

Article

# Data Exchange in Cluster Structure for Longevity of IoT

Soobin Jeon

School of Information Technology Engineering, Catholic University of Daegu, Gyeongsan-si 38430, Korea; marsberry@cu.ac.kr; Tel.: +82-10-5106-0388

Received: 10 January 2020; Accepted: 10 February 2020; Published: 12 February 2020

**Abstract:** In the Internet of Things (IoT), the scope of wireless sensor nodes is extended to things deployed in a pervasive world. For various IoT service applications, things can gather and share their information with each other through self-decision-making. Therefore, we cannot apply the existing information aggregation methods of wireless sensor networks to the IoT environment, which aim to transmit the collected data to only a sink node or a central server. Moreover, since the existing methods involve all the sensor nodes in the process of data exchange, they can cause an increase in the network traffic, delay of data transmission, and amount of energy consumed by things. In this paper, we propose a clustering-property-based data exchange method for efficient energy consumption in IoT networks. First, the proposed method assigns properties to each thing according to the characteristics of the obtained data. Second, it constructs a cluster network considering the location of things and their energy consumption. Finally, the things in a cluster communicate with other things in a different cluster based on their properties. In the experiment, the proposed method exhibits a better performance than the existing method. Owing to the energy-saving effect, we demonstrate that the proposed method results in a more reliable network and improves the longevity of IoT networks.

**Keywords:** thing property; clustering; data exchange; energy efficiency; IoT

---

## 1. Introduction

A wireless sensor network (WSN) is formed with many sensor nodes and deployed in our surroundings life to support smarter human life. The sensor node is composed of a small microprocessor, specific sensor devices, wireless communication modules, and small storage capacity. The sensor nodes achieve the information from their surrounding environment and transmit it into the sink node. Based on the collected data, the sink node recognizes the current state of location the WSNs deployed and can send the information to the external network to perform the advanced analysis [1,2].

Recently, with the extension of the WSNs, the concept of the Internet of things (IoT) has been proposed, and various application services have been developed in IoT networks. In the WSNs, the sensor nodes collect data through their sensing devices and transfer them to the sink node. The analysis of collected data is performed in the sink node or outside servers. However, in IoT networks, the scope of sensor nodes is extended to the things pervasively scattered. The things function as data collectors in addition to a sink node or base station. Furthermore, each thing attempts to share information by exchanging data with other things. Moreover, the state and action of things are decided by themselves. An information-sharing approach provides an abundant development environment for IoT services.

Sensor nodes have limited power, narrow network bandwidth, and low computing power. Moreover, since many sensor nodes are involved in configuring a network, their maintenance is difficult. Accordingly, efficient energy consumption is one of the rising research areas. In particular,

routing algorithms for reducing network traffic have been researched as an important issue for efficient energy consumption [3–5].

In the WSNs, each sensor node collects data from its surrounding area and transfers them to the sink node persistently. Accordingly, the purpose of routing algorithms is to transfer data to the sink node efficiently. However, in the IoT environment, various things not only communicate with each other, but also perform cooperative interactions to achieve the desired service or obtain advanced information. Therefore, instead of the transmission of data to the sink node, network configuration for data exchange between things is required, while maintaining efficient energy consumption.

The things deployed in various places are equipped with sensing devices to obtain information from the surrounding environment. Each sensor device measures data for a specific purpose. These data can be used to represent specific properties, such as distance, weight, vibration, sound, and illumination. As the properties required to increase, various sensor devices emerge. In the IoT environment, to obtain information for a specific property, various kinds of sensor devices are tied together, and the specific property can be obtained.

Since things are operated with limited power resources, efficient energy management is required to prolong the lifetime of IoT networks. Thus, if a network could be formed by assigning properties to the things, energy efficient data transfer is possible. When a thing interacts with other things to obtain the desired information, if property-based networks are applied, the network traffic can be reduced greatly. The result is that only specific things with the desired property have participated in the process of data exchange.

In this paper, we proposed a cluster configuration and data exchange method for IoT network environments. We configured the IoT network environment with the properties of things involved in the clusters. The proposed methods can solve the problems occurring when a hierarchical routing algorithm used in WSNs is applied to the IoT environment. In the proposed method, after a cluster is configured with local things, a cluster header is chosen, and it broadcasts the properties of member things to the surrounding cluster headers. Furthermore, when data exchange between clusters is required, since the property of the requested data is utilized, only the clusters relevant to the property participate in the data exchange process. In the experiment, the proposed method exhibits a better performance than the existing method. In particular, in terms of the average energy consumption rate, the proposed method exhibits a saving effect of more than 10 times. Through the experiments, we demonstrate that the proposed method contributes to a more reliable network and improves the longevity of IoT networks.

The remainder of this paper is organized as follows. Section 2 describes the related works based on WSNs, routing algorithms, and the IoT environment for efficient data transfer. Section 3 describes the proposed methods to construct clusters of local things and to exchange data on the basis of the cluster property in IoT network environments. Section 4 presents the performance evaluations of the proposed method. Section 5 concludes the paper and suggests future works.

