

Research Article

Agent-Based Multipath Management for Supporting Sink Mobility in Wireless Sensor Networks

Cheonyong Kim ¹, Hyunchoong Cho,² Kwansoo Jung,³ Yongbin Yim,⁴ Taehun Yang,⁵ Sang-Ha Kim,⁵ and Sangdae Kim ⁶

¹Advanced KREONET Center, Korea Institute of Science and Technology Information (KISTI), Daejeon 34141, Republic of Korea

²Research Institute for Computer and Information Communication, Chungbuk National University, Cheongju 28644, Republic of Korea

³Department of Fintech, Daejeon University, Daejeon 34520, Republic of Korea

⁴Network Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea

⁵Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea

⁶Division of Computer Science and Engineering, Kongju National University, Cheonan Chungnam 31080, Republic of Korea

Correspondence should be addressed to Sangdae Kim; sdkim.cse@gmail.com

Received 9 June 2020; Revised 6 July 2020; Accepted 1 August 2020; Published 5 September 2020

Academic Editor: Euisin Lee

Copyright © 2020 Cheonyong Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks, sink mobility support is one of the essential functionalities in many applications. With continuous advancement, future applications will require not only sink mobility support but also high-performance data delivery service. Multipath routing is one of the promising technologies for improving data delivery performance by collaboratively using alternative or redundant multiple routing paths. However, existing multipath routing protocols had not dealt with sink mobility. As a result, they lead to bad performance in terms of energy efficiency due to the end-to-end path reconstruction. Consequently, a novel multipath management scheme is required thereby supporting sink mobility without performance degradation. In this paper, we propose a multipath management scheme for supporting sink mobility. The proposed scheme dynamically constructs multipath along the moving path of a sink. In addition, the proposed scheme provides the path shortening schemes according to the sink's movement for reducing energy consumption. Our simulation results show that the proposed scheme is superior to existing path management schemes in terms of reliability and energy efficiency.

1. Introduction

Sink mobility is a fundamental requirement in wireless sensor networks (WSN) because mobile sinks appear frequently in many applications as data collectors, administrators, and users [1, 2]. For example, soldiers in military surveillance, rangers in forest management, and biologists in habitat monitoring are representative mobile sinks in WSN-based data collection systems. The mobile sinks execute their duties with the aid of the adjacent sensor data. In addition, automatic mobile sinks (e.g., unmanned aerial vehicles (UAVs) and mobile robots) can perform challenging tasks in inhospitable areas without

reliable infrastructure. The effective sensor data collection services that can be provided by sink mobility are promoting the practical development of wireless sensor networks.

Recently, various techniques for WSN have actively been researched because WSN is receiving attention as an infrastructure for the Internet of Things [3, 4]. In this situation, the new applications will naturally be developed and these applications will surely require more powerful network performance than existing applications. For example, in the habitat monitoring application, the biologist may want to collect not only the location of the target animals but also their statuses such as temperature, heartbeat, and posture. In this case, the network should

provide enhanced data throughput in real time. To realize the applications, the sensor network has to be evolved toward high-performance networks.

The multipath routing is one of the network techniques to provide high performance in terms of reliability and throughput of packet delivery [5]. In multipath routing, multiple disjointed paths are established between a source and a sink by path discovery and path construction processes. These disjointed paths improve the packet delivery success ratio by transmitting multiple copies of a packet through the paths. In the redundant packet transmission, the packet delivery fails if and only if all of the paths fail. In addition, the packet delivery throughput can be improved by delivering multiple packets concurrently. Theoretically, the improved data rate of the concurrent packet transmission is proportional to the number of paths. That is, discovering disjoint multipath enhances the packet delivery quality significantly.

Traditional multipath routing protocols [6, 7] assumed that there is a static sink at a fixed location. Therefore, to support sink mobility, the existing protocols need to reconstruct the multipath whenever the sink moves. Multipath construction includes several subprocesses such as searching available paths, selecting appropriate paths, and establishing selected paths. Since the processes require a lot of message exchanges among the sink and the participating nodes, existing multipath discovery schemes lead to a fatal problem. That is, a large amount of energy is needed for reconstructing multipath so the network lifetime would be considerably shortened. Moreover, the data delivery cannot be conducted while the multipath is reconstructed. Technically, the multipath is constructed between a source and an agent. The agent acts as a static sink during a mobile sink is within its radio range. That is, the agent is the last-hop node of each path. Therefore, the multipath has to be reconstructed whenever the sink moves outside the radio range of the agent. In this case, the multipath reconstruction interval decreases as the speed of the sink increases and as the radio range of nodes decreases. The frequent multipath reconstruction dissipates nodes' energy so that the network lifetime will be shortened without performance improvement. In addition, the connection between the source and the sink is disconnected during the multipath reconstruction process. If the reconstruction delay is larger than the reconstruction interval, the connection between the source and the sink will not be restored.

Meanwhile, in single-path routing, a few works have dealt with the path management scheme for supporting sink mobility. They maintain the connection between a source and a mobile sink without end-to-end path reconstruction by the agent control mechanism. For example, in the footprint chaining mechanism [8], a mobile sink makes a connection between the closest neighbor node and the agent. This path extension is continuously conducted and the routing path is constructed along the moving path of the sink. However, in multipath routing, they lead to performance degradation since their path management mechanisms result in constructing a single path along the moving path of a mobile sink.

In this paper, we propose a novel multipath management scheme, called MPM, for establishing end-to-end multiple routing paths during the sink moves. As mentioned above,

the existing studies lead to performance degradation because they deal with one agent. On the one hand, in the multipath routing schemes, the end-to-end multiple paths should be reconstructed when the link between the agent and the mobile sink is disconnected. The frequent path reconstruction leads to the excessive energy consumption of sensor nodes. On the other hand, in the single-path routing, a single path is constructed between the current location of a mobile sink and the initial agent. The single path management cannot provide high-performance data delivery. Intuitively, simply applying the single path management scheme to multiple paths seems to be a plausible solution for solving the problem. However, the independent management and disjointness maintenance are nontrivial challenge. As an innovative solution, MPM adopts the agent-based multipath management where an exclusive agent is assigned to each path. Figure 1 shows the radical difference between the existing approaches and MPM. For effective path management, we introduce two kinds of path management functions that are path shortening and path extending and provide different mechanisms for each function. In addition, we provide a discussion about the impact of the multipath reconstruction interval and neighbor list update interval which significantly influences the performance of the MPM.

