

A Backbone-Aware Topology Formation (BATF) Scheme for ZigBee Wireless Sensor Networks

Kuei-Li Huang · Li-Hsing Yen · Jui-Tang Wang ·
Chao-Nan Wu · Chien-Chao Tseng

Published online: 9 November 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract In a tree-structured ZigBee wireless sensor network, nodes close to the root of the tree (i.e., hot-spot nodes) may exhaust their power earlier than those distant from the root due to heavy loads on packet forwarding. This hot-spot problem is inherent in tree-structured networks and may demand extra energy to recover from failures of hot-spot nodes. In this paper, the backbone-aware topology formation (BATF) scheme is proposed to alleviate the hot-spot problem. BATF utilizes power-rich nodes to form a backbone tree that does not suffer from the hot-spot problem. Each power-rich node independently initiates a ZigBee tree network that attracts associations from ZigBee-compliant devices in order to distribute packet-forwarding loads over a larger set of nodes. Issues of BATF such as the partition of address space and ZigBee-compliant routing are discussed in detail. Simulation results confirm that BATF does alleviate the hot-spot problem as it improves network lifetime as well as data collection capability. Comparisons with native ZigBee protocols show that the improvement comes from our protocol design rather than simply introducing power-rich nodes.

Keywords ZigBee network · Power node · Backbone · ZigBee-compliance

K.-L. Huang · C.-N. Wu · C.-C. Tseng (✉)
National Chiao-Tung University, Hsin-Chu, Taiwan, ROC
e-mail: cctsen@cs.nctu.edu.tw

K.-L. Huang
e-mail: huangkl@cs.nctu.edu.tw

C.-N. Wu
e-mail: cnwu@cs.nctu.edu.tw

L.-H. Yen
National University of Kaohsiung, Kaohsiung, Taiwan, ROC
e-mail: lhyen@nuk.edu.tw

J.-T. Wang
Industrial Technology Research Institute, Hsin-Chu, Taiwan, ROC
e-mail: rtwang@itri.org.tw

1 Introduction

ZigBee [1] is a wireless sensor network (WSN) standard developed by ZigBee Alliance [2]. It has been widely applied to information collection applications, especially ones such as climate monitoring, soil status monitoring, river status monitoring, etc, that require periodical monitoring of status in target areas. ZigBee supports three networking topologies: star, tree, and mesh. Compared to the other two, tree topology has advantages in high connectivity and low routing overhead, which is more suitable for periodical monitoring applications. However, a tree-structured ZigBee network may suffer from the hot-spot problem, which refers to the phenomenon that nodes close to the root of the tree (i.e., hot-spot nodes) may exhaust their power earlier than those distant from the root due to a heavy load on packet forwarding. This is because, in most applications, data packets flow from every node into the root, and hot-spot nodes forward packets for all its descendants in the tree. As a result, a hot-spot node is bound to run out its energy earlier, resulting in a link breakdown between its parent and the subtree rooted at it. Extra efforts should be done to recover from such failures, which wastes precious energy, thus degrading network lifetime to some extent.

Several methods have been proposed to address the hot-spot problem, among which two main concepts have been developed. The first is clustering [3–9]. Through selecting power-rich nodes as cluster heads or partitioning nodes into clusters of proper sizes, the resulting topology is energy-efficient. However, clustering merely affects the efficiency locally. Without proper associations among cluster heads, however, packets may be delivered through some power-insufficient nodes. Therefore, the second concept, a network capable of organizing nodes of rich power into a backbone, is introduced in [10–14]. As most packet forwarding tasks take place in backbone, the forwarding load on other nodes is alleviated. Nevertheless, a ZigBee network does not embrace the backbone concept, which motivates us to design a ZigBee-compliant backbone for tree-structured ZigBee networks.

In this paper, we assume the availability of devices with virtually unlimited power supply or rich power. This can be achieved, for example, by replacing batteries on devices or using solar power or power-line on the devices. These devices are referred to as Power-Nodes (PNs). MMSN (mutually complementary multichannel sensor network), which is proposed in [15], assumes the use of power-line communication (PLC) in the backbone network. If two routers in the backbone network also have ZigBee interfaces, they have the ability to connect to each other via a ZigBee link. This is used to deal with an intrinsic constraint that two PLC nodes in two different circuit loops cannot reach each other. Only routers in the backbone network can forward packets for other nodes. Any other device can only establish a link with one of these routers via its sole PLC or ZigBee interface and, hence, only act as a leaf node of the network. In other words, ZigBee technology is only used for one-hop wireless access; no ZigBee trees developing from a backbone router are perceived. MMSN yields a robust backbone and long network lifetime. However, star topology limits the range of wireless coverage and, thus, entails low connectivity. To connect all sensors in a target area, the backbone should span almost the entire area to reach every corner. A wireless sensor network for data collection applications is expected to feature not only long network lifetime but also good data collection capability. A network of low connectivity may result in lower data collection capability.

To prolong the lifetime of a ZigBee network and improve its data collection capability, we propose a backbone-aware topology formation (BATF) scheme, where one ZigBee tree network can be rooted at each PN. All PNs form a backbone that connects all independently-developing ZigBee trees to one network. In BATF, all PNs and ZNs are required to

share the same 16-bit address space to uniquely identify every PN and ZN in the network. Furthermore, tree-based routing used by ZigBee, which is tightly-coupled with addressing, is required to work in each PN-rooted ZigBee tree. Therefore, how to manage the address space to accommodate all PNs while preserving the same tree-based routing rule becomes a key challenge. How to attract associations of ZNs with PNs while balancing the load of each PN is also an issue. The bottom line is that: we should not modify the behavior of ZN. As a result, PNs in BATF form a backbone that connects ZigBee trees independently formed by ZNs, reducing the forwarding burden on ZNs while retaining sufficient connectivity in the sense that the backbone needs not span the whole area of the network. Simulation results in Sect. 4 also show that BATF improves the data collection capability of a ZigBee network compared to that of the original ZigBee protocol and MMSN.

