This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Mobility-Robust Tree Construction in Zigbee Wireless Networks

Wei-Ho Chung and Pi-Cheng Hsiu
Research Center for Information Technology Innovation
Academia Sinica
Taipei, Taiwan 115, R.O.C.
{whc, pchsiu}@citi.sinica.edu.tw

Yuan-Yao Shih, Ai-Chun Pang, Yu-Kai Huang, and Kuan-Chang Hung
Graduate Institute of Networking and Multimedia
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan 106, R.O.C.
{r97005, acpang, d94944009, r96099}@csie.ntu.edu.tw

Abstract

Zigbee, formalized by the IEEE 802.15.4 standard, is a specification for wireless personal area networks with low power, low cost, and a low data rate. In Zigbee, tree topology is commonly practiced to form wireless sensor networks and perform data delivery applications. In Zigbee wireless applications, data delivery failures occur constantly due to the node movements and topology changes of networks. To tackle the topology changes, conventional route reconstruction often involves huge resource consumption. In this paper, we utilize the regularity of mobility patterns to reduce the frequency of route reconstructions and achieve higher efficiency in sending data to mobile nodes. To increase the data delivery ratio, we introduce the metric of mobility-robustness in a tree topology, and propose tree construction with an objective to maximize the mobility-robustness of the constructed tree. We develop an efficient algorithm for effective tree construction. The effectiveness of network topologies constructed using this mobility-robustness metric is demonstrated by NS2 simulations against a real-world scenario.

Keywords Mobility robustness, tree topology construction, Zigbee wireless networks.

1. Introduction

With the maturation of wireless communications and sensing, various sensor-based applications, e.g., tour guiding, industrial automation and health-care monitoring, generate tremendous social benefits. The potential of these benefits has driven extensive studies on wireless sensor networks in past decades [4, 5]. Initiated by the Zigbee Alliance [7], the Zigbee standard specifies the network and application layers for sensing data delivery. Many Zigbee applications require moving objects to be equipped with a sensor and connected to the backbone for data collection and dissemination [16]. Thus, these applicatons need an efficient and automatic approach for locating mobile sensor nodes.

In past decades, various wireless networks, such as cellular networks, wireless local area networks (WLANs), and mobile ad hoc networks (MANETs) have considered the importance of mobility. Many excellent mechanisms/protocols have been proposed for efficient data delivery within such networks, e.g., [6, 9]. However, these excellent research results cannot be applied to the handling of mobility issues for mobile sensor networks because of the unique features of Zigbee, including low power consumption, low data rate, and short communication range. For cellular networks, the *handover* (or handoff) is a typical mechanism [18], but frequent signal strength detection may result in the rapid energy exhaustion of Zigbee nodes [8]. In WLANs, mobile IP (or IP mobility) provides an efficient mechanism for a mobile object to roam across multiple LAN subnets [17], but it usually relies on self-awareness movements and requires mobile objects to rejoin a network and associate with a new based station. Therefore, Mobile IP is more suitable for objects with low-moving frequency. In regards to MANETs, in order to address mobility, [15] proposed a multicast protocol adaptive to dynamic network topologies and resources. However, Zigbee is meant for small amounts of data delivery with a low data rate. The control messages for invoking a route-discovery procedure, as normally used in on-demand routing protocols for MANETs, may congest network traffic. Therefore, to carry out envisioned Zigbee applications, researchers have recently started exploring mobility issues in wireless Zigbee networks. In particular, a numerical study of mobility support was conducted over Zigbee networks [10], and it concluded that Zigbee provisions for mobility are inadequate.

In this paper, to improve the downlink data delivery ratio, we exploit mobility regularity and propose an approach to construct mobility-robust tree topologies in Zigbee wireless networks. Our goal is to construct a topology so that mobile nodes move into its data forwarding path with high probability. The mobile nodes can receive data as long as they are located within the transmission range of any one of the routers on the forwarding path. To achieve this goal, we gather movement information on the environment and introduce the *mobility-robustness* of a tree topology with respect to the moving tendency of nodes. Tree construction is formulated as a graph optimization problem, and we propose an efficient algorithm for effective tree construction. NS2 [3], incorporated within our network deployment tool [12], demonstrates the effectiveness of the network topologies constructed by the proposed algorithm in light of mobility-robustness.

