

SoftFarmNet: Reconfigurable Wi-Fi HaLow Networks for Precision Agriculture

¹Nurzaman Ahmed, ²Flavio Esposito, ³Okwudilichukwu Okafor, ⁴Nadia Shakoor

^{1,4}Donald Danforth Plant Science Center

^{2,3}Department of Computer Science, Saint Louis University
St. Louis, MO, USA

Email: {¹nahmed, ⁴nshakoor}@danforthcenter.org, {²flavio.esposito, ³okwudilichukwu.okafor}@slu.edu}

Abstract—Networks deployed for Internet of Agricultural Things (IoAT) applications are often deployed in remote areas with limited coverage and a lack of standardization across sensing devices, posing challenges to reliable connectivity and resilient data exchange. The IEEE 802.11ah standard, commonly known as Wi-Fi HaLow, offers the potential for wide coverage and support for a large number of IoAT devices. However, such protocol still faces efficiency suboptimalities in channel utilization, particularly when handling heterogeneous IoT applications with diverse Quality of Service (QoS) requirements. To address these challenges, in this paper we propose *SoftFarmNet*, a reconfigurable IEEE 802.11ah (Wi-Fi HaLow) network management architecture, specifically designed for remote monitoring and control of agricultural-based IoAT. *SoftFarmNet* leverages a network traffic prediction-based slot scheduling and station grouping scheme to enhance channel utilization and support different QoS requirements of IoAT applications. The proposed architecture integrates Software-defined Networking (SDN) at the edge, enabling configuration of low-level Wi-Fi HaLow parameters and the management of multiple network slices to dynamically meet application requirements. Our performance evaluation demonstrates substantial improvements in throughput, delay, and energy consumption, effectively optimizing channel usage, with respect to state-of-the-art solutions.

Keywords—Internet of Things, Internet of Agricultural Things, IEEE 802.11ah, Software-Defined Networking, Quality of Service

I. INTRODUCTION

Precision agriculture is a modern farming approach that leverages technology to optimize crop yields, reduce waste, and increase profitability [1]. One of the most promising technological innovations in precision agriculture is the combination of the Internet of Things (IoT) and Unmanned Aerial Vehicles (UAVs) [2]. Integrating IoT sensors in the soil and on crops enables data collection and analysis for informed decision-making on planting, fertilization, and pest management. UAVs provide a bird's eye view, identifying crop issues and enabling swift corrective action [3]. However, challenges like network connectivity, data management, and QoS for different applications need addressing. Existing solutions such as LoRA, NB-IoT, and SigFox offer long-range communication but suffer from low data rates, limiting scalability [4], [5]. Wi-Fi, BLE, and 6LoWPAN offer higher data rates but limited coverage [6]. On the other hand, 4G/5G cellular networks have limited coverage, higher energy consumption, and cost. Existing solutions for scalable and remote networking solutions

in precision agriculture can be broadly categorized into two approaches. The first approach involves combining multiple existing communication technologies, such as Wi-Fi, 6LoWPAN, and LoRa [6], [7], creating complex and challenging-to-manage networks. The second approach focuses on utilizing Wi-Fi HaLow [8], [9], [10], offering a more streamlined and efficient solution.

The IEEE 802.11ah standard, also known as Wi-Fi HaLow, has great potential for covering vast areas and accommodating numerous devices [11]. With sub-1GHz channels, low power consumption, and Modulation and Coding Schemes (MCSs), it achieves up to 78Mbps data rates over a 1km range in a single hop [12]. This bandwidth is crucial for processing large amounts of field-collected imagery, such as (hyperspectral) cameras [13]. Wi-Fi HaLow's channel access and Restricted Access Window (RAW) features reduce contention and facilitate station grouping [11]. It also supports multi-hop or relays, expanding coverage with up to two hops, each covering over 1km. Thus, 802.11ah is a promising candidate for supporting numerous sensing and actuation devices in precision agriculture and other IoT applications.

Future AIoT networks deploy a wide range of heterogeneous IoT applications with diverse QoS requirements. For example, scenarios like deploying cameras in the field or utilizing drone-borne hyperspectral data for surveillance and phenotyping operations [14], coordinating fleets of robots for efficient fruit harvesting [15], enabling self-driving tractors to collaborate with UAVs [16], and even employing small bee-like drones for pollination assistance [17], all underscore the critical need for dynamic environments and real-time decision-making. Despite various enhancements in network management [18], slot scheduling schemes [19], [20] and node grouping schemes [21], [9] proposed for Wi-Fi HaLow, they do not suffice to meet the dynamic requirements and real-time decision-making demands of these diverse AIoT applications.

In this paper, we propose *SoftFarmNet*, a Wi-Fi HaLow-based network management architecture designed for connecting a large number of IoT-based agricultural devices with various applications such as farm monitoring, irrigation automation, and seasonal harvesting. An SDN controller at the edge collaborates with the edge and cloud to analyze application behavior using historical data and network traffic, enabling dynamic reconfiguration of the data plane's Wi-Fi

HaLow access points (APs). This dynamic reconfiguration optimizes channel allocation and ensures the required QoS in terms of throughput, delay, and energy consumption.

The key *contributions* of this paper are as follows:

- **Architectural contribution:** We introduce a scalable network management architecture based on Wi-Fi HaLow for smart precision agriculture. Leveraging SDN at the edge to configure low-level Wi-Fi HaLow parameters and create multiple network slices, enabling dynamic adaptation to meet varying application requirements.
- **Algorithmic contribution:** We present a network traffic and pattern prediction-based slot scheduling and station grouping scheme to enhance channel utilization and support QoS for different IoAT applications.
- **Implementation and evaluation:** We designed and implemented a prototype to generate IoAT network traffic. To predict the timing of future transmissions and facilitate slot scheduling, we employed suitable prediction models. The proposed network was implemented using the open-source SDN controller, Ryu [22], and for large-scale analysis, we used NS-3 simulator [23].

The rest of the paper is organized as follows: Section II discusses related works, Section III presents SoftFarmNet, our precision agriculture architecture. In Section IV, we evaluate its performance, and finally, Section V concludes the paper.

II. RELATED WORK

Network Management Architectures for IoT Applications. The related work on network management architectures for **precision agriculture** can be categorized into non-scalable network architectures, IEEE 802.11ah-based architectures, and SDN-based architectures.