We organize the remainder of this paper as follows. In Sect. 2, we describe the topology formation scheme of ZigBee network. Section 3 explains the BATF scheme in detail. In Sect. 4, we present and discuss simulation results. Finally, Sect. 5 concludes this paper.

2 An Overview of ZigBee’s Operations

In this section, tree-structured ZigBee network and its main mechanisms are introduced. As shown in Fig. 1, a tree-structured ZigBee network [1] includes a coordinator, the root of the tree, and several ZNs as tree nodes. Each ZN has a unique network address and a depth value which is the distance (in terms of hop count) between the ZN and the root of the tree. For example, ZN *F* has network address 37 and depth 3. Furthermore, ZNs can be further classified into two types: router nodes and leaf nodes. Router nodes can have children, while leaf nodes cannot. ZN *A* in Fig. 1 is a leaf node, and ZN *F* is a router node.

Packets issued by ZNs are routed along the tree. When a ZN receives an incoming packet, it determines the next-hop node of the packet without consulting a routing table. The destination address of the packet alone suffices for making the routing decision. That is, network address is not only used for node identification but also for routing. In order to perform the networking operation as described above, two main mechanisms are required: tree-structured ZigBee topology formation mechanism and the corresponding routing protocol, which are illustrated in the following two subsections.

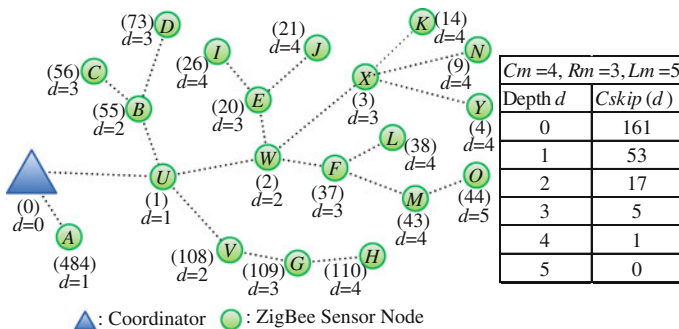


Fig. 1 An ZigBee network

2.1 Tree-Structured ZigBee Topology Formation

A ZigBee tree network can be characterized by the following three parameters:

- C_m : the maximum number of children a router node can have
- R_m : the maximum number of children that can be router nodes
- L_m : the maximum depth value of the tree

These parameters determine scale and structure of a ZigBee tree and the allocation of network addresses to ZNs. The network address allocated to a ZN is determined by the ZN's position in the tree. For any router node, the subtree rooted at it possesses a block of consecutive addresses. More specifically, the subtree rooted at a router node with depth value $d + 1$ is allocated $C_{skip}(d)$ sequential addresses, where $C_{skip}(d)$ is defined as

$$C_{skip}(d) = \frac{1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 - R_m}.$$

This rule applies to each router node in the tree. Figure 1 illustrates a sample tree network with the relationship between d and the C_{skip} 's shown. With this design, a parent node can locally assign a network address that is globally unique to its child. If the child joins the network as a router node and is the n th router-node child of the parent in depth d , it is allocated a network address

$$ID_{nth-router-node-child} = ID_{parent} + (n - 1) \times C_{skip}(d) + 1,$$

where ID_{parent} is the address of the parent. The subtree rooted at the child is allocated a block of consecutive addresses inclusively from $ID_{nthrouter-node-child}$ to $ID_{parent} + n \times C_{skip}(d)$. On the other hand, if the child joins the network as a leaf node and is the m th end-node child of the parent, then its address is

$$ID_{mth-end-node-child} = ID_{parent} + R_m \times C_{skip}(d) + m.$$

Note that no address block is allocated to any end-node children. For example, router node F in Fig. 1 is the 3rd router-node child of W . The depth value of W is 2. Therefore, W assigns network address $2 + (3 - 1) \times 17 + 1 = 37$ to F which possesses a block of addresses from 37 to 54 for the subtree rooted at F . On the other hand, A is the first end-node child of the coordinator, so it is assigned a network address $0 + 3 \times 161 + 1 = 484$. If beacon-mode is enabled, a router node periodically broadcasts beacon frames that contain the address and depth value of the router node. When a ZigBee device enters a ZigBee network, it first searches for neighboring router nodes by scanning all channels for possible beacon frames. Among the neighboring ZNs found, the device chooses and associates with the ZN that has the lowest depth value. The device obtains a network address from its parent node as part of the association procedure.

2.2 Tree Routing in ZigBee Networks

Since the location of every ZN in the tree is uniquely identified by the ZN's address, a router node is able to derive the next-hop node for a received packet from the packet's destination address. More specifically, when a router node with address A at depth d receives a packet with destination address D , it performs the following action:

1. If $A < D < A + C_{skip}(d - 1)$, send the packet to a descendent node with address N , where

- (a) $N = D$, if D is for a leaf node or
 - (b) $N = A + 1 + \left\lfloor \frac{D-(A+1)}{C_{skip}(d)} \right\rfloor \times C_{skip}(d)$, otherwise.
2. Otherwise, send the packet to the parent node.

