2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

# Centralized Graph based TSCH Scheduling for IoT Network Applications

Nikumani Choudhury*, Moustafa M. Nasralla†, Prakhar Gupta*, Ikram Ur Rehman§
* Dept. of Computer Science and Information Systems, Birla Institute of Technology & Science, Pilani, Hyderabad, India
† Dept. of Communications and Networks Engineering, Prince Sultan University, Riyadh, Saudi Arabia
‡ School of Computing and Engineering, University of West London, London W5 5RF, UK
Email: nikumani@hyderabad.bits-pilani.ac.in, mnasralla@psu.edu.sa,
f20171441@hyderabad.bits-pilani.ac.in, ikram.rehman@uwl.ac.uk

***Abstract*—The current specification of the IEEE 802.15.4 standard supports several application specific Quality of Service (QoS) requirements for Internet of Things (IoT) network applications. Specifically, the Time Slotted Channel Hopping (TSCH) MAC mode provides effective latency and throughput performance through the use of dedicated timeslots between two communicating devices. Despite the impact TSCH MAC can facilitate in low-power lossy networks (LLNs), the standard does not explore either the building or maintaining of a schedule. The challenge is to build an energy-efficient TSCH schedule that repeats periodically over several channels. To address this problem, we propose a centralized cluster-level TSCH scheduling mechanism from the energy-efficiency perspective. The proposed mechanism derives a collision graph for each of the clusters in the network topology to schedule non-overlapping timeslots. The Bron–Kerbosch algorithm is used as a sub-procedure for finding the complete sub-graphs of a graph. In addition, we analytically compute the transmission and energy overhead with the help of a Markov Model for TSCH.**

***Index Terms*—IEEE 802.15.4, Internet of Things, TSCH MAC, scheduling, Bron–Kerbosch Algorithm.**

## I. Introduction

In the last decade, there has been an enormous proliferation of embedded devices connected to the internet. This has resulted in the rise of several IoT applications suited for various domains such as home and industry. These diversified IoT networks also require different QoS performance metrics suited to a variety of real-time or time-critical applications. These demands are supposed to increase two-to-three fold every forthcoming year. Therefore, there is a need for a networking standard that adheres to the needs of the constrained embedded devices connected to an IoT network but also fulfills the QoS demands of these IoT applications. The IEEE 802.15.4-2011 [1] standard is one such standard that was designed for IoT applications with constrained resources and non-stringent QoS requirements. The revised IEEE 802.15.4-2020 [2] standard is designed for real-time applications with latency and throughput constraints that need to provide better reliability and robustness. Among the three available MAC modes of the standard, the TSCH MAC mode provides high reliability and time-critical assurance for several medium to large-sized IoT network applications. It is specifically suited for applications that are prone to interference from other wireless networks.

### A. Motivation

TSCH MAC enables devices to communicate using a special timeslot mechanism (described in the next subsection) for effective and non-overlapping transmissions. A scheduling mechanism is necessary to allot the available timeslots to the devices in the network across the different channels in order to achieve low-latency and high throughput QoS requirements. However, the IEEE 802.15.4 standard does not define and explore any scheduling mechanism. Additionally, the TSCH schedule should make optimal use of the available resources (maximize the use of channels with minimal timeslots), incurring minimal overhead. Prior works [3]–[13] in this direction suffer from several limitations such as high overhead, non-adaptive to changes in the network, or focuses on a single QoS parameter like throughput maximization. In this paper, we present a lightweight TSCH scheduling mechanism that aims to allot timeslots optimally.

