The Room Shortage Problem of Tree-Based ZigBee/IEEE 802.15.4 Wireless Networks

Li-Hsing Yen and Wei-Ting Tsai

Abstract

A number of IEEE 802.15.4 devices can form a tree topology as proposed by ZigBee specification. The ability to confine the shape and extent of the tree serves as the basis for address configuration and packet routing. This paper identifies the room shortage problem in tree-based ZigBee networks, which refers to the phenomenon that some devices are unable to get addresses while many addresses are still left unused. Room shortage problem occurs when pre-allocated address space does not well match the underlying physical topology. To alleviate the problem, we developed three alternatives to the standard addressing mechanism. These approaches manage address space with flexibility yet still support tree-based routing. Performance evaluations indicate that proposed approaches provide different levels of tradeoff between the ratio of addressable devices and storage overhead.

Keywords: Address configuration, Tree, ZigBee, Wireless network.

I. Introduction

IEEE 802.15.4 [1] is a standard for wireless Personal Area Networks (PANs), which comprise devices that are characterized by low data rate, short communication range, and low cost. Depending on their capabilities, these devices can be categorized into full function devices (FFDs) and reduced function devices (RFDs). FFDs are able to forward frames for other devices, while RFDs lack such capability. An FFD can initiate a PAN and act as the coordinator of the PAN. A coordinator can periodically broadcast beacon frames so nearby RFDs can discover it and thereby join the PAN, forming a star topology.

ZigBee specification [2] extends the basic star topology of an IEEE 802.15.4 PAN to a tree or mesh. In a tree topology, the root (called ZigBee Coordinator; ZC) and all internal nodes (called ZigBee Routers; ZRs) are FFDs, while RFDs can only be leaf nodes called ZigBee End Devices (ZEDs). When a ZR or ZED joins the network, it must be assigned a network address that is unique in the tree. A ZigBee network address is 16-bit long, so potentially 65,535 addresses can be assigned to all ZigBee devices in the tree (address 0 is reserved for the ZC). This amount should suffice for most applications.

A tree-based ZigBee network is characterized by *topological parameter*, which limits the height of the tree and the maximal number of children devices/ZRs that a ZC/ZR can have. By setting the topological parameter at the PAN initialization time, a ZC can confine the shape and extent of the tree. However, the actual topology also depends on the geographical distribution of devices. If the setting of the topological parameter does not well match the geographical distribution of devices, problems such as waste of address space, *room shortage*, and *routing detour* may occur. The room shortage problem refers to the phenomenon that some devices are unable to get addresses while many addresses are still left unused. The routing detour problem occurs when the established path from some device to the ZC is not the shortest (in terms of hop count) among all potential ones. Such path demands extra transmissions, wasting precious bandwidth and energy resource.

This research work was motivated by the need to accommodate as many devices as possible in the tree. Our first thought was to design an automatic way to yield a parameter setting that well matches the underlying physical topology. But we soon realized that this approach is impossible according to the specification since the ZC cannot be made aware of the presence of other devices prior to the initialization of the PAN. Therefore, we decided to get rid of the topology constraint imposed by the topological parameter. Since the topological parameter serves as a basis for addressing and routing in ZigBee tree networks, our mission is therefore to design alternative addressing methods that support routing while alleviating the room shortage problem to the most possible extent.

Distributed Address Assignment Mechanism (DAAM) is the default addressing method for ZigBee networks. With a given setting of the topological parameter, DAAM reserves a unique address for each possible location in the tree so that every subtree possesses a continuous address block. A desired property of DAAM is that a ZC/ZR can locally allocate addresses to its children with the knowledge of its own depth value and the global setting of the topological parameter. The tree structure and the regularity of addressing also simplify the task of routing. When a ZC/ZR receives a packet not destined for it, the next-hop node can be deduced directly from the destination address without consulting a routing table. The deduction-only routing rule eliminates the need for extra storage to keep routing information in ZC/ZRs.

In this paper, we propose three alternatives to DAAM. The first one, Centralized Stateful Address Configuration (CSAC), adapts conventional stateful addressing method to ZigBee trees. As CSAC creates no static binding between addresses and node locations in the tree, every available address is assignable to any node and addressing failure occurs only for address exhaustion. The weakness of CSAC is the need for additional storage in every ZC/ZR to keep a routing table. The second approach, called Hybrid Address Configuration (HAC), attempts to make a compromise between DAAM and CSAC by utilizing DAAM with priority and applies CSAC only when needed. The third one, Router-Based Address Configuration (RBAC), partitions address space into chunks and assigns one chunk to each ZR on demands. A ZR then locally allocates available addresses from its chunk to its child ZEDs.

The rest of this paper is organized as follows. Technical background and related work are given in the next section. Section 3 presents proposed approaches in details. We have conducted extended simulations to investigate the performance of the proposed schemes. The experimental results are described in Section 4. Section 5 concludes our work.

II. Preliminaries

A. Related Work

Since ZigBee is a particular type of mobile ad hoc networks (MANETs), let us start with MANET addressing protocols. MANET addressing can be done by electing a mobile node as *address agent*, from which all other mobile nodes request their addresses [3]. The address agent maintains address allocation status such as the mapping between allocated IP address and the associated MAC address. It periodically floods address allocation status to the whole MANET and waits renewal confirmations from all nodes that wish to retain their addresses. The agent retrieves all addresses for which corresponding confirmations are not received in time. MANETconf [4] treated the problem of dynamically allocating a unique address to each node as a distributed agreement problem and proposed adapting a distributed mutual exclusion algorithm to MANET address configuration. Some approach partitions address space (using binary split) among nodes so that each node can configure new node independently [5]. These schemes are designed to deal with problems like node failure, message loss, node mobility, network partitioning and merge, and multiple concurrent initiations of the protocol. The ultimate goal is to ensure address uniqueness in despite of these problems, as well as to minimize the amount of unaddressed nodes. Refer to [6], [7] for a comprehensive survey of current development of address configuration in MANETs.

Conventionally, MANET addressing methods view routing as an independent issue. They are not designed to assist routing tasks except for the guarantee of address uniqueness. In contrast, DAAM in ZigBee networks provides not only unique identification to every device, but also adequate routing information for every possible packet delivery path. The last property eliminates the need for an independent routing protocol.