The rest of this paper is organized as follows: Section 2 describes the system model and problem formulation. Section 3

This work was partly supported by National Science Council of Taiwan, under grant number 99-2218-E-001-004 and 99-2219-E-002-029. This work was also supported in part by the Excellent Research Projects of National Taiwan University under Contract 99R80304, the National Science Council under Contract NSC98-2221-E-002-200-MY3, the Institute of Information Industry, and Intel.

presents the proposed algorithm for mobility-robust tree topology construction. Simulation results and analysis are reported in Section 4. Section 5 concludes this work.

2. System Model and Problem Definition

2.1 System Model

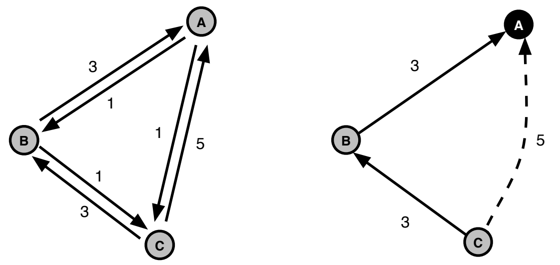
In a Zigbee network, three device types are defined: the Zigbee coordinator, Zigbee router, and Zigbee end device. A Zigbee network is formed by one Zigbee coordinator and multiple Zigbee routers/devices. A Zigbee coordinator performs the initializing, maintaining, and controlling functions for the network. A Zigbee router routes the data among end devices and the coordinator.

Our system model is based on a Zigbee cluster-tree network, which includes a coordinator, routers, and mobile end devices. We choose cluster-tree topology because it supports a superframe structure that achieves power saving and supports a very light-weight routing protocol without routing table maintenance. The coordinator acts the same as in a conventional Zigbee cluster-tree network: the difference being its operation on mobile end devices. Unlike address assignment in a conventional Zigbee network, every mobile end device in our network is randomly assigned a unique address different from those pre-allocated for the coordinator and routers. Consequently, tree construction is not subject to the constraint on the maximum number of children of a router/the coordinator (C_m), but only constraints on the maximum number of child routers of a router/the coordinator (R_m) and the depth of the network (L_m). In order to adapt to quick topology changes, there is no association mechanism between mobile end devices and routers. Instead, when having a packet to send, a mobile end device simply sends it. Upon receiving a packet sent by a mobile end device, a router helps to forward the packet to its parent (in the tree) if the packet has yet to be forwarded. We assume that the coordinator maintains the location of the mobile end device when it sends an uplink data packet to the coordinator. When a downlink packet is destined to a mobile end device, the coordinator deliveries the packet by using the previously recorded location (i.e., the last router receiving the packet from the mobile end device) as the destination. Upon receiving the downlink packet, the router then simply sends it out, and expects to receive an acknowledgement from the mobile end device. If the destined mobile node has already moved out of its previous location, the data delivery fails, and the system searches for the mobile node by sending a broadcast message to inquire its current location. This broadcasting operation will result in significant cost in bandwidth and power consumption, particularly when mobile end devices frequently moves among routers.

For many applications in mobile WSNs, e.g., museum guiding, the moving patterns of mobile end devices follow certain regularity, since these devices follow some explicit guidelines. We collect and utilize data on device movements for the topology formation. With these historical data on node movements, we form a tree consisting of the coordinator and routers to efficiently deliver downlink packets without frequent location tracking overheads. The rationale behind the approach is to increase data delivery success, and consequently reduce the amount of broadcasting triggered by the coordinator due to the location changes of the mobile end devices. For the simplicity of presentation, we assume that every mobile end device has a similar amount of data traffic and does not prefer to receive its data at some specific router, although our approach could be applicable without this assumption by giving end devices/routers different weights.