Most existing network architectures for precision agriculture rely on wireless sensor networks and employ various communication technologies such as ZigBee, Bluetooth, Cellular, Wi-Fi, 5G, and LoRa. For instance, Gsangaya et al. [24] utilized Wi-Fi-based ESP8266 devices for field node-to-AP communication to facilitate data acquisition. Ahmed et al. [6] combined 6LoWPAN and long-distance Wi-Fi networks to connect precision agriculture devices in rural regions. While those are all sound solutions, managing a network with modern and future farm cyber-physical systems remains a challenge. For example, while LoRa is capable of connecting sensor devices over long distances in precision agriculture [7], it is limited in terms of the high data rate and scalability requirements. Our architecture was designed to cope with those limitations. Additionally, studies have explored the suitability of 5G cellular networks for rural agriculture, highlighting features such as enhanced mobile broadband (eMBB), ultrahigh reliability and low latency (uRLLC), and massive machine type communications (mMTC) [25]. FarmBeats [26] presents an end-to-end IoT platform for agriculture that enables seamless data collection from various sensors, cameras and drones. IBM's Watson Decision Platform for Agriculture [27], and ThingsBoard's IoT platform [28]. Our architecture not only

leverages these innovations but also enhances network coverage and reduces implementation costs, making it a promising solution for precision agriculture applications.

While these existing networks are designed for specific communication needs, IEEE 802.11ah holds the potential for connecting a large number of IoAT [12]. Alam et al. [8] demonstrates the potential of IEEE 802.11ah for long-range connectivity and supporting various IoT applications, including precision agriculture.

Prior use of IEEE 802.11ah and Radio Channel Management in IoT. To improve channel utilization and capacity, Chang et al. [21] proposed a station grouping scheme for load balancing among RAW groups. Tian et al. [9] developed the Traffic-Aware RAW Optimization Algorithm (TAROA) to predict inter-packet times of stations, while Georgiev et al. [10] analyses selfish behavior of 802.11ah stations to improve fairness. In [29], a genetic algorithm (GA)-based approach is proposed for station grouping. By optimizing the GA parameters, the algorithm achieves highly efficient results within a short timeframe. These studies enhance channel utilization but often overlook QoS and priority considerations for heterogeneous network traffic.

Software-Defined Networks for IoT. Incorporating SDN into IoT networks has been explored to address various challenges and optimize service delivery [30]. For example, Huang et al. [31] proposed an SDN-based vehicular network for precision agriculture to minimize performance degradation during controller connection loss. A slicing scheme on 802.11ah network has been proposed in [18]. It creates and manages logical slices per service (e.g., video or audio) based on the available RPS configurations. However, further works are needed for supporting the required QoS demands of various agricultural services while efficiently utilizing the available channel bandwidth.

Considering the requirements of precision agriculture, a network architecture that seamlessly connects all components and facilitates remote management and control based on QoS and application priorities is of utmost importance, and this paper aims to achieve that. While an SDN-enabled IEEE 802.11ah network holds promise in addressing connectivity challenges, further efforts are required to support the specific QoS requirements of different applications while optimally utilizing the available channels.

III. SOFTFARMNET ARCHITECTURE OVERVIEW

A. System Model and Assumptions

Consider Fig. 1, which illustrates the architecture of a Wi-Fi HaLow-based IoAT network. IoT devices are organized into different groups irrespective of their geographical locations. The stations are associated with their nearest Access Point (AP) or relay node to forward their traffic to a gateway. The gateway or AP utilizes their available resources to host an Edge SDN controller (ESDN). Each of these groups is allocated to a RAW frame, which has a specific RAW parameter Set (RPS). Each group is associated with a slice, which is a set of states composed of virtual network states and other group

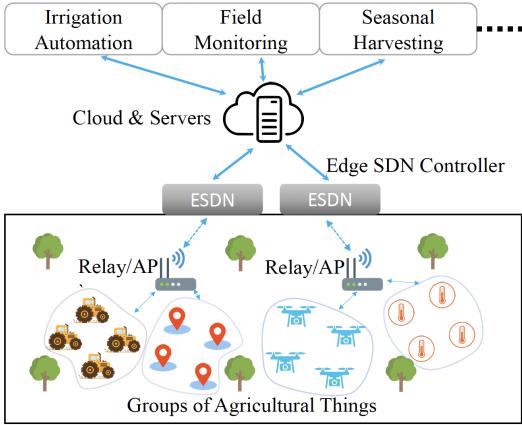


Fig. 1: Wi-Fi HaLow-based IoAT network architecture

states. In this model, we consider a total of N_{sta} IoT devices, including sensors and actuators, distributed among K_{gr} groups. Each group consists of $N_{gs} = N_{sta}/K_{gr}$ stations, although the number of stations in each group can dynamically change for load balancing purposes. We assume that the network utilizes RAW $R = R_1, R_2, \dots, R_{K_{gr}}$ for channel access, configured with a specific RPS. Additionally, we assume that multiple MCSs are available and we denoted them with, $M = M_1, M_2, M_3, \dots, M_m$, allowing the selection of an appropriate MCS for a relay or an access point, based on the required bandwidth for the current demand. In our model, an SDN controller oversees the network infrastructure, enabling flexible dynamic configuration of parameters related to APs, RPS, MCS, and grouping.

We categorize IoAT application's network traffic into three main classes: (i) control loop, (ii) periodic, and (iii) on-demand. The **control-loop** category includes real-time control and automation applications with high sensitivity to delays and direct impact on automation, such as machine automation and irrigation control. Traffic flows in this category are assigned *Priority 1*. The **periodic** category involves continuous or periodic data collection and transmission for monitoring and triggering actions, like collecting weather data every hour, capturing crop growth measurements daily, or monitoring livestock conditions at fixed time intervals. Traffic flows in this category are assigned *Priority 2*. The **on-demand** category includes flexible and less time-sensitive data collection applications, such as collecting hyperspectral images from the crops during early sessions. Traffic flows in this category are assigned *Priority 3*.

The SDN controller creates multiple QoS configurations for each type, enabling the creation of slices over the 802.11ah network infrastructure. The edge gateway possesses sufficient processing capabilities to perform edge computing operations such as actuation based on threshold values and running the SDN controller. The cloud performs traffic and data value pattern predictions based on historical agricultural data and network traffic. The cloud forwards the outputs, such as future expected traffic flows, to the edge computing module and SDN controller for data-level and network-level decision-making,

respectively. By utilizing this system model, the proposed solution achieves traffic differentiation and resource allocation.

B. Traffic Differentiation

In this section, we focus on modeling different classes of IoAT network traffic and employing prediction models to calculate expected future transmission.

