A Mobility-Aware Node Deployment and Tree Construction Framework for ZigBee Wireless Networks

Yuan-Yao Shih, *Student Member, IEEE*, Wei-Ho Chung, *Member, IEEE*, Pi-Cheng Hsiu, *Member, IEEE*, and Ai-Chun Pang, *Senior Member, IEEE*

***Abstract*—ZigBee is a specification formalized by the IEEE 802.15.4 standard for low-power low-cost low-data-rate wireless personal area networks. In ZigBee networks, a tree topology is often used to construct a wireless sensor network for data delivery applications. However, delivery failures constantly occur in ZigBee wireless applications due to node movements and network topology changes. The conventional route reconstruction method is designed to mitigate the effects of topology changes, but it consumes a large amount of resources. In this paper, we exploit the regularity in node mobility patterns to reduce the frequency of route reconstructions and ensure that the transmission of data to mobile nodes is efficient. To increase the data delivery ratio and mitigate the effects of packet loss caused by the node mobility, we propose a ZigBee node deployment and tree construction framework. In particular, the framework considers the regularity in mobility patterns during the construction of the routing tree and deployment of nodes. It also includes an overhearing mechanism for mobile nodes to further improve the data delivery ratio. We present details of the proposed algorithms for node deployment and tree construction in the framework. The effectiveness of network topologies constructed under the framework is demonstrated through comprehensive ns-2 simulations based on two real-world scenarios. The results show that our approach can construct ZigBee tree topologies with a high data delivery ratio and low routing overhead.**

***Index Terms*—Mobility robustness, tree topologies, ZigBee wireless networks.**

I. Introduction

With the increasing sophistication of wireless communications and sensing technologies, various sensor-based applications, such as tour guiding and industrial automation, generate tremendous economic and social benefits. The potential for even greater impact has motivated extensive studies on wireless sensor networks (WSNs) in recent years [1]–[5]. For example, the ZigBee standard, designed by the ZigBee Alliance [6], specifies the network and application layers for sensing data deliveries.

Many ZigBee applications, such as tour guiding and indoor/building monitoring systems, require moving objects to be equipped with an end device that is connected to a backbone network for data collection and dissemination [7]–[10]. Another category of applications use ZigBee routers as roadside units and end devices as in-vehicle units. In such applications, ZigBee cluster-tree networks can serve as vehicle-to-infrastructure communications in vehicular ad hoc networks (VANETs), because ZigBee can provide low power consumption, medium data rates, and reliable communications [11], [12]. With ZigBee technologies, various intelligent transportation system (ITS) applications in VANETs such as traffic control, system-aided navigation, location-based information pushing, and vehicular safety access control can be realized [13]–[15]. Normally, routers that are connected to the backbone network are static and equipped with reliable power supplies, whereas mobile end devices rely on batteries. In many applications such as drivers who receive traffic information from ITSs, tourists who receive recreational information, and workers who receive supervisory messages, the major function of mobile end devices is to receive data from the network coordinator rather than send data through the network. Chen *et al.* [16] showed that, even if both the end device and the router are mobile, a cluster tree performs better than a mesh structure does when the end device is receiving data. Therefore, this paper focuses on the ZigBee cluster tree as the main topology. Moreover, the ZigBee specification allows a small and simple protocol stack [17] and, thus, has lower implementation cost compared

Manuscript received June 8, 2012; revised December 26, 2012; accepted January 27, 2013. Date of publication February 7, 2013; date of current version July 10, 2013. The work of W.-H. Chung was supported in part by the National Science Council of Taiwan under Grant NSC101-2221-E-001-002, Grant NSC101-2221-E-001-008, and Grant NSC101-2622-E-001-001-CC3. The work of P.-C. Hsiu was supported in part by the National Science Council under Grant NSC101-2219-E-002-002. The work of A.-C. Pang was supported in part by the National Taiwan University through the Excellent Research Projects under Grant 102R890822, the National Science Council under Grant NSC101-2221-E-002-018-MY2 and Grant NSC101-2221-E-002-052, Chunghwa Telecom, Information and Communications Research Laboratories, Industrial Technology Research Institute, and the Research Center for Information Technology Innovation, Academia Sinica. This paper was presented in part at the 2011 IEEE International Conference on Communications. The review of this paper was coordinated by Prof. F. R. Yu.

Y.-Y. Shih is with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan, and also with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: ckm@citi.sinica.edu.tw).

W.-H. Chung and P.-C. Hsiu are with the Research Center for Information Technology Innovation and also with the Institute of Information Science (IIS), Academia Sinica, Taipei 115, Taiwan (e-mail: whc@citi.sinica.edu.tw; pchsiu@citi.sinica.edu.tw).

A.-C. Pang is with the Graduate Institute of Networking and Multimedia and the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, and also with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan (e-mail: acpang@csie.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2013.2245693

with Bluetooth and Wi-Fi. The much lower power consumption of ZigBee, compared with Wi-Fi [17], also facilitates a long lifetime of mobile end devices, which greatly benefits the aforementioned applications.

A network with highly mobile users raises challenging mobility issues. Based on the ZigBee specification [6], a *device discovery* procedure is triggered if the central server cannot locate a certain mobile end device. During the procedure, the central server simply floods the whole network with messages to locate the displaced end device. However, flooding the network is costly in terms of resources, and during the procedure, the network cannot accommodate multiple instances of rapid node mobility [16]. Thus, we need a more efficient and automatic approach for locating mobile end devices. In many applications, the mobility patterns of sensor nodes are inherently regular due to the geographical structure of the network or physical constraints. The regularity provides useful information that can be exploited to construct a proper routing topology for sensing data deliveries.

To improve the downlink data delivery ratio, we propose an approach that exploits the aforementioned information to optimize the locations of routers and construct a mobility-robust tree topology in a ZigBee wireless network. The approach deploys routers and constructs a topology with the property that mobile nodes will move along the constructed data-forwarding path with high probability. Data will reach the target mobile nodes as long as they are within the transmission range of any router on the forwarding path. In other words, we choose the positions of the routers and design the tree topology so that most movements are directed toward the root of the tree. To achieve our objective, we gather information about node movements in the environment and construct a ZigBee tree topology framework. In particular, the framework considers the regularity of the mobility patterns during the construction of the tree and deployment of the routing nodes, and it incorporates an overhearing mechanism for mobile nodes to further improve the data delivery ratio. We also design heuristic and low-complexity algorithms for node deployment and tree construction and analyze their performance in ZigBee networks. The effectiveness of network topologies that consider mobility regularity is demonstrated through the ns-2 network simulator, in which we incorporate a network deployment tool that we developed [18], [19]. The simulation results show that, compared with conventional approaches, the proposed approach achieves significant improvements in data delivery in real-world scenarios with different mobility patterns.

The remainder of this paper is organized as follows. In Section II, we review the related work, and in Section III, we describe the system model and formulate the problem. In Section IV, we discuss the proposed algorithms for tree topology construction. The simulation results and analysis are presented in Section V. Section VI contains some concluding remarks.

II. Related Work

In this section, we discuss mobility mechanisms that were proposed in the literature and explain the difficulties that arise in the direct application of existing mobility patterns to a ZigBee network. We also consider studies of mobility support in WSNs and research on the deployment of such networks.

A. *Enabling Connectivity With Node Mobility in MANETs*

The issue of mobility in mobile networks, e.g., mobile ad hoc networks (MANETs), delay-tolerant networks (DTNs), and mobile wireless sensor networks (MWSNs), has generated much research interest in recent years [20], [21]. In particular, for MANETs or DTNs, certain routing schemes that utilize "mobility prediction" have recently been proposed. Reference [22] presents a routing scheme using the trajectories of the mobile nodes to predict their future moving direction, and then, the sender uses the predicted information to choose the proper forwarding node moving toward the message recipient. Reference [23] builds a virtual coordinate of high-dimensional Euclidean space based on the mobility patterns of mobile nodes. Each dimension in the proposed system represents the frequency with which a node is found in a particular location. This system is then used to compute the routing path. Reference [24] proposes socially aware routing based on the property that mobile nodes usually move among a small set of socially significant points (hubs). The routing scheme tends to forward messages toward one of the hubs the recipient visits with high frequency. Reference [25] tries to achieve good delivery performance and low end-to-end delay in a sparse MANET where nodes may freely move by casting the route-finding problem as an information flow problem in a social network.

Although the aforementioned approaches are effective in their respective scenarios, challenges remain in the mobility issue in mobile sensor networks because of ZigBee's unique features, i.e., low power consumption, low data rates, and short communication ranges. ZigBee is designed for the delivery of small amounts of data at low data rates. Thus, routing schemes that utilize mobility prediction for MANETs and DTNs cannot directly be applied to a ZigBee network, because they require online movement data collection, and the route computation will introduce additional computation and message exchange overheads unaffordable in ZigBee.

B. *Localization and Mobility in WSNs*

A number of works focus on the localization in MWSNs [26], [27]. Many MWSN localization algorithms determine the location of nodes by transforming the measured strength of signals emitted by neighboring sensors into a position estimated through geometric manipulation, e.g., triangulation. Through an anchor, i.e., a fixed sensor node that has information about its location, the absolute coordinates of mobile sensors can be obtained. To pinpoint the precise location of a sensor node, MWSN localization algorithms often introduce large complexity because of information exchange and the consequent network delay. In contrast to applications that require accurate node positions, for data delivery in ZigBee applications, only the information about routers to which the destination node can be connected are needed.

Another strategy called "network repair" tries to resolve node mobility problems in a WSN by reconstructing the topology to maintain the connectivity. In [28] and [29], the authors consider generic network repair issues. Based on the approach proposed in [29], if a node finds that the link to the parent is broken, it broadcasts a message to find a neighboring node with the shortest path to the coordinator and adopts that node as its new parent. The drawback of this scheme is that it may cause loops in the repaired network topology. The approach introduced in [28] is a reinforcement scheme that allows the parent node of a failed link to participate in the recovery. However, as shown in [29], this scheme also increases the communication overhead and convergence delay.

There have been attempts to efficiently deliver data to mobile nodes in WSNs based on the property where the movement of the mobile nodes shows a certain degree of regularity and the path of those nodes can probabilistically be predicted. Reference [30] introduces a routing scheme called "data stashing" to minimize energy consumption and network congestion by exploiting knowledge about the mobility of the mobile sinks. Instead of directly sending data from a sensor to the sink, the data are sent to certain relay nodes located on the trajectories of the sink. The trajectory of the sink is either announced by the sink itself or predicted based on the observed mobility regularity of the sinks. As sinks traverse these relay nodes, the relay nodes then send the data directly to the sinks. Adopting a similar idea, [31] focuses on determining the optimal set of relay nodes from the trajectories of the sinks to minimize the total routing cost for the sensors to send collected data to the sinks. The idea of "data stashing" is designed for data collection applications and also introduces long and unsteady data delivery delay, because the data stashed in the relay nodes are waiting for the mobile sinks to come. As a result, the idea cannot directly be applied to the data delivery applications focused in this paper.

