Protocol-Aware Hybrid Clustering for IoT: Adaptive Reconfiguration and Secure Communication with MQTT/CoAP Integration

Osama Mohammed Dighriri
School of Electrical and Data Engineering,
Faculty of Engineering and IT,
University of Technology Sydney, Australia
Department of Computer Science,
College of Engineering and Computer Science,
Jazan University, Jazan, Saudi Arabia
osamamohammeda.dighriri@student.uts.edu.au

Priyadarsi Nanda
School of Electrical and Data Engineering,
Faculty of Engineering and IT,
University of Technology Sydney, Australia
priyadarsi.nanda@uts.edu.au

Manoranjan Mohanty
School of Information Systems,
Carnegie Mellon University,
Doha, Qatar
mmohanty@andrew.cmu.edu

Bashair Alrashed
College of Computer Science and Engineering,
University of Jeddah,
Jeddah, Saudi Arabia
baalrashed@uj.edu.sa

Ibrahim Haddadi
Department of Computer Engineering,
College of Computer Science and Engineering,
Taibah University, Medina, Saudi Arabia
ihaddadi@taibahu.edu.sa

***Abstract*—The fast evolution of Internet of Things (IoT) is placing increased demands on network infrastructures to be versatile and robust, especially in dynamic, heterogeneous, and resource-limited environments. Existing cluster-based and communication protocols struggle with protocol rigidity, insufficient integrated security, and limited reconfigurability. This paper proposes a novel security-aware hybrid clustering framework integrating BIRCH-DBSCAN algorithms, MQTT/CoAP switching adaptively, and AES-128 encryption with session-based key rotation for end-to-end confidentiality. By featuring a three-layer architecture designed with autonomous cluster recovery, layered verification, and a reconfiguration system upon performance, energy, and mobility changes. Evaluated and tested on Contiki-NG simulation, the approach provides 43.3% latency reduction, 22.8% energy efficiency improvement, 99.91% delivery reliability, and zero breaches over 39 adaptive switches with just 4.2% overhead. The results attest to the platform's strength and viability for future IoT deployments for efficient and responsive communications in changing conditions.**

***Index Terms*—IoT Security, Hybrid Clustering, Protocol-Aware Adaptation, AES-128, Adaptive Reconfiguration, Contiki-NG.**

I. Introduction

As the Internet of Things (IoT) devices are expected to increase to 29 billion by 2030 [11], these systems will generate massive amounts of data [4], exerting colossal pressure on networks to support extreme heterogeneity, constrained resources, and intensifying cybersecurity threats [19].

Despite ongoing research, three critical challenges remain under-addressed: (1) *Scalability-security tradeoffs*, where traditional clustering protocols fail in dynamic environments like vehicular or high-density IoT, struggling to balance encrypted transmission with energy efficiency [5]; (2) *Protocol rigidity*, as most IoT stacks remain static and cannot adapt to shifting latency, reliability, or traffic demands in fog-to-cloud settings [3]; and (3) *Resilience gaps*, with many systems lacking redundancy or autonomous recovery, exposing critical applications to disruptions [19].

Traditional cluster-based protocols such as Low-Energy Adaptive Clustering Hierarchy (LEACH) and Hybrid Energy-Efficient Distributed Clustering (HEED) [8], [13] offer energy efficiency but lack responsiveness or built-in security. Similarly, lightweight communication protocols like Message Queuing Telemetry Transport (MQTT) and the Constrained Application Protocol (CoAP) [10] enable efficient transmission, but typically fall short on adaptability and protection against evolving threats [18]. To address these limitations, we propose a *security-aware protocol clustering framework* for secure IoT environments. The framework integrates Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering, adaptive MQTT/CoAP switching, and end-to-end Advanced Encryption Standard (AES)-128 encryption to support secure, context-aware communication.

The major contributions include: (1) a *security-integrated hybrid clustering mechanism* combining BIRCH's efficiency and DBSCAN's noise resistance, augmented with encryption and threat monitoring; (2) a *secure adaptive protocol selection algorithm* that switches between MQTT and CoAP based on security context, energy profiles, and network conditions; and (3) a *resilient secure architecture* implementing AES-128 with dynamic key rotation and session management for sustained confidentiality.

II. Related Work

Traditional cluster protocols such as LEACH [15] and HEED [6] are widely employed in IoT due to their energy efficiency. However, both lack integrated security and adaptability, making them unsuitable for threat-aware or rapidly changing environments. BIRCH and DBSCAN [2] provide memory efficiency and noise tolerance, respectively, offering complementary benefits for IoT clustering. Yet, most hybrid approaches neglect secure-by-design strategies. Our framework addresses this gap by embedding cryptographic constraints and threat metrics within the clustering process to support trust-aware formation and operation.

MQTT and CoAP [12] are commonly used for lightweight communication in constrained IoT settings. However, current deployments adopt them as rigid stacks with no adaptation to shifting latency, reliability, or context-aware needs. Tripathi et al. [18] evaluated MQTT performance across QoS levels, and Hu et al. [7] examined protocol-layer security at the edge, but neither supports runtime switching or adaptability. Our system extends this line by enabling context-driven protocol switching while maintaining encryption continuity and resilience.