## 2. Related Work

### 2.1. Wireless Sensor Networks

A WSN is configured with small sensor nodes fabricated with low specification CPU, memory, a low-speed wireless communication module, and low-capacity battery. They are deployed in buildings, roads, mountains, and various environments and collect information from their surroundings. The collected data are transmitted to a specific destination via the sink node. According to the types of measured data, they are called a biosensor node, chemical sensor node, magnetic sensor node, light sensor node, motion sensor node, etc. The WSNs installed with various sensor nodes can be utilized for monitoring weather and temperature-humidity changes, surveillance of wild animals, control of vehicular traffic, etc. [6,7].

The sensor nodes are classified into a sink node and a general-purpose sensor node. A general-purpose sensor node obtains information from its surrounding environment and transmits it to the sink node. The sink node performs the role of a gateway to not only send the received information to

an external network, but also transmit external requests to the corresponding sensor nodes. Since the deployed sensor nodes have limited power, memory, and other low-level resources, many methods for effective data transfer have been researched. One of them is an ad-hoc network that configures the network quickly, preserves low data transfer rates, and can support multiple sensor nodes in a loosely coupled style. In order to operate wireless sensor nodes effectively under limited power resources, TinyOS is proposed, which chooses an event-driven operation and state transition machine method [8]. Furthermore, SOS [9] provides a runtime upgrade function for applications areas and kernel areas. MANTIS [10] has been proposed on the microscopic threads. NanoQplus [11] has been studied as the operating system for WSNs. Furthermore, various operating systems for WSNs have been researched [12,13].

## 2.2. Internet of Things and Edge Computing

The IoT is based on the existing WSNs. Recently, various IT high technologies have been applied to smarten things and to configure advanced networks among things. Particularly, the IoT employs networking technologies to support intelligent, cooperative interaction among smart things. As a result, many applications services operating on WSNs are moving to the IoT domain and moreover are attempting to provide more intelligent functions.

As the IoT network grows, the number of things in IoT increases significantly. Consequently, the amount of information produced by things increases highly. In IoT, each thing should co-share the desired information with other things or collect information actively. These intentions are represented as Brillo [14] to support various devices, HomeKit [15] for smart homes, and MinT [16] to support the integration of heterogeneous devices and provide the tools to develop IoT service applications. Furthermore, for effective cooperation between things, research on the middleware in IoT and interconnection of things and the information management of things has been undertaken [17,18].

The implementation of IoT is a challenging task, and all the implementation scenarios require the use of different technologies and the emergence of new ones, such as Edge Computing (EC). Over the last few years, there has been a great increase solution that decentralize communications, data collection and processing, moving all those tasks to the edge. Edge computing directs computational data, applications, and services away from cloud servers to the edge of a network. The content providers and application developers can use the edge computing systems by offering the users services closer to them. Edge Computing allows for improving the performance of computer systems by lowering latency, reducing the cost of resources and increasing responsiveness, scalability, reliability, security or privacy [19,20].

The management of IoT networks is challenging because of the heterogeneity of its resources, creating difficulties in communication protocols, real-time processes, data management, big data storage, security or privacy. In this regard, Edge Computing architectures offer a solution to IoT infrastructures because they can manage the heterogeneous data generated by IoT devices [21].

## 2.3. Routing Protocols

As IoT succeeds WSNs, things in IoT are similar to the sensor nodes in WSNs. Since sensor nodes are equipped with limited resources, effective operation methods have been researched to support the longevity of IoT networks. The highest proportion of energy in sensor nodes is consumed by the wireless communication module. Therefore, to reduce the energy consumed by the communication module, studies on routing protocols in WSNs have been conducted. According to the network architecture, the routing protocols are classified into planar structure and tree hierarchical structure.

In a planar structure, all the sensor nodes perform the same role. According to the time of formation of routing paths, a table-driven method or on-demand method can be employed. The table-driven method calculates the routing paths before the data transmissions are requested. On the other hand, the on-demand method configures the routing paths when the data transmissions are issued. When compared to the hierarchical tree structure, the planar structure requires a larger routing table and has less scalability.

In a hierarchical tree structure, the sensor nodes are divided into clusters composed of a header node and member nodes. It has a group-based architecture. When member nodes in clusters require data transmissions, communication is undertaken via the header node. Hence, the size of the routing table decreases [5].

If the planar structure method is applied to the IoT environment, since each thing has a routing table to support all the things, a significant amount of storage resources is exhausted. Moreover, even if the tree structure method can manage the routing tables effectively, it requires the participation of all the things for data collection. This approach causes an increase in the network traffic, and thus, the energy consumed by the things increases significantly, the network bandwidth is wasted, and the unnecessary time delay is inevitable.

### 2.3.1. Data Collection in Tree Structure

Data collection in a tree structure is the basic approach among hierarchical data collection methods. In a tree structure, several sensor nodes become the parent nodes and collect data from the child nodes, which are deployed in the surrounding working areas. The parent nodes play the role of a bridge such that the data received from other parent nodes are transmitted to the sink node. In order to designate some specific sensor nodes as the parent node, various metrics, such as the geometric location [22], remaining resources of sensor nodes [23], measured data types [24], and cost of data collection [25] have been employed.

As mentioned above, for data collection with a tree structure, each sensor node is classified into a parent or child node, but the parent node does not manage the child nodes. Furthermore, since data are always requested from the sink node in WSNs, the sink node must be the center of the tree structure. During the sharing of information in the IoT environment, many things can request information from other things frequently. Consequently, the number of tree structures increases significantly. Since all tree structures incur the burden of formation and maintenance, it is difficult to apply the tree structure method to IoT networks.

