A Lightweight Adaptive Holding-time Policy for Clustered Wireless Sensor Networks

Trong-Minh Hoang[1], Thanh-Long Tran[1], Huy-Long Tran[1], Ngoc-Bich Pham[2], and Sinh Cong Lam[3]

[1] *Posts and Telecommunications Institute of Technology, Hanoi, Vietnam,*
[2] *Thang Long University, Hanoi, Vietnam,*
[3] *VNU University of Engineering and Technology, Hanoi, Vietnam*

https://doi.org/10.26636/jtit.2026.2.2598

Abstract — In clustered wireless sensor networks (WSNs), reshaping the topology can redistribute cluster head load, but each such task consumes energy. This paper studies the refresh timing problem in static clustered WSNs, where the controller decides not only whether to rebuild the topology but also determines the time over which the selected topology remains active. The proposed method formulates topology maintenance as a semi-Markov adaptive holding-time control problem. At each control epoch, the controller selects a refresh indicator, a target cluster count, and a holding time. The topology builder uses explicit cluster head election, nearest head member association, and intra-cluster chain forwarding with one-hop cluster head transmission to the base station. Under nominal deployment, the proposed controller reaches a half-node death (HND) point of 1969.1 ±8.4 rounds with 0.104 J of control energy, while periodic refresh with T = 10 reaches 1819.7 ±32.6 rounds and consumes 1.133 J. Across seven tested deployment scenarios, the proposed method gives a higher HND point with lower control energy than the tested refresh-enabled baselines. Therefore, the method is positioned as a lifetime overhead control mechanism, favoring lower control energy and longer mid-life operation, whereas periodic refresh remains preferable when delivery performance is the primary objective.

Keywords — adaptive holding time, energy efficiency, semi-Markov control, topology refresh, wireless sensor networks

1. Introduction

Clustered wireless sensor networks (WSNs) are used in environmental monitoring, smart agriculture, industrial sensing, and infrastructure supervision, as local data aggregation at cluster heads is capable of reducing the requirement for long-range radio transmissions from battery-powered sensor nodes to the base station [1]–[4]. In a clustered deployment, the operating bottleneck is not only the selected network topology, but also the process of updating that topology. Over subsequent communication rounds, residual energy, cluster head forwarding load, and member-to-cluster assignment may drift from the state used during the topology design phase, motivating a topology refresh before energy imbalance becomes severe.

A topology refresh is useful only when its reduction in data plane energy and energy offsets the added control energy cost. Frequent refreshes increase the overhead of status reporting, cluster head announcement, member reassociation, and schedule dissemination. Delayed refresh keeps an aging cluster assignment active after the residual energy distribution has changed, which can increase forwarding load on energy-depleted nodes. This paper studies clustered topology maintenance as a re-synchronization control problem, where the controller decides both the re-synchronization action and the holding time of the resulting topology.

Existing WSN clustering studies focus primarily on improving topology after reconfiguration. LEACH and HEED provide the classical baselines for cluster head rotation and energy-aware hybrid clustering [5], [6]. Recent studies further refine cluster operation through graph-based construction, energy-balanced routing, fuzzy or heuristic decision rules, and reinforcement learning-aided control [7]–[11]. These methods improve cluster head selection, forwarding load distribution, or route quality when the network decides to update its topology. The present papers address a narrower timing issue. The holding time of the selected topology is usually set to a fixed period or a one-step control clock, so its effect on control energy cost and residual energy drift is not explicitly controlled.

This paper makes the holding time of a topology control decision an explicit control variable. At each control epoch, the controller selects a refresh indicator, a target cluster count, and a holding time, rather than simply deciding whether to refresh the clustered topology. This converts topology maintenance into a variable duration control problem, where refresh responsiveness and control energy expenditure are handled within the same decision layer. The topology design module is kept explicit through cluster head election, nearest head member association, and intracluster chain forwarding, while the proposed controller operates above it to determine when to update the topology and for how long the updated topology should remain active.

The presented work is organized around three technical elements. First, clustered WSN topology maintenance is formulated as an adaptive holding time problem with explicit control energy accounting, cluster head election, and member association. Second, a factorized semi-Markov controller separates the triggering of refresh, the selection of the cluster count, and holding time selection, making the action structure consistent with the operation sequence of clustered topology maintenance. Third, an evaluation examines nominal operation, fixed duration baselines, a holding-time-blind ablation, deployment changes, connectivity-based QoS proxies, imperfect state observation, cluster count sensitivity, and runtime complexity.

In the clustered WSN setting under consideration, exposing the topology holding time as a control variable reduces unnecessary topology refreshes, lowers the control energy cost, and improves the HND operating point relative to fixed period and holding-time blind refresh policies. This benefit is obtained at the cost of lower performance compared to the periodic refresh approach, which remains preferable when delivery performance is the primary objective. Therefore, the results should be interpreted as a lifetime overhead operating trade-off, and not as dominance over all lifetime or delivery metrics.

The remainder of the paper is organized as follows. Section 2 reviews clustering, adaptive topology refresh, and variable-duration control. Section 3 defines the network model, topology design, energy accounting, and the retention time objective. Section 4 presents the proposed control pipeline. Section 5 describes the experimental protocol. Section 6 presents the results and limitations, while Section 7 concludes the paper.

2. Related Works

Recent works on clustered WSNs first and foremost address the designed spatial structure developed after a topology update. LEACH rotates the cluster head role to distribute energy consumption among sensor nodes, while HEED incorporates residual energy and communication cost into the cluster head selection process to improve energy-sensitive cluster formation [5], [6].

Later studies refine this topology construction layer through graph-based clustering, energy-balanced path-tree design, dynamic clustering, and chain- or tree-based forwarding structures [7], [9], [10]. Stable clustering, redundancy-aware topology control, and energy-constrained cluster formation have also been studied to improve cluster head selection, member association, and forward load distribution [12]–[14].

These methods improve the topology installed after the network decides to update its structure. They do not directly control how long the installed topology should remain active before the next network state observation.

State-aware WSN control has also been studied through fuzzy inference, hedge algebra, and heuristic decision rules. These methods use residual energy, distance, load, or link-related descriptors to support cluster head selection and routing decisions under uncertain network conditions [15].

Fuzzy and heuristic WSN schemes further show that energy and load descriptors are useful inputs for distributed control when residual energy decreases unevenly across the network [8].

This line of work is relevant to the present paper, as it confirms the value of compact network state indicators. Its control object, however, is usually the cluster head choice, routing decision, or rule output inside a predefined update loop. The persistence time of the selected topology control decision is not treated as a separate operating variable.

Adaptive clustering and learning-based WSN control move beyond one-shot topology design by conditioning actions on the observed network state. Recent work on Q-learning-based routing, dynamic topology reconfiguration, and intelligent clustering adapts routing or cluster organization to residual energy evolution, traffic changes, or topology degradation [10], [11], [16]. These schemes are closer to the setting considered here, since the control action changes along with the network condition.

The remaining timing issue is more specific. In many adaptive schemes, the decision clock is still imposed by a fixed period or by a one-step update rule and, hence, the controller can choose what action to take but is not capable of determining how long the resulting topology control decision should persist.

This timing issue is important, because the cost of topology refresh and the effects of topology aging affect network performance over different time scales. A frequent refresh can correct cluster assignment and forwarding-load imbalance earlier, but it increases status reporting, cluster head announcement, member reassociation, and schedule dissemination overhead. A long holding time reduces the control energy cost, but the retained cluster assignment may drift from the current residual energy distribution. Therefore, fixing the update interval removes a degree of control freedom that is directly tied to the HND overhead operating point.