The rest of this paper is organized as the following. Section 2 provides the background and related works of this paper. Section 3 describes MPM in detail including the data structure and path management algorithms. In Section 4, the results from our computer simulation are presented. Finally, Section 5 concludes this paper.

2. Related Work

In multipath routing, the node or link disjoint multipath is constructed between a source and a sink since the nondisjoint may lead to the negative impact in terms of reliability and aggregated bandwidth. The disjoint multipath improves the performance in terms of the packet delivery success ratio and the throughput [5]. First, the packet delivery success ratio increases by exploiting path redundancy. In this case, the probability of failure decreases as the number of multipath increases if multiple copies are redundantly forwarded through each path. This is because a packet delivery fails only if all paths fail. Second, the throughput increases by bandwidth aggregation. When using a single path, a source transmits one packet at a time. On the other hand, when multiple paths are available, the source can transmit multiple separate packets at a time. This shows the same effect as aggregating bandwidths of multiple paths. Alternative path routing [9] is another usage of multiple routing path where the secondary path is used if data delivery through the primary path fails. Due to the fact that each path should be independently used in multipath routing, establishing disjointed multipath is important for enhancing data throughput and packet delivery ratio.

Due to the great advantages, multipath routing has been an important research subject in WSN. In addition, recently, multipath routing is actively used for various application scenarios based on WSN. Mohanty and Kabat proposed a multipath data transmission scheme for healthcare

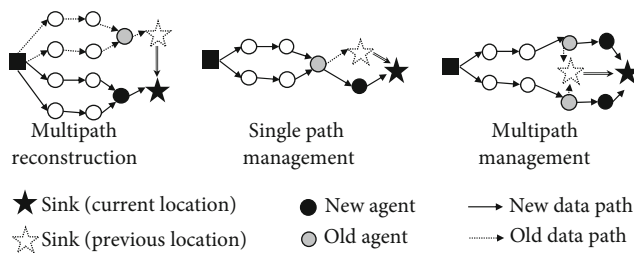


FIGURE 1: Path management classification.

application [10]. In this, the data packets are classified based on the criticality. The intermediate nodes are responsible for prioritized buffer management and congestion probability computation. In case of congestion, the emergency and sensitive data are transmitted through alternate paths. Hasan et al. proposed a multiconstrained multipath routing protocol for multimedia sensor networks [11]. Their mathematical model based on the Lagrangian relaxation adaptively controls the multipath routing protocol to balance the QoS parameters (i.e., the energy consumption and end-to-end delay constraints). Jaiswal and Anand [12] proposed the energy-efficient multipath routing protocol to improving QoS for IoT applications. Specifically, they focused on achieving high reliability in data packet delivery when the packets are unfairly generated in IoT environment. To this end, the sensors consider lifetime, traffic intensity, and reliability in the path construction phase so that the source can choose high-quality paths. However, the works focus on the performance enhancement and QoS provision in delivering data to a static sink. Therefore, frequent end-to-end multipath construction process should be conducted for supporting sink mobility.

The path management is the most important functionality for supporting sink mobility. The most fundamental management approach is reconstructing paths each time a sink moves, in other words, whenever the last-hop connection between the agent and the sink is lost. Primitive path construction schemes utilize the flooding method [13]. The mobile sink broadcasts its own location and sensor nodes store a neighbor who sends to the location update message as the downstream node while the location update message is disseminated in the network. When a sensor reports a data packet, it sends the packet through the downstream chain toward the sink. However, this scheme wastes lots of energy since the location update flooding is performed each time the sink moves.

The existing studies have focused on reducing flooding costs. Wang et al. exploited the local flooding approach [8]. In the study, a sink selects an agent among neighbors and then the agent floods its own location in the network. Then, the agent periodically performs flooding within the local area by a predefined range. The nodes outside of the local area could deliver data packets toward the agent, and nodes in the area send the data to the location that the sink currently exists. Luo et al. proposed the trajectory forwarding scheme for delivering data packets to the continuously moving mobile sinks [14]. In the trajectory forwarding, a mobile sink is associated with two agents, which are the primary and immediate agents. The primary agent acts as a static sink during the mobile sink is in a cell (i.e., distance-based periodic reconstruction). While keeping movement, the sink

continuously elects new immediate agent and sends the location of the immediate agent to the primary agent for future data forwarding. The trajectory forwarding is an effective single path management scheme. Yu et al. proposed a path management scheme based on the overhearing feature of the wireless medium [15]. They include the location information of the sink in the data packets. Therefore, the data packet from the agent has the up-to-date location of the sink. This updated location information is gradually propagated to the successive path nodes; consequently, the path would be smoothly modified toward the current location of the sink. Jain et al. proposed a query-driven routing protocol for WSNs with mobile sinks [16]. They exploit a virtual structure, called wheel, to deliver query and data between sensors and a mobile sink. Due to the fact that the virtual wheel is a closed chain of sensors having one-node width, the location of a mobile sink can be thoroughly traced. The path management schemes support sink mobility without flooding so that considerable energy can be saved. However, in multipath routing, the existing schemes do not guarantee the path disjointness since there is no established path but a location-based connection between the sink and the agent. If the multipath is jointed each other into a single path, it causes loss of not only reliability or bandwidth but also service consistency. For some parts made up of multipath, using an alternative path or increasing bandwidth under the trust of them causes service disruption. This problem should be resolved through a multipath management method guaranteeing disjointness.

Recently, researchers have proposed path management mechanisms for multipath routing. Wang et al. proposed a route adjustment scheme for supporting sink mobility [17]. In this, the sensing field is divided into equal-sized clusters and each cluster head acts as an agent. Therefore, a mobile sink updates its location to the closest cluster head so that the data from sensors are delivered through the cluster head. Aswale and Ghorpade proposed a multipath routing protocol based on the triangle link quality metric for enhancing the traditional link quality estimation method using link quality indicator or packet reception ratio [18]. The proposed protocol includes the path maintenance phase in which the multipath are recovered using route alert and route discovery messages to exclude the exhausted or low-quality path nodes. Sreeram et al. suggested the enhanced route recovery mechanism where the broken link is restored without route alert messages [19]. The path management mechanisms maintain the end-to-end connection. However, they suffer from the route discovery delay when the paths are broken due to the continuous movement of the mobile sink. The initial idea of MPM was presented in [20] but this primitive method is designed for a dense deployment of sensors. Therefore, it is hard to be applied when sensors are sparsely deployed or there are void areas. On the other hand, MPM is dealing with the multihop path management scenarios so that more general environment can be covered.

