Whisper: IoT in the TV White Space Spectrum

Tusher Chakraborty and Heping Shi, *Microsoft;*
Zerina Kapetanovic, *University of Washington;* Bodhi Priyantha, *Microsoft;*
Deepak Vasisht, *UIUC;* Binh Vu, Parag Pandit, Prasad Pillai, Yaswant Chabria,
Andrew Nelson, Michael Daum, and Ranveer Chandra, *Microsoft*

https://www.usenix.org/conference/nsdi22/presentation/chakraborty

This paper is included in the Proceedings of the
19th USENIX Symposium on Networked Systems
Design and Implementation.

April 4–6, 2022 • Renton, WA, USA

978-1-939133-27-4

Whisper: IoT in the TV White Space Spectrum

Tusher Chakraborty, Heping Shi, Zerina Kapetanovic[†], Bodhi Priyantha, Deepak Vasisht[‡], Binh Vu,
Parag Pandit, Prasad Pillai, Yaswant Chabria, Andrew Nelson, Michael Daum, and Ranveer Chandra
Microsoft, [†]University of Washington, [‡]UIUC

Abstract

The deployment of Internet of Things (IoT) networks has rapidly increased over recent years – to connect homes, cities, farms, and many other industries. Today, these networks rely on connectivity solutions, such as LoRaWAN, operating in the ISM bands. Our experience from deployments in multiple countries has shown that such networks are bottlenecked by range and bandwidth. Therefore, we propose a new connectivity solution operating in TV White Space (TVWS) spectrum, where narrowband devices configured for IoT can opportunistically transmit data, while protecting incumbents from receiving harmful interference. The lower frequency of operation extends the range by a factor of five over ISM bands. In less-densely populated area where larger swaths of such bandwidth are available, TVWS-based IoT networks can support many more devices simultaneously and larger transmission size per device. Our early experimental field work was incorporated into a petition to the US FCC, and further work influenced the subsequent regulations permitting the use of IoT devices in TVWS. We highlight the technical challenges and our solutions involved in deploying IoT devices in the shared spectrum and complying with the FCC rules.

1 Introduction

The growth in IoT is accelerating and expanding across a wide variety of industries. Networking has emerged as a fundamental challenge for IoT. Current solutions like LoRaWAN rely on narrowband (NB) connectivity in the ISM bands, such as US915, EU868, CN779, and so on [23]. However, as IoT networks continue to expand, they run into bottlenecks of these networking solutions. Consider an agriculture scenario, where IoT devices are used to enable precision agriculture techniques on farms in remote areas. These farms can span tens of thousands of acres. LoRaWAN has a communication range of up to 2.5 miles [4]. To connect such vast coverage areas, multiple LoRa gateways need to be deployed and maintained, where deploying a single gateway may cost thousands of US dollars. Besides, setting up proper backhaul connectivity for a gateway is cumbersome in remote areas. It holds for several other scenarios, such as oil and gas fields, power grids, wind farms, and so on. Furthermore, ISM bands have a limited bandwidth, e.g., only 8 MHz in EU868 and 26 MHz is US915 where bandwidth allocated for downlink communication is even smaller (see Section 2). Therefore, it becomes challenging to support IoT applications such as heatmap-based monitoring and plant stress monitoring using cameras, where comparatively larger volumes of data traffic is required to be transmitted over a low data rate long-range network [7, 17, 18]. In such cases, the combination of longer-range and larger available bandwidth is required.

To bridge the gap, we envision enabling IoT networks over the TV white spaces (TVWS). TV white spaces are the allocated, but unused channels in the VHF and UHF broadcast TV bands that can be leveraged for both high and low bandwidth data transmission. There are several advantages of utilizing TVWS spectrum for a NB IoT deployment over ISM bands. As the TV band spectrum consists of lower frequencies than the 800/900 MHz ISM bands, it facilitates longer-range connectivity which extends to 10s of miles, with non-line of sight (NLOS) operations, and even through some obstructions. Consequently, it opens the door of covering a large-area IoT deployment with one or a minimal number of gateways. It even facilitates higher data rates for distant IoT clients which in turn provides power savings on the IoT device. In addition, in less densely populated areas, there are typically several unused TV channels available for use by TVWS devices. The actual number of available channels vary by location. In the aggregate, these available channels can offer a large bandwidth, and hence support for many more simultaneous communication channels and increased traffic.

However, in the way of realizing this vision, the challenges are twofold – regulatory and technical. While operating in a dynamic spectrum as the unlicensed user, the precondition is to ensure the protection of incumbents from receiving harmful interference and the inability to claim protection from interference. Although this challenge has been addressed in the case of unlicensed broadband devices operating in the TVWS [5], it is non-trivial to extend the same to NB devices due to their limited power budget and distinct regulations for NB operation, such as channel occupancy limit (Section 3). We identify three corresponding challenges below:

- First, in a large-area deployment, the data rate of sparsely deployed clients (Figure 7c) served by a single gateway becomes highly variable as the data rate is inversely proportional to the distance. Single configuration setting of slow data rate (longer range) for all the clients, will result in throughput loss and power overhead. On the other hand, mainstream IoT MAC protocols, mainly designed for ISM bands, cannot make the best utilization of wide TVWS spectrum in serving large traffic even using a gateway with multi-data rate support on a single channel (Section 7.2).

- The second challenge is handling the spatio-temporal dynamism in TVWS channel availability and quality. The dynamism is mainly due to the channel occupancy by nearby licensed users (e.g., TV station, wireless microphone, etc.) and unpredictable unlicensed users (e.g., TVWS broadband network). Moreover, the long spatial separation between the gateway and client devices implies that uplink and downlink may operate on different channels with dissimilar quality. Finally, the power constraint of IoT client devices adds a curb on dynamic spectrum access and management.
- The final challenge is to develop an efficient carrier sensing solution to detect the presence of an interfering RF transmission from incumbents – both licensed and unlicensed – in a dynamic spectrum. With LoRa modulation, the devices can communicate even when the signal level is below the ambient RF noise floor. Hence, the conventional approach of simply measuring RF energy level in a NB channel to detect RF interference does not work.

Over the years, we have worked on devising Whisper, an end-to-end IoT network system over the TVWS spectrum which addresses both regulatory and technical challenges. Our early field work, authorized under an experimental license, led to a proposal on NB TVWS device operations which was the part of a broader 2018 petition for rulemaking to the US Federal Communications Commission (FCC) for expanding its rules for TVWS devices. Later, our work supported the FCC's December 2020 decision to adopt regulations on NB white space devices to operate in the VHF and UHF bands below 602 MHz [10]. In addition, we make the following contributions through this work.

Whisper Protocol: We design a new MAC protocol for a star-topology IoT network operating in the TVWS spectrum. Our Frequency Time Division Multiple Access (FTDMA) based design supports larger traffic along with diversity in the data rate of sparsely deployed clients. It leverages a dynamic binary counting table with the linear Diophantine equation for formalizing and optimally limiting the channel occupancy to protect the incumbents. The protocol further incorporates a smart approach for handling the dynamism in the TVWS spectrum given the power limitation of IoT devices.

Whisper Hardware: We design and develop a NB Whisper radio that operates in the continuous spectrum ranging from 150MHz to 960MHz. The radio uses LoRa modulation at the physical layer. Given that and the above-mentioned challenge in corresponding carrier sensing, we further develop a spectrum sensing module that uses a locally generated signal by a Whisper radio to measure the RF interference from the incumbents in individual NB TVWS channels.

Real-world Deployment: Finally, we make a real-world deployment of Whisper as an end-to-end IoT network system for more than 2.5 months. The deployment covers 17 fields in a 8500 acre farm with single gateway and 20 IoT devices. The sensor data collected via Whisper is used by third-party users in multiple agriculture applications including food tracing, data-driven farming [32], and carbon monitoring.

Data type	Area (acre)	#gateways	#clients	Traffic (bytes/hr)	Prominent issue
Sensor	1700	3	11	3.2k	Range
Sensor	350	2	20	0.9k	Range
Sensor	700	3	9	0.5k	Range
Image	8500	2(Abortive)	20	550k	Bandwidth

Table 1: Setup of multiple real-world IoT deployments where Whisper would befit. These are representative data from our deployments across the globe using ISM band LoRa.

From our real-world deployment, we find that Whisper facilitates at least 5x range improvement over LoRa operating in the 800/900 MHz ISM band and at least 3x over state-of-the-art modulation techniques proposed for NB operation in TVWS [27,28]. Furthermore, our simulation shows that using only 3 white TV channels (6 MHz each), Whisper can handle at least 5x traffic compared to ISM band.

2 Motivation from Real-world Experience

The need for deploying IoT devices in TVWS spectrum is motivated by the bottlenecks experienced in our real-world deployments using LoRa operating in the ISM bands. We highlight two application scenarios here.

The first application scenario emanates from one of our earlier projects, FarmBeats, that aims to enable data-driven agriculture [32]. To do so, we deploy sensors across a farm and aggregate data in a star-topology LoRa network operating in 800/900 MHz ISM bands. We have made more than 30 research deployments in farms across the globe (including US, South Asia, Europe, Africa, Asia Pacific, etc.) over a period of 4 years. The top three entries in Table 1 are representative of the setup of these deployments. The major challenge we have experienced in these settings is the relatively short range of the communication link compared to the size of a farm and sparsely deployed IoT clients. The average maximum achieved range is 1.12 miles combining both NLOS and LOS settings. Consequently, we need to deploy multiple gateways to cover a farm, even when we need to support just a small number of sensors spread across the farm. Whereas, TVWS spectrum offers a range of tens of miles, and thus, reduces the number of gateways as we show in Section 7.

