Article

Not peer-reviewed version

Improved Cell Allocation Strategies Using K-means Clustering in Congested 6TiSCH Environment

Fransiskus Xaverius Kevin Koesnadi and Sang-Hwa Chung *

Posted Date: 22 July 2024

doi: 10.20944/preprints202407.1627.v1

Keywords: 6TiSCH; cell allocation; K-means; node density; wireless sensor network

Article

Improved Cell Allocation Strategies Using K-means Clustering in Congested 6TiSCH Environment

Fransiskus Xaverius Kevin Koesnadi and Sang-Hwa Chung *

Department of Information Convergence Engineering, Pusan National University, Busan 46241, Korea; fxkevink@pusan.ac.kr

* Correspondence: shchung@pusan.ac.kr

Abstract: The 6TiSCH protocol (IEEE 802.15.4e) is crucial for the Industrial Internet of Things (IIoT), utilizing a time-slotted channel hopping (TSCH) mode based on node distribution. In our experiment, we explored node position clustering and adjusted the capacity of existing cells in congested areas to understand traffic patterns and optimize cell selection for TSCH resource allocation, particularly in slot frames. Efficient cell allocation prevents packet loss, manages congestion, and enhances data transmission reliability. Node clustering increases cell usage efficiency and reduces communication latency between nodes, with each cell allocation tailored to its traffic needs through the sink node from the clustering result. We evaluated the clustering method using the K-means algorithm through experiments using the 6TiSCH simulator. Additionally, we considered using the routing protocol for low power and lossy networks (RPL) as a comparison, such as OF0 and MRHOF, as the objective function to assess clustering results and their impact on throughput and packet delivery. Our method significantly improved average performance metrics: under the OF0 routing protocol, we achieved a 30.01% latency reduction, 15.95% faster joining time, 8% higher packet delivery ratio, and 13.82% throughput increase. Similarly, with the MRHOF routing protocol, we observed a 12.34% improvement in packet delivery ratio, 21.06% latency reduction, 12.68% faster joining time, and 25.97% higher throughput speed.

Keywords: 6TiSCH; Cell Allocation; K-means; Node Density; Wireless Sensor Network

1. Introduction

The Internet of Things (IoT) connects various entities by enabling communication between physical devices. Nodes, acting as wireless sensors in this network, share environmental data, allowing computational systems to interact with the real world. In recent years, wireless sensor networks (WSN) have garnered significant interest due to their applicability in various fields. The collected data analyzes current environmental conditions, influencing appropriate responses [1]. The goal of 6TiSCH is to thoroughly compare 6TiSCH and real-world environments using formal models or actual implementations. 6TiSCH is a standard protocol that combines IPv6 and IEEE802.15.4 time-slotted channel hopping (TSCH), making it a crucial component of the Industrial Internet of Things (IIoT) by offering industrial-grade performance with seamless internet integration [2]. Therefore, this work identifies and addresses various challenges to develop more effective routing strategies and security management, aiming for greater efficiency and simplicity [3].

The 6TiSCH architecture is vital for meeting the high demands of industrial IoT, ensuring reliable and coordinated communication. Effective cell allocation prevents packet loss in areas with high node traffic. In smart factory settings, field devices, sensors, and supervisory systems require high reliability and prompt information exchange [4]. A key challenge in managing a 6TiSCH network is efficiently allocating cells based on node positions, as this affects transmission paths. Cell allocation greatly influences the network's performance and power consumption [5]. This work uses K-means clustering to improve cell allocation in the 6TiSCH network. Designating centroid nodes as cluster heads and linking nodes to the nearest center reduces transmission hops, simplifies neighbor discovery, and enhances scalability. The approach manages intra-cluster subchannel allocation and inter-cluster interference through distributed learning. These clustering results inform slot frame strategies, optimizing cell use, decreasing overhead, and improving efficiency. The 6TiSCH network's topology shows role distributions, with parent nodes as cluster heads and sink nodes as data collection centers.

The goal is to enhance resource utilization and reduce packet loss in the 6TiSCH network. K-means clustering nodes based on nearby communication needs, optimizing packet routes, minimizing packet loss and latency, and ensuring reliable communication. This method also prevents slot collisions and maintains efficient cell placement by ensuring cells from the same communication node do not occupy the same slot frame. If slots are insufficient, additional slots are allocated in the next available frame. K-means clustering improves network performance by providing insights into node distribution, communication patterns, and resource allocation, enhancing scalability and efficiency for IIoT applications.

The contribution of this study can be summarized as follows:

- We improved cell allocation in the 6TiSCH network, optimizing resource utilization, reducing packet loss, and ensuring reliable communication;
- We utilized K-means clustering to group nodes based on nearby communication needs, which minimized packet loss and latency, improved delivery ratio and throughput, and prevented slot collisions by maintaining efficient cell placement;
- We enhanced network performance and scalability for IIoT applications by providing insights into node distribution, communication patterns, and resource allocation, enabling better management of network resources and avoiding slot clashes through dynamic slot allocation.

The rest of this paper is organized as follows. Section 2 introduces the basic background of the simulator and the utilized tools with its problem explanation of clustering technique, while Section 3 contains the comparison works of this work. Section 4 presents the mechanism of density clustering for node-centric cell allocation in the 6TiSCH Network. Section 5 evaluates the comparison performance, which consists of the packet delivery ratio, the latency performance, time allocation for every node to join the network, and the throughput as the data rate measurement from the proposed method. Lastly, section 6 offers the summarization of this paper.

2. Background and Problem

2.1. TSCH

The IEEE 802.15.4e TSCH (Time-Slotted Channel Hopping) is an amendment to the IEEE 802.15.4 standard, outlining the physical (PHY) and medium access control (MAC) layers for low-rate wireless personal area networks (WPANs). Designed for industrial and automation applications, TSCH offers synchronized time-based communication, making it dependable and energy-efficient. It is ideal for applications requiring precise timing and coordination, such as industrial control systems or environmental sensor networks [6]. TSCH specifies functionality, but its implementation depends on network architecture and application needs. Channels are managed by dividing time into fixed timeslots and periodically changing frequency channels, with each node assigned specific times for data transmission or reception, synchronized with neighboring nodes, minimizing channel access conflicts [7].

TSCH is a MAC mode in the IEEE 802.15.4e standard, allocating timeslots for data transmission or reception. These allocations can be static or dynamic based on network requirements. Timeslots are organized into slot frames, which define cycles of timeslots that repeat according to the slot frame size [8]. Nodes communicate within these timeslots, synchronized with a shared channel hopping schedule that periodically changes the frequency channels used, reducing interference. This ensures precise communication and time synchronization between nodes, with the slot frame length adjustable based on network and application needs. Utilizing about 16 available channels, TSCH operation allows high coordination within the network and reduces channel access conflicts [9].

