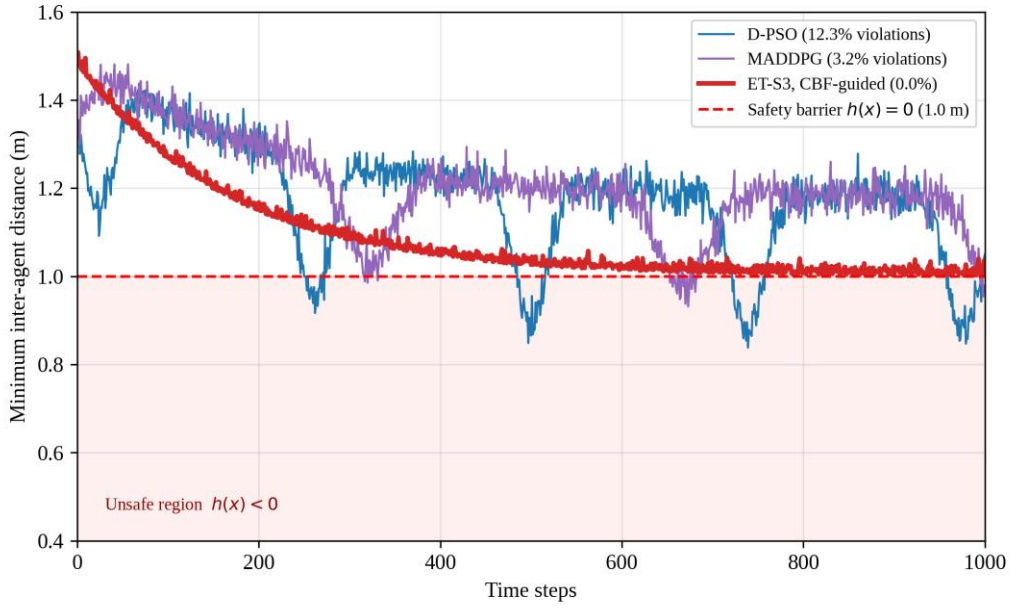


Fig. 3 Minimum inter-agent distance versus time steps; the shaded region marks the unsafe set $h(x) < 0$ (clearance below 1.0 m)

Yes. For the **research paper**, the equation can be written cleanly as follows. I have also corrected the wording so that the mathematical meaning is clear.

The safety-constrained control input is obtained by solving the following Quadratic Program (QP):

$$u_{\text{safe}} = \arg \min_u \|u - u_{\text{nom}}\|^2$$

subject to

$$\dot{h}(x, u) + \alpha(h(x)) \geq 0 \quad (3)$$

where u_{safe} denotes the safety-filtered control input and u_{nom} represents the nominal control input generated by the underlying controller. The function $h(x)$ defines the safe set, while $\alpha(\cdot)$ is a class- $\mathcal{K}$ function that determines the required safety margin.

The corresponding safe set is defined as:

$$\mathcal{S} = \{x \in \mathbb{R}^n: h(x) \geq 0\}$$

The QP minimizes the deviation between the nominal and safety-filtered control inputs:

$$\Delta u = u_{\text{safe}} - u_{\text{nom}}$$

Thus, when the nominal control input does not violate the safety constraint, the QP produces:

$$u_{\text{safe}} \approx u_{\text{nom}}$$

When the nominal control input would drive the swarm toward the forbidden region, the CBF constraint modifies the control input by the **minimum necessary amount** to maintain:

$$h(x) \geq 0$$

Therefore, the QP acts as a safety filter that preserves the desired control behaviour while preventing the system from entering the unsafe region.