### B. Time-slotted Channel Hopping

One of the newly adopted MAC modes in the IEEE 802.15.4 standard is the TSCH MAC that aims to provide high reliability and time-critical assurance for several medium to large-sized IoT network applications. The TSCH mode is significantly different from the legacy IEEE 802.15.4 MAC [14], [15]. In this mode, the devices synchronize to slotframe, which is periodic in nature. A slotframe is a collection of timeslots, as shown in Fig. 1. A timeslot within a slotframe is a pairwise communication slot between any two associated devices. Such a timeslot can be dedicated (one pair of devices) or shared (CSMA/CA) between several pairs of devices. Fig. 1 depicts a slotframe that consists of 3 timeslots and nodes A, B, and C are in transmission with each other. TSCH supports multi-channel communication that is built on a pre-defined channel hopping sequence. The frequency of a link [2] is computed as shown below:

$$f = F[(\text{ASN}) + \text{Channel Offset}]\%N_{\text{channels}}\ , \qquad (1)$$

where $F$ is the channel Hopping Sequence list and $N_{\text{channels}}$ is the number of channels used in the current network operation. TSCH defines a timeslot counter called Absolute Slot Number (ASN). When a new network is created, the ASN is initialized to 0; from then on, it increments by 1 at each timeslot.


DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00219

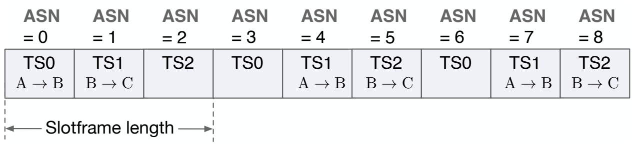



Fig. 1. Depiction of a 3-timeslot slotframe.

One of the important features of the TSCH mode is its CSMA/CA algorithm which is based on the IEEE 802.15.4 CSMA/CA mechanism. Additionally, the TSCH re-transmission mechanism is very energy-efficient. In the paper, we analyze the energy consumption and transmission time of the TSCH CSMA/CA and re-transmission algorithms with the help of a Markov model.

### C. Contribution & Organization

This paper proposes a centralized cluster-level TSCH scheduling mechanism with the help of Bron–Kerbosch algorithm. The proposed mechanism derives a collision graph for each cluster in the network topology to schedule non-overlapping timeslots. Additionally, the transmission and energy overhead is analytically computed with the help of a Markov Model for TSCH. The primary contributions of this paper are itemized below:

- A centralized cluster-level TSCH scheduling mechanism is proposed that schedules non-overlapping timeslots, that repeats periodically over several channels.
- We analytically compute the transmission and energy overhead with the help of a Markov Model for TSCH.

The rest of this paper is organized as follows. The related works are presented in Section II. The network model is described in Section III. The proposed TSCH scheduling mechanism is presented in Section IV, followed by the analytical results in Section V.

## II. Related Work

The recent inclusion of the TSCH MAC into the standard has resulted in an increasing interest in TSCH timeslot scheduling. An optimally devised schedule can serve several application's QoS requirements. The works in [3]–[6] propose scheduling mechanisms for TSCH networks. The authors in [3], proposed an Orchestra scheduling scheme that aims to achieve high throughput. It uses Routing Protocol for Low-power and Lossy Networks (RPL) [16] routing to schedule slots. The RPL implementation adds extra overhead to the network. Additionally, it is not adaptive to a change in the traffic rate on the channel. On the other hand, the adaptive static scheduling in [4] focuses on low and deterministic delay for the static networks. Wave [5] is another scheduling scheme that targets minimal latency based upon traffic flows. Several iterations, called wave, are required to arrive at the final schedule. Each next wave is an optimized subset of the first wave. Stripe [6] is a distributed scheduling mechanism that reconfigures random pre-allocated slots and later schedules additional slots based on traffic. It comprises of a relocation

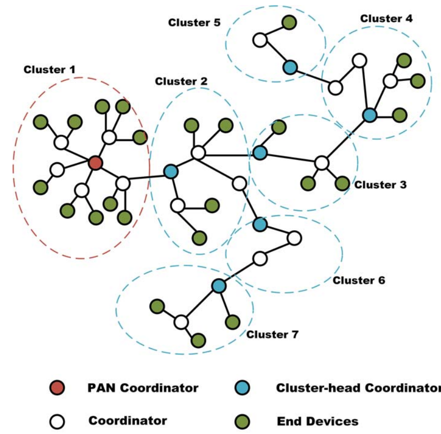



Fig. 2. TSCH based cluster-tree network topology.

phase and a reinforcement phase that schedules additional cells to support the traffic generated and relayed by each node towards the sink. The learning phase results in higher energy consumption by the constrained IoT devices. The authors in [7] proposes a priority-aware, scalable scheme called PriMulA that introduces a PriMulA message containing priority of the frames based on the deadline of message consumption. Each frame requires to carry an overhead of the extra priority bit and the associated computations by the set of devices. Authors in [17], [18] propose several adaptive channel selection mechanism for TSCH networks.