### 2.3.2. Data Collection in Cluster Structure

Data collection in a cluster structure is similar to that of the tree architecture. Specific nodes collect data and transfer them to the sink node. However, in the cluster structure, a network is configured with multiple clusters. A cluster is composed of a specific node and several surrounding nodes. These clusters form sub-networks among themselves. A specific node in a cluster is designated as the header node, and the surrounding nodes become member nodes in the cluster. The header node receives data from its member nodes and integrates them and transmits to the sink node. The member nodes of a cluster can communicate with only their header node and transmit their collected data to the header node [26,27].

The clustering algorithm is composed of two steps: Construction of the cluster and data collection. In a cluster structure, to reduce the energy consumption of the header node, a cluster is reconstructed within a certain period to elect a new header node. As the election of the header node is performed repeatedly, the fairness of energy consumption in a cluster can be sustained. Various metrics have been applied for the election of the header node, such as random, remained energy, priority, etc. [28,29].

For collecting data in a cluster structure, the existing researches focus on both cluster construction and header node election. After constructing the clusters, the sink node receives data from the clusters according to a method by which all the member nodes in a cluster transmit the collected data to the header and each header re-transmits them to the sink node.

WSNs have intrinsic characteristics in the flow of collected data. Since all data is unilaterally collected at the sink node, the traffic in the network flows in a fixed direction. For this reason, the installation of a routing path to reach the sink node is an important issue in both tree structure and cluster structure. However, in the IoT and EC environment, the working environment differs from that of WSNs. In IoT and EC, various things perform the role of a sink node because each thing can attempt to collect the data saved in other things to share information. This can occur in all the things

deployed in IoT and EC networks. Therefore, if the data collection methods used in WSNs are applied to the IoT environment, all the things have to store a routing table for all the other things and maintain the information of all the things to communicate with each other. These requirements induce significant overhead in the efficient management of IoT networks [30].

### 3. Proposed Property-Based Data Transfer

#### 3.1. Motivation and Strategies

In the WSNs, since routing algorithms are employed to transfer the data from the sensor nodes to the central server, each sensor node has to sustain only the routing path to the server. However, in the IoT environment, various things can directly interconnect with other things without a server. They can request data exchanges with other things. Hence, in addition to the central server, routing paths connected to various things are created and maintained in the working space of each thing. These working conditions require significant energy and storage resources. These requirements cause a heavy burden on things with limited resources. Therefore, we require a new strategy to solve these problems.

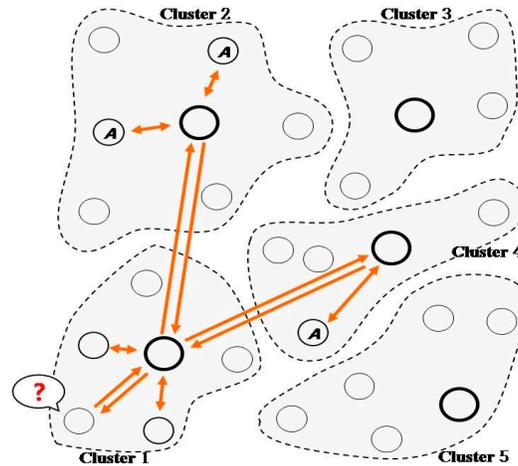
In the IoT environment, the data measured and produced by the things are classified according to the type of use. For example, if the distance between things is measured, the data measured by the ultrasonic sensor, infrared sensor, received signal strength indicator (RSSI) in Bluetooth, and other sources can be utilized. However, these data can be used to infer a specific property, i.e., "distance".

To share information, when data exchange between things is necessary, the properties of the things can be utilized. If only the things related to the specific property participate in the data exchange process, instead of involving all the things, the network traffic can be reduced significantly. Hence, energy consumption in the IoT network also decreases significantly. Therefore, if a group is formed with the things located close together and a leader node, and if the leader of each group sustains the property information of the other groups, a thing can exploit the property information of all the groups with a small amount of resources. The group is called a property-based cluster (PC), which is composed of a header node and several member nodes.

#### 3.2. System Architecture

In this paper, the cluster configuration method and data transfer method are proposed for efficient information-sharing in IoT networks. The proposed method is based on the cluster property of data transfer. It is abbreviated as CPDT (cluster property of data transfer). In the CPDT method, each thing has a property that represents its characteristic. A group of things becomes a cluster. The header node of a cluster has the property information of the member nodes. When a request for specific data arises, the header node queries only the things with the property corresponding to the specific data. Since only the relevant things participate in the process of querying, the network traffic and energy consumption in IoT networks are significantly reduced.

Figure 1 shows an example of a cluster configuration in the IoT environment with five clusters. The clusters are composed of several things with various functions, such as distance measurement, detection of human appearance, and monitoring of surrounding air state. Each cluster is based on the surrounding local things. As shown in this figure, several kinds of things are involved. The thick circular node represents the header of a cluster. The sensor node for monitoring the surrounding air state is titled as A.



**Figure 1.** Cluster configuration with properties of things (○: Cluster header, A: Air state monitoring node, ?: Query node, →: Message).