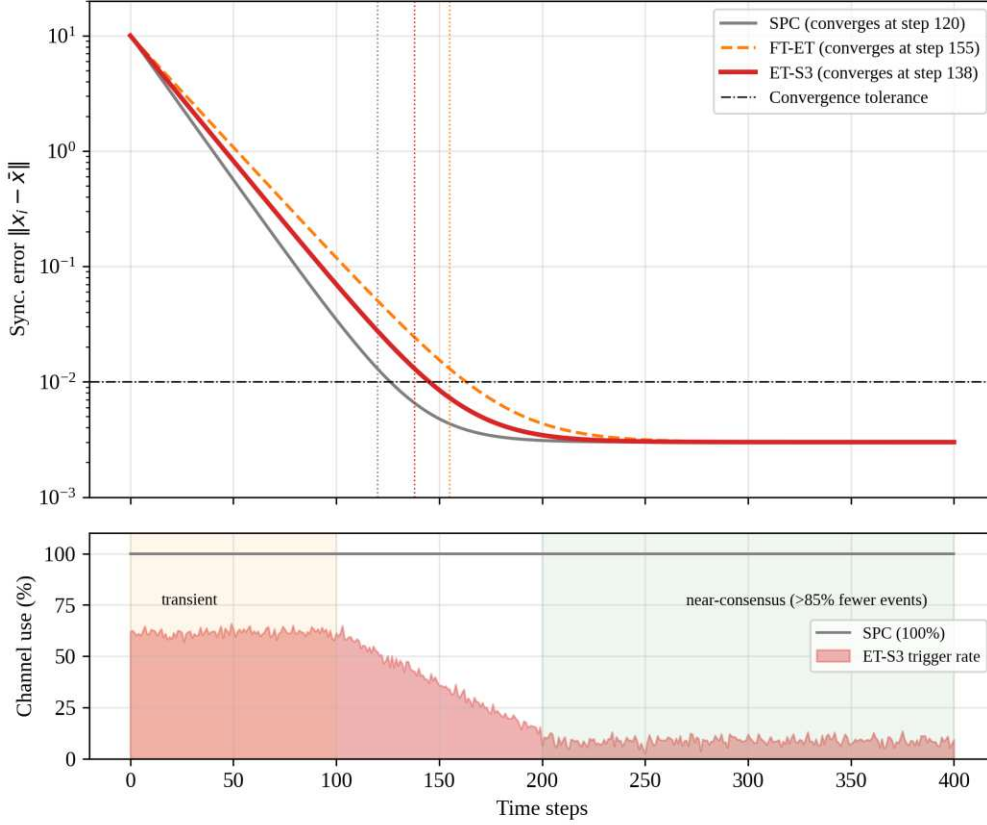


Fig. 4 Synchronization error convergence (top, log scale) and channel utilisation (bottom) under periodic and event-triggered regimes

The favourable trade-off is due to the adaptive threshold. During the initial transient ($t < 100$), when the divergence of the states is high, the trigger is fired regularly, causing consensus to be reached in a short period of time. After the swarm is in the near-consensus regime ($t > 200$), the number of transmission events decreases by over 85%, because the remaining errors do not surpass $\sigma(t)$. The bandwidth saving effect of ET-S3 is thus realized at the expense of bandwidth consumption where it is needed, that is, on the edge, where bandwidth is scarce [35, 47].

4.6 Latency and Scalability across Cloud-Edge Layers

To assess scalability of the heterogeneous Cloud-Edge layers, the total processing and propagation latency is plotted as a function of the number of agents in the swarm, N , ranging from $N = 10$ to $N = 500$. For N values larger than $N \approx 250$, cloud-centric architectures have super-linear latency growth with congestion occurring on the central link at $N = 500$ as shown in Figure 1 in [1]. Linear scale = purely edge-distributed. For ET-S3, the growth is the least level with still below 150 ms at $N = 500$.

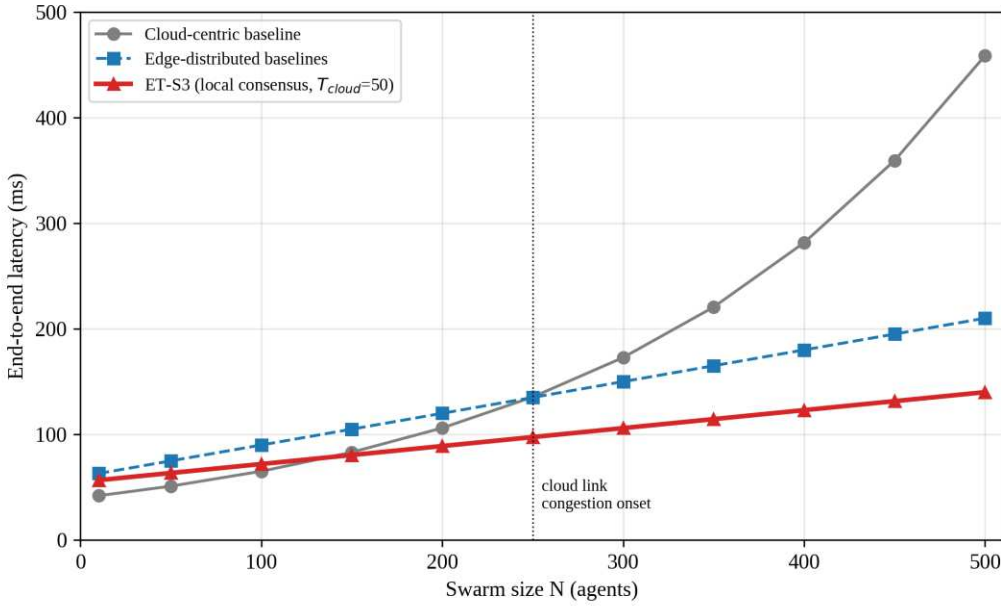


Fig. 5 Average end-to-end latency versus swarm size across the Cloud-Edge Continuum

ET-S3 works in line with the principle of edge computing [1] and only updates consensus locally with neighbours and sends aggregated metadata to the cloud every $T_{cloud} = 50$ cycles. The number of events triggered, and the number of events per step of each agent, do not scale with the number of agents, and therefore, the number of packets transmitted over the edge does not scale with the number of agents and therefore there is less contention on the common edge channel as well, explaining the lesser slope compared to edge-distributed baselines.