Address space in [8] is defined by an n-dimensional coordinate system. Each coordinate, expressed as an n-tuple of integers, is considered a logical address. Once configured, a parent node assigns to a child node joining the network an unallocated logical address such that addresses of these two nodes differ in only one element in the tuple: the element in the child's address is one larger than that in the parent's. In this manner, the number of children nodes that each node can have is limited by n. Each node should also inform neighboring nodes which addresses have already been assigned by it, as multiple nodes may be eligible to assign the same address. The issue with this approach is that it imposes an n-dimensional mesh structure on the network, which is too complicated for some PAN applications. We focus on tree structure in the rest of this paper.

For its simplicity and regularity, tree structure has been proposed for routing [9] and multicasting [10] in MANETs. Tree structure also has been used for data gathering in wireless sensor networks (WSNs), where all sensor nodes push their sensing data following a data gathering tree [11], [12], [13] to the root node called sink. If ZigBee is employed to implement a WSN, the sink is logically the ZC while other sensor nodes are ZRs and ZEDs. For tree-based ZigBee networks, Ding et al. [14] have studied how to reduce the number of rebroadcast nodes on broadcasting a data to the whole network. Yet another research issue in tree-based ZigBee networks is to schedule each

ZR's beacon frame transmission to avoid potential beacon collisions [15], [16], [17].

B. ZigBee Basics

The topological parameter associated with DAAM is a collection of three integer variables:

- Lm: the maximum depth value of the tree.
- Cm: the maximum number of children of a ZC/ZR.
- Rm: the maximum number of children of a ZC/ZR that can be ZRs.

According to ZigBee specification, the ZC is at depth 0 and devices at depth Lm can only be ZEDs, not ZRs. Although Lm, Cm, and Rm are all ranged from 0 to 14, their values are not independent as some value combinations are meaningless and some others require more addresses than allowed.

When operating in beacon-enabled mode, a ZC/ZR periodically broadcasts beacon frames to announce its presence and disclose related information. Any device should first scan for beacons before joining a PAN. The collected beacon information is used to build a neighbor table. The device then sends Association Request frames to a ZC/ZR that has the minimal depth value in the neighbor table. If the request is granted, an Association Response frame containing an allocated address (called *short address*) is sent back to the device requesting association. The procedure for a ZC/ZR to allocate an address to a device requesting association is detailed in Fig. 1. Note that an FFD can be configured as a ZED if its parent has no room to accommodate child ZRs.

DAAM's addressing is hierarchical in the sense that any subtree possesses a block of consecutive addresses. Let P be a ZC/ZR located at depth d and D be a child of P. If D is a ZR, the subtree rooted at D is allocated *Cskip*(d) sequential addresses, where *Cskip*(d) is defined as [2]

$$\textit{Cskip}(d) = \begin{cases} 1 + \mathrm{Cm} \times (\mathrm{Lm} - d - 1) & \text{if } \mathrm{Rm} = 1, \\ \frac{1+\mathrm{Cm}-\mathrm{Rm}-\mathrm{Cm}\times\mathrm{Rm}^{\mathrm{Lm}-d-1}}{1-\mathrm{Rm}} & \text{otherwise.} \end{cases}$$

With DAAM's hierarchical addressing, routing can be performed without consulting any routing table. Suppose a ZC/ZR at depth d with address A receives a packet destined for address $D \neq A$. If $A < D < A + \textit{Cskip}(d-1)$, this packet is for some node in the subtree rooted at A and should be passed to the child with address $A + 1 + \lfloor (D - A - 1)/\textit{Cskip}(d) \rfloor \times \textit{Cskip}(d)$. If D is not within the specified range, the packet should be passed to A's parent. [14]

Therefore, the tree structure used by DAAM for address configuration also serves the purpose of routing. The routing path between any two devices is along the tree. This design is justifiable if ZigBee is to implement WSNs, where network traffic mostly flows into or comes from the sink. In contrast, network traffic in MANETs is conventionally assumed peer-to-peer, calling for MANET routing protocols that suit best for that traffic type.

// $d(P)$ denotes the depth value of P
// $A(P)$ denotes the network address of P
// $N_r(P)$ denotes the number of P's children that are ZRs
// $N_e(P)$ denotes the number of P's children that are ZEDs
// $\textit{Cskip}(d) = \begin{cases} 1 + \mathrm{Cm} \times (\mathrm{Lm} - d - 1) & \text{if } \mathrm{Rm} = 1, \\ \frac{1+\mathrm{Cm}-\mathrm{Rm}-\mathrm{Cm}\times\mathrm{Rm}^{\mathrm{Lm}-d-1}}{1-\mathrm{Rm}} & \text{otherwise.} \end{cases}$

if $d(P) = \mathrm{Lm} - 1$ **and** $N_e(P) < \mathrm{Cm}$ **then**
  $N_e(P) \leftarrow N_e(P) + 1$ // D can only be a ZED
  allocate D the address $A(P) + N_e(P)$
else if D is an FFD **and** $N_r(P) < \mathrm{Rm}$ **then**
  $N_r(P) \leftarrow N_r(P) + 1$ // accommodating D as a ZR
  allocate D the following address
    $A(P) + \textit{Cskip}(d(P)) \times (N_r(P) - 1) + 1$
else if $N_e(P) < \mathrm{Cm}$ **then**
  $N_e(P) \leftarrow N_e(P) + 1$ // accommodating D as a ZED
  allocate D the following address
    $A(P) + \textit{Cskip}(d(P)) \times \mathrm{Rm} + N_e(P)$
else
  // no room for D
end if

Fig. 1. DAAM: The procedure for ZC/ZR P to allocate an address to a device D

A ZigBee device may initiate its own removal from the network by performing a leave procedure. If the device is a ZED, the leave procedure involves a unicast of a leave command frame from the device to its parent. If the device is a ZC/ZR, it broadcasts the leave command frame to inform all its neighbors of its leaving. The ZC/ZR may optionally instruct all of its children to also leave the network, and that instruction should be propagated downward to all of the device's descendants. Any device that leaves the network voluntarily or involuntarily may attempt to associate with other ZC/ZRs by performing a *rejoin* procedure. We define a rejoin to be *seamless* if the device is able to retain its short address after its rejoin. With DAAM, however, seamless rejoins are impossible as device's locations in the tree must change after the rejoin procedure. This limitation may be a problem for applications that demand an unchanging address for correct executions.

