A Lightweight Network Repair Scheme for Data Collection Applications in ZigBee WSNs

Meng-Shiuan Pan and Yu-Chee Tseng

***Abstract*—Data collection is a fundamental operation in *wireless sensor networks (WSNs)*. In [4], a quick convergecast solution is proposed for data collection in a ZigBee beacon-enabled tree-based WSN. However, it does not consider the network repair issue. When a ZigBee router loses its link to its parent, all its descendants have to rejoin the network. The rejoining procedure is time-consuming and may incur high communication overheads. The proposed network repair scheme consists of a *regular repair* and an *instant repair* schemes. Periodically, the network coordinator can issue regular repair to refresh the network (so as to keep the network in good shape). During normal operations, if a router loses its parent, it tries instant repair to reconnect to a new parent. Our design thus improves over ZigBee in that nodes can continue their operations even during instant repair.**

***Index Terms*—Convergecast, IEEE 802.15.4, reliability, wireless sensor network, ZigBee.**

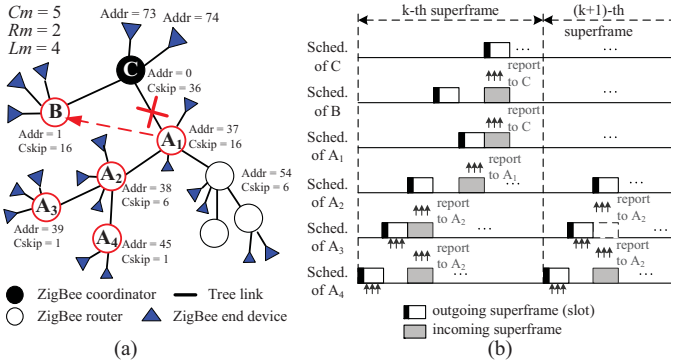

Fig. 1. (a) A ZigBee tree network. (b) A beacon scheduling example for quick convergecast.

I. Introduction

ZIGBEE [5] and IEEE 802.15.4 [2] are widely used in *wireless sensor networks (WSNs)*. Observing that data gathering is a major application of WSNs, [4] presents several beacon scheduling algorithms with low *convergecast latency*. Fig. 1(a) illustrates a ZigBee tree network with one *coordinator* (*sink*), some *routers*, and some *end devices*. Each router is responsible for collecting sensed data from its child routers or end devices and relaying incoming data to the sink. Each router can announce a beacon to start its a superframe, which consists of an *active portion* (called an *active slot*) followed by an *inactive portion*. On receiving its parent router's beacon, a child router/end device has to wake up during the former's active portion and forward its data to the former. The objective is to schedule routers' active portions to minimize the overall convergecast latency. Fig. 1(b) shows a scheduling example. The report latencies of routers A_3 and A_4 are 4 and 5 slots, respectively. The convergecast latency of the network is the longest report latency among all routers.

However, [4] does not consider the network repair issue. In reality, a wireless link may experience short-term or long-term failure. In ZigBee, when a router finds the link to its parent to be broken, it has to reassociate with a new parent and all its descendants need to perform the reassociation procedure. In Fig. 1(a), if link (C, A_1) is broken, totally 7 routers and 14 end devices need to rejoin the network. According to IEEE 802.15.4, an association procedure takes at least one *beacon interval (BI)*. Moreover, a reassociated router needs to find a new active slot. For example, if A_1 selects B as its new parent, it would be better to shift A_1's active slot to the left by two slots to reduce its report latency. This may even trigger its children to re-select their active slots. In addition, checking if a slot is interference-free takes at least one BI. Thus, the total rejoining time is at least two BIs. So the failure of (C, A_1) takes at least 6 BIs to repair. As pointed out in [4], a BI can be up to 4.19 minutes, and a node cannot continue its tasks until it successfully rejoins the network.

General network repair issues have been studied in [1] and [3]. In [3], the proposed reinforcement scheme may cause upstream nodes of a failed link to participate in recovery, thus incurring high communication overheads and convergence delay. In [1], if a node determines its upstream link to be broken, it immediately broadcasts a *search* packet to look for a neighbor with the shortest path to the sink as its new parent. But this may cause loops in some cases. However, these schemes are not designed for ZigBee.