Since addresses in a ZigBee tree topology are used for both node identification and packet routing, the address space should be pre-planned, making the addressing method inflexible. The inflexibility may significantly reduce the number of addressable nodes. Hence, simply introducing PNs to a ZigBee network may not take advantage of their rich power. For example, assume node F , U , V and G are PNs. Node F eventually joins node W , instead of G . This makes node F unable to become part of backbone, leaving the forwarding burden to node W . In addition, a PN with neighbor ZNs may have no children since the PN has a higher depth value and is possibly a leaf node. For instance, node Y joins node X , instead of PN L , because node Y has a lower depth value. PN O is unable to have children since it is a leaf node. To solve these problems, we propose the BATF mechanism which utilizes PNs to forms a ZigBee-compliant PN backbone in a ZigBee tree-structured network.

3 Backbone-Aware Topology Formation (BATF)

In a conventional ZigBee tree, only the root of the tree acts as a ZigBee coordinator. In BATF, however, each PN is utilized as a virtual coordinator, from which a ZigBee-compliant tree can independently develop. A ZigBee device joins one of these trees and sees the virtual coordinator as a conventional ZigBee coordinator. All PNs are connected by another tree network (referred to as backbone tree). The hot-spot problem is alleviated because BATF introduces more ZNs with a depth value 1 in order to share traffic loads, and the load for each of the ZNs is significantly reduced. Figure 2 shows the typical topology yielded by BATF where the backbone tree connects the coordinator and all PNs. In general, the resulting topology is effectively a connected forest consisting of several PN-rooted ZigBee trees. Consequently, the depth value of each ZN is generally reduced.

There are three design requirements in developing BATF. First, routing in a PN-rooted ZigBee tree should comply with conventional ZigBee routing, while routing in a backbone tree can be customized. Second, all devices share the same address space, so the address space should be partitioned to accommodate the needs of PN-rooted trees and the backbone tree. Finally, a virtual coordinator should be indistinguishable from a conventional ZigBee coordinator from a ZN's point of view but should have a depth value lower than ZigBee router nodes to attract associations from ZNs.

Fig. 2 A wireless network formed by BATF

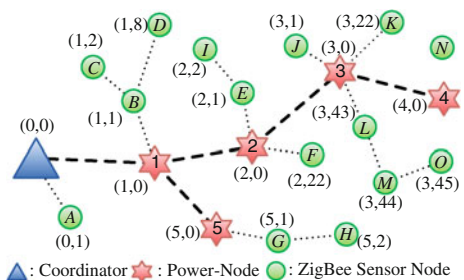
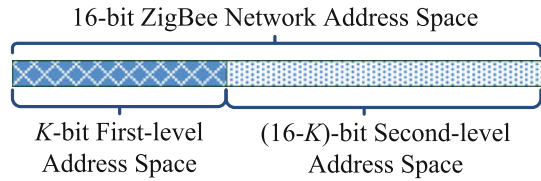


Fig. 3 Two-level network address space planning



3.1 Network Address Management

In BATEF, a 16-bit ZigBee network address is partitioned into two parts as shown in Fig. 3. The first K bits of the address field are used to identify PNs, where K is the smallest number such that 2^K addresses suffice for all PNs. This leaves the largest possible address space for ZNs, thus retaining the connectivity of each PN-rooted ZigBee tree.

For a PN, the network address can be expressed as $(A_1, 0)$, where A_1 ranges from 1 to 2^K . This address could be either assigned by the coordinator or pre-configured prior to the topology formation. Unlike standard ZigBee addressing scheme, the addressing scheme for PNs needs not be hierarchical. As a result, the number of children each virtual coordinator can have in the backbone tree is not limited. This increases flexibility of backbone tree formation.

For a PN with network address $(A_1, 0)$, the network address for any ZN that is the PN's descendant should be (A_1, A_2) , where A_2 is configured with the standard ZigBee addressing scheme. For example, M in Fig. 2 is assigned the address $(3, 44)$, where 3 is inherited from PN 3, and 44 is derived according to the ZigBee addressing rule. Note that to apply the ZigBee addressing rule, parameters C_m, R_m, L_m should be set first. The setting should meet the requirement that the length of the largest possible address should not exceed $(16 - K)$ bits.

This two-level network address formation does not affect the ZigBee routing performed by ZNs, which will be verified in Sect. 3.3.

3.2 Topology Formation

To let ZigBee-compliant ZNs associate with a virtual coordinator with priority, beacon frames issued by PNs are modified to have a depth value of zero. This modification prioritizes associations between ZNs and PNs. Beacon frames issued by a PN also include additional information called core depth, which indicates the depth value of the PN with respect to the backbone. This information helps a PN identify neighboring PNs that are closer to the coordinator than itself.

A PN P performs the procedure shown in Fig. 4 to join a ZigBee network:

Step 1. P searches for beacons of neighboring PNs and ZNs. The result is stored in P 's Neighbor Table (NT).

Step 2. From its NT, P chooses a PN with a depth value of zero to make association with. If two or more neighboring PNs are eligible, the PN with the smallest value of core depth will be selected.

Step 3. After its association, P sends a route update message toward the coordinator. This is to update the routing table of each PN along the path of P to the coordinator.