C. Problem Statement

A major weakness associated with DAAM is the lack of flexibility. In DAAM, the ZC's last

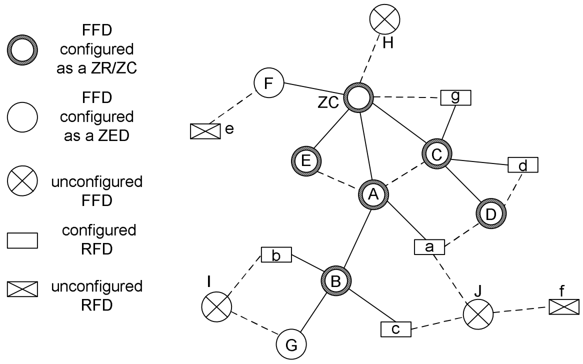

Fig. 2. A ZigBee tree with Cm = 4, Rm = 3, and Lm = 3. We have $S_1 = \{J\}$, $S_2 = \{I, e, f\}$, and $S_3 = \{H\}$. Solid lines stand for established associations while dashed lines are potential links.

child possesses the highest usable address, which is $\mathit{Cskip}(0)\times\text{Rm}+\text{Cm}-\text{Rm}$. Addresses higher than this are not used by DAAM. Therefore, if the topological parameter is not set appropriately, some address space will be wasted. This may lead to *address shortage*, i.e., the amount of usable addresses does not suffice for all devices in the network. The address shortage problem differs from the room shortage problem in that the former is purely due to short supply while the latter can be contributed by poor utilization of addresses. Even if the possibility of address shortage is precluded, room shortage may still occur to a device for one of the following causes:

- The device cannot reach any FFD within the communication range.
- The device can reach some FFDs, but none of them are ZRs.
- The device can reach some ZRs, but none of them is able to allocate an address to this device.

In the following, we preclude the possibility of address shortage and discuss the source of room shortage. Let us first define some notations to ease subsequent discussion:

- *UnC*: the set of unconfigured devices.
- *ZED*: the set of ZEDs.
- *ZR*: the set of ZRs.
- ZR_e: the set of ZR that are still able to allocate addresses to devices requesting associations.
- $N_F(d)$: For any device d, the set of FFDs that are within d's communication range.

The set *UnC* can be partitioned into three subsets, each corresponding to one of the aforementioned causes:

- $S_1 = \{d \mid N_F(d) = \emptyset\}$.
- $S_2 = \{d \mid N_F(d) \neq \emptyset \wedge N_F(d) \cap ZR = \emptyset\}$.
- $S_3 = \{d \mid N_F(d) \cap ZR \neq \emptyset \wedge N_F(d) \cap ZR_e = \emptyset\}$.

TABLE I
POSSIBLE SOURCES OF UNCONFIGURED DEVICE SETS

	FFD density	P1	P2
S_1	Yes	No	No
S_2	No	Yes	No
S_3	No	No	Yes

Figure 2 shows an example of ZigBee tree with some instances of these sets illustrated. Note that F is not configured as a ZR because when it attempts association with the ZC, the ZC already has three child ZRs and therefore can only accommodate F as a ZED. G is neither a ZR because it is at depth Lm. H does not even associate with the ZC, as the ZC has no room to accommodate any child when H attempts association with it.

After examining these instances, it is not difficult to identify that the room shortage problem is closely related to the following two protocol properties associated with DAAM:

P1: The possibility of FFDs being configured as ZEDs.

P2: The limitation on the amount of devices allowed to associate with a ZC/ZR.

The size of S_1 depends on the density of FFDs, a factor irrelevant to protocol design. In contrast, S_2 and S_3 are specific to these protocol properties. P2 is the only reason why $ZR \neq ZR_e$ and therefore the only source of S_3. For each device $d \in S_2$ and each $e \in N_F(d)$, we have either $e \in ZED$ or $e \in UnC$. The condition $e \in ZED$ is due to property P1. The other condition $e \in UnC$ is equivalent to $e \in S_1 \cup S_2 \cup S_3$, implying that any possible cause of addressing failure is also an indirect source of S_2. Table I summarizes possible sources of unconfigured device sets (S_1, S_2, and S_3), excluding the indirect source of S_2.

The routing detour problem also relates to the aforementioned protocol properties. To explain, we define a *potential path* for a device d to be a series of potential links between FFDs,

$$\{(d, f_1), (f_1, f_2), \cdots (f_{n-1}, f_n)\},$$

where $F = \{f_1, f_2, \ldots, f_{n-1}\}$ is a set of FFDs and f_n is the ZC. For example, $\{(g, C), (C, \text{ZC})\}$ and $\{(g, \text{ZC})\}$ in Fig. 2 are two potential paths starting from g. A potential path is not necessarily the routing path from d to the ZC; it becomes that only if (1) all the involved FFDs are ZRs and (2) each potential link ends up with an association. The routing detour problem occurs to d when none of d's shortest potential paths is able to serve as a routing path from d to the ZC. For each such path, the following condition holds when d requests association with the network

$$\exists f_i \in F : f_i \in ZED \vee f_i \in UnC \vee (f_i \in ZR \wedge f_i \notin ZR_e).$$

The source of the condition $f_i \in ZED \vee f_i \in UnC$ is exactly that of S_2 while $f_i \in ZR \wedge f_i \notin ZR_e$ comes from the same reason why S_3 exists.

Note that the routing detour problem concerns only routing paths from a non-ZC device to the ZC. Other paths in a tree-structured network certainly may also encounter routing detours. In particular, tree-routing rule demands that the routing path between two non-ZC devices must go along the tree even if there exists a series of potential links connecting them that is shorter (in terms of hop count) than the path along the tree. However, routing detours imposed by the tree-routing rule is inherent. In contrast, the routing detour problem of our concern may occur due to side effects of addressing protocols.