Semi-Markov decision processes and temporally extended actions provide the control structure needed for this degree of freedom. In a semi-Markov model, the selected action can remain active for a variable number of time steps, before the next decision is made [17]. The repetition follows the same principle by allowing the controller to choose both the action and its execution duration [18]. This mechanism aligns with clustered WSN topology maintenance, because a topology refresh incurs an immediate control energy cost, whereas the effect of the refreshed topology accumulates over several communication rounds.

This paper addresses a temporal control layer above clustered topology development. It does not replace the head election, member association, or the design of the forwarding structure. Instead, it treats the holding time of the installed topology as a controller output and evaluates whether this variable improves the HND overhead operating point at the expense of an explicit topology refresh cost. This separates the proposed problem from general WSN clustering and from learning-based routing methods that retain a fixed decision clock.

3. System Model and Problem Formulation

3.1. Network and Topology Model

Consider a static clustered WSN with N sensor nodes and one base station. The network evolves over discrete communication rounds indexed by t. Let V_t denote the set of alive nodes and let G_t denote the active clustered topology. The topology consists of the cluster head set H_t, the member association map, and the intra-cluster forwarding structure. In the evaluation model, each cluster uses a PEGASIS-style intracluster chain, and each cluster head forwards aggregated traffic to the base station through a one-hop uplink. This topology model is therefore a clustered chain-assisted structure, not an unrestricted multi-hop routing graph.

When a reconfiguration event is triggered, the topology builder first elects cluster heads from the alive node set. For each active node $i \in V_t$, the cluster head quality score is:

$$q_i(t) = 0.4\left(1 - \frac{d_{i,\mathrm{BS}}}{d_{\max}}\right) + 0.3\, c_i(t) + 0.3\, \frac{e_i(t)}{E_0}, \qquad (1)$$

where $d_{i,\mathrm{BS}}$ is the distance from node i to the base station, $d_{\max}$ is the largest node-to-base-station distance in the deployment, $c_i(t)$ is the normalized centrality descriptor used by the topology builder, $e_i(t)$ is the residual energy, and E_0 is the nominal initial energy.

Equation (1) specifies the election rule used in the evaluation. A node obtains a higher score when it is closer to the base station, more central within the deployment, and has higher residual energy.

For a target cluster count C_n, the cluster head set is:

$$\mathcal{H}_{t_n} = \mathrm{Top}_{C_n}\left\{q_i(t_n) : i \in \mathcal{V}_{t_n}\right\}, \qquad (2)$$

where Top_{C_n} returns the C_n highest scoring alive nodes, with $|\mathcal{H}_{t_n}| = \min(C_n, |\mathcal{V}_{t_n}|)$.

Equation (2) enforces the count of the requested cluster when enough alive nodes remain in the network. It also prevents depleted nodes from being selected as cluster heads.

Each non-head node is then associated with the nearest selected cluster head:

$$a_i(t_n) = \arg\min_{h \in \mathcal{H}_{t_n}} d_{i,h}, \qquad i \in \mathcal{V}_{t_n} \setminus \mathcal{H}_{t_n}. \qquad (3)$$

Equation (3) defines the member association rule used after the cluster head election. Once the members are assigned, an intra cluster chain is constructed inside each cluster, and the resulting clustered topology remains active until the next refresh event.

3.2. Topology Age, State, and Composite Action

The topology age is updated as:

$$\tau_t = \begin{cases} 0, & \text{if a new topology is installed at round } t, \\ \tau_{t-1} + 1, & \text{otherwise.} \end{cases} \qquad (4)$$

Equation (4) records the number of communication rounds during which the current clustered topology has remained active. A larger τ_t indicates that the retained cluster assignment and forwarding structure have been used for more rounds and may have drifted from the current residual energy distribution.

At the decision epoch n, the controller observes:

$$\mathbf{s}_n = \big[\bar{e}_n, \sigma_{e,n}, e_n^{\min}, e_n^{\max}, \bar{e}_n^{\mathrm{ch}}, \sigma_{e,n}^{\mathrm{ch}}, \phi_n, C_n^{\mathrm{cur}}, \tau_n, \kappa_n, \hat{E}_{n-1}^{\mathrm{data}}, \hat{E}_{n-1}^{\mathrm{ctrl}}, \mathbb{I}(u_{n-1} = 1)\big], \qquad (5)$$

where the entries describe the residual energy statistics, cluster head energy statistics, alive node ratio, active cluster count, topology age, energy imbalance and recent data/control energy.

Equation (5) supplies the controller with compact network state descriptors and the recent cost of topology maintenance. These descriptors are used to decide whether the current topology should be retained or updated.

The controller selects:

$$\mathbf{a}_n = (u_n, C_n, d_n), \qquad (6)$$

where $u_n \in \{0,1\}$ is the refresh indicator, $C_n \in \mathcal{C}$ is the target cluster count, and $d_n \in \mathcal{D}$ is the holding time.

The next decision epoch is:

$$t_{n+1} = t_n + d_n. \qquad (7)$$

Equation (7) makes the decision clock action dependent. The selected holding time determines how long the current control decision remains active before the state of the controller observes the network again.

3.3. Energy Accounting and Utility Interpretation

The round-level energy consumption is separated into data plane and control plane components:

$$E_t^{\mathrm{tot}} = E_t^{\mathrm{data}} + E_t^{\mathrm{ctrl}}. \qquad (8)$$

Equation (8) makes topology refresh part of the network energy cost rather than treating the reconfiguration as a cost-free operation. The data plane component includes transmission, reception, and aggregation energy:

$$E_t^{\mathrm{data}} = \sum_{i \in \mathcal{V}_t} E_{i,t}^{\mathrm{tx/rx}} + \sum_{h \in \mathcal{H}_t} E_{h,t}^{\mathrm{agg}}. \qquad (9)$$

Equation (9) collects the energy used for data forwarding and local aggregation. The second term captures the additional cost of aggregation at the selected cluster heads. When topology refresh is triggered at t_n, the control plane energy is formulated as cost:

$$E_{t_n}^{\mathrm{ctrl}} = u_n |\mathcal{V}_{t_n}| \left(e^{\mathrm{status}} + e^{\mathrm{config}}\right), \qquad (10)$$

$E_t^{\mathrm{ctrl}} = 0$ for non-refresh rounds within the same holding segment.

Equation (10) models refresh overhead, as the status reporting and configuration dissemination costs are paid by the alive nodes when the clustered topology is rebuilt.

For a candidate decision (u, C, d), the idealized d-round utility can be written as:

$$\Delta_t(C,d) = \sum_{i=0}^{d-1} \left(E_{t+i}^{\mathrm{data,keep}} - E_{t+i}^{\mathrm{data,act}}(C)\right) - E_t^{\mathrm{ctrl}}(u, C). \qquad (11)$$

Equation (11) is used only to interpret the refresh-timing trade-off. A refresh is useful when the data plane energy reduction obtained by rebuilding the topology is large enough to offset the one-time control plane cost. A shorter holding time improves responsiveness to residual energy drift, whereas a longer holding time gives more rounds over which the refresh cost can be amortized.

