Contention Based Proportional Fairness (CBPF) Transmission Scheme for Time Slotted Channel Hopping Networks
Bommisetty Lokesh and T G Venkatesh

Abstract—Time Slotted Channel Hopping (TSCH) is a Medium Access Control (MAC) protocol introduced in IEEE802.15.4e standard, addressing low power requirements of the Internet of Things (IoT) and Low Power Lossy Networks (LLNs). The 6TiSCH Operation sublayer (6top) of IEEE802.15.4e defines the schedule that includes sleep, transmit and receive routines of the nodes. However, the design of schedule is not specified by the standard. In this paper, we propose a contention based proportional fairness (CBPF) transmission scheme for TSCH networks to maximize the system throughput addressing fair allocation of resources to the nodes. We propose a convex programming based method to achieve the fairness and throughput objectives. We model TSCH MAC as a multichannel slotted aloha and analyse it for a schedule given by the 6top layer. Performance metrics like throughput, delay and energy spent per successful transmission are derived and validated through simulations. The proposed CBPF transmission scheme has been implemented in the IoT-LAB public testbed to evaluate its performance and to compare with the existing scheduling algorithms.

Index Terms—Internet of Things (IoT), IEEE 802.15.4e, Time Slotted Channel Hopping(TSCH), Stochastic modelling, Multichannel Slotted ALOHA.

I. INTRODUCTION

In recent times, deployment of Low Power and Lossy Networks (LLN) has been boosted by the emergence of Internet of Things (IoT) [1]–[3]. LLNs consists of low complexity resource-constrained devices with constraints on processing power, memory, energy as they are mostly battery operated. Wireless sensor and actuator networks (WSANs) are perceived as a dominating technology in many application domains and are expected to play a crucial role in the realization of the future of IoT. WSANs were already proved effective in many application domains such as smart cities, healthcare, industrial automation etc., which are a result of rapidly expanding IoT networks [1], [2]. Traffic generated by such devices will be sporadic but demand low latency and energy efficiency because of the devices being battery operated.

The IEEE 802.15.4e standard based on Time Slotted Channel Hopping (TSCH), addresses the low energy requirements of IoT networks. TSCH behavioral mode of IEEE 802.15.4e combines time slotting and multichannel communication with channel hopping capabilities to provide predictable latency, communication reliability and energy efficiency [3]. The channel hopping mechanism of TSCH allows overcoming the impact of multipath fading by changing the communication channel for every transmission. TSCH is widely implemented in several use cases like industrial monitoring and automation, smart homes, healthcare and environment monitoring [2]. TSCH devices transmit in resource units called cells which are defined by the timeslot index and the channel offset.

A sublayer called 6TiSCH operation sublayer (6top) is built on top of 802.15.4e TSCH MAC as shown in Fig. 1 to make the routing protocol for LLNs (RPL) compatible with 802.15.4e devices. RPL is the routing protocol developed to address several application specific optimizing objectives like minimizing energy, minimizing latency in LLNs [5]. 6top layer also enables the adoption of IPv6 over TSCH [4]. Apart from these functionalities, 6top sublayer additionally enables scheduling to provide differential treatment to network devices, hereafter termed as nodes, as the demands at each node is different to satisfy the application demands such as low end-to-end latency, high reliability and energy efficiency [6].

The 6top layer in IEEE 802.15.4e TSCH will only define the execution of a given schedule without specifying how to create and maintain a proper link scheduling. TSCH schedule defines the share of resources a node can use which is determined by the scheduling algorithms. In TSCH, scheduling helps to effectively allocate resource cells to wireless links. The design of a schedule will have direct impact on the network efficiency, including node energy consumption, latency, network throughput and reliability. These scheduling algorithms may be targeted to optimize different performance parameters constrained over the load conditions, application demand, latency requirement etc., at each node.

Several TSCH scheduling algorithms have been proposed in literature. They can be broadly classified into three categories:
centralized, distributed and autonomous scheduling [7]. In
centralized scheduling, a central entity or controller builds
the schedule and broadcasts it to all the nodes in network.
In distributed scheduling, the schedule of a node is built on
the basis of its negotiations with the neighbouring nodes. In
the third type of scheduling called autonomous scheduling,
the schedule of each node is constructed by that node au-
tonomously. These three types of scheduling are in the order
of reduction of network infrastructural costs.

In this paper, our attention is on the distributed scheduling
algorithms. Several distributed algorithms proposed in litera-
ture are contention free in nature. Generally, contention free
schedules require a well built and centralised infrastructure for
the network which is not cost effective [8]. Contention free
schedules also induce large overhead messages to the payload
data. The IoT and sensor networks in general generate sporadic
traffic in which case reservation based scheduling algorithms
are highly inefficient in terms of resource utilisation. Be-
sides, in dense networks reservation based schedules introduce
long transmission delays. Hence contention free scheduling
algorithms become highly inefficient in dense IoT networks.
Whereas the contention based scheduling achieves higher link
throughput for sporadic traffic however inducing the frequent
collisions resulting in retransmission of data packets.

Medium access algorithms are used in wireless networks
to control access to a shared wireless medium, and thereby
reduce collisions, ensure high system throughput, and distrib-
ute the available bandwidth fairly among the competing
streams of traffic. The problem of fair rate control at the
transport layer have been extensively researched, however the
fair resource allocation at the MAC design has not been
adequately addressed in TSCH networks [8].

Aim of this paper is to propose a contention based
distributed scheduling algorithm for TSCH networks that
achieves proportional fairness. We model the TSCH MAC
layer as a multichannel slotted aloha system where the trans-
mission probability of each node in a slot is given by the
scheduling algorithm. Because of the scheduling, the network
nodes behave heterogeneously while contending for the shared
resources. In view of growing IoT networks and application
diversity, it is of extreme importance that the performance of
a TSCH network under the heterogeneous behaviour to be
analysed.

The major contributions of our paper are as follows.
• We propose a proportional fairness contention based
  schedule for TSCH networks.
• We model and analyze the TSCH MAC protocol as
  a multichannel slotted aloha wherein the transmission
  probability of a node in a time slot is given by the
  schedule.
• We mathematically derive the performance metrics,
  namely throughput, packet transmission delay and energy
  spent per successful transmission of TSCH network for
  a given schedule.
• We have implemented and evaluated the proposed CBPF
  MAC protocol in the IoT-LAB [9] public testbed and
  compared its performance with the existing scheduling
  algorithms.