3. Proposed Scheme

3.1. *Preliminary.* To elaborate on MPM, we defined the network model as follows. The sensor nodes are uniformly

deployed on a square sensor field. There are a mobile sink and a source node that is randomly selected among sensor nodes. A sensor node is aware of the location of itself and neighbors through a localization method and beacon exchanges. At the initial stage of the network, the multipath is constructed between the source and the sink. Let P_N be a path consisting of a set of nodes having same unique path ID N , and we call the nodes the path nodes. The PN_N^H is a path node of P_N with hop distance H from the source. That is, $P_i = \{PN_i^1, PN_i^2, \dots, PN_i^j\}$ where i is the path ID and j is the length of the path. Among the path nodes, the agent is the node having the largest hop distance (i.e., PN_i^j is the agent of P_i). That is, the agent is a neighbor of the sink.

There are two path management functions: path shortening and path extending. First, path shortening is conducted when the sink meets a path node. Due to the fact that any path node was closer to the source than the agent of the corresponding path, altering the last-hop connection of the path from the agent to the discovered path node provides higher reliability and lower energy consumption. In this case, the discovered path node becomes a new agent, and the subpath from the new agent to the old agent is released. Second, path extending is conducted in case the sink moves out of the transmission range of an agent but there is no path node that has the same path ID with the agent within the sink's transmission range. In this case, the sink has to elect a new agent among neighboring nodes and to establish the connection between the new agent and the old agent for restoring the path. Figure 2 shows the two path management cases according to the above classification. It is worth noting that path management could be applied in a multihop manner as shown in Figures 2(c) and 2(d). To check the existence of a path node or the absence of an agent, the mobile sink should periodically update its neighbor list. We will discuss the impact of the neighbor list update at the later of this section.

The mobile sink manages the status of the agent using the agent table. Generally, a mobile sink is referred to as a powerful node having more energy and better processing power than sensor nodes. To prolong the network lifetime, we actively use the capacity of a mobile sink. In other words, MPM imposes the overheads of multipath management to the mobile sink to reduce the energy consumption of the sensor nodes. In MPM, a mobile sink is responsible for state management, condition check, and process initialization.

A mobile sink supervises the multipath management through the agent table. The number of agent table entries is equal to the number of paths. The agent table includes path ID, agent location, hop distance, path priority, and path state. Path ID is uniquely assigned to each path when the multipath is constructed. The agent location indicates the geographical coordinate of the current agent having the path ID. Hop distance contains the hop count from the source to the agent. To help path management, the packet header of MPM includes path ID, sender's location, and sender's hop distance from the source. A mobile sink can understand and manage the state of each path by extracting the header of the received packet from the sensors.

Path priority and path state are assigned by a sink and specially used for path extending. Most existing path discovery schemes exploit the shortest path or the most reliable path (i.e., a path consisting of most reliable nodes/links) as a primary path. The other paths are subpaths or alternative paths, constructed beside the primary path. In MPM, the priority of the primary path is 0. And the priorities of left- and right-side paths of the primary path are assigned alternately. As a result, the odd priority paths are left side, and the even priority paths are the right side of the primary path. Which side the odd- or even-priority paths reside is of no importance but we designate that for the clear explanation. The path priority values are initialized whenever the multipath is reconstructed. The values of the path state field are 'valid' when the connection from the sink to the corresponding agent is available. Figure 3 shows an example of the constructed multipath and the resulting agent table.

3.2. Path Management Algorithm. The mobile sink continuously checks the list of the neighboring sensors since the change of the neighbor list might require path management. There are two significant changes in the neighbor list: the addition of a path node and the elimination of an agent. The addition of a path node means the opportunity of shortening the path because the agent has the largest hop distance toward the source. On the other hand, the elimination of an agent means that a path is not available, and consequently, the routing path between the sink and the agent has to be reestablished. Therefore, the sink performs the path shortening or path extending process according to the presence of neighbor nodes and the condition of the agent table.

The addition of a path node and elimination of an agent may occur simultaneously according to the length of the neighbor list update interval and the positions of the path nodes. The concurrence of the addition and elimination events can happen on different paths or on the same path. In the different-path case, the sink has to deal with the elimination event first since recovering a path is related to the 'availability' of the path whereas shortening the length of a path is related to the 'efficiency' of the path. In the same-path case, on the other hand, the sink simply discards the eliminated agent because the availability of the path corresponding to the agent would obviously be restored by electing the added path node as the new agent.

3.2.1. Path Shortening. If the addition of any path node has been identified, the sink conducts the path shortening process. First, the sink notifies the added path node that it has become the new agent of the path to establish the last-hop connection of the path. Then, the sink updates the agent table with the information of the new agent. The sink finds the agent table entry having the same path ID with the new agent and updates the agent location and hop distance fields. The path state field does not change since the connection between the sink and the new agent has been established.

The path shortening process can also be performed by a data packet from a path node. In other words, a mobile sink can identify the existence of the new neighboring path node before updating the neighbor list by overhearing a data packet.

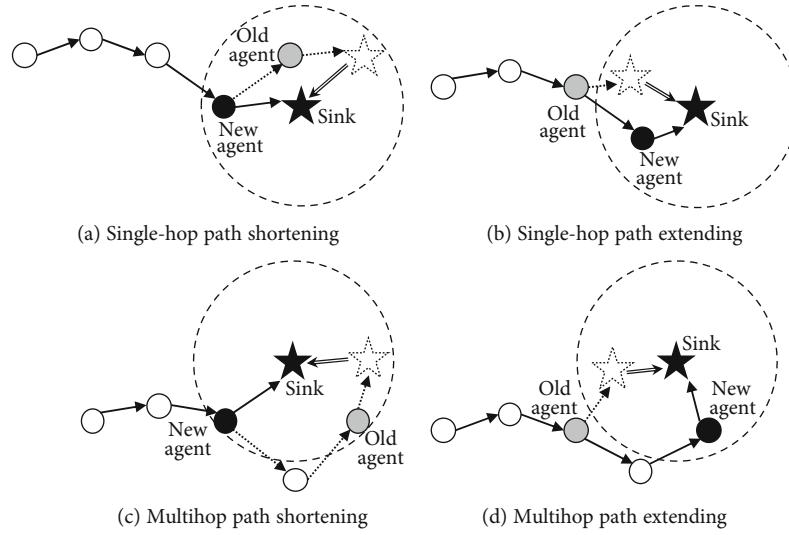


FIGURE 2: Path shortening and path extending.