To mitigate the effect of route interruptions caused by node mobility and realize ZigBee applications, a number of researchers have studied mobility issues in wireless ZigBee networks. In ZigBee, if a node loses the link to its parent, the node and its descendants must *rejoin* the network, which can be time consuming and may also increase the communication overhead. A recent empirical study of mobility support in ZigBee networks concluded that ZigBee's provisions for mobility management are inadequate [16]. The solution that was proposed in [32] attempts to address this problem by introducing a faster network repair scheme that comprised a regular repair scheme and an instant repair scheme. The network coordinator periodically applies the regular repair scheme to maintain the network topology, whereas the instant repair scheme is triggered if a node loses the link to its parent. Both schemes are designed for data collection applications, i.e., uplink transmissions; however, we focus on data delivery applications in this paper.

C. Deployment of WSNs

The method that is used to deploy a WSN has a major impact on the topology and efficiency of the network. A well-calibrated deployment can reduce the cost, increase the robustness, and improve the energy efficiency of a network. Various factors must be considered when we deploy WSNs in different applications. For example, coverage and connectivity are crucial factors in sensing and data collection applications. In certain situations where user behavior patterns are partially known, the sensor nodes can be deployed in predetermined locations to fully satisfy users' demands [33]. The performance of this kind of deployment is quite predictable, but this kind of deployment requires prior information about the environment and users. In contrast, random deployment does not require prior environmental information [34], [35]. Hence, it is suitable for ad hoc situations such as battlefield or disaster rescue operations.

Aside from coverage and connectivity, other aspects of sensor networks, such as energy consumption, are of interest to researchers. Tian and Georganas [36] propose a node-scheduling scheme that tries to reduce a system's energy consumption and thereby increase the system's lifetime. The scheme alternately disables redundant sensors but still maintains the original sensing coverage. In [37], Dhillon and Chakrabarty present two algorithms to support distributed applications for efficient deployment. The algorithms try to optimize the number of sensors and determine their locations. Chakrabarty *et al.* [38] formulate the sensor field as a 2-D or 3-D grid of coordinates and utilized the integer linear programming to determine the minimum number of sensors needed to cover the sensor field of interest.

In this paper, we focus on solving the mobility problem of data delivery in ZigBee networks while attempting to minimize the network overhead. As mentioned in Section II-A, many schemes that deal with mobility issues in MANETs and DTNs cannot directly be applied to ZigBee networks; however, they provide hints about how we can exploit the regularity in the mobility of nodes to improve mobility support in the network. In addition, to date, most research on ZigBee mobility support has focused on data collection applications. The network repair strategy demonstrates the importance of constructing or maintaining a connected network topology, but it increases the overhead when the network is operating; thus, it is not suitable for large networks. Meanwhile, the majority of works on the deployment of WSNs focus on the coverage, connectivity, and cost of sensors. On the other hand, the approaches in [36] and [37], as discussed in the previous paragraph, determine the locations of network routers based on different sensing interests. These observations motivate us to consider the mobility regularity and unique features of ZigBee when designing WSNs. By exploiting the end device's historical movement data in our node deployment and tree construction algorithms at the design stage, the operating overhead such as route searching packets and network control messages as in many conventional methods can substantially be reduced.

III. System Model and Problem Definition

In this section, we introduce the system model, the network architecture, the configuration parameters, and the assumptions underlying the system model. Then, based on the system model, we formally define the design objective and the problem under investigation.

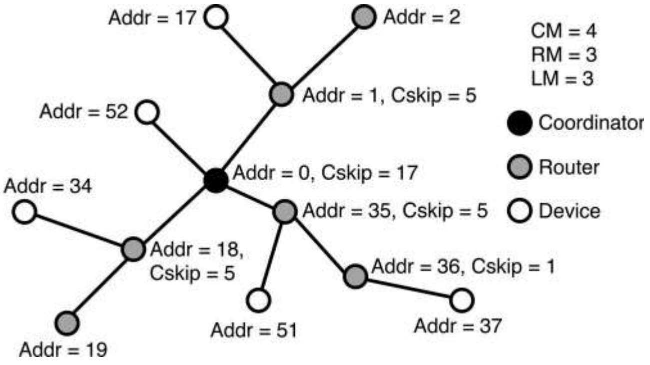


Fig. 1. ZigBee cluster tree.

A. System Model

A ZigBee network comprises the following three types of devices: 1) a coordinator; 2) multiple routers; and 3) multiple end devices. The coordinator performs the initialization, maintenance, and control functions in the network. A router is responsible for routing data between the end devices and the coordinator. An end device is not equipped with forwarding capability, and its hardware requirements are minimized to control costs. With the three types of devices, the ZigBee standard supports the following three network topologies: 1) star networks; 2) cluster-tree networks; and 3) mesh networks. In a star network, multiple end devices directly connect to the coordinator, and in a cluster-tree network, routers form clusters with their surrounding devices. Moreover, in cluster-tree and mesh networks, the devices communicate with each other in a multihop fashion.

Among the three ZigBee network topologies, the cluster tree is the most suitable for low-power WSNs, because it supports the superframe structure, which is responsible for power-saving operations in IEEE 802.15.4.[1] In addition, based on a distributed address assignment policy, the cluster tree supports a very lightweight routing protocol without maintaining a routing table. Before a topology is constructed, the following three system configuration parameters must be set: 1) the maximum number of children of a router or the coordinator (Cm); 2) the maximum number of child routers of a router or the coordinator (Rm); and 3) the depth of the network (Lm). Fig. 1 shows a ZigBee cluster-tree topology with the parameters $Cm = 4$, $Rm = 3$, and $Lm = 4$.

We consider the system model based on the ZigBee cluster-tree network with a coordinator, routers, and mobile end devices. The coordinator acts as the tree root, whereas the routers serve as internal nodes in the tree for data forwarding, as in the conventional ZigBee cluster-tree network. The difference between the proposed scheme and the conventional ZigBee network lies in the operations of the mobile end devices. In contrast to the address assignments in the conventional ZigBee network, every mobile end device in our network is randomly assigned a unique address that is different from those preallocated to the coordinator and routers. Consequently, the tree construction is not subject to the constraint on C_m but is subject only to constraints on R_m and L_m. To be adaptive to rapid topology changes, the proposed scheme does not impose any associations between the mobile end devices and routers. Instead, a mobile end device simply sends a packet, which is then forwarded to the coordinator through the routers. Upon the reception of a packet from a mobile end device, the router forwards the packet to its parent, as indicated by the tree structure. The location of the mobile end device is recognized by the network and maintained by the coordinator, which identifies the last router that was used to forward the end devices uplink data packets. When a downlink packet is sent to a mobile end device, the coordinator delivers the packet to the last recorded location, i.e., the last router that received the uplink packet from the mobile end device. Upon the reception of the downlink packet, the router simply forwards it to the mobile end device and waits for an acknowledgement message from the end device. If the mobile end device has moved from the last known location, the data delivery fails, and the coordinator starts a search by broadcasting a message that asks for information about the mobile end device's current location. Broadcast operations are expensive in terms of bandwidth and power consumption, particularly when mobile end devices frequently move between different routers' coverage areas.

In many applications of mobile WSNs, e.g., an indoor tour guide, the movement patterns of mobile end devices exhibit certain regularity, because there are explicit paths that visitors with mobile end devices follow. We collect and utilize the data about device movements to construct a topology. With the historical data of node movements, we form a tree topology that is composed of the coordinator and routers to efficiently deliver downlink packets without frequent location tracking overheads. We call the tree topology a *mobility-robust tree*. The objective of the approach is to increase the number of successful data deliveries and thereby reduce the number of broadcasts triggered by the coordinator due to the location changes of the mobile end devices. For ease of presentation, we have the following two assumptions: 1) mobile end devices carry similar amounts of data and 2) there is no preference for delivering data through specific routers. The proposed approach could be implemented without these assumptions by giving end devices and routers different weights.

B. Problem Formulation

The objective of our deployment framework is to increase the downlink data delivery ratio in ZigBee cluster-tree networks by exploiting the nodes' mobility regularity during the tree construction stage. We consider a joint problem of ZigBee router deployment and routing tree construction. In other words, we exploit the mobility regularity in the early stage of node deployment and during the tree construction. We assume that the WSN is in a closed region and the number and locations of the router nodes are to be determined. Although we consider the framework in a 2-D region, extending it to a region of higher dimension is straightforward. For router node deployment, we construct a virtual grid that covers the whole region. Each

[1] The superframe structure is defined in the IEEE 802.15.4-2003 Standard for Information Technology—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).

vertex, i.e., the intersection of lines, on the grid is a candidate location for a router node. The distance d between adjacent grid points is determined based on the particular scenario and application. The smaller the length of d is, the better the precision and performance will be. However, an extremely small d will result in a very large number of grid points, which will increase the computational complexity. Thus, the length of d should carefully be chosen according to the level of precision required by the application. In the framework, a router node can be placed at an arbitrary point in the region, not necessarily at a vertex on the grid. We assume that nodes are placed on the same xy plane. The communication range of the router node is represented by a polygon with an antenna gain profile ANT_g that indicates the different gains at different angles of the antenna and a placement angle degree r_{ant}, $0^\circ \leqslant r_{ant} \leqslant 359^\circ$ that indicates the antenna direction of the router node.

We model the collected movement data of the mobile end devices as a transition matrix based on the grid. Each candidate location (vertex) on the grid is represented by a state in the transition matrix. We use the grid point to approximate a node's location. In particular, if a mobile sensor's nearest candidate location is i at time t, it is assumed that the sensor is in state i at time t. By counting the number of the events of sensors that move away from or toward each candidate location (state), we can then derive the transition probability matrix, M. The matrix is called the *mobility profile*, and the information that it contains is used in the router node deployment stage to perform the following two subtasks: 1) router node placement and 2) coordinator node selection. Router node placement involves finding appropriate positions for ZigBee routers such that the target region is fully covered by router polygons, i.e., each candidate location (grid point) on the grid is covered by at least one router polygon. Because the proposed scheme exploits mobility regularity, it searches for a candidate location through which a large number of mobile end devices frequently pass. In the coordinator node selection task, the scheme selects a router node as the ZigBee network coordinator. The task can manually or automatically be performed. For manual deployment, the coordinator's position must be predefined according to the environment. Our scheme selects the position based on the mobility profile. More specifically, the scheme exploits the property whereby mobile nodes move toward the coordinator (root) of the ZigBee routing tree as frequently as possible; therefore, to select the position for the coordinator, it searches for a point (a transition state) with as many in-events as possible (an in-event of a state occurs when a mobile end device moves into that state).

After the router node deployment phase has been completed, the routing tree construction can be formulated as a graph problem, where a vertex represents an immobile node (i.e., a router), and a directed edge represents a possible transmission link from an immobile node to another immobile node. In other words, a ZigBee network is represented as $G_r = (V_r, E_r)$, where V_r is a set of immobile nodes, and E_r is a set of transmission links in the network. Based on the historical movement data collected from immobile nodes, each directed edge $e_{(u,v)} \in E_r$ is associated with a weight $W_r(e_{(u,v)})$, which represents the total count of transitions of all mobile nodes that move from immobile nodes u to v in the collected data. The weights on the edges are nonnegative. Our method is designed to construct a ZigBee cluster tree T in the bidirected weighted graph G_r. At the edge $e_{(v,u)}$, the node v is the parent of node u. The movements from u to v are in the opposite direction of downlink data forwarding from v to u. To minimize the number of missed data deliveries caused by mobile end-device mobility, the proposed scheme tries to maximize the total number of movements that are in the reverse direction of data-forwarding paths.

