An Improved Centralized Traffic Aware Scheduling Algorithm with Level and Density Metric for Time Slotted Channel Hopping in 802.15.4e

Jet Yu
yujianyuanhaha@gmail.com
National Taiwan University, ROC

***Abstract*—This report shows the procedure of a new designed traffic-aware link scheduling of TSCH for 802.15.4e. And later shows the preliminary result and analyse about how different metrics and modes effects the overall delay time performance, thus helping us to focus on selected valuable topic.**

I. Introduction

This report shows the procedure of a new designed traffic-aware link scheduling of TSCH, which is a new foundermental MAC behavior of prosperous 802.15.4e of IoT, the link scheduling problem is also a open issue itself.

The scheduling is composed of matching, coloring, truncating and sorting here, along with the combined level-density metric.

Matching divide the links into different time sets, and coloring further divides the time sets into different time-channel sets, and truncating is needed when the number of channel is not enough to hold for time-channel sets, the sorting finally arrange the time slot sequence of the scheduling.

We consider the time and frequency reuse and pursuit the **spatial reuse** by physical interference model.

We expect our method will have better performance than Palattella's [1].

II. Related Work

Since it's a new issue, few papers have been published to talk about it. But we can refer much from papers about TDMA, which is similar to TSCH, especially about spatial reuse and physical interference model. When it comes to scheduling, it's divided into centralized and decentralized, we first consider the former here.

We primarily rely on Palattella's work here. Palattella believes the **node with larger queue has the priority** to be schedule first, it makes sense but not good enough, I think the **lower nodes or links in sparse networks should have the priority** to transfer first.

Palattella provides a brilliant idea about assigning the links into different time-frequency cells under the interference free rule with the help of **graph theory**. But unfortunately, her algorithm **lacks the consideration for insufficient channels** (TSCH only has 16 channels), and her work only shows rough or basic conclusions in the simulation part. We believe our design can have better performance and more well-around comparison and validation, and her work can be set as a **baseline**.

III. Scenario and Problem Formation

We implement a multi-way tree topology here in 2, such topology set us free of routing, which is a classic and difficult problem in mesh network, that's to say, there exists only one way between two nodes.

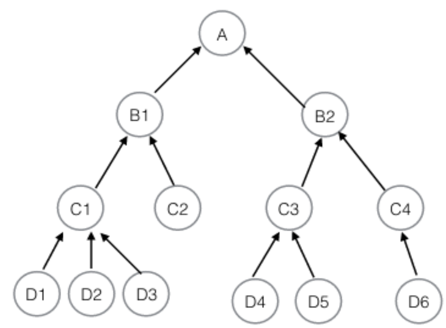

Fig. 1. tree topology

Some papers argues that denser network and more hierarchy architecture design help to reduce the interference, and thus lifting the capacity. However, the increasing aggregators lead more hardware cost and power consumption, the cost of a aggregator is higher than the leaf node or lower one.

The communication behavior maybe simple at first, but we can get more based on this, and this is enough for most application case, we list the case for **smart grid** to collect the electricity usage for each house in a city as an application, the sensor only has to uplink the daily information with tiny size everyday at certain interval. Besides, the aggregator itself is a sensor, the information it gather about its information is just like other sensors. And we have uplink only, each node transfers its cached information to its parent only.

Once it comes to aggregator, we must point out that the aggregator has the capacity to reduce the aggregated information size since the aggregated information is always similar, we can always have some way like entropy reduction to make it. To put it simple, we assume the aggregator can reduce 20% of the size of message from more than one node, we define a aggregate coefficient $\eta = 0.8$ here.

We introduce the concept of **transfer workload** here, which reflects the cost to transfer the message to final destination in that the cost is different for nodes with same size of package but different distance, so here $W = p * l$, p is the packet load and l is the distance.

It's interesting to estimate the quantity and topology for simulation, we want to show the primary result the depth of tree: the depth around 5 and degree around 10 is enough to support thousand of nodes, we can formulate a multi-way tree with depth d under 7, each node can have M around 10 children at most , the number of children of a node is a random(uniform distribution) number generated from M, here M is the limitation, so the expected number of children is $M/2$. Such case is somewhat like the back off of CDMA. We can get a topology random enough to verify our idea.

The coverage of 802.15.4e is aroud 100m, so the area is $0.05^2 * \pi = 0.0075km^2$, the area of Taipei is around $271,00km^2$, consider only half of it need to be covered on average, so we need almost $400,000$ sensors. So $1 + M/2 + (M/2)^2 + ... + (M/2)^d = 400,000$, the interger solution for it is aroud $d = 5, M = 13$.

In general, we want to achieve the minimum average delay time as much as possible, and this benefits much about the power consumption, similarly we can observe the result on users convergence side.