The authors in [8] formulated the scheduling problem as a throughput maximization problem and later proposed a graph theory approach to solving it. An equivalent maximum bipartite matching problem is developed to reduce the computation complexity, and a polynomial-time algorithm is adopted to develop the schedule. Ojo et. al [9] formulated an energy consumption model and aimed to address the scheduling problem as an energy efficiency maximization problem. The authors proposed two mechanisms; the first being a low-complexity energy-efficient scheduler, and the second being Vogel's Approximation Method Heuristic Scheduling Algorithm (VAM-HSA). Another centralized scheme was proposed in [19] that presents an Adaptive Multi-hop Scheduling (AMS) method to provide multi-hop scheduling and low latency. This algorithm also takes into consideration possible transmission required near the PANC, thus allocating additional resources to vulnerable links. The paper also introduces the idea of virtual traffic as a way to allocate additional resources for retransmission.

## III. System Model

### A. Network Topology

A cluster-tree network topology is considered depicting an IEEE 802.15.4 TSCH MAC network as shown in Fig. 2. Such a network topology is suited for various applications that aim to minimize energy consumption during data transmission by exploring the parent-child hierarchy links. The network

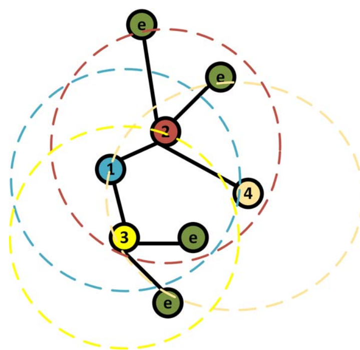



Fig. 3. Transmission range of various coordinator devices.

TABLE I
MAIN NOTATION DEFINITION

| Symbols | Definition |
|---|---|
| $T_{\text{next-link}_n}$ | $n$th constant time waiting for the next transmission link to destination before attempting CCA. |
| $T_{CCA}$ | Time required in $CCA$. |
| $T_{ta}$ | Turn around time |
| $T_l$ | Time taken to transmit frame of length $l$. |
| $ACK_{\text{wait}}$ | Time spent in waiting for acknowledgement from the coordinator. |
| $ACK_{\text{rec}}$ | Time required in receiving the ACK. |
| $E_x$ | Energy consumed after completing a specific operation. |
| $T_{n\text{BCK}_{\text{next-link}}}$ | Random number of shared links that must be skipped before re-transmission attempt. |

comprises of coordinators (FFDs) and end-devices (FFDs or RFDs). A coordinator initiating the network is termed as the Personal Area Network Coordinator (PANC). It is responsible for maintaining the PAN. The PANC forms the first cluster of the network. Devices join the network by associating themselves with a coordinator. End devices are devoid of routing capabilities and simply associated with a neighboring coordinator. They transmit all the sensed data to the associated parent coordinator.

### B. Collision Domain

A 2-hop collision domain is considered in this work. In general, 2-hop distance is considered as the transmission range for devices [20]. But, in dense networks, it is observed that collisions beyond 2-hops are also possible. The collision probability of such networks was studied in [21], [22]. It is important to understand that the carrier-sense range of devices is longer than its transmission range. This paper considers a sparse network topology with both transmission and sensing range (collision domain) up to 2-hops for simplicity. Fig. 3 shows a single cluster from Fig. 2. The sensing/transmission range is marked with dashed circles so as to demarcate the repetition of channels clearly. To avoid complexities and overlap in the figure, we avoided drawing the sensing range for the end-devices *e*.