In the second application scenario (bottom entry in Table 1), we study the feasibility of monitoring plant stress using a camera in US915 ISM band [24]. Each client sends an image of ~ 25 kB where the gateway can expect at least 22 images per hour. Here, the foremost problem is sending a large number of confirmed-up frames as the ISM band suffers from the paucity of bandwidth in two levels. First, the dwell time restriction of 0.4 sec enforces sending an image in a large number of small uplink frames. It, in turn, increases the load on the downlink (allocated bandwidth is 4 MHz) with a large number of ACKs. For example, if we consider the median uplink data rate (DR2) supported in US915 band, it takes around 230 uplink frames for an image [3]. On the downlink side, with a comparable data rate (DR12), a gateway can serve ACKs for maximum 7 images per hour without even considering any frame loss and the inefficiency of existing MAC protocols in handling confirmed-up frames [3, 16]. Multiple existing research work report similar problem [7, 17, 18]. Even one of the largest commercial LoRaWAN service providers, The Things Network, recommends finding an alternative platform in such scenarios [22]. Furthermore, with the aforementioned data rates, the maximum range can be up to a mile. In this scenario, TVWS spectrum offers a larger bandwidth that can easily handle the aforementioned traffic (Section 7.2).

3 Regulating NB Operation in TVWS

Although FCC has adopted regulations on NB operation in TVWS spectrum in 2020, we have been working with FCC on it for more than four years. FCC's Office of Engineering Technology granted us several experimental licenses for operating NB IoT TVWS devices in an agricultural setting. From the beginning, the experimental NB TVWS transmitter and network architecture have been designed with the understanding that the primary users of these frequency bands must be protected from receiving harmful interference. Based on the experience gained over the course of the field tests, we filed a petition for rulemaking at the FCC for expanding TVWS operations that included NB. Next, we describe the key regulations mandated by FCC for NB operation in TVWS spectrum [10].

- Incumbents are protected through a geolocation and database method. The location of the NB is provided through an incorporated geolocation decision, typically GPS [9]. The geolocation information is provided to a white spaces database (WSDB). The WSDB combines information on incumbent users from the FCC Licensing and Management System that is updated daily; information from other users input directly, such as wireless microphones that is updated hourly; and a calculation engine that determines the list of available channels for the TVWS device operating at that location.
- A TV channel in the US is 6 MHz wide. The conducted power and power spectral density limits for broadband TVWS devices are based on 100 kHz. Thus, the channel size limit for a NB device is proposed to be 100 kHz. The proposed channel plan requires NB TVWS devices to operate at least 250 kHz from the edge of a 6 MHz TV channel. It implies that NB devices are permitted to operate within 55 possible 100 kHz NB channels in the center 5.5 MHz of each TV channel.
- FCC limits transmissions by a NB TVWS device on each NB channel to a total of 36 sec per hour. It means that different NB channels may be required for the uplink/downlink data communication and interaction with the WSDB.
- The conducted power and power spectral density limits for NB devices per 100 kHz are the same as for broadband TVWS devices. Here, the EIRP of a NB TVWS device can be up to 18.6 dBm/100 kHz. The rules for protecting incumbents are based on a scenario where each of the 55 channels of 100 kHz in a TV channel is concurrently being used at its conducted power limit.

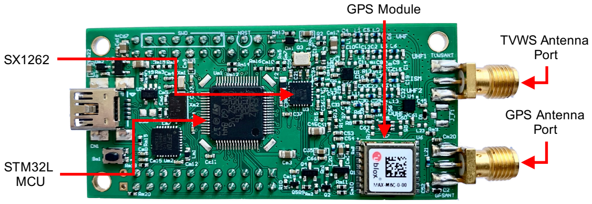


Figure 1: Whisper radio fabricated as an industrial-grade module that can operate from 150MHz to 960MHz.

We highlight that the NB devices can operate in 174-216 MHz in the VHF band and 470-602 MHz in the UHF band.

4 System Design

Whisper is a new IoT system that can support long-range communication at large-scale in the TVWS spectrum. We design Whisper to have two key components similar to a classic star-topology IoT network: an IoT client radio and gateway. In the following subsections, we describe each component.

4.1 Whisper radio

Whisper requires a radio that can operate over the whole TVWS spectrum in both VHF and UHF bands as mentioned in Section 3. We further intend to use LoRa modulation in the physical layer, which is very popular for long-range and low-power wide area network (LPWAN). Since commercial off-the-shelf LoRa radios are designed to operate over narrow ISM bands, we develop a custom NB IoT radio that can operate over a wider TVWS spectrum. Figure 1 shows the fabricated radio which can operate from 150 MHz to 960 MHz including the upper-VHF, UHF, and ISM bands. For modulation and demodulation of LoRa signal, our radio incorporates an SX1262 radio transceiver manufactured by Semtech [12].

Unlike an off-the-shelf radio designed using SX1262 for narrow ISM bands, our design for a wider TVWS spectrum requires careful consideration of avoiding leakage and harmonics in an adjacent TV channel or other licensed bands within the spectrum. The power amplifier stage of the off-the-shelf radio chips typically has low linearity to reduce power consumption. This non-linearity results in RF signals at harmonics of the carrier frequency. In a design for a wide spectrum, harmonics of the lower carrier frequencies can lie within the higher frequencies of the spectrum, resulting in spurious RF emissions. As a remedy, our radio design incorporates a collection of electronically switchable RF filters between the SX1262 radio transceiver and the antenna. The filter cut-off frequencies are selected in a way such that, with the appropriate filter selected, the harmonics of any carrier

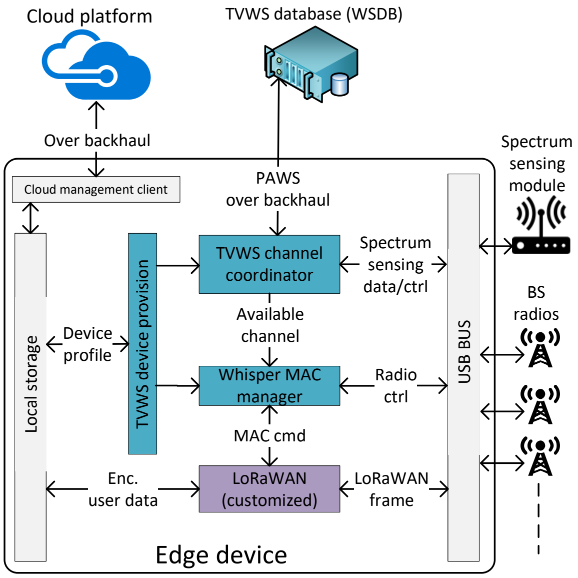


Figure 2: Gateway architecture. Whisper implements it to enable the gateway operation in the TVWS spectrum.

signal within 150 MHz to 960 MHz are filtered out preventing spurious RF emissions. We further carry out laboratory experimentation to ensure that the adjacent channel emissions limits of Whisper radio comply with FCC regulations. Find the setup and results of the experimentation in Appendix A.

Whisper radio has a low-power GPS module to provide its geo-coordinates and height to the WSDB, as required under the FCC rules for unlicensed white space devices (Section 3). To control all of these components and execute our communication protocol, we use an ultra-low-power microcontroller, STM32L151RE based on ARM cortex-M3 architecture [30].

The maximum current consumption by Whisper radio is 119 mA in transmission (at 20 dBm transmit power) and 12 mA in reception with 3.3V power supply. Although the TX energy consumption seems to be higher compared to off-the-shelf ISM band LoRa radio, Whisper radio consumes less energy per given throughput while communicating at a distance of more than a mile. Because the lower frequency of TVWS spectrum enables higher data rate, i.e., less time on air, at a longer distance compared to the ISM band (Section 7.1.4).

4.2 Whisper Gateway

Using the Whisper radio as an IoT client device, we also need a gateway to enable end-to-end communication. Figure 2 shows the architecture of Whisper gateway. It consists of multiple Whisper radios, an edge device, and a spectrum sensing module. First, the radios integrated with the gateway are referred to as base station (BS) radios and follow a similar design to the Whisper radio mentioned above. The number of BS radios can be adjusted depending on the scale of application. Next, the edge device is off-the-shelf can be a single board computer (e.g. Raspberry Pi, Up Board, etc.) or even a laptop PC. Finally, the spectrum sensing module and BS radios are connected to the edge device through a USB hub.

The edge device has the Whisper MAC manager at its core, which facilitates IoT communication over the TVWS spectrum. It administrates the medium access by the clients (Section 5), coordinates network bootstrap followed by data communication (Section 5.4), and handles the dynamism in TVWS spectrum (Section 6). To do so, it requires exchanging MAC commands with the clients. Here, MAC commands and user data are wrapped in standard LoRaWAN frame format [11]. To be specific, we use the security provided by LoRaWAN along with its frame format, however, not any associated MAC protocol. Consequently, we customize the MAC commands according to our protocol.

TVWS channel coordinator prepares the list of available channels to be used in the network. To do so, it directly communicates with the TVWS database (WSDB) for TV channel availability in the region using the standard protocol to access white space (PAWS) [8]. Furthermore, it conducts real-time screening of the uplink NB channels using the spectrum sensing module (Section 6).

5 Whisper MAC Protocol

With TVWS, we can reduce the number of gateways in a large-area deployment and still support endpoint devices dispersed at varying distances. A question that comes up is, why can we not take a similar approach that is used in mainstream LPWAN communication such as pure or slotted ALOHA? Unfortunately, these protocols do not perform well in our targeted scenarios [1, 14, 16, 26]. In particular, these protocols are not suitable for applications requiring confirmed-up frames (e.g., cameras for plant stress monitoring) and higher traffic load in a single-gateway network (Section 7.2). Besides, a gateway following these protocols appears to be even more inefficient in handling the diversity in data rate demands by the clients [13, 19]. Most importantly, the dynamic nature of the TVWS spectrum is not considered in existing MACs, since these are primarily designed for ISM band operation.

We design and implement FTDMA based MAC protocol to address these challenges. However, note that there are two apparent overheads of FTDMA based protocol for LPWAN: synchronization frame and scheduling control frame. These are neutralized by leveraging the complementary advantages from FCC-mandated compulsory parts of the system. Here, we utilize the onboard GPS module, mounted to comply with FCC regulations detailed in Section 3, for synchronization purposes. Furthermore, we piggyback the scheduling info in the FCC mandated regulatory control frames.