In parallel with our work, ZigBee Alliance has recognized the room shortage problem. As a remedy, the latest version of ZigBee specification (ZigBee2007 [2]) provides an optional addressing scheme: Stochastic Address Assignment Mechanism (SAAM). SAAM discards the use of *Cskip*, so addressing with SAAM is no longer hierarchical. ZigBee devices randomly and independently select their network addresses, and make extra efforts to detect and resolve address conflicts, which occur when two or more devices select an identical network address. To detect address conflicts, each device should maintain a network address map table (functionally similar to ARP cache found in Internet hosts) in addition to neighbor table. To resolve address conflicts, a broadcast is required to inform every device of a conflict, and a conflicting device may need to rejoin the network to obtain a new address. These activities may be costly in some environments. Moreover, tree-based routing is no longer feasible with SAAM. An independent routing protocol such as AODV [18] should be employed to deliver packets among devices. Our approaches differ from SAAM in that ours need neither an additional routing protocol nor detection and resolving of address conflicts.

III. Flexible Address Configuration

The goal of this research is to devise alternative addressing methods for ZigBee networks. These methods are required to guarantee address uniqueness and, as an integrated part of the addressing scheme, form a tree structure for routing as well while alleviating room shortage and routing detour problems.

Our proposal assumes that ZC acts as an address configuration server (ACS), which manages an address pool for entire network. An ACS is functionally equivalent to and can be implemented as a conventional DHCP server. It assigns unallocated addresses to devices on an on-demand basis. Each ZC/ZR is required to have adequate storage space to keep its routing table.

A. Centralized Stateful Address Configuration (CSAC)

Since ZC is the ACS, any one-hop neighbor of the ZC can acquire an address directly from the ZC on its association without difficulty (assuming no address shortage). A device that does not have a direct link to the ZC, however, cannot obtain an address directly from the ZC. When such a device attempts to associate with some ZR, the ZR should request an address from the ZC on behalf of the device. CSAC introduces two message types for this purpose: Address Request and Address Response. Detailed procedure follows.

- A ZigBee device attempts association with the network by sending Association Request to a neighboring ZR.
- On receiving the request, the ZR becomes the *proxy ZR* of the device. The proxy ZR then sends an Address Request message to the ZC on behalf of the device requesting association.
- The request message is delivered hop-by-hop to the ZC (We will discuss how to realize such a upward routing shortly.)
- The ZC allocates an unused address from the address pool and sends it to the proxy ZR by responding with an Address Response message.
- The response is delivered hop-by-hop to the proxy ZR (The issue concerning this downward routing will be addressed later.)
- The proxy ZR extracts the address from Address Response and sends it to the device attempting association by replying an Association Response.

Note this procedure is compliant to ZigBee's association procedure from the end device's point of view. With CSAC, FFDs are all ZRs while RFDs are all ZEDs. In contrast, FFDs may be degraded to ZEDs with DAAM.

Since CSAC no longer binds addresses to tree locations, it needs a new mechanism for routing. We shall now discuss how a tree-based routing can still be achieved. This relies on the following three properties.

Property 1: Each ZR/ZED sets its default route to its parent.

Property 2: Each ZC/ZR keeps one routing entry for each of its descendants (i.e., a host-specific route for each descendant). This entry points to the right next-hop device on the unique path that connects the ZC/ZR with the corresponding descendant.

Property 3: Each device contains no other routing entry.

It is not hard to see that these three properties together guarantee successful operations for all possible networking scenarios:

- A device can send packets to any of its ancestors with default routes.
- A device can send packets to any of its descendants with host-specific routes.
- A device can send packets to any other devices by first delivering them to the nearest common ancestor of the source and destination (with default routes), from which the packets are then delivered to the destination with host-specific routes.

CSAC ensures the first property by requiring each device to take its parent as the default gateway upon successful association. To retain the second property, each device is required to initiate a route update procedure after it has been configured with an address, say, A_d. The procedure is described as follows.

- The device sends a Route Update message destined for the ZC.
- When the ZC or any halfway ZR P receives Route Update from its child C, P first creates a host-specific route in its routing table for address A_d with next-hop address set to C's address. P then forwards the update message to its parent if P is not the ZC.

In this way, the path from the ZC to each associated device can be created. The procedure also implies that the size of routing table in a ZC/ZR is proportional to the number of associated devices residing in the subtree rooted at the ZC/ZR.

With CSAC, seamless rejoin is likely since CSAC does not bind addresses to tree locations. However, the routing path pertaining to the device's previous location is no longer accessible after the rejoin procedure. The device should therefore perform a *route maintenance* process, which consists of a route update procedure and a *route removal* procedure. The route update procedure creates a new routing path from the ZC to the device while the route removal procedure removes obsolete routing entries residing in ZRs along the previous routing path. Fortunately, nodes common to the new and the previous paths are not involved in the process. We need only propagate a Route Update message all the way to the joint node of the previous and the new routing paths, from which a Route Removal message is then initiated and propagated downward to the end point of the previous

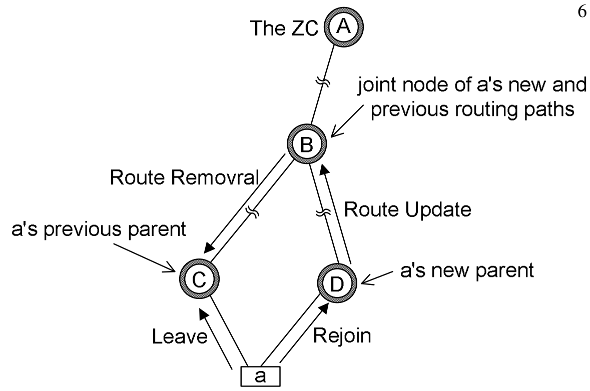

Fig. 3. Propagation of Route Update and Route Removal messages after node a rejoins the network and becomes D's child device.

// $d(P)$ denotes the depth value of P
// $N_r(P)$ denotes the number of P's children that are ZRs
// $N_e(P)$ denotes the number of P's children that are ZEDs

if P has been configured through CSAC **then**
  allocate D an address by running CSAC
else if D is an FFD **and** $d(P) < \text{Lm} - 1$ **and** $N_r(P) < \text{Rm}$ **then**
  $N_r(P) \leftarrow N_r(P) + 1$ // accommodating D as a ZR
  allocate D a ZR address as defined by DAAM
else if $d(P) \leq \text{Lm} - 1$ **and** $N_e(P) < \text{Cm}$ **then**
  $N_e(P) \leftarrow N_e(P) + 1$ // accommodating D as a ZED
  allocate D a ZED address as defined by DAAM
else // room shortage in case of DAAM
  allocate D an address by running CSAC
end if

Fig. 4. HAC: The procedure for ZC/ZR P to allocate an address to a device D

path to remove relevant routing information. Refer to Fig. 3 for an illustration.