Periodic Data Collection Type. Periodic traffic exhibits predictable patterns, making it relatively easier to anticipate and plan for. We determine the better periodicity of such traffic by utilizing Autocorrelation Function (ACF). Let's assume we have a time series of uplink traffic data denoted as X_t , where t represents the time index. The $ACF(k)$ at lag k can be calculated as follows:

$$ACF(k) = \frac{\sum_{t=k+1}^N (X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^N (X_t - \bar{X})^2} \quad (1)$$

where \bar{X} and N are the mean and total number of observations of the time series data, respectively. The reciprocal of the lag with highest ACF gives the periodicity of traffic [32]. The cloud stores data with a size of 1000, which is then processed to calculate the autocorrelation function (ACF). The periodicity of network traffic can be seen in Fig. 2a. The corresponding ACF and lag values are illustrated in Fig. 2b.

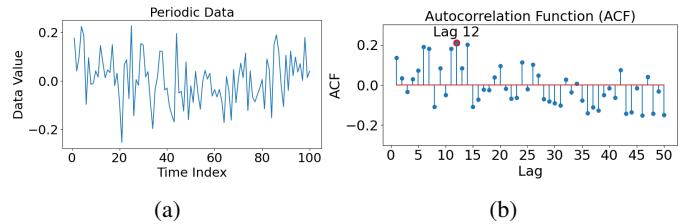


Fig. 2: Calculating periodicity for historical soil sensor data: (a) Sine-wave of Periodic Data , and (b) ACF and Significant Peaks in ACF

Control-Loop Type. Control-loop communication and on-demand data collection are less predictable, as they are triggered by specific events. In the case of control-loop communication in agricultural fields, it is possible to predict the occurrence of downlink frames based on the patterns observed in the uplink periodic data [33]. For instance, a decrease in moisture levels could indicate the need for action, leading to the generation of downlink traffic. We consider employing Autoregressive Integrated Moving Average (ARIMA). Let us denote with X_t and Z_t the time series data and error term at time t , respectively; with B we denote the backshift operator, representing the lag operator, with ϕ_1, \dots, ϕ_p the autoregressive (AR) coefficients, and with $\theta_1, \dots, \theta_q$ the moving average coefficients. Then, an $ARIMA(p, d, q)$ model can be represented as follows:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d X_t = (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) Z_t \quad (2)$$

where, p is the order of the AutoRegressive (AR) component, d is the degree of difference required to achieve stationarity, q the order of the moving average (MA) component. With the model of Equation 2, we can forecast future uplink traffic values using. Based on the forecasted uplink traffic, we can identify when downlink traffic is expected. We generated a dataset of 1000 data points representing control loop communication. We then applied an ARIMA-based prediction model to estimate the timing of a control loop pair, which includes the time for the uplink and downlink frames. Fig. 3a shows the forecasted uplink traffic and visualizes the predicted downlink traffic. Threshold markers are set at a value of 20 to indicate the threshold level.

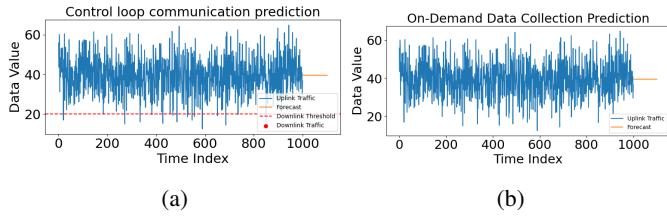


Fig. 3: ARIMA-based prediction of control-loop time based on smart irrigation data (b) SARIMA-based prediction of on-demand traffic timing based on smart irrigation data

On-Demand Data Collection Type. The on-demand data collection often takes place after specific seasons of crops, typically involving a larger amount of uplink traffic. In this case, we use Seasonal ARIMA (SARIMA) models to predict appropriate seasonal periods. Along with non-seasonal ARIMA parameters p , d , and q , let P , D , and Q , represent the seasonal orders of the autoregressive, differencing, and moving average components, respectively. We can fit the SARIMA(p, d, q)(P, D, Q) $_s$ model to the uplink traffic data, where s represents the seasonal period [34]. By utilizing this model, we can account for both the seasonal and non-seasonal patterns in the uplink traffic data. This improved model allows for more accurate predictions during the specific seasons of crops when on-demand data collection is expected. Fig. 3b illustrate the prediction of on-demand data over time. Due to

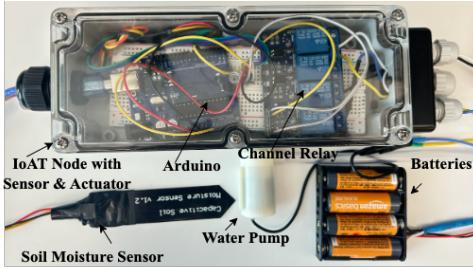


Fig. 4: Arduino-based sensor and actuator node for collecting control loop and on-demand application's traffic

the unavailability of publicly available datasets to understand control loop and on-demand traffic in agricultural scenarios, we have developed an IoAT node with sensing and actuation

capabilities, as depicted in Fig. 4. This module is designed to capture the service patterns of agricultural applications, disregarding timing variations caused by communication and networking constraints. The cloud/edge utilizes a southbound API to send the trained model associated with predicted timing for all three types of traffic, enabling efficient scheduling and actuation by the SDN controller.

C. The Resource Allocation

The SDN controller analyzes the available resources, specifically dedicated slots for communication, to enable QoS for the identified priorities.

1) RAW Parameter Set: The RAW is a crucial concept in IEEE 802.11ah networks that defines a dedicated time interval within a superframe to allow a particular group of stations for contention. The RPS in the IEEE 802.11ah standard contains essential information about the configuration of one or more RAWs, including the associated stations in each RAW and the duration of each RAW. A station determines its assigned RAW using the following formula:

$$x = (i + \text{offset}) \bmod S_{\text{RAW}} \quad (3)$$

Here, x represents the slot number within a RAW frame of size S_{RAW} , the offset value is utilized to enhance fairness among the stations within a RAW, and i denotes the position index or Association Identifier (AID) of the station. If the station has already been paged, it uses the AID; otherwise, it uses the position index. The station can access the RAW only if the RAW is restricted to stations with AID bits set to 1 in the TIM (Traffic Indication Map) element [35]. The duration of each slot (T_x) is calculated based on the slot duration count (S_c) specified in the RPS as follows:

$$T_x = 500\mu\text{s} + S_c \times 500\mu\text{s} \quad (4)$$

Here, S_c depends on the value of k ($S_c = 2^k - 1$), which represents the number of bits in a sub-field. The proposed scheme sets the slot size by determining an appropriate value of k . The information collected from the above section, we