5.1 FTDMA structure

Whisper's MAC protocol has a custom FTDMA structure at its core. We first introduce the structure and corresponding

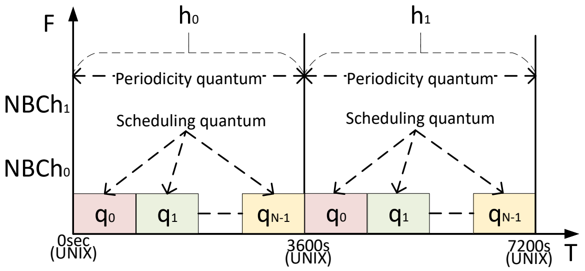


Figure 3: FTDMA structure of Whisper MAC. For communication, each client uses a slot that is a group of consecutive scheduling quanta inside a periodicity quantum.

components. Figure 3 shows a high-level depiction of the proposed FTDMA structure. The T-axis represents time tracked in seconds in the form of a UNIX timestamp. The origin point of the T-axis is starting of the UNIX timestamp which is universal. We divide the T-axis into two dividing time units: periodicity quantum and scheduling quantum. The required periodicity of periodic traffic is basically a multiple of the periodicity quantum. Each periodicity quantum is further divided into an equal number of scheduling quantum. Scheduling quantum (q) is the minimum time precision required in the scheduling. Both periodicity and scheduling quantum can be adjusted based on the computation capability of the edge device and the required precision of scheduling parameters for the application scenario. For the application in our real-world evaluation, we use the hour as the periodicity quantum. To keep the coherence in the rest of the paper, we use hour (h) in place of periodicity quantum. Here, each hour in T-axis is denoted with h_n (Figure 3), where n is the number of hours from the origin point of the T-axis. As mentioned above, we further break each hour down into N scheduling quanta where n^{th} one is denoted with q_n. Now, the F-axis represents the NB channels to be used. Note that if the gateway has multiple BS radios, the T-axis, as well as the set of scheduling quanta, is separate for each radio, however, the F-axis is shared. This allows multiple BS radios to operate simultaneously across different NB channels. The final component of the FTDMA structure is the slot, a group of *consecutive scheduling quanta*. For communication, each client is allocated at least one slot. Note that, in a slot, not more than one uplink (downlink) channel is used. However, multiple uplink (downlink) frames can be transmitted in a slot.

The IoT clients can generate two different patterns of traffic: periodic and event-driven. Here, the timing required for communication depends on the traffic pattern. Depending on the data rate and size of the payload, the time length required for communication varies. We formulate these requirements as the slot requirement in our FTDMA structure. To fulfill the requirement, the slot allocation algorithm of Whisper MAC optimally allocates slots along with communication channels in compliance with the occupancy limit. In the following subsections, we describe the slot allocation algorithm in detail for the two aforementioned traffic patterns.

5.2 Slot allocation for periodic traffic

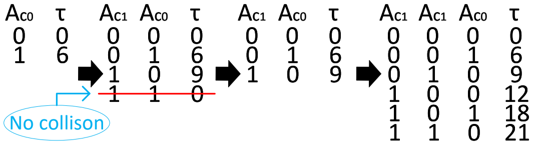
A client generating periodic traffic requires slots at a regular periodicity. We define the requirement as σ scheduling quanta are required with the periodicity of p, where σ depends on the number of frames, both uplink and downlink, to be communicated and system processing time. One or multiple slots can be allocated having at least σ scheduling quanta in total. Now, the frames to be communicated can have a variable size depending on the data rate and payload size. Therefore, each frame requires at least a certain number of scheduling quanta in a slot for communication. Here, we define continuity (α) as the number of scheduling quanta in a slot. Now, α_{min} is the minimum continuity required for the communication of a frame. Next, we delineate how the slot allocation algorithm of Whisper MAC allocates slots in three phases: scheduling quantum selection, channel selection, and slot allocation.

5.2.1 Scheduling quantum selection

The goal of the scheduling quantum selection process is to find a set of σ scheduling quanta at every p hours which are not a part of the existing allocated slots. As the T-axis is separate for each BS radio, we here describe the quantum selection process for single BS radio. We first define assignment, $A{<}h_s, p{>}$, for a scheduling quantum which implies it is occupied at every p hour starting from h_s hour to fulfill a slot requirement. When two assignments of a quantum take place in the same hour, we call it a collision. Consequently, two communication slots corresponding to these assignments collide which is not desirable. Hence, given the new slot requirement, the quantum selection process makes sure that the new assignment for a quantum does not collide with the existing ones. To do so, it leverages the linear Diophantine equation. According to the theorem for solution to linear Diophantine equation, two assignments $A_1{<}h_{s1}, p_1{>}$ and $A_2{<}h_{s2}, p_2{>}$ collides iff $|h_{s1} - h_{s2}|$ is a multiple of $gcd(p_1, p_2)$. For now, we assume that $p \in \mathbb{Z}^+, 1hr \leq p \leq 24hr$ (cases outside this boundary are discussed in Appendix B.4). Given the boundary of p, $h_s \in \mathbb{Z}^+, 0hr \leq h_s \leq 23hr$.

Now, what if any scheduling quantum is not found collision-free for the new assignment? In this case, we modify the new assignment by making it silent in the hours of collision with existing assignments. As a result, the new client will halt its transmission in the slots of those hours. It is apparent that silencing has an effect on the throughput of the new client, and thus, it should be minimum. Here, we introduce a metric $\omega_{q\%}$ that measures the frequency of silencing. It can be calculated from the generic solution of the Diophantine equation. See Appendix B.1 for more details.

The scheduling quantum selection process tunes the value of h_s from h_0 to $h_{(p-1)}$ for the required periodicity p and makes the list of collision-free scheduling quanta with and without modifying the new assignment to be silent. For the slot selection algorithm, it sets a priority value ($\rho_q \in \mathbb{R}, 0 \leq$



A_{c0}	τ
0	0
1	6

A_{c1}	A_{c0}	τ
0	0	0
0	1	6
1	0	9
~~1~~	~~1~~	~~0~~

A_{c1}	A_{c0}	τ
0	0	0
0	1	6
1	0	9

A_{c1}	A_{c1}	A_{c0}	τ
0	0	0	0
0	0	1	6
0	1	0	9
1	0	0	12
1	0	1	18
1	1	0	21

Figure 4: Construction of channel occupancy table.Whisper uses it to formulate and optimally limit the channel occupancy.

$\rho_q \leq 1$) to every scheduling quantum in the list as following.

$$\rho_q = \begin{cases} 1, & \text{has old } A[], \text{ no collision with new } A \\ 2/3, & \text{does not have assignment} \\ \frac{1-Max(\omega_{q\%})}{3}, & \text{has old } A[] \text{ and collides with new } A \end{cases}$$

A scheduling quantum having existing assignments and no collision with the new assignment gets the higher priority compared to the one with no existing assignment. It ensures optimal usage of a quantum by grouping non-colliding assignments together. Finally, a quantum having existing assignments colliding with the new one gets the lowest priority depending on the frequency of silence.

5.2.2 Channel selection

The channel selection process, completely independent of quantum selection, optimally finds the channels for new slot requirement complying with the occupancy limit. First off, if two assignments use the same channel in the same hour, it is a collision in channel usage. However, the collision is safe, i.e., assignments are valid, as long as the total occupancy of the channel in that hour is not more than 36s (Section 3). Based on it, validating a new assignment for a downlink channel is challenging. A downlink channel might be assigned to multiple clients, and these assignments can collide in different combinations and hours. Furthermore, in a gateway having multiple BS radios, although the T-axis is separate for each BS radio, the F-axis is shared, and the channel occupancy is calculated in aggregate for all BS radios. Now, simply avoiding the collision in channel usage by making the assignment silent in the colliding hour or using spare channels result in significant throughput loss and wastage of bandwidth respectively. Therefore, we need to formulate the channel occupancy and optimally limit the same despite collision among assignments in channel usage. Here, we propose binary counting based dynamic table on top of the linear Diophantine equation to validate an assignment for a downlink channel.

Now, we illustrate the dynamic construction of channel occupancy table with Figure 4. We here define the assignment for channel as $A_c{<}A, \tau, q[]{>}$, where τ denotes the channel occupancy (in seconds) per hour for the assignment, and $q[]$ denotes the list of scheduling quanta associated with it. $q[]$ is used in tracking the channel overlap in FDMA. As shown in Figure 4, each entry in the table contains the colliding assignments and corresponding total occupancy (τ) in the colliding hour of the assignments. For example, an entry $<A_{c3} = 1, A_{c2} = 0, A_{c1} = 1, A_{c0} = 1, \tau = 17>$ means A_{c3}, A_{c1}, and A_{c0} collide in channel usage, and the channel occupancy is 17s in the colliding hour. Now, if there are n assignments for a channel, there are 2^n possible combinations of assignments. However, the assignments in a possible combination may not collide, and thus it can be ruled out for future computation. For example, A_{c0} and A_{c1} do not collide, and thus the last entry in the second table has $\tau = 0$ that is removed for any further development. Find step by step description of table construction with example in Appendix B.2

The channel selection process leverages the channel occupancy table in finding a valid channel despite the collision between new and existing assignments in an hour. However, the question arises which channel is optimal to select? To do so, we here utilize a priority variable (ρ_c) for a channel similar to (ρ_q). However, $\omega_{c\%}$ is treated differently compared to $\omega_{q\%}$. In case of a collision between the new and existing assignments in channel usage, although the occupancy becomes higher only in the hour of colliding occurrence, the channel remains under-utilized in the rest of the hours of non-colliding occurrences. Consequently, a higher value of $\omega_{c\%}$ results in lesser under-utilized hours. Hence, instead of $\frac{1-Max(\omega_{q\%})}{3}$, we use $\frac{Avg(\omega_{c\%})}{3}$ in calculating ρ_c for a channel. Although it cannot ensure complete eradication of under-utilized hours for a channel, we later show how the channel can be utilized further in these hours for event-driven traffic. In aggregate, for a given assignment A_c, the channel selection process finds a valid downlink channel using the occupancy table with the maximum possible ρ_c.

5.2.3 Slot allocation

The slot requirement from a client can be satisfied with a single or a combination of slots. From a high level, the slot allocation algorithm works in two steps. First, it prepares different possible combinations of slots using the scheduling quantum selection and downlink channel selection (find the pseudo code in Appendix B.3). Then, it selects the best combination of slots based on the following weight function.

$$f = w_1\rho_q + w_2\rho_c + w_3\alpha - w_4\psi$$

We have discussed ρ_q and ρ_c earlier. Positive weight on higher continuity (α) value takes lesser slots to fulfill the requirement of σ quanta, which, in turn, saves the power of the client. On the other hand, ψ quantifies the wastage of quanta, i.e., how many extra quanta are used in total for a combination of slots. Consequently, a negative weight on ψ reduces the wastage in bandwidth-sensitive applications. All weights can be further tuned based on the nature of the application.