2.2. 6TiSCH Minimal Scheduling Function

6TiSCH (IPv6 over IEEE 802.15.4e TSCH mode) is an IETF standard for facilitating IPv6 communications over IEEE 802.15.4e TSCH mode [3]. It defines control plane protocols to align link-layer resources with application needs and integrates CoAP with link-layer security [10]. The 6TiSCH architecture, combining IPv6 with IEEE 802.15.4e TSCH MAC, organizes time into slots within slot frames, where each slot is a distinct time interval for node functions [11,12]. Cells, representing communication capability, are controlled by the 6TiSCH Operational Sublayer Protocol (6P), which manages cell allocations and time slot usage, ensuring reliable communication and resource efficiency in IIoT environments [12,13].

The Minimal Scheduling Function (MSF) in 6TiSCH networks adjusts cell allocation based on utilization metrics to meet dynamic bandwidth requirements [14]. Using TSCH at the MAC layer, MSF operates with minimal cells for network initialization, autonomous cells for default communication, and negotiated cells for managing communication load [15]. MSF coordinates time slot and cell distribution, synchronizes node transmission schedules, and assigns specific roles to nodes [16]. This protocol ensures efficient bandwidth use, low latency, and high dependability, suitable for real-time control and monitoring systems [12].

2.3. Routing Protocol for Low-Power and Lossy Networks (RPL)

The Routing Protocol for Low-Power and Lossy Networks (RPL) is specifically designed for networks with limited resources, such as WSN. RPL aims to provide an efficient and reliable routing mechanism within environments constrained by energy, bandwidth, and processing capabilities. It addresses challenges in wireless sensor networks, including node mobility, dynamic topologies, and communication in unpredictable network conditions. One of the key features of RPL is its support for hierarchical network structures, which facilitates efficient route establishment and maintenance in extensive networks. Using the Destination-Oriented Directed Acyclic Graph (DODAG), RPL enables nodes to create optimal communication paths toward specific destinations, considering network topology and data delivery requirements [17]. Additionally, RPL supports efficient network maintenance by dynamically reorganizing paths in response to changes in network topology or other environmental factors. This dynamic reorganization makes RPL a frequently used routing protocol in WSN and IoT applications, ensuring efficient and reliable data delivery while managing control overhead and limited memory constraints [18].

2.3.1. Objective Function Zero (OF0)

The Objective Function Zero (OF0) in RPL provides guidelines for nodes to select and optimize routes within an RPL instance using available information objects. Instead of specific instructions, OF0 offers a general concept to calculate a node's rank by adding a normalized scalar, to the rank of its preferred parent R_p. OF0 encodes rank in units of 256, allowing for hop ranges from 28 (worst) to 255 (best). In RPL, all parents are feasible successors for upward traffic, and nodes can consider parents in subsequent DODAG versions as potential successors. Multiply the step of rank S_p by the rank factor R_f and adjust it with a stretch factor S_r, constrained by the configured rank stretch, to compute the current node's rank R_N. This rank is added to the preferred parent's rank to find the node's overall rank. By calculating the rank, OF0 enables a node to participate in a DODAG version that provides satisfactory connectivity, where the minimal hop rank increase defines the lowest rank increase to any potential parent. OF0 does not guarantee optimization according to any specific metric, and connectivity validation is specific to implementation and link type, falling outside OF0's scope [19].

2.3.2. Minimum Rank with Hysteresis Objective Function (MRHOF)

Another RPL protocol, Minimum Rank with Hysteresis Objective Function (MRHOF), looks for the best path between source and destination nodes in resource-constrained wireless networks with changeable conditions. MRHOF aims to minimize routing overhead while balancing path quality and network stability. MRHOF evaluates each node's rank value to find the shortest path using the minimum rank algorithm. MRHOF uses hysteresis to prevent frequent changes and preserve path stability in the face of varying network conditions. Additive metrics are compatible with this protocol because they use metrics to minimize path costs [20].

2.4. Unsupervised Learning and K-Means Clustering

Unsupervised learning, a branch of machine learning, focuses on uncovering patterns within unlabeled data. Unlike supervised learning, which relies on labeled datasets, unsupervised learning aims to discover hidden structures. Its primary goals include clustering similar data, identifying correlations, and reducing data dimensionality without predefined labels. The clustering process depends heavily on similarity metrics, with the Euclidean distance—the sum of squared differences between two patterns—being the most commonly used [21]. Clustering algorithms may also employ similarity measures like the correlation coefficient. Cluster analysis identifies high-density groups where observations are more similar within clusters than between. The number of clusters, often user-defined, and the chosen similarity metric significantly impact the results [22]. The K-means algorithm is a popular unsupervised learning technique for clustering data points into predefined clusters [23]. It partitions a dataset of N observations into k clusters by iteratively assigning each observation to the nearest centroid, calculated as the mean of all data points in a cluster. The algorithm begins with random centroid initialization and iterates through assignment and update steps: observations are allocated to the nearest centroid based on Euclidean distance, minimizing the within-cluster sum of squares, and centroids are recalculated as the mean of all observations in each cluster. This repeats until centroids stabilize or a specified number of iterations [24]. As exemplified by K-means, partial clustering divides a dataset into k disjoint clusters, refining a local approximation to a global objective function iteratively. K-means aims to minimize the sum-of-squared error (SSE) or within-cluster sum of squares (WCSS) between data points and cluster centroids $\mu_1, ..., \mu_k$ using Euclidean distance. Challenges include non-spherical or differently sized clusters and sensitivity to outliers, potentially resulting in skewed clusters. Empty clusters may form, often resolved by reassigning outlying points. Persistent issues might suggest an inappropriate k value. The Elbow Method helps determine the optimal number of clusters by plotting WCSS against various k values to find the 'elbow' point. Silhouette analysis evaluates cluster compactness and separation, providing insights into the chosen k value.

2.4.1. Silhouette Score and Elbow Method

The Silhouette Score measures how well an object fits within its own cluster compared to how far it is from other clusters. This score ranges from -1 to 1, where positive values indicate that the objects are more similar to their cluster members than those in other clusters. In contrast, negative values suggest the opposite and measure the separation from the closest cluster to the i data point. Like the elbow method, the silhouette score helps determine the optimal number of clusters by identifying those with the highest scores [25].