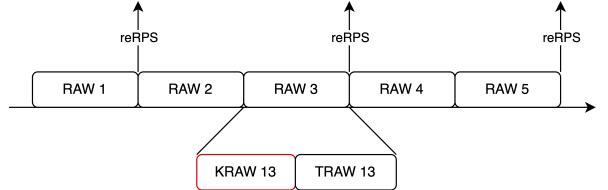


Fig. 5: Structure of RAW with logically separated TRAW and KRAW

use **KRAW** contains a dedicated set of slots for known traffic. Otherwise, traffic which are not known to the system, we use **TRAW** (traditional RAW) they use traditional DCF based channel access for uplink and downlink transmission (refer Fig. 5). We use an updated RPS (reRPS) to support dynamic

RPS. In of traffic's priorities are known, the time required for transmitting one frame can be calculated as:

$$T_{KDATA} = T_{SIFS} + T_{FH} + T_{DATA} + T_{ACK} + 2T_P \quad (5)$$

Here, $T_{FH} = T_{PHY} + T_{MAC}$ represents the frame header duration, including the PHY layer header T_{PHY} and the MAC layer header T_{MAC} . Additionally, T_{DATA} , T_{ACK} , T_{SIFS} , and T_P indicate the time required to transmit a data frame, acknowledgement, short inter-frame space, polling, and propagation, respectively. Otherwise, in such a case, normal traffic, the time required for transmitting one frame can be calculated as:

$$T_{DATA} = T_{FH} + T_{DATA} + T_{ACK} + T_{SIFS} + T_{POL} + 3T_P + T_{BO} \quad (6)$$

where, T_{POL} , and T_{BO} are the size of a power-save poll message and back-off. The control loop time (C_T) is the time difference between actuation and sensing. If T_{AD} is the propagation time from AP to Controller, and T_{CTL} is the controller processing time, and T_{WAIT} is the frame wait time at AP. Then, the control loop time, $C_T = 2(T_{KDATA} + T_{AD}) + T_{WAIT} + T_{CTL}$. Therefore to complete a control loop communication slots pair $slot_1$ and $slot_2$ are scheduled at time t_1 and $t_2 (= t_1 + C_T - T_{KDATA})$. The control loop achieves its fastest execution when the controller processing time is smaller than the beacon interval, and there is no contention during the transmission of frames by the sensor and actuator nodes [19]. While removing the T_{WAIT} , depends on the beacon interval, the shortest control loop time can be calculated as:

$$C_T = \frac{2 \times (T_{DATA} + T_{SIFS} + T_{ACK})}{T_{beacon} - T_{KDATA}} \quad (7)$$

where T_{beacon} is the beacon interval. To increase the number of C_T and to reduce T_{WAIT} time, the proposed scheme schedule a priority 1 traffic in the immediate slots after the beacon. In case of periodic traffic, a slot of size T_{KDATA} is reserved at $t_3 (= P_T - T_{KDATA})$ time. Moreover, in case of on-demand traffic, if the total size of the request segment is S_{req} , the number of dedicated slot should be:

$$S_T = \frac{S_{req}}{T_{KDATA}} \quad (8)$$

While supporting priority to the higher classes, it is important to note that time slots may not be consecutive.

2) Channel utilization and dynamic stations grouping: Considering the dynamic traffic load over time there maybe some of the RAW group get congested. Moreover, if there are large number of priority traffic in a single group, other traffic flows will face never transmission issues. Therefore, load balancing aware dynamic grouping mechanism is proposed. Lets assume there are N_{frames} number of frames in a RAW group and duration of a RAW is T_{RAW} . For known traffic (allocated in dedicated slots of K_{RAW}), we can calculate the load in the RAW group as follows:

$$\text{LoadK} = N_{frames} \times T_{KDATA} \quad (9)$$

where T_{KDATA} is the time required to transmit one frame for known traffic. Similarly, for normal traffic (utilizing traditional

RAW slots in T_{RAW}), we can calculate the load in the RAW group as: $\text{LoadT} = N_{frames} \times T_{TDATA}$, where T_{TDATA} is the time required to transmit one frame for normal traffic. To calculate the channel utilization, we need the total duration of the RAW group. Assuming all the RAW slots are fully utilized, the channel utilization can be calculated as:

$$\text{Channel Utilization, } U = \frac{\text{LoadK} + \text{LoadT}}{K \times T_{RAW}} \quad (10)$$

where K is the number of RAWs. We can form the integer programming formulation that optimizes the channel utilization while considering station assignments and the regrouping of stations between groups to maximize the incremental gain of channel utilization: *Objective:*

$$\text{Maximize } U = \frac{\text{LoadK} + \text{LoadT}}{K \times T_{RAW}} \quad (11)$$

Subject to:

$$\sum_{j=1}^K X_{i,j} = 1, \quad \forall i \in S \quad (12)$$

$$T_j \leq T_{RAW}, \quad \forall j = 1, \dots, K \quad (13)$$

$$X_{i,j} \in \{0, 1\}, \quad \forall i \in S, \forall j = 1, \dots, K \quad (14)$$

$$H_j = \sum_{i \in S} X_{i,j} \cdot D_i, \quad \forall j = 1, \dots, K \quad (15)$$

$$H_j \leq H_k, \quad \forall j, k \text{ where } j < k \quad (16)$$

To maximize the utility (U) based on the given equation, which considers the load of both K and T components over the total available RAW time (T_{RAW}). Equation 12 states that each station i must be assigned to exactly one group. The sum of $X_{i,j}$ over all groups j for a given station i should be equal to 1, indicating that the station is assigned to one group only. Equation 13 restricts the duration T_j of each group j to be less than or equal to the duration of the RAW group T_{RAW} . It ensures that the duration of each group does not exceed the overall duration of the RAW group. Equation 14 specifies that the decision variable $X_{i,j}$ takes binary values of 0 or 1, indicating whether station i is assigned (1) or not assigned (0) to group j . It enforces the assignment of stations to groups to be mutually exclusive. The additional constraint $H_j \leq H_k$ ensures that the load of RAW group j is less than or equal to the load of RAW group k for all pairs (j, k) where j represents a higher priority RAW group than k . This constraint allows the dynamic allocation of stations from higher load RAW groups to lower load RAW groups, prioritizing the stations with higher priority.