5.3 Slot allocation for event-driven traffic

In event-driven traffic, a frame is generated based on an event-trigger, e.g., rain monitoring. The frame needs to reach the gateway before a certain time. Therefore, we define the requirement as σ scheduling quanta are required with α_{min} con-

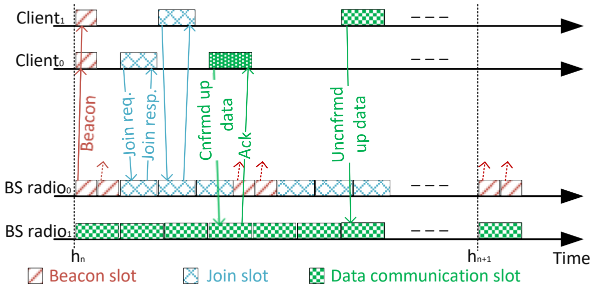


Figure 5: Time sequence diagram of Whisper protocol in a simple example scenario. This example depicts bootstrapping of two clients followed by data communication with the gateway having two BS radios.

tinuity before the expiry, ε. The underneath concept of slot allocation algorithm for event-driven traffic is mostly similar to the periodic traffic. We point out the *distinct factors* below.

During scheduling quantum selection, we check the collision of an assignment for event-driven traffic with the existing assignments for both types of traffic. However, there is no notion of silencing the new assignment for event-driven traffic as its time-sensitive. Accordingly, ρ_q has three different priority levels - highest priority to no collision with existing assignments, then to no existing assignments, and lowest to collision only with existing assignments for periodic traffic. Here, in the last case, even if the existing periodic assignment ends up transmitting frames in the same channel together with the new event-driven assignment, according to the nature of LoRa physical layer, there is a probability of one getting successfully demodulated. In this way, the scheduling quantum selection process enlists all quanta before the expiry, ε.

Although the underneath concept remains similar, the channel selection for event-driven traffic is more simplistic than periodic traffic. Here, rather than using the channel occupancy table, the validity of a new assignment for a channel is probed by simply checking collision with individual existing assignment for both types of traffic in the hour of the new assignment. If the new assignment collides with existing assignments in channel usage and the total τ of all colliding assignments including the new one remains within the limit, we consider it valid. The priority of a channel (ρ_c) for a valid new assignment is decided differently – highest priority to no collision with existing assignments, then to collision with existing assignments, and lowest to no existing assignments. It facilitates the utilization of a channel in no or low utilization hours of the periodic assignments.

Finally, we make a modification in the weight function for slot allocation as following.

$$g = w_1\rho_q + w_2\rho_c + w_3\alpha - w_4\psi - w_5\delta$$

where δ is the time difference between selected scheduling quantum and the expiry (ε). A negative weight on δ makes the choice for immediate slot less greedy. It, in turn, facilitates serving multiple concurrent requests of event-driven traffic without colliding in the immediate slot.

5.4 Bootstrap and Data Communication

Hitherto we discuss the process of slot allocation for controlling the medium access by the clients. We now describe how Whisper MAC manages client bootstrapping and data communication using the slot allocation. Client bootstrapping is a critical part of a TVWS based IoT network since it is required to comply with FCC regulations given the power constraint of the IoT devices. Recall that, each NB TVWS device must provide the WSDB its geo-coordinates and height to obtain the list of available channels for transmission at that location. Here, the clients can register via the gateway as they do not have Internet connectivity. Nevertheless, a key question remains. How does a client share its location with the gateway without knowing the available channel and time for transmitting registration request?

According to the FCC regulations, a client that is not registered in the WSDB can transmit only its location information or network join request on the channels registered against the gateway. To do so, a client requires to know the channels registered against the gateway where the list of these channels may even vary over time. As a remedy, Whisper MAC utilizes broadcast beacons that embed the info of join slots for clients to transmit WSDB registration, i.e., network join request. In the join slot, a client transmits the join request along with its location for WSDB registration and slot requirement for the data communication. Note that the beacon slots have predefined and fixed timing and channels. Therefore, the time synced (using GPS) clients can listen for the beacon frames without draining its power in random or continuous scanning. Figure 5 depicts an example scenario of client bootstrapping followed by data communication. Due to space limitations, we put the corresponding implementation details in Appendix B.5.

After receiving the WSDB registration response, i.e., join response, a client is ready for the data transmission in the allocated slots (piggybacked as MAC command in the response frame) until the expiry of channel. A TVWS device is required to contact (poll) the WSDB once every 24 hours and check the list of available channels at the location. The poll request is piggybacked as a MAC command in a confirmed-up data frame. The expiry generally gets extended after the polling, and updated expiry is sent in the downlink ACK. However, polling is not required for a client generating event-driven traffic as it freshly joins the network whenever it has a data frame to transmit.

6 Dynamic Spectrum Access

The dynamic nature of TVWS spectrum can lead to bottlenecks when it is utilized for NB IoT networks. First, the long distance between the gateway and a client brings about separate uplink and downlink having dissimilar quality at their corresponding locations. Second, the IoT clients are power

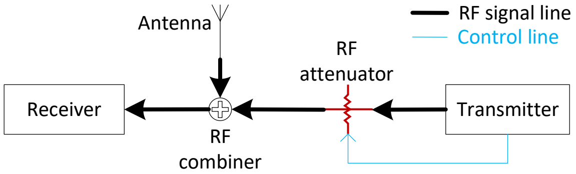


Figure 6: Block diagram of our spectrum sensing module that measures interference from the incumbent users.

constrained. Hence, any extra transmission and carrier sensing activity by the clients for spectrum management adds power overhead. Whisper addresses these challenges by implementing spectrum awareness capability and then incorporating smart spectrum exploitation approach in the protocol.

6.1 Spectrum awareness

Uplink communication is inevitably significant in most of the IoT applications as IoT traffic is push-based. The quality of this communication link directly relies on the interference nearby the gateway. Hence, we first try to enable spectrum awareness at the gateway. A part of spectrum awareness, both uplink and downlink, is achieved by using the WSDB. However, it is not enough to accumulate info about the real-time activity of incumbent users – both licensed and unlicensed – in the carrier. To do so, Whisper performs spectrum sensing.

First off, how can the interference level in a NB channel be measured given the Whisper radio uses LoRa modulation and its minimum detectable signal (MDS) is very low (-149 dBm)? The LoRa modulation enables frame reception even when the carrier signal level is below the ambient RF noise floor. Consequently, the conventional approach of measuring RF energy in a NB channel as an indication of interference level is not viable here. Instead, we design a custom off-the-shelf spectrum sensing module to evaluate the level of interference in NB TVWS channels. It can evaluate the impact of interference from licensed and unlicensed cognitive radio users at a very low signal level.

Spectrum sensing module locally generates RF signal with a controlled amplitude to determine MDS in a NB channel. The MDS gives the perception of interference from incumbent users in the NB channel – a lower MDS implies a lower RF interference. Figure 6 depicts the block diagram of the spectrum sensing module. It consists of a transmitter, a controlled attenuator, an RF combiner, and a receiver. We use two Whisper radios as the transmitter and receiver. The attenuator is controlled by the transmitter. We modify transmitter radio firmware to set the value of attenuator. To estimate the MDS in a channel, we tune the transmitter and receiver to the specific channel frequency, and the transmitter starts transmitting test frames. The RF signal of a test frame is attenuated by the attenuator to yield a weak signal similar to a reception from a remote transmitter, which is a commonly practised technique in laboratory emulation [15]. The combiner combines this weak signal with the ambient RF signals (includes interference from incumbent users) picked up by the antenna. The attenuator value is varied to identify the maximum attenuation that results in 90% successful frame reception. The lowest RSSI of the received frames is the MDS under the interference. Note that the spectrum sensing module only senses the RF channel. The generated RF signal is only used internally, and thus has no effect on data communication and channel occupancy limit. The spectrum sensing module is a part of the gateway as mentioned in Section 4. The channel control unit utilizes it every hour to update the interference level from the incumbent users.

6.2 Spectrum exploitation

The interference in the downlink at the location of a client cannot be measured with the spectrum sensing module, given the long distance. Furthermore, carrier sensing at the client's end would add huge power overhead. Here, Whisper enables the spectrum exploitation approach to mitigate the effect of interference through dynamic channel assignment that does not require any additional frame transmission.

Before jumping into the details, we first introduce the related terminologies and notations. For each slot, we define a channel-tube (ChT) that has two channels for a confirmed-up and one channel for an unconfirmed-up data frame. For a slot used in periodic traffic, N_{ChT} channel-tubes are assigned by adjusting the A_c for the associated channels during the slot allocation described in Section 5. In every occurrence of the slot, a ChT is picked in a sequential round-robin manner for communication. In case of event-driven traffic, a ChT is assigned which is different from last $(N_{ChT}-1)$ ChTs. In this way, when a slot completes hopping across all N_{ChT} ChTs, we call it a hopping cycle. Note that, a client may have multiple slots (as mentioned in Section 5) and each slot has an individual set of N_{ChT} ChTs. Furthermore, ChTs are preferably selected from different TV channels to facilitate robustness against broadband interference. Next, we delineate how the set of ChT is dynamically updated in three phases.