2.2 Problem Formulation

Mobility-robust tree construction can be formulated as a graph problem, in which a vertex represents an immobile node, i.e., the coordinator or a router, and a directed edge represents a possible transmission link from one immobile node to another. That is, a Zigbee network is represented as $G = (V,E)$, where V is a set of immobile nodes and E is a set of transmission links in the network. With the movement historical data collected among immobile nodes, each edge $e = (u,v) \in E$ is associated with a weight, $W(e)$, which represents the number of transitions of all mobile nodes moving from the transmission range of immobile node u to that of v in the collected data. For any directed edge $e = (u,v)$, there exists a directed edge $\overline{e} = (v,u) \in E$ in the reverse direction. The weights of these edges are non-negative. Our method is to construct a Zigbee cluster tree T in the bi-directed weighted graph G. In the edge $\overline{e} = (v,u)$, the node v is the parent of node u. Movements from u to v follows the upstream of down-link data forwarding from v to u. Our objective is to maximize the total counts of movements toward the upstream of data forwarding paths, so that we can minimize missed data deliveries caused by mobile device mobility. To achieve this objective, we define the *mobility-robustness* of the constructed tree T as the sum of the weights in the chosen directed edges, $W(e), \forall \overline{e} \in T$ and the weights of those unselected edges, $W(e), e \notin T$, which connect all descendant-to-parent pairs of vertices in the same branch in T. We illustrate the mobility-robustness of a tree by Figure 1, where the 1(a) is the input graph and 1(b) is the output spanning tree with root A. Although the edge (C,A) is not chosen as a part of the spanning tree, the movements from C to A are in the direction toward ancestors. We formalize the tree construction problem as the Maximum Mobility-Robustness Zigbee Tree (MRZT) Problem, described below.



(a) A connectivity graph

(b) A spanning tree with mobility-robustness 11

Figure 1. An illustration of the mobility-robustness of a tree

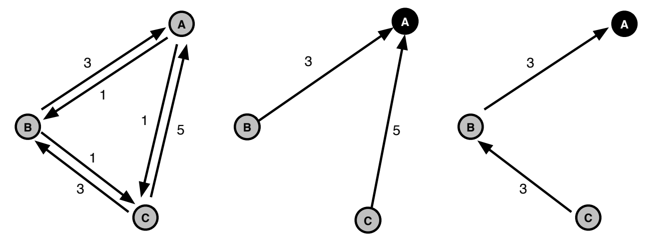
The Maximum Mobility-Robustness Zigbee Tree Problem
Instance: An instance includes a bi-directed graph $G = (V,E)$ with edge weight $W(e) \geqslant 0, \forall e \in E$, and two positive constraint integers R_m and L_m.
Objective: The objective is to find a rooted spanning tree T in G such that the mobility-robustness of T is maximized among

all possible trees in G. Also, the out-degree of every vertex in T does not exceed R_m, and the depth of T does not exceed L_m.

We can prove the $\mathcal{NP}$-hardness of the MRZT problem through a reduction from the decision version of the *degree-constrained spanning tree problem*, which is known to be $\mathcal{NP}$-complete [14]. The detailed proof is omitted here due to space limitations.

3. Mobility-Robust Zigbee Tree Construction

In this section, we present a heuristic algorithm for effective MRZT construction. First, we differentiate the target problem from existing ones in the literature. We notice that a maximum spanning tree [13] is not necessarily a maximum mobility-robustness Zigbee tree. We illustrate this claim by Figure 2. The total weight of the maximum spanning tree in Figure 2(b) is 8, but the total weight of the maximum mobility-robustness tree in Figure 2(c) is only 6. However, although the edge (C,A) is not in the tree in Figure 2(c), there is an edge from C to A in the connectivity graph in Figure 2(a). Considering the tree in Figure 2(c) is capable of collecting the mobility-robustness for a mobile device moving from C to A, the mobility-robustness of the tree is 11. The mobility-robustness of the tree in Figure 2(c) is larger than that of the tree in Figure 2(b). Instead of choosing the maximum weighted spanning tree, the goal is to find a maximum mobility-robustness Zigbee tree to mitigate the effects of node mobility.