The cluster labeled as B, which has the minimum distance to object i (when $d(i, B) = b(i)$), is considered the neighbor of object i. This cluster is the alternative choice for object i if it cannot be assigned to cluster A. Understanding the neighbor of each object provides insights into the closest alternative cluster when cluster A is not an option. It is crucial to note that determining $b(i)$ depends on the presence of other clusters besides A. Therefore, it is assumed throughout this discussion that the number of clusters k is should be more than one. The silhouette score $s(i)$ is computed by appropriately combining $a(i)$ and $b(i)$ [25].

In K-means clustering, the elbow method determines the optimal k value or the number of clusters. This method involves calculating and plotting the distortion, also known as the inertia value, for different values of k. Distortion is the total distance between any point in a cluster and its cluster center. As the value of k increases, the points tend to be closer to their cluster centers, causing the inertia value to decrease. Eventually, this decrease slows down, forming an elbow-like curve on the plot. The k

value at this elbow is the optimal k value because adding more clusters beyond this point does not significantly improve the explained variance in the data [26].

3. Related Work

This section provides an overview of studies on the clustering technique among wireless sensor networks. This method employs group nodes for administration and distributed job execution, including managing resources such as the congestion level of neighboring nodes. While clustering approaches primarily benefit reduced energy usage, they also achieve load balancing and energy conservation, addressing network heterogeneity and mobility.

One application of cluster formation is demonstrated by [27] in WSN, using particle swarm optimization and the Tabu algorithm approaches. These methods utilize cluster head role rotation based on energy levels, multi-hop routing algorithms, and fault tolerance mechanisms to enhance energy efficiency, network endurance, and protocol performance. However, parameter settings still limit cluster head selection, leading to high energy consumption. The space partition in K-means using Optimal K-means [28] addresses the area partition by forming space partitions in wireless sensor networks, yet the node issuance algorithm remains unclear. Additionally, modified grid-based wireless sensor networks using K-means [29] have been tested to increase energy efficiency and extend the lifespan of large-scale wireless sensor networks. By dividing the network area into a grid and applying K-means in each grid cell, overall energy usage is reduced, and network longevity is improved. However, it may limit the implementation of heterogeneous sensor networks based on a static parent node.

Hierarchical protocols like Low-Energy Adaptive Clustering Hierarchy (LEACH) reduce the amount of data transmitted across the network by dividing it into clusters with cluster heads responsible for data aggregation and relaying. This conserves energy and improves scalability. Modifying LEACH, as proposed by [30], combines the midpoint algorithm with the K-means approach to enhance network lifespan in wireless sensor networks, aiming to create balanced clusters and reduce cluster head load. Further optimization of this method could improve overall network efficiency and prolong network lifespan, addressing minor declines in the packet delivery ratio observed in initial tests. As mentioned earlier, LEACH can be adapted to anticipate reduced data transmission, as suggested by [31], which developed an algorithm selecting a primary and secondary cluster head to maintain network performance and efficiency, ensuring data flow continuity even in node failure. Another method by [32] focuses on improving average energy consumption per node each round and the number of surviving nodes per round, though routing algorithms still need to address high latency issues.

To minimize frequent re-clustering and associated energy overhead, [33] introduced a multi-hop routing algorithm to optimize energy use across the network, incorporating a fault-tolerant mechanism to address CH and relay node failures. However, this experiment's routing process emphasized close transmission distances without considering the need for nodes to be utilized, aiming to minimize energy consumption, even if the receiver node remains inactive without any expected packet reception.

Our research focuses on cell allocation and its impact on wireless sensor network implementation, differing from previous works that typically focus on node localization. We examine how node grouping performance affects each cell allocation, particularly in TSCH, and its indirect relation to the cell allocation system in WSN, considering node location, energy efficiency, and network performance. Unlike earlier studies, we investigate how group node positioning influences cell operation in TSCH, focusing on scalability, throughput delay, and reliable communication. Furthermore, to enhance network efficiency, we propose a dynamic allocation method that adjusts based on node mobility and network responsiveness during data transmission.

4. Nodes Clustering

4.1. Cluster Initialization and Parameter Enhancement

Initially, cluster centers are determined based on their coordinates in the random mote connectivity matrix. Each cluster C_i central node randomly initializes the centers within the data space, with boundaries defined by k values. A random integer generator within a specified range determines each cluster center, represented by the coordinates (x_N, y_N). Algorithm 1 in Figure 1 details the initialization and update procedures for the K-means cluster center. The process begins by randomly selecting the first cluster center from the given range for each coordinate (x, y). The method then iteratively updates the cluster center for each cluster using the average coordinates of the nodes within that cluster. This iterative process continues until convergence, ensuring the cluster center stabilizes and does not change significantly with subsequent iterations, achieving a *new_center* of the coordinate.

```
Algorithm 1 K-means Cluster Center Initialization and Update
Input: $nodes$ as number of nodes
Output: $cluster\_center$

Initialize: $cluster\_center \leftarrow []$, $new\_center \leftarrow []$

Class RandomConnectivity
function UpdateClusterCenter
    for $center, node$ in $clusters.items()$ do
        if $node$ is not empty then
            $x_{total} \leftarrow$ add values of $x$ coordinate of nodes
            $y_{total} \leftarrow$ add values of $y$ coordinate of nodes
            $x_{average} \leftarrow x_{total} / num\_nodes$
            $y_{average} \leftarrow y_{total} / num\_nodes$
            $new\_center$.append($x$: $x_{average}$, $y$: $y_{average}$)
        else
            $new\_center$.append({'$x$': $center[0]$, '$y$': $center[1]$})
        end if
  end for
  return $new\_center$ as new node centroid based on its coordinate
```

Figure 1. Pseudocode of K-means Cluster Center Initialization and Update.

Use the average coordinates of the nodes within each cluster C_i to iteratively update the coordinates of cluster centers $(x'c_i, y'c_i)$ through the K-means parameter optimization method. To assign each node to its nearest cluster, the algorithm calculates the Euclidean distance, denoted as $d(N, C_i)$ in equation (5), between each node N and the updated cluster centers $(x'c_i, y'c_i)$, starting from the previous cluster centers (x_{C_i}, y_{C_i}) to form the initial proximity-based clusters. At this stage, represent node N coordinates, which contribute to the cluster centers (x_{C_i}, y_{C_i}), by x_N and y_N.

$$d(N, C_i) = \sqrt{\left(x_N - x_{C_i}\right)^2 + \left(y_N - y_{C_i}\right)^2} \tag{1}$$

Algorithm 2.1 Node Granting and Optimal K Selection – Coordinate Clustering

```
Input: $nodes$ as the number of nodes
       $cluster\_center$ as node connectivity cluster

Initialize: $distances$ ← []

function $CoordinateClustering$
for $node$ in each $nodes$ do
    for each $center$ in $cluster\_center$ do
        $distance$ ⟵ $Euclidean\_distance(node, center)$
        $distances$.append($distance$)
    end for
    $nearest\_cluster\_index$ ⟵ index of $minimum\_value$ in $distances$
    assign $node$ to the $near\_cluster$
end for
```

Figure 2. Pseudocode of Node Granting and Optimal K Selection – Coordinate Clustering.