Owing to the characteristics of the cluster structure, the member nodes involved in a cluster can communicate with each other via only the elected header. The header configures the cluster and broadcasts the property information of the member nodes to other cluster headers. Moreover, it receives messages from the headers of other clusters. These messages include information on things in other clusters. Consequently, a header gathers the property information possessed in other clusters. Based on this knowledge, when data exchange is required for information-sharing, the header queries only the relevant cluster headers regarding the corresponding information. This information includes the property of things. Through a query to selective things, the network traffic decreases and hence, the energy consumption of the things also reduces.

For example, in Figure 1, when a member thing with question(?)mark in the cluster 1 requests the air condition state, it sends a query message to the header node. Subsequently, the header node of cluster 1 transfers the query message to only the corresponding cluster headers that include member nodes capable of obtaining information about the air condition state. In the previous cluster configuring stage, each header node becomes aware of the properties of the member nodes that can obtain the property information about the air condition state. In Figure 1, clusters 1, 2, and 4 include the things possessing information about the air condition state. The corresponding things are marked as A.

Owing to the selective query message, the cluster headers irrelevant to the air state information do not receive the query message. However, the cluster headers relevant to the air state information receive the query message. The headers send it to their member nodes that can obtain the air state information. In Figure 1, the headers of cluster 2 and cluster 4 are relevant. After receiving the messages from the header, the member nodes of clusters 2 and 4 obtain the information on the air condition state and send it to their header node. The header nodes of clusters 2 and 4, in turn, send the information of air condition state to the header node of cluster 1. This header node sends it to its member node that issued the query of air condition state initially.

As the data exchange between the cluster headers is processed according to these selective query messages, the total network traffic can be reduced significantly.

### 3.3. System Operation Process

The major functions of CPDT are effective cluster configuration based on the remaining energy of the things and data interchange of the properties assigned to the things. In order to enhance the lifetime of the IoT network, the cluster header is changed periodically. This period is named as a round. The operation of CPDT is composed of three phases: “Advertisement phase”, “setup cluster phase”, and “data exchange phase.”

### 3.3.1. Advertisement Phase

In the “advertisement phase”, each thing(node) broadcasts its remaining energy and participates in the header election. Figure 2 shows the sequence diagram of “advertisement phase” with four nodes. In Figure 1, it is regarded that node 1 has the most remaining energy. The steps of this phase are described as follows.

1) When the advertisement phase starts, all the nodes are initialized to the general node, but not the header node or member node. Each node broadcasts information of its remaining energy and receives the information of remaining energies from the surrounding nodes.

2) After exchanging the information of remaining energies, each node attempts to identify the node with the most remaining energy from the surrounding nodes. If the node itself has the most remaining energy, it assumes the role of a header node. However, if there are other nodes with more remaining energy than its own, it becomes a member node and is ready to participate in the other clusters. The node elected as the header broadcasts a message with its ID. This node not only becomes the header of a cluster, but also is ready to recruit the member nodes of the cluster.

3) The member nodes save the “advertisement” messages received from the surrounding header nodes and are ready to choose their cluster.

In Figure 2, node 1 becomes the header node because it has the most remaining energy. As the header node, node 1 broadcasts its information to the surrounding nodes. The other nodes become the member nodes and are included in the cluster. The sequence diagram in Figure 2 shows the process of configuration of a cluster.

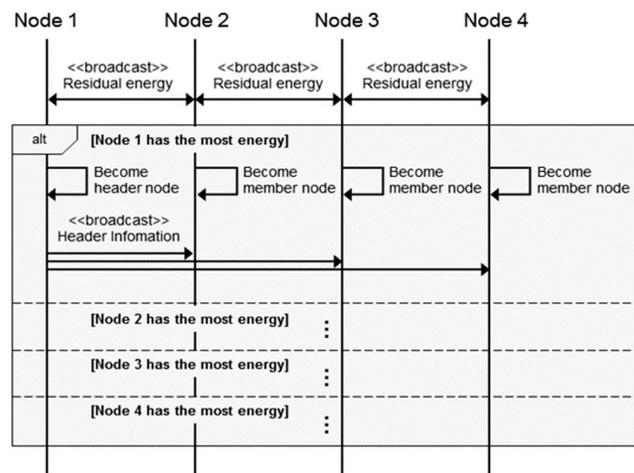


Figure 2. Advertisement phase sequence diagram.

### 3.3.2. Setup-Cluster Phase

In the “setup cluster phase”, the member nodes choose the most adjacent header node based on their locations. The member nodes utilize the ID information of cluster headers received in the advertisement phase. Finally, each member node involves in the cluster of header node they chose. Subsequently, the property information of the things is exchanged to support the cluster-property-based data transfer. If the memory of the things is sufficiently large, the information of all the things deployed in the IoT network can be stored successfully. However, generally, since the things in IoT have limited computing resources, they cannot store the information of all the things. Therefore, in the CPDT, each header node stores the property information of its member nodes. Furthermore, the header node also stores the information of properties retained by the other clusters to support the cluster-property-based data transfer.

In this study, to choose the most adjacent cluster header, RSSI is exploited. The RSSI indicates the strength of the radio signal received in the communication module. The strength of RSSI is stronger as the distance between things is lesser, and vice versa. Based on this characteristic of RSSI, the distance between things is calculated relatively [31].