B. Hybrid Address Configuration (HAC)

HAC is designed to reduce storage cost incurred by CSAC while retaining flexibility to a certain degree. HAC uses DAAM by default, and invokes CSAC only for devices that cannot be configured through DAAM. As mentioned, only addresses in $[0, \mathit{Cskip}(0) \times \text{Rm} + \text{Cm} - \text{Rm}]$ are reserved by DAAM. The rest can therefore be utilized by CSAC.

When a ZC/ZR P receives an association request from a device D, P uses the procedure shown in Fig. 4 to allocate an address to D. Note that once P has been configured through CSAC, it can only use CSAC to allocate an address to D because P's address is outside the scope of DAAM's address space. If P has been configured through DAAM, it first attempts using DAAM to allocate D's address. P uses CSAC only when it encounters room shortage

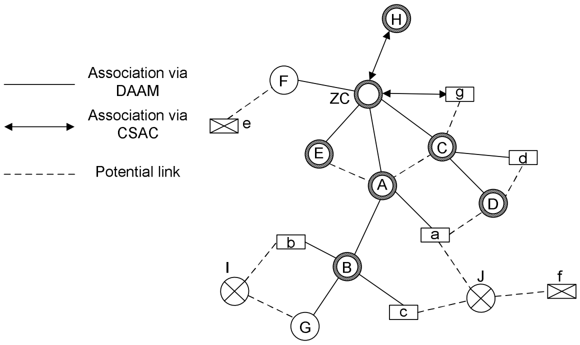

Fig. 5. Result of running HAC on the tree shown in Fig. 2 with Cm = 4, Rm = 3, and Lm = 3.

problem with DAAM. As a result, there are two types of ZRs in HAC: one configured via DAAM (called *D-ZR*) may execute both DAAM and CSAC protocols while the other configured via CSAC (called *C-ZR*) executes only CSAC. If a device is configured by means of CSAC, all its ancestors, C-ZR or D-ZR, should create associative host-specific routes for it as a result of executing the route update procedure. Consequently, the size of routing table in a ZC/ZR is proportional to the number of associated descendants that are configured via CSAC. This is the reason why D-ZR uses DAAM with priority: such strategy reduces storage cost.

The aim to save storage is also reflected by the following design philosophy. When an FFD cannot be configured as a ZR via DAAM, we may accommodate it either as a ZED through DAAM or as a C-ZR using CSAC. In Fig. 4, the former treatment takes precedence over the latter. While an alternative design that attempts these two options in reverse order may increase the ratio of associated devices, the adopted design is more likely to reduce storage cost.

Figure 5 shows the result of running HAC on the same tree shown in Fig. 2 with Cm = 4, Rm = 3, and Lm = 3. Note that FFD H now is configured as a C-ZR, and the routing detour problem occurring to g is resolved. However, F and G are still configured as ZEDs because the procedure in Fig. 4 prefers using DAAM to accommodate them. If CSAC is used instead, these two devices can be ZRs and devices I and e can therefore join the tree (Fig. 6).

The routing rule used by HAC is also hybrid. C-ZRs follow the routing rule of CSAC. For D-ZRs, whether a packet should be handled by DAAM's or CSAC's rule depends on the destination address. If the destination address is within the scope of DAAM's address space, DAAM's rule applies. Otherwise, CSAC's routing rule is in effect.

Unlike CSAC, HAC may not support seamless

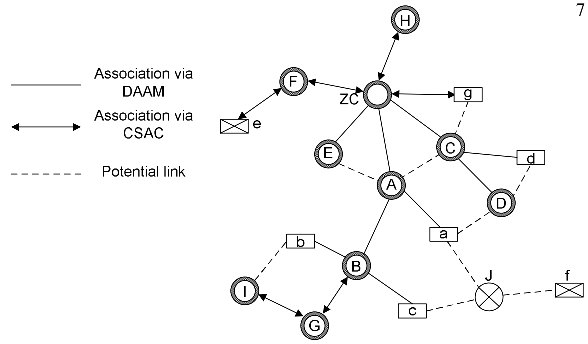

Fig. 6. The same as Fig. 5 except that CSAC is used when DAAM has no room to accommodate FFDs as ZRs.

TABLE II
CONTENTS OF ROUTE MAINTENANCE PROCESS PERFORMED BY DEVICES USING HAC.

Previous parent	New parent	
	D-ZR	C-ZR
D-ZR	None	Route update only
C-ZR	Route removal only	Route update & route removal

rejoins. A device can retain its short address after the rejoin procedure only if both of its previous and current parents are C-ZRs. It must use a different address in other cases. The contents of the route maintenance process in HAC are also different from that in CSAC. Depending on the types of the previous and the new parents of the rejoining device, the route update and the route removal procedures can be performed separately. Refer to Table II for the rule governing the contents to be performed.

C. Router-Based Address Configuration (RBAC)

One drawback of CSAC comes from the additional communication cost between proxy ZRs and the ACS for address allocations. This cost can be reduced if Proxy ZRs own some spare addresses so that they could locally grant address requests without communicating with the ACS. This idea motivates RBAC.

RBAC is similar to CSAC. It partitions the whole address space into fixed-size blocks, which are the basic unit allocated to ZC/ZRs. When a proxy ZR receives an association request, the proxy ZR sends Address Request to the ACS if (and only if) the association request is issued by an FFD. When the ACS receives the address request, it allocates an address block instead of a single address to the proxy ZR. The proxy ZR then informs the FFD of the block. The first address in the block is for the FFD and the rest are spares. After the association is completed, the FFD becomes a proxy ZR and can locally allocate spare addresses to RFDs that request associations with it.

TABLE IV
POSSIBILITIES OF UNCONFIGURED DEVICE SETS (S_1, S_2, AND S_3) WITH EACH PROTOCOL

	DAAM	CSAC	HAC	RBAC
S_1	Yes	Yes	Yes	Yes
S_2	Yes	No	Yes	No
S_3	Yes	No	No	Yes

As all ZEDs associated with the same ZR share an address block and the block size is known to every ZR, only ZRs need to initiate the route update procedure after their associations. ZEDs need not perform the procedure. As a result, each ZC/ZR keeps routing records only for ZRs residing in the subtree rooted at it. In contrast, a ZC/ZR with CSAC needs store addresses of *all devices* (both ZRs and ZEDs) in the same subtree. Therefore, the storage cost of RBAC is lower than that of CSAC.