The reminder of the paper is organised as follows. We
discuss the background and literature of TSCH scheduling
algorithms to form the motivation of our work in Section [II]
The IEEE 802.15.4 e TSCH network is modelled as a generic
multichannel slotted aloha network. We formally define the
possible conflicts during transmissions in our system model
mathematically in Section [III]. We formulate the problem of
finding optimal transmission scheme for the random access in
TSCH networks as a convex optimisation problem in Section
[IV]. In Section [V] we apply the random access transmission
scheme proposed in Section [IV] to determine the transmission
probabilities of the nodes in a data collector network. The
performance of the proposed transmission scheme on the data
collection network is mathematically analysed in Section [VI].
We present the analytical results and validate them using simu-
lations and testbed implementation along with the comparative
analysis with the existing scheduling algorithms in Section
[VII]. Finally we conclude our paper in Section [VIII].

II. BACKGROUND AND RELATED WORK

TSCH schedule can be explained in two steps. Firstly, a
scheduling algorithm (also known as scheduling function SF)
specifies the design of schedule to the 6top sublayer. Schedul-
ing algorithm decides the number of cells to be added/deleted
from the schedule of a node. However, the implementation of
a schedule is done in 6top layer by the 6top protocol (6P)
which is defined by IETF in the standard.

A. 6top Protocol (6P)
The 6top protocol (6P) is defined in RFC 8480 [10]. 6P
transaction is a pairwise negotiation mechanism for cell
negotiation thus allowing the neighbours to agree on the
schedule in distributed manner. A 6P transaction can be an
ADD/Delete/Relocate request of cells to/from/in a schedule
between two nodes. 6P transaction can consist of two or
three steps. To schedule a link from node A to node B, in
2-step 6P transaction, node A proposes the set of cells to
be allocated. While in 3-step 6P transaction, node B puts
forward the cells to be allocated. The scheduling algorithm
must specify whether to use 2-step or 3-step transaction. In
either type of transactions, once a node proposes a set of
cells for the schedule, it locks the cells and waits for the
confirmation of the final list of cells from its neighbour. Once
the 6P transaction is complete, the cells which are agreed upon
will be added/deleted to/from the schedule of link from node
A to node B.

B. Scheduling algorithms
In their survey [7], Kharb et al. presented more than forty
TSCH scheduling algorithms and classified them into cen-
tralized, distributed, autonomous and hybrid scheduling algo-
rithms. Distributed scheduling algorithms are well suited for
the networks with dynamic topologies and also for large and
dense networks. Hence they occupy the major share (55%) of
the total scheduling algorithms. Most of these scheduling algo-
rithms consider the required bandwidth, system throughput,
energy spent per packet transmission etc., as the optimizing
parameters in their design.

The IETF 6TiSCH adopts a default scheduling algorithm
called minimal scheduling function (MSF) in RFC8180. MSF uses a combination of Id-based scheduling and neighbor to neighbor scheduling which means a hybrid of autonomous and distributed scheduling. Paletella et al. proposed a distributed scheduling scheme called On-the-fly (OTF) [11]. OTF calculates the number of cells to be allocated to each link based on the estimated traffic on that link. But as the links are not aware of the traffic on the neighbouring links, OTF is prone to collisions and exhibits a poor performance for dense networks. Kravelska et al. proposed a queue length based load balancing distributed scheduling algorithm called local voting [12]. In local voting, the authors considered a weighted allocation of cells to a link depending on their neighbouring link’s traffic.

A static TSCH scheduling algorithm proposed by Park et al. reduces the control message collisions to achieve high packet delivery ratio [13]. This is achieved by individually defining the broadcast and unicast slots. Kim et al. proposed a link based scheduling algorithm called ALICE considering the possible collisions into the design of scheduling [14]. ALICE does a mapping between link and their transmission cells using a hash table, which becomes NP hard to find the optimal mapping for larger and time-varying networks. To handle the unpredictable changes in the network due to time varying traffic and network topology, Jeong et al. proposed a traffic aware scheduling algorithm called TESLA [15]. TESLA allows each to change the slotframe size according to its traffic load. However, this slotframe size is propagated to the neighbouring nodes. In OST [16], depending on the traffic load, nodes dynamically adapt the frequency of transmitting cells, minimising the energy consumption.

Javan et al. proposed a combinatorial multi armed bandit approach to achieve optimal throughput in TSCH networks [17]. In their model, the candidate cells for scheduling are determined using bipartite graphs and coloring methods in graph theory based on the load and network topology. This approach completely avoids the collision, but at higher loads, the greedy approach of cell allocation may limit resources to the nodes with lower demand resulting in an unfair resource allocation.

All these scheduling algorithms calculate the number of cells to be added/deleted from a particular node’s schedule. Once the scheduling algorithm decides the number of cells to be added/deleted, 6top protocol (6P) triggers a negotiation between neighbouring nodes to decide on the location of cells to be added/deleted in each node’s schedule. Besides, these scheduling algorithms [11], [13], [17] quantitatively does not measure the fair resource allocation. Ojo et al. emphasised the need for fair scheduling algorithms and commented on the incompatibility of the general wireless network scheduling algorithms to TSCH and IIoT networks. The authors proposed a heuristic min-max fair scheduling algorithm in their work [8].

C. Motivation

The 6P transactions occur simultaneously at all the nodes in network. Hence there is a chance that two interfering nodes (need not share a schedule) may put forward the same resource units as the candidate cells which are picked in random from the available set of cells inherently leading to contention. Hence negotiation procedure increases the convergence delay due to the collision prone scheduled cells. This phenomenon gets pronounced as the network grows large or becomes denser. Even worse, this approach is inefficient for dynamic network conditions as the renegotiation is time consuming and implies packet drops before the convergence of new schedule [18]. This inevitably lowers the throughput of the network. Besides, the candidate cells get locked for the entire duration of 6P transaction, irrespective of it being a successful or failure, reducing the utilization of resources. Scheduling algorithms in literature do not take into the account the possible contention during 6P transactions. Hence they are not designed to minimize the collisions during 6P transactions. But our algorithm takes into account the load of the network at each node and fixes the transmission probability of each node so as to maximise the fairness throughput. The transmission probabilities solved using our CBPF algorithm can be used to construct a slotframe instead of doing it in pseudo-random method. Proportional fairness was first introduced by Frank Kelly in [19] based on changing rate control for elastic traffic in computer network services. Proportional fairness maximises the overall utility rate allocations with a logarithmic utility function. Moreover, in Aloha networks, the optimization problem for attaining proportional fairness is convex and separable [20], which allows us to develop computationally simple algorithms for arriving at the transmission probabilities that attain the optimal rates.