Theorem 1. *For a horizon of T communication rounds, if $d_n \geqslant d_{\min}$ for every control epoch, then the number of control opportunities satisfies:*

$$N_{\mathrm{ep}}(T) \leqslant \left\lceil \frac{T}{d_{\min}} \right\rceil, \tag{12}$$

and the cumulative control energy satisfies:

$$E_{\mathrm{ctrl}}^{\mathrm{cum}}(T) \leqslant N\left(e^{\mathrm{status}} + e^{\mathrm{config}}\right) \left\lceil \frac{T}{d_{\min}} \right\rceil. \tag{13}$$

Proof. Each decision epoch covers at least $d_{\min}$ communication rounds. Hence, over a horizon of T rounds, the number of control opportunities is upper-bounded by (12). At each refresh event, the control-plane charge is at most $N(e^{\mathrm{status}} + e^{\mathrm{config}})$ because no more than N nodes can be alive. This gives the cumulative control-energy bound in (13). □

Theorem 1 does not define an optimal policy. It states the structural role of the holding time. Increasing the minimum holding time directly limits the maximum refresh frequency and the worst-case accumulation of control plane energy.

Because each action remains active for d_n communication rounds, the segment reward is:

$$\bar{r}_n = \sum_{i=0}^{d_n-1} \lambda^i r_{t_n+i}, \qquad 0 < \lambda \leqslant 1, \tag{14}$$

and the long-horizon objective is:

$$J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}} \left[\sum_{n=1}^{\infty} \gamma^{n-1} \bar{r}_n \right]. \tag{15}$$

Equations (14), (15) evaluate a topology control decision over its full holding segment, instead of only at the first communication round after refresh. This is the level at which topology aging, residual energy drift, and refresh overhead jointly affect the lifetime overhead trade-off.

4. Proposed Adaptive Holding-time Controller

4.1. Pipeline Overview

The proposed controller is organized as a four-block pipeline. The first block extracts compact state descriptors from the current network condition. The second block selects the factorized action (u_n, C_n, d_n). The third block rebuilds the clustered topology only when $u_n = 1$, using the topology-construction rules in Eqs. (1) – (3). The fourth block executes d_n rounds of data communication and returns the accumulated segment reward for the learning update.

This separation keeps the role of each module explicit. The learning policy controls refresh triggering, cluster-count selection, and holding time, while the cluster-head election and member-association rules remain fixed and are evaluated.

The reward in round t is:

$$r_t = \alpha\, \phi_t - \beta\, \hat{E}_t^{\mathrm{data}} - \gamma_c\, \hat{E}_t^{\mathrm{ctrl}} - \delta\, \kappa_t, \tag{16}$$

where ϕ_t is the alive-node ratio, κ_t is the energy-imbalance descriptor, and the energy terms are normalized by the initial network energy.

Equation (16) promotes node survival while penalizing the data plane energy, control plane energy, and residual energy imbalance. As a result, topology refresh is selected only when its expected benefit outweighs the additional control plane charge.

4.2. Factorized Policy and Learning Update

The policy is factored as follows:

$$\begin{aligned} \pi_{\boldsymbol{\theta}}\big(\mathbf{a}_n|\mathbf{s}_n\big) &= \pi_{\boldsymbol{\theta}}^{(u)}\big(u_n|\mathbf{s}_n\big)\pi_{\boldsymbol{\theta}}^{(C)}\big(C_n|\mathbf{s}_n, u_n\big) \\ &\quad \times \pi_{\boldsymbol{\theta}}^{(d)}\big(d_n|\mathbf{s}_n, u_n\big). \end{aligned} \tag{17}$$

Equation (17) separates the refresh decision from the topology scale decision and the hold time decision. This structure follows the operating sequence for clustered topology maintenance. The controller first decides whether to update the current topology. When a refresh is selected, the target cluster count determines the new topology scale. The holding-time branch then determines how many communications rounds the resulting control decision remains active before the next control epoch.

The critical target is:

$$y_n = \bar{r}_n + \gamma\, V_\psi\big(\mathbf{s}_{n+1}\big), \tag{18}$$

and the critical loss is:

$$\mathcal{L}_{\mathrm{critic}} = \mathbb{E}\big[\, (V_\psi(\mathbf{s}_n) - y_n)^2 \,\big]. \tag{19}$$

Equations (18), (19) train the value estimator using the return accumulated over a complete holding segment, rather than only the immediate communication round after a refresh decision. This is consistent with the operating sequence for clustered topology maintenance, where a selected topology remains active for multiple rounds before the next control epoch.

The advantage estimate is as follows:

$$\hat{A}_n = \bar{r}_n + \gamma\, V_\psi(\mathbf{s}_{n+1}) - V_\psi(\mathbf{s}_n), \tag{20}$$

and the actor update follows::

$$\nabla_{\boldsymbol{\theta}} J \approx \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\mathbf{a}_n|\mathbf{s}_n) \hat{A}_n\right]. \tag{21}$$

Equation (21) assigns credit to the complete control tuple (u_n, C_n, d_n) according to the multi-round network outcome produced by that tuple. This credit assignment is needed, because the control plane refresh cost is borne immediately, whereas the effect of the selected topology and holding time appears over subsequent communication rounds.

Algorithm 1 Adaptive holding-time topology control

Require: Deployment, energy parameters, cluster count set $\mathcal{C}$, holding-time set $\mathcal{D}$, actor parameters $\boldsymbol{\theta}$, critical parameters $\boldsymbol{\psi}$

for each training episode **do**
  Reset residual energy, alive node set, and initial clustered topology
  while the termination condition is not met **do**
    Design the control state $\mathbf{s}_n$ using Eq. (5)
    Sample the control tuple: $(u_n, C_n, d_n) \sim \pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_n)$
    if $u_n = 1$ **then**
      Elect cluster heads using Eq. (1)
      Associate members using Eq. (3)
      Charge the control plane energy using Eq. (10)
    end if
    Execute d_n communication rounds under the active clustered topology
    Accumulate the segment reward $\bar{r}_n$
    Design the next control state $\mathbf{s}_{n+1}$
    Update the critic using Eq. (19)
    Update the actor using Eq. (21)
  end while
end for

4.3. Algorithm and Complexity

Algorithm 1 separates the timing decision from the topology construction rule. The actor selects whether to refresh, which cluster count to use, and how long the resulting control decision remains active. The topology builder is invoked only when $u_n = 1$. Otherwise, the network continues data communication under the retained clustered topology.

The complexity of topology construction per refresh event is:

$$\mathcal{O}_{\text{topo}} = \mathcal{O}\left(|\mathcal{V}_t| \log |\mathcal{V}_t| + |\mathcal{V}_t| C_n + \sum_{h \in \mathcal{H}_t} |\mathcal{C}_{h,t}|^2\right), \quad (22)$$

where the three terms correspond to cluster head ranking, nearest head member association, and intra-cluster chain construction, respectively. Equation (22) also shows that topology construction is paid only at refresh events, not during every data packet transmission.

The policy inference cost is proportional to the number of actor parameters:

$$\mathcal{O}_{\text{policy}} = \mathcal{O}(P_{\boldsymbol{\theta}}). \quad (23)$$

Tab. 1. Deployment scenarios used for evaluation.