In this paper, we propose a *regular repair* and an *instant repair* schemes to maintain a failure-prone ZigBee network. The former is to periodically refresh the network to keep it in a good shape. It consists of a *tree reformation procedure* and a *slot assignment procedure*. In particular, the latter can reserve some extra address spaces for routers so that they can accept new children when failure occurs. The instant network repair is triggered when a router identifies that the link to its parent is broken. It first executes a *localized reconnecting procedure* to find a new parent among its neighbors by considering their capacities and slots. It then triggers an *address update procedure* to update its descendants' addresses. The instant repair scheme causes no loops and can preserve the convergecast latency of the network. In addition, most nodes can continue to work during instant repair.

Manuscript received March 1, 2009. The associate editor coordinating the review of this letter and approving it for publication was A. Smiljanic.

The authors are with the Department of Computer Science, National Chiao Tung University, Hsin-Chu, Taiwan (e-mail: mspan@csie.nctu.edu.tw, yctseng@cs.nctu.edu.tw).

Y.-C. Tseng's research is co-sponsored by MoE ATU Plan, by NSC grants 96-2218-E-009-004, 97-3114-E-009-001, 97-2221-E-009-142-MY3, and 98-2219-E-009-005, by MOEA 98-EC-17-A-02-S2-0048, and by ITRI, Taiwan.

Digital Object Identifier 10.1109/LCOMM.2009.090481

II. ZigBee's Address Assignment and Superframes

ZigBee assigns devices' network addresses by a distributed scheme. The coordinator determines the maximum number of children (Cm) of a parent router, the maximum number of child routers (Rm) among its Cm children, and the depth of the network (Lm). Each node then computes a $Cskip$ to reserve some address space for its descendants. An example is shown in Fig. 1. The coordinator also defines the superframe structure of a network, which is controlled by *beacon order* (BO) and *superframe order* (SO), where $0 \leq SO \leq BO \leq 14$. These parameters decide the lengths of a superframe, called *beacon interval* (BI), and its active portion, respectively. Given a $(BO - SO)$, a router can choose from 2^{BO-SO} slots as its active portion, called its *outgoing superframe*. Accordingly, its parent's active portion will serve as its *incoming superframe* (since beacons will be received from its parent). Note that when choosing *slots* (i.e., outgoing superframes), neighboring routers' slots should be shifted away from each other to avoid interference. As introduced in [4], two routers have *direct* or *indirect interference* if they are neighbor nodes or they have a common neighbor, respectively. We denote by $N_I(v)$ the set of interference neighbors of router v.

III. The Proposed Network Repair Scheme

Our approach consists of a regular repair and an instant repair parts. The former is triggered less frequently (say, in every hour or in case the coordinator determining that the network is in a very bad shape). Regular repair will construct a fresh new tree and instant repair will try to maintain the current tree. When a tree link is detected to be broken, instant repair will attempt to do a quick fix.

A. Regular Network Repair

The regular network repair consists of two phases: *tree reformation* and *slot assignment*. In the first phase, a ZigBee tree satisfying the constraints of Cm, Rm, and Lm is formed. We adopt the original ZigBee tree formation, so the detail is skipped. After the formation, let $R(v)$ be v's remaining capacity, i.e., the number of child routers that v can accept in the future.

The second phase is a distributed scheme to compute for each router v an active slot $s(v) \in [0, k-1]$, where $k = 2^{BO-SO}$ is the number of available slots, and a *delay index* $d(v)$. Intuitively, $s(v)$ is the slot in a superframe for v to receive data from its children, and $d(v)$ is the delay incurred to transmit these data to the sink. Our scheme follows a *larger-remaining-capacity-first* strategy. Each v will periodically broadcast a HELLO$(s(v), d(v), R(v), l(v), h(v))$ packet with period t_{hello}, where $l(v)$ and $h(v)$ are v's level and the height of the subtree root at v in the current tree, respectively. (*NULL* is used when a variable is unknown yet; the root's level is 1.) The algorithm is triggered by the sink t setting $s(t) = d(t) = k - 1$ and broadcasting an Assign$(s(t), d(t))$ packet. A router v that receives an Assign$(s(p), d(p))$ from another router p will execute the following procedure.