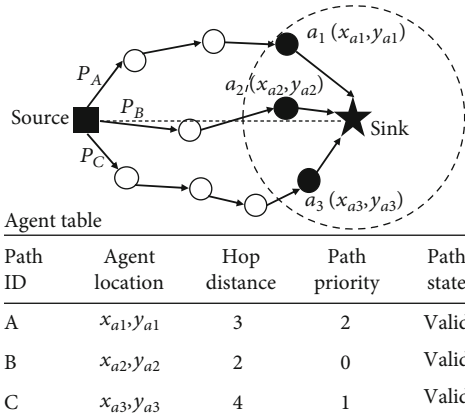


FIGURE 3: Path construction and agent table.

When a sink receives a data packet, the sink extracts the packet header and finds an agent table entry having the same path ID of the header. Then, the sink compares the hop distance of the header and the one of the agent table entry. If the hop count of the header is smaller, the sink conducts the abovementioned path shortening process with the sender of the data packet as the new agent. After that, the sink adds the new agent in the neighbor list so that the addition of the agent at the next neighbor list update can be ignored.

After the agent table updated, the sink sends the path release message to the new agent. The new agent forwards the message along the old path so that old path nodes including the old agent remove their path states. Figure 4 shows an example of path shortening. In the figure, nodes a_3 and a_4 are the old agent and the new agent, respectively. After node a_4 has become the new agent, the agent table entry about path C is updated with the information of new agent.

3.2.2. Path Extending. On the other hand, if the elimination of an agent was detected, path extending is performed. The sink finds the agent table entry corresponding to the elimi-

nated agent and the changes the path state to 'invalid' which means the last-hop connection is lost. Technically, path extension is the route discovery from the sink to the old agent. The criterion for selecting the next hop node is similar to the void handling techniques. The sink draws a virtual line between the location of itself and the location of the old agent which comes from the agent table and searches the next-hop node clockwise or counterclockwise from the virtual line. The searching direction depends on path priority. That is, the sink searches clockwise for even priority paths and counterclockwise for odd priority paths. In the case of the primary path (i.e. the priority is 0), the direction faces the farther agents of the second-order paths (i.e., the priority is 1 or 2). During the searching process, the firstly discovered node is selected as the new agent whereas the agents of other paths are skipped. The purpose of the strategies for selecting a new agent is to promote the path disjointness. Due to the fact that an extended path consists of the consecutively elected agents, the new agent should be located where there is no impact on other paths. Figure 5(a) shows an example of path extending. The sink searches counterclockwise from the virtual line between the location of itself and the location of a_3 (i.e., the old agent) because the priority of the path C is 1. Due to the searching direction, node n_1 is excluded and it might be the new agent of the primary path. Finally, node n_2 , which is the firstly discovered candidate, is elected as the new agent of path C. In the case of the absence of node n_2 , the path extension process can establish a multihop route from the sink to the old agent as shown in Figure 5(b). The sink selects node n_3 as the new agent and the searching process continues until the route reaches the old agent. During the searching process, the virtual line toward the old agent and searching direction are maintained from the sink.

The area for the candidates is restricted a half-side with the straight line between the sink and old agent as a center. Which side is the area for the candidates is decided by whether the path priority is odd or even. For example, as shown in Figure 5, the path priority of path C is 1, an odd

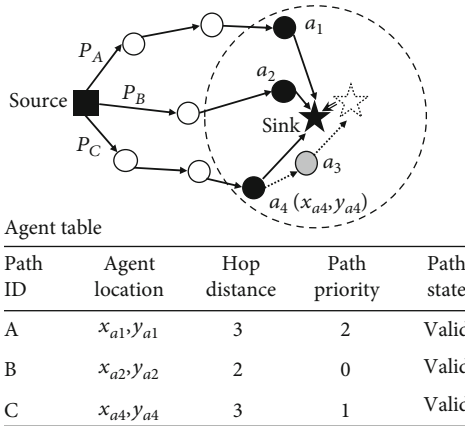


FIGURE 4: Agent table update for path shortening.

number. At the initial stage of path extension process, the candidate for the new agent is 4 (node a_2 , n_1 , n_2 , and n_3). Node a_2 , which is already a path node, is excluded. Node n_1 is also excluded because it is on the right side. Between node n_2 and n_3 , n_3 is finally elected as a new agent since node n_3 is further from the straight line between the sink and node a_3 , an old agent. In case that the path order is 0 (primary path), the new agent can be elected from both areas. Among the nodes in the area, the highest priority candidate is the node that is closest to the sink for reducing the frequency of the path extending process. After the new agent is elected, the agent table is updated.

3.3. Discussion

3.3.1. Path Reconstruction. After a mobile sink moves a long distance, overly long extended paths or inefficiently extended paths (e.g., zigzag) might be constructed. In this case, the end-to-end multipath should be reconstructed rather than keeping an immoderate path extension. There are two methods that can be adapted: cost-based and periodic reconstruction. The cost-based reconstruction is performed when the cost of conducting reconstruction is smaller than the cost of keeping extended paths. Therefore, defining the cost is not only the nominally important factor but also a key factor influencing the actual performance. Definitely, both costs (i.e., for conducting reconstruction and keeping extended paths) have to include not only the path construction cost but also future data delivery costs. This approach should be designed complementary with the path construction scheme and the purpose of the application. Moreover, it is seriously influenced by external factors, such as the sink's mobility pattern, data generation pattern, and deployment of sensors. Although the cost-based reconstruction seems very efficient, it is generally unavailable due to the uncertainty of those external factors. For example, to calculate the cost for delivering a data packet from the source to a mobile sink, the exact values of the external factors should be available such as the sending time at the source, receiving time at the sink, location of the sink at the receiving time, and hop count of each extended/shortened path. That is, the cost-based reconstruction can be adopted by the application where the external factors

are strictly controlled. Therefore, the cost-based reconstruction may lead to inefficient end-to-end multipath reconstruction in general applications where those external factors cannot be strictly controlled. On the other hand, in the periodic reconstruction, the multipath is reconstructed every certain period of time or the distance that the sink moved after the multipath is reconstructed. This is very simple because it needs only to check time or distance. In addition, it can provide steady performance against the mobility pattern of a mobile sink and the data generation patterns of sensors. In this paper, we use the periodic approach for our computational simulation for providing relatively steady operation thereby clearly comparing the path management performance.

