sensors

Article

Improved Cell Allocation Strategies Using K-Means Clustering in Congested 6TiSCH Environments

Fransiskus Xaverius Kevin Koesnadi and Sang-Hwa Chung *

Department of Information Convergence Engineering, Pusan National University, Busan 46241, Republic of 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 this study, we propose an innovative cell allocation strategy based on node position clustering using the K-means algorithm, specifically designed to address congestion and optimize resource distribution in the 6TiSCH network. Our mechanism effectively groups nodes into clusters, allowing for dynamic adjustment of cell capacities in congested areas by analyzing traffic patterns and the spatial distribution of nodes. This clustering approach enhances the efficiency of slot frame utilization and minimizes communication delays by reducing interference and improving routing stability. The proposed strategy leverages the clustering results to improve cell usage efficiency and reduce communication latency between nodes. By tailoring cell allocation to the specific traffic needs of each cluster, we significantly reduce packet loss, manage congestion more effectively, and enhance data transmission reliability. We evaluated the clustering method using the K-means algorithm through experiments with the 6TiSCH simulator. Additionally, we considered using objective functions in Routing Protocol for Low-Power and Lossy Networks (RPL), such as OF0 and MRHOF, to assess clustering results and their impact on throughput and packet delivery. Our method resulted in significantly improved average performance metrics. Under the OF0 routing protocol, we achieved a 30.01% latency reduction, a 15.95% faster joining time, an 8% higher packet delivery ratio, and a 13.82% throughput increase. Similarly, 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 with the MRHOF routing protocol. These findings highlight the effectiveness of the improved cell allocation strategy in congested 6TiSCH environments, offering a better solution for enhancing network performance in IIoT applications.

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

Citation: Koesnadi, F.X.K.; Chung, S.-H. Improved Cell Allocation Strategies Using K-Means Clustering in Congested 6TiSCH Environments. *Sensors* **2024**, *24*, 5608. https://doi.org/10.3390/s24175608

Academic Editor: Kit Yan Chan

Received: 18 July 2024
Revised: 22 August 2024
Accepted: 26 August 2024
Published: 29 August 2024

1. Introduction

The Internet of Things (IoT) connects physical devices, enabling wireless sensor nodes to share environmental data and facilitate system interaction with the real world. Wireless sensor networks (WSNs) have gained interest for their diverse applications in analyzing environmental conditions and guiding appropriate responses [1]. 6TiSCH addresses the needs of low-power wireless networks by ensuring end-to-end reliability and deterministic latency. It integrates IPv6 with IEEE802.15.4 time-slotted channel-hopping (TSCH), providing industrial-grade performance and seamless internet connectivity that are vital for the Industrial Internet of Things (IIoT) [2]. Therefore, the present work identifies and addresses the challenges in routing strategies and security management to enhance their efficiency and effectiveness while striving for greater simplicity in these areas [3,4]. In IEEE 802.15.4's TSCH mode, limited broadcast opportunities occur at specific times and channels. Routers use enhanced beacon (EB) frames to announce the network; these EBs can include critical information that allows new or long-sleeping nodes to make the most of these rare broadcasts, thereby ensuring efficient slot usage [5]. Scheduling techniques in TSCH networks vary, with some using centralized methods employing root node managing schedules and cell allocations and others using link-based schemes employing MAC addresses. However, current TSCH approaches may not meet future IIoT needs. Hence, more lightweight and energy-efficient scheduling and routing algorithms are necessary [6].

The 6TiSCH architecture is crucial for the IIoT, ensuring reliable communication through effective cell allocation to prevent packet loss in high-traffic areas. In factories, field devices and sensors require high reliability and timely information exchange [7]. Efficient cell allocation based on node position directly influences network performance and power consumption [8]. Optimizing cell allocation in 6TiSCH-based wireless sensor networks is essential for reliable data transmission, especially in densely deployed environments [9] comprising industrial automation, smart city infrastructure, and environmental monitoring implementations. This work employs the K-means algorithm for label-free clustering to optimize cell allocation in 6TiSCH networks by clustering nodes based on their spatial distribution and link performance, which helps to avoid the need for further iterations. Centroid nodes are designated as cluster heads and other nodes are associated with the nearest center, resulting in reduced transmission hops, simplified neighbor discovery, and enhanced scalability. This clustering strategy informs a dynamic cell allocation approach within the slot frame, which improves communication efficiency and minimizes collisions. Our proposed method efficiently manages intra-cluster subchannel allocation and inter-cluster interference through distributed learning, which optimizes cell utilization, reduces overhead, and enhances overall network efficiency. This optimization is reflected in the network topology, where parent nodes function as cluster heads and sink nodes serve as data collection centers. Dynamic cell allocation within each cluster is guided by communication demands and traffic density, thereby minimizing congestion and improving transmission efficiency. Furthermore, routing paths and network organization are determined by RPL while considering factors such as OF0 and MRHOF. Network performance is assessed based on the joining time, packet delivery ratio, end-to-end latency, and throughput. By maintaining a constant number of nodes, this study prioritizes the evaluation of clustering and cell allocation strategies while accounting for the static conditions typical of industrial and IIoT environments [10].

Our goal is to enhance resource utilization in 6TiSCH networks and to reduce packet loss based on the node distribution. K-means clustering is applied to nodes based on nearby communication needs to optimize packet routes, minimize packet loss and latency, and ensure reliable data communication by division based on nodes' distance in a grid area. This method also prevents slot collisions and maintains efficient cell placement by ensuring that cells from the same communication node do not occupy the same slot frame. The proposed system allocates additional slots in the next available frame if the available slots are insufficient.

The contribution of this study can be summarized as follows:

- We improve cell allocation in the 6TiSCH network, optimizing resource utilization, reducing packet loss, and ensuring reliable communication.
- We utilize K-means clustering to group nodes based on nearby communication needs, which minimizes packet loss and latency, improves delivery ratio and throughput, and prevents slot collisions by maintaining efficient cell placement.
- We enhance 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 cell allocation.

The structure of this paper is as follows: Section 2 provides an overview of the simulator and tools, and discusses the problem of clustering techniques; Section 3 reviews related works for comparison; Section 4 outlines the proposed density clustering mechanism for node-centric cell allocation in the 6TiSCH network; Section 5 presents the performance evaluation, covering metrics such as packet delivery ratio, latency, time allocation for every node to join the network, and throughput; finally, Section 6 concludes the paper with a summary of the findings.

2. Related Work

This section provides an overview of studies on clustering techniques in wireless sensor networks and related research. Our method employs group nodes for administration and distributed job execution, including management of resources such as the congestion level of neighboring nodes. Although the primary benefit of clustering systems is lower energy consumption, clustering also addresses network heterogeneity and mobility, load balancing, and energy conservation.