(a) A connectivity graph (b) A maximum spanning tree (c) A maximum mobility-robustness tree

Figure 2. Maximum spanning tree v.s. maximum mobility-robustness tree

3.1 A Tree Construction Algorithm

The algorithm is consisting of two phases: the *ZTG* phase and the *FIX* phase. The ZTG phase searches and connects the vertices that add the most mobility-robustness to the tree. Nevertheless, due to the restriction of *Rm* and *Lm*, after the ZTG phase, there might exist multiple trees instead of a single tree. Therefore, these small trees from the ZTG phase are merged in the FIX phase. The pseudocodes of the proposed algorithm are shown in Algorithm 1 and Algorithm 2.

3.1.1 The ZTG Phase

This section explains the ZTG phase using Algorithm 1. The total metric of mobility-robustness in the tree T is denoted as $robustness(T)$ in the pseudocodes. The $\triangle robustness(F+e)$ denotes the incremental robustness when e is added in F. In the ZTG phase, we initialize all the vertices by setting their parents as -1, and the degree and depth as 0 in Lines 4 - 6. Q is the set of weights of all edges in E. We sort Q in non-increasing order in Line 8. The main loop is contained in Lines 9 - 29, and the loop iterates until Q is empty or the tree F contains all the $|V|$ vertices. At the beginning of the loop, in Lines 10 - 14, we check the existence of pendant vertices, u, with only one directly adjacent vertex v. For the pendant vertices u and v, we connect u and v. Any pendant vertex must be connected at the beginning or it may not be able to connect if any other vertex occupies its adjacent vertex at the later stage of the algorithm. When a vertex u is connected to another vertex v, $parent[u]$ is set as v; and $depth[u]$ is increased by 1 to $depth[v]+1$.

Algorithm 1: ZTG

Input: A bi-directional graph $G=(V,E)$ with weight function W, the maximum number of routers Rm, and the maximum depth of the tree Lm.
Output: A forest of Zigbee trees F

```
1  F ← φ
2  Q ← E;
3  for all u ∈ V do
4      parent[u] ← −1;
5      degree[u] ← 0;
6      depth[u] ← 0;
7  end
8  Sort (Q);
9  repeat
10     for all u ∈ V, u ∉ F do
11         if v ← pendent (u) then
12             Add edge (u,v) into F
13         end
14     end
15     Dequeue the max weight edge e = (s,d) from Q;
16     p ← ShortestPath(s,d) in Q;
17     p̄ ← the reversing path of p;
18     switch max(△robustness(F+e), △robustness(F+p)/(|F+p|−|F|), △robustness(F+p̄)/(|F+p̄|−|F|)) do
19         case △robustness(F+e)
20             Add edge e into F;
21         end
22         case △robustness(F+p)/(|F+p|−|F|)
23             Add path p into F;
24         end
25         case △robustness(F+p̄)/(|F+p̄|−|F|)
26             Add path p̄ into F;
27         end
28     end
29 until |F| = |V| or Q = φ ;
30 return F;
```

In the iterations, i.e., Lines 9 - 29, we denote e as the edge with the maximum robustness in Q, and dequeue it from Q. Then, we apply a *shortest path algorithm* [11] to find an alternative path p with the smallest vertex count in Q. If there are multiple shortest paths with the same vertex count, we choose the heaviest one. We use a *breadth-first search algorithm* [11] to find an alternative path with the largest robustness and shortest vertex count. We denote $\overline{p}$ as the reversing path of p. At each iteration, the algorithm connects one of the three choices, i.e., $e = (s, d)$, p, or $\overline{p}$. We calculate the mobility-robustness and edge count of the three choices. The alternative path p or $\overline{p}$ is consisting of at least the mobility-robustness of e. However, p or $\overline{p}$ uses more edges than e does. Since a single tree can only contain $|V| - 1$ edges, we calculate and connect the path with the greatest mobility-robustness per edge, as shown in Lines 18 - 27. There are two heuristic reasons to use a shortest path algorithm here. First, the longer alternative path is more likely to exceed the depth constraint of the Zigbee tree. Secondly, the longer path tends to have lower mobility-robustness per edge. After the loop is completed, the algorithm outputs the forest F and proceeds into the FIX phase.