Algorithm 2.2 Node Granting and Optimal K Selection – Clustering Algorithm

```
Input: $r = \{r_1, r_2, \ldots, r_i\}$ as dataset range of nodes in a grid area
       $K_{range} = [k_{min}, k_{max}]$ as range of possible $k$ values
       $max\_iterations$ as maximum number of iterations
       $new\_center$ as new node centroid based on its coordinate
Output: Clusters $C = \{C_1, C_2, \ldots, C_k\}$ with $optimal\_k$

Initialize: $silhouette\_score$ ← [], $WCSS$ ← [], $distance$ ← []
Initialize: $best\_silhouette\_score$ ← -1, $best\_k$ ← 0

function ClusteringAlgorithm
for $k$ in $K_{range}$ do ← Start cluster centers at random inside $r$'s boundaries.
    for $i$ from 1 to $max\_iterations$ do
        for each data point $r_i$ in $r$ do
            assign $r_i$ to $near\_cluster$ ←Assign the closest cluster points in $r$
        end for
        for each $new\_center$ in $near\_cluster$ do
            $UpdateClusterCenter(cluster\_center)$ ← recalculate center
        end for
        if $cluster\_center$ not change then
            break
        end if
    end for
    $WCSS[k]$ ←For assignments, compute the within-cluster sum of squares
    if k > 1 then
        $silhouette\_scores[k]$ ← Compute the assignment's silhouette score
        if $silhouette\_scores[k] > best\_silhouette\_score$ then
            $best\_silhouette\_score = silhouette\_scores[k]$
            $best\_k = k$
        end if
    end if
end for
```

Figure 3. Pseudocode of Node Granting and Optimal K Selection – Clustering Algorithm.

Equations (2) and (3) ensure that the cluster centers $(x'c_i, y'c_i)$ consistently improve to accurately reflect the spatial distribution of nodes within each cluster C_i after the initial assignment, as represented by $|N_{C_i}|$. Additionally, our method employs the Elbow Method and Silhouette Score (5) and (6) to determine the optimal number of clusters k. The Elbow Method identifies the ideal number of node clusters by detecting the point where the reduction rate in the within-cluster sum of squares (WCSS) significantly changes. Similarly, the Silhouette Score measures the compactness and separation of clusters, with higher scores indicating more distinct clusters. Algorithm 2.1, illustrated in Figure 2, outlines the procedures involved in the optimization process, including node assignment and optimal k selection.

The algorithm begins by randomly initializing cluster centers and assigning nodes to the nearest cluster based on Euclidean distance. For each k value in the specified range, the algorithm computes the WCSS (4) using x and y coordinate points and (μ_i) centroids as a measure to evaluate clustering quality and iteratively updates the cluster centers. Consequently, algorithm 2.2 in Figure 3 determines the optimal number of clusters for the final K-means clustering phase.

$$x'c_i = \frac{1}{|N_{C_i}|} \sum_{N \in N_{C_i}} x_N \tag{2}$$

$$y'c_i = \frac{1}{|N_{C_i}|} \sum_{N \in N_{C_i}} y_N \tag{3}$$

5. Scheduling of TSCH

The MSF organizes these slot frames into TX and RX cells for each node, considering communication requirements and network grouping. Parent nodes in the MSF coordinate slot frame allocations, while child nodes receive these allocations to enable communication. Typically, nodes send packets to a parent node, but if the parent node is unavailable, a child node can act as a hop node to forward packets to the sink node. Slot and channel availability are considered during cell allocation; hop nodes select the slot offset, channel, and TX and RX cells for communication between child nodes. Simultaneously, the root or sink node applies the same algorithm to determine the optimal cell configuration. Custom cell allocation involves determining the specific cell configuration allocated exclusively to each child node. The hop node assigns a slot offset to each child node based on the number of allocated cells (N_{cells}), ensuring each cell has a distinct offset. The root or sink node employs the same algorithm to determine exclusive cell allocations for each child node through the predetermined cluster head.

In the slot frame, a component of the data transmission process, other nodes, such as Node A to B, may attempt to occupy the same slot allocation. This occurs due to the inherent allocation process in the slot frame, which can lead to potential conflicts and interference, thereby reducing the packet delivery ratio (PDR). Additionally, delayed cell allocation for every 0.1 seconds of 1 delayed packet could reduce latency. This application can be seen in Figure 4.

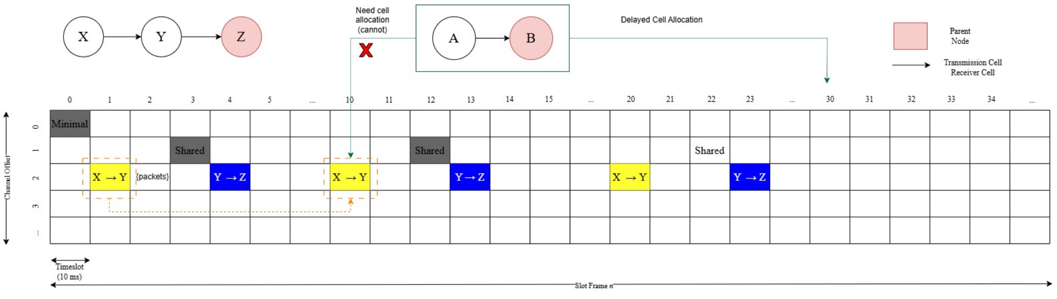


Figure 4. TSCH scheduling based on period traffic.