IV. Topology generation

As we list before, the depth and degree of the tree play an import part on the service performance like average delay time and average throughput.

here we shows 3 topology example with uniform distribution, normal distribution and a bias distribution given almost same expectation of number of nodes(by same generation and degree expectation).

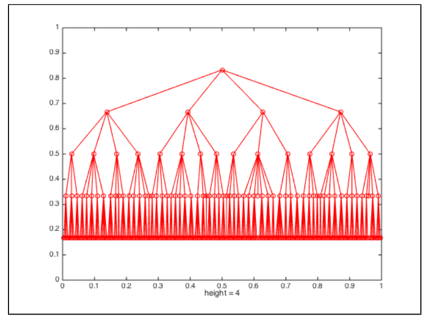

Fig. 2. normal

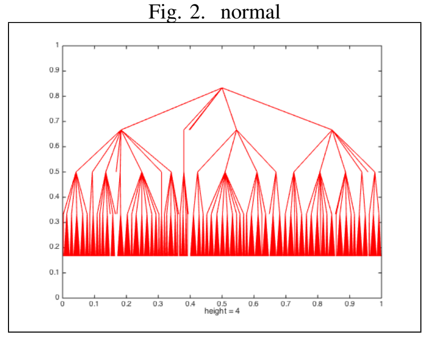

Fig. 3. uniform

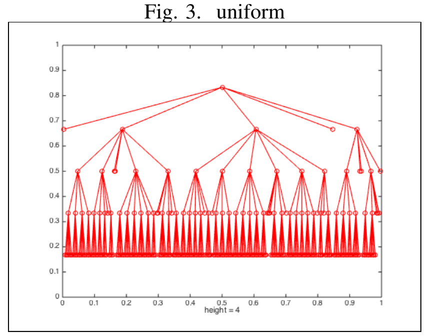

Fig. 4. bais

V. Connectivity Graph and Physical Interference Graph

The connectivity graph show the expected links to carry the communication workload, it's a multi-way tree topology in our case. We can directly duplex conflict relationship from connectivity graph.

The physical graph is not a easy problem, and no perfect solution has be found yet, we can implement the method in the further.

Obviously, the protocol graph is easier while with impaired spatial reusage. However, since our scheduling is minimal, the calculation is done at first, the physical interference model absolutely gains more. We can observe how much the physical interference model gains in the further work.

The generation of physical interference model is a relatively independent problem, we can generate it with some naive assumption at first, it doesn't effect the latter procedure, we plan to give a well-around solution in the further work.

In this case, similarly to Jimmy's method [**?**],we defined there is physical interference if the senders of two links have the

uncle or cousin relationship besides the duplex conflict, this doesn't violate our common sense, the sibling(and its children) of a node's parent is not far from the node most of the time.

VI. Matching

We name the set after matching as *time-set*, and call the case as the *duplex-conflict*, a node cannot send to the other node while it's sending due to half-duplex here, and either can it send to the other node while it's receiving due to collision.

Notice that this procedure only **rule out the possibility of duplex-conflicit**, the separated links cannot be scheduled in the same time duration(compose of several time slots), but it still worth efforts to separate the links in the same time duration into different channel, we can see further separation in coloring section and see how to select most beneficial links once the channel is not enough to support for large time-set.

The matching procedure itself is an classic problem in graph theory, we adapt the algorithm of BGL called **ermonds maximum cardinary matching**, which give the set with maximum number of links in each run, and we can get sets in decreasing order finally.

VII. Coloring

Similarly, we name the set after matching and coloring *time-frequency-set*, links in the same time-frequency-set can be scheduled in the same time slot and same channel, it's indeed spatial reused actually.

Again we mention that **coloring only** can separate the links into different time-channel cells since the physical-interference graph is derived from connectivity graph and thus more restrictive. However, the matching procedure makes the work more efficient, it's known that coloring a small sub-graph after the matching procedure is much easier than to color a gaint graph.

Besides, although the matching and coloring algorithms follow default order(set with larger links comes ahead), we don't care the default order or the subtle problem(like which sets comes ahead with same number of links) because we apply the sorting procedure in later time, the sorting may break the order then.

Notice that since we have not update the physical interference model here, the spatial reuse is still limited, we can expect better spatial reuse and average throughput after the refinement on this part.

VIII. Truncating

It's possible we get a time-frequency set of links more than the numbers of channel here, which is 16 in 802.15.4e. As a result, we need to **select the most beneficial links and truncate the set into several sets** of appropriate size. We will follow the metric here.