One application of cluster formation in WSN, as demonstrated by [11], employs particle swarm optimization and Tabu algorithm techniques. These methods improve energy efficiency, network longevity, and protocol performance by rotating cluster head roles based on energy levels, utilizing multi-hop routing algorithms, and incorporating fault tolerance mechanisms. However, the selection of cluster heads remains constrained by the parameter settings, which can result in elevated energy consumption. Space partition in K-means clustering using optimal K-means was employed in [12] to address area partition by forming space partitions in a wireless sensor network, although the node issuance algorithm remains unclear. A modified grid-based wireless sensor network utilizing the K-means approach [13] has been tested to enhance energy efficiency and extend the lifespan of large-scale wireless sensor networks. This method reduces overall energy consumption and improves network longevity by dividing the network area into grids and applying K-means clustering within each grid cell; however, it may impose limitations on deploying heterogeneous sensor networks that rely on a static parent node.

Hierarchical protocols such as 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 protocol conserves energy and improves scalability. Modifying LEACH, as proposed by [14], 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 the 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 [15], where an algorithm was developed to select a primary and secondary cluster head in order to maintain network performance and efficiency while ensuring data flow continuity even in the event of node failure. Another method [16] focuses on improving the average energy consumption per node in each round and the number of surviving nodes per round, although routing algorithms still need to address high latency issues.

To minimize frequent reclustering and associated energy overhead, [17] 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. In [18], the authors employed a joint optimization strategy using the A3C learning algorithm to minimize the system energy consumption of the NOMA cluster while ensuring safe and low-delay computational loading in dynamic vehicle networks.

Our research focuses on cell allocation and its impact on wireless sensor network implementation, unlike previous works that typically concentrate on node localization. We examine how node grouping performance affects each cell allocation, particularly in TSCH, as well as its indirect relation to the cell allocation system in WSN, while 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, we propose a dynamic allocation method for enhancing network efficiency that makes adjustments based on node mobility and network responsiveness during data transmission.

3. Background

3.1. TSCH

IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) 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 that require precise timing and coordination, such as industrial control systems or environmental sensor networks [19]. While TSCH specifies functionality, its implementation depends on network architecture and application needs. Channels are managed by dividing time into fixed time slots and periodically changing frequency channels, with each node assigned specific times for data transmission or reception and synchronized with neighboring nodes, thereby minimizing channel access conflicts [20].

TSCH is a MAC mode in the IEEE 802.15.4e standard that allocates time slots for data transmission or reception. These allocations can be static or dynamic based on network requirements. Time slots are organized into slot frames, which define cycles of time slots that repeat according to the slot frame size [21]. Nodes communicate within these time slots, where they are synchronized through a shared channel hopping schedule that periodically changes the frequency channels to reduce interference. This ensures precise communication and time synchronization between nodes, with the slot frame length being adjustable based on network and application needs. TSCH operation utilizes about 16 available channels, which allows for high coordination within the network and reduced channel access conflicts [22].

3.2. 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 [23]. The 6TiSCH architecture combines IPv6 with IEEE 802.15.4e TSCH MAC, organizing time into slots within slot frames in which each slot is a distinct time interval for node functions [24,25]. Cells representing communications capability are controlled by the 6TiSCH Operational Sublayer Protocol (6P), which manages cell allocations and time slot usage to ensure reliable communication and resource efficiency in IIoT environments [25,26].

The Minimal Scheduling Function (MSF) in 6TiSCH networks adjusts cell allocation based on utilization metrics to meet dynamic bandwidth requirements [27]. 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 [28]. MSF coordinates time slot and cell distributions, synchronizes node transmission schedules, and assigns specific roles to nodes [29]. This protocol ensures efficient bandwidth use, low latency, and high dependability, making it suitable for real-time control and monitoring systems [25].

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

The Routing Protocol for Low-Power and Lossy Networks (RPL) is designed for resource-constrained environments such as wireless sensor network. RPL offers efficient and reliable routing by addressing challenges such as node mobility, dynamic topologies, and unpredictable conditions. It supports hierarchical structures and uses the Destination-Oriented Directed Acyclic Graph (DODAG) to establish optimal paths toward specific destinations while considering topology and delivery needs [30]. RPL also dynamically reorganizes paths in response to network changes, making it ideal for WSN and IoT applications thanks to its efficiency, reliability, and ability to manage control overhead and memory constraints [31].

3.3.1. Objective Function Zero (OF0)

The Objective Function Zero (OF0) in the RPL guides nodes in selecting and optimizing routes within an RPL instance by using available information objects. Instead of providing 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. To compute the current node's rank R_N, the step of rank S_p is multiplied by the rank factor R_f and adjusted with a stretch factor S_r that is constrained by the configured rank stretch. This rank is then added to the preferred parent's rank to determine the node's overall rank. By calculating the rank, OF0 enables a node to participate in a DODAG version that provides satisfactory connectivity, with the minimal hop rank increase defining the lowest rank increase to any potential parent. OF0 does not guarantee optimization according to any specific metric; connectivity validation is specific to implementation and link type, which falls outside OF0's scope [32].

3.3.2. 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 [33].

3.4. Unsupervised Learning and K-Means Clustering

Unsupervised learning aims to uncover patterns in unlabeled data by discovering hidden structures without predefined labels. Unlike supervised learning, which uses labeled datasets, unsupervised learning focuses on clustering, identifying correlations, and reducing data dimensionality. The clustering process relies on similarity metrics, with the Euclidean distance being the most common [34]. Other metrics, such as the correlation coefficient, may be used as well. Cluster analysis identifies high-density groups with similar observations within clusters. The number of clusters and similarity metric both have significant impacts on the results [35]. The K-means algorithm is a widely used unsupervised technique for clustering data into predefined clusters [36]. It partitions a dataset of N observations into k clusters by iteratively assigning observations to the nearest centroid, calculated as the mean of the points in a cluster. Starting with random centroids, the algorithm alternates between assigning observations based on the Euclidean distance and recalculating the centroids until reaching either convergence or a set number of iterations [37]. Using the Euclidean distance, K-means clustering seeks to minimize the within-cluster sum of squares (WCSS). Challenges include handling non-spherical clusters, outliers, and empty clusters, which are often resolved by reassignment or adjusting the number of clusters k. The elbow method determines the optimal k by plotting the WCSS against various values of k to find the 'elbow' point. Silhouette analysis assesses cluster compactness and separation, aiding in the selection of an appropriate k value.