To facilitate routing task, the size of each block is set to a power of two. With this we can define a bit mask that indicates which bit in the address field should agree for two addresses being in the same address block. This notion is exactly the same as subnet mask used in subnetting IP networks. With this bit mask, a ZC/ZR can easily determine whether the destination address of a received packet is a ZR or ZED. If the destination is a ZED, the ZC/ZR can also determine the address of the ZR with which the ZED associates. The packet can then be forwarded to the destination ZR by consulting routing tables.

All ZRs can conduct a seamless rejoin by executing the same route maintenance process as CSAC. For ZEDs, however, seamless rejoins are unlikely. The only exception is when a ZED rejoins the tree through the same parent ZR, which may happen when they all perform the rejoin procedure due to the leaving of one of their common ancestors. The address of the ZED must change in other cases.

D. Protocol Properties Discussions

We shall now analyze whether these three protocols suffer from room shortage and routing detour problems. FFDs can only be ZRs with CSAC or RBAC, so these two protocols do not have property P1. Unfortunately, HAC inherits P1 from DAAM since it uses DAAM with priority. CSAC neither has property P2, since it does not limit the number of devices associated with a ZC/ZR. HAC does not inherit P2 from DAAM due to the introduction of CSAC as a remedy. P2 holds for RBAC since the address block size confines the number of ZEDs (but not ZRs) allowed to associate with a ZC/ZR. Table III summarizes protocol properties.

From Tables I and III, possibilities of unconfigured device sets (S_1, S_2, and S_3) with each protocol can be obtained (Table IV). It turns out that both the room shortage and routing detour problems are most serious with DAAM and least with CSAC. The behaviors of HAC and RBAC should be in-between. In particular, CSAC does not suffer from the routing detour problem as S_2 and S_3 are the only cause of the problem. Therefore, the routing path of each device to the ZC must be one of its shortest potential paths. With HAC or RBAC, however, the routing detour problem may still arise.

Recall that seamless rejoin is impossible with DAAM. CSAC supports seamless rejoin as it no longer binds addresses to tree locations. For HAC, only devices that change their parents from one C-ZR to another C-ZR can retain their addresses. The frequency of such case hence depends on the population of C-ZRs. For RBAC, all ZRs can retain their addresses while ZEDs hardly can. Therefore, the frequency of seamless rejoins depends on the ratio of ZRs to ZEDs.

IV. EMPIRICAL RESULTS

We conducted extended simulations to investigate the performance of proposed schemes. We assumed an 1000×1000 m^2 deployment field, within which 200 to 1000 ZigBee devices were uniformly deployed at random. An additional device acting as the ZC was placed at the center of the deployment field. All devices had a communication range of 100 m. The ratio of FFDs to RFDs was one to one. We considered three settings for the topological parameter, which respectively stand for *tall* (Cm=4, Rm=2, Lm=14), *regular* (Cm=12, Rm=4, Lm=7), and *flat* (Cm=14, Rm=8, Lm=5) trees. For HAC, the topology setting was fixed to (Cm=12, Rm=5, Lm=6) for a fixed address-space partition between DAAM and CSAC. The block size for RBAC was set to 8. Each parameter setting was repeated 100 times to obtain average results.

A. Percentage of configured devices

The first metric we measured is the percentage of devices that were successfully configured with addresses. Both CSAC and RBAC performed the best, followed by HAC and then DAAM. Fig. 7 shows the result.

Since S_1 is the only source of configuration failure with CSAC, the curve of CSAC here also indicates a relative amount of S_1 devices. We found that FFD density did not suffice for an 100% configuration ratio until over 500 devices were deployed (note only half of them were FFDs).

TABLE III
SUMMARY OF PROTOCOL PROPERTIES

	DAAM	CSAC	HAC	RBAC
Forcing FFDs to be ZEDs? (P1)	Yes	No	Yes	No
Limiting associable devices? (P2)	Yes	No	No	Yes

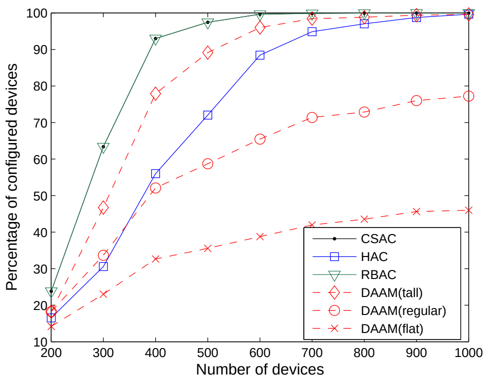

Fig. 7. Percentage of devices being successfully configured

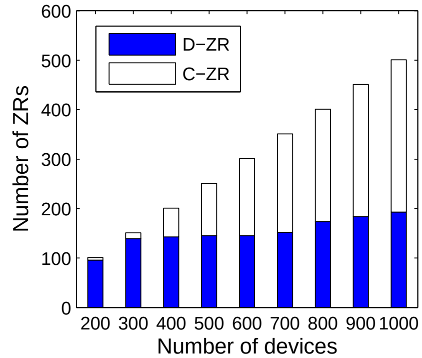

Fig. 8. Number of D-ZRs and C-ZRs

Besides S_1, only S_3 contributes to RBAC's addressing failures. Therefore, the performance gap between CSAC and RBAC can be regarded as the degree of S_3's impact on RBAC's performance. Since RBAC's performance is hardly distinguishable from CSAC's, we deduce that the impact is negligible.

DAAM's failures were contributed by S_1, S_2, and S_3. S_1's contribution diminished when over 500 devices were deployed. However, for DAAM with the regular-tree and the flat-tree settings, the configuration ratio only slightly increased with increasing device population. The significant performance gap between them and CSAC was mainly due to their relatively small settings on Lm. A small Lm will result in a large set of $FFD \cap UnC$ and hence a large set of S_2. This also explains why DAAM with the tall-tree setting had a better performance.