End-to-end protection remains limited in cyber-physical IoT systems, as cryptographic mechanisms are often decoupled from runtime dynamics. Ferretti et al. [14] studied the efficiency of AES and similar algorithms, but integration within a real-time adaptive framework remains sparse. Montasari [11] emphasized the national importance of IoT security, while Wyss et al. [19] underlined the need for scalable QoS and self-healing properties. Our solution consolidates these goals by introducing a hierarchical, secure, and protocol-aware architecture that supports reconfiguration, session agility, and attack resilience.

III. Proposed Framework

Our research addresses the core challenges of heterogeneous IoT environments through a comprehensive three-layered framework that integrates dynamic clustering, adaptive communication protocols, and secure data transmission. The system is designed to accommodate evolving requirements in real-time, optimizing for both performance and protection.

A. System Architecture

The proposed system employs a hierarchical three-layer architecture, as illustrated in Figure 1.

The **Edge Layer** serves as the root level and maintains global network statistics, orchestrates inter-cluster communication, and executes high-level routing logic. It also processes summarized data and oversees hybrid clustering configuration.

The **Cluster Center Layer** includes dynamically elected centers that coordinate intra-cluster communication, data aggregation, and local decision-making. These centers are selected based on multi-factor criteria including residual energy, historical responsiveness, connectivity index, and latency stability.

The **Client Node Layer** consists of low-power sensor nodes tasked with metric collection and basic communication. These nodes rely on lightweight protocols and offload computation to their respective cluster centers, enabling energy preservation and responsive behavior.

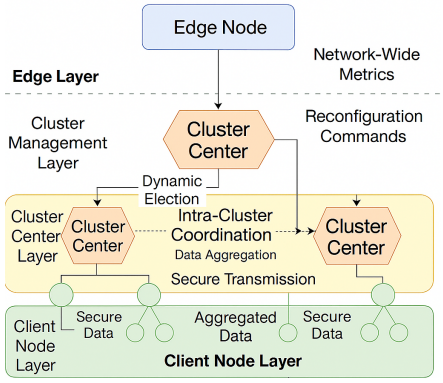

Fig. 1. Three-layer hierarchical architecture of the proposed framework.

B. Hybrid Clustering Mechanism

The framework employs a dual-phase clustering strategy that balances computational efficiency and real-time adaptability in resource-constrained IoT networks. It leverages the strengths of two complementary algorithms—BIRCH and DBSCAN—across two key stages: *initial cluster formation* and *boundary refinement*.

In the first phase, BIRCH performs scalable and memory-efficient clustering by summarizing node attributes into compact feature vectors. This allows rapid initial grouping based on location, usage frequency, and mobility patterns. The second phase applies DBSCAN to refine cluster boundaries by considering spatial density and noise tolerance. This two-step process ensures robust cluster formation even in heterogeneous or dynamic topologies.

Each cluster center operates as a distributed coordination point, maintaining full node profiles that include relocation history, traffic intensity, energy metrics, latency logs, and communication patterns. These detailed profiles enable intelligent protocol selection, routing optimization, and dynamic resource management. Clusters are continuously evaluated and updated through periodic observation, allowing real-time adjustments in response to network changes.

The formation logic also considers environmental factors such as node mobility, density variation, and energy distribution. This enables optimized and resilient clustering for diverse IoT applications, including mobile sensing, event detection, and constrained smart environments.

C. QoS-Driven Reconfigurability

The system enables QoS-driven reconfiguration via three dynamic triggers: (1) *Energy-based*, which activates cluster center re-election and load redistribution when residual energy falls below a threshold; (2) *Performance-based*, which adjusts cluster boundaries and routing paths in response to reduced packet delivery or increased latency; and (3) *Topology-based*,

which reconfigures clusters following mobility events or node failures.

A timer-based synchronization mechanism governs periodic updates across all layers. Each reconfiguration cycle ensures the regeneration of AES-128 encryption keys to preserve session integrity and security alignment with the updated cluster layout.

D. Secure Communication

The framework ensures protocol-agnostic, end-to-end secure communication across all layers, optimized for the heterogeneous nature of IoT systems. It combines multi-level encryption with lightweight, adaptive policies that respond to shifting network conditions and threat dynamics.

Security is anchored by a hierarchical key management scheme: cluster centers function as local security coordinators, managing session keys and enforcing authentication for intra-cluster communications. This decentralized model distributes security overhead, balancing protection with resource constraints.

Adaptive protection policies are triggered based on dynamic assessments of node capabilities, residual energy, and localized threat metrics. Each reconfiguration cycle includes AES-128 key regeneration, mitigating risks of key exhaustion or reuse. The system reacts to anomalies such as packet injection, replay patterns, or communication dropouts by increasing key rotation frequency or triggering selective re-authentication.

This layered, adaptive security model provides consistent protection without compromising efficiency, and ensures resilience against evolving attack vectors.

E. Adaptive Protocol Selection

The communication layer supports intelligent switching between MQTT and CoAP protocols based on real-time performance metrics. The decision logic evaluates three key parameters: energy availability, traffic volume, and latency sensitivity, allowing the system to adapt to diverse application demands across varying deployment conditions.

Protocol decisions are guided by a scoring function that monitors node health, message patterns, and delivery feedback in real time. Switching is triggered only when the performance gap between the current and alternate protocol exceeds a defined threshold, and a cooldown timer is enforced to prevent frequent oscillation.

This adaptive mechanism enables seamless transitions without disrupting ongoing sessions, ensuring stable and optimized communication throughout the system lifecycle. The result is a balanced trade-off between energy efficiency and service quality, tailored to the evolving requirements of heterogeneous IoT nodes.