6. Cell Allocation Implication of Nodes

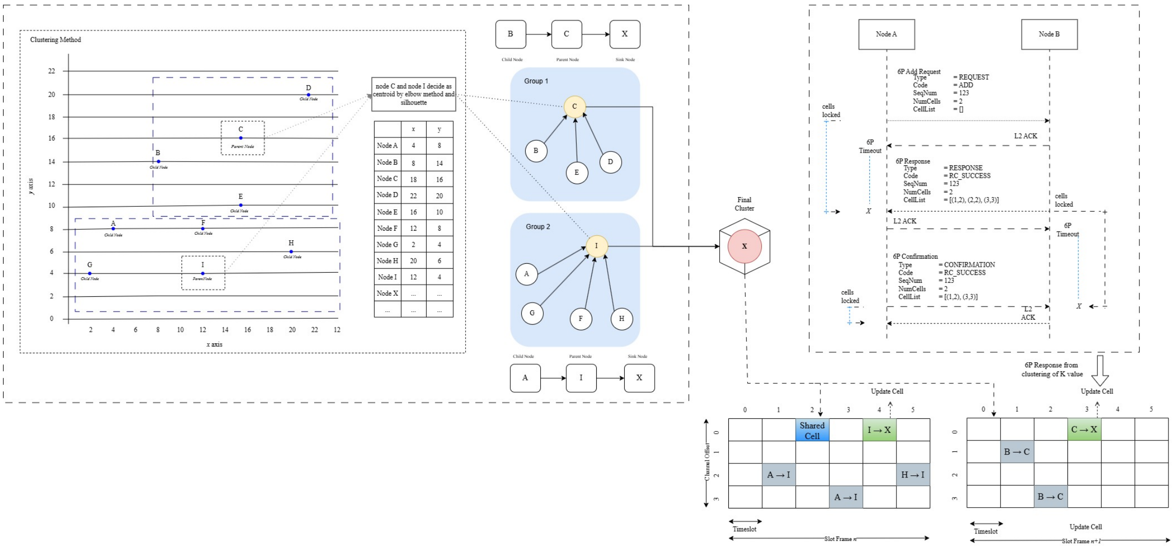
As illustrated in Algorithm 3 in Figure 5, the improved cell allocation optimizes communication cell allocation in the TSCH network. The algorithm begins with inputs like S, representing the number of slot frames used by cells in a cluster, and *current_cluster*, indicating the network's current clustering status. It compares a node's current cluster association with its neighbors using *GetCluster*. It evaluates each slot frame in S to collect cluster details of neighboring nodes by calculating the number of slot frames utilized during a given period for each cell involved. The main execution phase continuously processes cell assignments until termination conditions are met, with the node acting as a child node performing the scheduling function through 6p transactions.

```
Algorithm 3 Enhanced TSCH Cell Allocation based on Clustering
Input: $S = \{S_1, S_2, \ldots, S_i\}$ as the quantity of frame slots that each cluster cell uses
       $current\_cluster$ as current clustering network state
Output: Cell operations decisions based on clustering

▷Examine and contrast the current node's cluster with that of its neighboring node
Procedure: $GetCluster(neighbor, current\_cluster)$
for each slot frame $i$ in $S$ do
    $clustering[neighbor] \leftarrow$ Get the neighbor node's cluster information
end for

▷Assigning communication cells to network nodes
repeat
    $NeighborCluster = GetCluster(S.neighbor, current\_cluster)$
    $SelfCluster = GetCluster(Self, current\_cluster)$
    if $SelfCluster = NeighborCluster$ then
        ▷ If the requesting and the neighbor nodes are in the same cluster
        $ExecuteIntraClusterStrategy(S)$
        if $S > Current\_Clustering$
            add active $TXcell$
        else
            relocate active $TXcell$
        end if
    else
        ▷ If the requesting and the neighbor nodes are in a different cluster
        $ExecuteInterClusterStrategy(S)$
        if $S < current\_cluster$
            delete active $TXcell$
        else
            relocate active $TXcell$
        end if
    end if
until END
```

Figure 5. Pseudocode of Enhanced TSCH Cell Allocation based on Clustering.

Necessary information, such as time slots and a list of available channels, is then submitted to each clustered cell list. When the current node and its neighbors are part of the same cluster, as determined by *GetCluster*, an intra-cluster strategy is executed, involving nodes within the same cluster, interpreted as a group of nodes with coordinates close to each other. Forming a cluster aims to enhance communication efficiency, such as time slot utilization. During this phase, active TX cells are added if the number of used slot frames exceeds the current cluster capacity. In contrast, existing TX cells are dynamically reallocated if demand decreases, optimizing resource usage by moving cells to areas with higher user density or increasing throughput.

If nodes belong to different clusters, an inter-cluster strategy is applied using gateway or router nodes to facilitate communication between clusters. This allows the network to grow and adapt without disrupting operations. To prevent resource wastage, active TX cells can be removed if fewer slot frames are available, or relocated to suit inter-cluster needs. Effective node clustering improves packet transmission efficiency by assigning specific cells based on a predetermined schedule. Removing unnecessary cells consolidates resources, reduces packet queuing, and enables efficient communication between parent and child nodes. 6P transaction processing minimizes signaling overhead, and nodes intelligently control cell usage, reducing 6P requests and acknowledgments while maintaining reliable connections. Details of this proposed method are illustrated in Figure 6.



	x	y
Node A	4	8
Node B	8	14
Node C	18	16
Node D	22	20
Node E	16	10
Node F	12	8
Node G	2	4
Node H	20	6
Node I	12	4
Node X	...	...
...	...	...

Figure 6. Schema of the proposed clustering for the cell allocation.

7. Performance Evaluation

This section will explain the simulators and parameters used in the experiments and the methods used to compare the routing protocols. It will also explain how each RPL affects packet data transmission and how the node coordinate clustering results affect the performance metrics tested, including packet delivery ratio, throughput, latency, and joining time of every node.

7.1. Experimental Setup

To evaluate the effectiveness of our proposed K-means clustering approach, we compared it with other clustering methods, including GRID-based K-means (GBK) [29], ECRP clustering [33], and the default MSF [14]. All procedures were implemented on the same network layer in TSCH using the standard 6TiSCH simulator, with each node having a battery level of 2821.5 mAh. The evaluation involved randomly placing nodes in an area for four scenarios: 100, 150, 200, and 300 nodes, with each node positioned at specific coordinates. The number of nodes was chosen based on related work to ensure consistency. The optimal value for each parameter k was determined using the silhouette score and elbow method. Within the controlled environment of the 6TiSCH simulator, we carefully assessed the performance using the RPL. The assessment process included uniformly adjusting the packet transmission intervals up to 0.1 percent. Each network node produced 120 packets with 80 bytes of data, leading to a standardized transmission packet size of 1016 bits. This experiment was conducted using two different routing protocols. The first protocol is OF0 [19], which prioritizes paths with the highest packet delivery ratio among potential next-hop neighbors on the route to the root or parent node. The second routing protocol used is MRHOF [20], which integrates metrics like rank, which indicates link quality, and incorporates hysteresis to prevent excessive path changes, thereby minimizing latency and overhead.

$$\text{Throughput } (kbps) = \frac{\text{Received packet } \times \text{ Packet size}}{\text{Simulation time allocation}} \tag{4}$$

This work illustrates how a node-based clustering strategy can improve the 6TiSCH network's scalability and performance. Network performance was measured using reliability indicators such as PDR, joining time, end-to-end latency, and throughput. PDR measures the percentage of successfully delivered packets among all transmitted packets to determine how reliable the network is at providing data. Measure the time a packet travels from its source to its destination by end-to-end latency to indicate how responsive the network is. The joining time indicates how quickly a new node integrates into the network. Throughput, as shown by the formula in (4), measures the data rate successfully sent from one node to another within a certain period or by the number of packets from each source node to the destination node, considering the packet size used, and is measured in kilobits per second ($kbps$). Table 1 displays the parameter simulations along with our suggested approach.

