Efficient Scheduling with Retransmission Handling of IEEE802.15.4e TSCH Mode

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

***Abstract*—Reliability has been a key factor to consider in many stringent 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 under ideal channels, but they are unable to consider retransmission handling when it comes to lossy channels in TSCH networks. In this paper, we propose a refined method to permeate the retransmission into the schedule. Our simulation is expected to demonstrate that the proposed method is able to achieve robust data transmission in TSCH networks.**

I. Introduction

The Time Slotted Channel Hopping (TSCH) of IEEE802.15.4e [1] is a new MAC-layer protol that adapts for highly reliable transmission in industrial case. It applies the deterministic time slots to avoid contentions and channel hopping to mitigate the effects of multi-path fading. All nodes are synchronized by the superframe structure that composed of multiple time slots. The number of time slots in each superframe is up to the realistic requirement, and superframes with different size can coexist if necessary [2]. Besides, the sender should finish its transmission and receive ACK from receiver in a time slot. Every node keeps a global schedule to decide its behaviour at each time slot, once there were no transmission for it at current time slot, the node will turn into sleep mode to save energy, and wake up until its turn comes. By channel hopping [3], the transmission will hop to another channel for next transmission. All the node keep a hopping pattern to follow as to avoid collisions or stuck at deteriorated channels. Some advanced channel hopping [4] methods may help when considering the dynamic quality of each channel. Meanwhile, several papers have discussed about the open issue of scheduling, either in centralized and distributed way, centralized method can reach the theoretical optimum performance at the cost of huge computation, while distributed methods [5] [6] are simpler and more flexible. Some scheduling algorithms in TDMA [7] may also offer similar solutions.

Among the listed methods above, we mainly refer to Palattella's Traffic Aware Scheduling Algorithm (TASA) [8] [9] based on matching and coloring of graph theory. With her method, firstly we get a matching set free of the duplex collision by a traffic aware matching method for each time slot, and links in the matching set can be scheduled in the same time slot. Then we continue to divide such a matching set into several matching-coloring sets by coloring method to avoid the protocol interference collision, and links in different coloring set should be transmitted on different channels. Finally we can achieve a deterministic network free of collisions in this way, and it proves to reach the minimum active slots under certain conditions.

However, the TASA only considers lossless channels, which means there is no transmission failure. It's stated in her work redundant resources are reserved for retransmission, but it doesn't specify how much resourced is need nor when to execute the retransmission. Recent Jin has proposed a novel tentative cell allocation method AMUS [10] (Adaptive Multi-hop Scheduling) method to allocate additional resources to vulnerable links on the basis of known ideal transmission, but its calculation lacks consideration about protocol interference, as we will discuss later. His simulation scale is not realistic either. Meanwhile, Honsi [11] handle the retransmission with in distributed way, and upper bounding the end-to-end delay. His strategy is based on stratums to reserve time-bands for each depth in the routing structure constructed by RPL. We also need to figure out which is better between centralized and distributed method.

The rest of the paper is organized as follows. We will introduce some preliminary knowledge about the scenario first, and then follow the review about AMUS. We go on to explore the possibility. Finally, the benefits of our design will be shown by final performance.

II. Preliminary Idea

In centralized method, each node can only follow the schedule it receives at the beginning, and it cannot do retransmission at hand if there was transmission failure. What's worse, due to the multi-hop property, the failure of previous transmission may lead next transmission invalid and result in waste of resources or longer average delay. Notice there are some regular behaviour before retransmission, the coordinator need to collect the statistics from all the nodes to figure out which packet is lost and on which link it is lost, and then it makes adjustment like hopping channel or altering apath to make a retransmission. Such regularity may generate some overhead and cost certain resources, but we ignore the cost of such behaviour in following discussion since it's essential and the traffic is relatively small.

A. Make Up Retransmission

Here we propose two retransmission schemes, the make up and virtual traffic method. The make up scheme will put off all the failure transmissions till the regular schedule ends, the coordinator establishes additional schedule for retransmission after collecting the failure message. Although the probability of successful retransmission increases after adjustment like channel hopping, transmission failure may still happens, we repeat the make up schedule several times until it reaches satisfactory level.

B. Virtual Traffic

On the other hand, the virtual traffic method will recalculate the traffic in each nodes with consideration of their end-to-end packet error rate P_e. Here $Q^{'} = Q/P_e$, $P_e = \prod p_e$, and p_e stands for package error rate for each hop along the path from node to the root. Generally, the larger P_e is, the more likely transmission failure will happen, and more resources the link deserves. Here we adapt BPSK and choose proper setting as standard suggest, $SNR = 2dB, \gamma = 1.6$. The relationship between distance and P_e is shown as in Fig. 1.