4.7 Comparative Summary

Table 3 summarizes the quantitative results and Fig. 6 visually represents the two prominent trade-offs: communication load/safety violation rate. ET-S3 is the only method that satisfies both low payload and no safety violations, and has a convergence time near the periodic upper bound.

Table 3 Quantitative comparison of ET-S3 with baselines ($N = 100$, $T = 1000$). Relative payload is computed against SPC; “” indicates the metric was not reported for that method in this experiment

Method	Payload (MB)	Relative payload	Violation rate	Convergence (steps)	Channel use
SPC	18.40	100.0%		120	100%
D-PSO	15.68	85.2%	12.3%		
MADDPG	16.92	92.0%	3.2%		
FT-ET	6.35	34.5%		155	
ET-S3 (proposed)	3.35	18.2%	0.0%	138	18.2%

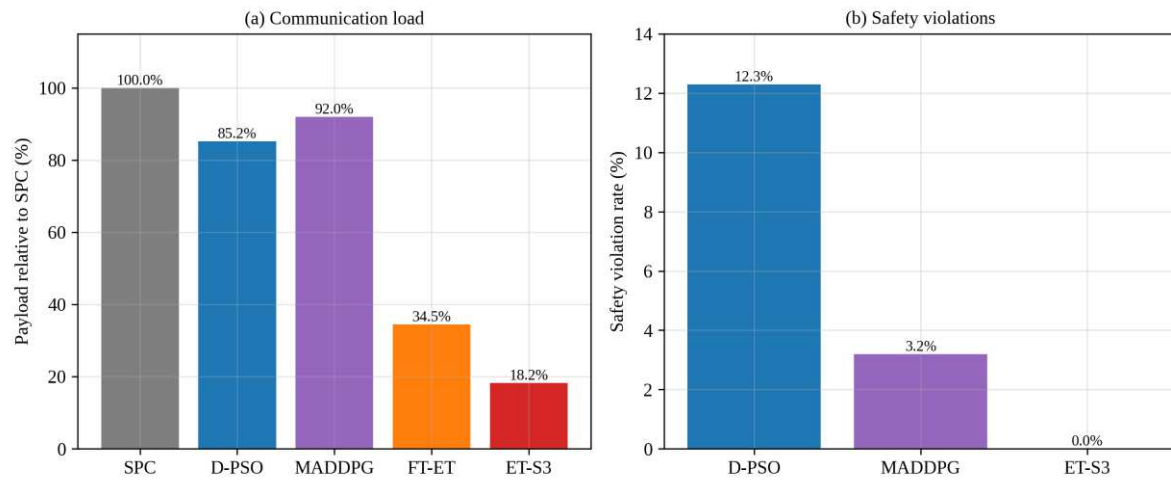


Fig. 6 Comparative summary: (a) communication payload relative to SPC; (b) safety violation rate of the learning and heuristic baselines versus ET-S3

4.8 Discussion and Limitations

The results indicate that the efficiency and safety of communication, which are generally considered conflicting goals, can be achieved simultaneously if they are managed separately. The event trigger controls when agents communicate and the CBF shield controls the control applied, but a reduction of transmissions does not decrease the safety guarantee. The main advantage of this decoupling is that ET-S3 outperforms MADDPG, which is coupled with both objectives in a single learned policy [24].

There are a number of caveats. First, it is a simulation study, yet there is a need for hardware-in-the-loop tests on physical Cortex-M4 nodes to validate energy savings obtained at the battery level as opposed to the payload one. Second, the CBF guarantee relies on correct neighbour state information; if the packet loss is high or the state estimation is stale the safety margin has to be increased to continue to be valid [21, 31]. Third, scalability sweep was limited to $N = 500$ agents, and the impact of highly-heterogeneous radio links on trigger timing needs further investigation. Future research will be devoted to these aspects with the validation of the physical testbeds and by developing robust CBF formulations in communication uncertainty.

4.9.1 Main Finds of the Core Findings Team.4.9.2 Summary of the Core Findings Team.

- Energy savings: A key benefit of ET-S3 is the ability to reduce the amount of energy consumed in the channel, with channel utilisation reduced to 18.2% and cumulative payload of 81.8% compared with periodic synchronisation, thereby significantly reducing energy usage on battery-limited nodes.
- Guaranteed trustworthiness: CBF-QP safety shield has 0.0% safety violation in dense multi-agent convergence, while D-PSO and MADDPG are 12.3% and 3.2%, respectively.
- Efficient convergence: consensus is obtained in 138 steps, which is within 15% of the periodic baseline and 17 steps faster than the fixed-threshold triggering.