Step 4. P now becomes a virtual coordinator. It starts periodically sending beacons with a depth value of zero to attract associations of new coming ZNs. P also selects from its NT neighboring ZNs that have high depth values. These ZNs could have lower depth values if they re-associate with P .

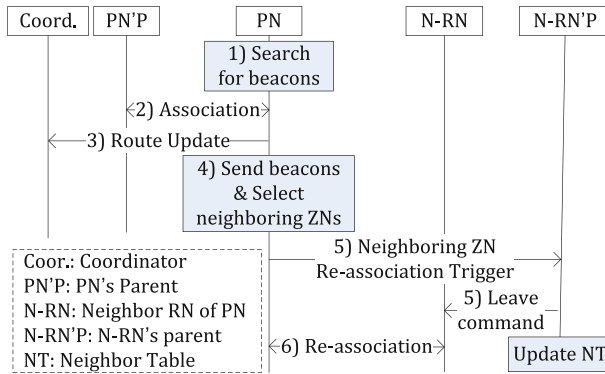


Fig. 4 PN's network join procedure

Step 5. For each ZN found in Step 4, *P* commands its parent node to disassociate from it. The parent node then updates its own NT accordingly.

Step 6. Each ZN disassociated in Step 5 performs re-association to rejoin the network with *P* being its new parent.

Through this procedure, a PN is able to act as a virtual coordinator, which attracts associations of both new coming ZNs and ZNs with high depth values. For example, suppose PN 4 in Fig. 2 learns of ZN *O* with depth 3 in the tree rooted at PN 3 from PN 4's NT. Hence, PN 4 sends a command to ZN *O*'s parent, ZN *M*. ZN *M* then sends a leave command to ZN *O*. In addition, when a new comer, ZN *N*, scans all channels for Beacon frames, it will find out PN 4 with depth zero and ZN *K* with depth 1; therefore, it will choose PN 4 as its parent.

3.3 Routing in BATF

Although network address is no longer flat, standard ZigBee routing rules still work in PN-rooted trees. That is, a ZN routes a packet to one of its children only if the destination address of the packet falls within the range of the address space allocated to that child; otherwise, the ZN sends the packet to its parent. It follows that a packet with destination address falling into the range of other PN-rooted ZigBee tree will be routed to the backbone tree first. We give a formal mathematical verification as follows.

Lemma 1 All ZNs in a PN-rooted tree can correctly forward received packets by ZigBee routing rules.

Proof Consider a ZN in depth *d* with network address $A = (A_1, A_2)$ and a received packet with destination address $D = (D_1, D_2)$ where the first-level address space is *K*-bit. We shall prove that (1) if *D* is a descendent, the ZN will forward the packet to the child that is *D*'s ancestor, and (2) if *D* is not a descendent, the ZN will forward the packet to its parent.

- (1) Since the second-level address is configured with standard ZigBee addressing rules, the condition that *D* is a descendent of the ZN implies

$$A_2 < D_2 < A_2 + C_{skip}(d - 1),$$

where $C_{skip}(d - 1)$ is smaller than $2^{(16-K)}$. The condition also implies that D and A belong to the same PN-rooted tree and hence $A_1 = D_1$. Therefore,

$$\begin{aligned} A_2 < D_2 < A_2 + C_{skip}(d - 1) \times 2^{(16-K)} \\ \Rightarrow A_1 + A_2 < 2^{(16-K)} \times D_1 + D_2 < 2^{(16-K)} \times A_1 + A_2 + C_{skip}(d - 1) \\ \Rightarrow A < D < A + C_{skip}(d - 1). \end{aligned}$$

The above derivation implies that the ZN will forward the packet to a child by ZigBee routing rules. Now we show that the packet will be forwarded to a correct child. Suppose that the ZN forwards the packet to a child with network address $N = (N_1, N_2)$, where $N_1 = A_1$. If D corresponds to a leaf node of the ZN, then there exists m such that $0 < m \leq C_m - R_m$ and

$$\begin{aligned} D &= D_1 \times 2^{(16-K)} + D_2 \\ &= D_1 \times 2^{(16-K)} + A_2 + R_m \times C_{skip}(d) + m. \end{aligned}$$

Actual network addresses of the end-node children of the ZN are $A_1 \times 2^{(16-K)} + A_2 + R_m \times C_{skip}(d) + i$, where $0 < i \leq C_m - R_m$. Since the ZN and its end-node child are in the same PN-rooted tree, $A_1 = D_1$, and we can find an i such that $m = i$ and

$$\begin{aligned} N &= A_1 \times 2^{(16-K)} + A_2 + R_m \times C_{skip}(d) + i \\ &= D_1 \times 2^{(16-K)} + A_2 + R_m \times C_{skip}(d) + m \\ &= D_1 \times 2^{(16-K)} + D_2 \\ &= D. \end{aligned}$$

Therefore, the ZN can find a correct node with network address N to forward the packet when the destination is a leaf node of the ZN. If D is a descendent of the ZN, then according to ZigBee routing rules, we have

$$N_2 = A_2 + 1 + \left\lfloor \frac{D_2 - (A_2 + 1)}{C_{skip}(d)} \right\rfloor \times C_{skip}(d).$$

From the perspective of the ZN, it computes a network address as follows:

$$\begin{aligned} N &= A + 1 + \left\lfloor \frac{D - (A + 1)}{C_{skip}(d)} \right\rfloor \times C_{skip}(d) \\ &= (A_1 \times 2^{(16-K)} + A_2) + 1 \\ &\quad + \left\lfloor \frac{D_1 \times 2^{(16-K)} + D_2 - (A_1 \times 2^{(16-K)} + A_2 + 1)}{C_{skip}(d)} \right\rfloor \times C_{skip}(d) \\ &= A_1 \times 2^{(16-K)} + A_2 + 1 + \left\lfloor \frac{D_2 - (A_2 + 1)}{C_{skip}(d)} \right\rfloor \times C_{skip}(d) \\ &= A_1 \times 2^{(16-K)} + N_2 \\ &= N_1 \times 2^{(16-K)} + N_2. \end{aligned}$$

As shown in the above equation, a ZigBee-compliant ZN can simply follow standard ZigBee routing rules to forward a received packet when the destination belongs to one of the ZN's descendents.