F. Mathematical Formulation

The proposed framework employs a multi-phase decision model integrating hybrid clustering, adaptive protocol switching, and AES-128 encryption. This formulation mathematically governs the behavior of each network node and cluster under dynamic conditions. All notations are summarized in Table I.

TABLE I
MATHEMATICAL NOTATION AND DEFINITIONS

Symbol	Description
n_i, n_j	IoT nodes in the network
x_i, y_i	Coordinates of node n_i
$d(n_i, n_j)$	Euclidean distance between nodes
C_k	Cluster k
$\|\mathcal{C}\|$	Total number of active clusters
$\mathcal{C}$	Set of all clusters
$\mathcal{C}_0$	Initial clusters from BIRCH
c	Preliminary cluster
CF	Cluster feature vector (N, LS, SS)
N	Number of nodes in the cluster
LS, SS	Linear and squared sums of node attributes
L_k, T_k, E_k	Latency, throughput, energy of C_k
$L_{\max}$, $T_{\max}$, $E_{\max}$	Maximum latency, throughput, energy
$QoS(C_k)$	Cluster quality of service score
α, β, γ	QoS weighting factors
μ_{qos}	Minimum acceptable QoS
σ	Sensitivity parameter
$\overline{QoS}$	Average historical QoS
θ_{qos}	Reconfiguration threshold
R_k	Reconfiguration state of cluster C_k
τ_{sync}	Synchronization timer interval
μ_e, μ_p, μ_t	Energy, performance, topology thresholds
DR_k	Message delivery rate for cluster C_k
Δ_{topo}	Topology change magnitude
p	Protocol (MQTT or CoAP)
ε_p	Protocol efficiency score
l_p, t_p, e_p	Latency, throughput, energy for protocol p
w_l, w_t, w_e	Protocol metric weights
δ	Protocol switching threshold
$t_{current}, t_{last\_switch}$	Timestamps
$T_{cooldown}$	Cooldown interval between switches
M_i, C_i	Plaintext and AES-128 ciphertext
K	Symmetric AES-128 session key
Γ	Global system QoS score
D	Input dataset for clustering
$\Pi(n_i)$	Protocol assignment for node n_i
ρ_i	Threat probability for node n_i

a) Cluster Feature Vector (BIRCH):

$$CF = (N, LS, SS) \tag{1}$$

b) Node Distance Calculation (DBSCAN):

$$d(n_i, n_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2}$$

c) Cluster QoS Score:

$$QoS(C_k) = \alpha \left(1 - \frac{L_k}{L_{\max}}\right) + \beta \frac{T_k}{T_{\max}} + \gamma \left(1 - \frac{E_k}{E_{\max}}\right) \tag{3}$$

d) QoS Threshold for Cluster Reconfiguration:

$$\theta_{qos} = \mu_{qos} - \sigma \cdot \overline{QoS} \tag{4}$$

e) Reconfiguration Trigger Conditions:

$$R_k = \begin{cases} 1 & \text{if } E_k < \mu_e \text{ or } DR_k < \mu_p \text{ or } \Delta_{topo} > \mu_t \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

f) Protocol Efficiency Score:

$$\varepsilon_p = w_l \left(1 - \frac{l_p}{L_{\max}}\right) + w_t \frac{t_p}{T_{\max}} + w_e \left(1 - \frac{e_p}{E_{\max}}\right) \tag{6}$$

g) Protocol Switching Condition:

$$|\varepsilon_{\text{MQTT}} - \varepsilon_{\text{CoAP}}| > \delta \quad \text{and} \quad (t_{current} - t_{last\_switch}) > T_{cooldown} \quad (7)$$

h) AES-128 Encryption and Decryption:

$$C_i = \text{AES128\_Enc}(M_i, K), \quad M_i = \text{AES128\_Dec}(C_i, K) \quad (8)$$

i) Global System Performance:

$$\Gamma = \frac{1}{|\mathcal{C}|} \sum_{k=1}^{|\mathcal{C}|} QoS(C_k) \quad (9)$$

j) System Parameter Initialization:

$$\alpha + \beta + \gamma = 1, \quad w_l + w_t + w_e = 1 \quad (10)$$

k) Adaptive Security Policy:

$$\Phi(n_i) = \psi(E_i, \rho_i, DR_k) \quad (11)$$

The formulation integrates protocol adaptability, clustering stability, and encryption continuity into a cohesive model. Thresholds (δ, μ_e, etc.) are empirically derived and updated dynamically to reflect operating environments. Equation 7 incorporates a hysteresis effect and a cooldown interval ($T_{cooldown}$) to prevent protocol oscillation. Cluster reconfigurations are guided by dynamic θ_{qos} and conditionally triggered to maintain efficiency. The global QoS score Γ reflects system scalability under increasing cluster cardinality $|\mathcal{C}|$, ensuring decentralized adaptation as node density grows. The optional security policy in Eq. 11 enables threat-responsive behavior by adjusting protection levels based on local conditions. This formulation serves as the computational foundation of the secure, reconfigurable IoT architecture.

G. Algorithm Design

Algorithm 1 initializes the secure network infrastructure with reconfigurability support. The `InitializeSystem` procedure sets core parameters, deploys AES-128 symmetric key K, and defines reconfiguration thresholds for energy, performance, and topology. The `HybridClustering` routine executes two-stage clustering: BIRCH forms initial groups using CF vectors (Eq. 1), followed by DBSCAN refinement based on distance metrics (Eq. 2). Clusters are assigned $R_k = 0$, and node metadata is AES-encrypted to ensure confidentiality.