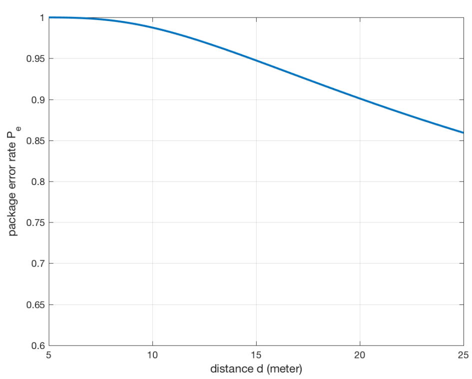

Fig. 1: Packet error rate with distance

C. Comparison

In the simulation of convergecast of all nodes shown as in Fig. 2, since the retransmission situation is much better after adjustment, we assume the retry is ideal and no additional retransmission is needed. We can find the make up method will stuck at plateaus around 20 for long time and then burst when the make up retransmission comes. While for the virtual traffic method, the curve will reach a higher plateaus around 40 and then move on again similarly, at the cost of more time slots. On the consideration of reliabilty, we will prefer the virtual traffic mechanism with higher plateaus.

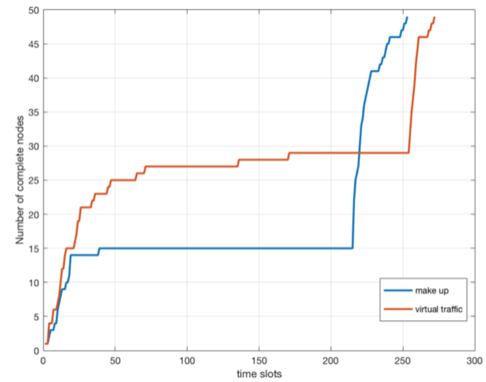

Fig. 2: Makeup vs VirtualTraffic

III. Review of AMUS

AMUS tries to inject more virtual traffic into the network during the scheduling stage and fill as many unscheduled cells as possible in a slotframe. And two key parameters which determines the number of tentative cells. 1). The number of existing traffic on links. 2). The number of cells available for a specific link that would not cause interference or collision to other links. Here $T_c(L_j^i)$ the amount of tentative cells, $R(L_j^i)$ for the aggregated data on link from node N_i to node N_j, T_total is the total number of timeslots in a slotframe.

$$T_c(L_j^i) = \lfloor R(L_j^i) \times min(w_{N_i}, w_{N_j}) \rfloor \tag{1}$$

$$w_{N_i} = \frac{T_{total} - (R_{tx}(l^{N_i}) + R_{rx}(l_{N_i}) + T_c(l^{Ni}))}{R_{tx}(l^{N_i})} \tag{2}$$

$$w_{N_j} = \frac{T_{total} - (R_{tx}(l^{N_j}) + R_{rx}(l_{N_j}) + T_c(l^{Nj}))}{R_{rx}(l^{N_j})} \tag{3}$$

For a simple example shown in Fig. 3, the corresponding schedule table is shown in Fig. 4, and links in red color is reserved for retransmissions.

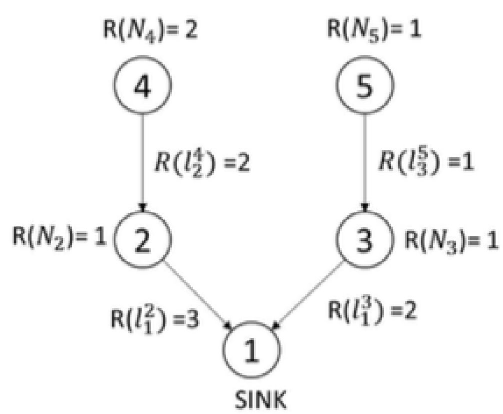

Fig. 3: Simple Example

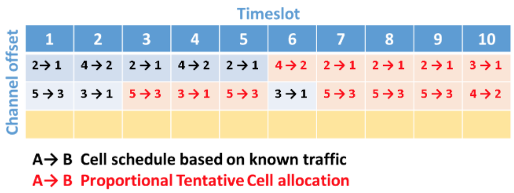

Fig. 4: Example of produced cells

Nevertheless, there are several shortcoming of AMUS. 1) The calculate of W_{N_i} or W_{N_j} is not accurate, for count of the invalid slot part of W_{N_i}, it's $R_{tx}(l^{N_i}) + R_{rx}(l_{N_i}) + T_c(l^{Ni})$. But it counts more if there was intersection set of $R_{tx}(l^{N_i})$ and $R_{rx}(l_{N_i})$, and counts less if protocol interference slot appear. So the count of the invalid slot part here should be $R_{tx}(l^{N_i}) + R_{rx}(l_{N_i}) - intersection(R_{tx}(l^{N_i}), R_{rx}(l_{N_i})) + T_c(l^{Ni}) + Interference(l^{Ni})$, similar refinement for W_{N_j}. 2) Too greedy to use up all remaining slots, we need to find a balance between power consumption and reliability. 3) Channel Hopping or dynamic channel quality rather than hops is not considered.