Unique features of our paper differentiating from the existing literature are as follows.

- To the best of our knowledge, ours is the first work to propose a contention based scheduling scheme that maximizes the proportional fairness throughput of the TSCH network.
- Several works in literature [21]–[23] have analysed the performance of TSCH network governed by a schedule only through simulations. In contrast, we mathematically analyse the performance of TSCH network governed by our proposed transmission scheme and validate the results using simulations.

III. SYSTEM MODEL

We model the IEEE 802.15.4e-TSCH as a generic multichannel slotted aloha network as follows. Our model considers a 6TiSCH network built by the Routing Protocol for Low-Power and Lossy Networks (RPL) [5]. The network consists of a finite set $\mathcal{N} = \{1, 2, 3, \ldots, N\}$ of nodes and operates in discrete time slots. Each node sends its data to the sink node over multiple hops. Hence a node’s transmissions consists of data generated by itself and also the forwarding data of other nodes to the sink node. Without loss of generality, we consider a single-sink model although the proposed algorithm works with multiple sinks. Let $\mathcal{D}_n \subseteq \mathcal{N} \setminus n$ denote the subset of relay nodes which can be used by node $n$ to send data to sink. A node $n$ may attempt transmission of a packet to one of the nodes $m \in \mathcal{D}_n$, in one of the channels chosen randomly in a time slot. We define a link $(n, m)$ to denote a directed communication from the transmitter node $n$ to the receiver.
node $m$, where $n \neq m$, in a specific time slot and a channel. Let $S = \{(n, m) | n \in N, m \in D_n\}$ denote the set of all links in a network.

Time is divided into slots with slot index $t = 0, 1, 2, \ldots$. A time slot $t$ is long enough for the transmission of a data packet from source node to destination node and for the reception of corresponding acknowledgement. The available bandwidth in IEEE 802.15.4e TSCH networks is divided into orthogonal channels. Let the number of non-interfering channels available to TSCH mechanism be denoted by $M$. The set of available channels is denoted by $\mathcal{M} = \{ch_i | 1 \leq i \leq M\}$. A transmission occurs on a link using a resource unit called cell which is denoted by the time slot index and the channel offset as $(t, ch_i)$. The transmission on a link $(n, m)$ using the cell $(t, ch)$ is denoted by $c_{(n,m)}^{(t, ch)}$, where

$$c_{(n,m)}^{(t, ch)} = \begin{cases} 1, & \text{if transmission occurs on } (n, m) \text{ in cell } (t, ch) \\ 0, & \text{if no transmission on } (n, m) \text{ in cell } (t, ch) \end{cases}$$

(A) Primary Conflict

We consider the following reasonable assumptions on the communication capability of nodes in the network. Nodes do not have multi packet transmission / reception capability. In other words

(i) A node cannot simultaneously transmit on two or more links and cannot simultaneously receive over two or more links.

(ii) A node cannot transmit and receive simultaneously.

The above two conditions are called primary conflicting conditions. Primary conflict refers to packet loss irrespective of the channel on which the communication occurs. Fig 2a illustrates different scenarios when primary conflict occurs. The set of links which are in primary conflict with the transmission on the link $(n, m)$ is denoted and defined as follows

$$\mathcal{I}_{nm}^p = \{(l, k) | (l, k) \cap \{n, m\} \neq \phi\}$$

Equation (3) indicates that transmission cannot occur on two primary conflicting links at the same time slot $t$, even on different channels $ch_i$ and $ch_j$.

(B) Secondary Conflict

Secondary conflict occurs between the transmissions taking place on distinct links in the same interference range choosing the same channel in the same slot. It stems from the fact that a receiver cannot decode an incoming packet in a channel $ch$ if another node in its neighborhood is also transmitting on the same channel $ch$ in the same time slot $t$ due to channel interference. Let $\mathcal{N}_n$ be the set of nodes within the interference range of node $n$. It can be seen that $D_n \subseteq \mathcal{N}_n$ (one hop transmission is possible only within the interference range). The set of links that interfere with the ongoing transmission on link $(n, m)$ is denoted as

$$\mathcal{I}_{nm}^s = \{(l, k) | m \in \mathcal{N}_l \lor k \in \mathcal{N}_n\}$$

Equation (4) gives the set of links which interfere with the transmissions on $(n, m)$ if operated on same channel and an example scenario is given in Fig. 2b. It can be seen from (2) and (4) that $\mathcal{I}_{nm}^p \subseteq \mathcal{I}_{nm}^s$. Also, note that the links in $\mathcal{I}_{nm}^s \setminus \mathcal{I}_{nm}^p$ will not cause interference to link $(n, m)$ if operated on a different channel even in the same time slot. We call the links in $\mathcal{I}_{nm}^s \setminus \mathcal{I}_{nm}^p$ as secondary conflicting links of $(n, m)$. The secondary conflicting links cannot transmit on the same time slot and the same channel which is formally written as follows.

$$c_{(n,m)}^{(t, ch)} c_{(l,k)}^{(t, ch)} = 0, \text{if } (l, k) \in \mathcal{I}_{nm}^s \setminus \mathcal{I}_{nm}^p$$

Equation (5) indicates the interference constraint or the secondary conflict constraint. Fig. 2b shows a scenario of secondary conflict between two links.

Following the “Multi Channel Slotted Aloha” random access strategy, each node $n$ in TSCH network transmits its data or relays the forwarded data to the sink node in each time slot through a relay node $m \in D_n$ with probability $\tau_{nm}$, independent of other nodes and of the past transmission history. When node $n$ does a transmission, it randomly chooses a particular channel $ch_i \in \mathcal{M}$ for transmission.