Table 1. Simulation Parameters.

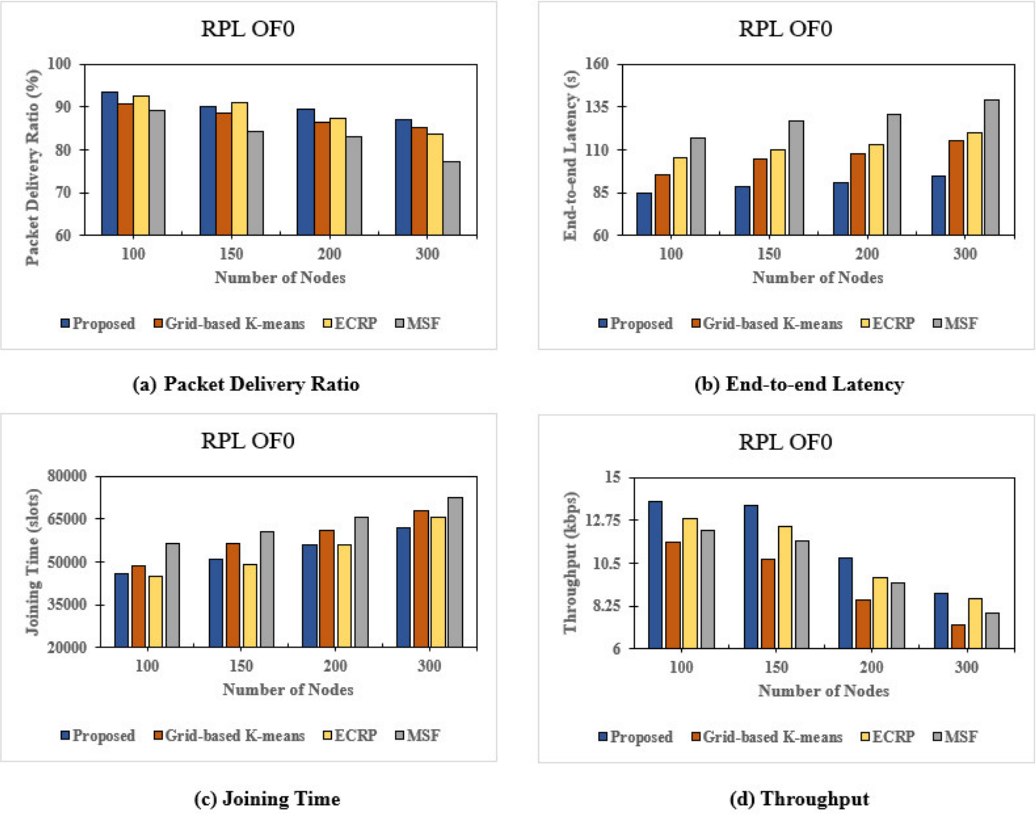
Parameter	Value
Simulation area (grid size)	2 km × 2 km
Simulation platform	6TiSCH
Battery level	2821.5 mAh
Number of nodes	100, 150, 200, 300
RPL extensions	Unicast
RPL DAO interval	60 seconds
RPL objective function	OF0, MRHOF
Traffic period	Periodic
Node distribution	Random
TSCH number of channels	16
TSCH TX queue size	12 frames
TSCH timeslot length	10 miliseconds
K maximum value	10
K-means maximum iterations	120
K-means features	3
Slot frame per run	4800
Packet size	1016
Silhouette score	-1
Silhouette cluster	3

8. Results

8.1. Objective Function Zero Routing Protocol

We initially conducted a performance test using the Objective Function Zero (OF0) routing protocol. This test compared our proposed clustering method with other methods and the MSF, as shown in Figure 7. Our method maintained a consistent delivery ratio across varying node counts. However, ECRP clustering slightly surpassed it at 150 nodes due to its cluster head selection based on node energy. The minimum hop count process in OF0 also affects inter-cluster communication. Despite this, our method's iterative clustering and scheduling processes effectively predicted packet arrival, outperforming the grid-based clustering method. Overall, our clustering process yielded positive results. On Figure 7 also demonstrates that the end-to-end latency with the OF0 routing protocol remained under 100 seconds even with 300 nodes, attributed to the effective distribution of slot frames and timeslots. The end-to-end latency for clustering on cell allocation also showed critical improvements compared to grid-based K-means.

Figure 7 highlights the consistent joining time across different node counts, with K-means clustering reducing the time needed for new nodes to receive advertisements from the sink node. Energy efficiency in clustering remains high, largely due to residual energy considerations for each node group. This allows our method to optimize how the sink node manages packet delivery for reserved nodes. Figure 7 shows favorable throughput results, with each packet's transmission speed enhanced by OF0's low-overhead path selection. Although energy efficiency clustering slightly narrows the throughput margin at 300 nodes, maintaining 8.89 kbps, the overall throughput remains high due to efficient packet transmission. In summary, our proposed method excels in packet delivery ratio and throughput, showing an 8% increase in delivery ratio, a 30.01% reduction in latency, a 15.95% decrease in joining time, and a 13.82% increase in throughput compared to the MSF under the OF0 routing protocol.



