Journal of Network and Computer Applications 46 (2014) 252–263

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Low latency scheduling for convergecast in ZigBee tree-based wireless sensor networks

Meng-Shiuan Pan *, Ping-Lin Liu

Department of Computer Science and Information Engineering, Tamkang University, New Taipei City, Taiwan

A R T I C L E I N F O

Article history:
Received 11 December 2013
Received in revised form
13 May 2014
Accepted 12 July 2014
Available online 24 July 2014

Keywords:
Convergecast
Graph theory
Scheduling
Tree construction
Wireless sensor network
ZigBee

A B S T R A C T

Convergecast is a fundamental operation in many wireless sensor network (WSN) applications. When gathering data, power saving and convergecast latency are two main concerns. This work adopts the ZigBee tree networks to address these two concerns. In a ZigBee tree network, to support energy efficient operations, each node is assigned to an active portion (or say slot). A node wakes up at its slot and its parent's slot to collect data from its children and to report data to its parent, respectively. Then, it can go to sleep to save energy. To support low latency convergecast, in this work, we propose a centralized and a distributed slot assignment schemes for ZigBee tree networks. We observe that when assigning slots, the latency can be further reduced by reconnecting some tree links. More specifically, by the designed rules, a node is allowed to locally modify some of its neighbors' parents, and then the node can be assigned to a better slot that can have the benefit of reducing the node's report latency. Simulation and implementation results show that the proposed schemes can effectively reduce the convergecast latency in ZigBee tree-based WSNs.

1. Introduction

Convergecast is a fundamental operation in many *wireless sensor network* (*WSN*) applications including monitoring (Hayes et al., 2008; Pan et al., 2008a), health care (Huo et al., 2011), dynamic path finding (Chen et al., 2009; Tseng et al., 2006), and smart home (Nguyen et al., 2010; Pan et al., 2008b). It involves a set of nodes in a network to report their sensory data via a *data reporting tree* to the sink. When gathering data, there are two main technical issues, *power saving* and *convergecast latency*. To support energy efficient operations, nodes can periodically sleep and wake up to conserve their power. For the latency concern, the wake-up timings should be carefully designated.

Recently, the ZigBee/IEEE 802.15.4 (ZigBee Alliance; IEEE standard for information technology, 2003) standards are widely used in many wireless sensor platforms. The IEEE 802.15.4 and ZigBee specifications define the physical/link layer protocols and the protocols above the link layer, respectively. The ZigBee network layer supports *star*, *tree*, and *mesh* networks. A *ZigBee coordinator* is responsible for initializing, maintaining, and monitoring the network. In a star network, there is a coordinator, which allows nodes associating to it. In tree and mesh networks, nodes can communicate with each other in a multihop fashion. The network contains one ZigBee coordinator and multiple *ZigBee routers*. A node can join a network by associating with the coordinator or a parent router. In a tree network, the coordinator and routers can periodically announce beacons to start superframes. In ZigBee, each superframe consists of an *active portion* followed by an *inactive portion*. In this work, we define an active portion as a *slot*. After sending a beacon, the beacon sender will keep awake for a slot time. On receiving its parent's beacon, a child node wakes up for a slot time, and can communicate with its parent during this period. However, in a ZigBee mesh network, regular beacons are not allowed. Since sending beacons is an important mechanism to support power management in ZigBee networks, the tree topology is preferred when energy saving is a desirable feature.

In this work, we aim to propose slot assignment schemes for supporting low latency convergecast in ZigBee tree-based WSNs. Figure 1 shows an example of the network scenario, where there are seven ZigBee nodes, i.e., one coordinator and six routers, and eight slots (labeled from 0 to 7). In this example, each node is assigned to a slot and broadcasts its beacons at the beginning of that slot. As shown in Fig. 1(a), the digit nearby a circle is the assigned slot for that node. After broadcasting a beacon, a node keeps awake for a slot to collect data from its child nodes, and then reports the collected data in its parent's slot. For example, as shown in Fig. 1(b), v collects and reports data at slots 1 and 4, respectively. In ZigBee, nodes' slots are periodically repeated. So, a

* Corresponding author.
E-mail addresses: mspan@mail.tku.edu.tw, 600420391@s00.tku.edu.tw (M.-S. Pan).

http://dx.doi.org/10.1016/j.jnca.2014.07.007
1084-8045/

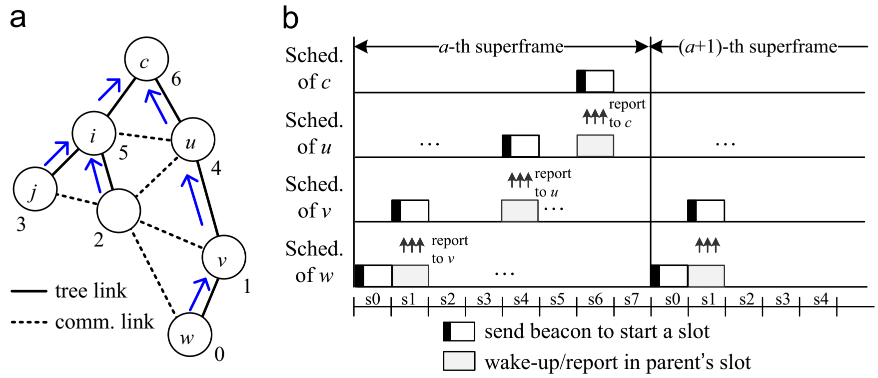

Fig. 1. The scenario of the slot assignment in a ZigBee tree-based WSN.

node periodically wakes up in its slot and its parent's slot, and can sleep in other slots to save energy.