3) Northbound API: The SDN controller collects priority, future RPS timing (predicted), MCS, and grouping information via the northbound API. Using these data, a Virtual Network Slicing Broker (VNSB) creates dynamic slices and transmits the updated parameters to the AP via the southbound API. This allows the AP to implement the required changes in its MAC layer, ensuring efficient resource allocation and improved QoS for different traffic flows.

4) *Data Plane*: The data plane in the AP is essential for efficient resource deployment and network slice configuration by the SDN controller. It facilitates optimal resource allocation, while also providing feedback on the current load conditions. Algorithm 1 present the slot scheduling process

Algorithm 1: Priority-based Resource scheduling

Data: Priority 1, Priority 2, and Priority 3 traffic
Result: Resource 1, Resource 2, Resource 3

```

1  $\{slot_1, slot_2, \dots\} \in KRAW$  ;
2  $\{SLOT_1, SLOT_2, \dots\} \in TRAW$  ;
3  $t \leftarrow$  time ;
4 for Incoming traffic do
5   if Priority == 1 then
6     Resource 1: Allocate  $slot_1$  for uplink, and
       $slot_2$  for downlink at time  $t$  and  $t + C_T$ 
      respectively
7   else if Priority == 2 then
8     Resource 2: Allocate  $slot_3$  at time
       $t \wedge t \notin \{slot_1, slot_2\}$  ;
9   else if Priority == 2 then
10    Resource 3: Assign
       $\{slot_4, \dots, slot_{S_T}\} \notin \{slot_1, slot_2, slot_3\}$  ;
11   else
12     Assign  $\{SLOT_1, SLOT_2, \dots\}$  for normal
      contention.
```

for allocating resources based on the priority of incoming traffic. It takes into account different levels of priority traffic, namely priority 1, priority 2, and priority 3. The algorithm aims to allocate appropriate resources (i.e., slot time $slot$) for each priority level based on their specific requirements. The algorithm operates in a time-based manner, with incoming traffic being processed at each time interval. For priority 1 traffic, which represents control loop traffic, the algorithm allocates two slots: one for uplink and the other for downlink communication. These slots are assigned at specific time duration. For priority 2 traffic (line #5-6), which represents periodic traffic, the algorithm allocates a single slot based on the closest periodicity of the traffic. This slot is assigned at a time that is not conflicting with the slots allocated for priority 1 traffic. For priority 3 traffic, which represents on-demand traffic, the algorithm assigns a set of slots to fulfill the immediate requirements of the application. These slots are selected from the available slots that have not been allocated for priority 1 or priority 2 traffic. In case the incoming traffic does not match any of the defined priorities, the algorithm falls back to normal contention, where resources are assigned based on standard contention mechanisms.

IV. PERFORMANCE EVALUATION

The proposed network architecture has been evaluated using a combination of emulation, simulation, and a real IoT setup. To develop accurate prediction models, real IoT datasets

TABLE I: Parameters used in Simulation and Analysis

Parameters	Value
Bandwidth	2MHz (MCS0, MCS1)
Basic Data rate (δ)	650Kbps, 1300Kbps
Payload size (L)/Traffic type	100 Bytes/UDP
Traffic rate	≈ 2 Kbps* (MCS0)
CWmin/CWmax, σ/σ_{max}	15/1023, 0.5/0.7 Sec.
Backoff slot time	52 μ s
SIFS time /DIFS time	160 /SIFS+2* slot time μ s
Distribution/Path loss model	Random/Outdoor Macro [36]
Symbol duration (T_{sym})/bits (β)	40 μ s/26 bits
Coding rate BPSK-MCS0 (γ)	0.5
Header PHY (T_{PHY})/MAC (m_h)	6 * T_{sym} (μ s)/14Bytes
Queue size/Group/RAW size	100/2-10 /15
No. of stations (Max.)	1000
$p_{tx}/p_{rx}/p_{id}/p_{st}$	255 /135/135 /1.5mW [37]
Simulation area/time	1000 \times 1000 m ² / 5 Min.

are considered for the service differentiation methods. The OpenVswitch, integrated with the NS-3 802.11ah module [23], is connected with the Ryu SDN controller [22] to simulate large-scale IoT networks. The default performance evaluation parameters are provided in Table I.

A. Performance Metrics

We analyze QoS performance using key metrics: throughput, delay, and energy consumption. Our study compares *SoftFarmNet* with traditional IEEE 802.11ah (*HaLow* [11]) and a slicing-based 802.11ah scheme, *CoHaLow* [18].

1) *Throughput*: In case of data transmission in traditional RAW, we can use the two-dimensional discrete Markov Chain Model as proposed by Bianchi et al. [38]. A station can transmit when backoff counter is zero and the probability of moving to the next state depends only on the event occurred in the previous state. It is possible to calculate the probability of at least one transmission (P_{TX}), and the probability of successful Tx in a slot (P_{SUC}) [38]. With average payload size $E[Payload]$, the saturation throughput (T_{thr}) using the traditional protocol can be calculated as:

$$T_{thr} = \frac{P_{TX} P_{SUC} E[Payload]}{(1 - P_{TX})\xi + P_{TX} P_{SUC} T_{SUC} + P_{COL} T_{COL}} \quad (17)$$

where ξ is the average duration of a slot, and $P_{COL} = (1 - P_{SUC})P_{TX}$ is the collision probability. T_{SUC} and T_{COL} are the busy times for successful Tx and when a collision occurs respectively. For 802.11ah, these values can be calculated as below:

$$T_{SUC} = T_{DATA}$$

$$T_{COL} = T_{PH} + T_{DATA} + T_{DIFS} + T_p + T_{Timeout}$$

where $T_{DATA}, T_{SIFS}, T_p, T_{ACK}, T_{Timeout}$ and T_{DIFS} are the Data, SIFS, Propagation, ACK, ACK-Timeout, and DIFS duration respectively. For long durable periodic station, we can neglect the initial contention time. The duration of data frame, T_{DATA} and control frame, T_{ACK} used in IEEE 802.11ah can be calculated as proposed in [37]. Then, saturation throughput

in the proposed scheme can be calculated as:

$$T_{prop} = \frac{E[Payload]}{T_{KDATA}} \quad (18)$$

For successfully transmitting a frame in the traditional scheme, T_{SUC} additionally includes T_{POL} , T_{DIFS} , and T_{SIFS} . From Eq. 17, and 18, it is clear that the throughput of the proposed scheme will be higher.