In nutshell, the methodology of our work to arrive at the optimal transmission scheme can be explained as follows. Firstly, all links in the network are assigned weights according to the queue length i.e., the number of packets waiting to be transmitted on that particular link. Transmissions in the network are restricted by two constraints. (i) No two primary conflicting links should be scheduled in the same time slot. (ii) The average number of transmitting nodes in a neighbourhood in the same time slot cannot exceed the number of available channels. With these two constraints, we determine the optimal transmission probabilities of all the links in $S$ by maximising the weighted proportional fairness throughput utility function using the convex optimisation methods. Graphical illustration of the proposed random access algorithm is shown in Fig. 3 using a block diagram. Once the optimal transmission scheme is determined, we mathematically analyse the performance of the TSCH network for a data collection network in terms of the link throughput and delay.
transmission on link \((n,m)\) in time slot \(t\) and \(h_{nm}(t) = 0\) otherwise. Intuitively, a good scheduling algorithm is expected to keep the number of packets waiting in the queues to be finite. To make this notion formal, we define that if \(\lambda \notin \Lambda\) then no scheduling algorithm can keep queue lengths finite, where

\[
\Lambda = \{\lambda \in \mathbb{R}_+^S \mid \lambda \leq \mu \text{ component wise, for some } \mu \in C\}
\]

where \(\Lambda\) is the set of all arrival rate vectors such that the arrival rate at any node does not exceed its service rate. Each node assigns a weight to each outgoing link based on the number of packets queued for transmission on that link and updates it at the end of every slot. We denote the link weight by \(w_{nm}\) for a link \((n,m)\) ∈ \(S\). These link weights of each node \(n\) are shared among the neighboring nodes \(N_n\) for all \(n \in N\) periodically. The weight of each link is calculated as a function of queue length of that link as \(w_{nm} = f(Q_{nm})\). Several earlier works proposed different classes of weight functions. We adapt the class of weight functions proposed by authors in [24], in particular we choose the weight function to be \(f(x) = \log(1 + x)\), a positive, increasing and concave function of \(x\). Hence the weight of a link \((n,m)\) in terms of its queue length \(Q_{nm}\) is expressed as

\[
w_{nm} = \log(1 + Q_{nm})
\]

Proportionally fair link success probabilities \(\mu^*\) are attained by the transmission probability vector \(\tau^*\) that maximize the weighted proportional fairness objective function also known as utility function as given below

\[
\tau^* = \arg\max_{\tau \in \mathcal{P}} \sum_{(n,m) \in S} w_{nm} \log(\mu_{nm}(\tau))
\]

Lemma 1: The weighted proportional fairness objective function \(F = \sum_{(n,m) \in S} w_{nm} \log(\mu_{nm}(\tau))\) is concave with respect to \(\tau\).

Proof: Proved in Appendix A.

The maximum number of successful transmissions that can occur in a slot within the neighbourhood of a link is limited by the number of non-interfering channels \(M\) or the number of competing nodes whichever is lower. This gives us a constraint on the feasible link transmission probability vector. Hereby, we model the optimal link transmission probability problem as an optimization problem.

\[
\begin{align*}
\text{(P1):} \\
\text{minimize} & \quad \tau \in \mathcal{P} \quad -F = - \sum_{(n,m) \in S} w_{nm} \log(\mu_{nm}(\tau)) \\
\text{subject to} & \quad \tau_{nm} + \sum_{(l,k) \in \mathcal{E}_{nm}} \tau_{lk} \leq M \quad \forall (n,m) \in S
\end{align*}
\]

The objective function of the above optimization problem (P1) is convex since the function \(F\) is concave according to Lemma 1. The constraints are linear with \(\tau\). The domain \(\mathcal{P}\) of the function \(F\) is a hypercube in \(\mathbb{R}^N\) and hence the domain of (P1) is also convex. Hence the problem (P1) is a convex optimization problem [25]. It can be seen that the linear constraints always hold true for atleast one point in the domain \(\tau \in \mathcal{P}\). Hence the primal problem is said to be feasible.
As (P1) is a feasible convex optimisation problem, there exists a unique minimal point for the objective function and a unique link transmission probability vector $\tau^*$. In the following section, we derive the closed form expression of the optimal transmission probability vector at the border router level of a TSCH network where it acts as a data collector.

V. Optimal Transmission Probabilities for Data Collection Network

In this section, we mathematically derive the optimal transmission probabilities of nodes in the TSCH network at the border router. The border router receives data over multiple channels from multiple nodes forming a star topology. A network with this topology is often referred to as a data collection network.

In this network setting, each node will have only one outgoing link and therefore the links can be identified by the transmitter node itself. The link access probabilities and the link success probabilities of the link connected to node $i \in N$ are now denoted by $\tau_i$ and $\mu_i$, respectively. Further, all the links of the network are in interference conflict with each other. Therefore, success probability of a link $i$ is the product of $\tau_i$, probability of transmission on link $i$, and the probability that no other link $j$ transmits on the same channel as that of link $i$.

$$\mu_i = \tau_i \prod_{j=1, j \neq i}^{N} (1 - \frac{\tau_j}{M})$$

The convex optimization problem given by (P1) in (12) can be adapted for the data collection network as

(P2):

minimize$_{\tau \in P}$ $F(\tau) = -\sum_{i=1}^{N} w_i \log(\mu_i)$

subject to $\sum_{i=1}^{N} \tau_i \leq M$

where the condition that the optimization problem is subjected to refers to the average number of transmissions not exceeding the number of channels. The convexity of the objective function has been already proved for the general case in Appendix A. The objective function $F(\tau) : R^N_+ \rightarrow R$ is also called as primal function. The constraint in (14) is clearly a linear function of $\tau$, with the domain of $\tau$ being a hypercube in $R^N_+$. Therefore the above problem (P2) is clearly a convex optimisation problem. Let the primal point $f^*$ of $F(\tau)$ is defined as follows

$$f^* = \min_{\tau \in P} F(\tau)$$

(15)

Lagrange function $L$ for the convex optimization problem (P2), $L : R^N_+ \times R_+ \rightarrow R$ is given as

$$L(\tau, \gamma) = \sum_{i=1}^{N} -w_i \log(\mu_i) + \gamma \left( \sum_{i=1}^{N} \tau_i - M \right)$$

(16)

where $\gamma \in R_+$ is the Lagrangian multiplier. The dual function $G$ of the convex problem (P2) and the dual point $g^*$ are given as follows

$$G(\gamma) = \inf_{\tau \in P} L(\tau, \gamma)$$

(17)

$$g^* = \max_{\gamma \geq 0} G(\gamma)$$

(18)

where $P$ is the domain of $\tau$. Since the constraint in (14) is feasible in $P$, $(\exists \gamma^* \in P : \sum_{i=1}^{N} \tau_i^* \leq M)$, (P2) is a feasible convex optimisation problem for which the strong duality holds ($f^* = g^*$) by Slater’s condition [25]. Hence the Karush-Kuhn-Tucker (KKT) conditions [25] can be used to find the optimal solution ($\tau^*, \gamma^*$). The KKT conditions for the above optimisation problem (P2) are given by