Algorithm 2 governs runtime operations and enables dynamic reconfiguration. The `EvaluateClusterQoS` module measures $QoS(C_k)$ (Eq. 3) and checks triggers (Eq. 5). If thresholds are violated, `ReconfigureCluster` is invoked. This synchronizes updates to prevent storms and adjusts clusters via re-election, DBSCAN reruns, or routing table updates. Each reconfiguration regenerates AES keys to maintain confidentiality. `AdaptiveProtocolSelection` compares MQTT/CoAP efficiency (Eq. 6) and switches protocols when δ deviation and cooldown conditions (Eq. 7) are met. Messages are secured with AES (Eq. 8). `CalculateGlobalPerformance` aggregates $QoS(C_k)$ (Eq. 9) for network health monitoring and global tuning.

Algorithm 1 System Initialization and Hybrid Clustering

procedure INITIALIZESYSTEM
  Init routing tables, protocol states, monitoring timers
  Generate and distribute AES-128 key K
  Set weights: $\alpha + \beta + \gamma = 1$, $w_l + w_t + w_e = 1$
  Configure thresholds: θ_{qos}, δ, $T_{cooldown}$, μ_e, μ_p, μ_t
  Start synchronization timer τ_{sync}
end procedure
procedure HYBRIDCLUSTERING(D)
  Extract node features, $\mathcal{C}_0 \leftarrow$ BIRCH(CF)
  $\mathcal{C} \leftarrow \emptyset$
  for each $c \in \mathcal{C}_0$ **do**
    Refine with DBSCAN using $d(n_i, n_j)$ (Eq. 2)
    Remove noise, finalize C_k
    $R_k \leftarrow 0$
    for each $n_i \in C_k$ **do**
      Encrypt M_i: $C_i \leftarrow AES128\_Enc(M_i, K)$
    end for
    Append C_k to $\mathcal{C}$
  end for
  return $\mathcal{C}$
end procedure

Algorithm 2 QoS Monitoring with Adaptive Reconfiguration

procedure EVALUATECLUSTERQOS(C_k)
  Measure L_k, T_k, E_k, DR_k; compute $QoS(C_k)$
  Check triggers (Eq. 5)
  if $QoS(C_k) < \theta_{qos}$ or $R_k = 1$ **then**
    Call RECONFIGURECLUSTER(C_k)
  end if
end procedure
procedure RECONFIGURECLUSTER(C_k)
  Sync broadcast, await ACKs
  if $E_k < \mu_e$ **then**
    Elect new cluster center, redistribute
  else if $DR_k < \mu_p$ **then**
    Optimize boundaries, adjust power
  else if $\Delta_t > \mu_t$ **then**
    Re-run DBSCAN, update routing
  end if
  Distribute new AES key K_{new}, reset $R_k \leftarrow 0$
end procedure
procedure ADAPTIVEPROTOCOLSELECTION(n_i)
  Monitor l_p, t_p, e_p; compute ε_p
  if Switch criteria met (Eq. 7) **then**
    $\Pi(n_i) \leftarrow \arg\max(\varepsilon_p)$
  end if
  Encrypt: $C_i \leftarrow AES128\_Encrypt(M_i, K)$
  Transmit C_i via $\Pi(n_i)$
end procedure
procedure CALCULATEGLOBALPERFORMANCE($\mathcal{C}$)
  $\Gamma \leftarrow \frac{1}{|\mathcal{C}|} \sum_{k=1}^{|\mathcal{C}|} QoS(C_k)$
end procedure

IV. Implementation Details

A. Simulation Environment

The proposed framework was prototyped using Contiki-NG, an advanced operating system designed for IoT deployments involving memory-constrained devices. Its modular stack enables full support for lightweight protocols (e.g., CoAP, MQTT), multi-hop communication, and integrated security [17]. The simulation was executed through the Cooja network emulator, which replicates low-power wireless communications and realistic energy models, offering fine-grained control over radio and sensor behaviors. The experimental setup models a mesh topology with 61 nodes deployed in a

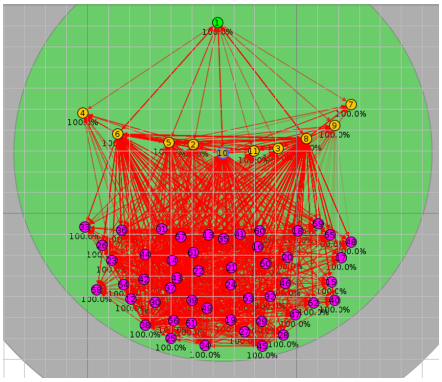

Fig. 2. Contiki-NG simulation environment showing network topology with edge root (green), cluster centers (yellow), and client nodes (purple).