Silhouette Score and Elbow Method

The Silhouette Score measures how well an object fits within its cluster compared to other clusters, ranging from -1 to 1. Positive values indicate that objects are more similar to their cluster members than those in other clusters, while negative values suggest the opposite [38]. For an object i, the cluster B with the minimum distance to i (i.e., $d(i, B) = b(i)$) is considered the closest alternative if i cannot be assigned to cluster A. Determining $b(i)$ requires considering clusters other than A; hence, k must be greater than 1. The Silhouette Score $s(i)$ combines $a(i)$ and $b(i)$ to evaluate cluster quality [38].

In K-means clustering, the elbow method identifies the optimal number of clusters k by plotting the distortion (inertia value) against k. The distortion measures the total distance between points and their cluster center. As k increases, points become closer to their centers, reducing the distortion. The rate of decrease slows, forming an elbow in the plot. The k at this elbow is optimal, as further increases in k yield minimal improvements in the explained variance [39].

4. Node Clustering

4.1. Cluster Initialization and Parameter Enhancement

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

```
Input: $nodes$ as number of nodes
Output: $cluster\_center$

1:  Initialize: $cluster\_center \leftarrow []$, $new\_center \leftarrow []$
2:
3:  Class RandomConnectivity
4:  function UpdateClusterCenter
5:      for $center, node$ in $clusters.items()$ do
6:          if $node$ is not empty then
7:              $x_{total} \leftarrow$ add values of $x$ coordinate of nodes
8:              $y_{total} \leftarrow$ add values of $y$ coordinate of nodes
9:              $x_{average} \leftarrow x_{total} / num\_nodes$
10:             $y_{average} \leftarrow y_{total} / num\_nodes$
11:             $new\_center$.append($x$: $x_{average}$, $y$: $y_{average}$)
12:         else
13:             $new\_center$.append($\{'x': center[0], 'y': center[1]\}$)
14:         end if
15:     end for
16:     return $new\_center$ as new node centroid based on its coordinate
```

Figure 1. Pseudocode for K-means cluster center initialization and updating.

4.2. Clustering Parameter Enhancement

The coordinates of the cluster centers (x'_{C_i}, y'_{C_i}) are iteratively updated using the K-means optimization method based on the average coordinates of the nodes within each cluster C_i. To assign nodes to their nearest clusters, the algorithm computes the Euclidean distance $d(N, C_i)$ between each node N and the updated cluster centers (x'_{C_i}, y'_{C_i}) as shown in Equation (1), starting from the previous centers (x_{C_i}, y_{C_i}). This process establishes proximity-based clusters with each node's coordinates denoted by x_N and y_N, contributing to the calculation of the updated cluster centers.

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

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 in Equation (4) 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. The algorithm illustrated in Figure 2, outlines the procedures involved in the optimization process, including node assignment and optimal k selection.

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

1:  Initialize: $distances \leftarrow []$
2:
3:  function $CoordinateClustering$
4:  for $node$ in each $nodes$ do
5:      for each $center$ in $cluster\_center$ do
6:          $distance \leftarrow Euclidean\_distance(node, center)$
7:          $distances$.append($distance$)
8:      end for
9:      $nearest\_cluster\_index \leftarrow$ index of $minimum\_value$ in $distances$
10:     assign $node$ to the $near\_cluster$
11: end for
```

Figure 2. Pseudocode for node granting and optimal K selection: Coordinate clustering.

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 using Equation (4), with x and y coordinate points and (μ_i) centroids used to evaluate the clustering quality and iteratively update the cluster centers. Consequently, the algorithm shown in Figure 3, determines the optimal number of clusters for the final K-means clustering phase.

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

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