1) $\sum_{i=1}^{N} \tau_i^* - M \leq 0$
2) $\gamma^* (\sum_{i=1}^{N} \tau_i^* - M) = 0$
3) $\frac{\partial L}{\partial \tau_i} = 0$

Here the first condition shows the feasibility of (P2). The second condition is called the complementary slackness condition. It says that if a dual variable is greater than zero (slack) then the corresponding primal constraint must be an equality (tight). It also says that if the primal constraint is slack then the corresponding dual variable is tight (or zero). To find the optimal transmission probabilities, we apply condition 3 on (16) as follows,

$$\frac{\partial L}{\partial \tau_i} = \frac{-w_i}{\tau_i} + \frac{\sum_{j=1, j \neq i}^{N} w_j}{M - \tau_i} + \gamma = 0$$

(19)

Rearranging (19), we get a quadratic equation as follows.

$$\gamma \tau_i^* + (\gamma M - \sum_{j=1}^{N} w_j) \tau_i^* + w_i M = 0$$

(20)

This quadratic equation has two roots of the form $\tau_i(\gamma)$ out of which only one of them will lie in the domain of (P2). Since the problem (P2) is convex and is feasible in its domain, the uniqueness of the optimal point is expected.

The unique optimal solution $\tau^*$ of (P2) is determined by finding the optimal dual variable $\gamma^*$ from (18) using first order optimisation methods like gradient descent or numerical methods. This concludes our discussion on the procedure to obtain the optimal transmission probability vector $\tau^*$ which can be used to achieve the proportional fairness scheduling in TSCH network.

VI. Performance Analysis

In this section, we analyse the performance of TSCH network operating using contention based proportional fairness scheduling algorithm derived in Section V. In particular, we derive the throughput, delay and energy spent per successful packet transmission by a node in the data collection network.

A. Number of nodes transmitting in a slot

Let $\tau_i, i = 1, 2, \ldots, N$ being the transmission probabilities of nodes in a data collection network. Let the random variable $K$ denote the number of nodes transmitting in a slot. $K$ can be treated as the sum of $N$ independent and non-identical Bernoulli random variables $I_j$. Where $I_j$’s indicate whether node $j$ transmits in a given time slot or not.

$$K = \sum_{j=1}^{N} I_j$$

(21)
Let us denote the IDFT of the expression from (21) and ϕ using characteristic function approach as follows. The characteristic function of $I_j$ is given as $\varphi_{I_j}(t) = 1 - \tau_j + \tau_j \exp(it)$ where $i = \sqrt{-1}$ [26]. Hence the characteristic function of $K$ is

$$\varphi_K(t) = \mathbb{E}[\exp(itK)] = \sum_{k=0}^{N} P_k(k; N) e^{ikt}$$

(23)

where $P_k(k; N)$ is the probability that out of $N$ nodes, $k$ nodes transmit and $N - k$ nodes differ transmission in a given time slot. Since $K$ is the sum of $N$ Bernoulli random variables, $\varphi_K(t)$ can be expressed as the product of individual characteristic functions $\varphi_{I_j}(t)$. Hence, (23) can be alternatively expressed from (21) and $\varphi_{I_j}(t)$ as follows

$$\varphi_K(t) = \prod_{j=1}^{N} \varphi_{I_j}(t) = \prod_{j=1}^{N} (1 - \tau_j + \tau_j \exp(it))$$

(24)

Let us denote the IDFT of the $N + 1$ length sequence $P_k(k; N)$ as $D(l)$. It can be seen that $D(l)$ can be obtained by substituting $t = \omega l$ in $\varphi_K(t)$ in (23) where $\omega = \frac{2\pi}{N+1}$. $D(l)$ which can be obtained from (23) and (24) as follows [26],

$$D(l) = \varphi_K(t) \big|_{t=\omega l} = \prod_{j=1}^{N} (1 - \tau_j + \tau_j \exp(i\omega l))$$

(25)

Since $D(l)$ is the IDFT of the sequence $P_k(k; N)$, DFT of $D(l)$ will result in $P_k(k; N)$ which is the probability that $k$ out of $N$ nodes transmit. Hence $P_k(k; N)$ is given by

$$P_k(k; N) = \frac{1}{N+1} \sum_{l=0}^{N} D(l) e^{-i\omega lk} = \frac{1}{N+1} \sum_{l=0}^{N} e^{-i\omega lk} \left( \prod_{j=1}^{N} (1 - \tau_j + \tau_j e^{i\omega l}) \right)$$

(26)

**B. Throughput**

Let us introduce the indicator random variables $X_i, i = 1, \ldots, M$ such that

$$X_i = \begin{cases} 1 & \text{if only one out of } k \text{ nodes transmit in } i^{th} \text{ channel} \\ 0 & \text{otherwise} \end{cases}$$

(27)

Probability that only one node out of $k$ nodes transmitting in the $i^{th}$ channel can be determined as follows. One out of $k$ nodes will choose $i^{th}$ channel with probability $1/M$, since there are $M$ channels, and the rest $k - 1$ nodes chooses other channels with probability $(1 - 1/M)^{k-1}$ as given below

$$P(X_i = 1) = \binom{k}{1} \frac{1}{M} (1 - \frac{1}{M})^{k-1}$$

(28)

$Y(k, M)$ be the random variable indicating the number of successful transmissions when $k$ nodes transmit in $M$ channels in a time slot and hence $Y(k, M) = X_1 + X_2 + \cdots + X_M$. The conditional expectation of the number of successful transmissions given $k$ nodes are transmitting in a time slot over $M$ channels is given by

$$E(Y(k, M)) = \sum_{i=1}^{M} E(X_i) = \sum_{i=1}^{M} 1P(X_i = 1) + 0P(X_i = 0) = \sum_{i=1}^{M} k \frac{1}{M} \left(1 - \frac{1}{M}\right)^{k-1} = k \left(1 - \frac{1}{M}\right)^{k-1}$$

(29)

System throughput denoted by $T$ is the average number of successful transmissions in a slot. $T$ is obtained from (29) by averaging $E(Y(k, M))$ over the distribution of number of transmitting nodes, $P_k(k, N)$.

$$T = \sum_{k=0}^{N} P_k(k; N) E(Y(k, M)) = \sum_{k=0}^{N} P_k(k; N) k \left(1 - \frac{1}{M}\right)^{k-1}$$

(30)

Equation (30) gives the system throughput of the considered TSCH network.