Figure 3 shows the sequence diagram of the setup cluster phase. The following are the steps of this phase.

1) Among header nodes, if the header node of cluster A is the closest header, the member node receives the strongest RSSI value from this header node. The member node transmits a message to the header of cluster A to participate in this cluster. This message includes the property of the member node.

2) The header node stores the property information of participating member nodes and allocates slot numbers. The slot number indicates the order in which data is collected from the member nodes by the header node. The slot number is also transmitted to the member node. Otherwise, if the member node is closest to cluster B, the strongest RSSI value is detected from the header of cluster B. This node participates in cluster B, as shown in Figure 3.

3) When all the corresponding member nodes are involved in a cluster, the configuration of a cluster is complete. Subsequently, the cluster header broadcasts the property information of the member nodes. The other cluster headers receive the broadcasted message and store the property information included in the message. Based on the stored property information, when the data interchanges between clusters are executed, only the clusters with the necessary property information are chosen, and the issued queries are sent to them.

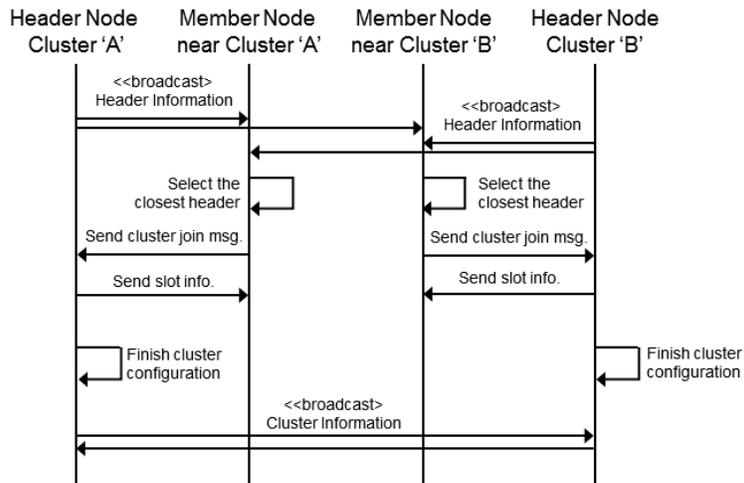


Figure 3. Advertisement phase sequence diagram setup-cluster phase sequence diagram.

### 3.3.3. Data-Exchange Phase

This phase performs the data exchange between things in different clusters. In a cluster, any nodes can request the information they do not possess. In “data exchange phase”, to obtain the desired information, the data exchange occurs directly between clusters. After this phase is completed, the data exchange between nodes is suspended temporarily. All the nodes begin to undergo the advertisement phase to configure new clusters.

Figure 4 shows that a member node of cluster A requests the information of “weight” property, which is produced in the member node of cluster B. This request is handled according to the following steps.

1) A member of cluster A transmits a query to the header to obtain the weight property information.

2) The header of cluster A receives the query from the member, and sends the query to the headers of other clusters that include the member nodes capable of obtaining the weight information. At this time, since the header of cluster A utilizes the stored information of weight property, the query is sent to only the headers of clusters relevant to this property.

3) The header of cluster B sends a query to the member node relevant to the weight property. The header requires weight information.

- 4) The member node receiving the query obtains the weight information through measurement, calculation, etc. Finally, this member node transmits the weight information to the header node according to its own slot time in data collection, which is assigned in the setup cluster phase.
- 5) The header of cluster B sends the weight information to the header of cluster A.
- 6) The header of cluster A transmits the weight property information to the corresponding member node that issued the query originally.

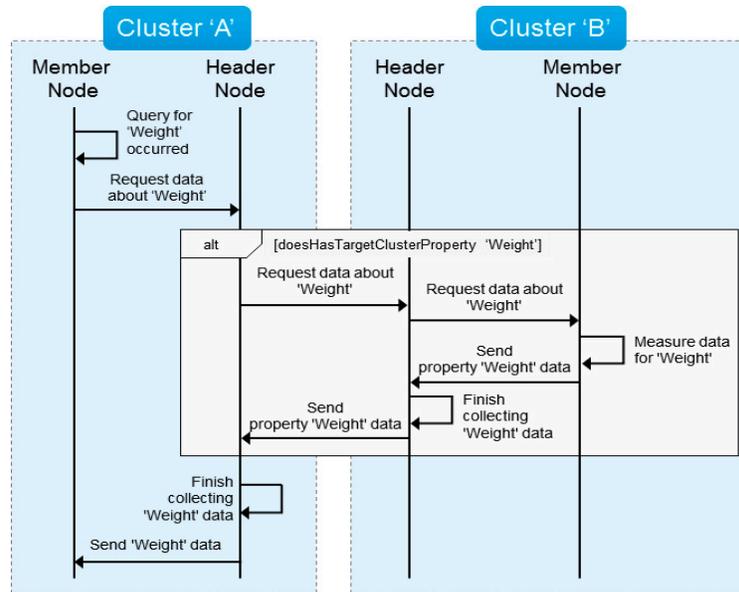


Figure 4. Data-exchange phase sequence diagram.

