

Minimal Idle-Listen Centralized Scheduling in TSCH Wireless Sensor Networks

Mary Nsabagwa*, Joshua Muhumuza*, Robert Kasumba*, Julianne Sansa Otim*, and Roseline Akol†

*Department of Networks, Makerere University, Kampala, Uganda

†Department of Electrical Engineering, Makerere University, Kampala, Uganda

Email: mnsabagwa@cit.ac.ug, josh.jesusreigns@gmail.com, robein@ymail.com, sansa@cit.ac.ug, makol@cedat.mak.ac.ug

Abstract—Centralized scheduling in IEEE 802.15.4e Time Slotted Channel Hopping (TSCH), uses a centralized entity to perform scheduling of nodes in cells to generate collision-free communication amongst neighboring nodes. Although many centralized scheduling algorithms have been proposed, they still suffer from idle-listening, a state in which nodes stay awake without transmitting or receiving data, which causes wastage of the limited sensor node energy. This paper proposes a minimal-idle-listen centralized 6TiSCH scheduling algorithm (MILS) with the aim of minimizing idle listening amongst sender nodes. MILS is formulated as a Constraint Satisfaction Problem (CSP) problem, which schedules relay links in parallel with leaf links while maximizing the channels available in order to minimize waiting at bottleneck regions. Minimizing the number of packets in nodes close to the sink reduced delays by approximately 19% and idle-listening by 50% in MILS compared to OTF.

Keywords—centralized 6TiSCH scheduling; Constraint Satisfaction Problem; Idle-listening; TSCH; Wireless Sensor Networks;

I. INTRODUCTION

Applications such as tracking many wild animals using wireless sensor networks benefit from IEEE 802.15.4e [1] in conserving the limited sensor node energy. IEEE 802.15.4e is an amendment of the IEEE 802.15.4 MAC protocol standard, supporting multi-channel communication using Time Slotted Channel Hopping (TSCH), which is a medium access mode using time synchronization to allocate cells within a slotframe. A slotframe consists of timeslots, grouped together, repeating over time and forming a TSCH schedule. A TSCH schedule consists of cells, characterized by slotOffset and channelOffset, hence, collision-free communication amongst neighboring nodes and indicating when a node should transmit, receive or sleep in a given cell. During network setup, an Absolute Slot Number (ASN), which is used to calculate the channelOffset is initialized to 0 and is incremented by 1 for each timeslot. For each slotframe, a different channel is obtained for two neighbors with the same channelOffset.

Centralized scheduling algorithms such as those proposed in [2][3] generate schedules using a Path Computation Element (PCE), which may be located either in or outside the network. The schedules are distributed to nodes using level layer protocols such as CoAP[4]. One of the problems in the proposed algorithm is idle-listening, which causes energy wastage [5].

We are proposing a minimal-idle listen centralized 6TiSCH scheduling algorithm (MILS) with the main aim of minimizing

idle listening. We formulate a Constraint Satisfaction Problem (CSP) problem which schedules relay links in parallel with the leaf links to minimize waiting at bottleneck regions. Minimizing the number of packets sent to nodes near the sink, at a given time reduces delays and packet dropping, which would be brought about as a result of buffer overflows. Our contributions are as follows

- Designing a minimal idle-listen centralized 6TiSCH scheduling algorithm, formulated as a Constraint Satisfaction Problem (CSP)
- We establish an optimal slotframe size while maximizing the available channels, a function of network size and topology

II. SCHEDULING IN 6TiSCH

6TiSCH proposes scheduling schemes including autonomous, distributed and centralized schemes. Autonomous schemes such as orchestra[6] and ALICE [7] operate without exchange of control packets while distributed scheduling requires neighboring nodes to negotiate the allocation and de-allocation of transmit, receive or sleep cells using Enhanced Beacons [8], [9], [10], [11]and [12]. Distributed scheduling generates non-optimal routes since nodes have limited visibility of the network, hence increased delays and idle listening.

In centralized scheduling, the Path Computation Element (PCE) uses network routes from routing protocols such as RPL[13] to generate optimized schedules. The schedules are then installed on sender nodes using application layer protocols such as CoAP. Centralized scheduling algorithm's performance is negatively affected by communication overhead[14] and idle listening, which waste node energy.