$100\times100\,\mathrm{m}^2$ area as depicted in Figure 2, reflecting realistic urban IoT environments. The node roles are as follows: A single **edge root node** (green) acts as the coordination gateway and protocol switch controller; Ten **cluster centers** (yellow) handle local aggregation, hybrid clustering, and adaptive protocol control; Fifty **client nodes** (purple) emulate diverse IoT device types, ranging in energy capacity and link stability. Devices engage in reconfigurable clustering and secure communication based on the framework logic, dynamically shifting between MQTT and CoAP when system conditions demand it. However, Real-time metrics including latency, delivery success, node energy usage, and throughput are monitored using instrumented Contiki timers and Cooja plugins. Key events (e.g., re-clustering, protocol switch, AES key updates) are logged across both central and cluster-local interfaces, enabling visualization of response patterns. The simulator was also modified to support periodic cluster reshaping under topology or performance triggers.

B. Clustering Implementation

The hybrid clustering mechanism combines the strengths of BIRCH and DBSCAN to ensure both scalability and noise resilience—two essential qualities for dynamic and large-scale IoT environments. The algorithm executes in two stages: an initial coarse-grained cluster formation using BIRCH, followed by fine-grained boundary refinement through DBSCAN.

BIRCH-Based Formation: BIRCH builds an initial clustering tree using Cluster Feature (CF) vectors computed incrementally from incoming node attributes (Eq. 1). This method supports memory-efficient clustering by summarizing data in CF subclusters without full dataset traversal. For IoT-specific adaptation, we tuned the branching factor and threshold radius to reflect memory and latency trade-offs. A background process compacts sparse or inactive CF nodes to optimize memory usage during runtime.

DBSCAN Boundary Refinement: DBSCAN is applied selectively to clusters formed by BIRCH—particularly those near density boundaries or in transitional zones. Using spatial indexing and Euclidean distance (Eq. 2), DBSCAN reclassifies noisy nodes, adjusts boundaries, and merges or splits clusters based on local density thresholds. The ε parameter is adapted dynamically using CF variance statistics, minimizing fragmentation while preserving spatial integrity.

Dynamic Cluster Maintenance: The system continuously adapts clusters through real-time maintenance. Cluster centers are re-elected based on multi-metric criteria including residual energy, connectivity index, and historical stability. These factors support long-term cluster reliability while minimizing reorganization overhead. Cluster updates are incremental—no complete CF-tree reconstruction is required. Node joins/splits are automatically managed based on membership deviation from predefined thresholds, maintaining optimal size and topology awareness without global resets.

C. Security Implementation

The framework employs AES-128 encryption at the application layer using Contiki-NG's built-in crypto libraries. This balances confidentiality with computational efficiency under constrained environments. All messages are secured before protocol-specific processing to ensure end-to-end integrity.

Key Management: A hierarchical key model is used, where each cluster center distributes symmetric session keys to its client nodes based on a shared master secret. Session keys are generated via cluster-local random number generators and periodically rotated using a time-triggered or event-based mechanism tied to cluster stability. This ensures forward secrecy while minimizing key management overhead. If hardware encryption is unavailable, the system defaults to software-based memory isolation with secure credential storage.

Encryption and Integrity: Messages are padded using PKCS #7 to align with AES block size and encrypted using CBC mode. Hash-Based Message Authentication Code (HMAC) ensures data authenticity and protects against tampering and replay attacks. For efficiency, only sensitive payload components are ciphered, and in-place memory operations are used to reduce encryption overhead during runtime.

Threat Monitoring and Adaptive Response: Each node maintains lightweight security triggers that monitor communication anomalies, packet loss patterns, and replay attempts. Upon detecting a potential violation, the framework initiates countermeasures including session key regeneration, re-authentication of cluster membership, and local cluster reconfiguration. This ensures resilience against real-time threats such as man-in-the-middle (MITM), replay, and impersonation attacks.

D. Protocol Integration

MQTT and CoAP were integrated into the Contiki-NG application layer through custom extensions to support protocol-aware adaptability. The system enables dynamic, metric-driven switching between them without service interruption, based on Equation (6). Both protocol stacks operate in standby, and nodes alternate roles seamlessly according to runtime thresholds.

MQTT Integration: A lightweight broker is embedded at cluster center nodes, enabling a decentralized publish-subscribe model. It supports QoS levels 0 and 1, with message

queues ensuring resilience against short-term link loss. Topic namespaces are managed locally at each center, with inter-cluster routing through the edge root node to reduce broadcast flooding and ensure structured topic propagation.

CoAP Integration: CoAP is implemented per RFC 7252, supporting both confirmable and non-confirmable messages. Message types are automatically selected based on sensed urgency and network feedback. CoAP's discovery features are enhanced by cluster-based routing tables to improve service location without global broadcasts.

Adaptive Switching Control: The switching decision engine tracks energy, message rate, payload size, and latency. CoAP is selected when energy drops below 20% or latency requirements are strict; MQTT is preferred when the payload exceeds 256 bytes or message frequency surpasses 3 per minute. To prevent protocol oscillation, a cooldown interval of 60 seconds is enforced, and switching occurs only if the efficiency delta exceeds threshold δ (Equation 7). This mitigates unnecessary toggling and reduces overhead.

E. Latency Performance Analysis

Latency remains a pivotal QoS metric for time-sensitive IoT applications, particularly in actuator control, intrusion detection, and emergency response. Figure 3 presents end-to-end latency comparisons across fixed and adaptive configurations. Achieved an average latency of 210 ms ($\sigma = 52$ ms), outperforming static MQTT (370 ms $\pm$ 89 ms) and fixed CoAP (285 ms $\pm$ 67 ms) by 43.3% and 26.3%, respectively. These measurements encompass all layers, including transmission, queuing, and AES encryption/decryption overhead (2.0 ms and 1.2 ms). Compared to LEACH-RLC (298 ms) [8] and LAECIPS (265 ms) [7], the proposed system improves latency by 29.5% and 20.8%, respectively. Gains stem from: (1) protocol switching aligned with traffic and energy dynamics, (2) hybrid clustering that minimizes hop counts, and (3) responsive reconfiguration that mitigates transient congestion.