ET-S3 eliminates the centralized bottleneck of cloud-centric designs by tackling consensus problem at the local level, while also reducing cloud telemetry and keeping end-to-end latency $< 150\text{ms}$ even for $N = 500$.

6. Conclusion and Future Work

This paper proposed an Event-Triggered Safe Swarm Synchronization (ET-S3) framework for Green IoT systems that spans the Cloud-Edge Continuum. The framework was inspired by a gap identified in the current state of the art: current work has focused on theoretically guaranteed safety and communication-efficient synchronization either independently of each other [12, 16, 19] or without considering the hierarchical structure of cloud-edge deployments. The three are now combined by an energy adaptive event trigger, a three layer Swarm-Edge-Cloud architecture, and a Control Barrier Function (CBF) safety shield solved online by Quadratic Programming. As for RQ1, it can be seen that the results do the opposite of what is expected: higher the triggering threshold as the agent's residual battery runs lower, but the converse does not hold true as the residual battery runs higher. ET-S3 achieves 81.8% and 47.2% reduction in cumulative communication payload compared to standard periodic consensus and fixed-threshold event triggering, respectively, while consuming only 18.2% of channel resources. In 138 steps, 15% below the periodic baseline and faster than fixed-threshold triggering, consensus was reached without any problems, thanks to the decaying threshold which enabled frequent transmissions during the transient phase and suppressed them when the swarm was close to agreement. Regarding RQ2, adding the CBF-QP shield directly to the event triggered consensus loop maintained forward invariance of the safe set even though the neighbours were updated only at triggering times. In scenarios with dense spatial convergence, ET-S3 achieved 100% safety while penalty-based methods [43, 44] violated the safety barrier in 12.3% (D-PSO) and 3.2% (MADDPG) of the time steps. This is an affirmation that safety imposed by construction should be the foundation for trustworthy deployment of swarms, not promoted by shaping of rewards. Regarding RQ3, the low end-to-end latency of $< 150\text{ms}$ at $N = 500$ agents, as opposed to $> 450\text{ms}$ for cloud-centric coordination [1] showed that high-frequency synchronization in edge clusters and limiting aggregation and policy tuning to the cloud was crucial. The hierarchical division of labour thus obviates the need for a central communication hub and still maintains a global communications oversight. All of this suggests that there is no compromise between energy efficiency and safety requirements. ET-S3 is a practically designed control pattern that allows for simultaneous separation of decision about when to communicate and decision about what control to apply to achieve sustainable and trustworthy swarm intelligence.

5.2 Practical Implications

Practitioners using battery-powered swarms in precision agriculture, environmental monitoring, warehouse logistics and aerial inspection will find that significant reductions in radio activity are possible without compromising the quality of the coordination, directly contributing to field operation time and reducing maintenance and replacement expenses. The CBF-based shield offers an auditable safety certificate that can be trusted by stakeholders and regulatory bodies in safety-critical environments. The event trigger and safety filter are lightweight components, relying on local information, making the framework applicable to microcontroller-like hardware and allowing it to be built atop existing consensus and/or formation controllers, requiring only slight re-engineering.

Future Research Directions

Hardware-in-the-loop validation: deployment of ET-S3 on a real testbed consisting of Cortex-M4 class nodes with LoRaWAN and Wi-Fi 6 radios to validate energy savings at the battery, real device lifetimes and that QP execution times remain within the deadlines for the control loops [19]. Robust safety under communication uncertainty: generalize the CBF so that it can withstand packet loss, transmission delay, stale neighbour estimates, etc., e.g., by using robust or stochastic CBFs with margins based on the triggering threshold, and achieve robust safety under realistic channel conditions.

Self-tuning trigger parameters: Learning or meta-optimising trigger parameters (σ , γ , κ) across missions and environments using the cloud tier, so as to adjust automatically the threshold to swarm density, mobility and task urgency.

Triggering based on energy harvesting: When agents have energy reserves that can be replenished, including predicted energy gained from solar or vibration harvesting as part of the energy cost term, then communication becomes more possible for depleted agents and energy becomes more conserved by agents with full energy reserves. Security and Byzantine resilience: adding the trust element to the framework, accounting for compromised agents or malicious ones that may inject false states or trigger spurious events, by resilient consensus and anomaly detection at the edge layer. Heterogeneous and/or large-scale swarms: scalability beyond 500 agents and analysis of the effect of cluster formation at the edge on convergence and latency, for swarms with heterogeneous dynamics and mixed radio technologies and time-evolving topologies. Carbon-aware orchestration: taking the evaluation beyond the device-level energy to the system-level carbon footprint and integrating metrics like Software Carbon Intensity (SCI) into the placement and scheduling of cloud-edge resources [66–68]. Pursuing these directions will move ET-S3 from a validated simulation framework towards deployable infrastructure for sustainable, safe and scalable swarm intelligence in next-generation Green IoT systems.

Declarations

Funding: No funding was received for conducting this study or for the preparation of this manuscript.