TABLE I
Variable used

W	workload
p	packetload
l	distance
d	depth of tree
sub()	subtree of a node in a tree
M	maximum number of children of a node
η	aggregate coefficient
θ	complete coefficient
M_{lev}	level metric
M_{den}	density metric
M_{com}	combined metric

A. the level metric

In order to get the minimum average delay time as much as possible, we follow the principle, similiar to [**?**],the lower nodes transfer first.

However, it doesn't absolutely mean the lower always comes first, if the higher node is a leaf or has collected enough.

So each node maintain a **workload list**, which contain the **message has been transfered, cached and not collected yet**, we name the message has been transfered part as completed coefficient θ, and it reflects how jobs has been finished. If the completed coefficient has became larger than a certain threshold(let it be 0.5 in our example), we admit the higher node to transfer before the lower in such case. We can see a example as shown in 5, here $M_{lev} = d + 2 * \theta$.

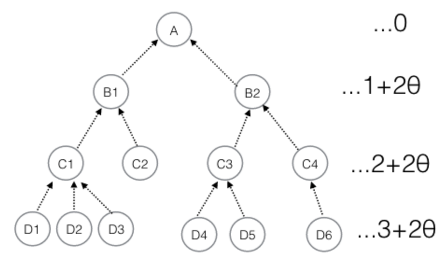

Fig. 5. level metric

B. the density metric

Obviously, the network density also matters the priority, the level metric listed before only reflects the attribute of level of a single node, and the density metric here can reflect the neighborhood.

We can simply validate our idea for density metric by the simple example as 6. Here weight of node $W = 1 + sub(W)$ in a recursive way, and metric $M_{den} = (W_{src} + W_{des})/2$, and $B = max\{M_{lev}\}$.

C. the combined metric

We can combine the two-order sort into one to make the sorting more efficient, we generation a combine metric level-density:
since level is more important we multiple it with a number big B enough to cover the difference of network density, we have

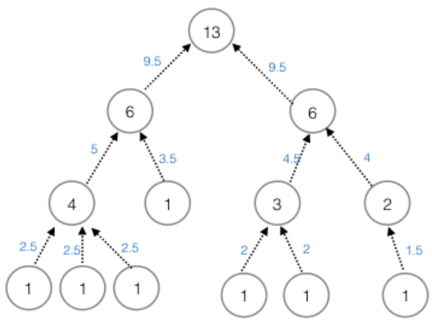

Fig. 6. density metric

to compare the secondary density metric when the primary level metric is the same. So $M = M_{lev} * B + M_{den}$.

IX. Sorting

Once we finish the matching and coloring, think of the classic example of queueing, we always let the person with less service time ahead to get the minimum average serve time on customers side though the server has the same serving time on its side. So how to sort the time slot matters.

We can two ways here, the bulk sort takes the time-frequency set as listed before as one, it calculate the overall level-density metric for each bulk, and sort these sets in decreasing order.

Also, notice the **time-frequency sets can further break down into time slots,** and these slot-level sets can be treated sets similiarly, we can also sort them up in deceasing order and compare with the former to see which performance better.

A. bulk-level sort

B. slot-level sort

Notice we need to run the four steps in several runs, at least the number of run is equal to that of depth.

X. Download case

The download case is much similar to the uplink case, the only differences are the changed physical interference graph(not binary) and the inverted level metric(the higher level download first).

XI. Result and Analyze

A. Metric

We can see the id-based, level and refined level perform best, we can further see the comparison of the three in detail in the second. At least, the queue or subtree queue metric is not favoured.

Result: time sequence set of links
initialization: calculate $\mathbf{q}, \mathbf{m}$ based on G ;
// G: connectivity graph, G_i: physical interference graph ;
// $\mathbf{q}$: completed workload vector for each node ;
// $\mathbf{m}$ combined metric vector ;
while $q \leq 1$ **do**
    $vec\ TimeSet[i] \leftarrow Matching(G)$;
    $vec\ TimeFreqSet[i][j] \leftarrow Coloring(G_i)$ for all $vec\ TimeSet[i]$;
    if $j >$ *num of channels* **then**
        $vec\ TimeFreqSet[i][j] \leftarrow Truncating(TimeFreqSet[i][j], \mathbf{q}, \mathbf{m})$ for all links in $vec\ TimeSet[i][j]$;
    else
    end
    Schedule the first time slot by default;
    $Sequence[] \leftarrow Sequnce[] + Next(TimeFreqSet[][], \mathbf{m}, \mathbf{q})$;
    $Update(\mathbf{q}, \mathbf{m})$;
end