### C. Markov Model

This subsection aims to compute the transmission time and the associated energy consumption during frame transmissions for the TSCH CSMA/CA mechanism, including re-transmissions. An IEEE 802.15.4 based TSCH network topology with *n* devices is considered. It is safely assumed that a schedule exists for communication. Each of the timeslots can either be a dedicated link or a shared link according to the schedule. In dedicated links, the allotted pair of devices start the transmissions immediately at the beginning of the timeslot. On the other hand, all devices must initially perform a single clear channel assessment (CCA) in the shared links. In the case of frame transmissions, the retransmission backoff mechanism of TSCH is used [2].

The proposed Markov model for TSCH CSMA/CA and retransmission mechanism is presented in Fig. 4. Here, we represent each state with a 4-valued tuple (i, j, CCA, rnd) (*i*, *j*, *CCA*, *rnd*), where $i = 0, \ldots, 7$ signifies the `macMaxFrameRetries` parameter, $j = 0, \ldots, 5$ signifies the `macMaxCSMABackoffs` and *rnd* ranges from $0$ to $2^{BE} - 1$ that signifies the random number of shared links that must be skipped before attempting transmission. Prior to attempting a frame transmission, each node performs CCA operation.

The transmission time of a frame in a shared link is given by

$$\begin{aligned} Tx_n \;=\; & \sum_{i=1}^{n} T_{\text{next-link}_n} + nT_{CCA} + T_{ta} + T_l \\ & + ACK_{wait} + ACK_{rec} \end{aligned} \tag{2}$$

For transmissions in dedicated links, the transmission time is given by

$$Tx \;=\; T_{\text{next-link}} + T_{ta} + T_l + ACK_{wait} + ACK_{rec} \tag{3}$$

The energy consumption in shared transmission links is

$$\begin{aligned} \mathrm{E}_{tx_n} \;=\; & E_x\left(T_{CCA}\right) + E_{ta}T_{ta} + E_{tx}T_l \\ & + E_x\left(ACK_{wait} + ACK_{rec}\right) \\ & + (n-1)E_x\left(T_{CCA}\right) \end{aligned} \tag{4}$$

and for dedicated communication link is expressed as

$$\mathrm{E}_{tx} \;=\; E_xT_{ta} + E_xT_l + E_x\left(ACK_{wait} + ACK_{rec}\right) \tag{5}$$

Retransmissions may be encountered only during the shared links as dedicated links are used by a single pair of devices. For shared links, retransmission denoted is by *i* in the Markov model(`macMaxFrameRetries` parameter in TSCH backoff algorithm). The transmission time and energy consumed is given by

$$\begin{aligned} \mathrm{RTx}_n \;=\; & Tx_n + \sum_{i=1}^{n} T_{n\text{BCK}_{\text{next-link}}} + nT_{CCA} + T_{ta} + T_l \\ & + ACK_{wait} + ACK_{rec} \end{aligned} \tag{6}$$

$$\begin{aligned} \mathrm{E}_{Rtx_n} \;=\; & \mathrm{E}_{tx_n} + E_x\left(T_{CCA}\right) + E_{ta}T_{ta} + E_{tx}T_l \\ & + E_x\left(ACK_{wait} + ACK_{rec}\right) \\ & + (n-1)E_x\left(T_{CCA}\right) \end{aligned} \tag{7}$$