**C. Delay**

In this subsection we derive the average delay experienced by a packet in the TSCH network. As the transmission probability of each node is different, the average delay experienced by different nodes are different. Let us derive the delay experienced by an arbitrary node $i$, whose transmission probability is $\tau_i$. Let $P_k(k; N - 1)$ denotes the probability of $k$ other nodes transmitting in a slot apart from the tagged node. This is calculated from (26) as follows

$$P_k(k; N - 1) = \frac{1}{N} \sum_{l=0}^{N} e^{-i\omega lk} \left( \prod_{j=1, j\neq i}^{N} (1 - \tau_j + \tau_j e^{i\omega l}) \right)$$

(31)

where $\omega = \frac{2\pi}{N}$. Let $P(S/k)$ indicate the probability that a tagged transmission is successful given there are $k$ other transmissions. This is the probability that none of the $k$ transmissions choose the same channel as that of the tagged channel and is given as follows

$$P(S/k) = (1 - \frac{1}{M})^k$$

(32)

Let $p_i$ be the probability that a packet transmitted by a tagged node is successful. Therefore $p_i$ can be calculated from (31) and (32) by averaging $P(S/k)$ over the distribution of $k$ given by $P_k(k; N - 1)$ as follows

$$p_i = \sum_{k=0}^{N-1} P_k(k; N - 1) (1 - \frac{1}{M})^k$$

(33)

Let the random variable $S_i$ denotes the service time of a packet in units of slots. $P(S_i = n)$ is the probability that a packet takes $n$ slots to get served. In other words it experiences $n - 1$ failure slots followed by a successful transmission.

$$P(S_i = n) = (1 - \tau_i p_i)^{n-1} \tau_i p_i \quad n = 1, 2, 3, \ldots$$

(34)
From (34), it can be seen that the service times are geometrically distributed. The mean and second moment of the service time are calculated as below (26),

\[
\bar{S}_i = E[S_i] = \frac{1}{\tau_i p_i} \tag{35}
\]

\[
\bar{S}_i^2 = E[S_i^2] = \frac{2 - \tau_i p_i}{\tau_i^2 p_i^2} \tag{36}
\]

Consider that each node \( i \) maintains a queue to which packets arrive according to a Poisson process with rate \( \lambda_i \) and packet service times having a geometric distribution. Hence the total delay experienced by a packet from the time it entered the queue till it got served can be calculated by the Pollaczek–Khinchin (P-K) formula for the \( M/G/1 \) (27). With the average service time and the second moment of service time given by (33) and (36) respectively, the total delay experienced by a packet of \( i^{th} \) node is given as

\[
D_i = S_i + \frac{\lambda_i \bar{S}_i^2}{2(1 - \lambda_i \bar{S}_i)} \tag{37}
\]

### D. Energy

Consider that the tagged node takes \( n \) slots to transmit its packet successfully, whose distribution is given in (34). Let’s define an indicator random variable \( Y_j \), \( j = 1, 2, \ldots, n - 1 \) for all the unsuccessful slots. \( Y_j \) indicates the reason for the failure of transmission made by the tagged node in \( j^{th} \) slot.

\[
Y_j = \begin{cases} 
1 & \text{if transmission in } j^{th} \text{ slot encounter collision} \\
0 & \text{node differs from transmission in } j^{th} \text{ slot}
\end{cases} \tag{38}
\]

Probability that the tagged node transmits a packet and collision occurs in \( j^{th} \) slot is \( P(Y_j = 1) = \frac{\tau(1-p)}{1-\tau p} \). The expected number of collisions \( C \) a packet experiences before getting successfully transmitted in \( n^{th} \) slot is given by

\[
C = \mathbb{E} \left[ \sum_{j=0}^{n-1} Y_j \right] = \mathbb{E}[n-1] \mathbb{E}[Y_j] = \left( \frac{1}{\tau p} - 1 \right) \cdot \left( \frac{\tau(1-p)}{1-\tau p} \right) \tag{39}
\]

(39) is obtained from (39) \( a \) by Wald’s theorem (26), using the fact that \( Y_j \)’s are IID and independent to \( n \). Let \( E_{tx} \) be the energy spent per transmission attempt. In total, a node takes \( 1 + C \) transmission attempts for successfully transmitting a packet.
Fig. 8: Normalised link weights against the average packet arrival rates in a TSCH network with $N = 86$, $M = 15$

Hence the average energy $E$ spent to successfully transmit the packet of tagged node is

$$E = E_{tx} \cdot (1 + C) = \frac{E_{tx}}{p}$$  \hspace{1cm} (40)

VII. RESULTS AND DISCUSSION

A. Experimental Setup and Simulation Methodology

We implement our CBPF transmission scheme in Contiki OS [28], an open source operating system for IoT network devices. Contiki uses hybrid modular architecture based event-driven model. To verify analytical results, we evaluate the proposed CBPF scheme (Contiki implementation) on the IoT-LAB [9] public testbed installed in Grenoble, France. We deploy our contiki implementation in 86 M3 nodes (ARM Cortex M3) whose radio chip was designed to implement the MAC layer of the IEEE 802.15.4 standard. The COAP (Constrained application protocol) generates the payload data at each node. Application payload is carried in UDP/IPv6 datagrams over 6LoWPAN. Contiki-RPL is used at the routing layer to construct the network topology. Objective function of RPL is chosen as MRHOF [29] which builds a tree topology using estimated link transmissions (ETX) and hop count of nodes. The application payload data is carried in UDP/IPv6 datagrams over 6LoWPAN adaption layer. 6LoWPAN protocol enables the transmission of IPv6 datagrams over the IEEE 802.15.4 radio links by adding an adaptation layer between the data link layer (IEEE 802.15.4 e TSCH) and the network layer [30]. The CBPF transmission scheme proposed in this paper is used at the MAC layer to trigger data transmission at nodes to the border router.