## 4. Performance Evaluation

### 4.1. Experiment Environment

In order to evaluate the proposed method in IoT networks, we measure the lifetime of the network and average energy consumption rates. To evaluate the performance of the proposed system, we have used MinT [16] to configure the simulation nodes for IoT device and customized the TOSSIM [32] to simulate the IoT environment. The MinT is middleware to configure the IoT device easily and quickly and also used for in-loop simulation. The TOSSIM is a simulator of TinyOS to simulate the WSN environment. We called the new simulator MinTOSSIM. The performance of the proposed method is measured using the MinTOSSIM. We configure the testbed with a size of 100 m × 100 m and deploy the nodes randomly. As the number of nodes increases to 50, 70, and 100, we measure the variation of network traffic. All the nodes deployed in the network have data exchanging opportunities of 100, 150, or 200 times per round. As the probability of 0.3 per one round, each node issues a query to an arbitrary property. Figure 5 shows the deployment map of our IoT networks, which are comprised of 100 randomly deployed sensor nodes.

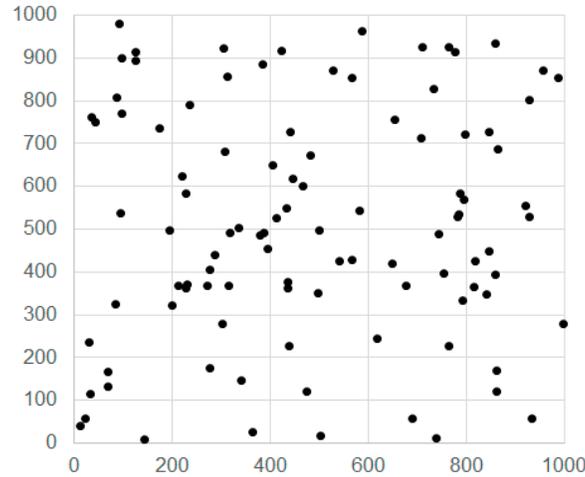


Figure 5. Deployment map with 100 things (nodes).

We use the first-order radio model as the communication style [32]. Figure 6 shows the structure of the chosen radio model. After a data packet of  $k$  bits in the sender is sent through the transmitter and the signal is amplified, it can be transmitted as a radio signal. The receiving node receives the data as a radio signal and decodes them into data packets. In this model, the energy for data transmit-receive operations was proposed according to the following Equations (1), (2). In these equations,  $k$  is data size(bit),  $d$  is distance(meter),  $E_{amp}$  is the power consumption in the signal amplifier (5 pJ), and  $E_{elec}$  is the power consumption in the radio module (1 nJ).

$$E_{T_x}(k,d) = E_{elec} \times k + E_{amp} \times k \times d^2 \tag{1}$$

$$E_{R_x}(k,d) = E_{elec} \times k \tag{2}$$

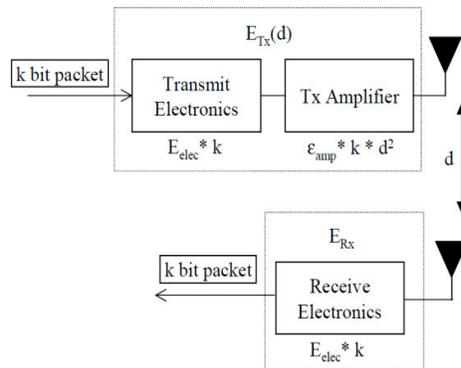


Figure 6. First order radio model.

#### 4.2. Round of the Last Node

In order to compare the performances of data transfer methods, we measure the dead round of the last node. The last node indicates the node that is worked on the cluster till the last. A round means the changing period of the cluster header in a cluster. Figure 7 shows the dead round of the last node in two data transfer methods. One method is CPDT we proposed. The other is non-CPDT which is an existing cluster structure method that does not use the property of things. As shown in this figure, the dead round of CPDT is 238 rounds with 100 deployed nodes. Otherwise, the non-CPDT performs the dead round at only 15 rounds.

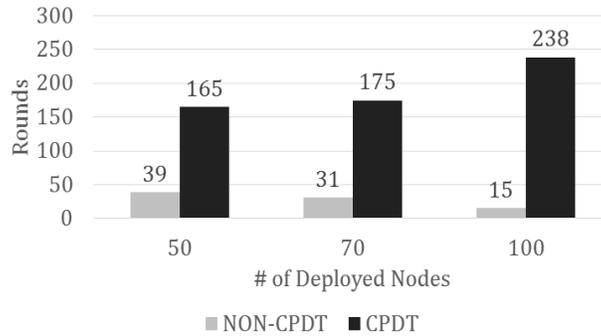


Figure 7. Dead round of the last node.

In particular, in the non-CPDT, as the number of deployed nodes increases, the number of nodes participating in the communication also increases. Consequently, as the dead round of the last node decreases, it is inevitable that the lifetime of the IoT network shortens. However, using the CPDT method, even if the number of nodes increases, the number of participating nodes in the query process does not increase proportionally. On the contrary, as the number of nodes increases, the CPDT prolongs the lifetime of the IoT networks because it significantly reduces the network traffic in IoT networks. In addition, as the number of nodes increases, the probability of each member being elected as a header node decreases. Hence, the energy consumed by the header node can be balanced, and thus, the CPDT method can contribute to the longevity of IoT networks.