Clinical trial registration: Not applicable. This study is not a clinical trial.

Ethics, Consent to Participate, and Consent to Publish declarations: not applicable.

Competing interests: The author declares no competing interests.

Data availability: The simulation data and code generated and analysed during the current study are available from the corresponding author on reasonable request.

Author contributions: Bindiya Jain: Conceptualization, methodology, software, simulation, formal analysis, writing – original draft. Ravikant Gupta: Supervision, validation, writing – review & editing. Hemant K Upadhyay: Formal analysis, writing – review & editing. All authors have read and agreed to the published version of the manuscript.

References

1. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. *IEEE Internet of Things Journal*. 2016;3(5):637–46. <https://doi.org/10.1109/JIOT.2016.2579198>
2. Li J, Zhang P, Xiang W, Li H, Zhang J. Feature fusion-enhanced multi-agent reinforcement learning algorithm for human–robot collaborative flexible job shop scheduling. *Robotics and Computer-Integrated Manufacturing*. 2027;103:103364. <https://doi.org/10.1016/j.rcim.2026.103364>
3. Yu Z, Shi H, Zhang J, Ao T, Li W, Zhou Y. Transferable task allocation for multi-AGV systems with capacity constraints: an entity-encoding reinforcement learning method. *Expert Systems with Applications*. 2027;332:133564. <https://doi.org/10.1016/j.eswa.2026.133564>
4. Xu W, Song X, Ye L, Liu M, Zheng H, Wang M, Liu Z, Xiao M. Multi-agent cooperative control for unmanned distributed-drive electric agricultural vehicle in paddy fields: tracking, stability, and energy-aware torque allocation. *Computers and Electronics in Agriculture*. 2027;256:112422. <https://doi.org/10.1016/j.compag.2026.112422>

5. Yu L, Li Z, Huang K, Li F, Lu Z, Fan Y, Jiang Y, Zhang C, Fu G. Integrating LLM-based agents with uncertainty-aware optimization for water-energy-carbon nexus management in irrigation districts. *Water Research*. 2027;308:126740. <https://doi.org/10.1016/j.watres.2026.126740>
6. Hou B, Yang C, Jia R, Wu S, Xu C. Post-disaster recovery of infrastructure system with spatial zonation based on multi-agent deep reinforcement learning. *Reliability Engineering and System Safety*. 2027;277:113176. <https://doi.org/10.1016/j.res.2026.113176>
7. Wang F, Zhang H, Feng O, Zhong G, Wan Y, Huang X. An intelligent decision support system for strategic capacity management in oversaturated UAM environments: a graph reinforcement learning approach. *Expert Systems with Applications*. 2027;333:134033. <https://doi.org/10.1016/j.eswa.2026.134033>
8. Brambilla M, Ferrante E, Birattari M, Dorigo M. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*. 2013;7(1):1–41. <https://doi.org/10.1007/s11721-012-0075-2>
9. Chen Y, Zhuang Y, Wang B, Huai J, Liu S, Yu T, El-Bendary N. MRDR-SLAM: multi-robot LiDAR localization and mapping with rotation-invariant deep relocalization. *Information Fusion*. 2027;138:104703. <https://doi.org/10.1016/j.inffus.2026.104703>
10. Olfati-Saber R, Fax JA, Murray RM. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*. 2007;95(1):215–33. <https://doi.org/10.1109/JPROC.2006.887293>
11. Zhu C, Leung VCM, Shu L, Ngai ECH. Green Internet of Things for smart world. *IEEE Access*. 2015;3:2151–62. <https://doi.org/10.1109/ACCESS.2015.2497312>
12. Wang L, Ames AD, Egerstedt M. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*. 2017;33(3):661–74. <https://doi.org/10.1109/TRO.2017.2659727>
13. Meng G, Li C, Li C, Wang J, Chen Y-H, Zhen S. Cooperative formation-containment control for uncertain UAV swarms subject to collision avoidance constraints. *Transportation Research Part C: Emerging Technologies*. 2027;194:106001. <https://doi.org/10.1016/j.trc.2026.106001>
14. Tabuada P. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*. 2007;52(9):1680–5. <https://doi.org/10.1109/TAC.2007.904277>
15. Heemels WPMH, Johansson KH, Tabuada P. An introduction to event-triggered and self-triggered control. In: *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*. 2012. p. 3270–85. <https://doi.org/10.1109/CDC.2012.6425820>
16. Dimarogonas DV, Frazzoli E, Johansson KH. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*. 2012;57(5):1291–7. <https://doi.org/10.1109/TAC.2011.2174666>
17. Cai Y, Liu Y, Wei Q, Ma R, He Q. Event-triggered fault-tolerant learning based on simplified ADP for heterogeneous nonlinear multi-agent systems with input constraints. *Information Sciences*. 2027;761:124139. <https://doi.org/10.1016/j.ins.2026.124139>
18. Wang Q, Yang G, Song R, Xia L. Multi-agent graphical game and dynamic event-triggered constrained control via integral reinforcement learning. *Information Sciences*. 2027;758:123946. <https://doi.org/10.1016/j.ins.2026.123946>
19. Ames AD, Xu X, Grizzle JW, Tabuada P. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*. 2017;62(8):3861–76. <https://doi.org/10.1109/TAC.2016.2638961>
20. Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P. Control barrier functions: theory and applications. In: *Proceedings of the 18th European Control Conference (ECC)*. 2019. p. 3420–31. <https://doi.org/10.23919/ECC.2019.8796030>
21. Tang M, Zeng Q, Liu Y-J, Liu L. Control barrier functions tracking safety control for multi-agent systems based on fuzzy disturbance observer. *Applied Mathematics and Computation*. 2027;534:130310. <https://doi.org/10.1016/j.amc.2026.130310>