CBPF uses the default slotframe structure define in the IEEE 802.15.4e standard. The CBPF control packets containing the queuelength details, weights of the nodes and the schedule are transmitted in the broadcast slots of the slotframe. The RPL control packets are transmitted based on the trickle timer mechanism as described in RFC 6206 [31] of the standard.

Each Cortex M3 node uses Tx power of -16.65 dBm and Rx power of -15.98dBm. Each node accommodate 4 sensors (light, pressure and temperature, magnetometer and gyroscope) which contribute to the traffic generated at that node. The control node (border router) records sensors data collected over the network from all nodes in TSCH network over multiple channels. Different traffic rates at different nodes required for our analysis of heterogeneous network is made possible by setting different update frequencies as well as by enabling only a subset of available sensors at a node. The sniffed data at the border router is analysed using Wireshark [32] and we calculate the necessary performance metrics of the network. Each experiment lasts for 1 hour so that sufficient number of packet transactions occur and 10 $\sim$ 15 such iterations are run over which the results are averaged.

Besides, we have performed Monte-Carlo simulations of TSCH network for the scheduled and unscheduled cases using MATLAB. Network consisting of $N$ nodes ranging from 1 to 30, all transmitting data to a border router is considered. The number of channels $M$ is fixed as 5, 10 or 15. We consider that the nodes transmit data to border router over a maximum of 5 hops. Throughput, delay and energy spent per transmission are obtained by averaging over 10000 iterations. Analytical results are obtained through numerical computation of the derived equations using MATLAB in line with the simulation parameters. The estimated link transmission is obtained through the simulations performed in 6TISCH simulator [21].

In this section, we present the simulation and experimental results for the CBPF algorithm proposed in section IV. We compare CBPF scheme with the minimal scheduling function (MSF) [33] described in the standard and also with a link scheduling algorithm, ALICE [14].

B. Homogeneous Network

We first consider a homogeneous network where all links are of equal importance i.e., all the links have same traffic arrival rate $\lambda_i$. Hence we consider $\lambda_i = 0.4$ for all links to simulate a network with number of nodes $N = 86$ and the number of channels $M = 15$. Since all links are homogeneous the weights of all links according to (10) are found to be same and the weight is $w_i = 0.18257$. With these given weights, we solve the optimization problem to find the transmission probabilities of network that optimizes the objective function. Fig. 4 shows that the dual function achieves its maximum
at $\gamma = 0.96$. The transmission probabilities of the nodes determined using (20) for $\gamma = 0.96$ is $\tau_i = 0.1744$. With this solution of the transmission probabilities, the system throughput achieved is 5.6115 packets/slot.

In Fig. 5, we plot the average system throughput with variation in the network size. The system throughput increases with the network size initially until it attains a maximum and gets saturated for the further increase in network size. The maximum value and hence the saturation occurs when the number of nodes in network is same as the number of available channels i.e., $N = M$. In traditional slotted aloha networks, the throughput attains its maximum at $N = M$, however it reduces for the higher values of $M$. It can also be observed that the peak throughput achieved is $T = 1.853.7$ and $5.52$ for $M = 5, 10$ and $15$ respectively, which is the maximum achievable theoretical throughput of slotted aloha systems ($T = 0.37 \times M$) [34].

The proposed optimal transmission scheme results in a linear increase of delay with $N$ as shown in Fig. 6, which otherwise would be an exponential. The average number of transmissions required by a node is shown in Fig. 7. The number of transmissions increases with increase in $N$ due to the increased contention but saturates after $N = M$ which can be explained as follows. From [39], it is clear that the number of collisions experienced by a node is inversely proportional to the successful packet transmission probability $p$. Yang et al. derived that $p = T/G$, where $T$ is the system throughput and $G$ is the traffic load i.e., the average number of transmission attempts in a slot [35]. In our model, due to the constraint in (14), the load will be $G = N$ when $N \leq M$ and $G = M$ for higher $N$. Also it can be seen from Fig. 5 that $T$ saturates after $N = M$. Putting these two facts together, the number of transmissions required also saturates after $N = M$. As the number of collisions is saturated, the energy required for a packet transmission also will be bounded and saturated. The theoretical results have been validated using simulations as well as the results obtained from the IoT-LAB testbed implementation.

C. Heterogeneous Network

Now, we consider a heterogeneous TSCH network where the packet arrival rates at each node in the network is specific to the application it serves. Hence the packet arrival rate for each node in the network can be different from that of the other nodes. We consider a network consisting of $N = 86$ nodes transmitting over $M = 15$ channels. Each node generates
In this subsection, we compare the performance of our CBPF transmission scheme, ALICE and MSF for a heterogeneous network of size 86 nodes packets according to a Poisson process whose mean value is different for different nodes in the range of (0, 0.2). Hence the links are assigned weights based on their queue lengths as described in Section IV. In Fig. 8, we plot the link weights against the packet arrival rate of the links. As the weightage function in (10) is logarithmic with the queue length, the weights assigned to the links also follow a logarithmic curve with respect to $\lambda_i$ in expectation as shown in Fig. 8. The proposed contention based scheduling algorithm calculates the transmission probabilities of the links as shown in Fig. 9 such that the weighted proportional throughput is maximized. With the obtained optimal transmission probabilities, the throughput of links is plotted in Fig. 10.

As the objective is to maximize the weighted proportional throughput, the throughput of each link should be proportional to its weight. Therefore, the link throughput should vary linearly with the weights. Hence the link throughput varies in logarithmic fashion with the packet arrival rate as shown in Fig. 10. In Fig. 11, we plot the service time and the total delay experienced by a packet against the packet arrival rate. We have validated the theoretical results with the IoT-LAB tested results. As the arrival rate increases, the weight assigned to the links also increases which results in the higher link throughput. Therefore the time required by a packet in the higher weight link is less compared to that of the lower weighted links. But on the other hand, as the arrival rate is higher, the number of packets in the queue builds up quickly leading to a larger waiting time of a newly arrived packet. While the links with very low arrival rate will almost see a zero queuing delay since the packets are serviced as soon as they arrive in the queue. Total delay experienced by a packet is the sum of its queuing delay and the service time. Since queuing delay is very less at the nodes with low arrival rates, service time and the total delay coincides with each other.

