Building Cost-Balanced Routing Trees for Fast Data Collection in IEEE 802.15.4e TSCH Networks

Jian-Yuan Yu and Hung-Yun Hsieh
Graduate Institute of Communication Engineering
National Taiwan University
Taipei, Taiwan 106
Email: hungyun@ntu.edu.tw

***Abstract* - Delay time has been a key factor to consider in many real-time wireless applications, especially in industrial applications that the new TSCH mode proposed in IEEE 802.15.4e targets. However, existing methods focus on minimizing the use of communication resources, but they are unable to exploit the availability of channel multiplicity for achieving shorter average delay time in TSCH networks. In this paper, we propose a new method to construct cost-balanced routing trees that can utilize multiple interfaces at the coordinator. Our simulation results demonstrate that the proposed method is able to achieve low-delay data collection in TSCH networks.**

I. INTRODUCTION

Time Slotted Channel Hopping (TSCH) is a new MAC-layer protocol in IEEE 802.15.4e [1] for supporting highly reliable transmission in industrial applications. It utilizes deterministic synchronized time slots and channel hopping to avoid contention and mitigate the effects of multi-path fading. All nodes are synchronized by the super-frame structure composed of multiple time slots, where the sender can transmit data and receive ACK from the receiver in a time slot. Every node keeps a global schedule to decide whether to communicate or sleep at each time slot. In addition, every node keeps a hopping pattern to allow each communicating pair to alter to another channel for each time slot and avoid collisions or get stuck at deteriorated channels.

A highly cited scheduling method called TASA (traffic aware scheduling algorithm) has been proposed for such a network based on matching and coloring in graph theory [2] [3]. The problem with TASA, however, is that it considers only a single interface at the sink (or coordinator, the root of routing tree). Since the sink with one interface cannot receive signals from multiple children simultaneously, the children can only take turns to talk to the sink, thus resulting in performance bottleneck. Another problem is that TASA forms the routing tree based on the shortest-path algorithm to minimize the number of hops from every node to the sink. Such a design can easily result in unbalanced subtrees, and the largest subtree could turn out to be the bottleneck that requires a long time for message forwarding while nodes in other smaller branches have finished their transmissions.

To address these problems, we first extend TASA for supporting multiple interfaces at the sink to allow parallel transmissions via different channels. To utilize multiple interfaces, we divide the whole network into multiple subtrees, with nodes in each subtree using a subnet of orthogonal channels, so different branches can work simultaneously. We then propose a new metric for building the routing tree. The new metric calculates the communication cost more precisely than conventional metrics using only the hop counts or traffic load. We first extend the Load-Balanced Tree (LBT) [4] into a Cost-Balanced Tree (CBT), and then refine the algorithm to avoid the ping-pong effect and early-stop problem. Finally, we combine routing and scheduling together to better use the provided channels. Although several papers have proposed the concept of the Balance Tree [4] [5] to ease the unbalance, they lack the ability to use all available channels for scheduling. We show through evaluation results that our approach can result in fewer time slots and less delay time compared to TASA.

In the following, we first state the network scenario and then explain in details how we construct the cost-balanced routing tree. Finally, we show the result of the the proposed method in terms of the balance factor and delay time.

II. COST-BALANCED TREE USING MULTIPLE INTERFACES

A. Network Scenario

We consider a scenario with nodes generating packets at the beginning of each super-frame. We consider raw-data convergecast as proposed in [5], such that data cannot be compressed at intermediate nodes. All traffic needs to be forwarded to the sink through the constructed routing tree. To reduce end-to-end delay time, the routing tree should use all available channels and the subtrees should have balanced load.

B. Building the Cost-Balanced Tree

We define the communication cost *C* as the cost to transfer the message from the source to the final destination. Let $c = p * l$, where *p* is the packet size (in bits) and *l* is the distance (in hop counts). The overall communication cost of the network thus is $C = \sum_{i=1}^{N} c_i = \sum_{i=1}^{N} p_i * l_i$ (in bit times hop counts). Obviously, for the same set of nodes with certain local traffic, different topologies result in different communication cost *C*. A thin tree may have larger *C* than a fat tree, and a flat tree is more likely to reach the theoretic lower bound of the schedule length.