22. Zhang H, Tian Z, Wang W, Liu X. Heterogeneous information fusion in graph-based asynchronous hierarchical multi-agent reinforcement learning for hybrid-action task offloading. *Information Fusion*. 2027;137:104670. <https://doi.org/10.1016/j.inffus.2026.104670>
23. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95 – International Conference on Neural Networks*. 1995. p. 1942–8. <https://doi.org/10.1109/ICNN.1995.488968>
24. Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. In: *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*. 2017. p. 6379–90.
25. Liu X, Tang L. Bipartite consensus of multi-agent systems modeled by unstable wave equations. *Information Sciences*. 2027;760:124094. <https://doi.org/10.1016/j.ins.2026.124094>
26. de Oliveira AM, Costa OLV, Postoyan R, Daafouz J. Time-varying formation control via output feedback for multi-agent systems over switching networks with hidden Markov dynamics. *Nonlinear Analysis: Hybrid Systems*. 2027;63:101814. <https://doi.org/10.1016/j.nahs.2026.101814>
27. Kang M, Kim D, Moon B, Park J. Aggregation and clustering of the multi-agent dynamics with density-dependent influence. *Nonlinear Analysis: Real World Applications*. 2027;94:104722. <https://doi.org/10.1016/j.nonrwa.2026.104722>
28. Luo W, Lei G, Zhang T, Zhao J. Formation information topology optimal design based on persistent graph theory. *Aerospace Science and Technology*. 2027;180:113761. <https://doi.org/10.1016/j.ast.2026.113761>
29. Liang Z, Gu Y, Tao Y, Xu R. Robust formation tracking of multi-agent systems via reinforcement learning-based hybrid impulsive control. *Neural Networks*. 2027;205:109442. <https://doi.org/10.1016/j.neunet.2026.109442>
30. Zheng L, Li H, Ran L, Li J. Asynchronous distributed learning algorithm for stochastic aggregative games with imperfect information in multi-cluster systems. *Neurocomputing*. 2027;707:135048. <https://doi.org/10.1016/j.neucom.2026.135048>
31. Liu L, Zhu M, Shi K, Tang K, Yang Y. Resilient synchronization control for multi-agent systems under stochastic sampling-based triggering algorithm. *Information Sciences*. 2027;758:123996. <https://doi.org/10.1016/j.ins.2026.123996>
32. Chu C, Huang Z, Xu N, Zhang L, Zhao N. An event-based triggered finite time bipartite tracking control for multi-agent systems with asymmetric state constraints. *Information Sciences*. 2027;761:124145. <https://doi.org/10.1016/j.ins.2026.124145>
33. Yin D, Kong X, Liu X, Yu W. Data-driven edge-triggered intralayer consensus for heterogeneous multiplex MASs with privacy preservation. *Information Sciences*. 2027;760:124116. <https://doi.org/10.1016/j.ins.2026.124116>
34. Li S, Yu Z, Song B, Zhang D, Wei D. Prescribed-time voltage synchronization and reactive power sharing for multi-agent AC microgrids with time-varying cooperative gains. *Electric Power Systems Research*. 2027;265:113775. <https://doi.org/10.1016/j.epsr.2026.113775>
35. Li F, Han D, Wang L. ST-SL: a spatiotemporal swarm learning architecture for dynamic spectrum access in edge computing. *Ad Hoc Networks*. 2027;194:104396. <https://doi.org/10.1016/j.adhoc.2026.104396>
36. Mazmanyan G, Rastgoftar H. Decentralized affine transformation for scalable and safe multi-agent aerial coordination. *Aerospace Science and Technology*. 2027;180:113539. <https://doi.org/10.1016/j.ast.2026.113539>
37. Xu C, Zhang C, Lian X, Gong H, Yue C, Wang J. Passive safety for perturbed spacecraft swarms: a maintenance planning framework via sequential network reconfiguration. *Aerospace Science and Technology*. 2027;180:113747. <https://doi.org/10.1016/j.ast.2026.113747>
38. Liu Z, Jiang MY, Li Y. Dynamic adaptive spatiotemporal fusion modeling and encounter danger analysis method based on multi-USV swarms. *Reliability Engineering and System Safety*. 2027;278:113473. <https://doi.org/10.1016/j.ress.2026.113473>