Many centralized scheduling algorithms have been proposed to solve the above problems. Adaptive Multi-hop Scheduling (AMUS)[2] proposed allocation of more resources to vulnerable links to reduce idle listening by using a notification mechanism to relay packets which are scheduled in later time slots compared to their source nodes. This mode of operation causes increased idle listening at the bottleneck region since all packets which originate hops away from the sink get delivered at the same time to relay nodes near the sink. Traffic Aware Scheduling Algorithm (TASA)[11] tries to build interference conflict free links through graph matching and coloring, based on network topology and traffic load. The algorithm tries to minimize latency by performing parallel cell allocation, hence high

throughput. A new centralized link scheduling (CLS)[5] and Plexi[15] are other proposals, which focus on considering the bandwidth requirements of the nodes when minimizing idle-listening. Plexi proposes an API, which performs slotframe and cell allocation based on network performance and based on resources exposed by nodes. It is however unclear how much energy is used to transmit resource information to the PCE. In order to reduce traffic between nodes and the scheduling entity, CLS proposed a mechanism in which nodes send cell allocation request messages to the sink only when the need arises. CLS tries to limit the number of packets exchanged by allocating and de-allocating time slots without rescheduling the entire schedule, a process which reduces communication overhead. Despite the strengths of the above centralized scheduling algorithms, little efforts have been made to reduce idle-listening. MILS proposes a scheduling approach in which leaf nodes are scheduled in parallel with the relay links, while maximizing the available channels in order to reduce idle-listening at bottleneck regions of the network.

III. MINIMAL IDLE-LISTEN SCHEDULING (MILS)

A. Network Model

We consider a multi hop IEEE 802.15.4 Wireless Sensor Network (WSN) with one sink node, which doubles as the PCE or the scheduling entity. The sink receives routing information from RPL and uses them to generate schedules, which are distributed to the nodes using CoAP. Given a network in form of a weighted Directed Acyclic Graph (DAG) $G = (V, E)$, where V is a set of vertices representing nodes and E is a set of edges joining vertices and representing links between nodes sending data to their neighbors. Each node must have sent a packet by the end of a timeslot through its preferred parent to the sink. A node at time t should either receive, transmit or sleep. Based on the routing topology, the amount of upward traffic for a given node is given by equation 1, where n is the network size and k is the number of child nodes.

$$f(n) = \begin{cases} n - 1, & \text{where node is sink} \\ n - k, & \text{where node is relay} \\ 1, & \text{node is leaf} \end{cases} \quad (1)$$

At the start of a slotframe, each node receives its schedule information from the sink/ PCE. Two neighbors may communicate in the same time slot and channel if one is a sender and the other a receiver. Otherwise, a different channel or timeslot is used.

B. Scheduling as a Constraint Satisfaction Problem

We formulated MILS as a Constraint Satisfaction Problem (CSP), which schedules edges or links between nodes in non-conflicting cells using the following inputs:

- Variables, which are nodes in a network identified by their addresses
- Domains, which are a set of {variable: [possible_value]} entries including cells (time slots and channels) available, in which a node can be scheduled
- Neighbors, which are a set of {variables: [variables]} containing nodes within range of each other that participate in a given constraint

- Constraints: A function $f(A, a, B, b)$ that returns true if neighbors A, B satisfy the constraint when they have values $A=a, B=b$. The four constraints on any pair of nodes include
 - A link can only be scheduled in a time slot if none of the nodes involved in that link appear in the same time slot
 - Once a node wakes up in a timeslot, it should perform all its duties before going to sleep
 - The minimum number of time slots a node has in a slotframe is equal to the number of children it has plus one
 - Child links are scheduled in the same time slot in parallel with their parents as long as no other constraint is compromised.

Below is the pseudo code for the algorithm

Inputs:

Set of links (sender->receiver),
max_timeslot_number, Number_of_channels

Output: Schedules

START

```
LinksList=getCompatibleLinks(LinksSet)
slotframe [TIMESLOTS] [CHANNELS]=NULL
```

```
while(LinksList_length>0) {
    cur_link= nextLink(LinksList)
    cur_link.assigned = FALSE
    timeSlot =0, channelOffset =0;
    while(timeSlot < TIMESLOTS) {
        if(channelOffset >=CHANNELS) {
            timeSlot++
            channelOffset=0;
        }
        if(!empty(slotframe [t] [chan])) {
            channelOffset++;
            CONTINUE
        }
        if(!violation(constraint,t)) {
            current_cell= cur_link
            assigned_links[i]= cur_link
            break
        }else { t++
                chan= 0
            } }
        if(cur_Link.scheduleTimes==0) {
            LinksList.remove(cur_Link)
        } } END;
```

C. Minimum slotframe size

Slotframe S is proportional to $N-H$, where N is the number of nodes and H is the number of hops

$$S \propto N - H$$

$$S = k(N - H); k \leq 1$$

$$k = \frac{N}{\sum_{i=0}^n Di}; \text{ Where } Di \text{ is the node degree for node } i$$

Therefore,

$$\text{Therefore, Min Slotframe_size} = \frac{N(N-H)}{\sum_{i=0}^n Di} \quad (2)$$

We proved the accuracy of equation (2) by testing it with varying network topologies and node sizes.