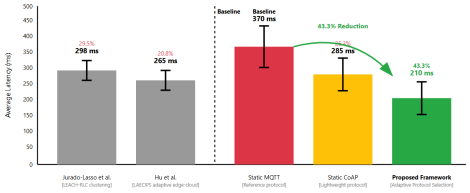

Fig. 3. Latency performance comparison showing average latency with standard deviation error bars and percentage improvements over baselines.

F. Energy Efficiency Analysis

Energy efficiency is critical for IoT deployments where many devices are battery-constrained. As shown in Figure 4, the proposed framework consumed 6,622 mJ in total, representing a 22.8% reduction compared to static MQTT (8,578 mJ), and lower than two recent works: Bideh et al. [1] (7,890 mJ, 16.1% higher) and Amirhossein et al. [16] (7,234 mJ, 8.5% higher). The total energy accounts for all components, including CPU, radio TX/RX, listening, and AES-based security (278 mJ or 4.2%). Our results align with Bideh et al., who found that MQTT retransmission yields power savings in lossy environments, and Amirhossein et al., who observed that MQTT-SN is most efficient, followed by CoAP and MQTT. Our gains, however, come from adaptive protocol switching that dynamically exploits such trade-offs based on node state. Compared to static MQTT, per-component savings were observed in CPU (from 1,200 mJ to 860 mJ), Radio TX (2,200 mJ to 1,595 mJ), RX (1,500 mJ to 1,130 mJ), and listening operations (3,678 mJ to 3,037 mJ). Lightweight CoAP is triggered under 20% battery, while MQTT is used for larger payloads and event bursts—yielding a balanced energy-performance profile across conditions.

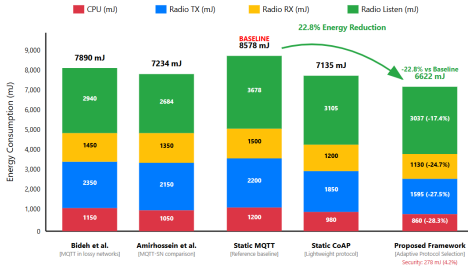

Fig. 4. Energy consumption breakdown by component showing CPU, Radio TX/RX, and Listen operations across approaches.

G. Message Delivery Success Rate

Our proposed framework demonstrated 99.91% message delivery success across 2,154 connections with just two failures—one caused by a timeout and the other by a collision. This performance largely surpasses both baseline and state-of-the-art approaches, as illustrated in Table II. Tripathi et al. [18] achieved 97.8% using only MQTT under varied QoS settings. Barati et al. [13] reached 98.2% via a hybrid learning-based forwarding method, while Kim and Lee [9] obtained 96.5% using an adaptive sensor relocation technique. In comparison, our solution achieves improvements of 2.11%, 1.71%, and 3.41%, respectively. This high reliability stems from the real-time protocol adaptation mechanism, which selects the most effective communication mode based on instantaneous network conditions. Smart retransmission control and path optimization techniques further enhance this success rate.

H. Protocol Adaptation Analysis

The system's adaptive logic successfully executed 39 seamless protocol transitions during simulation, achieving a 100% transition reliability under volatile conditions. As shown in Figure 5, the proposed mechanism outperformed comparable schemes such as Yang et al. [20], who reported a 96.9% switching accuracy using dynamic routing in heterogeneous IoT networks. Operationally, three dominant adaptation phases were observed: energy-constrained intervals (33–67 min), high-traffic bursts (100–125 min), and battery depletion stages (142–167 min). Transitions were triggered by energy drops (18 times), burst traffic (12), latency-sensitive operations (6), and congestion-induced delays (3). These events activated protocol

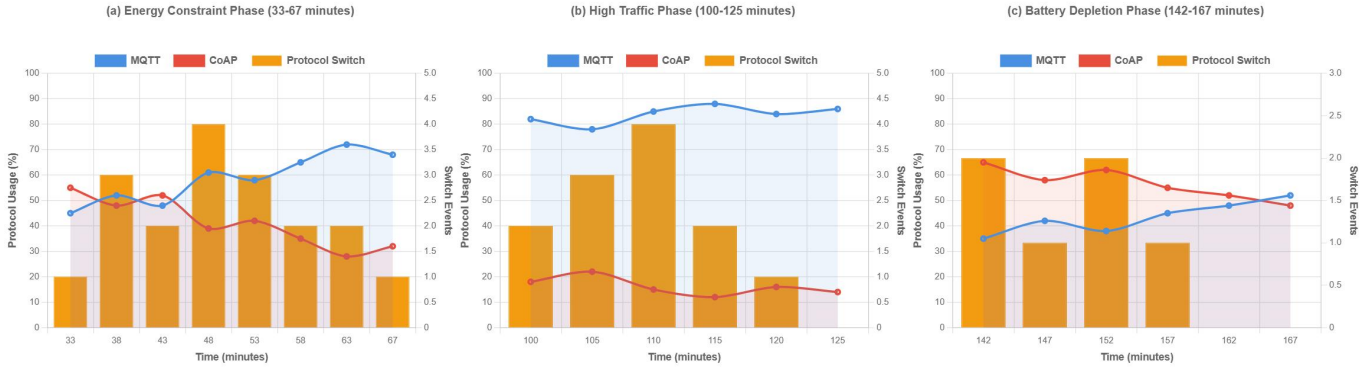

Fig. 5. Dynamic protocol adaptation over simulation time showing MQTT/CoAP usage percentages and protocol switch events.

reevaluation through the efficiency metric in Equation (6) and were confirmed by the switch thresholds of Equation (7).