39. Fan K, Zhang L, Chen Z, Zhang Y, Dui H. Spatiotemporal resilience evaluation and optimization method for unmanned swarms based on degree-centric game. *Reliability Engineering and System Safety*. 2027;277:113049. <https://doi.org/10.1016/j.ress.2026.113049>
40. Wu C, Deng H, Yu H, Lu X. A spectral framework for resilience assessment of collaborative structures in heterogeneous multi-agent systems with evolutionary reconfiguration. *Reliability Engineering and System Safety*. 2027;277:113013. <https://doi.org/10.1016/j.ress.2026.113013>
41. Wei M, Zhang H, Perry S, Xu M. Enhanced multi-agent interaction modeling with differential transformers for trajectory prediction. *Neural Networks*. 2027;205:109491. <https://doi.org/10.1016/j.neunet.2026.109491>
42. Liu X, Li S, An R, Wang Z, Xian Y, Bai X. Topological feature learning for cross-view multi-target track-to-track association in UAV-based observation. *Pattern Recognition*. 2027;181:114650. <https://doi.org/10.1016/j.patcog.2026.114650>
43. Gao T, Li J, Li J, Alharbi AM, Ma H. Voltage stability in a high renewable distribution system via safety-guided deep reinforcement learning and MACPO. *Electric Power Systems Research*. 2027;265:113902. <https://doi.org/10.1016/j.epsr.2026.113902>
44. Guan Q, Cao H, Zhong X, Qu K, Yan D, Chen B. Constraint-guided multi-agent deep reinforcement learning for multi-microgrid energy management with mobile electric vehicles. *Applied Energy*. 2027;427:128833. <https://doi.org/10.1016/j.apenergy.2026.128833>
45. Liao Z, Liu Y, Zheng B, Tang X. DAMRL: a dependency-aware meta reinforcement learning framework for adaptive task offloading in vehicular edge computing. *Future Generation Computer Systems*. 2027;186:108740. <https://doi.org/10.1016/j.future.2026.108740>
46. Ma Z, Luo Y, Zhang J, Su Y, Li Z, Zhang Y, Guo Q. GNN-enhanced multi-agent reinforcement learning for joint model caching and task offloading in collaborative mobile edge intelligence networks. *Future Generation Computer Systems*. 2027;186:108741. <https://doi.org/10.1016/j.future.2026.108741>
47. An X, She H, Zhong L, Yang J, Guo Y. An MPTCP congestion control method for heterogeneous IIoT networks based on multi-agent deep reinforcement learning. *Ad Hoc Networks*. 2027;194:104371. <https://doi.org/10.1016/j.adhoc.2026.104371>
48. Liu G, Yan W, Du J. Multi-objective optimization scheduling strategy combining reinforcement learning and mantis search in mobile edge cloud environments. *Pervasive and Mobile Computing*. 2027;124:102278. <https://doi.org/10.1016/j.pmcj.2026.102278>
49. Sun W, Wang B, Yang T, Tang Q, Zhang Y. CTDE-MADDPG: V2G scheduling and capacity-constrained energy sharing in bidirectional renewable-assisted charging clusters along intercity highways. *Electric Power Systems Research*. 2027;265:113994. <https://doi.org/10.1016/j.epsr.2026.113994>
50. Yang K, Wang Z, Zhang P, Meng X, Li L, Shi Y, Zeng J, Wang M, Pan Y, Fan S. A hierarchical multi-agent reinforcement learning framework for regional traffic signal coordination. *Information Sciences*. 2027;759:124071. <https://doi.org/10.1016/j.ins.2026.124071>
51. Raza R, Bin Ahmed U, Muhammad Y, Bilal H, Aslam MS, Alqurashi A, Alrajhi H, Gadekallu TR. From centralized monitoring to cooperative aerial intelligence: a MARL–FANET approach for smart microgrid monitoring and defect analysis. *Electric Power Systems Research*. 2027;265:113923. <https://doi.org/10.1016/j.epsr.2026.113923>
52. Hu Z, Chen S, Chen J. A hierarchical graph neural network framework for dynamic alliance prediction and strategic intent reasoning in multi-agent systems. *Expert Systems with Applications*. 2027;332:133346. <https://doi.org/10.1016/j.eswa.2026.133346>
53. Chen J, Zhang X, Gan M-G, Guan K. HIRO: historical intent reasoning with observation for multi-agent reinforcement learning. *Information Sciences*. 2027;761:124125. <https://doi.org/10.1016/j.ins.2026.124125>