2) *Energy Consumption*: In the context of 802.11ah, a transceiver can be in different modes such as receiving (T_{RX}), transmitting (T_{TX}), idle (T_{ID}), and sleeping (T_{SL}) within a DTIM (Delivery Traffic Indication Message) period [11]. The total energy consumption by a station within a DTIM can be calculated by multiplying the duration of each operation with their respective power consumption [39]:

$$E_{tr} = T_{RX} p_{rx} + T_{TX} p_{tx} + T_{IDP} p_{id} + T_{SLP} p_{sl}$$

where p_{tx} , p_{rx} , p_{id} , and p_{sl} are the power consumption required in the transmitting, receiving, idle, and sleep modes, respectively. However, in the proposed scheme, the frequency of updates depends on the controller messages. If there are no changes during one or two DTIM periods, a station might stay in sleep mode for a longer duration, resulting in energy savings.

3) *Delay*: In IoT, one of the major concerns is the delay caused by channel access, as a large number of devices compete for the same channel. When multiple stations choose the same backoff slot, a collision is likely to occur. Assuming a negligible frame drop probability, we can calculate the average frame delay (D_{tr}) for traditional WiFi HaLow as follows [40]:

$$D_{tr} = T_{BW} \times \xi \quad (19)$$

where T_{BW} represents the time duration that a station needs to wait before successfully transmitting a frame after encountering a series of empty slots [41]. However, the proposed approach uses a dedicated frame time T_{KDATA} to successfully complete a transmission.

B. Results on Channel Utilization

Fig. 6a shows the channel utilization of three different schemes: SoftFarmNet, CoHaLow, and HaLow, as the number of stations increases in the network. The graph demonstrates the impact of station density on channel utilization and allows for a comparison of the performance of each scheme. With the default configuration, a single station generates an average data rate of nearly 2 Kbps, and to fully utilize a half-duplex link operating at MCS0 (2 MHz), approximately 160 stations are required. As the number of stations increases beyond this point, traditional schemes such as HaLow face challenges related to contention and collisions, resulting in a decrease in channel efficiency. However, the proposed scheme, SoftFarmNet, utilizes slot scheduling for dedicated transmission and dynamic grouping, enabling optimization of slot allocation and improved channel utilization. We further investigated the effect of the number of groups on channel utilization (refer to Fig. 6b). The results indicate that all

schemes perform better with 10-15 groups when there are 1000 stations. However, the proposed scheme consistently demonstrates the highest efficiency. The dynamic grouping capability of the proposed scheme has a lesser impact on channel utilization. It is important to note that the trends may vary with different numbers of stations. To critically assess the performance of the scheme with an increasing number of control loop communications among the 1000 stations, we conducted additional experiments (see Fig. 6c). In a single control loop, an uplink and downlink traffic pair is involved, consuming double the required resources. While the other schemes fail to maintain a stable channel utilization as the number of control loop communications increases, the proposed scheme shows nearly 50% higher utilization efficiency. We evaluated the performance of the proposed scheme with varying numbers of stations, demonstrating its stability and consistent channel utilization (refer to Fig. 6d). The prediction-based slot scheduling and dynamic grouping features contribute to the scheme's ability to maintain optimal channel utilization even with increasing numbers of stations.

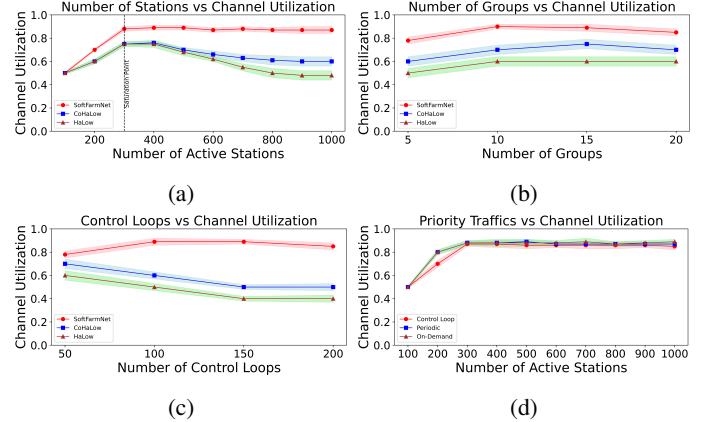


Fig. 6: Channel utilization with: (a) increasing active number of stations, (b) increasing number of groups, (c) increasing control loops, (d) different priorities

C. Results on Throughput Performance

The prediction-based slot allocation in the proposed scheme improves dedicated transmission by reducing collisions due to contention, thereby reducing transmission time (refer to Fig. 7a). However, CoHaLow and HaLow encounter collision issues during transmission using DCF-based channel access. Consequently, as the number of stations increases, the throughput decreases in the proposed scheme. Furthermore, when multiple MCSs are available (e.g., MCS0 (2 MHz) and MCS1 (2 MHz)), the proposed scheme dynamically switches to the MCS with a better data rate, resulting in improved throughput when required (see Fig. 7b). We also investigated how Packet delivery ratio (%) of different priorities behave under increasing load conditions, as shown in Fig. 7c. Due to priority handling, control loop communications exhibit stable throughput performance. However, a slight decrease

in throughput can be observed for priority 2 (Overdue) and priority 3 (On-demand) with increasing loads. Additionally, to enable a shorter time frame for dynamic configuration, the beacon interval plays a crucial role. In the proposed scheme, the configuration can be adapted immediately with a changing interval, whereas in the traditional HaLow scheme, all stations can only be updated during the DTIM periodic update. Therefore, a DTIM with a lower value yields better results in the traditional HaLow scheme (refer to Fig. 7d).