IV. EVALUATION

We used 6TiSCH simulator[16], a free and open-source event-driven simulator, which implements 6TiSCH elements including IEEE 802.15.4e, RPL and 6top. We customized the simulator such that a stable topology based on node ranks, which are calculated based on distance from sink. Each node gets only one parent. We compared performance of MILS with OTF distributed scheduling algorithm in order to benchmark performance with distributed scheduling, which has been considered to perform better than centralized scheduling.

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Burst Timesptamp	20
Number of nodes	50 and 100
Number of channels	16
Slotframe length	3000
Cycles per run	100
Packets per burst	5
Slot duration	0.1
Threshold	8

A. Algorithm Time and Space Complexity

Time complexity is affected by the following: -

- Generating compatible links uses $O(n^2)$ time.
- Choosing the next link, operates in $O(n \log n)$ time.
- The function isConstraints, which checks if a particular assignment violates any constraints operates in $O(n)$ time.

The overall time complexity of the algorithm can be calculated from

$$a) \quad \text{Time Complexity} = n^2 + n(n \log n + \text{TIMESLOTS} + n)$$

$$= n^2 + n^2 \log n + n * \text{TIMESLOTS} + n^2$$

$$= O(n^2 \log n)$$

b) The algorithm uses the LinksList data structure of size n and the slot frame of size $\text{TIMESLOT} * \text{CHANNELS}$ which is always a constant value. So the overall Space complexity is $O(n)$.

B. Results

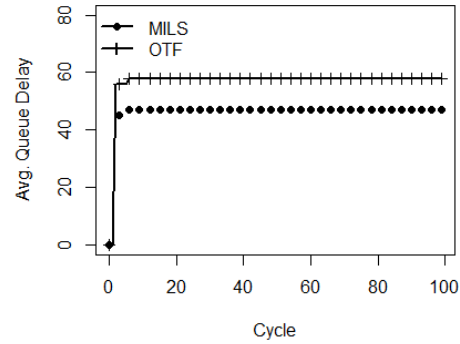


Fig. 1. Average queue delay for MILS and OTF

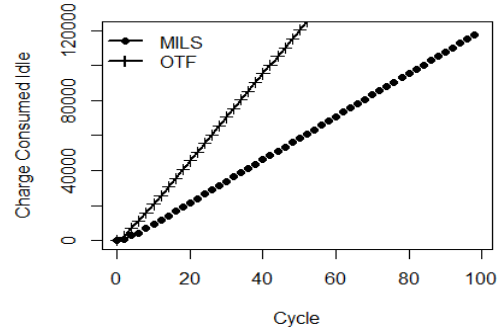


Fig. 2. Charge consumed in idle state

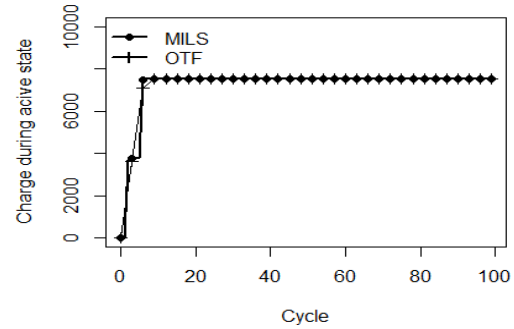


Fig. 3. Charge consumed during node active state

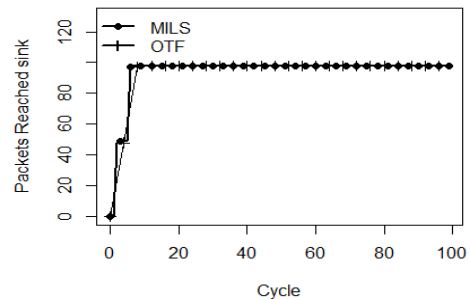


Fig. 4 Packets that reached the DAG root

C. Discussion of Results

Fig. 1 gives the average queue delay, which is the average time taken from when packet is generated to the time it arrives at the DAG root. The maximum queue delay for MILS is 47 while OTF reaches 58. MILS incurs a lower queue delay than

OTF because of its global view of the network, providing knowledge to nodes and enabling them to wake up at the right time of transmission and because relay packets are transmitted before those from children nodes arrive.