In a ZigBee tree network, when assigning slots, *interferences neighbors* should not be assigned to the same slots to avoid beacon collisions. Conventionally, a node takes its two-hop neighbors as its interference neighbors. Based on this restriction, in Fig. 1, nodes v and j cannot use the same slot, and v's data needs 5 slots to reach the coordinator. In this work, we adopt the interference model in Pan et al. (2013) and Wu and Tseng (2009) (which will be described in Section 3). By the model, in Fig. 1, v and j are allowed to use the same slot, and then v can be assigned to slot 3. As a result, v's report latency will be reduced to 3.

By the above interference model, we further observe that when choosing a slot, a node can locally modify some of its nearby tree links to reduce its latency. More specifically, a node is allowed to demand some of its neighbors to reconnect to other parents. After the reconnecting procedure, the node can use a better slot that can have the benefit of reducing the node's report latency. Based on the above observation, in this work, we propose a centralized and a distributed slot assignment schemes. The designed schemes have three properties. First, the node that reconnects to another parent will not induce interferences to those nodes that have been assigned to slots. Second, the network can still be loop-free after reconnecting some nodes. Third, the report latencies of the reconnected nodes will not increase. Moreover, according to our simulation and implementation results, our schemes can effectively reduce convergecast latency. To the best of the authors' knowledge, this is the first work that discusses how to reconnect tree links to reduce the convergecast latency in ZigBee tree-based WSNs.

The rest of this paper is organized as follows. Section 2 introduces ZigBee superframe structure and related works. Section 3 formally defines our network model. Section 4 presents the proposed centralized and distributed slot assignment schemes. Simulation and prototype implementation results are given in Sections 5 and 6, respectively. Finally, Section 7 concludes this paper.

2. Preliminaries

2.1. ZigBee superframe structure

In a ZigBee tree network, the coordinator and routers follow the definition in the revised version of IEEE 802.15.4 (IEEE 802.15.4b) (IEEE Computer Society, 2006) to form superframes. Before forming the network, the coordinator decides the superframe structure by two parameters, named *beacon order* (*BO*) and *superframe order* (*SO*). The *BO* and *SO* decide the lengths of a superframe and its active potion, respectively. To support low power operations, the setting of *BO* and *SO* should satisfy the relationship $0 \le SO < BO \le 14$. Changing the relationship of $(BO-SO)$ allows the coordinator to adjust the on-duty time of network nodes. According to the definitions of *BO* and *SO*, a superframe can be divided into $2^{(BO-SO)}$ active portions. For example, in Fig. 1, we assume that the $BO-SO=3$. So, there are 8 slots in that network. In ZigBee, a node selects one active portion as its *outgoing superframe*, i.e., its *slot*. For a node, the active portion selected by its parent is called its *incoming superframe*. In an outgoing (resp., incoming) superframe, a node transmits (resp., receives) a beacon to (resp., from) its child nodes (resp., parent node). To avoid beacon collisions, a node should not select the slot that is used by any of its two-hop neighbors. In this work, we will discuss how to loosen this restriction.

2.2. Related works

Energy efficient convergecast based on wake-up scheduling has been investigated in several works. References Choi et al. (2009), Hohlt et al. (2004), Incel et al. (2012), Lu et al. (2007) and Malhotra et al. (2011) introduce link-based slot assignment schemes, where each slot is assigned to a source and destination pair. In Lu et al. (2007), the authors propose an energy-efficient and low-latency MAC, called *DMAC*. Sensors are connected by a tree and stay in sleep state for most of the time. When sensors change to active state, they will be set to the receive mode and then to the transmit mode. *DMAC* achieves low-latency by staggering wake-up schedules of sensors at the time instant when their children switch to the transmit mode. Reference Hohlt et al. (2004) arranges wake-up schedule of sensors by taking traffic loads into account. Each parent periodically broadcasts an advertisement containing a set of empty slots. Children nodes can request empty slots according to their demands. Reference Choi et al. (2009) presents a centralized solution for convergecast. Their algorithm divides nodes into many segments such that the transmission of a node in a segment does not cause interference to other transmissions in the same segment. The aim is to increase the degree of parallel transmissions to decrease latencies. Although the Choi et al. (2009), Hohlt et al. (2004) and Lu et al. (2007) are designed for quick convergecast, they do not discuss how to avoid interferences when assigning slots. Reference Incel et al. (2012) introduces low-latency slot assignments by considering convergecast traffic patterns. When assigning slots, sensors use different channels to eliminate interferences. If there are sufficient channels, sensors use less slots, and thus the report latency can be shortened. Although using multi-channel can effectively reduce interferences, the designs in Incel et al. (2012) cause some overheads to control channel switching. The authors in Malhotra et al. (2011) observe that balanced tree can have the benefits of reducing report latency. They propose a tree construction algorithm, which runs in a level-by-level fashion. For those nodes in a level, their algorithm utilizes the concept of bipartite graph to balance the number of their child nodes. After forming the tree, the proposed link scheduling method is applied to achieve low-latency convergecast. We can see that in