```
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 $optimal\_k$

1:  Initialize: $silhouette\_score \leftarrow$ [], $WCSS \leftarrow$ [], $distance \leftarrow$ []
2:  Initialize: $best\_silhouette\_score \leftarrow$ -1, $best\_k \leftarrow 0$
3:
4:  function ClusteringAlgorithm
5:  for $k$ in $K_{range}$ do $\leftarrow$ Start cluster centers at random inside $r$'s boundaries.
6:      for $i$ from 1 to $max\_iterations$ do
7:          for each data point $r_i$ in $r$ do
8               assign $r_i$ to $near\_cluster$ $\leftarrow$Assign the closest cluster points in $r$
9:          end for
10:         for each $new\_center$ in $near\_cluster$ do
11:             $UpdateClusterCenter(cluster\_center)$ $\leftarrow$ recalculate center
12:         end for
13:         if $cluster\_center$ not change then
14:             break
15:         end if
16:     end for
17:     $WCSS[k]$ $\leftarrow$For assignments, compute the within-cluster sum of squares
18:     if k > 1 then
19:         $silhouette\_scores[k]$ $\leftarrow$ Compute the assignment's silhouette score
20:         if $silhouette\_scores[k] > best\_silhouette\_score$ then
21:             $best\_silhouette\_score = silhouette\_scores[k]$
22:             $best\_k = k$
23:         end if
24:     end if
25: end for
```

Figure 3. Pseudocode for node granting and optimal K selection: Clustering algorithm.

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 that 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, which is a crucial component of the data transmission process, multiple nodes (such as Node A and Node B) may attempt to occupy the same cell allocation. This inherent allocation process can lead to conflicts and interference, reducing the packet delivery ratio and increasing latency. In addition, the latency can increase significantly for every 0.1 second delay per packet. In the absence of node clustering, nodes compete for optimal cell allocation within the slot frame. As network density increases, this competition becomes more intense, complicating the allocation process and reducing efficiency. The high volume of nodes vying for limited slots can disrupt allocation and lead to significant delays in data delivery. Additional cell allocations may be required to manage traffic spikes and meet strict latency requirements in order to ensure smooth and timely communication, as illustrated in Figure 4.

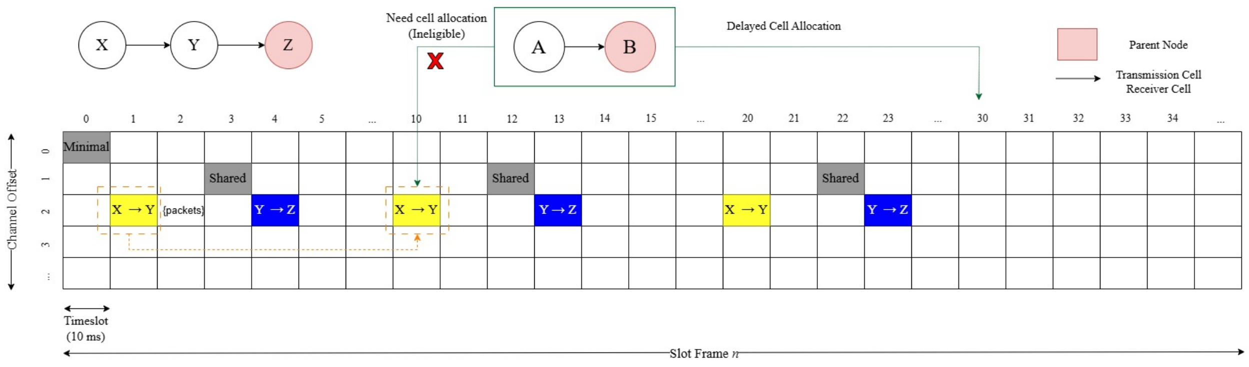


Figure 4. TSCH scheduling based on period traffic.

As shown in Figure 5, the cell allocation process at each node begins by connecting multiple nodes back to the central point, which is the parent node, in order to obtain available cells. If cell availability is low due to high density, nodes must wait or request an alternative cell, leading to delays and packet loss. When cells are available, data can be sent with low latency; however, conflicts and inefficiency can still arise, especially when node density is low.

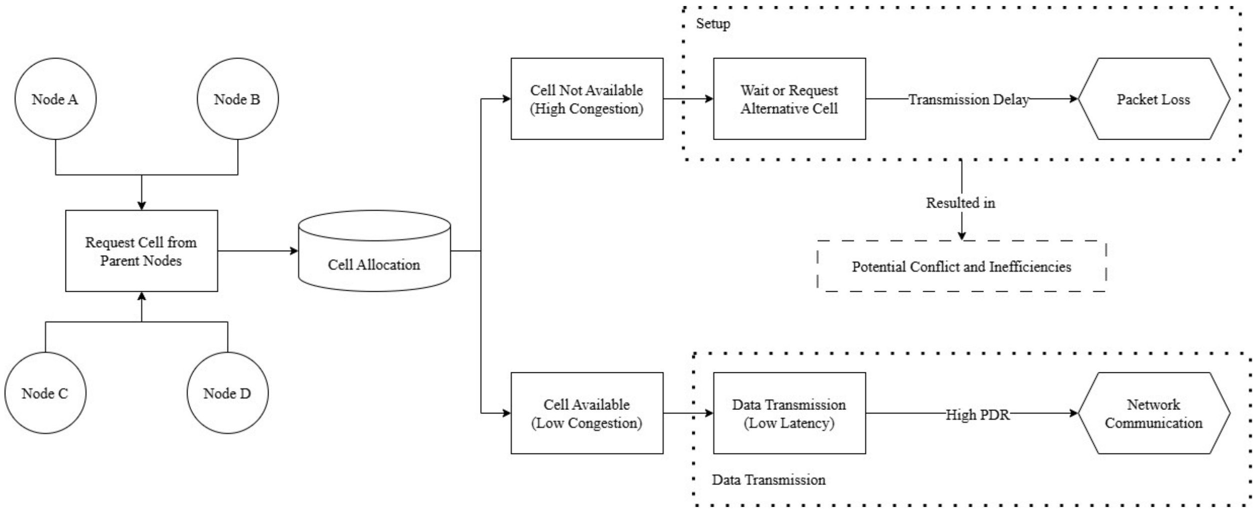

Setup
Node A
Node B
Node C
Node D
Request Cell from Parent Nodes
Cell Allocation
Cell Not Available (High Congestion)
Wait or Request Alternative Cell
Transmission Delay
Packet Loss
Resulted in
Potential Conflict and Inefficiencies
Cell Available (Low Congestion)
Data Transmission (Low Latency)
High PDR
Network Communication
Data Transmission

Figure 5. As-is diagram of default cell allocation in TSCH.