D. Comparative Analysis

In this subsection, we compare the performance of our CBPF transmission scheme, presented in [14], with the minimal scheduling function (MSF) [33] and ALICE [14]. Throughput and delay performance of the three algorithms are presented in Fig. 12 and Fig. 13 respectively. These results are collected from the IoT-LAB test bed by implementing MSF and ALICE scheduler in the deployed protocol stack. MSF allocates cells to links without taking into consideration of the possible collisions in network. Beyond that nodes have to poll for their share of resources following the 6P transactions, which causes additional collisions. Hence our algorithm outperforms MSF in all aspects, but following the same trend. But in case of ALICE, the transmission conflicts are taken into consideration while designing the schedule. However due to the autonomous nature of ALICE, a node doesn’t have the knowledge of neighbouring nodes, resulting in the sub-optimal schedule. Besides, ALICE does a mapping of cells to be used by links for transmission. Attaining an optimal mapping between links and the cells in slotframe is a NP hard problem. Hence, the schedule becomes highly inefficient for large traffics and large network sizes causing larger latency as can be observed from Fig. 13. In Fig. 14, we present the radio duty cycle performance of CBPF in comparison with MSF and ALICE. Since the proposed CBPF gives the optimal frequency of transmissions, the nodes in the network keep their radio on for the optimal number of slots and hence the duty cycle is kept low in comparison to MSF and ALICE, which are sub-optimal schedules. Hence, optimal transmission scheme ensures energy efficient communication.

VIII. Conclusion

The future of industrial communication is relied on wireless networks and Internet of Things. As the number of devices connected to the network is increasing rapidly and they generate sporadic traffic, reservation based scheduling is highly inefficient. Hence we propose a contention based transmission scheme for the TSCH network such that the weighted proportional throughput of the network is maximised. Also the proposed transmission scheme limits the number of collisions experienced by a packet, the energy consumption of the nodes during communication can be reduced enormously. We have implemented our proposed CBPF transmission scheme on a real time test bed IoT-Lab and have compared the performance of our algorithm with the existing schemes namely ALICE and MSF. The performance results showed that the proposed transmission scheme achieves the theoretical maximum throughput even when the network size is large.

Appendix A

Proof of the Convexity of Weighted Sum Throughput Function

Lemma 1: The weighted sum throughput function $F = \sum_{(n,m) \in S} w_{nm} \log(\mu_{nm}(\tau))$ is a concave function with respect to $\tau$.

Proof: Let the links in the TSCH network are indexed with $i, i = 1, 2, \ldots, S$ rather than with the source and destination nodes for simplicity. Let us denote the log rate function $\log(\mu_i(\tau))$ by $R_i(\tau)$ for link $i$. Now, the weighted sum throughput function $F$ is given in terms of $R_i(\tau)$ as follows.

$$F = \sum_{i=1}^{S} \mu_i R_i(\tau) \quad (41)$$
where from (6), \( R_i(\tau) \) can be written as

\[
R_i(\tau) = \log \left( \tau_i \prod_{j \in I_i^p} \left( 1 - \tau_j \right) \prod_{k \in \mathcal{I}_k \cap \mathcal{I}_S^p} \left( 1 - \frac{\tau_k}{M} \right) \right)
\]

\[
= \log(\tau_i) + \sum_{j \in I_i^p} \log(1 - \tau_j) + \sum_{k \in \mathcal{I}_k \cap \mathcal{I}_S^p} \log(1 - \frac{\tau_k}{M})
\]

(43)

Denote the Hessian of \( R_i(\tau) \) as

\[
\nabla^2 (R_i(\tau)) = \begin{bmatrix} a_{x,y}^{(i)} \end{bmatrix}, \quad 1 \leq x, y \leq S
\]

(44)

where \( a_{x,y}^{(i)} \) denotes the element of \( \nabla^2 (R_i(\tau)) \) at the \( x \)th row and \( y \)th column and can be expressed as

\[
a_{x,y}^{(i)} = \frac{\partial^2 (R_i(\tau))}{\partial \tau_x \partial \tau_y}
\]

(45)

From \( 42 \) and \( 44 \), the elements of hessian matrix \( \nabla^2 (R_i(\tau)) \) are determined as,

\[
a_{x,y}^{(i)} = \begin{cases} \frac{-1}{\tau_{xy}} & x = y = i \\ \frac{-1}{(1-\tau_{xy})} & x = y = j, j \in I_i^p \\ \frac{1}{\tau_{xy}} & x = y = j, j \in I_i^p \setminus I_i^p \\ 0 & \text{otherwise} \end{cases}
\]

(46)

From \( 45 \), it is clear that all entries of the Hessian matrix are non positive. Therefore the Hessian of log rate function of a link is negative semidefinite matrix, \( \nabla^2 (R_i(\tau)) \leq 0 \). Hence \( R_i(\tau) \) is a concave function of \( \tau \).

The function \( F \), as can be seen from \( 44 \), is the non negative weighted sum of \( R_i(\tau) \). Following the well established fact that non negative weighted sum of concave functions is a concave function \( [25] \), it can be seen that \( F \) is a concave function of \( \tau \).

REFERENCES

[1] M. Hashemi, W. Si, M. Laienfeld, D. Starobinski, and A. Trachtenberg, “Intra-car multihop wireless sensor networking: a case study,” IEEE Communications Magazine, vol. 52, no. 12, pp. 183–191, 2014.

[2] A. Elsts, S. Kim, H.-S. Kim, and C. Kim, “An empirical survey of autonomous scheduling methods for tsch,” IEEE Access, vol. 8, pp. 67147–67165, 2020.

[3] D. T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, R. Struik, J. Vasseur, R. Alexander et al., “Rpl: Ip6 routing protocol for low-power and lossy networks. rfc 6550,” 2012.

[4] Y. Ha and S. H. Chung, “Enhanced 6p transaction methods for industrial 6tisch wireless networks,” IEEE Access, vol. 8, pp. 174115–174131, 2020.

[5] S. Kharb and A. Singhroha, “A survey on network formation and scheduling algorithms for time slotted channel hopping in industrial networks,” Journal of Network and Computer Applications, vol. 126, pp. 59–87, 2019.

[6] M. O. Ojo, S. Giordano, D. Adam, and M. Pagano, “Throughput maximizing and fair scheduling algorithms in industrial internet of things networks,” IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3400–3410, 2019.

[7] C. Adjih, E. Baccell, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandeae et al., “Fit iot-lab: A large scale open experimental iot testbed,” in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, 2015, pp. 459–464.
[35] Yang Yang and T. P. Yum, “Delay distributions of slotted aloha and csma,” *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1846–1857, 2003.