(a) Packet Delivery Ratio

(b) End-to-end Latency

(c) Joining Time

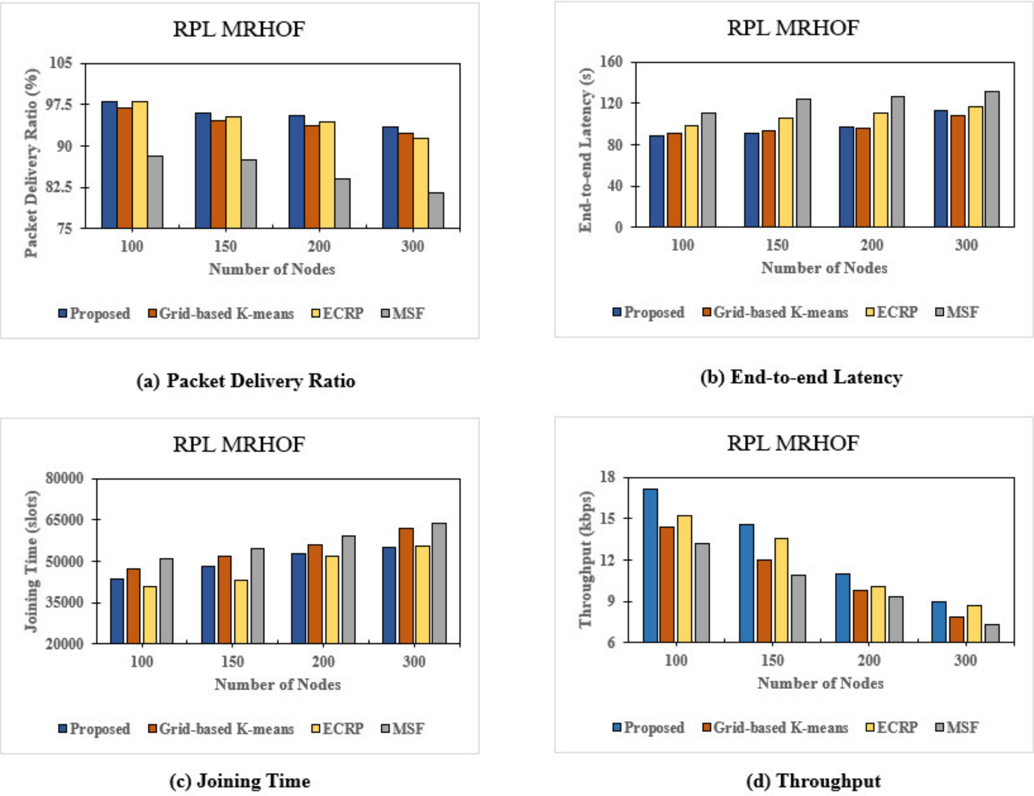
(d) Throughput

Figure 7. Graph result of the Objective Function Zero (OF0) routing protocol. (**a**) Graph of Packet Delivery Ratio (PDR). (**b**) Graph of the End-to-end Latency. (**b**) Graph of the Joining Time. (**b**) Graph of the Throughput.

8.2. Minimum Rank with Hysteresis Routing Protocol

Our next experiment utilized the Minimum Rank with Hysteresis Objective Function (MRHOF) routing protocol, using the same parameters as OF0 and comparing the results with the default MSF. Our proposed clustering method directly allocates cells to each slot frame, facilitating packet retransmission to the destination node, especially in high-density clusters. This method achieved slightly better PDR percentage results than other clustering techniques and supports multi-path routing, unlike OF0. The highest PDR, observed with 100 nodes, reached 98.15% and decreased to 93.35% with 300 nodes, reflecting increased traffic load on network resources. End-to-end latency performance metrics, remained stable despite a slight increase compared to grid-based K-means, reaching 112.91 seconds for 300 nodes due to fewer retransmissions.

Joining time was consistent with other K-means clustering methods, achieving 55219 slots compared to 55394 slots for ECRP clustering. Increased data traffic or competition for cluster resources likely caused congestion between cluster heads, delaying the transmission queue. Figure 8 also shows throughput values for data transmission speed, differing from RPL OF0. Throughput, closely related to PDR, decreased from 17.12 kbps to 9 kbps as nodes increased from 100 to 300, likely due to higher message transmission during the 6p transaction in grid-based K-means. Figure 8 highlights an anomaly where fewer nodes encountered difficulties due to clustering on low nodes in 6top protocol cell allocation, resulting in performance similar to MSF without clustering. Compared to MSF, our method showed a 12.34% increase in the average PDR, a 21.06% reduction in latency, a 12.68% decrease in joining time, and a 25.97% increase in throughput speed.



(a) Packet Delivery Ratio

(b) End-to-end Latency

(c) Joining Time

(d) Throughput

Figure 8. Graph result of the Minimum Rank with Hysteresis (MRHOF) routing protocol. (**a**) Graph of Packet Delivery Ratio (PDR). (**b**) Graph of the End-to-end Latency. (**b**) Graph of the Joining Time. (**b**) Graph of the Throughput.

9. Conclusion

This paper demonstrates the application of node clustering results using the K-means method for cell allocation in the 6TiSCH network, utilizing node position data and segmenting based on OF0 and MRHOF, two RPL factors influencing routing paths and network organization. Our method involves positioning nodes, clustering node coordinates, determining cluster centers, designating centroids as cluster heads, and associating other points with the nearest centers without repeated iterations. The goal is to minimize packet hops in packet routing and reduce neighbor discovery, improving performance efficiency and network traffic stability in the 6TiSCH network. RX and TX cells are essential for creating an efficient slot schedule in a slot frame. The system allocates each node a time slot for receiving RX cells and transmitting TX cells. Performance can vary while adhering to the standard scheduling function in the 6TiSCH network. Utilizing node position patterns allows K-means clustering to optimize resource allocation, managing resources based on cluster density. Clustering results determine the number of slots arranged in a time frame, adjusting cell operations through 6p transactions using K-means. The algorithm forms clusters of nodes based on proximity, optimizing specific cell allocation. In high-density scenarios, it adds dedicated cells to accommodate increased demand and removes redundant cells in low-density situations. The strategy prioritizes allocating unicast cells immediately after the receiving cell to minimize latency. Performance results regarding packet delivery ratio, packet latency, node joining time, throughput, and data transmission success are promising. Evaluations across different node scenarios and comparisons with other clustering methods show significant improvements in average performance metrics: under the objective function zero (OF0) routing protocol, we achieve a 30.01% decrease in latency, a 15.95% reduction in joining time, an 8% increase in packet delivery ratio, and a 13.82% increase in throughput. Using the minimum rank with hysteresis objective function (MRHOF) routing protocol, the average packet delivery ratio increased by 12.34%, latency decreased by 21.06%, joining time decreased by 12.68%, and throughput speed increased by 25.97%.