In contrast, Figure 6 illustrates an advanced method that utilizes K-means algorithm-based clustering to partition nodes based on their spatial positions and designates an optimized parent node, called the sink node. The resulting clusters inform a more efficient cell allocation process by assessing the cell requirements for each allocated frame slot. This method ensures that the allocation can continue seamlessly to the subsequent frame slot with minimal delay, even in the absence of an available cell. This approach effectively reduces node competition to maintain a high packet delivery ratio.

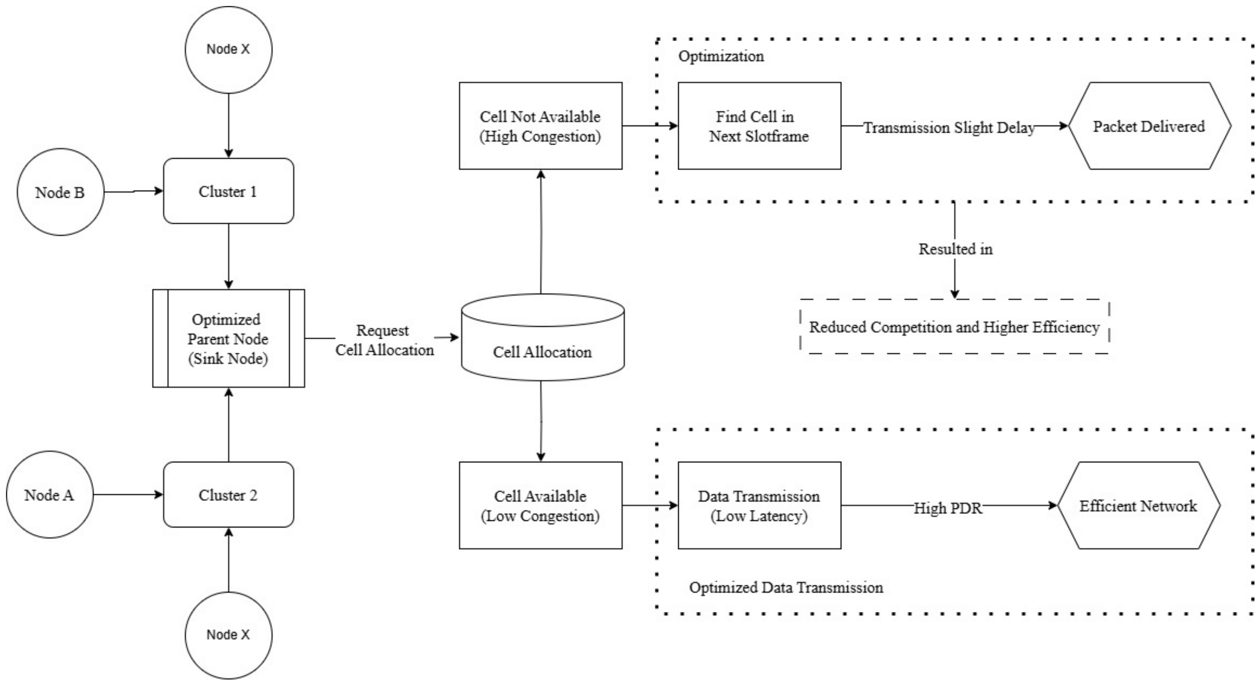

Node X
Node B
Cluster 1
Optimized Parent Node (Sink Node)
Request Cell Allocation
Cell Allocation
Node A
Cluster 2
Node X
Cell Not Available (High Congestion)
Optimization
Find Cell in Next Slotframe
Transmission Slight Delay
Packet Delivered
Resulted in
Reduced Competition and Higher Efficiency
Cell Available (Low Congestion)
Data Transmission (Low Latency)
High PDR
Efficient Network
Optimized Data Transmission

Figure 6. To-be diagram of cell allocation through clustering approach.

6. Cell Allocation Implication of Nodes

As illustrated by the algorithm in Figure 7, the improved cell allocation method optimizes communication cell allocation in the TSCH network. The algorithm begins with inputs such as *S*, representing the number of slot frames used by cells in a cluster, and *current_cluster*, indicating the network's current clustering status. It then compares a node's current cluster association with its neighbors using *GetCluster* and 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. Each randomly determined node location result is selected by looking at the distance between the modes to be determined in the inter-cluster or intra-cluster category, which is then used as a reference when determining the sink node. 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. This phase aims to divide the determination of each cell for the allocation of frame slots that have been determined based on the shared cells available for each frame slot used, in which case the cell is updated.

Information such as time slots and available channels is submitted to each clustered cell list. If the current node and its neighbors belong to the same cluster (as identified by *GetCluster*), then the intra-cluster strategy focuses on nearby nodes. Cluster formation enhances communication efficiency and improves time slot utilization. During this phase, active TX cells are added when the number of used slot frames exceeds the cluster capacity. When demand decreases, TX cells are reallocated to areas with higher user density, thereby optimizing resource usage and increasing throughput.

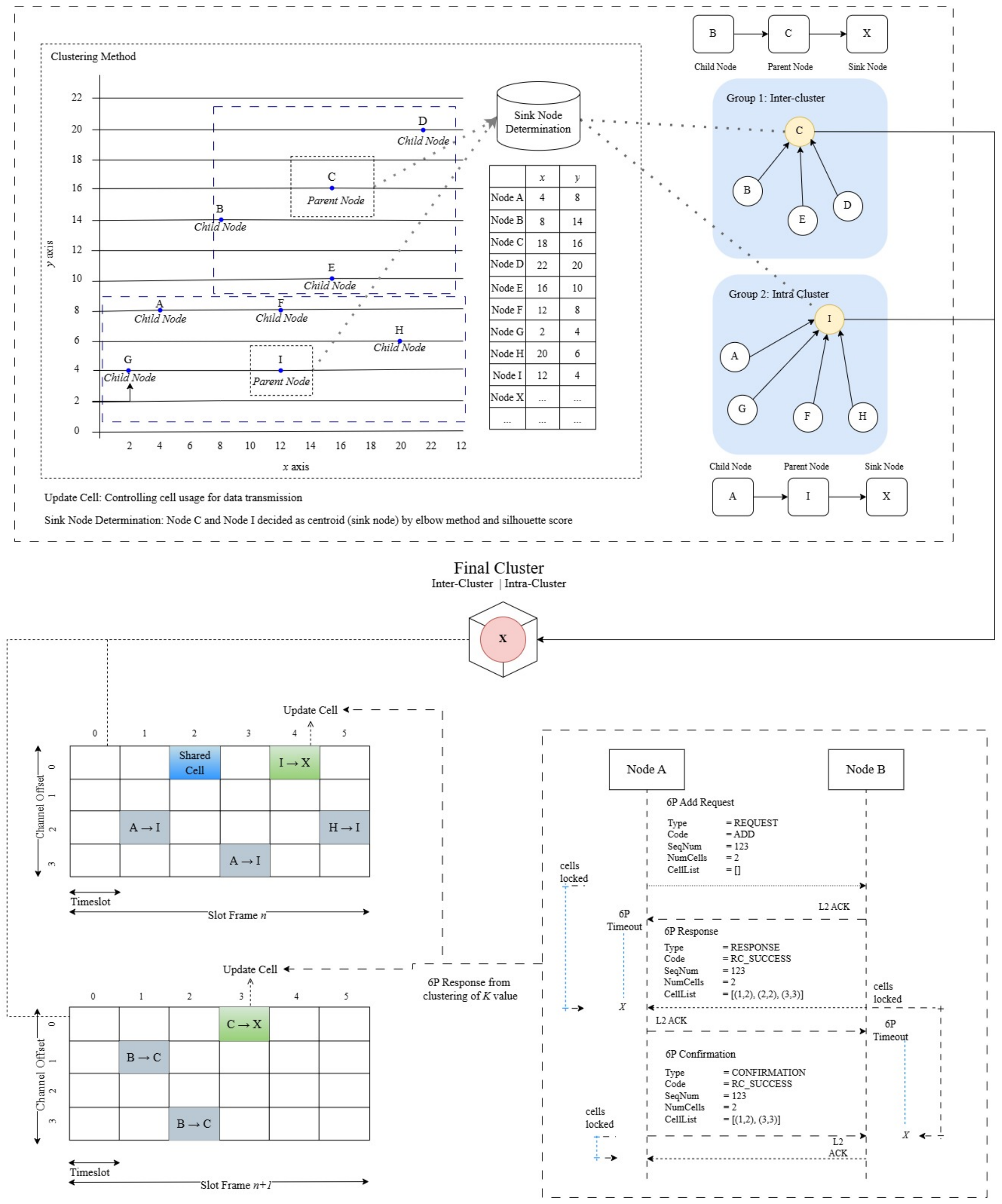
An inter-cluster strategy is proposed for nodes in different clusters using gateway or router nodes to facilitate communication between clusters, allowing the network to expand and adapt without disrupting operations. To prevent resource wastage, active TX cells are removed if fewer slot frames are available or are relocate to meet inter-cluster needs. Effective node clustering improves packet transmission efficiency by assigning specific cells based on a predetermined schedule. We consolidate resources by removing unnecessary cells, which reduces packet queuing and enables efficient communication between parent and child nodes. The 6P transaction processing minimizes signaling overhead and the nodes intelligently control cell usage, reducing 6P requests and acknowledgments while maintaining reliable connections.