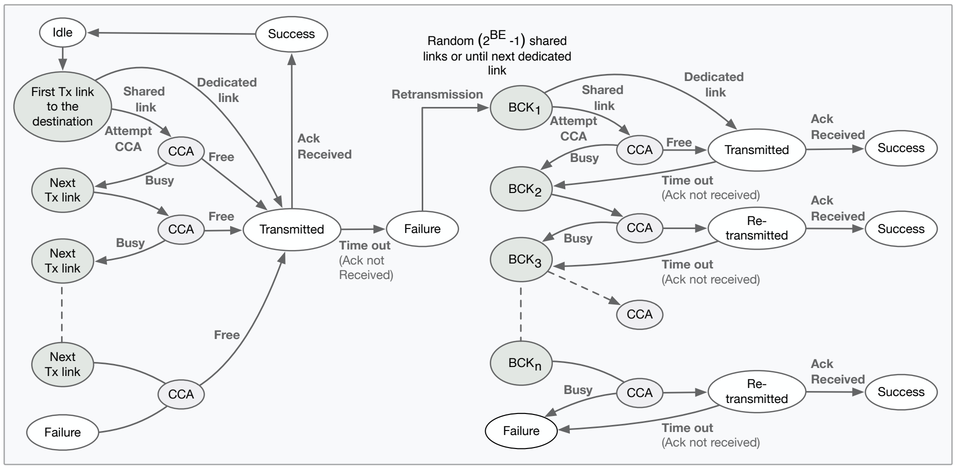



Fig. 4. Markov model for TSCH CSMA/CA and retransmission backoff mechanism.

```
algorithm BronKerbosch1(R, P, X) is
    if P and X are both empty then
        report R as a maximal clique
    for each vertex v in P do
        BronKerbosch1(R ∪ {v}, P ∩ N(v), X ∩ N(v))
        P := P \ {v}
        X := X ∪ {v}
```

Fig. 5. Bron–Kerbosch algorithm. [23]

## IV. Proposed TSCH Scheduling Scheme

### A. Forming the Collision Graph and Sub-graph

The set of links which, when scheduled together, will **not** result in interference and collision form a complete subgraph in the collision graph. All the active links in the network are represented as nodes in the collision graph. Two nodes (links of network) are adjacent in a collision graph, if and only if they won't interfere with each other if scheduled simultaneously. The edges of the original topology are represented as nodes in the collision graph.

A clique is defined as subset of vertices wherein every two distinct vertices in the clique are adjacent. Therefore, for a given graph G, the clique is an induced complete sub-graph. In the proposed mechanism, we make use of Bron–Kerbosch algorithm as a subprocedure for finding complete sub-graphs of a graph. The Bron–Kerbosch algorithm [23] is an enumeration algorithm for finding all maximal cliques in an undirected graph. The basic form of the Bron–Kerbosch algorithm is a recursive backtracking algorithm that searches for all maximal cliques in a given graph G. Given three disjoint sets of vertices R, P, and X, it finds the maximal cliques that include all of the vertices in R, some of the vertices in P, and none of the vertices in X. The algorithm is shown in Fig. 5

### B. Proposed TSCH Scheduling Mechanism

The proposed TSCH scheduling mechanism is a centralized scheme. The PANC is assumed to be aware of the entire topology, hence, creates a graph of the topology. It derives a collision graph for each cluster in the topology to schedule non-overlapping timeslots as described in the previous subsection. From the collision graph, Bron–Kerbosch algorithm is recursively applied to find the complete sub-graphs. The nodes in complete sub-graphs represent the links that are to be scheduled. These links denote the pairwise communication between two associated (parent-child) devices. These pairs of devices can be scheduled in the same timeslot but at different channels. The pseudo-code for the proposed mechanism is shown in Algorithm 1.

**Algorithm 1: Proposed TSCH Scheduling**

**1. Declare sets B = {}, S = {}, L = {}.**
**2. N(u,v) = Maintaining a 1 hop neighbor Set for each pair of adjacent nodes.**
**3. forEach : t $\epsilon$T, do :**
**4. S = Set of packets that need to be transferred in timeslot t (packets that have not reached PANC yet).**
**5. if ( S.empty()) {**
**6. break; }**
**7. L = Links that will be active in advancing all packets of S.**
//for every link in L, it is a pair of 2 nodes, u and v, therefore (u,v) is a link.
**8. CollisionMatrix : (u1,v1),(u2,v2) 1, iff (u1,v1),(u2,v2) don't result in collision of interference, 0 otherwise.**
// CollisionMatrix can be a adjacency matrix of a graph where (u1,v1) is a node and (u2,v2) is another node.
// All links (u,v) will have an edge to other links iff they don't result in a collision.
**9. { Call Subprocedure (Bron–Kerbosch algorithm) to find a complete subgraph of CollisionMatrix G. }**
**10. Advance packets remaining in buffer B.**
**11. Advance packets of the links present in G, if sufficient channels are available, otherwise select packets with top priority, and increase priority of remaining packets, and keep them in buffer B.**

**Illustrative example**: In this section, we perform numerical analysis depicting timeslot allotment using the proposed algorithm. Let us consider a cluster-tree network topology as shown in Fig. 6. We calculate the collision matrix G, then obtain complete subgraphs from it, and finally arrive at the links to be scheduled in this timeslot.