- (2) Since the ZN can correctly determine whether or not the destination of the packet is a descendent of the ZN, the ZN can correctly forward the packet to its parent node if the packet is not for its descendent.

Conditions (1) and (2) together show that a packet with a two-level network address can be correctly routed through the ZigBee routing rules.

We also give an example to illustrate the ZigBee compliance to the backbone tree and the design of the two-level network address. Take the network in Fig. 2 as an example, if a packet is sent from ZN *C* with network address (1, 2) to ZN *M* with network address (3, 44), the packet would first arrive at ZN *B* with network address (1, 1). The value of *B*'s address is $1 \times 2^{13} + 1 = 8,193$. As *B* is at depth 1 and C_{skip} has a value of 6, the subtree rooted at *B* possesses a block of consecutive addresses ranging from 8,193 to $8,193 + 6 = 8,199$. Since the destination address $(3, 44) = 3 \times 2^{13} + 44 = 24,620$ is not within this range, *B* forwards the packet to its parent node, the power node (1,0). PN (1,0) forwards the packet to PN (2,0) by consulting to its routing table for the first-level address of the destination. After a series of forwarding through the backbone tree, the packet reaches PN (3,0). PN (3,0) then finds out that the destination belongs to the tree rooted at itself and hence, follows ZigBee routing rules to compute the next hop as follows:

$$\begin{aligned}
 N &= 3 \times 2^{13} + 1 + \left\lfloor \frac{3 \times 2^{13} + 44 - (3 \times 2^{13} + 1) - (A + 1)}{C_{skip}(0)} \right\rfloor \times C_{skip}(0) \\
 &= 3 \times 2^{13} + 1 + \left\lfloor \frac{43}{21} \right\rfloor \times 21 \\
 &= 3 \times 2^{13} + 43.
 \end{aligned}$$

Hence, the packet arrives at ZN (3, 43). Finally, ZN (3, 43) follows the ZigBee routing rules to forward the packet to the destination ZN (3,44).

4 Simulations

We used NS-2 [16] as the simulation tool to evaluate BATF in periodical data collection wireless sensor network applications. We adopted NS-2.32 as our simulation platform, in which the 802.15.4 modules developed by Zheng [17] have been included. On the basis of the 802.15.4 architecture, we developed the ZigBee tree routing module and BATF module. As shown in Fig. 5, the module blocks with shadows are the modules we developed, while the others are the 802.15.4 modules. In the BATF module, two main functions are included: (1) Backbone Routing and (2) ZN Attraction Function. Backbone routing in BATF uses the first-level address and routing tables to forward a packet to a correct PN. ZN Attraction Function is responsible for depth information modification, neighbor ZN selection, and re-association triggering.

The evaluation metrics used in the simulations are network lifetime and data collection capability. Network lifetime is the time duration from network startup to the time that no ZN is alive. We define data collection capability (*DCC*) as the accumulated number of successfully delivered packets from ZNs during the network lifetime, formulated as follows.

$$DCC = \int_0^T N_{srp}(t)dt$$

T is the network lifetime, and $N_{srp}(t)dt$ is the number of successfully delivered packets in time interval *dt*.

Two different simulations were set up: (1) PNs and ZNs were randomly deployed and (2) PNs and ZNs were regularly deployed. In the first setup, a number of ZNs are uniformly

Fig. 5 Architectures of power nodes and ZigBee nodes in NS-2 network simulator

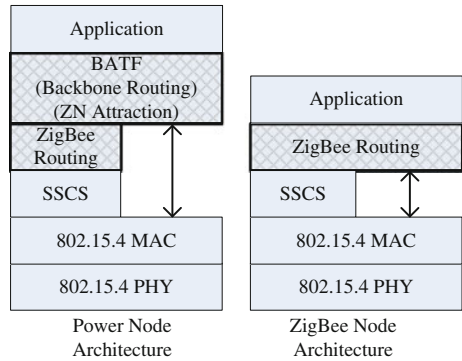


Table 1 Parameter configuration in simulation I

Parameter	Value
PN deployment area	33 × 23 m ²
ZN deployment area	43 × 43 m ²
Node communication radius	10 m
Initial ZN energy	4 J
Traffic type	Constant bit rate
Report packet size	70 bytes
ZN reporting period	20 s
K (Size of first-level address space)	4 bits
Number of ZNs	90
Number of PNs (the coordinator included)	11
Number of trials	200
Simulation stop time	20,000 s

yet randomly distributed over the target area, while PNs are randomly deployed near the coordinator. This scenario mimics the situation when a lot of sensors are dropped by vehicles or airplanes. The second setup applies to a target area that is easy to access, e.g., the soil moisture content in a farm.