```
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

1:  ▷ Examine and contrast the current node's cluster with that of its neighboring node
2:  Procedure: $GetCluster(neighbor, current\_cluster)$
3:  for each slot frame $i$ in $S$ do
4:      $clustering[neighbor] \leftarrow$ Get the neighbor node's cluster information
5:  end for
6:
7:  ▷ Assigning communication cells to network nodes
8:  repeat
9:      $NeighborCluster = GetCluster(S.neighbor, current\_cluster)$
10:     $SelfCluster = GetCluster(Self, current\_cluster)$
11:     if $SelfCluster = NeighborCluster$ then
12:         ▷ If the requesting and the neighbor nodes are in the same cluster
13:         $ExecuteIntraClusterStrategy(S)$
14:         if $S > Current\_Clustering$
15:             add active $TXcell$
16:         else
17:             relocate active $TXcell$
18:         end if
19:     else
20:         ▷ If the requesting and the neighbor nodes are in a different cluster
21:         $ExecuteInterClusterStrategy(S)$
22:         if $S < current\_cluster$
23:             delete active $TXcell$
24:         else
25:             relocate active $TXcell$
26:         end if
27:     end if
28: until END
```

Figure 7. Pseudocode for enhanced TSCH cell allocation based on clustering.

The proposed scheme, depicted in Figure 8, introduces innovative clustering-based cell allocation and dynamic reallocation strategies that enhance slot frame utilization and reduce signaling overhead, significantly improving network performance, as previously discussed regarding the K-means implementation. The K-means clustering process here has a complexity of approximately $O(n \times k \times i)$, where n is the number of nodes, k is the number of clusters, and i is the number of iterations until convergence is achieved. The cell allocation process in the slot frames involves evaluating and assigning cells to each node in the cluster, with an estimated complexity around $O(s \times n)$, where s is the number of slot frames and n is the number of nodes. The optimized 6P transaction algorithm also considers cell usage based on dynamic demand, with the complexity being generally linear with the number of nodes.



	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	...	...
...	...	...

6P Add Request

Type	= REQUEST
Code	= ADD
SeqNum	= 123
NumCells	= 2
CellList	= []

6P Response

Type	= RESPONSE
Code	= RC_SUCCESS
SeqNum	= 123
NumCells	= 2
CellList	= [(1,2), (2,2), (3,3)]

6P Confirmation

Type	= CONFIRMATION
Code	= RC_SUCCESS
SeqNum	= 123
NumCells	= 2
CellList	= [(1,2), (3,3)]

Figure 8. Scheme of the proposed clustering for cell allocation.

7. Performance Evaluation

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

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) [13], ECRP clustering [17], and the default MSF [27]. 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 on the same minimal scheduling function. The evaluation involved randomly placing nodes in an area for four scenarios involving 100, 150, 200, and 300 nodes, with each node positioned at specific coordinates. The number of nodes was chosen based on related works 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 was OF0 [32], 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 was MRHOF [33], which integrates metrics such as rank, which indicates link quality, and incorporates hysteresis to prevent excessive path changes, thereby minimizing latency and overhead.

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

Our 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. Measuring the time required for a packet to travel from its source to its destination provides the end-to-end latency, indicating how responsive the network is. The joining time indicates how quickly a new node integrates into the network. Throughput, as assessed by Equation (4), measures the data rate successfully sent from one node to another within a certain period or the number of packets sent from each source node to the destination node while considering the packet size, 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 s
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 ms
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 (OF0) 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 9. 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, as this value may be slightly higher due to the previously described clustering process affecting the scheduling process for each frame slot. 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. Figure 9 also demonstrates that the end-to-end latency with the OF0 routing protocol remained under 100 s even with 300 nodes, which can be attributed to the effective distribution of slot frames and time slots. The end-to-end latency for clustering on cell allocation also showed critical improvements compared to grid-based K-means. Noteworthy here is that slot frame distribution slightly affects latency, as the node coordinates used in slot frame distribution are divided appropriately for slot frames and individual time slots.

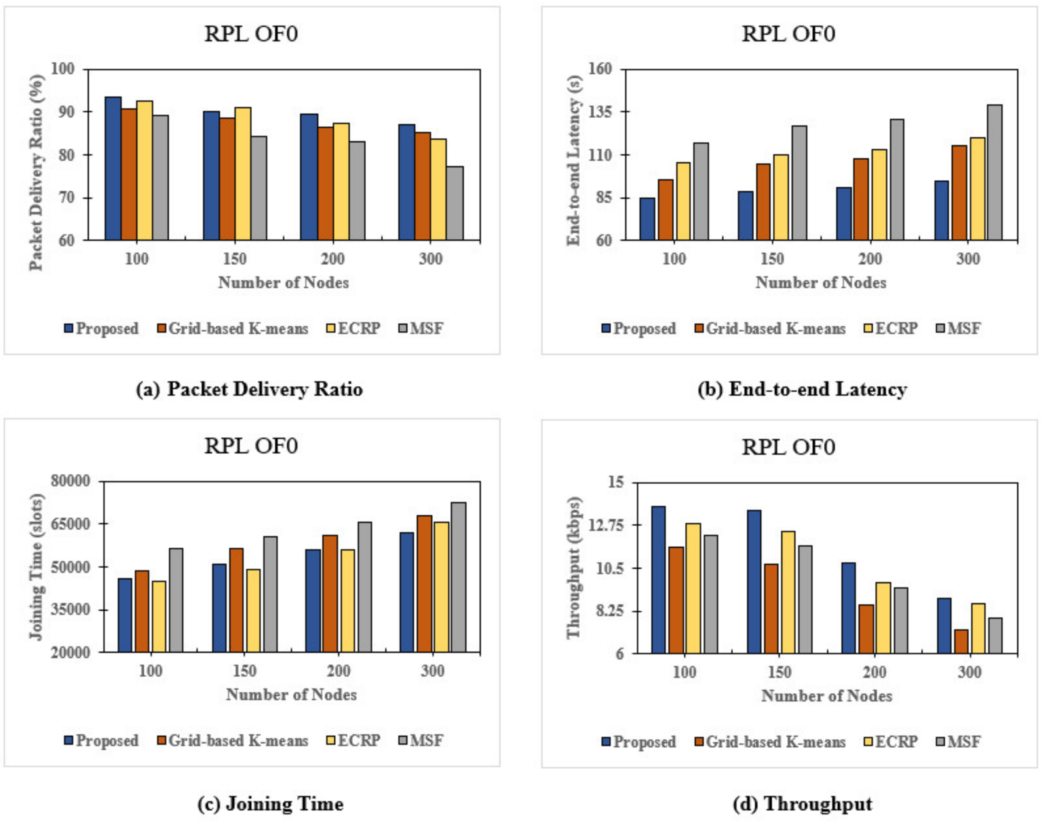


Figure 9. Graphs showing results for the Objective Function Zero (OF0) routing protocol: (**a**) Packet Delivery Ratio (PDR), (**b**) end-to-end latency, (**c**) joining time, and (**d**) throughput.