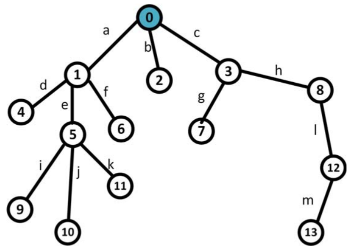


Fig. 6. Illustrative example.

| | a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| b | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| c | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| d | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| e | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| f | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| g | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| h | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| i | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| j | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| k | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| l | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| m | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Fig. 7. Collision Matrix G.

The collision matrix G has been calculated by taking into account the fact that a node in the original network can either receive or transmit at a time but not both simultaneously. Here, 'a' represents (0,1) edge 'b' represents (0,2) edge and so on. Since link 'a' won't interfere with links {g, h, i, j, k, l, m}, the entry G['a']['g','h'...] is 1, whereas links {b, c, d, e, f} do not interfere with link 'a' if they simultaneously transmit/receive data . Therefore, they have entry 0 in the G matrix. In this manner, the entire collision matrix G is built as shown in Fig. 7. The collision matrix G also represents a graph where the nodes represent the links in the original network topology. Next, we find complete subgraphs of G using with Bron–Kerbosch algorithm. The subgraphs are: {(a g i l), (a g i m), (a g j l), (b f g k l), (b f g k m), (b f h i m), (a g j), (a g i), (c f i m), (c f j l), (b f g i m)}. The nodes in the above complete subgraphs of G, represent the links that are to be scheduled. We can choose one of the above subgraphs to schedule links in TSCH. Let us consider {a, g, i, l} that represents (0,1),(5,9),(3,7),(8,12) edges in actual topology. Assuming 4 channels to be available. Since the number of channels is more than or equal to the available links that can be scheduled simultaneously, we schedule the links easily. This is shown in Fig. 8. Similarly, {b, f, g, i, m} represents (0,2),(1,6),(3,7),(5,9),(12,13) Since the number of channels is less than the available links that can be scheduled simultaneously, one of the links that have to be scheduled is kept in a buffer and those packets aren't advanced towards the PANC in this timeslot and is scheduled for the next slot. This is shown in Fig. 9. Since the traffic conditions in the network changes as more and more frames reach the PANC, graph G is recomputed for every timeslot, repeating the above process to get a complete link schedule.

| Channel | 0 | 1 | 2 |
|---|---|---|---|
| 1 | 0->1 | | |
| 2 | 5->9 | | |
| 3 | 3->7 | | |
| 4 | 8->11 | | |

Fig. 8. Schedule for {a, g, i, l}

| Channel | 0 | 1 | 2 |
|---|---|---|---|
| 1 | (0->2) | (12-13) | |
| 2 | (1->6) | | |
| 3 | (3->7) | | |
| 4 | (5->9) | | |

Fig. 9. Schedule for {b, f, g, i, m}

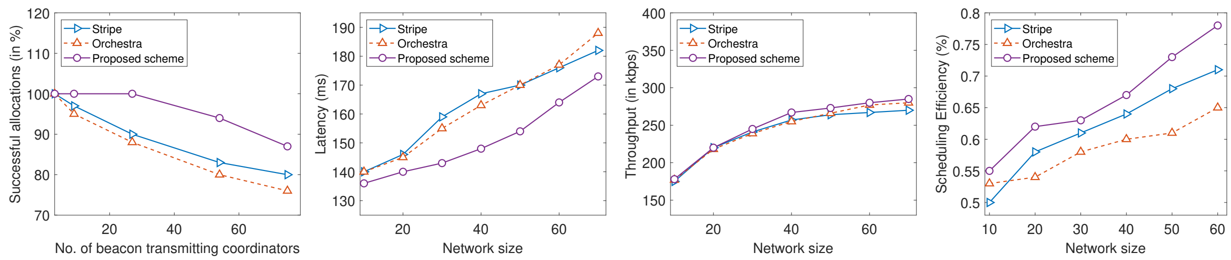
TABLE II
SIMULATION PARAMETERS

| Parameters | Values |
|---|---|
| Slot duration | 0.010 millisecond |
| Slotframe length | 101 symbols |
| Initial Energy | 10 J |
| Energy consumed to receive a frame | 0.003 J |
| Energy consumed to transmit a frame | 0.006 J |
| Energy consumed during sleep-state | 0.000 030 J |