We formalize the deployment framework as the mobility-robust ZigBee tree deployment (MRZTD) problem and define the following three subproblems: 1) router node deployment; 2) coordinator selection; and 3) routing tree construction. The overall objective is to minimize the number of missed data deliveries caused by mobile end-device mobility, thereby minimizing the cost of broadcasting incurred by node searching. Next, we formally define the MRZTD problem.

MRZTD problem

Input: An instance comprises a 2-D closed region G with grid points, a mobility profile M, a positive radius r that represents the communication range of the network routers, and two positive constraint integers R_m and L_m.

1. Router node deployment

Instance: A closed region G, a mobility profile M, and an antenna gain profile ANT_g.

Output: A graph $G_r = (V_r, E_r)$ such that G_r is connected and has edge weight $W_r(e_{(u,v)}) \geqslant 0, \forall e_{(u,v)} \in E_r$, where $W_r(e_{(u,v)})$ is derived from the mobility profile M. The union of a polygon with a placement angle degree r_{ant} centered at each $v_r \in V_r$ covers G.

Objective: Minimize the sum of edge weight $W_r(e_{(u,v)})$, $\forall e_{(u,v)} \in E_r$.

2. Coordinator selection

Instance: A graph $G_r = (V_r, E_r)$ with edge weight $W_r(e_{(u,v)}) \geqslant 0, \forall e_{(u,v)} \in E_r$.

Output: A coordinator node $p_r \in V_r$.

Objective: Maximize $\sum W_r(e_{(u,v)}), \forall e_{(u,v)} \in E_r$ such that the edge $e_{(u,v)}$ is directed toward p_r.

3. Routing tree construction

Instance: A graph $G_r = (V_r, E_r)$ with edge weight $W_r(e_{(u,v)}) \geqslant 0, \forall e_{(u,v)} \in E_r$, a coordinator node $p_r \in V_r$, and two positive constraint integers R_m and L_m.

Output: A rooted spanning tree T in G.

Objective: Construct a rooted spanning tree T in G_r such that the sum of all transition probabilities for edges directed toward the root is maximal among all possible trees in G. In addition, the outdegree of every vertex in T should not exceed R_m, and the depth of T should not exceed L_m.

IV. Mobility-Robust ZigBee Tree Topology Deployment

In this section, we present a heuristic algorithm for the deployment of an effective mobility-robust ZigBee tree (MRZT) topology. To better elaborate the problem formulation and the design of the algorithm, we design a simple example. The example will be presented along with the algorithm description in this section. The input comprises a 2-D closed region G with distance d between adjacent grid points, a mobility profile M based on the historical movement data of mobile end devices, the routers' antenna gain profile ANT_g, and the two constraints Rm (the outdegree constraint) and Lm (the tree-depth constraint). The algorithm is implemented in the following three phases: 1) ZigBee node deployment (ZND); 2) ZigBee coordinator decision (ZCD); and 3) ZigBee tree construction (ZTC). The ZND phase determines the number and locations of router nodes, the ZCD phase selects one of the routers as the coordinator, and the ZTC phase constructs an MRZT based on the deployment in the previous two phases.

The pseudocodes of the proposed algorithms are detailed in Algorithms 1, 2, and 3.

A. ZND Phase

In this section, the implementation of the ZND phase using Algorithm 1 is elaborated. In the ZND phase, the algorithm adopts a greedy approach. As previously mentioned, the input parameter set comprises a closed-region graph G, a mobility profile M, and the routers' antenna gain profile ANT_g. We model the transition probability matrix (mobility profile) as a directed edge-weight function on the virtual grid. In particular, if the transition probability from states i to j is p, the weight on the directed edge $e_{(i,j)}$ of graph G is p. The larger weight on an edge indicates a larger likelihood that mobile end devices will move on that edge.

Algorithm 1: ZND.

```
Input: A grid graph $G = (V, E)$, a mobility profile $M$, and the antenna gain profile $ANT_g$
Output: A network graph $G_r = (V_r, E_r)$
 1: $(Q, R_g, R, R_{tmp}, r_{ant}) \leftarrow (E, \emptyset, \emptyset, \emptyset, 0)$ and define an empty graph $G_r = (V_r, E_r)$
 2: Sort($Q$)
 3: Dequeue the maximum-weight edge $e_{(a,b)}$ from $Q$ and add it to $R$
 4: for all $e_{(u,v)} \in Q$ do
 5:     if DiskCover($e_{(u,v)}, R, ANT_g) = TRUE$ then
 6:         Remove $e_{(u,v)}$ from $Q$ and add it to $R_{tmp}$
 7:     end if
 8: end for
 9: $(R, r_{ant}) \leftarrow$ FindMaxAngle($R_{tmp}, ANT_g$)
10: Add $(R, r_{ant})$ to $R_g$ and add $(R_{tmp} - R)$ back to $Q$
11: repeat
12:     $(R, r_{ant}) \leftarrow$ FindMaxPolygonCover($R_g, Q, ANT_g$)
13:     Add $(R, r_{ant})$ to $R_g$
14:     Remove all $e_{(u,v)} \in R$ from $Q$
15: until $Q$ is empty
16: for all $(R, r_{ant}) \in R_g$ do
17:     $p \leftarrow$ MakePolygon($R$)
18:     Add $p$ to $V_r$ with a node weight $r_{ant}$
19:     AddNewRouter($p, G_r, ANT_g$)
20: end for
21: $E \leftarrow$ MakeConnection($V_r, r$)
22: for all $p \in V_r$ such that its degree $= 0$ do
23:     AddNewRouter($p, G_r, ANT_g$)
24: end for
25: return $G_r$
```

The first step of this algorithm (lines 1–10) attempts to find an appropriate location for the first router. At the beginning of this step, the edge with the largest weight is added to R. Here, Q is the set of weights on all the edges in E. We sort Q in nonincreasing order such that the first element in Q is the maximum weight in Q in line 2 (function **Sort**). We denote e as the edge with the maximum weight in Q, dequeue the edge $e_{(a,b)}$ from Q, and add it into an empty set R. Then, we iteratively check the remaining edges in Q one by one between lines 4 and 7. In line 5, we check if the edge $e_{(u,v)}$ in Q and the edges in set R can be covered by a single disk with a radius that is equal to the average of the router's communication range according to the antenna gain profile (function **DiskCover**). In other words, we check if those edges can be covered by the average communication range of one router. If the edge $e_{(u,v)}$ and the edges in set R can be covered by a single disk, we add the edge $e_{(u,v)}$ to set R. After the iteration, no more edges can be added to R under the restriction that the edges are covered by one disk with a radius equal to the average communication range of one router. Now, we decide the router's location as the center of the disk. The remaining task is to decide the angle of the antenna on this router. In line 9, we calculate the sum of edge weight under the router's coverage for each possible angle. We choose the angle with the maximum sum of edge weights to be r_{ant} (the angle of the antenna for this router) based on the design rationale where most of the node movement will be covered by a single router (function **FindMaxAngle**). Finally, we add the couple (R, r_{ant}) to a group set R_g in line 10.

At the second step of the ZND algorithm (lines 11–15), we keep locating more routers until Q is empty. This step ensures that the map is fully covered by the routers' communication range. Here, every router's communication range is at least partially overlapped with another deployed router's communication range such that at most two forwarding routers need to be added for each pair of two routers whose communication ranges are partially overlapped to ensure the communications of these two routers. Under this constraint, we choose the location and the antenna angle with the sum of weights as large as possible (function **FindMaxPolygonCover**).

After the loop has been completed, the loop between lines 16 and 20 marks the position of each router based on the group R_g. We model the center of each disk formed by edge sets in group R_g as a vertex (function **MakePolygon**) and then add the vertex to V_r. In line 21, we form bidirectional edges between each pair of two vertices in V_r (function **MakeConnection**). There will exist an edge $e_{(u,v)}$ in E_r if and only if the two routers represented by vertices u and v are within each other's communication range based on the antenna gain profile. In other words, the edges in E_r represent the transmission links between routers. Then, at the loop in lines 22–24, the algorithm attempts to add additional vertices (forwarding routers) to connect the

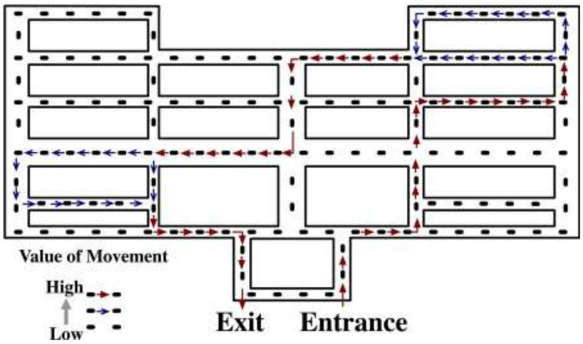


Fig. 2. Example environment.

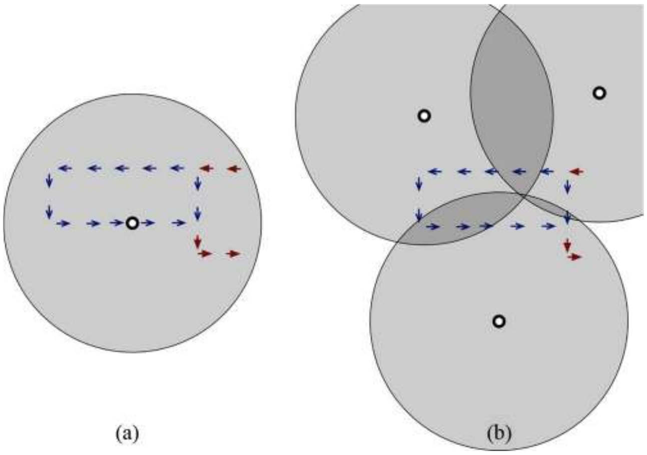


Fig. 3. Router deployments to cover the movement pattern.

disconnected vertices to ensure that the graph G_r is connected (function **AddNewRouter**). Finally, the algorithm outputs the graph G_r (the deployment of ZigBee routers), where V_r is a set of routing nodes, and E_r is a set of transmission links.

We use a simple example to further elaborate the algorithm. Fig. 2 depicts a simple exemplary environment. First, our framework requires the end device's historical movement data to be mapped on the grids. As shown in Fig. 2, each grid point represents a candidate deployment point of the routers. For simplicity, instead of showing the exact value of movement counts, we use the arrow representation in this example. The red wide arrow represents a large number of movement counts, the blue narrow arrow represents a medium number of movement counts, and no arrow means a small or even zero number of movement counts. At this stage, routers are to be deployed based on the principle of pursuing more movements that are covered under a single router. Fig. 3 shows two possible deployments for covering a movement pattern. The deployment in Fig. 3(a) is preferred, because a single router covers most movements instead of multiple routers in Fig. 3(b). The ZND algorithm first finds an edge with the maximum weight. Then, starting from that edge, the ZND algorithm tries to include more edges with as larger weights as possible on the constraint that those edges need to be covered by only one router. Those edges will decide one router's position, and the ZND algorithm repeats the routine until the whole environment has been covered by the routers. Finally, the deployment of routers can be completed by the ZND algorithm in Fig. 4.