4.1 Simulation I: PNs and ZNs are Randomly Deployed

In the first simulation, PNs and ZNs were randomly deployed. The parameter configuration in Simulation I is shown in Table 1. The coordinator was located at the center, whereas the area to deploy PNs was centered at the coordinator.

We are interested in how the number of PNs affects network lifetime and data collection capability. Figures 6 and 7 respectively show the average number and variance of living nodes versus time. The result shown indicates that simply adding PNs into a ZigBee network (ZigBee+PN in these figures) without proper utilization of the power resource in solving the hot-spot problem does not necessarily bring in performance enhancement. In contrast, the hot-spot problem was significantly alleviated when PNs were added with BATF employed, yielding an drastic improvement on network lifetime. Furthermore, although MMSN has a

Fig. 6 Number of living nodes during network lifetime in the simulation I

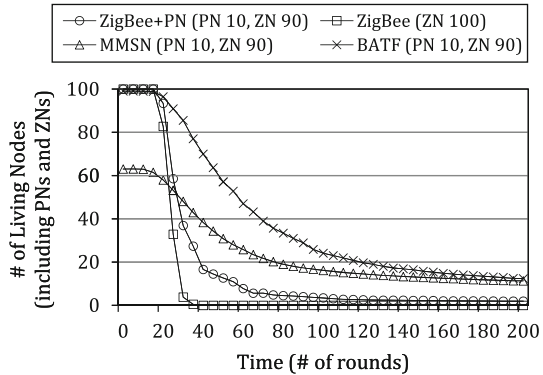
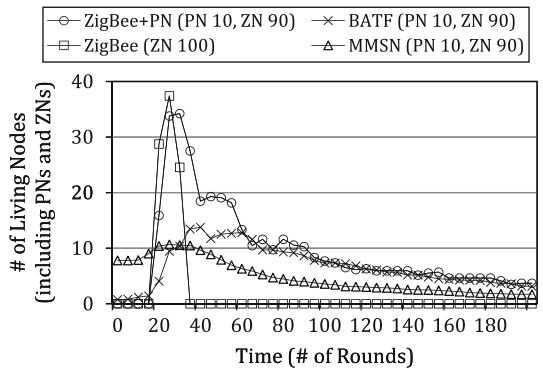


Fig. 7 The variance of number of living nodes during network lifetime in the simulation I



network lifetime similar to BATF, its connectivity in terms of the number of living ZNs in round 0 was lower than those of the other three methods.

Notably, ZigBee networks (both with extra PNs or ZNs) had high variance values during round 20 to round 70 due to events of power exhaustion and re-associations occurring frequently during this period. This caused a significant amount of power spent on re-associations of ZNs when their ancestors died. On the other hand, BATF distributed these events over a longer time period. Such distribution lowered power consumption rate and alleviated the hot-spot problem. In contrast, MMSN, which does not entail the hot spot problem, had higher variance in the first 20 rounds since the number of ZNs connected to PNs varies. From round 20 to round 40, the variance slightly increased because some hot-spots die in some cases and not yet in other cases. After round 40, the hot-spot problem took place in most cases. The number of living ZNs decreased gradually thereafter.

A wireless sensor network for data collection applications prefers not only long network lifetime but also good data collection capability. Figure 8 shows how the accumulated number of packets increases with network lifetime. These results were obtained from 200 trials. We observed that both BATF and ZigBee+PN can reach network lifetime of 1,000 rounds in some trials, but BATF collected more packets than ZigBee+PN during this period in most cases. BATF also outperforms MMSN in terms of data collection capability due to better connectivity.

We further classified the BATF trials in the number of PNs connected to backbone. Figure 9 shows the result, and we can see that data collection capability increased as more PNs were introduced to the backbone tree. Although network lifetime is not fully proportional to the

Fig. 8 Data collection capability versus network lifetime in simulation I

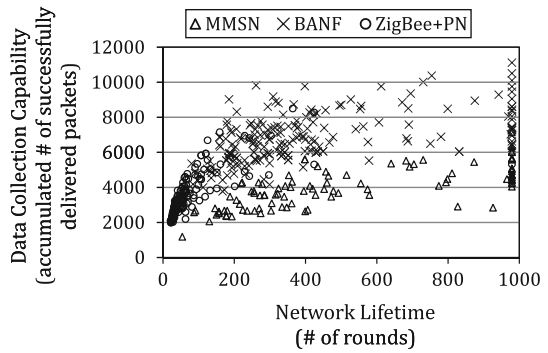


Fig. 9 Number of living ZNs during network lifetime with different number of PNs in the backbone tree in simulation I

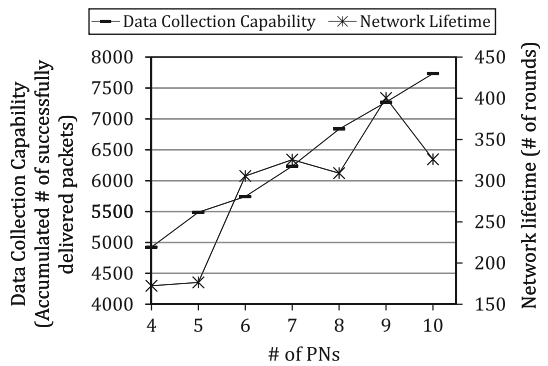


Table 2 Total number of (re)joins during network lifetime, average network lifetime and average data collection capability

	# of ZN (re)joins	# of PN (re)joins	network lifetime (# of rounds)	data collection capability (# number of packets)
BATF	231.05 (158.25)	9.4	421.89	6,910.52
ZigBee+PN	133.95 (89.97)	21.82	67.52	3,357.36
ZigBee	115.27 (99.97)	0.00	25.61	2,273.14
MMSN	54.40 (54.40)	9.37	482.01	3,790.25

number of PNs, network lifetime tends to increase with the number of PNs. The oscillatory behavior of the network lifetime with the number of PNs may be a result of the hot-spot problem. When a hot-spot is dead, ZNs in its subtree re-associate with other subtrees. However, a subtree may be geographically isolated from other subtrees. In that case nodes in the subtree do not re-associate with any other ZigBee routers; therefore, they are able to sustain life longer, yielding a longer lifetime for the network.