## V. PRELIMINARY RESULTS & INSIGHTS

This section presents a few experimental results on the proposed TSCH scheduling mechanism to measure its performance. The experiments were performed in 6TiSCH [24] simulator. Table II presents the parameter values of the simulation. The insights are as follows:

- The main objective of any scheduling mechanism is to successfully allot non-overlapping slots for transmission, i.e., no transmission conflicts with other coordinators. Fig. 10(a) shows the percentage of successful slot allotment with an increase in network size. A 7-12% improvement is achieved compared to the related schemes.
- Latency and throughput are one of the essential performance metrics of scheduling schemes as it gives a measure of the time consumed in transmission, and successfully transmitted bits. Fig. 10(b) and Fig. 10(c), depicts the latency (15% improvement) and total MAC goodput (2% improvement) for proposed mechanism, Orchestra [3] and Stripe [6], respectively.
- We define scheduling efficiency as the optimal usage of available resources, i.e., minimal timeslots and maximum channels, with respect to the allocation of TSCH timeslots. This helps us achieve a reduction in channel wastage, and the overall network throughput and scalability can be significantly increased and simultaneously minimize latency. Fig. 10(d), shows the comparison of the scheduling efficiency of the proposed mechanism with Orchestra and Stripe. There is approximately 20% improvement in the proposed scheduling mechanism.




(a) (b) (c) (d)

Fig. 10. (a) Successful slot allotment, (b) latency comparison, (c) throughput comparison, and (d) scheduling efficiency comparison.

## VI. Conclusion

In this paper, we presented a centralized mechanism to schedule TSCH timeslots with optimal usage of resources. The proposed mechanism is based on complete sub-graphs derived from the collision matrix of the topology. The links in a sub-graph can be simultaneously be scheduled in a single timeslot but across different channels. The proposed mechanism is computed by the PANC, which is assumed to have complete knowledge of the topology. We also proposed a Markov model to estimate the transmission time and energy consumption during transmission of frames using the TSCH MAC. The performance of the proposed mechanism is shown to outperform other related schemes.

## Acknowledgement

The authors would like to acknowledge Prince Sultan University (PSU) and Smart Systems Engineering lab for their valuable support and provision of research facilities that were essential for completing this work. Also, the authors would like to acknowledge the support of PSU for paying the registration fees of this publication.

## References

[1] "IEEE Std. 802.15.4- Low-Rate Wireless Personal Area Networks (WPANs)," http://www.ieee.org/Standards/, pp. 1–314, Sept. 2011, [Online accessed: 25-Jan-2020].

[2] "IEEE Std. 802.15.4- Low-Rate Wireless Networks," http://www.ieee.org/Standards/, pp. 1–800, July. 2020, [Online accessed: 25-Aug-2020].

[3] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust Mesh Networks through Autonomously Scheduled TSCH," in *Proc. Conf. Embedded Netw. Sensor Sys.*, Nov. 2015, pp. 337–350.

[4] X. Fafoutis, A. Elsts, G. Oikonomou, R. Piechocki, and I. Craddock, "Adaptive Static Scheduling in IEEE 802.15.4 TSCH Networks," in *World Forum on Internet of Things*, May 2018, pp. 263–268.

[5] R. Soua, P. Minet, and E. Livolant, "Wave: A Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks," *Trans.Emerging Telecommun. Technol.*, vol. 27, no. 4, pp. 557–575, Apr. 2016.

[6] I. Juc, O. Alphand, R. Guizzetti, M. Favre, and A. Duda, "Stripe: A Distributed Scheduling Protocol for 802.15.4e TSCH Networks," LIG Research Report RR-LIG-54 HAL Open Archive, Inria, Tech. Rep., 2017, [Online accessed: 25-Jan-2020]. [Online]. Available: https://hal.inria.fr/hal-01585904/document

[7] G. Patti and L. L. Bello, "A Priority-aware Multichannel Adaptive Framework for the IEEE 802.15.4e-LLDN," *IEEE Trans. Ind. Electron.*, vol. 63, no. 10, pp. 6360–6370, Oct. 2016.