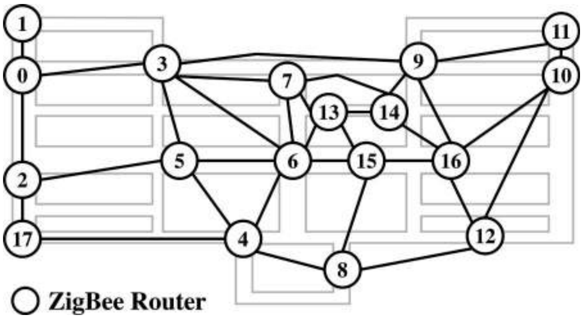


Fig. 4. Deployment of routers to cover the whole environment.

B. ZCD Phase

This section explains the implementation of the ZCD phase using Algorithm 2. Based on the deployment completed in the first phase, the ZCD phase selects one vertex in the region as the root (coordinator) of the routing tree. This phase also builds an edge-weight function based on the mobility profile for the graph G_r. The function will be used to construct the routing tree in the ZTC phase.

Algorithm 2: ZCD.

```
Input: A network graph $G_r = (V_r, E_r)$, a grid graph $G = (V, E)$, and a mobility profile $M$
Output: A vertex $p_r$ and an edge-weight function $W_r$ for graph $G_r$
1: $p_r \leftarrow \emptyset$
2: Define a weight function $W_r$ for graph $G_r$ such that
   $W_r \leftarrow \text{CombineState}(M, G, G_r)$
3: if $(p_r \leftarrow \text{UserAssignRoot}(G_r)) \neq \emptyset$ then
4:     return $(W_r, p_r)$
5: end if
6: Define a weight function $W_{IV}$ such that
7: for all $d \in V_r$ do
8:     $W_{IV}(d) \leftarrow \sum_{s \in V_r} W_r(e_{(s,d)})$
9: end for
10: $p_r \leftarrow \text{FindMax}(V_r, W_{IV})$
11: return $(W_r, p_r)$
```

The input of the algorithm in this phase comprises the graph G_r constructed in the previous phase, the original graph G with a virtual grid, and the mobility profile M. The algorithm outputs the root vertex p_r and the edge-weight function W_r. First, we define the edge-weight function W_r by merging states in the mobile profile M (function **CombineState**). Each state in M has a corresponding vertex in G. Some states are merged into one state if the positions of their corresponding vertices in G are covered by one disk centered at a vertex in G_r. Then, the new mobility profile will comprise the edge-weight function of G_r.

In line 3, we check if the root has been selected (function **UserAssignRoot**). If a vertex has been designated as the root, the algorithm is completed and outputs the root vertex p_r and the edge-weight function W_r. Otherwise, we try to find an

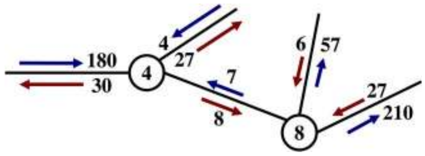


Fig. 5. In- and out-edge weights for routers 4 and 8.

appropriate vertex to be the root to maximize the benefit of our mobility-robust framework. As mentioned in the previous section, the algorithm chooses a vertex with the maximum sum of in-edge weights, where the in-edge weight represents the number of in-events. We define an edge-weight function W_{IV} in lines 7–9, where $W_{IV}(d)$ represents the sum of in-edge weights for vertex d. With the weight function W_{IV}, we find the vertex with the maximum sum of in-edges' weight and assign that vertex as the root in line 10 (function **FindMax**). At the end of this phase, the algorithm outputs the root vertex p_r and the edge-weight function W_r.

For the simple example, the deployment of routers is shown in Fig. 4. The edges between routers represent the communication links and have bidirectional weights. The weight on each edge represents the end-device movement counts from one router's coverage area to another router's coverage area. We intend to choose a router with the maximum sum of in-edge weights as the coordinator. For example, in Fig. 5, the sum of in-edge weights $(180 + 4 + 7 = 191)$ for router 4 is larger than that $(8 + 6 + 27 = 41)$ for router 8. Thus, we will choose router 4 as the coordinator, given these two routers. The ZCD algorithm calculates each router's sum of in-edge weights and chooses the router with the maximum sum of in-edge weights as the coordinator.

C. ZTC Phase

This section explains the implementation of the ZTC phase using Algorithm 3. So far, we have derived the router deployment graph G_r with an edge-weight function W_r that represents the movement tendency of mobile end devices. We have also assigned a vertex as the root. In this phase, we use those results, along with the network constraints R_m and L_m, as the input to construct a single rooted tree T as the ZigBee routing tree.

Algorithm 3: ZTC.

```
Input: A network graph $G_r = (V_r, E_r)$ with $W_r$ and $p_r$, $R_m$, and $L_m$
Output: A ZigBee routing tree $T$
 1: Define a tree $T = (V_T, E_T)$ whose root node is $p_r$
 2: repeat
 3:     $Q \leftarrow \emptyset$
 4:     for all $e_{(u,v)} \in E_r, u \notin T, v \in T$ do
 5:         Add $e_{(u,v)}$ to $Q$
 6:     end for
 7:     Sort($Q$)
 8:     for all $e_{(a,b)} \in Q$ do
 9:         if CheckLegal($e_{(a,b)}, T, R_m, L_m) = TRUE$ then
10:             Remove $e_{(a,b)}$ from $Q$
11:             Add $a$ to $V_T$
12:             Add $e_{(a,b)}$ to $E_T$
13:             break
14:         end if
15:     end for
16: until $|V_r| = |T|$
17: return $T$
```

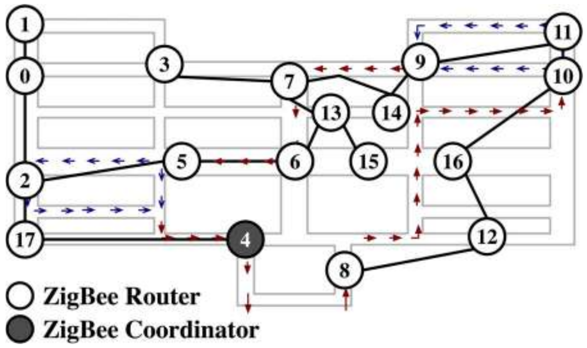


Fig. 6. Final deployment and tree topology.

The algorithm defines a tree and initially takes p_r as the root node. The main loop is in lines 2–16, and the tree is constructed in this loop until the number of vertices in G_r is equal to that in T. First, we put into an empty set Q each directed edge whose source vertex is not in T and destination vertex is in T. Then, in line 7 (function **Sort**), the algorithm sorts Q in nonincreasing order such that the first element in Q is the edge with the largest weight. The iteration in lines 8–15 sequentially adds the edges with the largest weights to T while maintaining the tree's legality. In line 9, we check if T is still legal after the edge $e_{(a,b)}$ in Q has been added to T (function **CheckLegal**). If T is legal, we add the edge $e_{(a,b)}$ to the tree. Otherwise, we try to add the edge with the second largest weight in Q, and so on, during this iteration. Finally, the algorithm outputs the tree T.

Recall that, for the tree construction, our design rationale is to prefer that the paths of downlink data delivery and end-device movement patterns are as close as possible and in reverse direction. To achieve this goal for the simple example, the ZTC algorithm constructs the ZigBee routing tree that includes only the coordinator at first. The edge with the maximum weight among all edges directed at the coordinator will then be selected to be included as part of the tree. Then, the edge with the maximum weight among all remaining edges directed at the tree will be chosen to be included, and so on, until all routers are connected by the tree, as shown in Fig. 6. We can observe in Figs. 6 and 7 that the tree has the tendency to "grow" along the end-device movement paths (the arrows), and the directions of downlink data delivery and end device movement paths tend to be close and in reverse direction.

D. Complexity Analysis

In this section, we discuss the complexity of the proposed algorithms. The initial steps in Algorithm 1 take $O(|E|)$ time, and the sorting procedure takes $O(|E| \times \log |E|)$ time. For the first loop in lines 4–7, the **DiskCover** function must perform

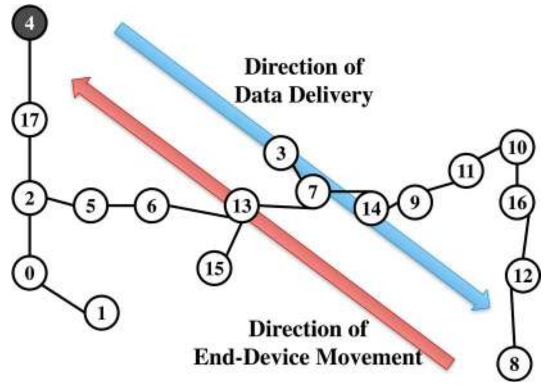


Fig. 7. Tree structure.

$O(|E|^2)$ operations in the worst case to check if the edges are covered by a single disk. In particular, it checks the distance between each pair of edges. Thus, this loop takes $O(|E|^3)$ time. Then, for the **FindMaxAngle** function, we assume that the angle of an antenna can precisely be set to the degree. Thus, the **FindMaxAngle** function takes at most $360 * O(|E|)$ time to find the best angle, even with using brute-force search. The function **FindMaxPolygonCover** performs $O(|E|^3)$ operations for edge coverage checking on each possible angle. Thus, the second loop takes $O(|E|^4)$ time. The second loop performs at most $O(|E|)$ operations to enable the **MakePolygon** function to calculate the center point of each polygon. The **MakeConnection** function performs $O(|V_r|^2)$ operations to connect each pair of vertices in V_r. Then, the **AddNewRouter** function takes $O(|V_r|^2)$ time to connect the disconnected vertices. Thus, the loop in lines 22–24 takes $O(|V_r|^3)$ time. Because $|V_r|$ should be less than $|E|$, the worst case time complexity of the ZTG phase is $O(|E|^4)$.

In Algorithm 2, the **CombineState** function performs $O(|V|)$ operations to merge the states. The W_{IV} function assignment in lines 7–9 performs $O(|V_r|)$ operations, and the *FindMax* function performs $O(|V_r|)$ operations to achieve the maximum. Thus, the worst case time complexity of the ZCD phase is $O(|V|)$, because $|V|$ is larger than or equal to $|V_r|$.

In Algorithm 3, the main loop in lines 2–16 has two inner loops. The first inner loop takes $O(|E_r|)$ time, and the second inner loop performs $O(|E_r|)$ operations in the worst case, because the **CheckLegal** function only takes constant time. The sorting procedure in the main loop takes at most $O(|E_r| \times \log |E_r|)$ time. Hence, the ZTC phase takes $O(|V_r| \times |E_r| \times \log |E_r|)$ time.

The overall time complexity of a complete process, including the three phases, is $O(|E|^4)$, because $|E|$ is at least equal to or larger than $(|E_r|)$ and $(|V|)$ for grids with more than two rows and two columns.