3.3.2. Neighbor List Update. In MPM, the mobile sink periodically updates its neighbor node list for identifying the opportunity of path shortening and the necessity of path extension. For neighbor list update, the sink broadcasts a beacon and the neighbor nodes respond to the beacon. During the beaconing process, the ordinary nodes just notify its existence to the sink whereas a path node includes the path ID and its hop distance in the response message for agent table update. The neighbor list update interval is an important tunable factor of MPM since it might influence the energy efficiency and data throughput. On the one hand, the frequent neighbor list update provides agile path management at the expense of the energy consumption of the sink and neighbor nodes for beaconing. On the other hand, the infrequent neighbor list update achieves energy savings but might suffer from the disconnection from the moment that the sink moves out of the agent's radio range to the next beaconing. In this context, the neighbor list update interval should be carefully configured with consideration of the mobility pattern of the mobile sink and the data packet generation patterns of sensor nodes.

4. Performance Evaluation

In this chapter, we present our simulation results to evaluate the performance of the proposed scheme. We implement the MPM in MATLAB. We also implement the multipath reconstruction (MPR) and single path management (SPM) schemes as the comparison group. In MPR, the end-to-end multipath between a source and a mobile sink is reconstructed whenever the sink moves outside the radio range of the agent. In SPM, new agents are continuously elected for constructing a single path along the moving path of the mobile sink after the multipath are constructed. In common, three schemes exploit the ideal multipath discovery method to minimize the effect of the multipath reconstruction overhead. In the ideal multipath discovery, a mobile sink is aware of the locations of every node. The sink constructs an ideal multipath toward the source without any signaling. The sink sends multiple copies of a message through all paths, and the source replies multiple copies of the ACK message along the reverse routes.

The default simulation setting is as follows. 1000 sensor nodes are uniformly deployed in the $500\text{ m} \times 500\text{ m}$ square field. The radio range of all nodes is 25 m. Each node

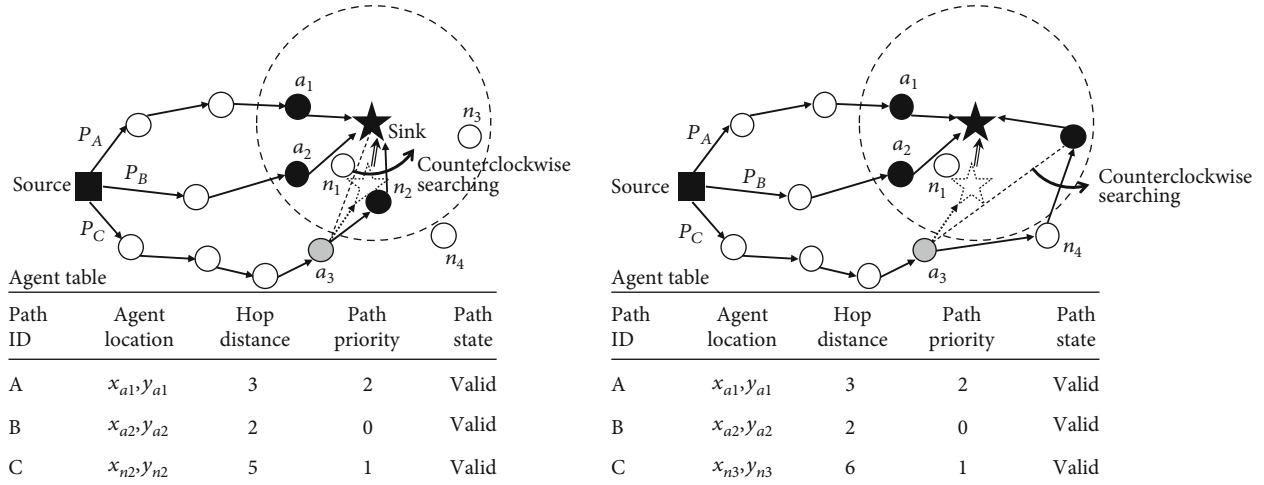


FIGURE 5: Agent table update for path extending.

consumes 20 mW and 15 mW for transmitting and receiving a packet, respectively. There are a source and a mobile sink. Since the data delivery performance is significantly affected by the hop count, we set the initial distance between the source and the mobile sink about 350 m. Under the condition, the source is arbitrary elected among all sensor nodes, and similarly, the initial location of the sink is designated randomly. The source generates a data packet every 5 seconds. The mobility of the sink follows the random waypoint mobility model [21] and the speed of the sink is 5 m/s. The mobile sink updates the neighbor list every 0.5 seconds. After the neighbor list update, the multipath reconstruction (in case of MPR) or path management (in case of SPM and MPM) might be conducted. The number of paths is 3, and they are reconstructed every 50 seconds (periodic multipath reconstruction). The packet delivery success ratio of all links is 95% and the retransmission strategy is excluded. Each simulation lasts 1000 seconds and the results are the average of 20 simulations for complying 95% confidence interval.