ID	Nodes/field	BS position	Purpose
S0	$100, 100 \times 100$	$[150, 50]$	Nominal
S1	$50, 100 \times 100$	$[150, 50]$	Sparse density
S2	$150, 100 \times 100$	$[150, 50]$	Dense density
S3	$100, 150 \times 150$	$[225, 75]$	Larger field
S4	$100, 100 \times 100$	$[50, 50]$	Centered BS
S5	$100, 100 \times 100$	$[200, 50]$	Far BS
S6	$100, 100 \times 100$	$[150, 50]$	Heterogeneous E_0

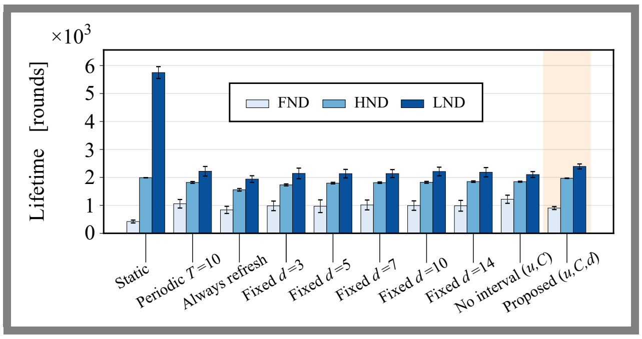

Fig. 1. Comparison of FND, HND and LND comparison under the nominal deployment.

Equation (23) applies only at control epochs and is incurred on the controller side. The sensor nodes do not execute the learning-based controller. They only receive and follow the disseminated topology configuration.

5. Experimental Design

The nominal setting uses $N = 100$ nodes in a 100×100 m field, a base station (BS) at $[150, 50]$, initial energy $E_0 = 0.5$ J, and 12 000 maximum rounds. The holding-time set is $\mathcal{D} = \{3, 5, 7, 10, 14\}$, and the baseline cluster count candidate set is $\mathcal{C} = \{3, 4, 5, 6, 7, 8\}$. Each main result is evaluated over 30 random seeds.

The baselines are static topology, periodic re-update with $T = 10$, always refresh, fixed-holding-time controllers with $d \in \{3, 5, 7, 10, 14\}$, a holding-time-blind adaptive controller (u, C) with fixed $d = 7$, and the proposed controller (u, C, d). A flat-action DQN baseline is also evaluated as an internal action factorization ablation. The evaluation scenarios are summarized in Tab. 1. They are designed to determine whether the result is limited to a single nominal deployment. The scenarios vary in node density, field size, base station location, and initial energy heterogeneity.

WSN performance is evaluated through lifetime (Fig. 1), topology maintenance overhead, and service-related metrics. Lifetime is measured based on first node death (FND), half-node death (HND), and last node death (LND). Overhead is measured based on control energy (CtrlE), refresh count, and average holding time. Service behavior is summarized by connectivity-based delivery, hop-count delay, throughput, coverage at HND, and Jain fairness at HND. These service-related metrics are treated as proxies, because the simulator does not include a full MAC layer scheduler, contention, retransmission, or stochastic physical layer decoding.

6. Results and Discussion

6.1. Nominal Lifetime Overhead Performance

Table 2 and Fig. 2 present the nominal comparison. The proposed controller attains 1969.1 ±8.4 rounds in HND with only 0.104 J control energy. Periodic refresh with $T = 10$ achieves 1819.7 ±32.6 rounds and consumes 1.133 J. Always

Tab. 2. Comparison of nominal lifetime and control energy (30 runs).

Method	FND	HND	LND	CtrlE [J]
Static	421.9 ±52.7	1988.3 ±2.1	5749.2 ±212.2	0.000
Periodic $T = 10$	1056.7 ±152.7	1819.7 ±32.6	2219.4 ±171.1	1.133
Always refresh	837.5 ±128.0	1556.0 ±47.1	1939.8 ±119.3	8.969
Fixed $d = 7$	1014.4 ±176.9	1811.5 ±27.0	2136.8 ±142.6	1.577
Fixed $d = 14$	984.8 ±191.9	1848.5 ±27.6	2184.9 ±166.3	0.803
No interval (u, C)	1218.4 ±145.9	1846.3 ±17.6	2101.6 ±105.1	1.651
Proposed (u, C, d)	903.6 ±54.6	1969.1 ±8.4	2391.5 ±89.1	0.104

refresh performs worse, because the topology is updated too aggressively and 8.969 J is spent on control energy.

The observation is metric-specific. The proposed controller does not maximize the FND, as the no-interval ablation reaches a higher FND point. Its advantage appears in HND, LND, and controlled energy. Compared to no-interval ablation, the proposed controller improves HND by 122.8 rounds and reduces the control energy by approximately 15.9×. This shows that the holding-time branch suppresses unnecessary topology refreshes and shifts the operating point toward longer mid-life and late-life network operation at lower control energy cost.

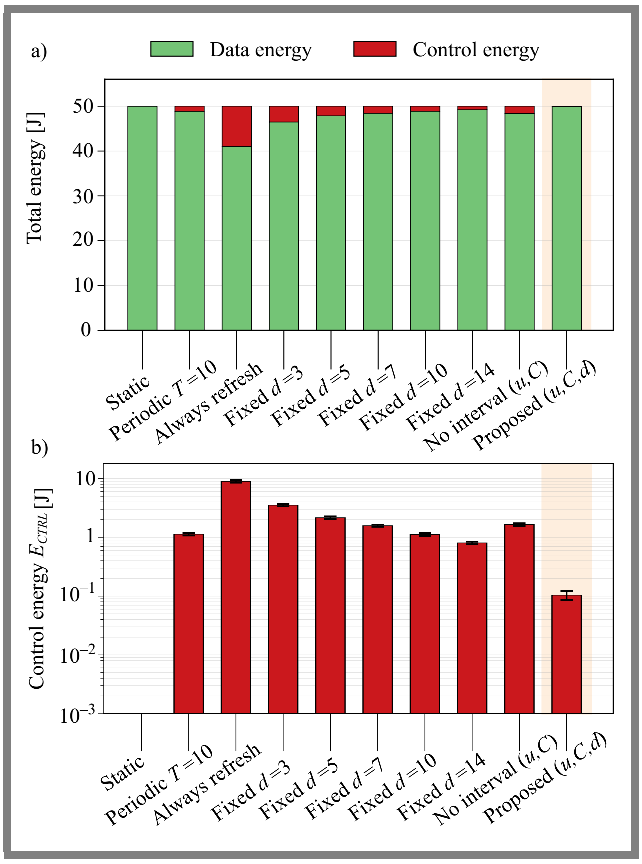

Fig. 2. Energy decomposition under nominal deployment.

6.2. Fixed Holding Time and Factorization Effects

The fixed-holding-time baselines show the limitations of using a single designer-selected update period. The best fixed baseline in Tab. 2 is $d = 14$, which reaches 1848.5 HND rounds with 0.803 J of control energy. The proposed controller reaches 1969.1 HND rounds with 0.104 J of control energy. Therefore, the improvement is not explained by selecting a long holding time. The controller jointly decides whether to refresh the topology and how long the selected topology control decision should remain active, so the refresh is triggered only when the observed network state justifies the additional control plane charge.

The flat-action DQN ablation further examines the factorized action structure in Eq. (17). With the same state descriptors and reward, the flat-action DQN reaches 1930.5 ±15.9 HND but consumes 0.302 J of control energy, which is about 2.84× higher than the proposed factorized controller in the matched ablation setting. This result supports separating refresh triggering, cluster count selection, and hold-time selection. It is used only as an internal action-structure ablation, not as a substitute for external WSN clustering or routing baselines.