#### 4.3. Dead Node Rates

In order to evaluate the efficiency of network operation, the dead node rates are measured according to the course of time. Figure 8 shows the average values of dead node rates when the number of nodes is 50, 70, and 100 and the number of queries exchanged in a round is 100, 150, and 200, respectively. In the existing data transfer method, as the operating time progresses, the number of packet interchanges to handle a query increases excessively. Hence, the energy consumption of nodes also increases significantly. Consequently, the dead node rates increase abruptly. From the figure, we can observe that the nodes over 90% are dead in 16 rounds in non-CPDT. However, the CPDT method requires 150 rounds to obtain the death rate of 90% because only the nodes relevant to the property information participate in the data exchange. Thus, the CPDT method can improve the reliability of IoT networks.

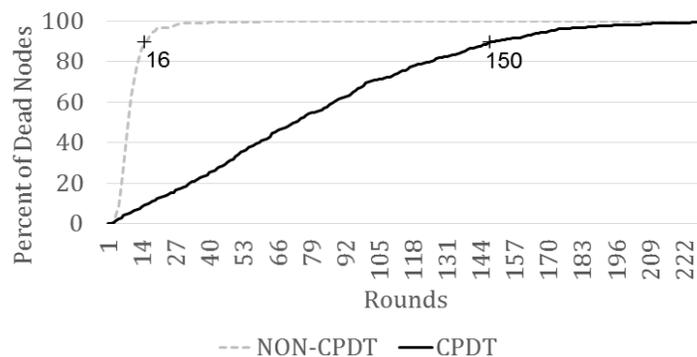


Figure 8. Percent of dead node per round.

#### 4.4. Average Energy Consumption Rate Per Round

Figure 9 shows the average energy consumption rates per round according to the number of nodes. In the two methods, as the number of nodes increases, the energy consumption rates also

increase. As the number of nodes increases, the number of clusters increases. Accordingly, the probability of data interchanges between clusters is also increased. However, as shown in this figure, the non-CPDT method has an energy consumption rate more than 10 times faster than the CPDT method because all nodes in clusters participate in the data interchange. However, in the CPDT method, the data transfers are executed with only the nodes relevant to the property information for queries. Consequently, unnecessary network traffic is eliminated, and thus, the energy consumption rate is low.

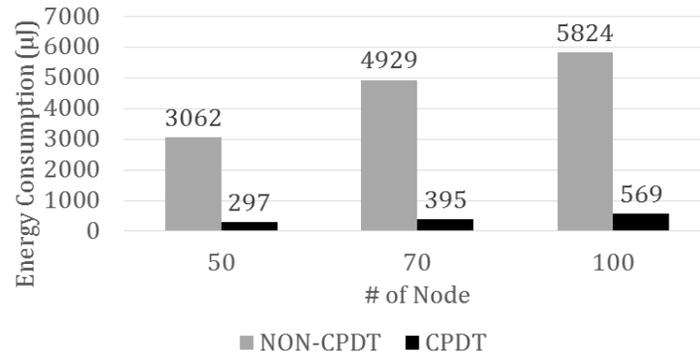


Figure 9. Average energy consumption per round.

## 5. Conclusion and Future Works

In the IoT environment, various things can communicate with other things for information-sharing. In this mechanism, many service applications have been developed and launched to support smarter human life. In WSNs, a specific sink node is used to transfer the data collected from leaf nodes. In order to avoid fast energy consumption at the sink node, many clustering methods have been proposed to achieve efficient energy consumption in WSNs. However, in contrast to WSNs, all the things deployed in IoT networks can perform the role of a sink node to collect data. Therefore, the existing clustering methods for WSNs have limitations for IoT service applications. If these existing methods are applied to IoT applications, they generate severe network traffic in the process of data collection. This results in inefficient energy consumption by the things and shortens the lifetime of IoT networks.

In this paper, to enhance energy efficiency in IoT networks, after assigning properties to each thing, a cluster-property-based data transfer method was proposed for the data exchange processes. The proposed method achieves effective data exchange to handle the issued queries because it forms energy efficient clusters from the property of things. In the experiments, to measure the performance of the data transfer methods, the dead round of the last node, dead node rates, and average energy consumption in IoT networks were chosen as the metrics. The proposed method, named CPDT showed vastly superior performances in terms of all the metrics as compared to the existing method. In particular, in terms of the average energy consumption rate per round, the CPDT method achieved the energy-saving effectiveness of more than 10 times as compared to the existing method because the CPDT only executes the things relevant to the property in the queries, instead of utilizing the participation of all the things. From the experiments, we demonstrated that the CPDT worked on the IoT networks as the more reliable state. Moreover, it contributes to the longevity of IoT networks.