When fewer than 400 devices were deployed, HAC performed worse than DAAM(regular). This can be explained by their Lm settings. When 400 or more devices were deployed, however, HAC overtook the counterpart, which was contributed by HAC's ability to accommodate extra devices with the introduction of C-ZRs. The benefit of C-ZRs became more significant when more devices were set, since the ratio of C-ZRs to D-ZRs increased with the number of deployed devices (Fig. 8).

B. Hop Count

Hop count measures the path length between two potential packet-exchanging nodes, which is affected by tree depth as well as the routing detour problem. Fig. 9 shows the average hop count from every device to the ZC. The average hop count between each pair of nodes in the tree is similar to that given by Fig. 9 and is not shown here. The result indicates that DAAM with both regular- and flat- tree settings had relatively low hop-count values. However, the result with the tall-tree setting was the worst when 400 or more devices were deployed.

The superiority of DAAM with both regular- and flat- tree settings comes from its ability to confine tree depths. In contrast, all proposed approaches place no limitation on tree depths, leading to high hop-count values. The difference between Figs. 7 and 9 reveals that DAAM is able to trade configuration ratios for hop counts by changing the value of Lm. A problem with that ability is the lack of automatic way to determine a suitable value for Lm that maximizes configuration ratio while minimizing depth value to the greatest possible extent.

CSAC and RBAC performed identically. Their hop-count results rose initially with increasing nodes, but slightly fell with more nodes. The reason for these results is due to the following two competing factors:

- A tree formed by a few nodes is expected small in scale. So the average path length is also small. On the other hand, a large-scale tree may contain many nodes that have long paths to the ZC, increasing the average value.
- Upon joining a tree, every device seeks the shortest path (in terms of hop count) from it to the ZC and selects one of its neighbors that

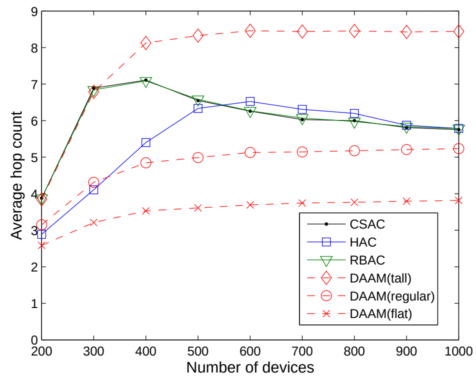

Fig. 9. Average hop count from every device to the ZC

leads to this path as its parent. The selected path is *optimal* if the path length is exactly $\lceil d/r_t \rceil$, where d is the distance between the node and the ZC while r_t is the communication range. If node density is sufficiently high, devices are likely to find and select optimal paths. Otherwise, many nodes are forced to join the tree with sub-optimal paths connecting them.

When 300 or fewer nodes were deployed, the first factor dominated the results. But it was overtaken by the second factor when more than 400 devices were deployed. The first factor can also explain why HAC outperformed CSAC and RBAC when fewer than 600 devices were set: HAC accommodated fewer nodes than either CSAC or RBAC did in that range. As the number of C-ZRs increased, HAC could accommodate as many devices as CSAC or RBAC did, and hence the superiority of HAC disappeared. HAC's performance roughly coincided with that of CSAC and RBAC thereafter.

C. Storage Cost

All proposed approaches demand storage to keep routing tables. We therefore took the size of routing tables as a gauge of storage cost. Fig. 10 displays average-case and worst-case storage costs associated with all the proposed methods. As expected, CSAC had the highest storage cost among all under all circumstances. Its average cost declined with increasing device population simply because high storage cost was amortized over all devices. It can be seen that RBAC halved CSAC's storage cost in all cases. This result is reasonable as devices were uniformly distributed and the radio of FFDs to RFDs was one to one.

HAC's performance was the best with 400 or fewer devices. Recall that in HAC, a ZC/ZR's routing table

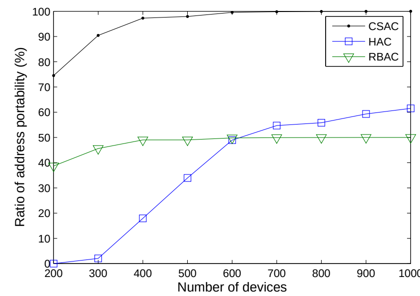

Fig. 11. Ratios of seamless rejoins

size is proportional to the number of its descendants that are configured via CSAC. Since all descendants of a C-ZR must be configured via CSAC, HAC's storage cost could be low only with few C-ZRs. Considering the observation that the amount of C-ZRs increased with device quantity (Fig. 8), the raise of HAC's storage cost when 500 or more devices were set can be expected.

Knowing the exact storage demand is important for real implementations. All routing entries except for the default route are host-specific routes. This implies that each routing entry can be represented by a pair of short addresses, one for the destination and the other for the next-hop addresses. Since a short address takes 16 bits in length, a routing entry calls for four bytes of storage space in total. Therefore, concerning the worst case, 4 KB of storage suffices even for a network consisting of 1000 ZigBee devices. This amount of storage demand is within the capacity of many off-the-shelf ZigBee products. For example, a recent single-chip platform from Freescale [19] provides 96 KB of RAM.

D. Ratios of Seamless Rejoins

We measured the ratio of devices that were able to retain their addresses after rejoining the same network via a different ZR. Since seamless rejoin is impossible with DAAM, only the proposed schemes were examined and compared. For each device that had already joined the ZigBee tree, we sought a ZC/ZR that was different from its current parent yet had the lowest possible depth value to be the device's new parent. We randomly selected one when there were multiple candidates. Fig. 11 shows ratios of seamless rejoins, where each result was averaged over all configured devices.