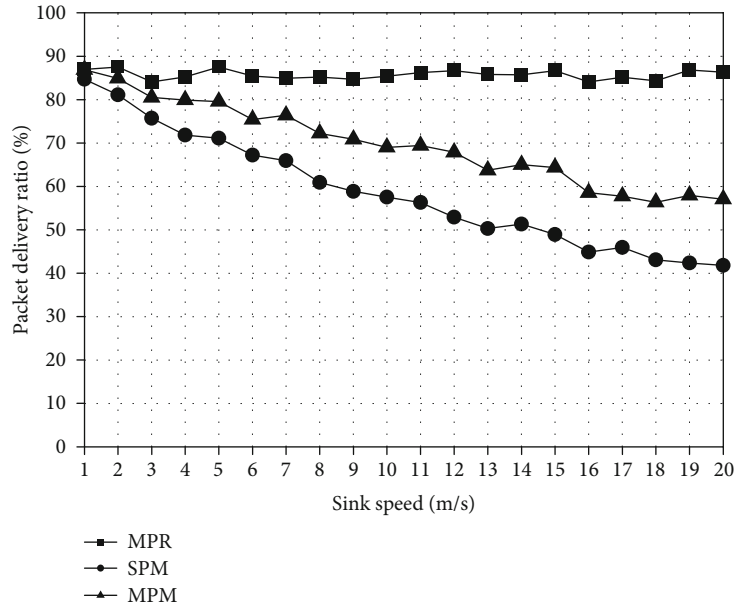
We choose two metrics: the packet delivery ratio and total energy consumption. The packet delivery ratio is the proportion of the number of received data packets of the sink to the number of sent data packets of the source. The total energy consumption is the sum of the consumed energy of all nodes including the sink. These metrics are the typical pros and cons of multipath routing. As a guiding mention, our all simulation results say that the MPM mitigates serious degradation of packet delivery success ratio without considerable energy consumption.

Figure 6(a) shows the packet delivery ratio according to the speed of the mobile sink. We vary the sink speed from 1 m/s to 20 m/s for including various types of mobile sinks ranging from handheld devices to connected vehicles. The MPR shows the highest packet delivery ratio because it tends to construct the shortest multipath than others. Since MPR reconstructs an end-to-end multipath whenever the sink moves outside the agent, the results are relatively steady. On the other hand, the packet delivery ratio of SPM decreases as the sink speed increases. This is because the portion of the single path increases as the sink speed increases in SPM. In

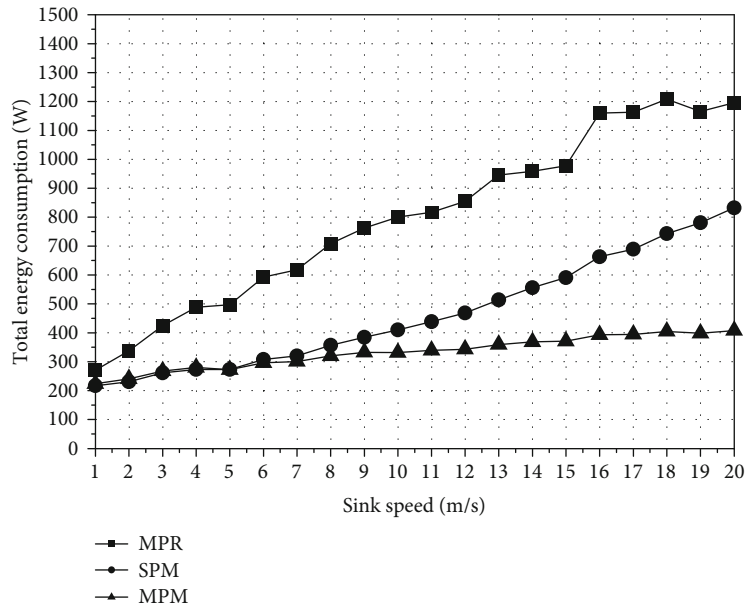
the case of the MPM, the packet delivery ratio is higher than SPM but lower than MPR. This is because the extended end-to-end paths of MPM are longer than the reconstructed paths of MPR although the number of multiple paths is the same. Figure 6(b) shows the total energy consumption according to the speed of the mobile sink. The noticeable tendency is that the total energy consumption of MPR increases rapidly as the sink speed increases. This is because the reconstruction frequency of MPR becomes higher as the sink speed increases. The total energy consumptions of SPM and MPM increased slowly as the speed of the mobile sink increased for constructing extended paths while MPM consumes more energy than SPM for extending multiple paths.

Figure 7(a) shows the packet delivery ratio according to the number of paths. We vary the number of paths from 2 to 5. All schemes show the increasing tendency because the path redundancy increases as the number of paths increases. The packet delivery ratios of MPR and MPM are higher than SPM but the one of MPM is slightly lower than MPR because the extended paths are longer than the reconstructed paths. The packet delivery ratio of SPM is lowest due to the extension of the single path. The difference between the SPM and others increases as the number of paths increases since other schemes use reconstructed or extended multipath while SPM uses the extended single path constantly. Figure 7(b) shows the total energy consumption according to the number of paths. The total energy consumption of the MPR increases sharply because the reconstruction overhead increases as the number of paths increases. On the other hand, the total energy consumptions of SPM and MPM increase gradually and almost the same. Although MPM manages more paths than SPM, the path shortening process makes the overall path management efficient.

Figure 8(a) shows the packet delivery ratio according to the packet generation interval varying from 1 to 10 seconds. As the packet generation interval increases, the source sends the decreased number of data packets but the routing paths are not affected. Therefore, the packet delivery ratios of all schemes are almost not changed. Figure 8(b) shows the total energy consumption according to the packet generation



(a) Sink speed vs. packet delivery ratio



(b) Sink speed vs. total energy consumption

FIGURE 6: Simulation result according to the speed of the mobile sink.

interval. The total energy consumptions of all schemes decreased exponentially since the energy consumption for data delivery decreased as the data generation interval decreased. In addition, the total energy consumption of all schemes converged into a specific value which implies the cost for path management. Specifically, the convergence value of MPR is much higher than others which means MPR consumes much more energy for multipath reconstruction. Meanwhile, SPM and MPM consume almost similar energy for path management.