1) If v is *not* p's child, it terminates the procedure.
2) Router v computes the smallest positive integer a such that $(s(p) - a) \bmod k \neq s(u)$ for each $u \in N_I(v)$ and $s(u) \neq NULL$. Then v chooses its slot $s(v) = (s(p) - a) \bmod k$ and sets $d(v) = d(p) - a$. Intuitively, a is the latency from the time when a packet arrives at v to the time when it is sent to v's parent p.
3) Then, v waits for a duration t_{wait}. Recall that each node will periodically broadcast HELLOs, so t_{wait} should be larger than t_{hello} by about an order to ensure that v receives sufficient HELLOs. From these HELLOs, if v finds that $s(v) = s(u)$ for any $u \in N_I(v)$ and one of the following conditions is true, it has to choose a new slot by going back to step 2.
   a) $R(u) > R(v)$.
   b) $R(u) = R(v)$ and $l(u) < l(v)$.
   c) $R(u) = R(v)$, $l(u) < l(v)$, but u's address is smaller than v's address.
4) After t_{wait}, v can finalize its slot and broadcast an Assign$(s(v), d(v))$.

The above slot assignment works in a top-down manner along the ZigBee tree. In step 2, a is the delivery latency incurred by an upward link. So $d(v)$ is a "countdown" of latency from the root toward each leaf of the tree. In fact, since root t enforces that $s(t) = d(t)$, the above procedure guarantees that $s(v) = d(v) \bmod k$ for any v. In addition, $d(v)$ will be used in our instant repair process to ensure loop freedom. In step 3(a), v will yield to u if u has more remaining capacity (because u may accommodate more children when instant repair is needed). In step 3(b), v will yield to u if u is closer to the root (because a node attaching to u will be closer to the root, too).

To prepare for future instant repair, each router v will maintain a *potential parent* set $P(v)$. We let $u \in P(v)$ if all the following conditions are true: (i) $R(u) > 0$, (ii) $d(u) > d(v)$, and (iii) $l(u) + h(v) + 1 \leq Lm$. Note that (ii) is to guarantee loop freedom and to preserve the convergecast latency when instant repair is taken.

After this phase, it is not hard to see that the report latency from v to its parent u is $(s(u) - s(v)) \bmod k = d(u) - d(v)$. The report latency for router v to root t is $L(v) = d(t) - d(v)$. The overall convergecast latency is $L(T) = \max_{\forall v}\{L(v)\}$, where T is the final ZigBee tree.

B. Instant Network Repair

According to IEEE 802.15.4, a device considers the link to its parent to be broken when it does not receive four consecutive beacons from its parent. We will focus on routers losing their parents. (To deal with an end device losing its parent, the original ZigBee procedure already works well.) Our instant repair consists of two phases: *localized reconnection* and *address update*. The former is to instantly reassociate a router to a new parent. Following that, the latter will update its subtree nodes' addresses. The instant repair guarantees no loop and preserves the original convergecast latency of the network.

Localized reconnection is triggered when a router, say, v identifies the link to its parent is broken or v is disconnected by its parent. It involves two steps.

1) It will sort the elements in its potential parent set $P(v)$ according to their levels in an ascending order. Then v

TABLE I
COMPARISON OF REPORT LATENCY $L(T)$.

Rm	3	4	5	6
DSA	53.4	57.6	59.6	61.1
Ours	57.1	62.9	65.6	67.3

TABLE II
COMPARISON OF THE NUMBER OF REASSOCIATIONS.

Blocked routers (%)	1%	2%	3%	4%	5%
ZigBee	29.9	56.9	83.8	112.7	142.7
Ours	6.7	12.1	19.4	24.4	29.2

will extract the first node in $P(v)$ and try to associate with it. If the association succeeds, v will terminate this phase and trigger the address update phase. Otherwise, v will extract the next node in $P(v)$ and repeat again, until $P(v)$ becomes empty.