6.3. Cluster Head Election and Topology Verification

The topology construction verification checks the lower layer clustering process used in the evaluation model. Across the verified refresh events, no depleted node is selected as a cluster head. Every alive non-head node is associated with exactly one cluster, and the realized number of cluster heads matches the requested C_n whenever $|\mathcal{V}_t| \geqslant C_n$. The mean topology development time is 104.3 ms in the verification run, while the broader runtime profile in Subsection 6.8 reports a mean topology build time of 147.6 ms. In the evaluation model, a topology update is therefore executed within one control epoch. This timing result should not be interpreted as distributed reconfiguration latency in a physical WSN deployment.

This verification fixes the interpretation of the proposed controller. The learning policy does not depend on an unspecified clustering routine. It decides when to invoke the topology builder, which cluster count to request, and how long the resulting clustered topology should remain active. Cluster head election, member association, and intra-cluster chain construction are defined by the topology construction module described in Section 3.

6.4. Multi-scenario Validation

Table 3 and Fig. 3 show the validation for seven scenarios. The proposed method provides a better HND-CtrlE operating point than the best non-static refresh-enabled competitor. The largest gains occur when the field is enlarged, or the base station is moved farther away, because topology refresh becomes more expensive and holding-time control can amortize that cost.

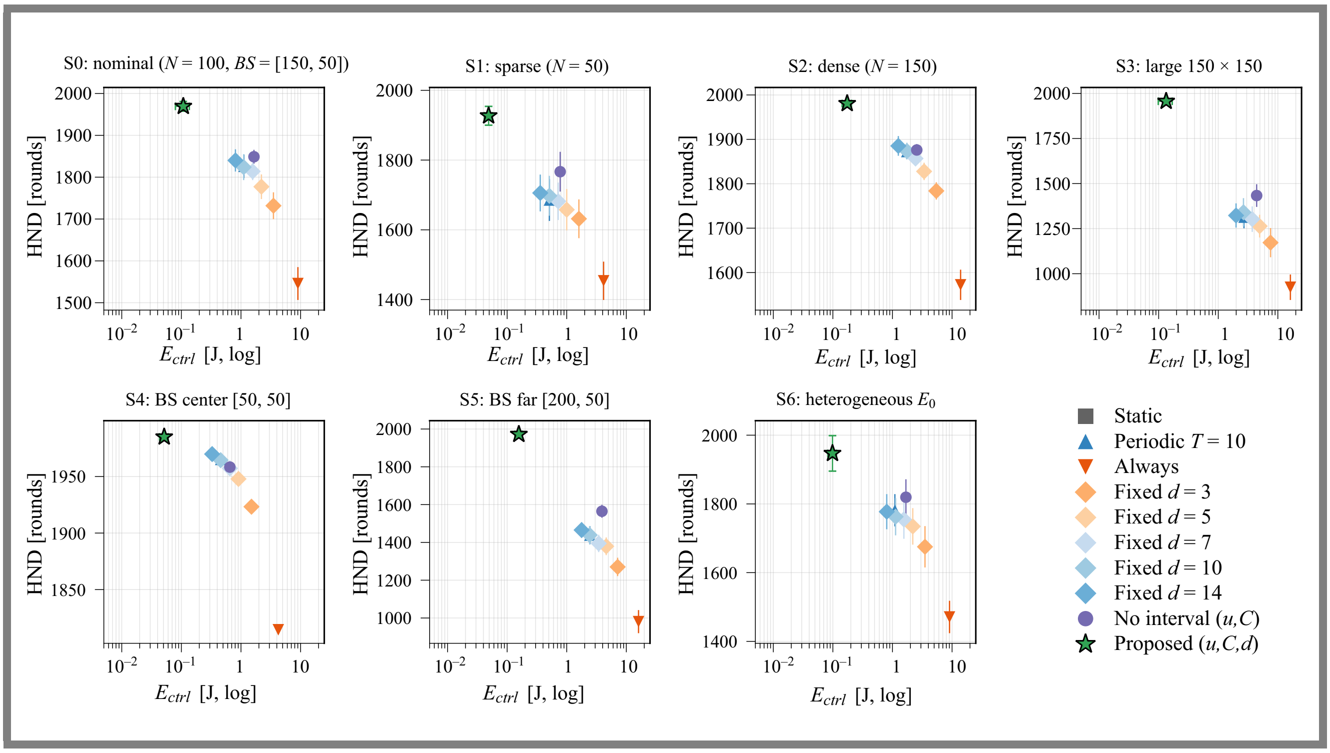

Fig. 3. HND versus control energy in seven deployment scenarios.

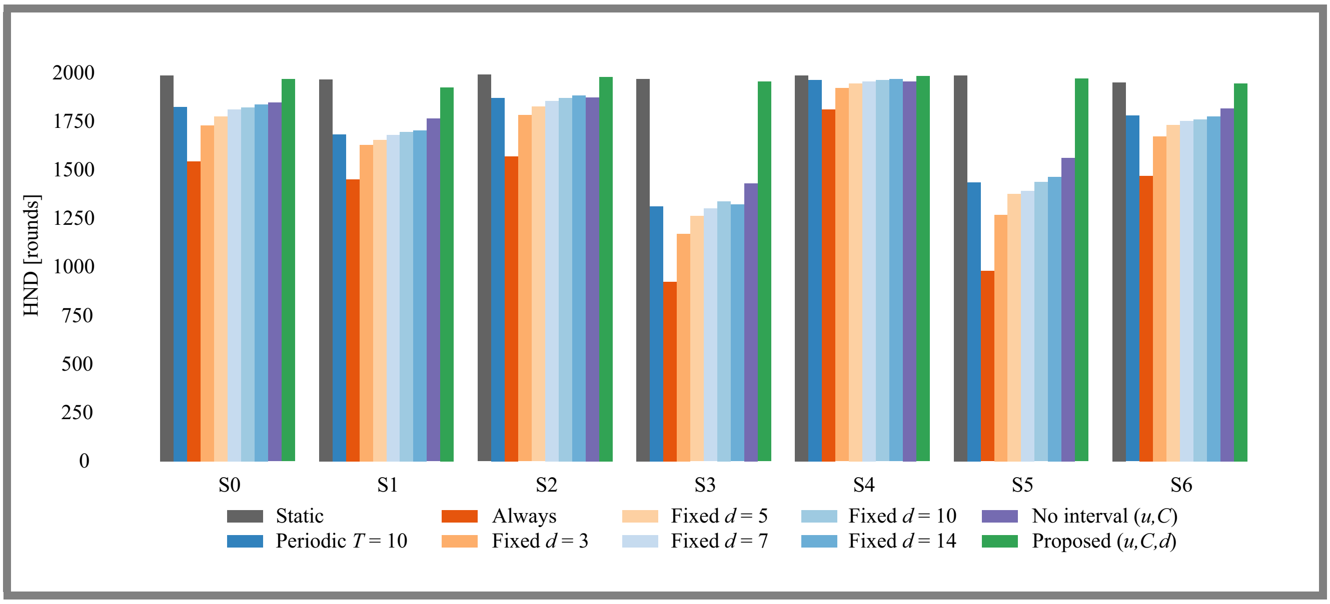

Fig. 4. Comparison of HNDs for tested scenarios.

The comparison also reveals where the gain is small. In S4, the base station is placed in the center of the field, reducing communication costs and making simple refresh policies more competitive. The HND gap is only 15.1 rounds. This indicates that adaptive holding time is most useful when refresh cost and topology staleness create a pronounced trade-off.