Figure 9(a) shows the packet delivery ratio according to the neighbor list update interval. The neighbor list update interval implies how often the mobile sink recognizes the

disconnection with the agent. In other words, the disconnection duration increases as the neighbor list update interval increases. Therefore, the packet delivery ratio of all schemes decreased as the neighbor list update interval increased. The distinct feature is the peak points at some neighbor list update interval values, that is, 2.5 and 5 seconds. This result is related to the packet generation interval which is designated by 5 seconds in the default simulation setting. In other words, if the value of the neighbor list update interval is an aliquot of the packet generation interval, the duration of the disconnection can be ignored. Figure 9(b) shows the total energy consumption according to the neighbor list update interval. In cases of the SPM and MPM, the total energy

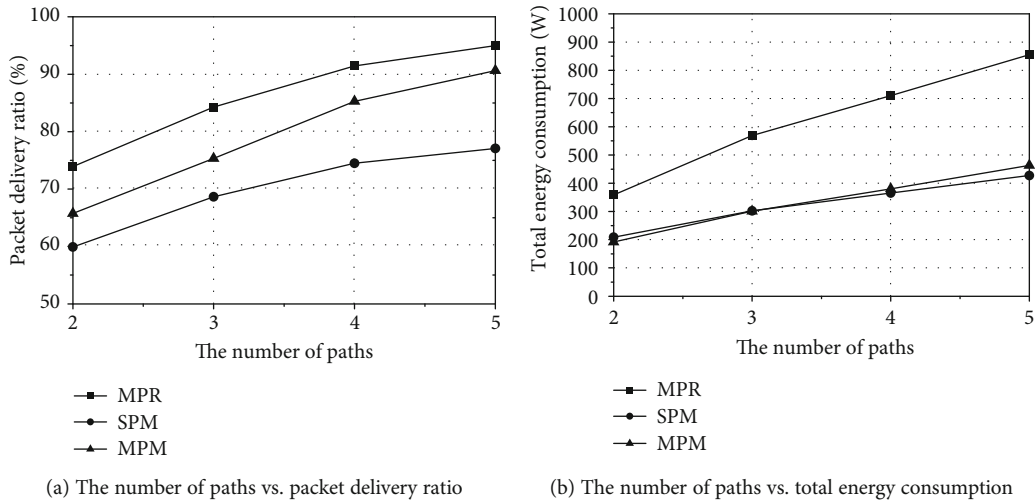


FIGURE 7: Simulation results according to the number of paths.

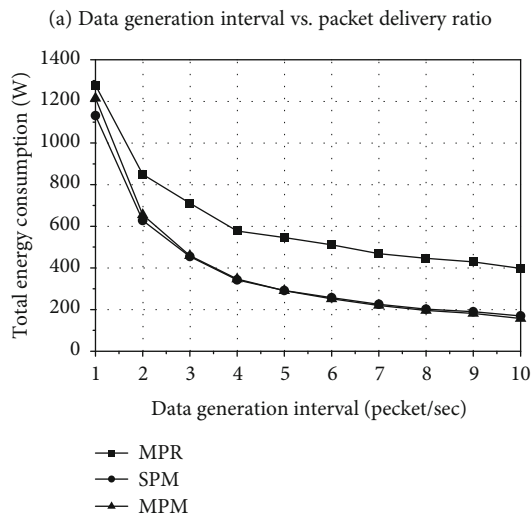
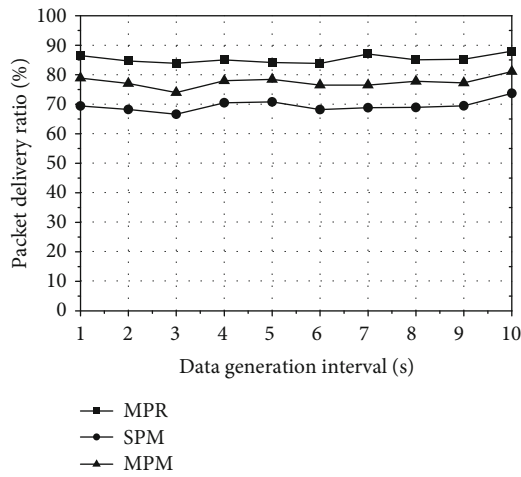


FIGURE 8: Simulation results according to the data generation interval.

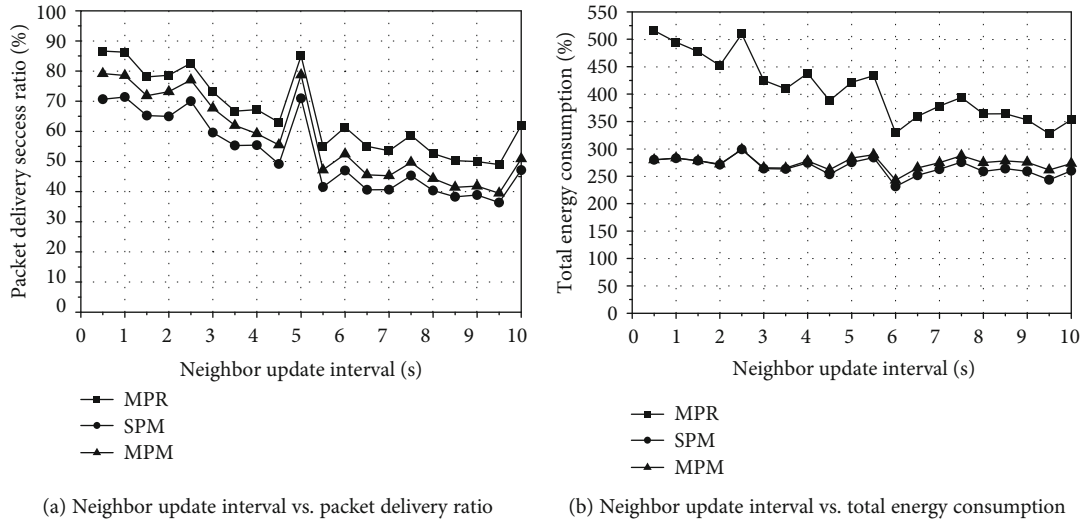


FIGURE 9: Simulation results according to the neighbor update interval.

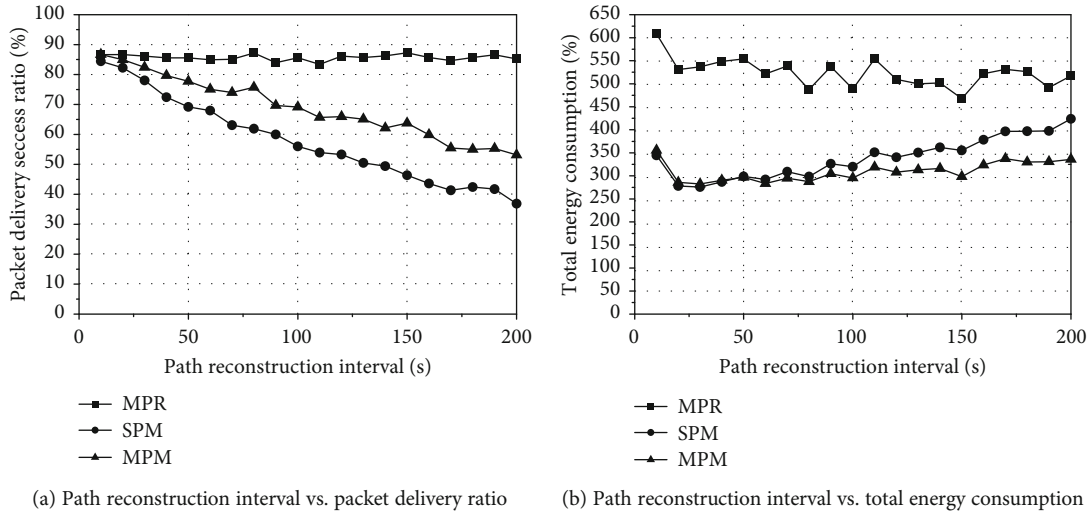


FIGURE 10: Simulation results according to the path reconstruction interval.