For more reliable data exchanges, the data exchange protocol will be operated in the course of the advertisement phase and setup cluster phase proposed in the CPDT. Especially, when urgent information transmission and speedy operation are required in IoT networks, the time delay will induce a serious impact on the IoT service applications. In our future work, we will study an advanced data exchanging protocol to support these issues, which will operate on the CoAP (constrained application protocol) in the IoT environment.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Yick, J.; Biswanath M.; Dipak, G. Wireless sensor network survey. *Comput. Netw.* **2018**, *52*, 2292–2330.
2. Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: a survey. *Comput. Netw.* **2002**, *38*, 393–422.
3. Xu, R.; Donald, W. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678.
4. Pantazis, N. A.; Stefanos, A. N.; Dimitrios, D. V. Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Commun. Surveys Tutor.* **2013**, *15*, 551–591.
5. Liu, X. A survey on clustering routing protocols in wireless sensor networks. *Sensors* **2012**, *12*, 11113–11153.
6. Al-Karaki, J. N.; Kamal, A. E. Routing techniques in wireless sensor networks: a survey. *IEEE Wirel. Commun.* **2004**, *11*, 6–28.
7. Youngtae, J.; Jinsup, C.; Inbum J. Traffic Information Acquisition System with Ultrasonic Sensors in Wireless Sensor Networks. *Int. J. Distrib. Sensor Netw.* **2014**, *10*, 961073.
8. Levis, P.; Madden, S.; Polastre, J.; Szewczyk, R.; Whitehouse, K.; Woo, A.; Gay, D.; Hill, J.; Welsh, M.; Brewer, E.; Culler, D. TinyOS: An operating system for sensor networks. *Ambient Intell.* **2005**, *35*, 115–148.
9. Han, C.C.; Kumar, R.; Shea, R.; Kohler, E.; Srivastava, M. A dynamic operating system for sensor nodes. In Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, WA, USA, 6–8 June 2005; 163–176.
10. Bhatti, S.; Carlson, J.; Dai, H.; Deng, J.; Rose, J.; Sheth, A.; Shucker, B.; Gruenwald, C.; Torgerson, A.; Han, R. MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Netw. Appl.* **2005**, *10*, 563–579.
11. Park, S.; Kim, J.W.; Shin, K.Y.; Kim, D. February. A nano operating system for wireless sensor networks. In Proceedings of 2006 8th International Conference Advanced Communication Technology, Pyongchang, Korea, 20–22 February 2006; 4.
12. SeungMin, P. Technical Trend of Sensor Network Node Platform & OS. *Electron. Telecommun. Trends* **2006**, *21*, 14–24, doi:10.22648/ETRI.2006.J.210102.
13. SeoKu, K. Sensor Network Research and Development Commercialization Instance. *Tech. Trends Every Weeks* **2007**, *1325*, 1–14.
14. Android Things. Available Online: <https://developer.android.com/things/hardware/index.html> (accessed on 9 January 2020).
15. Apple Homekit. Available Online: <https://developer.apple.com/homekit/> (accessed on 9 January 2020).
16. Jeon, S.; Jung, I. MinT: Middleware for cooperative interaction of things. *Sensors* **2016**, *17*, 1452.
17. Chongmyung, P.; Youngtae J.; Inbum J. Cooperative Processing Model for Wireless Sensor Networks. *Int. J. Distrib. Sensor Netw.* **2013**, *2013*, 317214.
18. Whitmore, A.; Agarwal, A.; Da Xu, L. The Internet of Things—A survey of topics and trends. *Inf. Syst. Front.* **2015**, *17*, 261–274.
19. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE IoT J.* **2016**, *3*, 637–646.
20. Balasubramanian, V.; Kouvelas, N.; Chandra, K.; Prasad, R.V.; Voyiatzis, A.G.; Liu, W. A unified architecture for integrating energy harvesting iot devices with the mobile edge cloud. In Proceedings of 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 13–18 February 2018.
21. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE IoT J.* **2017**, *4*, 1125–1142.
22. Solis, I.; Obraczka, K. Isolines: Energy-efficient mapping in sensor networks. In Proceedings of 10th IEEE Symposium on Computers and Communications (ISCC'05), Murcia, Spain, 27–30 June 2005.
23. Erramilli, V.; Malta, I.; Bestavros, A. On the interaction between data aggregation and topology control in wireless sensor networks. In Proceedings of 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, USA, 4–7 October 2004.
24. Ding, M.; Cheng, X.; Xue, G. Aggregation tree construction in sensor networks. In Proceedings of 2003 IEEE 58th Vehicular Technology Conference, Orlando, USA, 6–9 October 2003.
25. Luo, H.; Luo, J.; Liu, Y. Energy efficient routing with adaptive data fusion in sensor networks. In Proceedings of the 2005 joint workshop on Foundations of mobile computing, Cologne, Germany, 2 September 2005.

26. Wang, P.; Yao, C.; Zheng, Z.; Sun, G.; Song, L. Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems. *IEEE IoT J.* **2018**, *6*, 2872–2884.
27. Ren, J.; He, Y.; Huang, G.; Yu, G.; Cai, Y.; Zhang, Z. An edge-computing based architecture for mobile augmented reality. *IEEE Netw.* **2019**, *33*, 162–169.
28. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000.
29. Younis, O.; Sonia F. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans. Mobile Comput.* **2004**, *3*, 366–379.
30. Khan, W.Z.; Ahmed, E.; Hakak, S.; Yaqoob, I.; Ahmed, A. Edge computing: A survey. *Future Generat. Comput. Syst.* **2019**, *97*, 219–235.
31. Benkic, K.; Malajner, M.; Planinsic, P.; Cucej, Z. Using RSSI value for distance estimation in wireless sensor networks based on ZigBee. In Proceedings of 2008 15th International Conference on Systems, Signals and Image Processing, Bratislava, Slovak Republic, 25–28 June 2008.
32. Levis, P.; Lee, N.; Welsh, M.; Culler, D. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In Proceedings of the 1st international conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).