3.1.2 The FIX Phase

The FIX phase merges the trees constructed in the ZTG phase. These trees are directed trees, which means that we can only connect one tree to another by their roots. However, there might exist no edge between the root of one tree and the root of the other, or these two trees cannot be connected due to the constraint of depth. Therefore, to merge these trees, we must adjust them first. In order to achieve greater mobility-robustness, we tend to keep the tree of greater mobility-robustness and modify other trees of smaller mobility-robustness.

Algorithm 2: FIX

Input: $G = (V, E)$, and a forest of Zigbee trees F
Output: A Zigbee tree T

1 *max_robustness* $\leftarrow 0$;
2 $T \leftarrow$ the largest mobility-robust tree in F
3 **repeat**
4 　Pick a tree t adjacent to T, $t \in \{F - T\}$
5 　Find $e = \{b, a\}$ between t and T, $a \in T$,$b \in t$, $e \in E$, such that a, b are the vertices with the smallest depth in all the possible edges.
6 　Change t's root to vertex b;
7 　$T \leftarrow T \cup t$;
8 **until** *All trees in F are merged* ;
9 **for** *all* $v \in V$ **do**
10 　$u \leftarrow Root(T)$;
11 　*max_robustness* $\leftarrow$ *robustness*(T);
12 　Change T's root to vertex v;
13 　**if** *robustness*$(T) <$ *max_robustness* **then**
14 　　Change T's root to vertex u;
15 　**end**
16 **end**
17 **return** T;

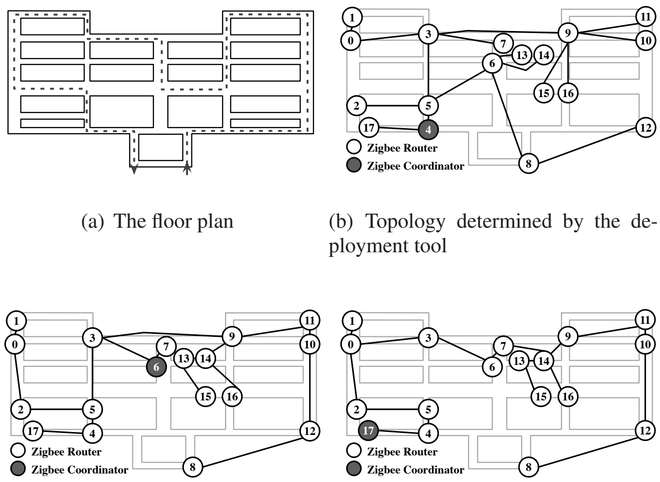
The pseudocodes of the FIX phase are illustrated in Algorithm 2. The main loop is in Lines 3 - 7, where the tree merging is performed. First, we pick the tree with the greatest mobility-robustness from F, denoted as T. Then we check if there is any other tree t adjacent to T. There may be several directly adjacent edges between t and T, denoted as E_t . To meet the constraint of depth, we pick the edge $e = (b, a) \in E_t$, i.e., $a \in T$ and $b \in t$, where a is the vertex with the smallest depth in T and b is the vertex with the smallest depth in t. Then we reconstruct t to be rooted in b and connect (b, a) to merge t and T. This procedure is repeated until all trees are merged into one tree. At the end of the FIX phase, we search different vertices as roots of T to create a tree with greater mobility-robustness in Lines 9 - 15.

The worst case time complexity of the ZTG phase is $O(|E| \times |V|^2)$ time, and the time complexity of the FIX phase is $O(V \times E)$. Overall, the time complexity of a complete process including the two phases is $O(|E| \times |V|^2)$.