1. Monitoring phase: During this phase, the performance of assigned ChTs for all the slots of a client is monitored at the gateway, and a flag is raised in case of substandard performance. It is raised based on the overflow of a bucket that is filled up with the tickets for an event of missing frame in a ChT. The number of tickets for a missing event is equal to the number of previous consecutive occurrences starting from the current event in the ChT. In the event of successful communication in a ChT, we remove its tickets in a similar manner. Now, how do we detect an incident of the missing frame? We here leverage the frame count block of the standard LoRaWAN frame header. Finally, we need to determine the optimal bucket size (S_{bukt}). It is directly related to $\sum_{i=1}^{ND_{ChT}} k_i$ where ND_{ChT} is number of distinct ChTs and k is number of times it is used in a cycle. S_{bukt} depends on the length of hopping cycle too. For example, a client with a smaller cycle overflows the bucket faster in presence of a short-term interference, whereas it gets slower in case of a longer cycle with long-term interference. Both these scenarios are undesired. Here, we express S_{bukt} as $m\sum_{i=1}^{ND_{ChT}} k_i$ for different bands of cycle. From our emulation and real-world deployments, we found the following optimal values of m with $N_{ChT} = 4$ for each slot. Here, m typically exhibits an exponential downward pattern with respect to the length of cycle.

$$m = \begin{Bmatrix} 2.5, & \text{cycle length} < 1hr \\ 1.6, & 1hr \leq \text{cycle length} < 3hr \\ 1.05, & 3hr \leq \text{cycle length} < 6hr \\ 0.7, & 6hr \leq \text{cycle length} < 11hr \\ 0.5, & 11hr \leq \text{cycle length} \end{Bmatrix}$$

2. Decision making phase: After a flag has been raised, the decision making phase decides which ChT and associated channels are to be replaced. We note that a ChT is selected for replacement if its contribution in bucket overflow is at least a certain percentage defined as, $\frac{k_i}{\sum_{i=1}^{ND_{ChT}} k_i}$. If no such ChT is found, older events are removed to accommodate new events in the bucket. Next, we find which channel(s) in the ChT(s) needs to be replaced based on the knowledge from the monitoring phase. The replacement ChT is placed exactly at the same sequential position of the replaced one in the current list of ChTs for a slot.

3. Execution phase: For a client sending confirmed-up frames, the execution of ChT update is initiated immediately after the decision making. Whereas, in case of a client sending unconfirmed-up data frames, it is carried out when the client polls the channel from WSDB since this is the only time when the client sends confirmed-up data frames. The execution is initiated by the gateway by sending an update notice in the downlink frame. The client sends a notification of notice reception in the following uplink data frame. To ensure the robustness against frame loss, the update notice (if a confirmed up frame is received) and notification are sent in the following N_{ChT} slots. The timing of the update is set accordingly in the notice. Note that the execution process is separate for each slot. Furthermore, no extra frame is exchanged for the execution process since all the associated MAC commands are piggybacked in the data frames.

6.3 Fallback

Finally, in a rare event of complete communication loss with the gateway, a client calls for the fallback. Although a client sending confirmed-up data frames detects such an event straight away, a client sending unconfirmed-up data frames detects only at the time of channel polling from WSDB. In such an event, the client does not get a downlink packet in any slot and ChT. As a fallback alternative, the client rejoins the network. It reports the event in the join request so that gateway can assign a different set of ChTs.

	Avg.	Max	Min
Throughput (in bps)	0.0626	0.0681	0.0601
FDR (in %)	98.4	100	97.1
Latency (in sec)	2.39	2.43	2.37

Table 2: Performance of Whisper in real-world deployment

7 Evaluation

In this section, we focus on evaluating the performance of Whisper through a real-world deployment and simulation. We further make a comparison with the ISM band IoT solutions.

7.1 Real-world deployment

We have an ongoing deployment of Whisper on 8500 acre dry-land wheat farm in Eastern Washington, which has been operating for over 2.5 months. Here, Whisper is used by the third-party users in collecting sensor data from 17 different fields for multiple data-driven agriculture applications.

7.1.1 Setup

The deployment includes 20 Whisper client radios that communicate with a single gateway. The client radios are retrofitted inside a weatherproof box, powered by solar, and connected to a sensor interface to collect data from five sensors – temperature, humidity, CO_2, and soil moisture and temperature (Figure 7b). As configured by third-party users, 11 clients report data of all sensors at different periodicity starting from 30 min to 12hr. Similarly, the remaining clients report all sensor data in an event-driven manner based on CO_2 level. All of the data frames are confirmed-up – 51 bytes uplink, 20 bytes downlink. The modulation parameters are configured to have a coding rate (CR) of 4, 62.5 kHz channel bandwidth (ChBW), and preamble length of 8. We tune the spreading factor (SF) based on the distance of the clients from the gateway as shown in Figure 7c.

The gateway is deployed on the farm, and due to a lack of power, is powered by solar power (Figure 7a). Since there is no Internet connectivity in the middle of the farm, we use an Adaptrum TVWS broadband radio at the gateway to create a wireless link to the nearest source of connectivity (farmer's home) [2]. It in turn enables the evaluation of Whisper in coexistence with off-the-shelf unlicensed TVWS broadband network. We use a Raspberry Pi 3B as the edge device at the gateway. For WSDB, we use Wave DB Connect by RED Technologies [31]. Finally, both BS radio and clients use omnidirectional antennas with approximately 5 dBi gain where the TX power is set to the max according to Section 3.

7.1.2 Results

Thus far, 4766 sensor data points have been collected via Whisper. We evaluate the performance of Whisper in terms of three metrics: throughput, frame delivery ratio (FDR), and

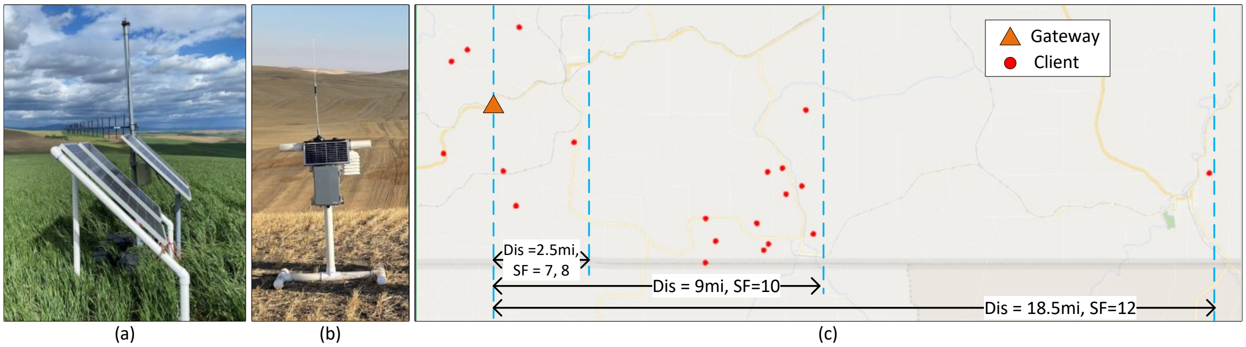


Figure 7: Whisper Deployment. The solar powered IoT Hub shown in (a) includes a broadband TVWS backhaul and Whisper gateway. In (b), Whisper client integrated with sensors and retrofitted inside a weatherproof box. (c) shows the deployment map.

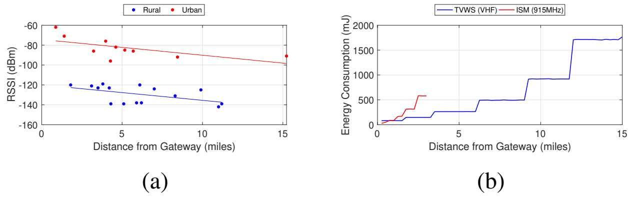


Figure 8: (a) Range of Whisper radio and (b) Comparison of energy consumption between Whisper radio in TVWS VHF band and off-the-shelf LoRa radio in ISM band.

latency. The value of each metric is averaged over a period of 24hr. We do not observe any significant deviation in the average values of these metrics throughout the deployment period. Table 2 summarizes the results. We also observe that Whisper has shown robustness against multiple real-world incidents such as power outages at gateway and clients due to thunderstorms, disrupted Internet connectivity, and WSDB disruption due to maintenance. Furthermore, Whisper shares the spectrum (470 to 488 MHz) with Adaptrum broadband TVWS radio in the deployment as we mentioned above. However, we do not observe any mutual interruption in the communication of Adaptrum radio and Whisper.

7.1.3 Range

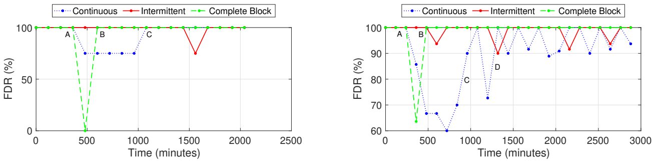
We perform range experiments to evaluate the sensitivity of the Whisper radio. Range tests are conducted in both urban and rural environments, where both have line-of-site (LOS) and non-line-of-sight (NLOS) connectivity. For the urban settings, we place a BS radio on the rooftop of an industry campus building. Then, we move the client to different locations using a vehicle and record the RSSI. The radios are configured for LoRa modulation having a ChBW of 31.25 kHz, SF of 12, and CR of 4 in the VHF band. Figure 8a shows the RSSI of received frames with respect to distance in miles. We can see that frames are successfully received at 11.3 miles with RSSI of −85 dBm. For the evaluation in rural settings, we use our aforementioned farm deployment. In this scenario, the performance is even better in comparison to the urban settings. We can see that packets are successfully received at 15 miles with a measured RSSI of −87 dBm which is significantly higher than the MDS (−149 dBm) of Whisper radio. These results imply that clients can be deployed as far as 15 miles away from the gateway, and likely even further. For example, in Figure 7c, one of the clients communicates with the gateway from a distance of 18.5 miles. Note that, the range of LoRa in ISM band is found to be 1 - 3 miles in literature and our real-world deployment [4].

7.1.4 Energy profile

We next conduct energy profiling of Whisper radio operating in TVWS spectrum and compare it with an off-the-shelf LoRa radio, SX126xMB2xAS [20], operating in the US915 ISM band. The physical setup for the experimentation is similar to the range test. Both radios are configured to send an uplink frame of 51 bytes every hour to the gateway with the same LoRa modulation configurations (CR: 1, ChBW: 62.5 kHz, TX power: 20 dBm). We then vary the distance of the radios from the gateway and tune the SF accordingly for successful communication. In Figure 8b, we show the energy consumption for each frame transmission in TVWS VHF band (174.3 MHz) and ISM band (915 MHz). Note that the energy consumption of the Whisper radio in locking GPS (once in 24hr) is prorated over its transmitted frames. Up to the initial distance of around a mile, the energy consumption in the TVWS spectrum is higher. However, as the distance increases, the time on the air for frame transmitted in the ISM band increases due to the higher spreading factor. Whereas, the lower frequency of the VHF band facilitates longer distance with comparatively lower SF. Consequently, the energy consumption becomes at least 2x higher in the ISM band compared to the TVWS VHF band after a mile of distance.