Author Contributions: Conceptualization, F.X.K.K.; Methodology, F.X.K.K.; Software, F.X.K.K.; Implementation, F.X.K.K.; Validation, F.X.K.K. and S.-H.C.; Formal analysis, F.X.K.K. and S.-H.C.; Investigation, F.X.K.K. and S.-H.C.; Resources, F.X.K.K. and S.-H.C.; Data curation, F.X.K.K.; Writing—original draft preparation, F.X.K.K.; Writing—review and editing, F.X.K.K. and S.-H.C.; Visualization, F.X.K.K.; Supervision, S.-H.C.; Project administration, S.-H.C.; Funding acquisition, S.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2023-00260098) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant RS-2023-00256816).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Kalita, A.; Khatua, M. 6TiSCH - IPv6 Enabled Open Stack IoT Network Formation: A Review. *ACM Transactions on Internet of Things* **2022**, *3*, 3. https://doi.org/10.1145/3536166.
2. Municio, E.; et al. Simulating 6TiSCH Networks. *Transactions on Emerging Telecommunications Technologies* **2018**, *30*, 3. https://doi.org/10.1002/ett.3494.
3. Vilajosana, X.; Watteyne, T.; Chang, T.; Vucinic, M.; Duquennoy, S.; Thubert, P. IETF 6TiSCH: A Tutorial. *IEEE Communications Surveys and Tutorials* **2020**, *22*, 1, 595–615. https://doi.org/10.1109/COMST.2019.2939407.
4. Wu, Y.; et al. IEEE TCCN Special Section Editorial: Intelligent Resource Management for 5G and beyond. *IEEE Trans Cogn Commun Netw* **2020**, *6*, 2, 422–427. https://doi.org/10.1109/TCCN.2020.2993976.
5. Kalita, A.; Khatua, M. Autonomous Allocation and Scheduling of Minimal Cell in 6TiSCH Network. *IEEE Internet Things J* **2021**, *8*, 15, 12242–12250. https://doi.org/10.1109/JIOT.2021.3062115.
6. Man, L.; Committee, S. IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. **2012**.
7. Domingo-Prieto, M.; Chang, T.; Vilajosana, X.; Watteyne, T. Distributed PID-Based Scheduling for 6TiSCH Networks. *IEEE Communications Letters* **2016**, *20*, 5, 1006–1009. https://doi.org/10.1109/LCOMM.2016.2546880.
8. Pradhan, N. M.; Chaudhari, B. S.; Zennaro, M. 6TiSCH Low Latency Autonomous Scheduling for Industrial Internet of Things. *IEEE Access* **2022**, *10*, 71566–71575. https://doi.org/10.1109/ACCESS.2022.3188862.
9. Hermeto, R. T.; Gallais, A.; Theoleyre, F. Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey. *Comput Commun* **2017**, *114*. https://doi.org/10.1016/j.comcom.2017.10.004.
10. Accettura, N.; Vogli, E.; Palattella, M. R.; Grieco, L. A.; Boggia, G.; Dohler, M. Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation. *IEEE Internet Things J* **2015**, *2*, 6, 455–470. https://doi.org/10.1109/JIOT.2015.2476915.
11. Duy, T. P.; Dinh, T.; Kim, Y. Distributed cell selection for scheduling function in 6TiSCH networks. *Comput Stand Interfaces* **2017**, *53*, 80–88. https://doi.org/10.1016/j.csi.2017.03.008.
12. Wang, Q.; Vilajosana, X.; Watteyne, T. 6TiSCH Operation Sublayer (6top) Protocol (6P). **2018**.
13. Vilajosana, X.; Watteyne, T.; Vucinic, M.; Chang, T.; Pister, K. S. J. 6TiSCH: Industrial Performance for IPv6 Internet-of-Things Networks. *Proceedings of the IEEE* **2019**, *107*, 6, 1153–1165. https://doi.org/10.1109/JPROC.2019.2906404.
14. Chang, T.; Vučinić, M.; Guillén, X. V.; Dujovne, D.; Watteyne, T. 6TiSCH minimal scheduling function: performance evaluation. *Internet Technology Letters* **2020**, *3*, 4. https://doi.org/10.1002/itl2.170.

15. Chang, T.; Vučinić, M.; Vilajosana, X.; Duquennoy, S.; Dujovne, D. RFC 9033 6TiSCH Minimal Scheduling Function (MSF). **2021**.
16. Hauweele, D.; Koutsiamanis, R.-A.; Quoitin, B.; Papadopoulos, G. Z. Pushing 6TiSCH Minimal Scheduling Function (MSF) to the Limits. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.
17. Alexander, R.; et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. **2012**.
18. Kim, H. S.; Ko, J.; Culler, D. E.; Paek, J. Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey. *IEEE Communications Surveys and Tutorials*, Institute of Electrical and Electronics Engineers Inc., **2017**, 2502–2525. https://doi.org/10.1109/COMST.2017.2751617.
19. Thubert, P. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). **2012**.
20. Gnawali, O.; Levis, P. The Minimum Rank with Hysteresis Objective Function. **2012**.
21. Barlow, H. B. Unsupervised Learning. *Cambridge*, **1989**.
22. Serra, A.; Tagliaferri, R. Unsupervised learning: Clustering. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, Elsevier, **2018**, pp. 350–357. https://doi.org/10.1016/B978-0-12-809633-8.20487-1.
23. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Los Angeles*, **1967**.
24. Hartigan, J. A.; Wong, M. A. Algorithm AS 136: A K-Means Clustering Algorithm. **1979**.
25. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* **1987**, *20*, 53–65.
26. Cui, M. Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. *ACCAF* **2020**, *3*, 9–16. https://doi.org/10.23977/accaf.2020.010102.
27. Vijayalakshmi, K.; Anandan, P. A multi objective Tabu particle swarm optimization for effective cluster head selection in WSN. *Cluster Comput* **2019**, *22*, 12275–12282. https://doi.org/10.1007/s10586-017-1608-7.
28. El Khediri, S.; Fakhet, W.; Moulahi, T.; Khan, R.; Thaljaoui, A.; Kachouri, A. Improved node localization using K-means clustering for Wireless Sensor Networks. *Comput Sci Rev* **2020**, *37*. https://doi.org/10.1016/j.cosrev.2020.100284.
29. Ben Gouissem, B.; Gantassi, R.; Hasnaoui, S. Energy efficient grid based k-means clustering algorithm for large scale wireless sensor networks. *International Journal of Communication Systems* **2022**, *35*, 14. https://doi.org/10.1002/dac.5255.
30. Kaur, L.; Kad, S. Modified EECPK-means mid-point algorithm for enhancing network life expectancy in WSN. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, **2018**, pp. 1262–1265.
31. Nasr, S.; Quwaider, M. LEACH Protocol Enhancement for Increasing WSN Lifetime. In *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, Institute of Electrical and Electronics Engineers Inc., **2020**, pp. 102–107. https://doi.org/10.1109/ICICS49469.2020.239542.
32. Kim, J.; Lee, D.; Hwang, J.; Hong, S.; Shin, D.; Shin, D. Wireless Sensor Network (WSN) Configuration Method to Increase Node Energy Efficiency through Clustering and Location Information. *Symmetry (Basel)* **2021**, *13*, 3, 1–11. https://doi.org/10.3390/sym13030390.
33. Moussa, N.; Hamidi-Alaoui, Z.; El Belrhiti El Alaoui, A. ECRP: an energy-aware cluster-based routing protocol for wireless sensor networks. *Wireless Networks* **2020**, *26*, 4, 2915–2928. https://doi.org/10.1007/s11276-019-02247-5.

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