With the communication cost *C* as the weight, we design the balancing tree forming algorithm to construct CBT. Initially, we select *K* nodes with the largest number of neighbors (major) and cache size (minor) as the top-level nodes, and make them

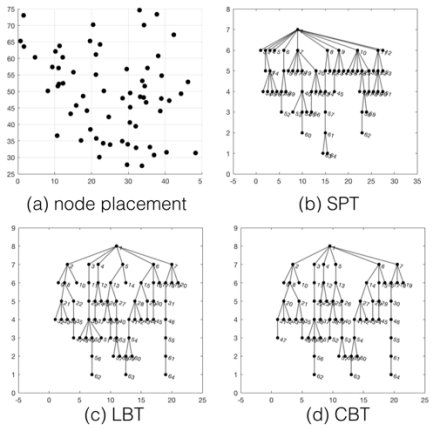

Figure 1. Different routing trees for same node placement.

act as the root of its corresponding branch. The selected nodes are then moved to the marked set, while other nodes remain in the unmarked set. While the unmarked set is not empty, we start from the branch with the minimum communication cost *C* and growing space defined in [4] and then search among the neighbors of the current branch for the unmarked node with the maximum cost *C* and growing space. The selected unmarked node then chooses the marked node with the least depth in the current branch. Once a connection is established, the tree graph, branch sets, marked and unmarked sets, neighborhood and weight of branch are updated.

As we have mentioned, the original adjusting algorithm may easily result in unexpected ping-pong effect and an early stop of the algorithm. To avoid a node repeatedly altering its connection from one branch to another, we keep the history of the former altered node. If the newly found node is the same as the former, then we can identify the ping-pong effect and skip it to go on for the next one. Note that original adjustment algorithm only starts from the biggest branch in one direction and may miss the requirement from smaller branches. We adopt a new approach to start from the current biggest and smallest branches one by one to reach the balance state faster.

C. Performance Evaluation

We randomly distribute nodes in the network, and obtain the results based on the average of 200 runs. Fig.1 shows different routing trees for the same node placement. It can be seen that nodes at the upper level of SPT have a large degree, with very large and small branches being present in the trees, while the branches of CBT and LBT are more balanced.

The balance factor θ defined in [4] evaluates how the weight of each branch differs. As Fig.2 shows, the communication cost of each branch of SPT (shortest-path tree) varies a lot as the network size gets larger. LBT (load-balanced tree) has similar topology and balance factor as CBT in a small network, but the difference becomes significant with larger network size. Notice that since we apply random node placement and random

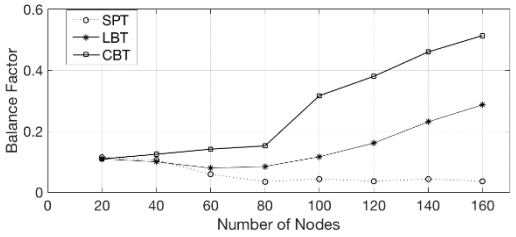

Figure 2. Balance factor over number of nodes with different routing.

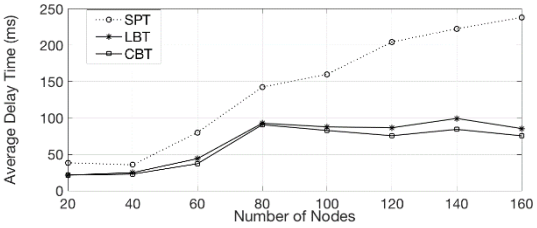

Figure 3. Average delay time over number of nodes with different routing.

generated traffic, the balance factor does not reach the perfect value of 1.0.

For the target application of raw-data convergecast, Fig.3 shows the relationship of average delay time and number of nodes when planar length L = 150 and communication distance d_c = 8. It can be observed that the average delay time of SPT increases much faster than LBT and CBT. CBT performs better than LBT, and the gain becomes more significant when the network size becomes larger. On average, the CBT can save approximate 52% of delay time in CBT and 11% in LBT.

III. CONCLUSIONS

To reduce the latency for nodes operating in the TSCH mode, we address the problems of long-tail effect and unbalance of subtrees in TASA. By leveraging multiple interfaces at the sink, we build Cost-Balanced Tree for routing to allow more efficient parallel transmissions, such that nodes at different branches use different subset of channels. We have observed that the proposed method can result in lower latency of about 50% off the original algorithm under raw-data convergecast.

REFERENCES

[1] "Part. 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANS)", Wireless Networks, 2011.

[2] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks," in Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium, pp. 327–332, IEEE, 2012.

[3] M. R. Palattella, "On optimal scheduling in duty-cycled industrial ioT applications using IEEE802. 15.4 e TSCH," Sensors Journal, vol. 13, no.10, pp. 3655–3666, 2013.

[4] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," in Global Telecommunications Conference, 2003, IEEE, 2003.

[5] O. D. Incel, "Fast data collection in tree-based wireless sensor networks," in IEEE Transactions on Mobile computing 11.1 (2012), IEEE, 2012.