consumptions were almost steady which means the cost for neighbor list update and the local path management is very small compared to the end-to-end path reconstruction and data delivery. On the other hand, the total energy consumption of MPR decreased slowly as the neighbor list update interval increased. In the case of MPR, the end-to-end multipath reconstruction interval increases as the neighbor list update interval increases.

Figure 10(a) shows the packet delivery ratio according to the end-to-end multipath reconstruction interval which is the key factor in the periodic path reconstruction approach. The packet delivery ratio of MPR was almost steady because the end-to-end multipath reconstruction is performed whenever the sink leaves the agent regardless of the reconstruction interval. In cases of SPM and MPM, on the other hand, the packet delivery ratio decreased slowly whereas the MPM achieved a higher packet delivery ratio. It means that MPM manages multiple extended paths effectively even the reconstruction interval is significantly long. Figure 10(b) shows

the total energy consumption according to the multipath reconstruction interval. In the case of MPR, the total energy consumption was almost not changed because the multipath reconstruction is performed not only periodically but also actively at the neighbor list update. On the other hand, SPM consumed a slightly increased amount of energy due to data delivery through the long extended routing path. In the case of MPM, the total energy consumption was almost not changed because it manages multiple routing paths very efficiently during the path reconstruction interval.

5. Conclusion

In this paper, we propose a novel multipath management scheme called MPM for supporting sink mobility in wireless sensor networks. Existing multipath routing protocols need a significant amount of overhead for reconstructing an end-to-end multipath for supporting sink mobility. Meanwhile, the existing single path management schemes maintain the

end-to-end connection without reconstruction by extending the constructed path. However, the single path management schemes cannot provide high-performance data delivery as multipath routing. To solve the problems, MPM extends multiple paths using the agent-based multipath management thereby maintaining the high-quality data delivery through multipath. Our simulation results show that MPM provides cost effectiveness compared to the existing multipath reconstruction and single path management schemes. Depending on the network parameter such as the speed of the mobile sink and the data generation interval of sensors, tuning neighbor list update interval and end-to-end multipath reconstruction interval influence the performance of MPM significantly.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This work was supported by the Korea Institute of Science and Technology Information (KISTI).

References

- [1] S. Yu, B. Zhang, C. Li et al., "Routing protocols for wireless sensor networks with mobile sinks: a survey," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 150–157, 2014.
- [2] E. Hamida and G. Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 31–37, 2008.
- [3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.
- [4] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRANet of things to a future INTERNet of things: a wireless- and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.
- [5] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, 2012.
- [6] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849–857, 2010.
- [7] W. Lou and Y. Kwon, "H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1320–1330, 2006.
- [8] G. Wang, T. Wang, W. Jia, M. Guo, and J. Li, "Adaptive location updates for mobile sinks in wireless sensor networks," *The Journal of Supercomputing*, vol. 47, no. 2, pp. 127–145, 2009.
- [9] J. Yim, J. Bang, Y. Nam, Y. Shin, and E. Lee, "Efficient Multipath Routing Protocol Against Path Failures in Wireless Sensor Networks," in *12th IFIP Wireless and Mobile Networking Conference (WMNC)*, vol. 2019, pp. 136–140, Paris, France, 2019.
- [10] P. Mohanty and M. R. Kabat, "Energy efficient reliable multipath data transmission in WSN for healthcare application," *International Journal of Wireless Information Networks*, vol. 23, no. 2, pp. 162–172, 2016.
- [11] M. Z. Hasan, H. Al-Rizzo, and F. Al-Turjman, "A survey on multipath routing protocols for QoS assurances in real-time wireless multimedia sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1424–1456, 2017.
- [12] K. Jaiswal and V. Anand, "EOMR: an energy-efficient optimal multi-path routing protocol to improve QoS in wireless sensor network for IoT applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2493–2515, 2020.
- [13] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRADient broadcast: a robust data delivery protocol for large scale sensor networks," *Wireless Networks*, vol. 11, no. 3, pp. 285–298, 2005.
- [14] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "TTDD: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 161–175, 2005.
- [15] F. Yu, S. Park, E. Lee, and S.-H. Kim, "Elastic routing: a novel geographic routing for mobile sinks in wireless sensor networks," *IET Communications*, vol. 4, no. 6, pp. 716–727, 2010.
- [16] S. Jain, K. K. Pattanaik, and A. Shukla, "QWRP: query-driven virtual wheel based routing protocol for wireless sensor networks with mobile sink," *Journal of Network and Computer Applications*, vol. 147, p. 102430, 2019.
- [17] J. Wang, J. Cao, S. Ji, and J. H. Park, "Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3277–3290, 2017.
- [18] S. Aswale and V. R. Ghorpade, "Geographic multipath routing based on triangle link quality metric with minimum inter-path interference for wireless multimedia sensor networks," *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [19] K. Sreeram, A. Unnisa, V. Poornima, and S. Chaudhari, "QoS aware multi-constrained node disjoint multipath routing for wireless sensor networks," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 382–385, Coimbatore, India, 2019.
- [20] C. Kim, Y. Yim, T. Yang, S. Kim, and S.-H. Kim, "Multipath management scheme for supporting sink mobility in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2017, pp. 1–6, San Francisco, CA, 2017.
- [21] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257–269, 2003.

Copyright © 2020 Cheonyong Kim et al. This work is licensed under <http://creativecommons.org/licenses/by/4.0/>(the “License”). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License.