4. Simulations and Results

In this section, we conduct extensive simulations and provide better insight into mobility-robust tree topologies in Zigbee wireless networks. We use the deployment tool developed in our previous work [12] to deploy the routers in a real-world indoor scenario so that the coverage and connectivity constraints are satisfied. We then use the NS2 simulator [3], with parameters set based on the standard Zigbee specifications [7], to verify the performance of the proposed approach.

4.1 Simulation Setup



(a) The floor plan
(b) Topology determined by the deployment tool
(c) Topology determined by our algorithm with *Rm* and *Lm* limitations
(d) Topology determined by our algorithm without *Rm* and *Lm* limitations

Figure 3. The Taipei world trade center exhibition hall

Our simulation uses the Taipei world trade center exhibition hall as a study environment. The dimensions of the hall are approximately 100 meters by 60 meters. The arrangements of the booths, shown in Figure 3(a), are based on a real-world business showcase. There is a recommended touring path in the exhibition hall. The mobile devices, i.e., the Zigbee end devices, carried by visitors probabilistically follow the recommended touring path. The radiation pattern of the F-shaped antenna (embedded in the Chipcon CC2420 RF transceiver [2]) used in the simulations is measured by Ansoft HFSS [1]. NS2 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.

Based on the deployment environment and antenna pattern, the deployment tool [12] determines the locations of the coordinator/routers and the connections between them, as shown in Figure 3(b). Figures 3(c) and 3(d) show the tree topologies determined by our algorithm based on a trace scenario collected for 2 hours, where 100 mobile end devices move at an average speed 1.5m/s in the exhibition hall with a probability of 80% following the recommended touring path. The tree topology in Figure 3(c) is derived with the limitations of *Rm* and *Lm* set as 5 and 6, respectively. The topology in Figure 3(d) is derived without the consideration of these limitations. Two performance metrics are adopted for evaluating the mobility-robustness of the three tree topologies: *packet delivery ratio* and *effective path duration*. The packet delivery ratio is the ratio of the number of data packets successfully delivered to mobile end devices to the number of data packets supposed to be received. The effective path duration is defined as the average time for a mobile end device to stay in the same branch of the tree.

Specification	Setting
Network Standard	IEEE 802.15.4
Deployment Environment	100m × 60m Rectangle with Walls
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
Average Communication Range	48m
Average Carrier Sense Range	120m
Traffic Pattern	Poisson Destination
Packet Size	70 bytes
Packet Inter-arrival Rate	0.1 packets/second per node

Table 1. Parameter settings in NS2

We measure the variations of the performance metrics against the *number of mobile devices*, *moving speed*, *location update period*, and *mobility regularity*. The location update period represents how often a mobile device informs the coordinator of its current location. The mobility regularity is defined as the probability of moving by the recommended path when a mobile device encounters a crossroad in the scenario. Once a mobile device does not follow the recommended path, it randomly selects one of the other directions with an equal probability. As the impacts of a parameter of interest are explored, the default settings of the others are set as follows: the number of mobile devices is set as 150, moving speed is set as 1.5 m/s, location update period is set as 240 seconds, and mobility regularity is set as 80%. The parameter settings in NS2 are listed in Table 1, which follows standard Zigbee specifications [7]. Unlisted specifications have the default values of NS2. The results are derived as an average value over 10 independent simulations, each of which runs for a simulation time of 7200 seconds. For fairness, the three topologies were evaluated over the same data sets.

4.2 Results and Observations

Based on the above simulation settings, the numerical results are presented and discussed in this section. In Figure 4, we study the impact of moving speed. Here, the delivery ratio decreases as moving speed increases. This is because the mobile devices change to the communication ranges of different routers more frequently when they move faster. The frequent location changes cause the route information to go stale within a shorter time duration. Therefore, the probability of a successful packet delivery decreases. We observe that the proposed MRZT attains a higher delivery ratio than the conventional tree, regardless of speeds of mobile devices.