IV. Scenario and Problem Formation

In each superframe, each node generates certain package at the beginning, and we aim to forward all the traffic to the root with the least average end to end delay time. After node placement and the establish of mesh network, we get a routing tree G_t by different metric and then form the interference tree G_i, and go on to finish the task with scheduling method. The key here is how to form a efficient scheduling

Here we also consider both the aggregated data and raw-data convergecast as proposed in [12]. For former case, packets are aggregated at each hop, and thus leaving space to be summarize with aggregation functions such as MEAN, MAX, MIN etc. Without loss of generality, we adopt MIN here. While for raw-data convergecast, data from each sensor is equally important or the correlation between them is minimal, so packets from children are directly added to parent's cache without any loss.

V. Simulation and Analyze

Network performance is generally evaluated by scheduling length, average delay time, power consumption, packet error rate and so on. Among them, scheduling length tells the overall time units to complete the whole schedule or decides the duty cycle in a superframe, and average delay time is an important metric for applications that requires fresh data. It's known that the larger schedule length the longer average delay time, but they are not the same measurement actually. e.g for a simple star topology, different schedule method has the same schedule length, but different schedule order results in different average delay time. Besides, the convergency graph of a instance can reveal how many nodes have finished their task at different timestamp, and the traffic graph can tell the traffic volume during the whole schedule. In our simulations, we consider the ideal lossless transmission here with packet error rate $p_{er} = 0$, and ensure same number of branches K for different routing trees. We will show detailed results of some essential arguments first, then show the overall performance later. All simulations are based on own written Matlab codes, and each figure is the average of 200 runs.

A. Node Placement and Related Settings

To put it simple, we formulate a tree topology here although the standards have proposed the DODAG (Destination Oriented Directed Acyclic Graph) to generate the topology G_t. There is only one path between any pair of nodes and all links are uplinks. Obviously, node that is not leaf can play the role as relay. We randomly place the nodes and build the mesh topology with two rules. Firstly, distance between any two nodes should be larger than *distinction distance* d_s. Secondly, nodes should be within the *communication distance* d_c of at least one node. The former rule tells that nodes should not stay too near to each other, otherwise it leads to a waste and may bring strong wireless interference. The latter one makes sure that all nodes are connected in the mesh network, and no node is isolated. Notice d_c is directly decided by the power of transmitter, and we assume the same transmit power and d_c for each node besides the root. It's known that any node placement method generally requires to cover as much area as possible with limited nodes, and we define the *coverage distance* $d_v = \sqrt{S/N}$ here, S for whole planar area, N for number of nodes. Generally, the d_s is fixed and d_c is decided by the coverage distance d_v, too small d_c may lead to insufficient coverage and too big d_c brings stronger interference or power consumption, so we set $d_c \approx 0.62 d_v$ by heuristic experiences. We set the communication distance d_c of the sink a bit larger than other nodes to get more candidate top-level nodes (which are children of the root) to ensure enough K branches, since the sink is better equipped and line-powered.

VI. Conclusion and Further Work

TODO

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. Min, "Traffic aware multiple slotframes scheduling algorithm in industrial iot applications using IEEE802.15.4 e TSCH," in *IEEE 16th International Conference on Communication Technology (ICCT)*, IEEE, 2015.

[3] A. M. Watteyne, Thomas and K. Pister, "Reliability through frequency diversity: why channel hopping makes sense," *roceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, 2009.

[4] P. Du and G. Roussos, "Adaptive time slotted channel hopping for wireless sensor networks," in *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*, IEEE, 2012.

[5] N. Accettura, "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops*, IEEE, 2013.

[6] T. W. Tinka, Andrew and K. Pister, "A decentralized scheduling algorithm for time synchronized channel hopping," *Ad Hoc Networks*, vol. 49, no. 10, pp. 201–206, 2010.

[7] S. C. Ergen and P. Varaiya, “TDMA scheduling algorithms for wireless sensor networks,” *Wireless Networks*, vol. 16, no. 4, pp. 985–997, 2010.

[8] 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.

[9] 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.

[10] J. W. Jin Yichao, Parag Kulkarni and M. Sooriyabandara, “A centralized scheduling algorithm for IEEE 802.15. 4e TSCH based industrial low power wireless networks,” in *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, IEEE, 2016.

[11] N. H. Ine‘s Hosni, Fabrice The oleyre, “Localized scheduling for end-to-end delay constrained low power lossy networks with 6TiSCH,” in *Computers and Communication (ISCC), 2016 IEEE Symposium*, IEEE, 2016.

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