Algorithm 1: Centralized algorithms for level-density based TASA

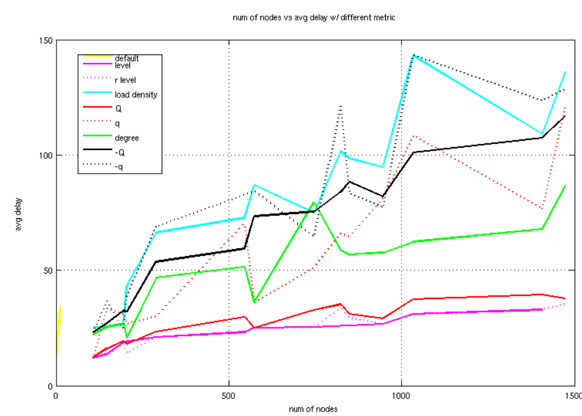

Fig. 7. Metric

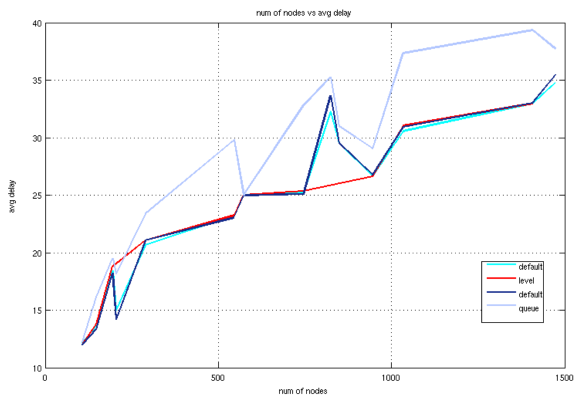

Fig. 8. Metric Detail

B. Metric Average vs Overall

Whether the sorting metric is by overall or average is a problem, both seems reasonbale. We decide to verify in simulation and its turns out both are nearly the same(slight different does exist), so we ignore related picture here. The superposed lines tell us that our metric is robust or significant enough to be distinguished from each other.

C. Batch vs Time Sloted

Since the time slotted have huge computing load and the disadvantage of empty transmit, and actually it doesnt perform any better than bundle, I dont think time slotted is acceptable.

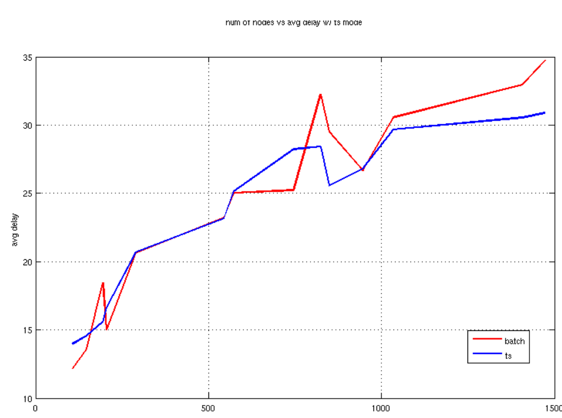

Fig. 9. Batch VS Time Sloted

D. Hold vs Breaking Matching Set

Similarly, breaking matching set may rule out the case about the empty transmit, but the increasing matching computing is a problem, and it actually doesnt perform any better.

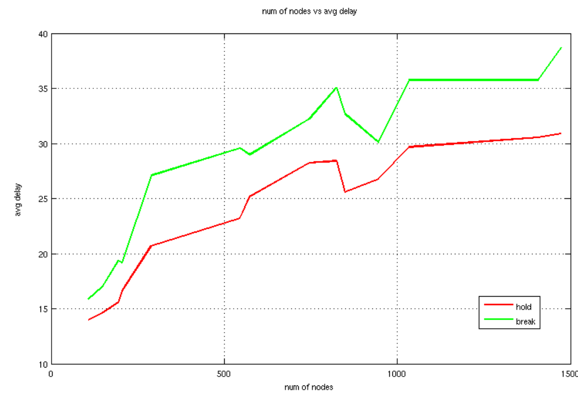

Fig. 10. Hold VS Breaking Matching Set

E. Sort and Split vs Default

As listed before, here we want to show the difference if my sort and split method works in tha case default 16 channel is not enough, but unfortuately, here comes superposed lines again. After analyze, I find the number of channel used, i.e. the number of coloring set in a mathing time slot is ratherly small, even smaller than 4, under the rough interference model. It seems that I should revisit the TSCH standards to make sure if a transfer over multi channels is allowed, which will bring in new problems here.

XII. Conclusion and Further Work

The result show some part of my initial idea is right, especially for the level metric perform better than prevailing queue metric, and the time slotted mode also benefits, these at least makes contribution so far. We will go on to investigate why the level metric and time slotted mode works with math tools, and of course, we need to hand on the mex file about matching and colouring.

References

[1] 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 on*. IEEE, 2012, pp. 327–332.

[2] J. Grönkvist, "Traffic controlled spatial reuse tdma in multi-hop radio networks," in *Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on*, vol. 3. IEEE, 1998, pp. 1203–1207.

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