V. Simulations and Results

In this section, we discuss the simulations that we conducted and consider the insights gained on mobility-robust tree topologies for ZigBee wireless networks. The proposed framework is used to deploy routers in a real-world indoor scenario with a custom-made mobility model and a real-world outdoor scenario with a real data set of human mobility trace so that the coverage and connectivity constraints[2] can be satisfied. We use the ns-2 simulator, with the parameter settings specified in the ZigBee standard [6], to evaluate the performance of the proposed approach.

A. Simulation Setup

The two scenarios used for the performance evaluation are the Taipei World Trade Center's Nangang Exhibition Hall and the Magic Kingdom, which is one of four theme parks at the Walt Disney World Resort, Orlando, FL. The layout of the two scenarios is shown in Figs. 8 and 9, respectively.

We use the Ground Level Exhibition Hall to evaluate the performance of our framework for indoor applications, e.g., trade shows or museum tours. The hall's dimensions are approximately 200 × 150 m, and there is 22 680 m^2 of exhibition space. The arrangements of the booths, shown in Fig. 8(a), are based on a real business exhibition. There are four recommended touring paths in the exhibition hall, as shown in Fig. 8(b) and (c). The paths can be designated by exhibition organizers, or they may simply be the paths frequently used by a large number of people who carry mobile end devices. It is assumed that people who carry mobile end devices will follow the recommended tour path. The mobility profile that is used by the framework in the simulation of this scenario is based on a 2-h trace in which 100 people with mobile end devices move at an average speed of 1.5 m/s in the exhibition hall. There is an 80% probability that people will follow the recommended tour paths.

To evaluate the performance of our framework in outdoor applications such as the World Expo or amusement parks, we selected the Magic Kingdom, Disney World (Orlando), as shown in Fig. 9(a). The red lines on the map depict the boundary of the coverage area of the network at the park. The area of the park is about 433 014 m^2, which is nearly 20 times larger than the Nangang Exhibition Hall. Instead of the recommended tour paths, for this scenario, we adopt a real human mobility data set collected by Rhee *et al.* at North Carolina State University (NCSU) [39]. The Disney World traces were obtained from 19 volunteers (41 traces) spending their holidays in various theme parks in the Disney World. We only pick the traces by those visiting the Magic Park (15 traces) for the simulation. Fig. 9(b) shows one sample trace for the Disney World scenario.

We use F-shaped irregular antennas in the simulations, but other types of antennas can easily be adapted. The radiation pattern of the F-shaped antenna (embedded in the Chipcon CC2420 RF transceiver [40]) used in the simulations is measured by Ansoft HFSS [41]. The ns-2 simulator assumes that nodes are placed on the same xy plane. The average and the standard deviations of the gain values (after normalization) on the selected horizontal plane are 0.69 and 0.19, respectively. The average communication range is about 48 m.

[2] The constraint on coverage ensures the coverage of any spot in the deployment space by at least one router, and the constraint on connectivity ensures the existence of a direct or indirect way for any spot to connect with the root.

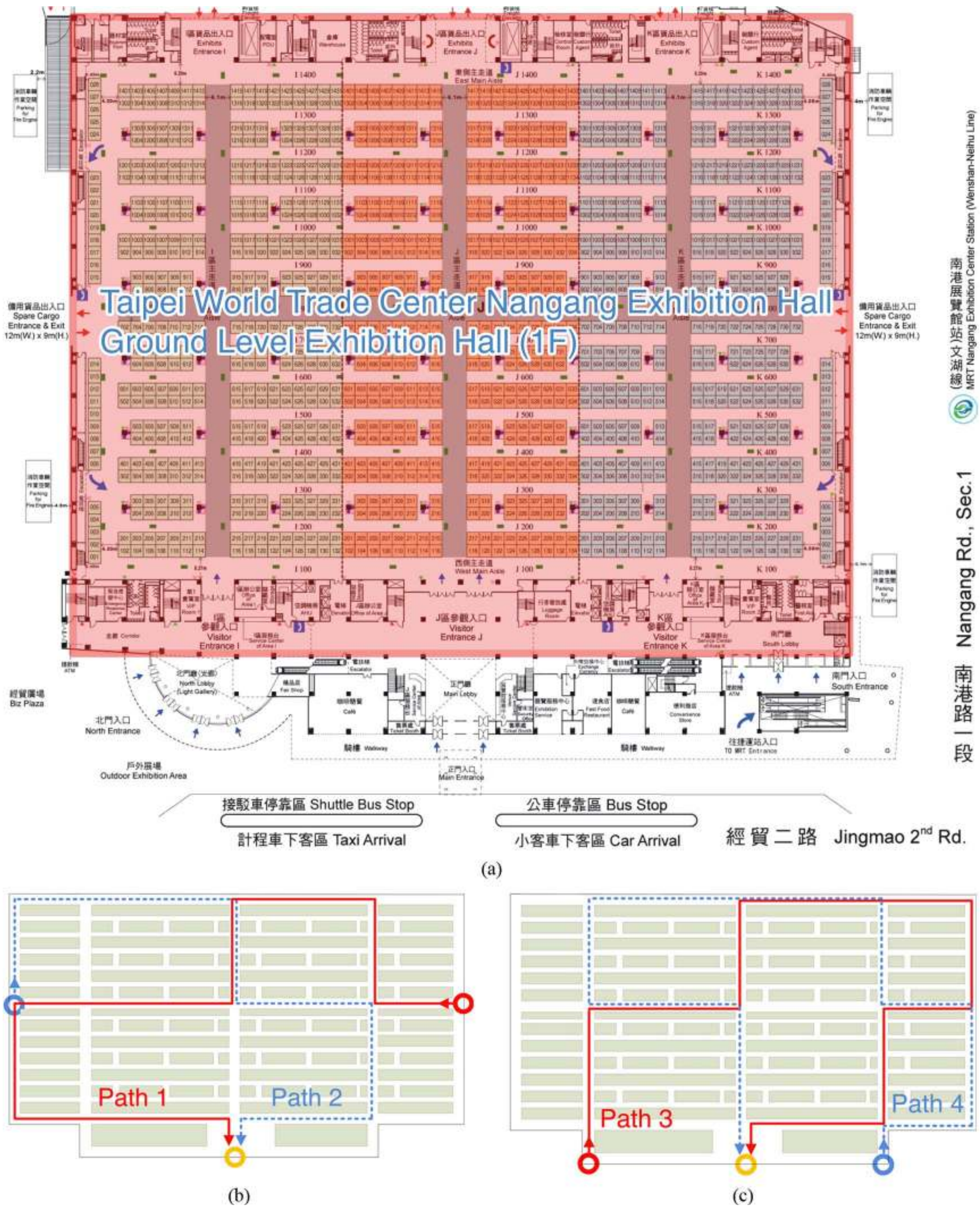


Fig. 8. Taipei World Trade Center's Nangang Exhibition Hall. (a) Floor plan. (b) Recommended paths 1 and 2. (c) Recommended paths 3 and 4.

In each scenario, we compare the performance of our framework with that of the minimum spanning tree (MST) approach in [42] and ad hoc routing by the ZigBee standard. Recall that the proposed scheme utilizes an overhearing technique. We test the MST with three routing mechanisms. The first mechanism uses the proposed overhearing technique, the second mechanism uses the original routing mechanism specified by the ZigBee standard without flooding, and the third mechanism is the original routing mechanism with flooding. Flooding is made by the coordinator when a data delivery fails and the network loses track of the target mobile end device. The locations of the coordinator and routers in our routing tree and in ad hoc routing are determined by the proposed approach. For the MST, the locations of the coordinator or routers and the connections between them are determined by the deployment tool proposed in [18] and [19]. The tool uses the minimum number of nodes to cover the space and executes a breadth-first-like algorithm to derive a feasible tree topology.

To evaluate the aforementioned routing approaches, we utilize the following four performance metrics:

1) packet delivery ratio;
2) one-way delay;
3) routing overhead;
4) effective path duration.

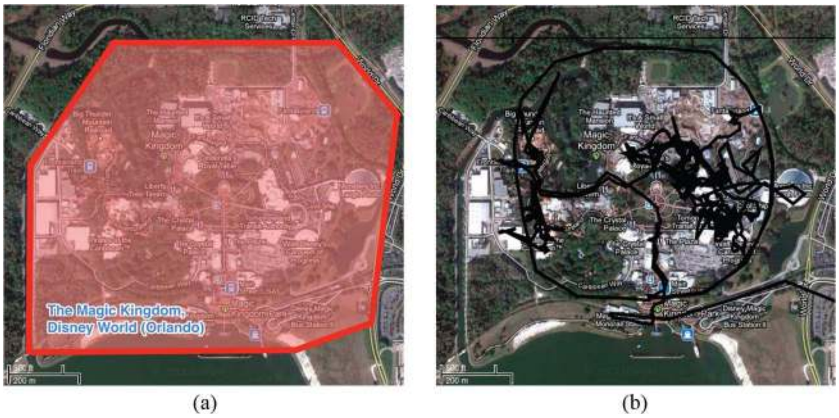


Fig. 9. Magic Kingdom, Disney World (Orlando). (a) Map. (b) One sample trace.

The packet delivery ratio is the ratio of the number of data packets that were successfully delivered to mobile end devices over the number of data packets that should be delivered. The one-way delay is the average time that a packet from the coordinator takes to travel to the target mobile end device. The routing overhead is the ratio of the number of packets used to exchange routing information to the total number of packets in the network. The effective path duration, which is conceptually similar to the path duration in [43], is defined as the average time for a mobile end device to stay on the same branch of the tree, where the packet delivery to the end device can be successful by overhearing. This metric is used to assess the characteristics of different tree topologies in the indoor scenario, because for the outdoor scenario, we focus on verifying the performance on the real human traces.

In each topology, every mobile end device is expected to receive a 70-kB packet dispatched by the coordinator under a *Poisson distribution*, where the mean interpacket interval is set at 10 s. We measure the variations of the performance metrics against the *number of mobile end devices*, *the speed of the devices*, the *length of the location update period*, and the *mobility regularity*. The location update period is the time between packets sent by a mobile end device to the coordinator, and the coordinator can derive the current location of the mobile end device from the packets. Mobility regularity is defined as the probability that a mobile end device will continue on the recommended path when it encounters a path intersection. If the device deviates from the recommended path, it randomly selects one of the other paths at the intersection with equal probabilities. For the indoor scenario, all four control factors, except the *length of the location update period*, are adopted, because intuitively, the effective path period is irrelevant to the location update period. For the outdoor scenario, because the mobility trace data set is given, we only measure the performance metrics against the *length of the location update period*. Although the impacts of the aforementioned parameters of interest are assessed, the default settings of other parameters are set as follows. The number of mobile end devices is 150. The speed of each device is 1.5 m/s. The location update period is 240 s, and the mobility regularity is 80%. The parameter settings in the ns-2, based on the standard ZigBee specifications [6], are listed in Table I. The unlisted specifications are set as default values in ns-2. The results are derived by averaging ten simulations, each of which runs for 7200 s. To ensure that the comparisons were fair, the three topologies were evaluated on the same data set.