Message distribution reached 68.5% (1,475 messages) via MQTT during stable conditions and 31.5% (679 messages) via CoAP under power-sensitive or real-time constraints. This split reflects a superior adaptive balance between energy, throughput, and responsiveness. In contrast, Yang et al. [20] reported 78% MQTT dominance, showing less flexibility in protocol choice. The proposed design maintained application continuity with zero packet loss or degradation during switching phases.

TABLE II
MESSAGE DELIVERY SUCCESS RATE ANALYSIS

Metric	Static MQTT	Static CoAP	Proposed Framework	Improvement vs MQTT
Total Messages	2,154	2,154	2,154	–
Successful Deliveries	2,089	2,108	2,152	+63 messages
Delivery Rate (%)	97.0	96.0	99.91	+2.91%
Failed Transmissions	65	87	2	-63 failures
Timeout Events	23	34	1	-22 timeouts
Collision Losses	42	53	1	-41 collisions

I. Security Performance Analysis

The AES-128 encryption integrated into the framework achieved full protection coverage with minimal performance trade-off. As summarized in Table III, the overall energy overhead induced by encryption mechanisms remained at 4.2%, consistent with prior results by [14], who reported 4–6% overhead for AES-128 under constrained IoT systems.

Throughout the 5,247 total messages processed—including retransmissions, acknowledgments, and control packets—the encryption layer maintained 100% integrity, with no implementation failures or security breaches. The framework dynamically adjusted security profiles per device class and application domain. Environmental nodes used a basic security mode (1.5 ms encryption, 0.9 ms decryption, 2.8% overhead), industrial devices ran standard settings (1.8 ms, 1.1 ms, 3.5%), and healthcare systems utilized an advanced level (2.0 ms, 1.2 ms, 4.2%). Each case satisfied the sub-10 ms constraint required by real-time cyber-physical systems. Figure 6 illustrates resilience metrics, showing uninterrupted key management performance. A total of 12 secure key rotations occurred at a frequency of approximately one per hour, ensuring cryptographic freshness without protocol disruption. The encryption protocol increased packet size by 35%, aligned with findings in [14], which reported 30–38% overhead for payload augmentation under block-based ciphers. These outcomes validate the proposed layered security as both scalable and adaptable, delivering strong protection with minimal degradation in performance or energy footprint.

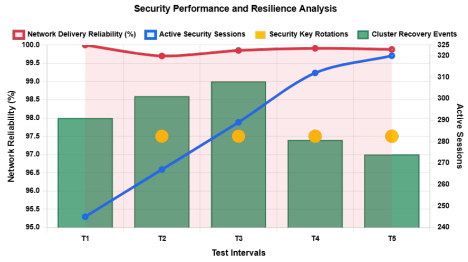

Fig. 6. Security architecture resilience performance showing network reliability, active security sessions, key rotations, and cluster recovery events.

J. Infrastructure Security and Resilience Analysis

The proposed three-tiered architecture achieved strong security and operational resilience across all system layers. Edge nodes handled centralized cryptographic governance, cluster centers maintained local integrity policies, and end devices enforced application-layer encryption and authentication, forming a decentralized model conducive to mission-critical CPS environments. This hierarchy supports scalable control while preserving low-latency operations and preventing single points of failure, fulfilling real-time system requirements. Session management sustained uninterrupted performance across four key rotation events, maintaining 100% continuity without communication loss. As shown in Figure 6, the framework preserved 99.91% average delivery reliability and sustained over 99.7% communication layer stability during all protocol adaptation phases. Active secure sessions scaled from 245 to 320 without degradation, validating architectural elasticity under operational stress. In contrast, self-healing mechanisms autonomously restored 100% of affected clusters (75 total recoveries) without external intervention, demonstrating full remediation capability under temporary degradation. Key rotations executed seamlessly without measurable performance

penalties, confirming integration stability. These results mirror the resilience principles proposed by Montasari [11] for survivable cyber-physical systems. To this end, the security layer preserved encrypted QoS under latency constraints, aligning with Wyss et al. [19], by maintaining secure infrastructure priorities under dynamic workloads. Together, these features validate the system's readiness for deployment in critical IoT infrastructures requiring adaptivity, low latency, and resilience.