2) Since the subtree rooted at v cannot be directly attached to any node, v will disconnect all its child routers and end devices. Then v will keep on trying to find a new parent or report the failure status to the upper layer.

Correctness of the above process relies on correctly maintaining the set $P(v)$. With nodes' periodical HELLOs, v can eventually construct its $P(v)$. However, during transition time or instant repair of other nodes, v may try to associate with a u with an incorrect level and v may even have incorrect information about its own subtree height. So after instant repair, a subtree may be too tall. Fortunately, this will not cause problem because the inconsistency will eventually be discovered in our address update phase. A node that can not correctly update its address will be disconnected by its parent, thus triggering the above localized reconnection. Most importantly, our scheme will never change a node's delay index. This leads to Theorems 1 and 2.

Address update will be triggered when router v associates with a new parent and receives a new address. When v's slot arrives, it can announce a beacon using its old network address with a special reserved bit set to indicate that this is for address update. Also, v attaches its children's current addresses in the pending list in the beacon and its new address in the beacon payload. A recipient, say, u will realize that it needs to update its network address. Following the philosophy of IEEE 802.15.4, u will send a *data request command* to v, which will respond an *association response command* containing u's new address. After obtaining a new address, u will wait for v's beacon with v's new network address. If u is a router, it will also recursively execute the address update procedure for its children (if any). However, if u finds that it is already at the bottom level Lm, it will terminate the process and disconnect all its children. The above special beacon (with special bit) can be sent for at most three times to ensure reliability (since four times will be interpreted as missing parent). Although there could exist orphans after instant repair, our scheme maintains two important properties.

Theorem 1: Our scheme ensures that the network is always loop-free.

Proof: Our regular repair ensures that a node's parent always has a larger delay index than itself. In each instant repair, a node's potential parents only contain those with higher delay indices. During instant repair, no node can modify its delay index. Since delay indices are strictly descending along the tree in the downward direction, the network is loop-free. □

Theorem 2: Let T be the tree after a regular repair. The convergecast latency is always bounded by $L(T)$ after each instant repair until the next regular repair is taken.

Proof: While nodes' active slots indicate their timing to collect data, their delay indices indicate the relative timing for children to deliver collected data to their parents. Let p be a router v's parent. Recall that delay indices are always descending. Before T is underwent any change, if v collects some sensing data at slot $i \times k + d(v)$ from its children, where i is an integer and $k = 2^{BO-SO}$, the earliest time for v to send to p is slot $i \times k + d(p)$. After some instant repairs, let p' be v's new parent. Since we guarantees that $d(p') > d(v)$, sensing data collected by v at slot $i \times k + d(v)$ can still be delivered to p' no latter than slot $i \times k + d(p')$. We can inductively prove this property for the parent of p'. Considering the whole network's convergecast latency, it is always bounded by $L(T)$ as long as the network remains connected. □

IV. SIMULATION RESULTS

In our simulations, 300 routers are randomly distributed in a $100^2\pi\ m^2$ circular region and a sink is placed at the center. We set $k = 64$, $Lm = 7$, and the transmission range of routers $= 25\ m$. We first compare our slot assignment algorithm against the DSA algorithm in [4] on $L(T)$ (in unit of slots) with various Rm. The result in Table I indicates that our scheme only slightly increases the convergecast latency. Next, we show the fault-tolerant capability of our instant repair. When there are link failures, we count the number of reassociation procedures appearing in the network, which reflects the number of nodes temporarily leaving the network. We fix $Rm = 5$ and randomly and sequentially mark some routers as *blocked* (a blocked router will miss all links to its current children). Table II shows that our scheme can greatly facilitate network operations even with frequent link failures.

REFERENCES

[1] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications," in *Proc. ACM/IEEE Int'l Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2004.
[2] IEEE Std 802.15.4, 2006.
[3] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 2–16, 2003.
[4] M.-S. Pan and Y.-C. Tseng, "Quick convergecast in ZigBee beacon-enabled tree-based wireless sensor networks," *Elsevier Computer Communications*, vol. 31, no. 5, pp. 999–1011, 2008.
[5] ZigBee specification version 2006, ZigBee document 064112, 2006.