7.1.5 Performance in presence of interference

To evaluate the performance of Whisper under interference in the above-mentioned real-world deployment, we create interference with a separate transmitter. The interference is introduced on both uplink and downlink channels in three different ways - intermittent, continuous, and complete block.



(a) (b)

Figure 9: Whisper's response to different types of interference on (a) downlink and (b) uplink channels in real-world.

In the case of intermittent interference, the interferer hops across the channels of $N_{ChT} = 4$ ChTs and creates short-term interference for the period equals to the time on air of a frame. Whereas, in continuous interference, the interferer picks one of the channels from the 4 ChTs and creates continuous interference on that channel. Finally, in case of complete block, the interferer is time-synced and aware of the hoping cycle schedule of the target radio. It accordingly creates interference on the active channel for data communication.

We evaluate the impact of these three types of interference on downlink channels. The interferer is placed very close to a client with high transmit power. The interferer is aware of the downlink channels used by the client. We then record the FDR on the downlink channels and the activity of switching channels for the client. In this way, we investigate the impact of interference for the clients, having different periodicity and traffic patterns, in our aforementioned deployment. Since all the clients generating periodic traffic exhibit similar behavior, we only discuss a client having a periodicity of 30 min. Figure 9a shows downlink FDR of the client averaged over a period of 120 min (one hopping cycle). As shown in the figure, the intermittent interference has a very minor impact on the FDR. No ChT update is initiated in response, whereas, for continuous interference, we observe a significant drop in FDR after the interferer is activated (point A in Figure 9a). As a result, a ChT update is initiated by the gateway. After the execution of same, the FDR reaches 100% (point C in Figure 9a). Finally, for complete block interference, we see the complete cease in downlink communication for the client. Consequently, the client triggers the fallback mechanism and rejoins the network (point B in Figure 9a). As a new set of ChTs is assigned after rejoining, the FDR reaches the maximum. We observe a similar impact of intermittent interference and a lesser impact of continuous interference on clients generating event-driven traffic. This is due to the members of the set of 4 ChTs varying with time.Nevertheless, it is difficult to create complete block interference for event-driven traffic.

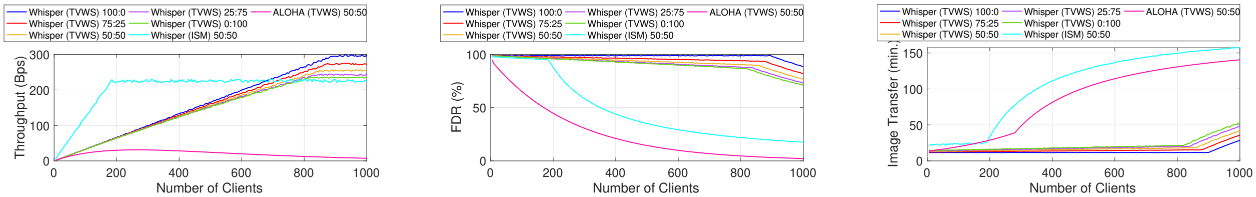
Next, we evaluate the impact on uplink channels by placing the interferer close to the gateway. We select four uplink channels of the same client for the interferer, however, these channels are used by more than half of the clients in different combinations. To get a precise understanding of the effect of interference, we record uplink frames transmitted by any client on only these four channels. Figure 9b shows corresponding uplink FDR averaged over a period of 120 min. We observe very minimal effect of the intermittent interference on the network and no ChT update took place for the same. In case of continuous interference, we see a drop in the FDR after the activation of interferer (point A in Figure 9b). Over the period of two days, ChTs of two clients are updated followed by increment in FDR (point C and D). Besides, the spectrum sensing unit detects the interference on the channel, and consequently, the channel is not further assigned to any client for sending event-driven traffic in this period by the channel coordinator. Finally, the complete block interference only affects the communication of one client. The client exhibits similar response as described above in case of the downlink.

7.2 Simulation

We now focus on evaluating Whisper's capability in terms of scale. Since deploying a very large-scale network (100s of clients) is challenging, we carry out the simulation. In particular, we evaluate two things: (1) how the larger bandwidth of TVWS spectrum facilitates scaling and (2) how Whisper MAC makes better utilization of this bandwidth compared to mainstream IoT MAC protocols. For this purpose, IoT devices transmitting images is a compelling application scenario (Section 2). We simulate this scenario in three network setups: (S1) Whisper MAC in TVWS spectrum, (S2) Whisper MAC in ISM band, and (S3) pure ALOHA (used by LoRaWAN) in TVWS spectrum. Comparison between S1 and S2 shows the role of bandwidth in our application scenario, whereas S1 and S3 show how Whisper MAC makes better utilization of this larger bandwidth.

7.2.1 Setup

We first integrate the LoRa physical layer from FLoRa simulator and the simulation environment of OMNet++ with our protocol [29]. We set up a single-gateway star-topology network where every IoT client device sends a 25 kB image. The gateway has eight BS radios. The number of clients is varied from 5 to 1000 at an increment of 5. The ratio of clients generating periodic traffic (N_p) to event-driven traffic (N_e) is also adjusted in five levels. We make an even distribution of image generation periodicity starting from 1hr to 24hr among the clients generating periodic traffic. A client with event-driven traffic sends an image once in every 24hr following random distribution. For S1 and S3, the uplink frame size is 255 bytes and downlink ACK frame is 18 bytes. The LoRa modulation parameters are set as following - CR: 1 and ChBW: 62.5 kHz. To simulate the effect of distance and associated data rates, we make an even distribution of six possible SF values $(7-12)$ among the clients. Beacon periods are separated by two minutes with a periodicity of $p_{bcn} = 1$hr (Appendix B.5). For simulation purposes, since radios are not certified real devices, we use a local proxy of WSDB where each device, including gateway, has 3 TV channels available for transmis-



(a) Throughput

(b) Frame delivery ratio

(c) Image Transfer Time

Figure 10: Simulation results showing the role of large TVWS band and how Whisper MAC better utilizes the same in scaling.

sion. For S2, we use the four uplink data rates, DR0-DR3, permitted in the US915 ISM band where corresponding SF ranges from 10 to 7 with the channel bandwidth of 125 kHz. The dwell time restriction limits the uplink payload size (in bytes) for these data rates – DR0:11, DR1:53, DR2:125, and DR3:242. On the downlink side, we use data rates, DR10-DR13, having same SFs as uplink with ChBW of 500 kHz.

7.2.2 Results

We analyze the performance in terms of throughput, FDR, and image transfer completion time.

1. Throughput: As shown in Figure 10a, we analyze the throughput of the network under a various number of clients and ratios of traffic types. Each point in the graph represents the average throughput of the network in a period of 24hr. Here, we only consider the data frames, not any frames associated with the bootstrapping process. For S1 (Whisper in TVWS), the throughput increases linearly with the number of clients for every ratio of traffic types ($N_p : N_e$) until it reaches the network capacity. As the proportion of event-driven traffic increases in a group of clients, the throughput slightly drops due to the higher collision rate among the randomly triggered events. For S2 (Whisper in ISM band) and S3 (ALOHA in TVWS), we only show the representative results from the traffic ratio of $50:50$. Although S2 shows similar patterns as S1, there are two distinct factors. First, due to the limited bandwidth in ISM band compared to the TVWS spectrum, S2 reaches the saturation point with $\sim$ 5x fewer clients than S1. Second, for the same number of clients in the network, S2 exhibits higher throughput as per-channel bandwidth is higher and supported SF is lower in the US915 ISM band compared to the TVWS spectrum. However, the lower value of supported SFs in the ISM band results in a much shorter range. In case of S3, both maximum throughput and number of supported clients are significantly lower than S1. This shows why the existing mainstream IoT MAC protocol is not suited for making better utilization of large TVWS bandwidth.

2. FDR: In computing FDR, we consider the frame generated but not successfully transmitted, even due to implied silence by Whisper MAC, as a frame loss. Figure 10b provides similar insights as the throughput. Although the initial throughput is found higher in S2 than S1, FDR is lower in the ISM band due to the larger number of small frame (both uplink and downlink) transmission for sending each image (Section 2). It, in turn, increases the probability of frame loss.

3. Image transfer completion time: We further report the time required to send an image including the re-transmission of frames. The image is dropped after 3hr, if it is not completely transmitted. As shown in Figure 10c, the completion time linearly increases for S1 and S2 until it hits the saturation point, whereas it increases exponentially for S3. Now, although S2 exhibits higher throughput than S1 at the cost of a much shorter range, S2 reports at least 2x higher completion time. Because, in ISM band, a client requires to transmit a larger number of small frames to send an image due to the cap on uplink payload size (Section 2).

8 Related Work

Even though unlicensed operations in TVWS spectrum have been extensively studied in literature [5,6,21]. Nearly all prior work have focused on broadband scenarios and corresponding FCC regulations. However, there are fundamental differences in NB operation such as power constraint of IoT devices, number of devices in the network, channel occupancy limitation, etc. These make the challenges in unlicensed NB operation non-trivial to solve. There are some prior work on devising suitable signal modulation techniques for NB operation in TVWS spectrum [25, 27, 28]. However, dynamic spectrum access and management in NB operation, compliance with the NB FCC regulations, and corresponding comprehensive MAC protocol design are beyond the scope of these work.

9 Conclusion

In this paper, we have presented Whisper, a new solution to enable long-range and wider spectrum communication for IoT by leveraging TVWS spectrum. With Whisper, we get at least 5x longer range compared to LoRa operating in ISM band. Consequently, in our real-world deployment, Whisper has covered 17 fields in a 8500 acre farm with single gateway. Besides, the lower frequency of TVWS spectrum facilitates 2x less energy consumption than ISM band at a range of more than a mile. Note that, for a deployment spanning not more than a mile from the gateway, LoRa in ISM band would be preferable over TVWS spectrum. It also holds for an indoor scenario. Finally, with only 3 white TV channels, Whisper can accommodate 5x more traffic than ISM band.