TABLE I
PARAMETER SETTINGS IN ns-2

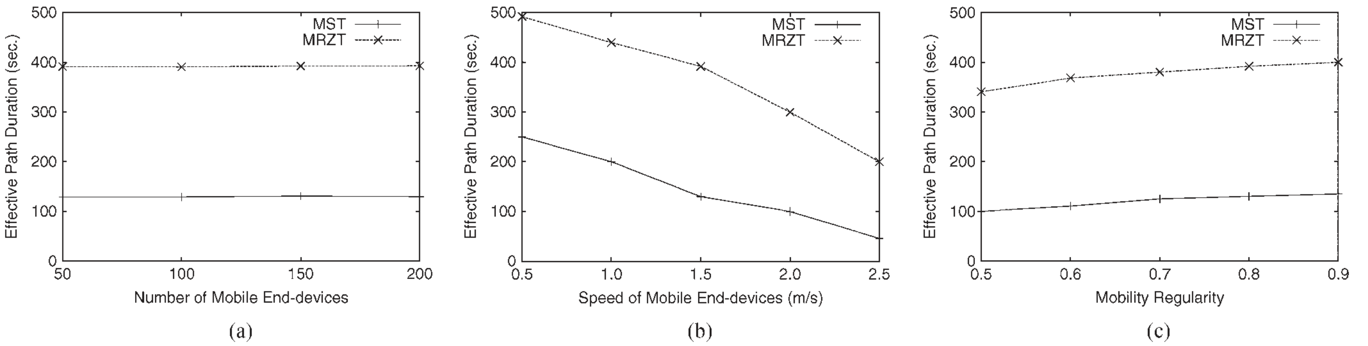
Specification	Setting
Network Standard	IEEE 802.15.4
Antenna Type	F-shaped Antenna
Frequency	2.4GHz ISM Band
Data Rate	250Kbps
Media Access Control	CSMA/CA
Propagation Model	TwoRayGround
Transmission Power	0dBm
Receiver Sensitivity	-94dBm
Communication Range	48m
Carrier Sense Range	120m

B. Results and Observations

In this section, we present the numerical results of the simulations and also consider the implications of the results. The abbreviations for the routing tree and routing mechanisms are given as follows: 1) **MRZT**, the tree that is generated by our framework using the proposed overhearing technique; 2) **MST**; and 3) ad hoc routing with the ZigBee specification (**Ad Hoc**). The routing mechanisms adopted by the MST are Original ZigBee tree routing with the ZigBee specification (**Ori**), original ZigBee tree routing with flooding (**Ori Flooding**), and mobility-robust routing (**MRR**; using the overhearing technique proposed in this paper).

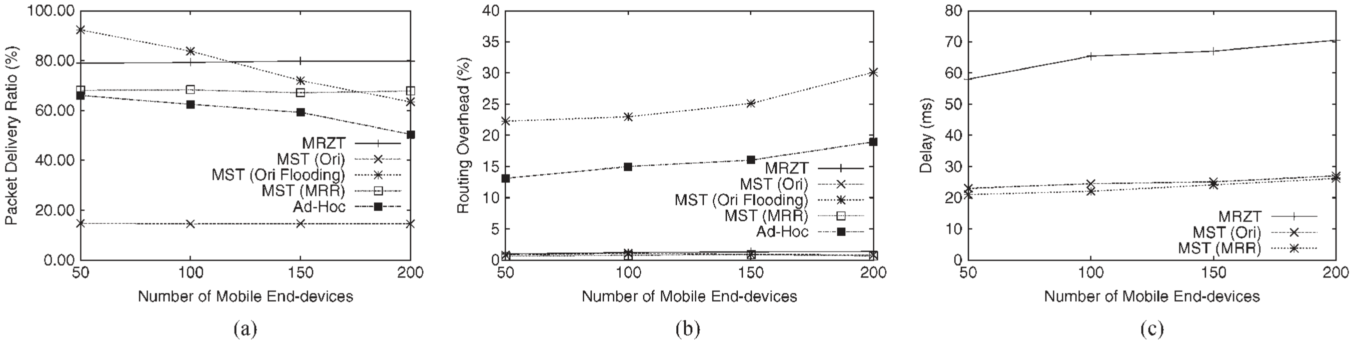
1) Case 1—Indoor Environment Scenario: In this section, we consider a scenario in an indoor environment (the Nangang Exhibition Hall). The tree topologies discussed in this case are the MRZT determined by our framework for multiple recommended touring paths and the MST.

Fig. 10(a) shows the relationship between the effective path duration and the number of nodes. We observe that the effective path period remains the same, despite the change in the number of nodes. Intuitively, the effective path period is irrelevant to the number of mobile end devices, and the latter has a negligible impact on the effective path duration. The graph in Fig. 10(b)



(a) (b) (c)

Fig. 10. Effective path duration for different tree topologies. Default values of the parameters are given as follows: the number of mobile end devices is 150, the speed of each device is 1.5 m/s, and the mobility regularity is 80%. (a) Effective path duration versus the number of nodes. (b) Effective path duration versus the speed of mobile end devices. (c) Effective path duration versus mobility regularity.



(a) (b) (c)

Fig. 11. Impact of the number of mobile end devices (the speed of each device is 1.5 m/s, the location update period is 240 s, and the mobility regularity is 80%). (a) Packet delivery ratio versus the number of mobile end devices. (b) Routing overhead versus the number of mobile end devices. (c) Packet delivery delay versus the number of mobile end devices.

shows that the effective path duration decreases as the mobile node's speed increases. This is because mobile end devices more frequently change with respect to the communication ranges of different routers when they move faster. Fig. 10(c) shows the impact of the mobility regularity on the effective path duration. The mobility regularity measures the probability that a mobile node will follow the recommended path. Again, the MRZT is generated by using the historical data (mobility profile) of 80% mobility regularity. The effective path duration increases as the mobility regularity increases. We find that the MRZT optimized by 80% mobility regularity yields higher effective path duration in different levels of mobility regularity, even when the latter is as low as 50%. This is because mobile end devices tend to remain on the branch of the MRZT that utilizes the mobility profile.

The aforementioned analysis shows that the MRZT always yields higher effective path duration than the conventional MST when there are changes in the environment such as the number of nodes, the speed of mobile nodes, the mobility regularity, and the location update period. Thus, we can expect that the MRZT will achieve better performance than the MST with or without the proposed overhearing mechanism. In the following discussion, we evaluate the performance differences between the MRZT and MST using different routing mechanisms (Ori, Ori Flooding, and MRR), and the ad hoc routing protocol without the tree topology.

Fig. 11(a)–(c) shows the impact of the number of mobile end devices on the packet delivery ratio, routing overhead, and packet delivery delay, respectively. In Fig. 11(a), we observed that, except for ad hoc routing, the delivery ratio hardly changes as the number of mobile end devices increases. Under ad hoc routing, the delivery ratio slightly decreases as the number of mobile end devices increases. This finding implies that ad hoc routing is more sensitive to the number of mobile end devices than tree routing. The MST (Ori Flooding) achieves a high packet delivery ratio when the number of devices is small, but the ratio decreases as the number of devices increases, because the *Device Discovery* procedure ensures that data will be delivered by broadcasting messages to locate a displaced end device, but because we only count the very first flooding message as the successful one, when the number of devices is large, the flooding messages cause serious network congestion, which results in a low packet delivery ratio. Regardless of the number of mobile end devices, the MRZT achieves a higher delivery ratio than the conventional MST (Ori) and MST (MRR), and the ratio of the MRZT is higher than the MST (Ori Flooding) when the number of devices is large. This demonstrates that node deployment and tree construction under the proposed framework have a positive impact on the network performance. The results of the routing overhead versus the number of mobile end devices in Fig. 11(b) show that the MST (Ori Flooding)'s routing overhead is much higher than the other

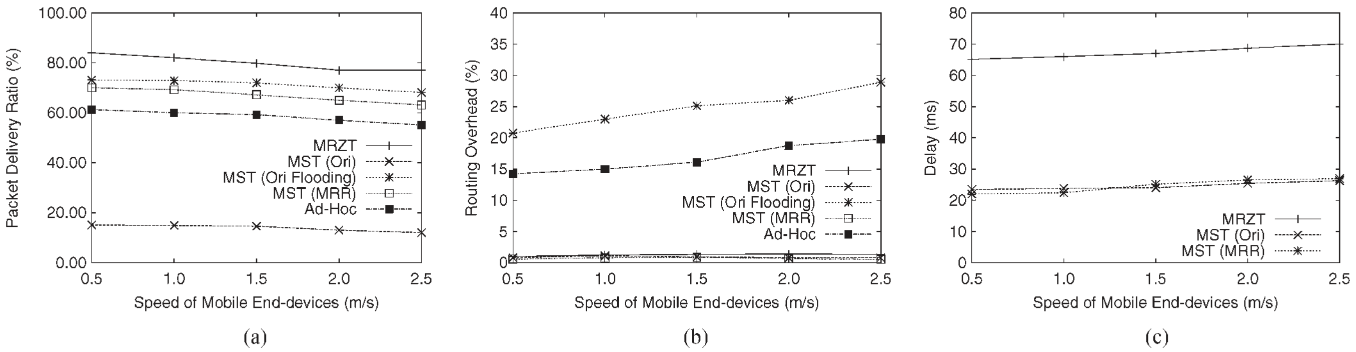


Fig. 12. Impact of the speed of mobile end devices (the number of mobile end devices is 150, the location update period is 240 s, and the mobility regularity is 80%). (a) Packet delivery ratio versus the speed of mobile end devices. (b) Routing overhead versus the speed of mobile end devices. (c) Packet delivery delay versus the speed of mobile end devices.