We further detailed the cost and benefit of BATF. Table 2 shows the number of joins of PNs and ZNs, average network lifetime, and average data collection capability for different methods, respectively. Numbers in parentheses stand for the number of (re)joins (successful (re)associations) before data collection, i.e., those required to form the initial tree topology. We observed that BATF had around 159 ZN (re)joins before data collection, about 68 higher than that in ZigBee+PN. This means that PNs' attraction operations in BATF yield 68 more

Fig. 10 Deployments of sensor nodes in simulation II

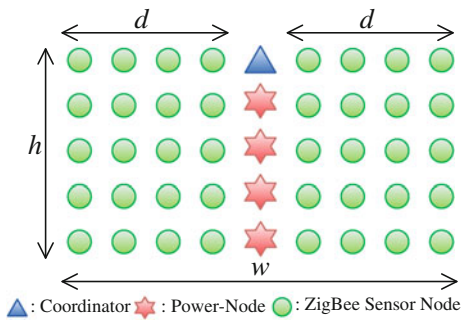


Table 3 Parameter configuration in simulation II

Parameter	Value				
Node communication radius	10 m				
Initial ZN energy	4 J				
Traffic type	Constant bit rate				
Report packet size	70 bytes				
ZN reporting period	20 s				
Simulation stop time	20,000 s				
Topology	5×20	7×14	10×10	14×7	20×5
K (Size of first-level address space) (bits)	4	4	4	4	5
Number of ZNs	95	91	90	86	80
Number of PNs (the coordinator included)	5	7	10	14	20

(re)joins to reform the network. Although the re-formation is costly, BATF significantly improved the network lifetime and data collection capability, compared to ZigBee+PN and ZigBee. MMSN, free from the hot spot problem, had a longer network lifetime but a lower data collection capability than BATF. Furthermore, more PN (re)joins in ZigBee+PN indicates that some PNs associate with ZNs, further re-associating with other nodes when the hot-spot in its subtree is exhausted, putting more burdens on hot-spots.

4.2 Simulation II: PNs and ZNs are Regularly Deployed

Deployments of sensor nodes in Simulation II setting formed a grid structure. Figure 10 shows the deployment pattern, where PNs were arranged in one column across the middle of the grid. Let h and w be the number of rows and columns in the grid, respectively. We set up five different topologies consisting of roughly 100 nodes, in which $h \times w$ was 5×20 , 7×14 , 10×10 , 14×7 , and 20×5 , respectively. The number of PNs (excluding the coordinator) in each of these topologies was $h - 1$. The parameter configuration is summarized in Table 3.

Figure 11 shows the result of network lifetime. We can see that the network lifetime with BATF is longer than that without (i.e., ZigBee+PN). Also, network lifetime is not in proportion to the number of PNs in the two methods. For example, a 7×14 grid featured a longer network lifetime than a 14×7 one, both with and without BATF. Among others, MMSN exhibits the longest network lifetime. The network lifetime in a 20×5 grid was

Fig. 11 Network lifetime in different topologies in simulation II

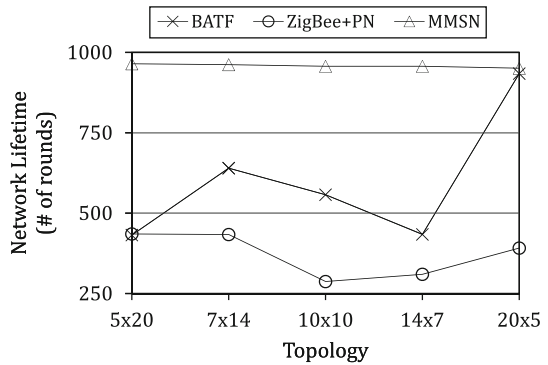
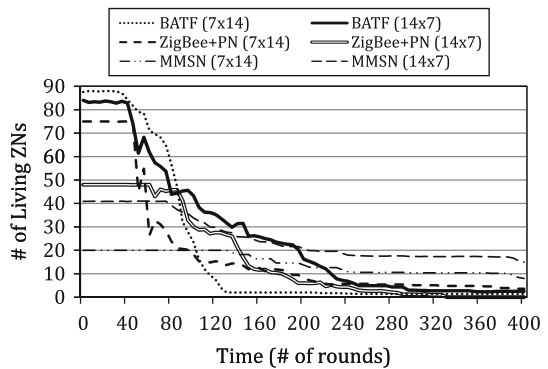


Fig. 12 Number of living ZNs versus time for 7×14 and 14×7 grids, with and without BATF

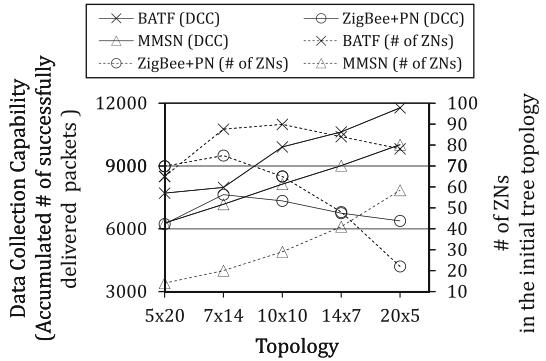


slightly shorter than that in 20×5 because the transmission of a ZN may be intercepted by the transmission of neighboring PNs which forward packets for the ZNs distant from the coordinator.