References

[1] Khaled Q Abdelfadeel, Dimitrios Zorbas, Victor Cionca, and Dirk Pesch. *FREE* — Fine-grained scheduling for reliable and energy-efficient data collection in LoRaWAN. *IEEE Internet of Things Journal*, 7(1):669–683, 2019.

[2] Adaptrum. Adaptrum TVWS broadband radio, 2021. https://www.adaptrum.com/Content/docs/acrs2_datasheet_1016.pdf.

[3] Arjan. Airtime calculator for LoRaWAN, 2021. https://avbentem.github.io/airtime-calculator/ttn/us915-dl/38.

[4] A. Augustin, J. Yi, T. Clausen, and W.M. Townsley. A study of LoRa: Long range low power networks for the Internet of Things. *Sensors*, 2016.

[5] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. White space networking with Wi-Fi like connectivity. *ACM SIGCOMM Computer Communication Review*, 39(4):27–38, 2009.

[6] Ranveer Chandra, Thomas Moscibroda, Paramvir Bahl, Rohan Murty, George Nychis, and Xiaohui Wang. A campus-wide testbed over the TV White Spaces. *SIGMOBILE Mob. Comput. Commun. Rev.*, 15(3):2–9, November 2011.

[7] Tonghao Chen, Derek Eager, and Dwight Makaroff. Efficient image transmission using LoRa technology in agricultural monitoring IoT systems. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 937–944. IEEE, 2019.

[8] V Chen, S Das, L Zhu, J Malyar, and P McCann. Protocol to access white-space (PAWS) databases. *Internet Engineering Task Force (IETF)*, 2015.

[9] Federal Communications Commission. FCC rules for unlicensed white space devices. March 2020.

[10] Federal Communications Commission. Unlicensed white space device operations in the television bands. *FCC ET Docket No. 20-36, Report Order and Further Notice of Proposed Rulemaking*, October 2020.

[11] LoRa Alliance Technical Committee. LoRaWAN 1.0.3 specification, 2018. https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf.

[12] Semtech Corporation. Semtech SX1262, 2020. https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1262.

[13] Joseph Finnegan, Ronan Farrell, and Stephen Brown. Analysis and enhancement of the LoRaWAN adaptive data rate scheme. *IEEE Internet of Things Journal*, 7(8):7171–7180, 2020.

[14] Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. Low overhead scheduling of LoRa transmissions for improved scalability. *IEEE Internet of Things Journal*, 6(2):3097–3109, 2018.

[15] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke. LoRa scalability: A simulation model based on interference measurements. *Sensors*, 17(6):1193, 2017.

[16] Md Tamzeed Islam, Bashima Islam, and Shahriar Nirjon. Duty-cycle-aware real-time scheduling of wireless links in low power WANs. In *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 53–60. IEEE, 2018.

[17] Akram H Jebril, Aduwati Sali, Alyani Ismail, and Mohd Fadlee A Rasid. Overcoming limitations of LoRa physical layer in image transmission. *Sensors*, 18(10):3257, 2018.

[18] Mookeun Ji, Juyeon Yoon, Jeongwoo Choo, Minki Jang, and Anthony Smith. LoRa-based visual monitoring scheme for agriculture IoT. In *2019 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2019.

[19] Shengyang Li, Usman Raza, and Aftab Khan. How agile is the adaptive data rate mechanism of LoRaWAN? In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 206–212. IEEE, 2018.

[20] ARM MBED. SX126xMB2xAS, 2021. https://os.mbed.com/components/SX126xMB2xAS/.

[21] Rohan Murty, Ranveer Chandra, Thomas Moscibroda, and Paramvir Bahl. Senseless: A database-driven white spaces network. *IEEE Transactions on Mobile Computing*, 11(2):189–203, 2011.

[22] The Things Network. Fair use policy explained, 2021. https://www.thethingsnetwork.org/forum/t/fair-use-policy-explained/1300.

[23] The Things Network. Regional parameters, 2021. https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/.

[24] NCSU College of Agriculture and Life Sciences. Low-cost cameras could be sensors to remotely monitor crop stress, 2022. https://shorturl.at/evxGW.

[25] Mahbubur Rahman, Dali Ismail, Venkata P Modekurthy, and Abusayeed Saifullah. LPWAN in the TV White Spaces: A practical implementation and deployment experiences. *arXiv preprint arXiv:2102.00302*, 2021.

[26] Brecht Reynders, Qing Wang, Pere Tuset-Peiro, Xavier Vilajosana, and Sofie Pollin. Improving reliability and scalability of LoRaWANs through lightweight scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, 2018.

[27] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. SNOW: Sensor network over white spaces. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 272–285, 2016.

[28] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Jie Liu, and Ranveer Chandra. Low-power wide-area network over white spaces. *IEEE/ACM Transactions on Networking*, 26(4):1893–1906, 2018.

[29] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. Adaptive configuration of LoRa networks for dense IoT deployments. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2018.

[30] STMicroelectronics. STM32L151RE, 2020. https://www.st.com/en/microcontrollers-microprocessors/stm32l151re.html.

[31] RED Technologies. Wave DB Connect forTVWS, 2021. https://www.redtechnologies.fr/sas-technology-copy.

[32] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. FarmBeats: An IoT platform for data-driven agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 515–529, 2017.

A Compliance of Emission

We conduct laboratory measurements to ensure that the emission generated by Whisper radio complies with the FCC leakage regulations. Our evaluation is twofold. First, we look at the changes in the desired to undesired (D/U) signal ratio on the first adjacent channel of select DTV receivers when the source of the undesired signal is changed from a broadband white space device (WSD) to a NB WSD (Whisper radio). Second, we evaluate the changes in the D/U ratio on the first adjacent channel of select DTV receivers in response to the airtime of Whisper radio.

A.1 Setup

As the bandwidth of a NB TVWS channel is limited to 100 kHz, we select the lowest and highest possible bandwidths supported by the Whisper radio within the limit. Recall that airtime in LoRa modulation is determined by the spreading factor (SF). In this case, we use the lowest (SF7) and highest (SF12) possible spreading factors to change the airtime from low to high, respectively. The duty cycle of the Whisper IoT radio was set at 78% (ON time 780 ms, OFF time 220 ms for each second transmission). Note that the FCC proposed duty cycle is considerably less: 1%.

As shown in Table 3, a set of DTVR - RX1, RX4, RX5, RX10, and RX12 - are identified for this evaluation to diversify over different price ranges, resolutions, dimensions, and form factors. Channel 9 in the high-VHF band (center frequency: 189 MHz) and Channel 16 in the UHF band (center frequency: 485 MHz) are selected to provide the desired DTV signal. The D/U ratio for the Whisper radio is average across four desired signal levels – moderately strong (-43 dBm), moderate (-53 dBm), moderately weak (-65 dBm), and very weak (-80 dBm) at ±3 MHz and ±6 MHz from the edge of the broadcast DTV channel. These are the same desired signal levels used in the broadband WSD laboratory testing. In this way, the two sets of measurements for the D/U ratio on the first adjacent channel can be compared. The video loop used in the D/U measurements for the broadband WSD measurements is used for this test. For each measurement, the undesired signal power is increased until artifacts are observed.

A.2 Results

Table 4 and 5 summarize the results of D/U ratio evaluation for different DTVR receivers.

For the ATSC 1.0 DTVs, (RX1, RX4, RX5, and RX10), the D/U ratio on the first adjacent channel for the NB TVWS IoT radio indicates the receivers are even more selective (i.e., the value of the D/U ratio more negative) with respect to an undesired NB WSD than an undesired broadband WSD. Note that the D/U ratio at ±6 MHz from the broadcast channel's edge is usually a few dB better (more selective) than the D/U ratio at ±3 MHz from the broadcast channel's edge.

For the ATSC 3.0 receiver (RX12) tests at a modulation level of 256QAM, the D/U ratio for NB and broadband WSDs for moderate and weak desired signals are about the same, within error. There is a 5-6 dB difference for moderately strong and very weak desired signal, where the NB WSD is more selective than the broadband WSD for moderately strong signals and the NB WSD is less selective than the broadband WSD for very weak signals. In both instances, the D/U ratio represents very high ATSC 3.0 receiver selectivity. As RX12 is tested only at 256 QAM, it could be a function of the higher threshold at this modulation level.

DTVR ID	Manufacturer	Model	Standard	Profile	Resolution	Display size
RX1	Samsung	UN65NU8000	ATSC1.0	TV	4K	65in
RX4	Samsung	UN32N5300AFXZA	ATSC1.0	TV	1080p	32in
RX5	Insignia	NS-24DF310NA19	ATSC1.0	TV	720p	32in
RX10	Mediasonic	HOMEWORX HW130STB	ATSC1.0	Set-up box	1080p	-
RX12	RedZone	TVXPLORER BUNDLE	ATSC3.0	USB dongle	HD	-

Table 3: Information of DTVR receivers

DTV center frequency = 189MHz (Channel 9)				
	Change in D/U when SF is increased from SF7 to SF12		Change in D/U when b/w is increased from 7.8kHz to 62.5 kHz	
DTV Receiver	7.8kHz	62.5kHz	SF7	SF12
RX1	-0.1	1.4	1.8	0.3
RX4	0.9	1.4	0.8	0.3
RX5	0.1	1.2	1.5	0.4
RX10	0.1	8.1	8.0	0.0
RX12	0.5	2.5	3.1	0.2

Table 4: Change in the D/U Ratio of DTV Receivers for the DTV Transmitter Operating on Channel 9

DTV center frequency = 485MHz (Channel 16)				
	Change in D/U when SF is increased from SF7 to SF12		Change in D/U when b/w is increased from 7.8kHz to 62.5 kHz	
DTV Receiver	7.8kHz	62.5kHz	SF7	SF12
RX1	-0.2	1.7	1.9	0.0
RX4	0.6	1.0	0.6	0.2
RX5	0.1	0.7	1.0	0.2
RX10	0.1	4.7	5.0	0.2
RX12	0.3	1.1	1.3	0.0

Table 5: Change in the D/U Ratio of DTV Receivers for the DTV Transmitter Operating on Channel 16

In general, the NB WSD operating at 62.5 kHz bandwidth with a spreading factor of SF12 displays higher impact on DTVR operation compared to other configurations. For the traditional ATSC 1.0 DTV receivers (RX1, RX4, and RX5), there is negligible change in the D/U ratio for the NB WSD operating at bandwidths of 7.8 kHz (at spreading factors SF7 and SF12) and 62.5 kHz (at spreading factors SF7 and SF12), for measurements in both the UHF and high-VHF bands. For the low-cost digital-to-digital converter (RX10), there is significant change in the D/U ratio with the 62.5 kHz bandwidth and SF12, with a greater change observed in the high-VHF band than the UHF band. For the ATSC 3.0 receiver, there was a $2-3$ dB change in the D/U ratio observed in the high-VHF frequency band for the 62.5 kHz bandwidth and SF12. The change in the UHF frequency band was minimal: $1.1-1.3$ dB.