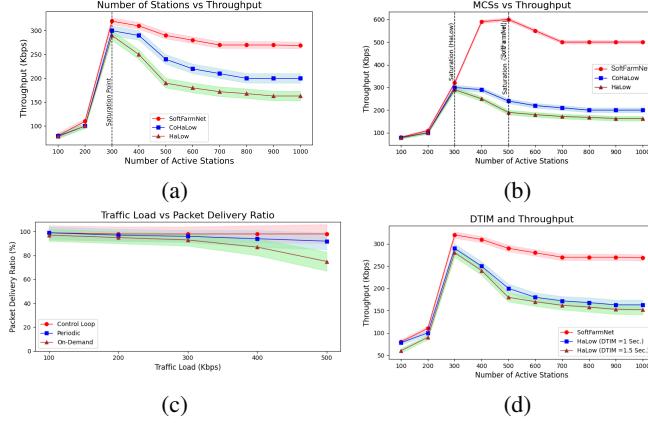


Fig. 7: Throughput: (a) increasing number of stations, (b) different MCSs, (c) increasing loads, (d) different DTIM sizes

D. Results on Delay

The proposed scheme effectively reduces channel access delay and wait time through throughput prediction-based slot scheduling. By utilizing dedicated transmission and ensuring transmission when slots are available, the scheme minimizes collisions. This is evident in Fig. 8a, where the delay is the lowest in the proposed scheme. Although CoHaLow reconfigures the RPS for service-aware scheduling, the lack of prediction of upcoming traffic results in failure to reduce delay in channel access. Furthermore, we examined the average time required for a frame to be successfully transmitted, as shown in Fig. 8b. Due to the highest priority, an immediate slot is allocated to the control loop, resulting in the lowest latency compared to other traffic flows. On-demand traffic is scheduled without strict deadlines, hence exhibiting the lowest delay. Similarly, when multiple MCSs are available, the proposed scheme dynamically utilizes them, making it easier to transmit a frame in the MCS with a higher data rate (refer to Fig. 8c). Finally, as depicted in Fig. 8d, with 1000 devices, the delay for all schemes is better in the case of 10-15 groups, where the proposed scheme demonstrates the lowest delay.

E. Results on Energy Consumption

The proposed scheme leverages traffic periodicity, predicts control loop communication, and forecasts on-demand communication to efficiently utilize available resources in a reduced time frame. Additionally, it optimizes energy consumption by extensively configuring the RPS to keep devices

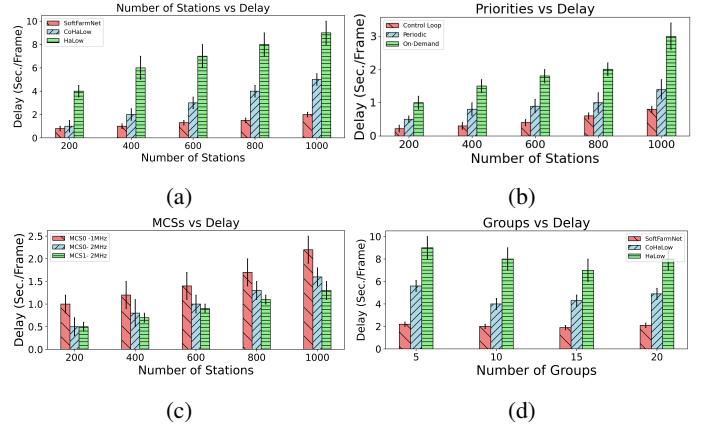


Fig. 8: Delay incurred: (a) increasing number of stations, (b) different priorities, (c) different MCSs, (d) increasing number of groups

in sleep mode for longer periods. As shown in Fig. 9a, the proposed scheme requires the lowest energy consumption to successfully transmit a frame, while the state-of-the-art solutions exhibit significantly higher energy consumption, possibly due to collisions and retransmissions. Moreover, the wait time (e.g., idle or Rx mode of the station) for high-priority applications is minimized in the proposed scheme, leading to lower energy consumption, as depicted in Fig. 9b.

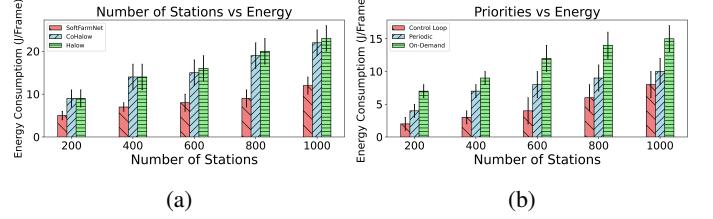


Fig. 9: Energy consumption: (a) increasing active number of stations, (b) different priorities

We plan to analyze controller overhead cost and algorithmic complexities in the future.

V. CONCLUSION

In this paper, we presented an SDN and edge computing enabled Wi-Fi HaLow network that effectively supports a large number of stations with diverse agricultural applications. By addressing challenges related to channel utilization, application prioritization, and adaptability, our proposed scheme offers significant improvements over existing state-of-the-art solutions for precision agriculture networks. Through extensive performance analysis, we have demonstrated enhanced throughput, reduced delay, and optimized power consumption, while effectively utilizing available channels and maintaining traffic flow priorities. Our future work includes the use of multiple prediction models and implementation of the network over a real testbed in an agricultural farm field.