For CSAC, the ratio was not 100% with low device density simply because some devices failed to rejoin due to the absence of other ZC/ZRs in their

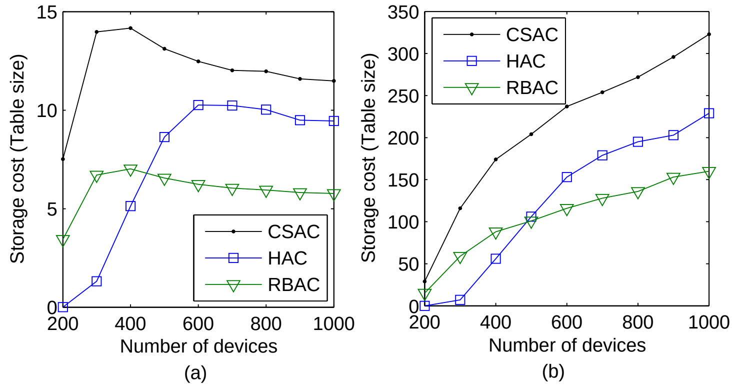

Fig. 10. Size of routing tables. (a) Average-case results (b) Worst-case results

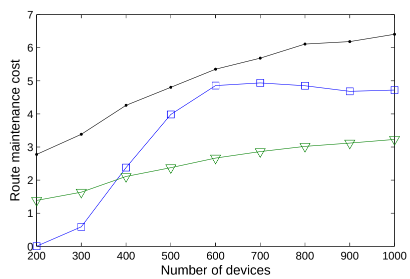

Fig. 12. Average route maintenance costs

neighborhood. Such devices were rare when node density became high. RBAC generally halved CSAC's measures, which is reasonable since only ZRs in RBAC could make a seamless rejoin. HAC's result was basically consistent with the observed ratio of C-ZRs to all ZRs.

A successful rejoin, seamless or not, may demand a route maintenance process to renew relevant routing path. Fig. 12 shows average route maintenance costs for all the proposed schemes. The cost was measured in the amount of messages transmissions required to complete a route maintenance. Generally speaking, costs associated with CSAC or RBAC increased with the number of deployed devices. The curve of HAC's cost roughly coincided with that of its average storage cost (Fig. 10(a)). This was anticipated as for each device that is configured through CSAC, the total storage cost and the route maintenance cost that it incurs are both proportional to the length of the routing path from it to the ZC.

V. Conclusions

This paper has identified the room shortage problem associated with DAAM, the standard address configuration schemes recommend by ZigBee specification. As a remedy, we have considered three alternatives: CSAC, HAC, and RBAC. Extended simulations have been conducted to investigate their performance compared with DAAM. Table V summarizes the result.

If storage (and accompanying route maintenance) cost is the only concern, then DAAM is the only choice. Otherwise, the proposed approaches should be adopted for their ability to alleviate the room shortage problem. CSAC is the most flexible scheme such that it has achieved the highest percentage of configured devices in simulations. However, extra storage in every ZigBee router is required by CSAC, incurring a cost that is directly proportional to the number of associated descendants. HAC aims to reduce the storage cost while retaining flexibility to a certain degree. Its ability to make such a tradeoff has been demonstrated through simulations. RBAC's performance in terms of configuration ratio is hardly distinguishable from that of CSAC. Since the storage cost of RBAC is generally lower than those of CSAC and HAC, it is recommended as the best treatment for the room shortage problem associated with DAAM.

References

[1] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4, 2003.

[2] "ZigBee specification," ZigBee Alliance, Jan. 2008. [Online]. Available: http://www.ZigBee.org/

[3] M. Güneş and J. Reibel, "An IP address configuration algorithm for Zeroconf mobile multihop ad hoc networks," in *Proc. Intl Workshop on Broadband Wireless Ad Hoc Networks and Services*, Sophia Antipolis, France, Sept. 2002.

[4] S. Nesargi and R. Prakash, "MANETconf: Configuration of hosts in a mobile ad hoc network," in *Proc. IEEE INFOCOM*, 2002, pp. 1059–1068.

[5] M. Mohsin and R. Prakash, "IP address assignment in a mobile ad hoc network," in *Proc. MILCOM*, 2002, pp. 856–861.

TABLE V

PERFORMANCE COMPARISON AMONG ALL APPROACHES

	DAAM	CSAC	HAC	RBAC
Percentage of configured devices	Low to high	The highest	Mid to high	Also the highest
Hop count	Low to high	Mid	Mid	Mid
Storage/route maintenance cost	None	The highest	Low to high	Mid
Ratio of seamless rejoins	Zero	The highest	Low to mid	Mid

[6] Y. Sun and E. M. Belding-Royer, "A study of dynamic addressing techniques in mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 4, pp. 315–329, 2004.

[7] *Survey of IP Address Autoconfiguration Mechanisms for MANETs*, IETF Internet Draft, Oct. 2007.

[8] G. Bhatti and G. Yue, "Structured addressing scheme for wireless networks," U.S. Patent 0 060 134, Mar., 2007.

[9] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Proc. IEEE ICC*, June 1997, pp. 376–380.

[10] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Proc. 5th ACM/IEEE Int'l. Conf. on Mobile Computing and Networking*, August 1999, pp. 207–218.

[11] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proc. 22nd Int'l Conf. on Distributed Computing Systems Workshops*, 2002, pp. 575–578.

[12] V. Annamalai, S. K. S. Gupta, and L. Schwiebert, "On tree-based convergecasting in wireless sensor networks," in *Proc. IEEE WCNC*, Mar. 2003, pp. 1942–1947.

[13] Y. Yu, B. Krishnamachari, and V. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 244–255.

[14] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava, "Tree-based data broadcast in IEEE 802.15.4 and ZigBee networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 11, pp. 1561–1574, Nov. 2006.

[15] Y.-C. Tseng and M.-S. Pan, "Quick convergecast in ZigBee/IEEE 802.15.4 tree-based wireless sensor networks," in *ACM Intl Workshop on Mobility Management and Wireless Access*, Terromolinos, Spain, Oct. 2006, pp. 60–66.

[16] A. Koubâa, M. Alves, M. Attia, and A. V. Nieuwenhuyse, "Collision-free beacon scheduling mechanisms for IEEE 802.15.4/Zigbee cluster-tree wireless sensor networks," in *7th Int'l Workshop on Applications and Services in Wireless Networks*, Santander, Spain, May 2007.

[17] L.-H. Yen, Y. W. Law, and M. Palaniswami, "Risk-aware beacon scheduling for tree-based ZigBee/IEEE 802.15.4 wireless networks," in *Proc. 4th Int'l Wireless Internet Conf.*, Hawaii, USA, Nov. 2008.

[18] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Los Alamitos, California, USA, Feb. 1999, pp. 90–100.

[19] *MC13224V: Advanced ZigBee-Compliant Platform-in-Package (PiP) for the 2.4 GHz IEEE 802.15.4 Standard*, 1st ed., Freescale, Nov. 2008.