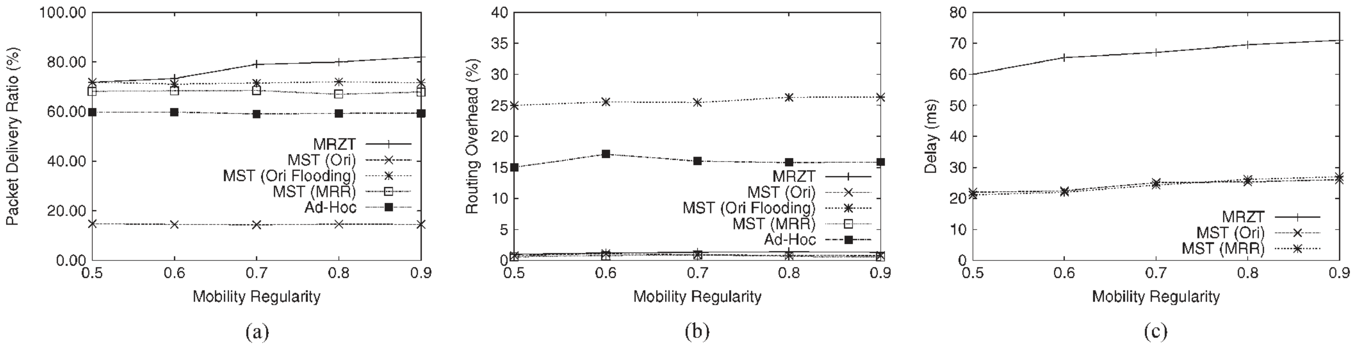
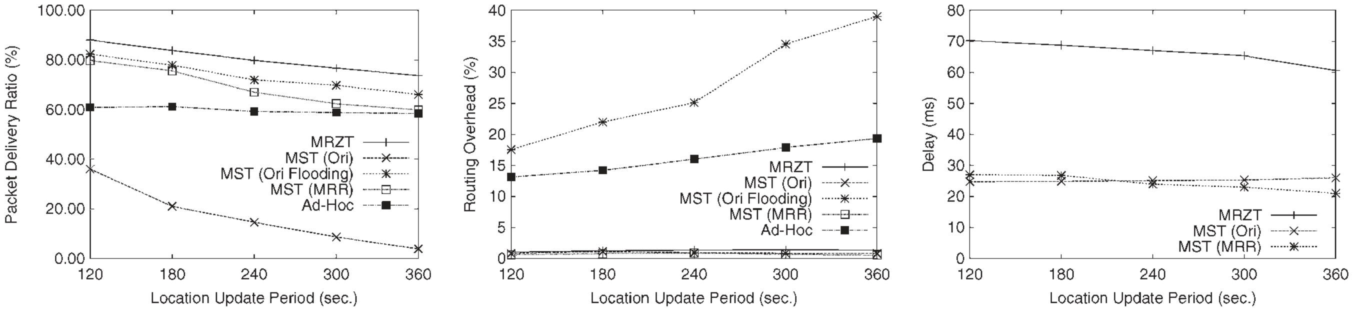


Fig. 13. Impact of mobility regularity (the number of mobile end devices is 150, the speed of each device is 1.5 m/s, and the location update period is 240 s). (a) Packet delivery ratio versus mobility regularity. (b) Routing overhead versus mobility regularity. (c) Packet delivery delay versus mobility regularity.

tree routing mechanisms and it is even higher than the ad hoc routing scheme, which has higher routing overhead than the tree routing protocols. Moreover, the routing overhead of the MST (Ori Flooding) increases as the number of mobile end devices increases, which means that the MST (Ori Flooding) would be unaffordable if the network was congested with mobile end devices. We also find that the tree routing schemes used by the MRZT, MST (Ori), and MST (MRR) have very low routing overhead and they are not affected by the number of mobile end devices. Fig. 11(c) shows the one-way packet delivery delay versus the number of mobile end devices on the MRZT, MST (Ori), and MST (MRR). The ad hoc routing and MST (Ori Flooding) schemes are not compared, because we focus on the impact of different tree construction approaches on the packet delivery delay. We observed that the MRZT incurs longer delay than the MST (Ori) and MST (MRR). The reason is that our framework tends to construct a tree with fewer branches and as much depth as possible, which is acceptable for non-time-critical scenarios such as museum touring and other recreational applications. The delay of the MST (MRR) is slightly lower than the MST (Ori), because mobile end devices may catch packets in the middle of the delivery path. The delay slightly increases for all trees as the number of mobile end devices increases due to network congestion. The results in Fig. 11(a)–(c) verify the robustness of the proposed MRZT scheme under various network sizes.

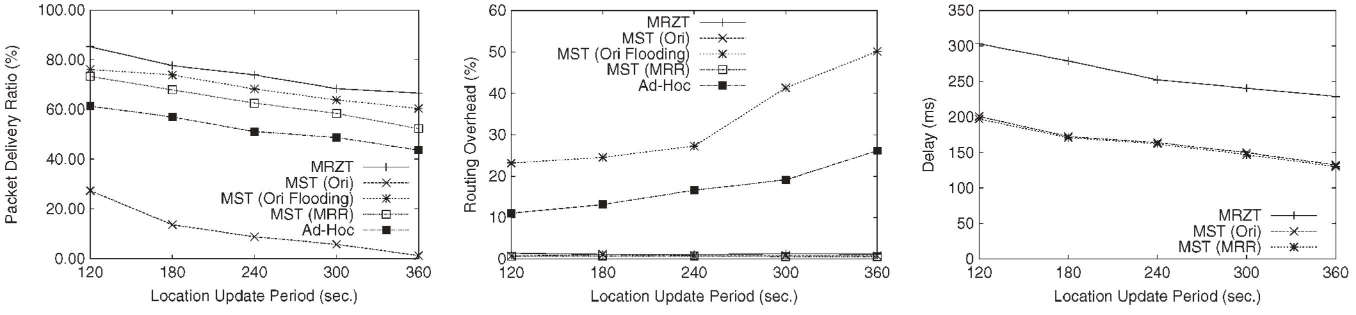
Fig. 12(a)–(c) shows the impact of the speed of end devices on the packet delivery ratio, routing overhead, and packet delivery delay, respectively. The delivery ratio decreases as the speed increases, as shown in Fig. 12(a). This is because mobile end devices more frequently change with respect to the communication ranges of different routers when they move faster, and the frequent location changes cause the route information to more quickly become stale; therefore, the probability of successful packet delivery is lower. We observed that the proposed MRZT scheme achieves a higher delivery ratio than the other schemes, regardless of the speed of mobile end devices. Fig. 12(b) shows that the MST (Ori Flooding) still has the highest routing overhead. Unlike the other trees, MST (Ori Flooding) and ad hoc routing are sensitive to changes in speed. The MRZT incurs longer delay than the MST (Ori) and MST (MRR), as shown in Fig. 12(c); however, the delay increases for all trees as the speed increases. The reasons are the same as those mentioned in the discussion for Fig. 11.

Fig. 13(a)–(c) show the impact of mobility regularity on the packet delivery ratio, routing overhead, and packet delivery delay, respectively. The MRZT is generated by using the mobility profile of 80% mobility regularity. The results in Fig. 13(a) demonstrate that the MRZT is sensitive to changes in mobility regularity. It still achieves a higher delivery ratio than the conventional MST and ad hoc routing schemes for various levels of mobility regularities. Fig. 13(a) also shows that



(a) (b) (c)

Fig. 14. Impact of the location update period (the number of mobile end devices is 150, the speed of each device is 1.5 m/s, and the mobility regularity is 80%). (a) Packet delivery ratio versus location update period. (b) Routing overhead versus location update period. (c) Packet delivery delay versus location update period.



(a) (b) (c)

Fig. 15. Impact of the location update period. (a) Packet delivery ratio versus location update period. (b) Routing overhead versus location update period. (c) Packet delivery delay versus location update period.

the MRZT optimized by 80% mobility regularity can maintain higher packet delivery ratios under different levels of mobility regularity, even when the latter is as low as 50%. This is because mobile end devices tend to remain on the branch of the MRZT where packets can be delivered to the mobile node or overheard. For the routing overhead versus mobility regularity in Fig. 13(b), the trends for all trees and ad hoc routing are the same as those mentioned in the discussions for Figs. 11 and 12. The changes in mobility regularity do not have a significant impact on the routing overheads of different routing approaches. Fig. 13(c) shows that, for the MRZT and MST (MRR), the delay slightly decreases as the mobility regularity increases, because the proposed overhearing mechanism allows mobile end devices to catch packets on the delivery path as they move.

Fig. 14(a)–(c) shows the impact of the location update period on the packet delivery ratio, routing overhead, and packet delivery delay, respectively. The delivery ratio decreases as the location update period increases, as shown in Fig. 14(a). This is because longer location update periods increase the chances that the location information will be out of date, resulting in more failed deliveries. Regardless of the length of the location update period, the proposed MRZT scheme achieves a higher delivery ratio than the conventional MST and ad hoc routing schemes. Fig. 14(b) shows that the overheads of the MST (Ori Flooding) and ad hoc routing increase as the location update period increases. This is because a longer location update period increases the possibility of delivery failures; therefore, the schemes' recovery mechanisms will more frequently be triggered. Finally, Fig. 14(c) shows that the delay of the MRZT and MST (MRR) slightly decreases as the location update period increases. The reason is that mobile end devices have more opportunities to catch packets on the delivery path as they move.

2) Case 2—Outdoor Environment Scenario: In this case, we consider a scenario in an outdoor environment, i.e., the Magic Kingdom, Disney World (Orlando) shown in Fig. 9. The park is much larger than the exhibition site in the previous scenario.

We study the performance differences of the MRZT and MST under different routing mechanisms (Ori, Ori Flooding, and MRR) and the ad hoc routing protocol without the tree topology. As previously mentioned, we only investigate the impact of different location update period, but by comparing the packet delivery ratios, routing overheads, and the packet delays of different tree construction and routing mechanisms, we can observe the advantages and performance gains of the proposed MRZT in the outdoor scenario with a real human mobility data set.

Fig. 15(a)–(c) shows the impact of the location update periods on the packet delivery ratio, routing overhead, and packet delivery delay, respectively. The delivery ratio decreases as the location update period increases, as shown in Fig. 15(a). The trends are the same as those in the indoor scenario; for all routing schemes, the variation is larger than in the indoor scenario. Fig. 15(b) shows that the overheads of the MST (Ori Flooding) and the ad hoc routing scheme also increase as the location update period increases and the overheads of

TABLE II
OVERALL PERFORMANCE COMPARISON

Metrics\Approaches	**MRZT**	**MST (Ori)**	**MST (Ori Flooding)**	**MST (MRR)**	**Ad-Hoc**
Packet Delivery Ratio	Very High	Low	High	Medium	Medium
Routing Overhead	Very Low	Very Low	High	Very Low	Medium
Delay	High	Low	Low	Low	Not Measured

the schemes are higher than in the indoor scenario. Finally, Fig. 15(c) shows that, in contrast to the indoor environment, the delay of the MRZT and MST (MRR) slightly decreases as the location update period increases. We can see that, in such a complex and large-scale scenario with more realistic human mobility traces (compared to the smaller indoor scenario with the recommended touring path probabilistic mobility model), the proposed MRZT still retains a higher packet delivery ratio and lower routing overhead than other tree construction and routing mechanisms.

3) Summary of Insights Gained From the Case Studies: Table II summarizes the overall performance among different approaches based on the comprehensive experiment that we conducted in the previous sections.

As we can observe in Table II, our framework (MRZT) achieves a much higher packet delivery ratio under all circumstances. The MST (Ori Flooding) also achieves a similar delivery ratio, but with much higher routing overhead (up to 30%). In contrast, the MRZT has a merely 0% routing overhead. The MRZT does introduce higher data delivery delay; however, if we compare the delay differences between the MRZT and other methods under indoor and outdoor environments, we can observe that the delay difference is smaller under the outdoor environment (1.5 times) than the indoor environment (2.3 times). This implies that, with larger environmental scale, the impact of our approach on the delivery delay is diminishing. In addition, even with 2.3 times longer delay under the indoor environment, the data delivery delay of the MRZT is still less than 70 ms, which is satisfactor under the voice over Internet Protocol (VoIP) G.114 regulation for one-way transmission duration [44]. With regard to outdoor and larger environments such as Disney World, our simulation results show that the delay is smaller than 320 ms, which is still well-applicable to our targeted non-real-time data delivery applications such as touring information and official announcement delivery [45].