REFERENCES

- [1] Y. Jararweh, S. Fatima, M. Jarrah, and S. AlZu'bi, "Smart and sustainable agriculture: Fundamentals, enabling technologies, and future directions," *Computers and Electrical Engineering*, vol. 110, p. 108799, 2023.
- [2] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.
- [3] V. Sagan, M. Maimaitijiang, P. Sidike, K. Ebblimit, K. T. Peterson, S. Hartling, F. Esposito, K. Khanal, M. Newcomb, D. Pauli, R. Ward, F. Fritschi, N. Shakoor, and T. Mockler, "Uav-based high resolution thermal imaging for vegetation monitoring, and plant phenotyping using ici 8640 p, flir vue pro r 640, and thermomap cameras," *Remote Sensing*, vol. 11, no. 3, 2019.
- [4] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A Survey on LoRa for Smart Agriculture: Current Trends and Future Perspectives," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3664–3679, 2022.
- [5] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot," in *2018 ieee international conference on pervasive computing and communications workshops (percom workshops)*, pp. 197–202, IEEE, 2018.
- [6] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4890–4899, 2018.
- [7] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A survey on lora for smart agriculture: Current trends and future perspectives," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3664–3679, 2023.
- [8] M. Alam, N. Ahmed, R. Matam, and F. A. Barbhuiya, "IEEE 802.11ah-Enabled Internet of Drone Architecture," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 174–178, 2022.
- [9] L. Tian, E. Khorov, S. Latré, and J. Famaey, "Real-time station grouping under dynamic traffic for ieee 802.11 ah," *Sensors*, vol. 17, no. 7, p. 1559, 2017.
- [10] Y. Georgiev, R. Verhoeven, and N. Meratnia, "Selfish behavior in ieee 802.11 ah networks: A detection algorithm and mitigation strategies," *Sensors*, vol. 22, no. 12, p. 4472, 2022.
- [11] "IEEE Approved Draft Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 2: Sub 1 GHz License Exempt Operation," *IEEE P802.11ah/D10.0, Sep 2016*, pp. 1–660, Jan 2016.
- [12] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.
- [13] M. Maimaitijiang, V. Sagan, P. Sidike, S. Hartling, F. Esposito, and F. B. Fritschi, "Soybean yield prediction from uav using multimodal data fusion and deep learning," *Remote Sensing of Environment*, vol. 237, p. 111599, 2020.
- [14] N. Virlet, K. Sabermanesh, P. Sadeghi-Tehran, and M. J. Hawkesford, "Field scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring," *Functional Plant Biology*, vol. 44, no. 1, pp. 143–153, 2016.
- [15] H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen, "Intelligent robots for fruit harvesting: Recent developments and future challenges," *Precision Agriculture*, vol. 23, no. 5, pp. 1856–1907, 2022.
- [16] V. Barrile, S. Simonetti, R. Citroni, A. Fotia, and G. Bilotta, "Experimenting agriculture 4.0 with sensors: A data fusion approach between remote sensing, uavs and self-driving tractors," *Sensors*, vol. 22, no. 20, p. 7910, 2022.
- [17] NPR, "Rise Of The Robot Bees: Tiny Drones Turned Into Artificial Pollinators." <https://www.npr.org/sections/thesalt/2017/03/03/517785082/rise-of-the-robot-bees-tiny-drones-turned-into-artificial-pollinators>. [Accessed on July 27, 2023].
- [18] P. P. Libório, C. T. Lam, B. Ng, D. L. Guidoni, M. Curado, and L. A. Villas, "Network slicing in ieee 802.11ah," in *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, pp. 1–9, 2019.
- [19] A. Seferagić, I. Moerman, E. De Poorter, and J. Hoebeke, "Evaluating the Suitability of IEEE 802.11ah for Low-Latency Time-Critical Control Loops," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7839–7848, 2019.
- [20] N. Ahmed and M. I. Hussain, "Periodic traffic scheduling for ieee 802.11 ah networks," *IEEE Communications Letters*, vol. 24, no. 7, pp. 1510–1513, 2020.
- [21] T.-C. Chang, C.-H. Lin, K. C.-J. Lin, and W.-T. Chen, "Load-balanced sensor grouping for ieee 802.11 ah networks," in *2015 IEEE global communications conference (GLOBECOM)*, pp. 1–6, IEEE, 2015.
- [22] "Framework community: Ryu SDN framework," *Online*. <http://osrg.github.io/ryu>, 2015.
- [23] "802.11ah ns-3." <https://github.com/imec-idlab/IEEE-802.11ah-ns-3>. Last accessed: 14th June, 2023.
- [24] K. R. Gsangaya, S. S. H. Hajjaj, M. T. H. Sultan, and L. S. Hua, "Portable, wireless, and effective internet of things-based sensors for precision agriculture," *International Journal of Environmental Science and Technology*, vol. 17, pp. 3901–3916, Apr. 2020.
- [25] W. Arrubla-Hoyos, A. Ojeda-Beltrán, A. Solano-Barliza, G. Rambauth-Ibarra, A. Barrios-Ulloa, D. Cama-Pinto, F. M. Arrabal-Campos, J. A. Martínez-Lao, A. Cama-Pinto, and F. Manzano-Agugliaro, "Precision Agriculture and Sensor Systems Applications in Colombia through 5G Networks," *Sensors*, vol. 22, p. 7295, Sept. 2022.
- [26] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "FarmBeats: An IoT platform for Data-Driven agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, (Boston, MA), pp. 515–529, USENIX Association, Mar. 2017.
- [27] IBM, "Ibm watson." <https://www.ibm.com/downloads/cas/ONVXEB2A>. Accessed on 30 July, 2023.
- [28] Thingsboard, "Thingsboard: Open-source iot platform." <https://thingsboard.io/>. Accessed on 30 July, 2023.
- [29] E. Garcia-Villegas, A. Lopez-Garcia, and E. Lopez-Aguilera, "Genetic algorithm-based grouping strategy for ieee 802.11 ah networks," *Sensors*, vol. 23, no. 2, p. 862, 2023.
- [30] W. Rafique, A. S. Hafid, and S. Cherkaoui, "Complementing IoT Services Using Software-Defined Information Centric Networks: A Comprehensive Survey," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23545–23569, 2022.
- [31] T. Huang, S. Yan, F. Yang, T. Pan, and J. Liu, "Building SDN-Based Agricultural Vehicular Sensor Networks Based on Extended Open vSwitch," *Sensors*, vol. 16, p. 108, Jan. 2016.
- [32] D. Churkin, S. Sugavanam, N. Tarasov, S. Khorev, S. Smirnov, S. Kobtsev, and S. Turitsyn, "Stochasticity, periodicity and localized light structures in partially mode-locked fibre lasers," *Nature communications*, vol. 6, no. 1, p. 7004, 2015.
- [33] S. Trimpe and D. Baumann, "Resource-aware iot control: Saving communication through predictive triggering," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5013–5028, 2019.
- [34] K. K. R. Samal, K. S. Babu, S. K. Das, and A. Acharaya, "Time series based air pollution forecasting using sarima and prophet model," in *proceedings of the 2019 international conference on information technology and computer communications*, pp. 80–85, 2019.
- [35] N. Ahmed, D. De, F. A. Barbhuiya, and M. I. Hussain, "Mac protocols for ieee 802.11ah-based internet of things: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 916–938, 2022.
- [36] B. Bellekens, L. Tian, P. Boer, M. Weyn, and J. Famaey, "Outdoor IEEE 802.11 ah range characterization using validated propagation models," in *Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2017.
- [37] O. Raeisi, J. Pirsikainen, A. Hazmi, T. Levanen, and M. Valkama, "Performance evaluation of ieee 802.11 ah and its restricted access window mechanism," in *2014 IEEE international conference on communications workshops (ICC)*, pp. 460–466, IEEE, 2014.
- [38] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.
- [39] Q. Fan, J. Fan, J. Li, and X. Wang, "A multi-hop energy-efficient sleeping MAC protocol based on TDMA scheduling for Wireless Mesh Sensor Networks," *Journal of Networks*, vol. 7, no. 9, pp. 1355–1361, 2012.
- [40] A. Leon and J. Cesar, "Evaluation of IEEE 802.11ah Technology for Wireless Sensor Network Applications," Master's thesis, Tampere University of Technology, Tampere, Finland, 2015.
- [41] E. Felemban and E. Ekici, "Single hop ieee 802.11 dcf analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3256–3266, 2011.