Figure 9 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, mainly 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. The energy efficiency of clustering tends to be high in the following moments, as it heavily relies on the residual energy of each group of nodes; thus, our proposed method could determine how the sink node might accept and deliver some packets for reserved nodes. Figure 9 shows favorable throughput results, with each packet's transmission speed enhanced by OF0's low-overhead path selection. We believe that the throughput value will still be high through packet sending, as it might barge the transmission cells. Energy efficiency clustering reduces the throughput margin to 8.89 kbps at 300 nodes, yet overall throughput stays high due to effective 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.

8.2. Minimum Rank with Hysteresis Routing Protocol

Our next experiment utilized the Minimum Rank with Hysteresis Objective Function (MRHOF) routing protocol. We used the same parameters as for 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; unlike OF0, it also supports multi-path routing. Clustering improves PDR by allocating cells directly. The clustering result improves the PDR value by directly determining the cells to be used. The highest PDR, observed with 100 nodes, reached 98.15%. This decreased to 93.35% with 300 nodes, reflecting the increased traffic load on network resources. The end-to-end latency performance metric remained stable despite a slight increase compared to grid-based K-means, reaching 112.91 s for 300 nodes due to fewer retransmissions.

The joining time was consistent with other K-means clustering methods, achieving 55,219 slots compared to 55,394 slots for ECRP clustering. Congestion at the cluster head delays the transmission queue when transmitting to other slot frames, as depicted in Figure 8. Increased data traffic or competition for cluster resources likely caused congestion between cluster heads, delaying the transmission queue. Figure 10 also shows the throughput values for the data transmission speed, which differ from the OF0 RPL. Throughput, which is closely related to PDR, decreased from 17.12 kbps to 9 kbps as the number of nodes increased from 100 to 300, likely due to higher message transmission during the 6p transaction in grid-based K-means; the throughput decreases linearly as the number of nodes increases. Figure 10 highlights an anomaly in which fewer nodes encountered difficulties due to clustering on low nodes in top-6 protocol cell allocation, resulting in performance similar to MSF without clustering. Compared to MSF, our method showed a 12.34% increase in average PDR, a 21.06% reduction in latency, a 12.68% decrease in joining time, and a 25.97% increase in throughput speed.

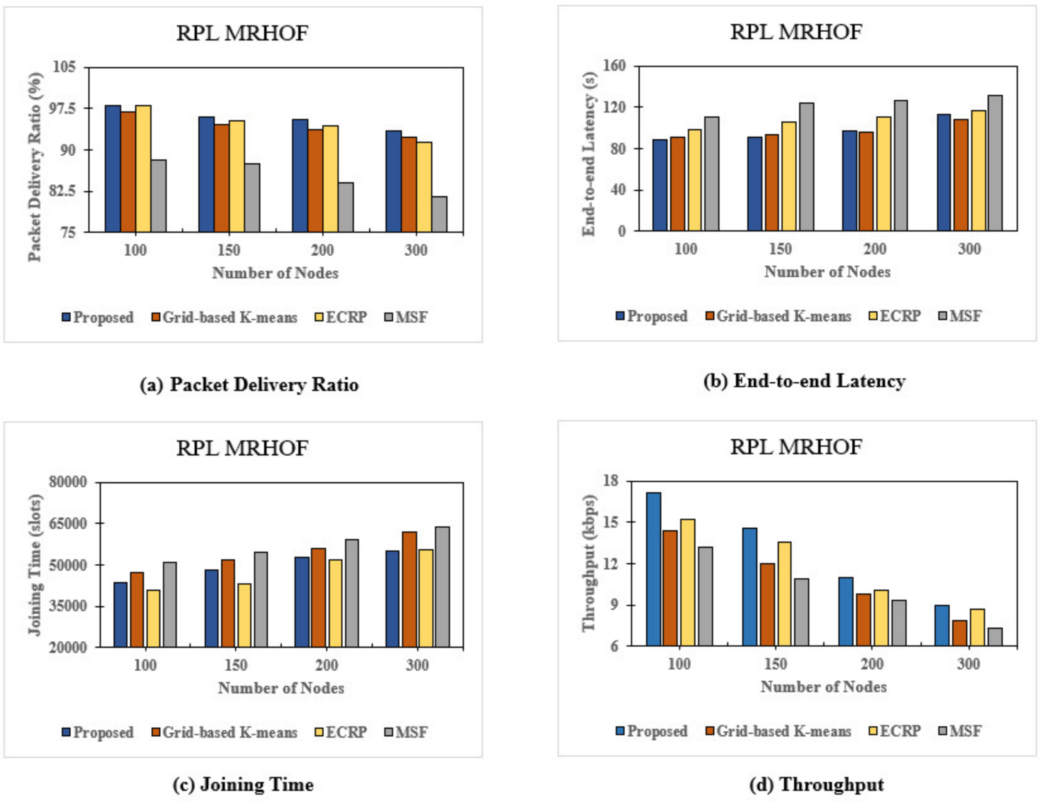


Figure 10. Graph results of the Minimum Rank with Hysteresis (MRHOF) routing protocol: (**a**) Packet Delivery Ratio (PDR), (**b**) end-to-end Latency, (**c**) joining time, and (**d**) throughput.

9. Conclusions

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, which influences 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, while 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 in order to manage resources based on cluster density. The clustering results determine the number of slots arranged in a time frame, while cell operations are adjusted 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. At the same time, the strategy prioritizes allocating unicast cells immediately after the receiving cell, which is intended to minimize latency. Our 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 achieved 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%. The number of nodes in the network was assumed to be constant in order to maintain the focus on optimizing cell allocation and clustering techniques. In real-world scenarios, node loss from hardware failure, signal weakening, or environmental shadowing can alter network topology. Future research could incorporate models that address dynamic conditions and varying network sizes, or explore hybrid systems that combine multiple clustering techniques involving multi-objective allocation. These advancements could enhance algorithm performance by achieving energy savings through optimal time slot selection and improving scalability for faster response times with the alignment of other objective functions.

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 and Communications Technology Planning and Evaluation). This work is supported by a 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.

Data Availability Statement: Data are contained within the article.

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 Trans. Internet Things* **2022**, *3*, 3. [CrossRef]
2. Municio, E.; Daneels, G.; Vučinić, M.; Latré, S.; Famaey, J.; Tanaka, Y.; Brun, K.; Muraoka, K.; Vilajosana, X.; Watteyne, T. Simulating 6TiSCH Networks. *Trans. Emerg. Telecommun. Technol.* **2018**, *30*, 3. [CrossRef]
3. Vilajosana, X.; Watteyne, T.; Chang, T.; Vucinic, M.; Duquennoy, S.; Thubert, P. IETF 6TiSCH: A Tutorial. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 595–615. [CrossRef]
4. Tanaka, Y.; Brun-Laguna, K.; Watteyne, T. Demo: Simulating a 6TiSCH Network using Connectivity Traces from Testbeds. In Proceedings of the IEEE Conference, Paris, France, 7–12 July 2019.
5. Kurunathan, H.; Severino, R.; Koubaa, A.; Tovar, E. IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1989–2010. [CrossRef]
6. De Guglielmo, D.; Seghetti, A.; Anastasi, G.; Conti, M. A Performance Analysis of the Network Formation Process in IEEE 802.15.4e TSCH Wireless Sensor/Actuator Networks. In Proceedings of the 2014 IEEE Symposium on Computers and Communications (ISCC), Funchal, Portugal, 23–26 June 2014; pp. 1–6. [CrossRef]