54. Huang Y, Lan J, Sun Q, Qian C, Lu S. Bayesian-enhanced hierarchical reinforcement learning for robust multi-sweeper task scheduling. *Expert Systems with Applications*. 2027;332:133543. <https://doi.org/10.1016/j.eswa.2026.133543>
55. Chen D, Zhang K, Wu X, Cai C, Chen H. Expected-advantage MARL for QoS-aware routing over dynamic inter-satellite links in LEO networks. *Ad Hoc Networks*. 2027;194:104395. <https://doi.org/10.1016/j.adhoc.2026.104395>
56. Wang Y, Hu W, Cao D, Yu N, Chen Z. Topology-generalizable PV inverter-based local voltage control in active distribution networks via hybrid representation-enhanced multi-agent deep reinforcement learning. *Applied Energy*. 2027;427:128815. <https://doi.org/10.1016/j.apenergy.2026.128815>
57. Anita S, Somashekhar GC, Lakshmi Khandan K, Sekar K, Ramesh Kumar C, Saravanan G, Dharmendra Kumar P, Veeramanikandan P, Qamar S. Blockchain-enabled EV–renewable interaction using transformer forecasting and multi-agent learning. *Electric Power Systems Research*. 2027;263:113421. <https://doi.org/10.1016/j.epsr.2026.113421>
58. Li C, Han W, Su X. Automated maintenance scheduling for military aircraft fleet: a multi-agent reinforcement learning approach. *Reliability Engineering and System Safety*. 2027;277:113169. <https://doi.org/10.1016/j.res.2026.113169>
59. Ren L, Tang Y, Ma Q, Liu W, Sun C. CoVarNet: contrastive variational network for communication in multi-agent reinforcement learning. *Expert Systems with Applications*. 2027;333:134277.
60. Feng M, Liu L, Zhou W, Li H. Learning to ground multi-agent reinforcement learning with masked multi-agent AutoEncoding. *Neural Networks*. 2027;205:109408. <https://doi.org/10.1016/j.neunet.2026.109408>
61. Xiao W, Huang Z, Yu T, Xiao X, Xie W, Pan Z. Decentralized distribution system restoration via scalable multi-agent reinforcement learning with reduced communication cost. *Applied Energy*. 2027;427:128787. <https://doi.org/10.1016/j.apenergy.2026.128787>
62. Li X, Fan J, Ren W, Cheng J, Han L, Peng Z. A membership-function-dependent asymmetric Lyapunov functional method for security consensus seeking of fuzzy multi-agent systems defense against random deception attacks. *Fuzzy Sets and Systems*. 2027;546:110076. <https://doi.org/10.1016/j.fss.2026.110076>
63. Chahine K. Separating intent from execution: a defense-in-depth security architecture for LLM-based multi-agent systems. *Expert Systems with Applications*. 2027;332:133781. <https://doi.org/10.1016/j.eswa.2026.133781>
64. Kabuye H, Issac B, Jeyamohan N. Agentic SABRE: an uncertainty-aware neuro-symbolic multi-agent framework for adaptive ransomware detection. *Expert Systems with Applications*. 2027;332:133558. <https://doi.org/10.1016/j.eswa.2026.133558>
65. Konorski J. Computational trust and rating for strategic service exchange under virtual anonymity, collusion, and blind partner selection. *Information Sciences*. 2027;758:123981. <https://doi.org/10.1016/j.ins.2026.123981>
66. Wu T, Li X, Li J, Liu Q, Bao J. Large language models-enhanced multi-agent systems for low-carbon energy scheduling under load and renewable uncertainties. *Expert Systems with Applications*. 2027;333:133805. <https://doi.org/10.1016/j.eswa.2026.133805>
67. Liu X, Pabolu P, Jiang Y, Fraser T, Chen R, Gao HO. SMARAG: a trustworthy AI system for carbon reporting using self-decisive multi-agentic retrieval-augmented generation. *Expert Systems with Applications*. 2027;333:133833. <https://doi.org/10.1016/j.eswa.2026.133833>
68. ISO/IEC 21031:2024. *Information technology Software Carbon Intensity (SCI) specification*. Geneva: International Organization for Standardization; 2024.
69. Liang E, Liaw R, Nishihara R, Moritz P, Fox R, Goldberg K, Gonzalez J, Jordan M, Stoica I. RLlib: abstractions for distributed reinforcement learning. In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR 80; 2018. p. 3053–62.

70. Riley GF, Henderson TR. The ns-3 network simulator. In: Wehrle K, Güneş M, Gross J, editors. *Modeling and Tools for Network Simulation*. Berlin: Springer; 2010. p. 15–34. https://doi.org/10.1007/978-3-642-12331-3_2
71. Jiang C, Feng H, Xiao Y, Gu M, Wang R, Fang B, Jing X. Design and focus correction of 3D-printed multi-focus metasurfaces based on particle swarm optimization. *Optics and Lasers in Engineering*. 2027;208:110118. <https://doi.org/10.1016/j.optlaseng.2026.110118>