[8] M. Ojo and S. Giordano, "An efficient centralized scheduling algorithm in ieee 802.15. 4e tsch networks," in *IEEE Conference on Standards for Communications and Networking (CSCN)*, Nov. 2016, pp. 1–6.

[9] M. Ojo, S. Giordano, G. Portaluri, D. Adami, and M. Pagano, "An Energy-efficient Centralized Scheduling Scheme in TSCH Networks," in *IEEE Intl. Conf. Commun. Workshops*, July 2017, pp. 570–575.

[10] M. M. Nasralla and I. U. Rehman, "Qci and qos aware downlink packet scheduling algorithms for multi-traffic classes over 4g and beyond wireless networks," in *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Nov. 2018, pp. 1–7.

[11] M. M. Nasralla and M. G. Martini, "A downlink scheduling approach for balancing qos in lte wireless networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept. 2013, pp. 1571–1575.

[12] M. M. Nasralla, "A hybrid downlink scheduling approach for multi-traffic classes in lte wireless systems," *IEEE Access*, vol. 8, pp. 82 173–82 186, Apr. 2020.

[13] M. M. Nasralla, N. Khan, and M. G. Martini, "Content-aware downlink scheduling for lte wireless systems: A survey and performance comparison of key approaches," *Computer Communications*, vol. 130, pp. 78–100, July 2018.

[14] N. Choudhury, R. Matam, M. Mukherjee, and J. Lloret, "A Performance-to-Cost Analysis of IEEE 802.15.4 MAC With 802.15.4e MAC Modes," *IEEE Access*, vol. 8, pp. 41 936–41 950, Feb. 2020.

[15] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, "Beacon Synchronization and Duty-cycling in IEEE 802.15.4 Cluster-tree Networks: A Review," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1765–1788, June 2018.

[16] H. Kurunathan, R. Severino, A. Koubâa, and E. Tovar, "RPL over DSME: A Technical Report," CISTER-Research Centre in Realtime and Embedded Computing Systems, Tech. Rep., 2018.

[17] P. H. Gomes, T. Watteyne, and B. Krishnamachari, "MABO-TSCH: Multihop and Blacklist-based Optimized Time Synchronized Channel Hopping," *Trans. Emerging Telecommun. Technol.*, vol. 29, no. 7, p. e3223, July 2018.

[18] P. Li, T. Vermeulen, H. Liy, and S. Pollin, "An Adaptive Channel Selection Scheme for Reliable TSCH-based Communication," in *Intl. Symp. Wireless Commun. Sys.*, Apr. 2016, pp. 511–515.

[19] 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," in *IEEE Wireless Commun. Netw. Conf.*, Sept. 2016, pp. 1–6.

[20] N. Choudhury, R. Matam, M. Mukherjee, and J. Lloret, "LBS: A Beacon Synchronization Scheme with Higher Schedulability for IEEE 802.15.4 Cluster-tree Based IoT Applications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8883–8896, Oct. 2019.

[21] N. Choudhury, R. Matam, M. Mukherjee, and L. Shu, "Distributed Beacon Synchronization Mechanism for 802.15.4 Cluster-tree Topology," in *Springer Intl. Wireless Internet Conf.*, Dec. 2016, pp. 10–20.

[22] N. Choudhury and R. Matam, "Distributed Beacon Scheduling for IEEE 802.15.4 Cluster-tree Topology," in *Proc. IEEE Intl. Conf. India*, Dec. 2016, pp. 1–6.

[23] H. Johnston, "Cliques of a graph-variations on the bron-kerbosch algorithm," *International Journal of Computer & Information Sciences*, vol. 5, no. 3, pp. 209–238, 1976.

[24] "Simulator for TSCH-based Networks," https://bitbucket.org/6tisch/simulator/src/master/, [Online accessed: 25-July-2021].