B MAC Protocol

B.1 Handling collision in quantum selection

If two assignments – A_1 and A_2 – of a scheduling quantum are colliding, we can make these collision free by ensuring that one of these keeps silence in the colliding hour. For example, if A_2 keeps silence, then we represent collision free form of it as $A_2{<}h_{s2}, p_2, Silent[{<}h_{s1}, p_1{>}]{>}$. Inside $Silent[]$, we can keep all the assignments colliding with A_2. In this way, the new assignment is made collision free with respect to the existing assignments of a quantum. Now, the question arises, how to measure the effect of silencing? Here, we introduce a new metric, frequency of silence (ω_q), representing how frequently the new assignment requires to be silent. We can get the value of ω_q from the generic form of solution to Diophantine equation. In the aforementioned example, ω_q for A_2 is quotient of p_1 divided by $gcd(p_1, p_2)$. If $\omega_q = 1$, then the new assignment requires to be silent in every occurrence of it to avoid collision. Consequently, the quantum is of no use for the new assignment. Based on it, if the quantum has existing assignments (mutually non-colliding by definition) that collide with the new one, we use $\omega_{q\%} = \frac{number\,of\,assignments\,impelling\,\omega_q frequency\,of\,silence}{\omega}$ for quantum selection. If $\omega_{q\%} \geq 1$, then the new assignment requires to be silent in every occurrence of it. Note that the value of $\omega_{q\%}$ is clamped at 1 for any further calculation.

B.2 Construction of channel occupancy table

In Figure 4, A_{c0} is the first assignment for the channel with the occupancy of $\tau = 6$. The first table in the figure represents the entry of A_{c0}. The first entry in the table with no assignment is to facilitate future entries. A_{c1} is the next assignment with the occupancy of $\tau = 9$ and does not collide with A_{c0}. Now, there are 2^2 possible combination of these two assignments as shown in the second table which is basically a binary counter of two digits. However, A_{c0} and A_{c1} do not collide, and thus the last entry in the second table has $\tau = 0$. Consequently, any combination in future having these two assignments will not have a common colliding hour. So we can remove this entry and get the third table. A_{c2} is the next assignment with the occupancy of $\tau = 12$ and collides with both A_{c0} and A_{c1}. Using the method of binary counting we derive the fourth table from the third one. Finally, if the value of τ crosses the limit of $36sec$ in an entry while adding a new assignment, the assignment is not valid for the channel.

B.3 Pseudo code of slot allocation algorithm

In Algorithm 1, we show how possible combinations of slots are prepared. We first get the dictionary of collision free scheduling quanta for all base station radios using scheduling quantum selection as mentioned above. The output of

Algorithm 1: Pseudo code: Generate possible combination of slots for slot allocation algorithm

```
Function Get_PossibleSlotCombinations($p$, $\sigma$, $\alpha_{min}$):
  GET $qDictionary$ FROM Quantum_Selection($p$);
  INIT $possibleSlotCmbntns[]$ TO $empty$ ;
  while $qDictionary\, != empty$ do
    POP $entry{<}A, collisionFreeq[]{>}$ FROM
      $qDictionary$;
    for $\alpha = \alpha_{max}$ TO $\alpha_{min}$ do
      if $collisionFreeq\, != empty$ then
        INIT $listOfGrpdq[]$ TO (SELECT * FROM
          (SELECT * FROM $collisionFreeq$ GROUPBY
          ($continuity = \alpha$)) ORDERBY (AVG($\rho_q$));
        INIT $cntGrpdq$ TO Count($listOfGrpdq$);
        INIT $numOfSlotRqrd$ TO $\lceil \frac{\sigma}{\alpha} \rceil$;
        INIT $cntSlotGotChnl$ TO $zero$;
        INIT $slotsWithChnl[]$ TO $empty$;
        for $i = 0$ TO $cntGrpdq - 1$ do
          INIT $A_c$ TO $<A, \tau, listOfGrpdq[i]>$;
          GET $channel$ FROM Channel_Selection ($A_c$);
          if $channel\, != null$ then
            ADD $<listOfGrpdq[i], channel>$ TO
              $slotsWithChnl[]$;
            INCREASE $cntSlotGotChnl$ BY 1;
            if $cntSlotGotChnl == numOfSlotRqrd$ then
              ADD $<slotsWithChnl[], \alpha, Avg(\rho_q), \rho_c, A>$
                TO $possibleSlotCmbntns[]$;
              break;
```

scheduling quantum selection is a dictionary having different possible (with(/out) modification for silencing) assignments ($A{<}h_s, p, Silent[]{>}$) as the key and corresponding list of collision free scheduling quanta as the value. For each entry in the dictionary, we then vary the continuity (α) of the scheduling quantum from max to given min. For each continuity value, we group the α consecutive scheduling quanta. Groups are ordered by the average ρ_q of member scheduling quanta in each group. Here, to serve the requirement of σ scheduling quanta, we take $\lceil \frac{\sigma}{\alpha} \rceil$ groups of scheduling quantum. Next, we select downlink channel for each group using the channel selection algorithm. Now, each group is an individual slot, and the combination of $\lceil \frac{\sigma}{\alpha} \rceil$ slots together satisfy the requirement from the client. In this way, we make a list of the possible combinations of slots. Next, to select a combination from the list, we use the above mentioned weight function.

B.4 Cases where p is out of boundary

Now, what if the traffic has a periodicity of p_{low} seconds that is less than 1hr? We convert it as $\lceil \frac{p_{low}}{3600} \rceil$ requirements having a periodicity of 1hr. During the slot allocation, we select slot combinations for each requirement having a time gap of no more than p_{low} seconds from the selected slot combinations for earlier requirement. If a client generates periodic traffic with $p > 24hr$, it freshly joins the network before data transmission. Because as mentioned in Section 5.4, a static client with periodic traffic requires to poll channel status from WSDB once in every 24 hours with the recent GPS reading. The poll request is piggybacked as a MAC command in a confirmed up data frame. As a client generating traffic at a periodicity of more than 24hr, it does not send any data frame within 24hr of the pervious one where the poll request can be piggybacked. Besides, sending a frame just for channel polling would be power inefficient. Thus, it it freshly joins the network every time when it has data frame to transmit. It in turn saves the power of polling channel in every 24 hours.

B.5 Client bootstrapping

B.5.1 Beacon

In one hour, there are N_{bcnprd} pre-specified beacon periods, each having $N_{bcnslot}$ of the same length. Every beacon slot has identical periodicity of p_{bcn}. For example, in Figure 5, each beacon period has $N_{bcnslot} = 2$ beacon slots with a periodicity of $p_{bcn} = 1hr$. A NB channel for downlink transmission is associated with each beacon slot. These NB beacon channels are distinctly picked from the TV channels registered for the gateway which increases robustness against noisy link and interference. The beacon slot structure along with channels are pre-specified and pre-loaded in the client radio. Here, the question arises what would happen if the WSDB in future ceases transmission on a particular beacon channel for the gateway? Although it is very infrequent, it needs to be addressed to comply with the regulation. In such a case, the gateway goes silent in the beacon slot associated with ceased channel. However, the clients continue to listen on that channel without violating the regulation. Finally, N_{bcnprd}, $N_{bcnslot}$, and p_{bcn} are adjusted based on application and expected traffic pattern. For example, for a deployment expecting high event-driven traffic, N_{bcnprd} is set to a higher value with lower p_{bcn}.

B.5.2 Join

On reboot, a client radio first locks the GPS and retrieves the current location and time with a precision of one second. It then hops across the beacon channels according to the previously specified beacon slot structure and listens for the beacon. Since the client is already time synced using GPS and knows beacon schedule, it requires minimal hopping depending on the channel quality. Each beacon frame embeds join slot information as MAC command in an encrypted LoRaWAN multicast frame. To be specific, each beacon frame contains the scheduling info of N_{jnslot} join slots between the current and next beacon period. Hence, the info received in a beacon is only valid till the next beacon period. For example, in Figure 5, each beacon contains the scheduling info of following $N_{jnslot} = 3$ join slots. Here, the gateway books the join slots as an event-driven traffic. The benefit is twofold. First, N_{jnslot} can be adjusted depending on the expected join requests in a deployment over the time. Second, it utilizes no or low-utilized channels in that hour assigned for periodic assignments. It in turn impels variation in the channels used in the join slots with the time which increases robustness against noisy channels and long-term interference. Note that the uplink and downlink channels are same in the join slot.

Upon receiving the beacon, a client selects one of the join slots from the info embedded in the beacon. Since multiple clients may attempt to send join request at the same time, we need to ensure that clients are selecting join slots in a distributed manner to reduce the collision. Here, the client leverages dynamic quadratic hash function where the unique device id is used as the key. The gateway decides the coefficients of the hash function equation. To do so, it estimates the set of clients likely to send join request using set the difference between the provisioned clients for the deployment and clients already joined the network for periodic traffic. The coefficients chosen based on this estimated set are embedded in the beacon. For example, in Figure 5, $Client_0$ and $Client_1$ select two different join slots.

In the join slot, a client transmits the join request along with its location and slot requirement for the data communication. Upon receiving the join request, the gateway sends the client's location to the channel coordinator for the registration on the WSDB (see Figure 2). Once the registration of the client is done, the Whisper MAC manager gets the available channel for it and allocates the slots for data communication. The gateway then sends a join response incorporating the allocated slots and expiry of the associated channels.