A closer investigation (as shown in Fig. 12) reveals the effect of the hot-spot problem. In ZigBee+PN, the number of living ZNs in a 7×14 grid decreases quickly in the beginning but remains stable subsequently due to unbalanced ZigBee subtrees in grid deployments. That is, there are large subtrees nearby the coordinator and small subtrees connecting to PNs distant from the coordinator. When a hot-spot node in a large subtree dies, a large number of its descendants could re-associate with other subtrees but small subtrees may not accept these re-associations due to the maximum depth limitation. As a result, a large amount of ZNs die due to the death of hot-spot nodes in large subtrees, all other nodes in small subtrees live for a long time for light traffic load in small subtrees. The 14×7 grid exhibits the same fashion as the 7×14 , but not as obvious due to a lower connectivity. In contrast, because of the two-level network address space planning, PNs in BATF have address spaces of same size for their subtrees; thus subtrees rooted at PNs are formed of similar sizes. This results in a slower decreasing rate in the number of living ZNs. Due to the absence of the hot-spot problem, MMSN shows the lowest decreasing rate in the number of living ZNs but exhibits the lowest connectivity.

Although network lifetime does not increase with the number of PNs, data collection capability does. Figure 13 displays a strict increase in data collection capability for BATF. Data collection capability in ZigBee+PN does not exhibit such trend. ZigBee+PN achieves the best data collection capability for the 7×14 grid due to the mixing effect of the hot-spot

Fig. 13 Data collection capability for different grids in simulation II



problem and connectivity. While the 7×14 grid accommodates the most ZNs, it still suffers from the hot-spot problem. In fact, the hot-spot problem in the 7×14 grid is more severe than in the 10×10 , 14×7 , or 20×5 grids. However, the numbers of ZNs in the 10×10 , 14×7 , and 20×5 grids were much lower than that of 7×14 grid, which results in a lower data collection capability. Without entailing the hot-spot problem, data collection capability in MMSN is proportional to its connectivity. Although featuring the longest network lifetime, MMSN presents a weaker data collection capability than BATF due to a weaker connectivity.

We summarize the main findings from the simulations as follows. First of all, connectivity and network lifetime jointly decide the data collection capability. Although ZigBee+PN has good topology connectivity, the data collection capability is low due to limited network lifetime resulting from the severe hot-spot problem. In contrast, MMSN features long network lifetime but lower connectivity, failing in collecting data from ZNs distant from backbone. Data collection capability is consequently lower. Nevertheless, BATF maintains connectivity to a sufficient degree and prolongs the network lifetime. We also observe that data collection capability is directly proportional to the number of PNs while network lifetime is weakly correlated with the number of PNs due to the effect of the hot spot problem.

5 Conclusion

We have presented BATF, a network infrastructure for tree-structured ZigBee networks that utilizes PNs as virtual coordinators for regular ZigBee devices. BATF distributes the traffic load of hot-spot nodes by organizing all PNs into a backbone tree. Yet it is ZigBee-compatible from the perspective of ZNs. In other words, BATF demands no modification from regular ZigBee devices. The simulation result has shown that BATF mitigates the hot-spot problem as it successfully improves network lifetime as well as data collection capability.

Acknowledgment This work was supported in part by National Science Council, Taiwan under Grants NSC 97-2221-E-009-051-MY3 and NSC 99-2220-E-009-046, and by D-Link Co., Taiwan.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.



Li-Hsing Yen received the B.S. (1989), M.S. (1991), and Ph.D. (1997) degrees in computer science, all from National Chiao Tung University, Taiwan. He was with the Department of Computer Science and Information Engineering at Chung Hua University, Taiwan, from 1998 to 2006. Dr. Yen has been with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Taiwan, since 2006 and is currently a full professor. His research interests include mobile computing, wireless networking, and distributed algorithms.



Jui-Tang Wang received his B.S. degree in the Department of Industrial Engineering and Management from National Chin-Yi Institute of Technology, Taichung, Taiwan, in 1995 and his MS degree in Computer Science and information engineering from National Cheng-Kung University, Tainan, Taiwan, in 2000. Finally, he received Ph.D. degree in the Department of Computer Science and Information Engineering at National Chiao Tung University in 2008. He is currently working with Information and Communications Research Laboratories at Industrial Technology Research Institute, Taiwan. His research interests include Mobile Computing, Network Security and Wireless Internet.



Chao-Nan Wu received the B.S. in computer science and information engineering from National Changhua University of Education in 2006 and the M.S. in computer science from National Chiao-Tung University in 2008. Currently, He is a computer engineer in CyberLink Corporation. His research interests include wireless communications and computer networks.



Chien-Chao Tseng is currently a professor in the Department of Computer Science at National Chiao-Tung University, Hsin-Chu, Taiwan. He received his B.S. degree in Industrial Engineering from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981; M.S. and Ph.D. degrees in Computer Science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. His research interests include Wireless Internet, Handover Techniques for Heterogeneous Networks, and Mobile Computing.