6.5. Connectivity-based Delivery, Coverage, and Fairness

Table 4 and Fig. 4 report the service-related proxies used to complement lifetime and overhead metrics. Periodic refreshes achieve a higher connectivity-based delivery value, because they update the clustered topology more frequently. The proposed controller refreshes less frequently, so the retained topology can age for more communication rounds. This produces a clear operating trade-off. The proposed method improves the overhead operating point, whereas periodic refreshing better preserves delivery-oriented performance.

The main limitation appears in S5, where the base station is farther from the sensing field. The proposed controller still improves the HND in this scenario, but its connectivity-based delivery value decreases to 0.352. This indicates that reduc-

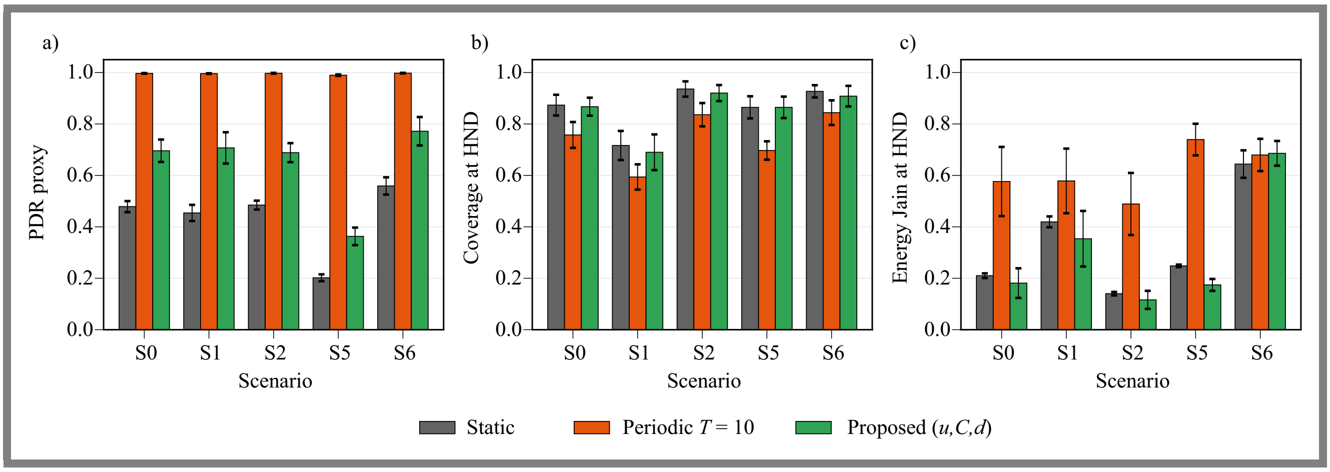

Fig. 5. Service-related proxy, coverage, and fairness.

ing the topology refresh frequency can extend the lifetime of the network while weakening delivery-oriented performance. For deployments where delivery performance is the primary objective, the controller would require a stronger delivery-oriented reward term, a different relay/base station placement, or a topology builder designed explicitly for delivery preservation.

6.6. Sensitivity to Imperfect Network-state Observation

Table 5 and Fig. 5 report the sensitivity of the proposed controller to residual energy observation noise and delayed network state information. The residual energy observation noise with $\sigma \in \{0.02, 0.05, 0.10\}$ changes HND by less than 0.2% relative to the clean proposed-policy reference. State-

Tab. 3. Comparison of multi-scenario HND overhead against the best non-static refresh-enabled competitor.

Scen.	Proposed HND	CtrlE	Best competitor	HND gap
S0	1969.3	0.108	No interval	+120.5
S1	1927.0	0.049	No interval	+160.3
S2	1980.7	0.174	Fixed d =14	+95.5
S3	1956.6	0.134	No interval	+523.1
S4	1985.0	0.051	Fixed d =14	+15.1
S5	1972.0	0.157	Fixed d =14	+506.6
S6	1947.1	0.098	No interval	+127.7

Tab. 4. Delivery, coverage, and fairness under periodic refresh and the proposed controller.

Scen.	Method	Delivery	Cov.@HND	Jain@HND
S0	Periodic	0.996	0.743	0.528
S0	Proposed	0.700	0.869	0.192
S2	Periodic	0.997	0.829	0.538
S2	Proposed	0.700	0.929	0.117
S5	Periodic	0.989	0.691	0.738
S5	Proposed	0.352	0.867	0.175
S6	Periodic	0.998	0.844	0.679
S6	Proposed	0.772	0.908	0.686

Tab. 5. Sensitivity of the proposed controller to imperfect observation of the state of the network.

Perturbation	HND	CtrlE [J]	HND change
Clean	1967.3 ±10.7	0.109	0.00%
Noise $\sigma = 0.02$	1969.8 ±6.9	0.104	+0.13%
Noise $\sigma = 0.05$	1967.9 ±8.3	0.104	+0.03%
Noise $\sigma = 0.10$	1970.6 ±9.4	0.092	+0.17%
Delay $\Delta = 1$	1980.4 ±2.7	0.074	+0.67%
Delay $\Delta = 2$	1979.6 ±2.4	0.085	+0.63%
Delay $\Delta = 5$	1979.9 ±2.8	0.100	+0.64%

observation delay with $\Delta \in \{1, 2, 5\}$ control epochs keeps HND within 0.67% of the clean reference.

These results show limited sensitivity to the perturbation levels in the state vector. The evaluation covers residual energy observation noise and short state observation delays. It does not cover packet loss, node mobility, MAC contention, bursty traffic, or adversarial sensing errors. The result should therefore be interpreted as state observation sensitivity within the tested operating range, not as general robustness to all deployment impairments.

6.7. Cluster Count Selection Sensitivity

The cluster count sensitivity study explains how the controller constrains the choices for cluster count. The controller does not optimize a continuous cluster count. It selects C_n from a predefined finite candidate set. Table 6, Figs. 6 – 7 show that the baseline set $\mathcal{C} = \{3, 4, 5, 6, 7, 8\}$ gives the highest proposed-policy HND among the evaluated candidate sets.

Tab. 6. Sensitivity of the candidate set with cluster count for the proposed controller.

Set	Values	HND	CtrlE [J]	Avg. C
C_{base}	{3, 4, 5, 6, 7, 8}	1967.5 ±11.8	0.107	3.34
C_1	{4, 6, 8}	1863.2 ±38.4	0.829	5.22
C_2	{5, 8, 10}	1916.8 ±25.4	0.285	5.41
C_3	{6, 10, 14}	1921.3 ±23.9	0.191	6.13

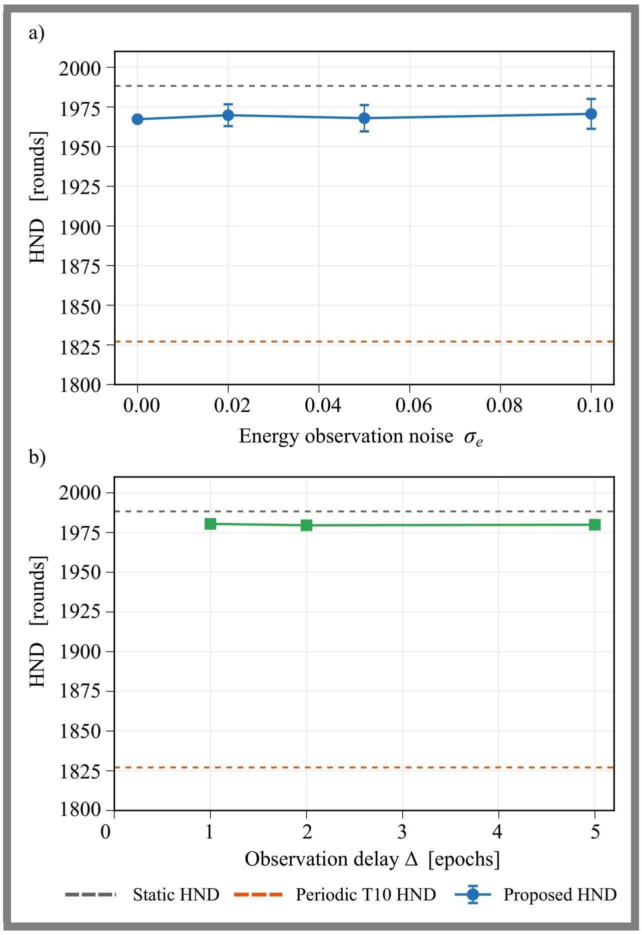

Fig. 6. HND of the proposed controller under residual energy observation noise and delayed network state observation.