7. Wu, Y.; Simeonidou, D.; Wang, C.X.; Yu, F.R.; Choi, S.; Xue, G.; Ksentini, A. IEEE TCCN Special Section Editorial: Intelligent Resource Management for 5G and beyond. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 422–427. [CrossRef]
8. Kalita, A.; Khatua, M. Autonomous Allocation and Scheduling of Minimal Cell in 6TiSCH Network. *IEEE Internet Things J.* **2021**, *8*, 12242–12250. [CrossRef]
9. Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of wireless sensor network: An Up-to-Date Survey. *Appl. Syst. Innov.* **2020**, *3*, 14. [CrossRef]
10. Ahmed, S.F.; Alam, M.S.B.; Hoque, M.; Lameesa, A.; Afrin, S.; Farah, T.; Kabir, M.; Shafiullah, G.M.; Muyeen, S.M. Industrial Internet of Things enabled technologies, challenges, and future directions. *Comput. Electr. Eng.* **2023**, *110*, 108847. [CrossRef]
11. Vijayalakshmi, K.; Anandan, P. A multi objective Tabu particle swarm optimization for effective cluster head selection in WSN. *Cluster Comput.* **2019**, *22*, 12275–12282. [CrossRef]
12. El Khediri, S.; Fakhet, W.; Moulahi, T.; Khan, R.; Thaljaoui, A.; Kachouri, A. Improved node localization using K-means clustering for wireless sensor network. *Comput. Sci. Rev.* **2020**, *37*, 100284. [CrossRef]
13. Ben Gouissem, B.; Gantassi, R.; Hasnaoui, S. Energy efficient grid based k-means clustering algorithm for large scale wireless sensor network. *Int. J. Commun. Syst.* **2022**, *35*, 14. [CrossRef]
14. Kaur, L.; Kad, S. Modified EECPK-Means Mid-Point Algorithm for Enhancing Network Life Expectancy in WSN. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 1262–1265.
15. Nasr, S.; Quwaider, M. LEACH Protocol Enhancement for Increasing WSN Lifetime. In Proceedings of the 2020 11th International Conference on Information and Communication Systems, ICICS 2020, Irbid, Jordan, 7–9 April 2020; pp. 102–107. [CrossRef]
16. 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* **2021**, *13*, 390. [CrossRef]
17. Moussa, N.; Hamidi-Alaoui, Z.; El Belrhiti El Alaoui, A. ECRP: An energy-aware cluster-based routing protocol for wireless sensor network. *Wireless Netw.* **2020**, *26*, 2915–2928. [CrossRef]
18. Ju, Y.; Cao, Z.; Chen, Y.; Liu, L.; Pei, Q.; Mumtaz, S.; Dong, M.; Guizani, M. NOMA-Assisted Secure Offloading for Vehicular Edge Computing Networks With Asynchronous Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 2627–2640. [CrossRef]
19. *IEEE Std 802.15.4-2011*; IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer. IEEE: Piscataway, NJ, USA, 2012.
20. Domingo-Prieto, M.; Chang, T.; Vilajosana, X.; Watteyne, T. Distributed PID-Based Scheduling for 6TiSCH Networks. *IEEE Commun. Lett.* **2016**, *20*, 1006–1009. [CrossRef]
21. Pradhan, N.M.; Chaudhari, B.S.; Zennaro, M. 6TiSCH Low Latency Autonomous Scheduling for Industrial Internet of Things. *IEEE Access* **2022**, *10*, 71566–71575. [CrossRef]
22. 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*, 84–105. [CrossRef]
23. 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*, 455–470. [CrossRef]
24. Duy, T.P.; Dinh, T.; Kim, Y. Distributed cell selection for scheduling function in 6TiSCH networks. *Comput. Stand Interfaces* **2017**, *53*, 80–88. [CrossRef]
25. Wang, Q.; Vilajosana, X.; Watteyne, T. 6TiSCH Operation Sublayer (6top) Protocol (6P). 2018. Available online: https://datatracker.ietf.org/doc/draft-ietf-6tisch-6top-protocol/12/ (accessed on 25 August 2024).
26. Vilajosana, X.; Watteyne, T.; Vucinic, M.; Chang, T.; Pister, K.S.J. 6TiSCH: Industrial Performance for IPv6 Internet-of-Things Networks. *Proc. IEEE* **2019**, *107*, 1153–1165. [CrossRef]
27. Chang, T.; Vučinić, M.; Guillén, X.V.; Dujovne, D.; Watteyne, T. 6TiSCH minimal scheduling function: Performance evaluation. *Internet Technol. Lett.* **2020**, *3*, 4. [CrossRef]

28. Chang, T.; Vučinić, M.; Vilajosana, X.; Duquennoy, S.; Dujovne, D. RFC 9033 6TiSCH Minimal Scheduling Function (MSF). 2021. Available online: https://datatracker.ietf.org/doc/rfc9033/ (accessed on 25 August 2024).
29. Hauweele, D.; Koutsiamanis, R.-A.; Quoitin, B.; Papadopoulos, G.Z. Pushing 6TiSCH Minimal Scheduling Function (MSF) to the Limits. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
30. Alexander, R.; Brandt, A.; Vasseur, J.P.; Hui, J.; Pister, K.; Thubert, P.; Levis, P.; Struik, R.; Kelsey, R.; Winter, T. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. 2012. Available online: https://datatracker.ietf.org/doc/html/rfc6550 (accessed on 25 August 2024).
31. 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 Commun. Surv. Tutorials* **2017**, *19*, 2502–2525. [CrossRef]
32. Thubert, P. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). 2012. Available online: https://datatracker.ietf.org/doc/rfc6552/ (accessed on 25 August 2024).
33. Gnawali, O.; Levis, P. The Minimum Rank with Hysteresis Objective Function. 2012. Available online: https://datatracker.ietf.org/doc/rfc6719/ (accessed on 25 August 2024).
34. Barlow, H.B. Unsupervised Learning. *Neural Comput.* **1989**, *1*, 295–311. [CrossRef]
35. Serra, A.; Tagliaferri, R. Unsupervised Learning: Clustering. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 1–3; pp. 350–357. [CrossRef]
36. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, CA, USA, 18–21 July 1965.
37. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser.* **1979**, *28*, 100–108. [CrossRef]
38. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
39. Cui, M. Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. *ACCAF* **2020**, *3*, 9–16.