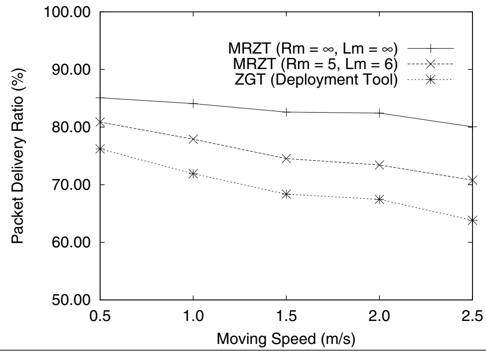


Figure 4. Packet delivery ratio v.s. moving speed

In Figures 5 and 6, we study the impact of mobility regularity. The MRZT in these two figures is obtained using the 80% mobility regularity from the historical data. In Figure 5, the delivery ratio increases as mobility regularity increases. The MRZT attains a higher delivery ratio than the conventional ZGT for various mobility regularities. We observe that when the MRZT is optimized by 80% mobility-regularity, it is capable of retaining higher packet delivery ratios in different mobility regularities, even when the mobility regularity is as low as 50%. This is due to the fact that mobile end devices have the probabilistic tendency to remain in the same branch of the MRZT, where the packet delivered to the mobile node can be received or overheard along the branch. In Figure 6, the effective path duration increases as the mobility regularity increases. The proposed MRZT attains a higher effective path duration than those of the conventional ZGT. The higher effective path duration means the longer the average time for a mobile end device to stay in the same branch of the tree, where the packet delivery to the end device can be successfully overheard. Observations in Figures 5 and 6 verify the robustness of the MRZT.

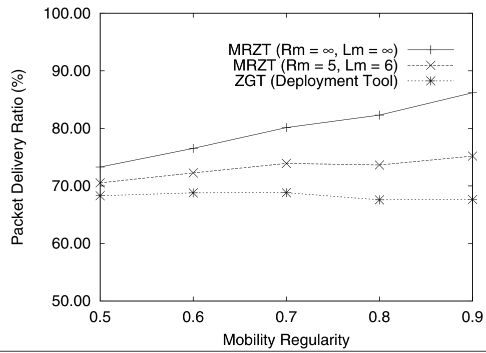


Figure 5. Packet delivery ratio v.s. mobility regularity

In Figure 7, we study the impact of the location update period. The delivery ratio decreases as the location update period increases. This is due to the fact that longer location update periods lead to more chances of stale location information, and thus failures in packet deliveries. Regardless of the values of the location update periods, the proposed MRZT attains a higher delivery ratio than the conventional ZGT. Observations in Figures 7 verify the robustness of the proposed MRZT under various location update periods.

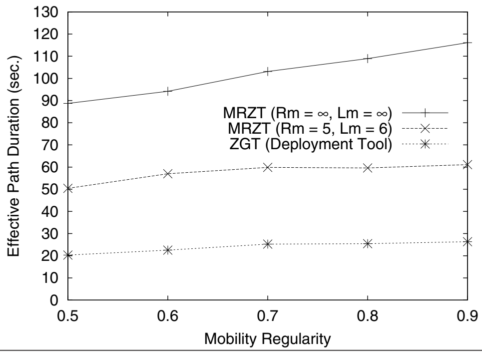


Figure 6. Effective path duration v.s. mobility regularity

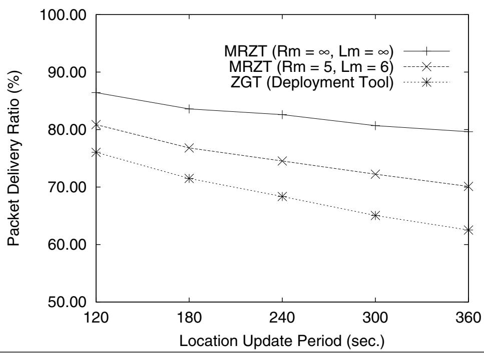


Figure 7. Packet delivery ratio v.s. location update period

In Figure 8, we study the impact of the number of mobile devices. The delivery ratio is almost unchanged as the number of mobile devices increases. Regardless of this quantity, the proposed MRZT attains a higher delivery ratio than the conventional ZGT. Observations in Figures 8 verify the robustness of the proposed MRZT under various network sizes.