The results indicate that the low cluster count options, especially the three- and four-groups, are useful under the evaluated deployment geometry and energy model. When the candidate set starts at 4 or 5 clusters, the controller is forced toward larger topologies, increasing control energy cost and reducing HND. The cluster count set is therefore an engineering design choice that should match the node density, the sensor field size, and the placement of the base station.

6.8. Training, Runtime, and Deployment Complexity

The controller uses a 13-dimensional state vector. Both the actor and critical models use two hidden layers with 256 units, LayerNorm, and ReLU activation. The actor has three output heads for the refresh indicator u, cluster count C, and holding time d. Adam is used with a learning rate of 3×10^{-4} and a discount factor of 0.99. The implemented model has 73 741 actor parameters and 70 657 critic parameters, for a total of 144 398.

The runtime profile separates off-line training from online controller execution (Fig. 8). Policy inference takes 2476 ±797 µs per control call, with a 95-th percentile of 3628 µs. Topology construction takes 147.6 ±38.4 ms per refresh event, with a 95-th percentile of 197.4 ms. These values are measured on the controller side. The sensor nodes do not execute the learning-based controller. They only receive the topology configuration and follow the resulting cluster assignment and forwarding schedule.

7. Conclusions

This article studied clustered WSN topology maintenance from the perspective of adaptive holding time. The controller decides whether to refresh the topology, which cluster count to use after the refresh, and how long the resulting topology control decision should remain active. The lower layer topology builder is specified through cluster head scoring, nearest head member association, and chain-assisted intra-cluster forwarding with one-hop cluster head transmission to the base station.

The results show a consistent lifetime-overhead trade-off. Under nominal deployment, the proposed controller improves HND from 1819.7 ±32.6 rounds with periodic refresh to 1969.1 ±8.4 rounds, while reducing the control energy from 1.133 J to 0.104 J. Across the seven tested deployment scenarios, the proposed method provides a better HND overhead operating point than the tested refresh enabled baselines. The state observation tests also show limited sensitivity to the evaluated residual energy noise levels and short observation delays.

The operating gain is not universal across all metrics. The proposed controller does not maximize FND and does not preserve connectivity-based delivery or periodic refresh. In the far base station scenario, the delivery value is low even though HND remains high. Therefore, adaptive holding time should be interpreted as a lifetime overhead control mechanism for the considered clustered chain topology model, not as a general QoS-maximizing topology policy.

Future work should extend the evaluation to MAC layer packet scheduling, dynamic traffic, mobility, and external learning-based clustering baselines under matched topology and energy accounting models.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 38, pp. 393–422, 2002 (https://doi.org/10.1016/S1389-1286(01)00302-4).

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey", *Computer Networks*, vol. 52, pp. 2292–2330, 2008 (https://doi.org/10.1016/j.comnet.2008.04.002).

[3] M.M. Afsar and M.H. Tayarani-N, "Clustering in Sensor Networks: A Literature Survey", *Journal of Network and Computer Applications*, vol. 46, pp. 198–226, 2014 (https://doi.org/10.1016/j.jnca.2014.09.005).

[4] I. Daanoune, B. Abdennaceur, and A. Ballouk, "A Comprehensive Survey on LEACH-based Clustering Routing Protocols in Wireless Sensor Networks", *Ad Hoc Networks*, vol. 114, art. no. 102409, 2021 (https://doi.org/10.1016/j.adhoc.2020.102409).

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks", *Proc. of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 3005–3014, 2000 (https://doi.org/10.1109/HICSS.2000.926982).

[6] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks",

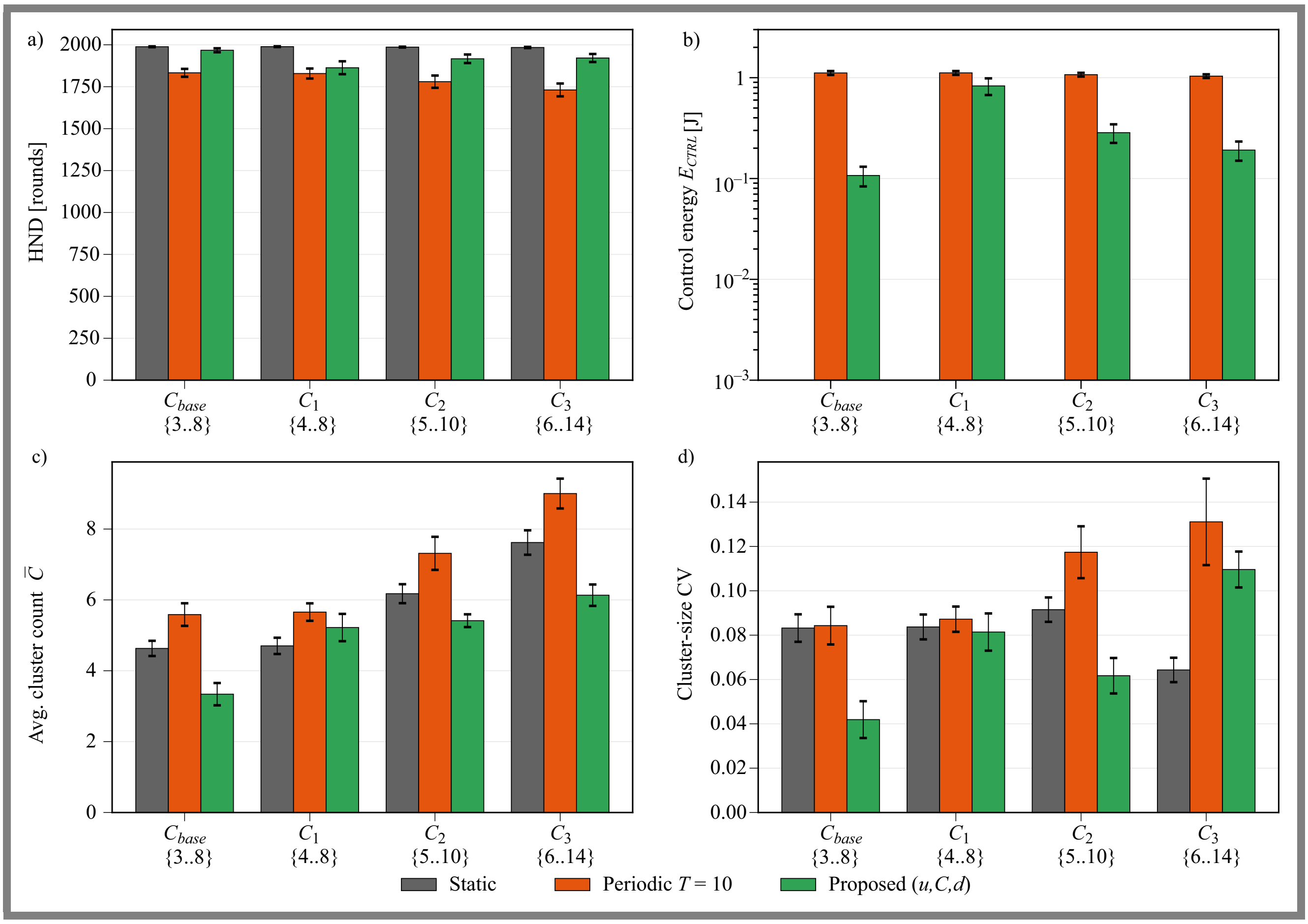

Fig. 7. Sensitivity of the cluster count candidate set. The proposed controller performs best with the evaluated baseline set, because it can select low cluster counts when fewer cluster heads are sufficient under this deployment model.