In addition, compared to the indoor environment, all the routing schemes yield a poor performance in the outdoor environment. This may be because the scale of the outdoor network is large and may be challenging for a ZigBee network. In particular, for the MST with flooding and ad hoc routing, the routing overhead becomes very high when the network is relatively large. In contrast, the proposed MRZT achieves a good packet delivery ratio, and its routing overhead is very low. Although the MRZT does not perform as well in the outdoor environment (compared to the indoor environment), it is still the best choice among the compared routing schemes in terms of the packet delivery ratio and routing overhead. Overall, the proposed MRZT scheme strikes an appropriate balance between the packet delivery ratio, routing overhead, and packet delivery delay for ZigBee wireless mobile applications in both indoor and outdoor environments.

In addition, with regard to the complexity of the proposed framework, as shown in Section IV-D, the overall time complexity of the proposed algorithms is $O(|E|^4)$. Although the complexity order may seem slightly high, based on our simulations, even in the complex Disney World scenario, our proposed framework can be conducted within a few hours on a standard PC with an Intel Core i7 (Sandy Bridge) processor. This shows that the proposed framework is feasible for practical target scenarios of reasonable complexity orders.

VI. CONCLUSION

In this paper, we have proposed a scheme that exploits the regularity to improve the data delivery ratio in ZigBee WSNs. The scheme deploys the network nodes and constructs the tree topology by using the mobility regularity imposed by the physical environment. In a ZigBee network, packets are forwarded to mobile end devices through routers. The primary objective of the proposed approach is to deploy the routers and construct a tree topology that enables mobile end devices to move with high probability in the direction of the routing paths. By using the historical movement data of mobile nodes, we construct the tree so that most movements are highly probabilistic to move toward the root, i.e., the opposite direction to downlink transmissions. By enabling mobile end devices to overhear the packets during movement, the data delivery can be completed if the destined mobile end device is located along the path of the data delivery.

The proposed ZigBee routing tree topology deployment and construction framework incorporates the mobility information, and algorithms are developed to implement the framework. Compared to existing approaches, our framework achieves higher data delivery ratios and longer path duration with much lower routing overhead in scenarios where the movements of mobile end devices are with regularity. We used ns-2 to conduct simulations in two real-world scenarios. The simulation results demonstrate the efficacy of the proposed approach.

REFERENCES

[1] W. Chung, P. Hsiu, Y. Shih, A. Pang, Y. Huang, and K. Hung, "Mobility-robust tree construction in ZigBee wireless networks," in *Proc. IEEE ICC*, 2011, pp. 1–6.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM WSNA*, 2002, pp. 88–97.

[5] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 325–349, May 2005.

[6] Z. Alliance, ZigBee Specifications 2006. [Online]. Available: http://www.zigbee.org/

[7] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.

[8] X. Xueliang, T. Cheng, and F. Xingyuan, "A health care system based on PLC and ZigBee," in *Proc. IEEE WiCom*, 2007, pp. 3063–3066.

[9] A. G. Ruzzelli, R. Jurdak, G. M. O'Hare, and P. V. D. Stok, "Energy-efficient multihop medical sensor networking," in *Proc. ACM SIGMOBILE HealthNet*, 2007, pp. 37–42.

[10] J. Lee, C. Chuang, and C. Shen, "Applications of short-range wireless technologies to industrial automation: A ZigBee approach," in *Proc. IEEE AICT*, 2009, pp. 15–20.

[11] M. Khanafer, M. Guennoun, and H. Mouftah, "WSN architectures for intelligent transportation systems," in *Proc. NTMS*, 2009, pp. 1–8.

[12] S. Chumkamon, P. Tuvaphanthaphiphat, and P. Keeratiwintakorn, "The vertical handoff between GSM and ZigBee networks for vehicular communication," in *Proc. ECTI-CON*, 2010, pp. 603–606.

[13] Z. Wu, H. Chu, Y. Pan, and X. Yang, "Bus priority control system based on wireless sensor network (WSN) and ZigBee," in *Proc. IEEE ICVES*, 2006, pp. 148–151.

[14] R. Zhou, C. Zhao, L. Fu, A. Chen, and M. Ye, "ZigBee-based vehicle access control system," in *Proc. IITSI*, 2010, pp. 232–235.

[15] A. Al-Abdallah, A. Al-Emadi, M. Al-Ansari, N. Mohandes, and Q. Malluhi, "Real-time traffic surveillance using ZigBee," in *Proc. ICCDA*, 2010, vol. 1, pp. VI-550–VI-554.

[16] L. Chen, T. Sun, and N. Liang, "An evaluation study of mobility support in ZigBee networks," *J. Signal Process. Syst.*, vol. 59, no. 1, pp. 111–122, Apr. 2010.

[17] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proc. IEEE IECON*, 2007, pp. 46–51.

[18] J. Chang, P. Hsiu, and T. Kuo, "Search-oriented deployment strategies for wireless sensor networks," in *Proc. IEEE ISORC*, 2007, pp. 164–171.

[19] Y. Huang, P. Hsiu, W. Chu, K. Hung, A. Pang, T. Kuo, M. Di, and H. Fang, "An integrated deployment tool for ZigBee-based wireless sensor networks," in *Proc. IEEE/IFIP EUC*, 2008, pp. 309–315.

[20] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Aug. 2002.

[21] Y. Fan, J. Zhang, and X. Shen, "Mobility-aware multipath forwarding scheme for wireless mesh networks," in *Proc. IEEE WCNC*, 2008, pp. 2337–2342.

[22] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. IEEE VTC*, 2005, vol. 4, pp. 2289–2293.

[23] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobility pattern space routing for DTNS," in *Proc. IEEE INFOCOM*, 2006, pp. 1–10.

[24] J. Ghosh, H. Q. Ngo, and C. Qiao, "Mobility profile based routing within intermittently connected mobile ad hoc networks (ICMAN)," in *Proc. IWCMC*, 2006, pp. 551–556.

[25] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 5, pp. 606–621, May 2009.

[26] L. Girod, M. Lukac, V. Trifa, and D. Estrin, "The design and implementation of a self-calibrating distributed acoustic sensing platform," in *Proc. ACM SenSys*, 2006, pp. 71–84.

[27] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Proc. 2nd Int. Conf. MELT GPS-Less Environ.*, 2009, pp. 235–254.

[28] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[29] A. Boukerche, R. Pazzi, and R. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications," in *Proc. MSWiM*, 2004, pp. 157–164.

[30] H. Lee, M. Wicke, B. Kusy, and O. G. L. Guibas, "Data stashing: Energy-efficient information delivery to mobile sinks through trajectory prediction," in *Proc. ACM/IEEE IPSN*, 2010, pp. 291–302.

[31] S. Choudhury, K. Islam, and S. Akl, "A primal–dual approximation algorithm for the minimum cost stashing problem in wireless sensor networks," in *Proc. IEEE IPCCC*, 2010, pp. 146–152.

[32] M. Pan and Y. Tseng, "A lightweight network repair scheme for data collection applications in ZigBee WSNS," *IEEE Commun. Lett.*, vol. 13, no. 9, pp. 649–651, Sep. 2009.

[33] Y. Wang, C. Hu, and Y. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," in *Proc. IEEE WICON*, 2005, pp. 114–121.

[34] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad hoc sensor networks," in *Proc. IEEE INFOCOM*, 2001, vol. 3, pp. 1380–1387.

[35] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. IEEE INFOCOM*, 2003, vol. 2, pp. 1293–1303.

[36] D. Tian and N. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. ACM WSNA*, 2002, pp. 32–41.

[37] S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proc. IEEE WCNC*, 2003, pp. 1609–1614.

[38] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[39] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the levy-walk nature of human mobility," in *Proc. IEEE INFOCOM*, 2008, pp. 924–932.

[40] Texas Instrument Chipcon cc2420. [Online]. Available: http://focus.ti.com/lit/ds/symlink/cc2420.pdf

[41] Ansoft HFSS. [Online]. Available: http://www.ansoft.com/products/hf/hfss/

[42] J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, Feb. 1956.

[43] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy, "PATHS: Analysis of PATH duration statistics and their impact on reactive MANET routing protocols," in *Proc. ACM MobiHoc*, 2003, pp. 245–256.

[44] Y. Chen, T. Farley, and N. Ye, "QoS requirements of network applications on the Internet," *Info. Knowl. Syst. Manag.*, vol. 4, no. 1, pp. 55–176, 2004.

[45] R. Pries, T. Hobfeld, and P. Tran-Gia, "On the suitability of the short message service for emergency warning systems," in *Proc. IEEE VTC*, 2006, vol. 2, pp. 991–995.

Yuan-Yao Shih (S'09) received the B.S. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 2008 and the M.S. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 2010. He is currently working toward the Ph.D. degree in wireless networking with the Department of Computer Science and Information Engineering, National Taiwan University.

He is currently a Research Assistant with the Research Center for Information Technology Innovation, and a Joint Assistant Research Fellow with the Institute of Information Science, Academia Sinica, Taipei. His research interests include wireless personal area networks, wireless body sensor networks, and vehicular ad hoc networks.

Wei-Ho Chung (M'11) was born in Kaohsiung, Taiwan, in 1978. He received the B.Sc. and M.Sc. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles, CA, USA, in 2009. His M.Sc. thesis focused on routing protocols in mobile ad hoc networks.

From 2002 to 2005, he was a System Engineer with the ChungHwa Telecommunications Company, where he worked on data networks. In 2008, he was a Research Intern with Qualcomm Inc., where he worked on code-division multiple acccess systems. Since January 2010, he has been an Assistant Research Fellow with the Research Center for Information Technology Innovation and a Joint Assistant Research Fellow with the Institute of Information Science, Academia Sinica, Taipei. His research interests include communications, signal processing, and networks.

Dr. Chung received the Taiwan Merit Scholarship from 2005 to 2009 and the Best Paper Award at the 2012 IEEE Wireless Communications and Networking Conference.

Pi-Cheng Hsiu (M'10) received the B.S. degree in computer information science from the National Chiao Tung University, Hsinchu, Taiwan, in 2002 and the M.S. and Ph.D. degrees in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 2004 and 2009, respectively.

In 2007, he was a Visiting Scholar with the Department of Computer Science, University of Illinois, Urbana-Champaign, IL, USA. He is currently an Assistant Research Fellow with the Research Center for Information Technology Innovation and a Joint Assistant Research Fellow with the Institute of Information Science, Academia Sinica, Taipei. His research interests include networked embedded systems, real-time systems, and wireless mobile networks.

Dr. Hsiu is a Member of the IEEE Computer Society. He received an Honorary Membership from the Phi Tau Phi Scholastic Honor Society in 2002 and the Ph.D. Dissertation Award from the Institute of Information and Computing Machinery in 2009.

Ai-Chun Pang (SM'09) received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1996, 1998, and 2002, respectively.

In 2002, she joined the Department of Computer Science and Information Engineering (CSIE), National Taiwan University (NTU), Taiwan. She is currently a Professor with CSIE and is with the Graduate Institute of Networking and Multimedia, NTU, and also is a Joint Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica,Taipei.