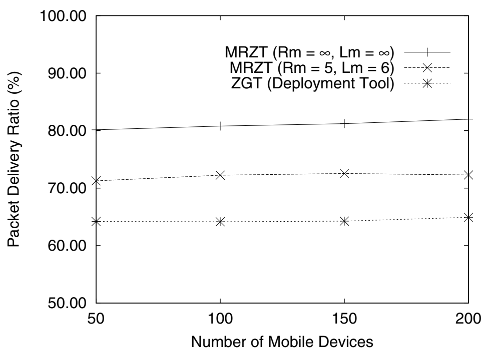


Figure 8. Packet delivery ratio v.s. number of mobile devices

5. Conclusion

In this paper, we introduce the concept of the mobility robustness of a tree topology. A method is proposed to realize the concept and to improve the data delivery ratio of Zigbee wireless sensor networks by constructing a tree topology using the mobility regularities imposed by the physical factors in the environment. The main goal is to construct the tree topology so that mobile devices move with high probability in the directions of the routing paths. By using collected movement data, we construct the tree so that most movements are toward the root, i.e., opposite to the direction of the downlink transmission. By overhearing packets along the branch, data delivery is completed if the destined mobile device is located along the path of the data delivery. We define the maximum mobility-robustness Zigbee tree problem to incorporate the mobility information into the tree construction. We prove the $\mathcal{NP}$-hardness of the problem and proposed an efficient algorithm for effective tree construction. The proposed method achieves a higher data delivery ratio and effective path duration in the scenarios where the movements follow certain regularity. We simulate the proposed approach by NS2 [3] using a real-world business showcase scenario. The simulation results verify and demonstrate the robustness of our proposed approach.

References

[1] Ansoft - HFSS, http://www.ansoft.com/products/hf/hfss.

[2] Texas Instrument - Chipcon CC2420, http://focus.ti.com/lit/ds/symlink/cc2420.pdf.

[3] The Network Simulator - ns2, http://www.isi.edu/nsnam/ns/.

[4] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. In *Ad Hoc Networks*, volume 3, pages 325–349, 2005.

[5] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. In *IEEE Communications Magazine*, volume 40, pages 102–114, 2002.

[6] I.F. Akyildiz, J. Xie, and S. Mohanty. A survey of mobility management in next-generation all-ip-based wireless systems. In *IEEE Wireless Communications*, volume 11, pages 16–28, 2004.

[7] ZigBee Alliance. Zigbee specifications, http://www.zigbee.org/. 2006.

[8] B. An and S. Papavassiliou. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. In *International Journal of Network Management*, volume 11, 2001.

[9] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *Wireless Communications & Mobile Computing*, volume 2, pages 483–502, 2002.

[10] L. Chen, T. Sun, and N. Liang. An evaluation study of mobility support in zigbee networks. In *Journal of Signal Processing Systems*, volume 59, pages 111–122, 2010.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.

[12] Y.K. Huang et al. An integrated deployment tool for zigbee-based wireless sensor networks. In *IEEE/IFIP EUC*, volume 1, pages 309–315, 2008.

[13] H.N. Gabow, Z. Galil, T. Spencer, and R.E. Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. In *Combinatorica*, volume 6, pages 109–122, 1986.

[14] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[15] P.C. Hsiu and T.W. Kuo. A maximum-residual multicast protocol for large-scale mobile ad hoc networks. In *IEEE Transactions on Mobile Computing*, volume 8, pages 1441–1453, 2009.

[16] J. Lee, C. Chuang, and C. Shen. Applications of short-range wireless technologies to industrial automation: A zigbee approach. In *AICT*, pages 15–20, 2009.

[17] C.E. Perkins and D.B. Johnson. Mobility support in IPv6. In *ACM MobiCom*, pages 27–37, 1996.

[18] A.J. Viterbi, A.M. Viterbi, K.S. Gilhousen, and E. Zehavi. Soft handoff extends cdma cell coverage and increases reverse link capacity. In *IEEE Journal on Selected Areas in Communications*, volume 12, pages 1281–1288, 1994.