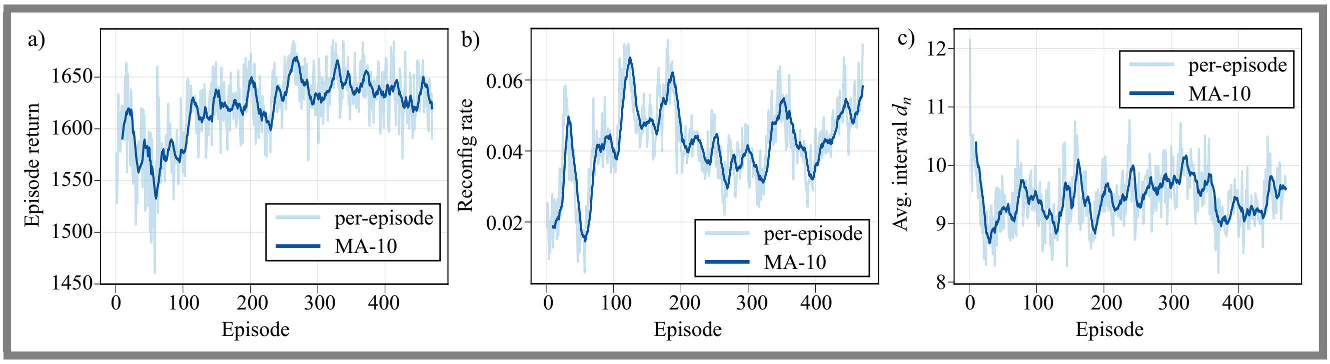

Fig. 8. Convergence of factorized policy.

IEEE Transactions on Mobile Computing, vol. 3, pp. 366–379, 2004 (https://doi.org/10.1109/TMC.2004.41).

[7] M. Saadati, S.M. Mazinani, A.A. Khazaei, and S.J.S.M. Chabok, "Energy Efficient Clustering for Dense Wireless Sensor Network by Applying Graph Neural Networks with Coverage Metrics", *Ad Hoc Networks*, vol. 156, art. no. 103432, 2024 (https://doi.org/10.1016/j.adhoc.2024.103432).

[8] M. Shokouhifar *et al*, "AI-driven Cluster-based Routing Protocols in WSNs: A Survey of Fuzzy Heuristics, Metaheuristics, and Machine Learning Models", *Computer Science Review*, vol. 54, art. no. 100684, 2024 (https://doi.org/10.1016/j.cosrev.2024.100684).

[9] B. Fan and Y. Xin, "EBPT-CRA: A Clustering and Routing Algorithm Based on Energy-balanced Path Tree for Wireless Sensor Networks", *Expert Systems with Applications*, vol. 259, art. no. 125232, 2025 (https://doi.org/10.1016/j.eswa.2024.125232).

[10] S. Najjar, M. David, W. Derigent, and A. Zouinkhi, "Dynamic Reconfiguration of Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 262, art. no. 111176, 2025 (https://doi.org/10.1016/j.comnet.2025.111176).

[11] C. Wang, H. Hu, and X. Fan, "Intelligent Clustering and Routing Protocol for Wireless Sensor Networks Using Quantum Inspired Harris Hawk Optimizer and Deep Reinforcement Learning", *Ad Hoc Networks*, vol. 178, art. no. 103914, 2025 (https://doi.org/10.1016/j.adhoc.2025.103914).

[12] N. Mittal, U. Singh, and B.S. Sohi, "A Stable Energy Efficient Clustering Protocol for Wireless Sensor Networks", *Wireless Networks*, vol. 23, pp. 1809–1821, 2017 (https://doi.org/10.1007/s11276-016-1255-6).

[13] B. Zebbane, M. Chenait, and N. Badache, "A Distributed Lightweight Redundancy Aware Topology Control Protocol for Wireless Sensor

Networks", *Wireless Networks*, vol. 23, pp. 1779–1792, 2017 (https://doi.org/10.1007/s11276-016-1248-5).
[14] H.R. Farahzadi, M. Langarizadeh, M. Mirhosseini, and S.A.F. Aghda, "An Improved Cluster Formation Process in Wireless Sensor Network to Decrease Energy Consumption", *Wireless Networks*, vol. 27, pp. 1077–1087, 2021 (https://doi.org/10.1007/s11276-020-02485-y).
[15] T. Nguyen, T.-A. Nguyen, and T.-M. Hoang, "A Study on Multi-hop Routing Scheme for Wireless Sensor Networks", *JST: Smart Systems and Devices*, vol. 31, pp. 1–9, 2021 (https://doi.org/10.51316/jst.152.ssad.2021.31.2.1).
[16] Z. Zhang, J. Zhou, and J. Li, "Q-learning-based Semi-fixed Clustering Routing Algorithm in WSNs", *Ad Hoc Networks*, vol. 174, art. no. 103837, 2025 (https://doi.org/10.1016/j.adhoc.2025.103837).
[17] R.S. Sutton, D. Precup, and S. Singh, "Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning", *Artificial Intelligence*, vol. 112, pp. 181–211, 1999 (https://doi.org/10.1016/S0004-3702(99)00052-1).
[18] A. S. Lakshminarayanan, S. Sharma, and B. Ravindran, "Dynamic action repetition for deep reinforcement learning", *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 31, pp. 2133–2139, 2017 (https://doi.org/10.1609/aaai.v31i1.10918).

Trong-Minh Hoang, Ph.D.
https://orcid.org/0000-0001-8486-2940
E-mail: hoangtrongminh@ptit.edu.vn
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
https://ptit.edu.vn

Thanh-Long Tran, Ph.D.
https://orcid.org/0009-0006-5843-6044
E-mail: longtt.product@gmail.com
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
https://ptit.edu.vn

Huy-Long Tran, Ph.D.
https://orcid.org/0009-0008-2968-3905
E-mail: longth@ptit.edu.vn
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
https://ptit.edu.vn

Ngoc-Bich Pham, Ph.D.
https://orcid.org/0009-0002-3302-5813
E-mail: bichpn@thanglong.edu.vn
Thang Long University, Hanoi, Vietnam
https://thanglong.edu.vn

Sinh Cong Lam, Ph.D.
https://orcid.org/0000-0003-4546-3378
E-mail: congls@vnu.edu.vn (corresponding author)
VNU University of Engineering and Technology, Hanoi, Vietnam
https://uet.vnu.edu.vn