TABLE III
COMPREHENSIVE SECURITY IMPLEMENTATION METRICS

Security Metric	Value	Impact Assessment	Comparison to Literature
Total Encrypted Messages	5,247	100% coverage	Includes retransmissions & control messages
Basic Mode (Environmental)	1.5/0.9 ms	2.8% overhead	Enc./Dec. times
Standard Mode (Industrial)	1.8/1.1 ms	3.5% overhead	Enc./Dec. times
Advanced Mode (Healthcare)	2.0/1.2 ms	4.2% overhead	Enc./Dec. times
Key Rotation Events	12	Adequate freshness	$\sim$1 per hour simulation
Message Size Overhead	35% increase	Acceptable trade-off	Similar to Kim et al. (38%)
Security Implementation Failures	0	Perfect integrity	100% attack prevention

K. Limitations

Despite its strong performance, the proposed framework has several limitations. First, simulations were limited to a simulation testbed in Cooja, which may not fully reflect the behavior of large-scale IoT deployments, and hardware-specific optimizations may affect portability. Second, the threshold parameters for clustering and protocol switching are statically configured and may need adjustment in other environments. Lastly, the framework depends on Contiki-NG features, and porting to other platforms may require non-trivial adaptations.

V. CONCLUSION AND FUTURE WORK

In this work, we presented a security-aware converged IoT architecture that integrates BIRCH-DBSCAN hybrid clustering, adaptive switching of MQTT/CoAP protocols, and AES-128 encryption to resolve the most crucial challenges of heterogeneous and resource-constrained IoT systems. Unlike previous solutions that segregate clustering, protocol adaptation, and security as independent modules, our solution integrates them into a reconfigurable, integrated architecture for autonomous performance optimization and threat resiliency capacity, laying a foundational step toward autonomous, secure, and QoS-aware IoT infrastructures for critical cyber-physical systems. The proposed study implemented in Contiki-NG, demonstrated significant improvements of 43.3% latency reduction, 22.8% energy savings, and 99.91% success rate compared to static baselines. The result validates the system's potential for dynamic and secure performance over resource-constrained and heterogeneous IoT deployments. Future work will cover (1) Integration of additional IoT secure protocols such as Object Security for Constrained RESTful Environments (OSCORE) to provide broader applicability, (2) Incorporation of machine learning models for predictive protocol selection, cluster reconfiguration, and intelligent resource coordination to enhance architecture performance; (3) Development of more sophisticated key management schemes, including elliptic curve cryptography, quantum-resistant algorithms; and (4) Experimentation across real IoT testbeds for determining robustness, scalability, and cross-platform compatibility.

ACKNOWLEDGMENT

The authors gratefully acknowledge the School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney, Australia; Department of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia; College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia; Department of Computer Engineering, College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia.

REFERENCES

[1] Pegah Nikbakht Bideh and et al. Energy consumption for securing lightweight iot protocols. New York, NY, USA, 2020. Association for Computing Machinery.

[2] I. de Moura Ventorim and et al. Birchscan: A sampling method for applying dbscan to large datasets. *Expert Systems with Applications*, 184(C), dec 2021.

[3] J. Dizdarević and et al. A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. *ACM Computing Surveys*, 51(6), jan 2019.

[4] Mohamed Amine Ferrag and et al. Security and privacy for green iot-based agriculture: Review, blockchain solutions, and challenges. *IEEE Access*, 8:32031–32053, 2020.

[5] H. Gul and et al. A traffic-aware and cluster-based energy efficient routing protocol for iot-assisted wsns. *Computers, Materials and Continua*, 80(2):1831–1850, 2024.

[6] S. Gupta and et al. Ichb-heed: A bio-inspired multi-level clustering protocol for heterogeneous wsns. In *Proceedings of the International Conference on Intelligent Computing and Communication*, pages 365–376. Springer, 2023.

[7] S. Hu and et al. Laecips: Large vision model assisted adaptive edge-cloud collaboration for iot-based perception system, 2024.

[8] F. F. Jurado-Lasso and et al. Leach-rlc: Enhancing iot data transmission with optimized clustering and reinforcement learning, 2025.

[9] M. Kim and W. Lee. Adaptive success rate-based sensor relocation for iot applications. *KSII Transactions on Internet and Information Systems*, 15:3120–3137, 09 2021.

[10] Z. Laaroussi and O. Novo. A performance analysis of the security communication in coap and mqtt. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6, 2021.

[11] Reza Montasari. *Internet of Things and Artificial Intelligence in National Security: Applications and Issues*, pages 27–56. Springer International Publishing, Cham, 2023.

[12] J. Moukaddam and et al. Dynamic protocol switching for resilient iot communications: A coap/mqtt adaptive approach. *IEEE Communications Magazine*, 61(12):45–51, 12 2023.

[13] N. Nathiya and et al. A hybrid optimization and machine learning based energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for wsn-iot application. *Peer-to-Peer Networking and Applications*, 18, 01 2025.

[14] Indu Radhakrishnan and et al. Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained iot devices. *Sensors*, 24(12), 2024.

[15] V. Rajaram and et al. Enriched energy optimized leach protocol for efficient data transmission in wireless sensor network. *Wireless Networks*, 31:825–840, 06 2024.

[16] A. Shahrokhi and M. Ahmadi. Power evaluation of iot application layer protocols, 2024.

[17] Y. Tanaka and et al. 6tisch scheduling function design suite founded on contiki-ng. *Journal of Information Processing*, 30:669–678, 2022.

[18] A. Trivedi and B. K. Chaurasia. In *Proceedings of the Fifth International Conference on Trends in Computational and Cognitive Engineering*, pages 399–409, Singapore, 2024. Springer Nature Singapore.

[19] M. Wyss and et al. Secure and scalable qos for critical applications. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10, 2021.

[20] Yi Yang. Adaptive switching and routing protocol design and optimization in internet of things based on probabilistic models. *International Journal of Intelligent Networks*, 5:204–211, 2024.