In regard to idle-listening Fig. 2, MILS takes half the amount of idle listening as that of OTF. This is true for small networks because as the network is doubled from 50 nodes to 100 nodes, idle-listening in MILS doubles. The reason behind the increased idle-listening in OTF only negotiates and only plans for schedules within its neighborhood. As packets move towards the sink, they are forced to wait for their parents to be available, causing idle-listening. While idle-listening in OTF is big in smaller networks, an increase in network size from 50 nodes to 100 caused negligible effect on idle listening. This implies that in big networks, OTF performs better than MILS.

The simulator propagates packets in the first 10 cycles. This is the time when nodes actively wake up, send and receive packets. MILS first sends packets from nodes close to the DAG root, hence, propagation stops earlier than OTF which first sends packets from nodes that are far from the DAG root. There is a maximum of 5.1% increase in charge dissipated during active state in MILS compared to that of OTF during the first 8 cycles. In the same period of time, MILS delivers more packets to the sink, compared to OTF. The number of packets generated by OTF are 2.5 times more than those generated in MILS. This is mainly attributed to the control information exchanged by nodes during the node negotiation process in OTF, a process which is performed at the DAG root in MILS. The total charge dissipated in MILS ends up being smaller than that in OTF because idle listening is much higher than active state charge, making the sum of charge in MILS smaller.

V. CONCLUSION AND FUTURE WORK

The rate of energy consumption has an impact on the life of wireless sensor nodes, hence, the need for energy conservation mechanisms. Centralized TSCH scheduling at the MAC layer schedules nodes in timeslots and channel offsets such that collisions are minimized. Existing centralized scheduling algorithms however face challenges with idle listening, hence consuming the limited energy. We have reviewed the proposed centralized scheduling algorithms and based on the predominant idle listening problem, have proposed a Minimal Idle-Listen (MILS) centralized scheduling algorithm, using constraint Satisfaction Problem. Using the 6tisch simulator, the algorithm reduces idle-listening to 50% compared to OTF and approximately 19% reduction in delays. In future, we shall assess the impact of schedule installation on the total charge.

VI. ACKNOWLEDGEMENT

We would like to acknowledge the financial support of NORAD and Peter Tahinduka for the initial input into the algorithm.

REFERENCES

- [1] B. Soediono, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement," *J. Chem. Inf. Model.*, 2015.
- [2] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks," *IEEE Wirel. Commun. Netw. Conf. WCNC*, vol. 2016-Sept, no. Wcnc, 2016.
- [3] M. Ojo, S. Giordano, G. Portaluri, D. Adami, and M. Pagano, "An Energy Efficient Centralized Scheduling Scheme in TSCH Networks," 2017.
- [4] Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol (CoAP)," *CoRE Work. Gr.*, 2013.
- [5] K. H. Choi and S. H. Chung, "A new centralized link scheduling for 6TiSCH wireless industrial networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.
- [6] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems - SenSys '15*, 2015.
- [7] C. K. K. (Seoul N. U. Seo Hyang Kim (Seoul National University), Na Eon Kim (Seoul National University), Nguyen Duc Lam (Seoul National University), "Autonomous Link-based TSCH Cell Scheduling," 2017.
- [8] M. R. Palattella et al., "On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks," *IEEE Sens. J.*, 2016.
- [9] I. Hosni, F. Tholeyre, and N. Hamdi, "Localized scheduling for end-to-end delay constrained Low Power Lossy networks with 6TiSCH," in *Proceedings - IEEE Symposium on Computers and Communications*, 2016.
- [10] M. Domingo-Prieto, T. Chang, X. Vilajosana, and T. Watteyne, "Distributed PID-Based Scheduling for 6TiSCH Networks," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1006–1009, 2016.
- [11] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, "Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 455–470, 2015.
- [12] T. P. Duy, T. Dinh, and Y. Kim, "Distributed cell selection for scheduling function in 6TiSCH networks," *Comput. Stand. Interfaces*, 2017.
- [13] R. K. A. Tim Winter, Pascal Thubert, Anders Brandt, Jonathan W. Hui, Richard Kelsey, Philip Levis, Kris Pister, Rene Struik, JP. Vasseur, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," 2012.
- [14] and T. W. Erwan Livolant, Pascale Minet, "The Cost of Installing a 6TiSCH Schedule," *Springer Int. Publ. Switz.*, pp. 17–31, 2016.
- [15] G. Exarchakos, I. Oztelcan, D. Sarakiotis, and A. Liotta, "plexi: Adaptive re-scheduling web-service of time synchronized low-power wireless networks," *J. Netw. Comput. Appl.*, 2017.
- [16] X. V. Thomas Watteyne, Kazushi Muraoka, Nicola Accettura, "The 6TiSCH Simulator." [Online]. Available: <https://bitbucket.org/6tisch/simulator/src/57f10f3138773d7776f62acc85ca0b59c5cd4d7b?at=master>.