IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 21, 2021, accepted March 30, 2021, date of publication April 5, 2021, date of current version April 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070976

IoT Network Design Using Open-Source LoRa Coverage Emulator

BASSEL AL HOMSSI, (Member, IEEE), KOSTA DAKIC, (Graduate Student Member, IEEE), SIMON MASELLI, HANS WOLF, SITHAMPARANATHAN KANDEEPAN, (Senior Member, IEEE), AND AKRAM AL-HOURANI, (Senior Member, IEEE)

School of Engineering, RMIT University, Melbourne, VIC 3000, Australia

Corresponding authors: Bassel Al Homssi (bhomssi@ieee.org) and Akram Al-Hourani (akram.hourani@rmit.edu.au)

This research has been partially funded by the Smart Cities and Suburbs Program, Northern Melbourne Smart Cities Network Enabling Data to Drive Change.

ABSTRACT Many wireless Internet-of-Things applications require extended battery life ranging from a few months to a few years. Such applications have motivated the recent developments in low power wide area networks, including the rise of Long Range (LoRa) technology. LoRa has a simple modulation scheme designed for extended converge, low battery consumption, and resistance to high interference levels. Thus LoRa is primarily targeted for shared spectrum applications where interference levels are typically higher than controlled spectrum applications where a single operator usually has a dominant control on the quality of service. As a result, it is of paramount importance to carefully design IoT networks while taking into account the impending impacts of interference and propagation environments. This paper presents a novel LoRa network design framework that utilizes a developed open-source emulator to provide a reliable network coverage estimation. The framework is tested in one of the largest open-access IoT network designs in Australia, which enabled the deployment of 294 sensors and 48 gateways. Both the framework and the emulator are implemented using MATLAB scripting, enabling integration with built-in and external radio planning tools. The framework leverages real interference measurements captured using software defined radio that records the spectrotemporal behavior of the existing traffic in the shared band.

INDEX TERMS LoRa, chirp spread spectrum, software-defined radio, IoT, unlicensed spectrum, emulator, radio network design, interference analysis.

I. INTRODUCTION

Recent estimations predict that more than 50 billion Internet of Things (IoT) devices will join the global network by 2022 [1]. This exponential increase results from emerging business applications that are continuously expanding, such as smart cities, smart agriculture, smart farming, utility metering, among many others [2]. Many applications utilize IoT sensors that are solely reliant on battery power to enable their remote deployment. In a bid to cope with this trend, a class of technologies grouped under the name *low power wide area networks* (LPWAN) have emerged [3]. These technologies aim to attain extended communication ranges with low power consumption and low bit-rates. A prominent LPWAN technology that relies on chirp spread spectrum (CSS) is the Long Range (LoRa) modulation adopted by *LoRa Alliance* [4]. LoRa uses a linear modulated chirp pulse to spread the data waveform over the communication frequency channel. This spreading makes the transmission more resilient in adverse shared spectrum environments where uncoordinated interference from other devices and technologies is substantially high [5].

Spectrum authorities regulate the access to the radio spectrum to enable the coexistence of different wireless technologies [6]. Wireless IoT access technologies are typically designed to operate in either; (i) licensed spectrum bands, e.g., leased by Telco providers, or (ii) license-free spectrum, e.g., conventional Wi-Fi, Zigbee, and Bluetooth. Technologies that operate in the licensed spectrum are subject to controlled levels of interference within their dedicated frequency bands. However, network operators are expected to pay a license fee in order to utilize these bands. Alternatively, technologies operating in the unlicensed spectrum require minimum or no licensing and therefore share the spectrum with many other technologies making the interference levels more significant than the licensed frequency bands. LoRa is

The associate editor coordinating the review of this manuscript and approving it for publication was Tie Qiu.

usually operated in the Industrial Scientific and Medical band (ISM-band), which is part of the unlicensed spectrum. The ISM-band is not exclusive to IoT applications but is accessible by many other technologies, and therefore is susceptible to higher levels of interference [7]. In order to provide a fairer sharing opportunity, many restrictions are imposed on the devices operating in the ISM-band by spectrum regulators, such as restrictions on EIRP envelopes, caps on the limited number of frequency hops, and hopping rate, and limitations on the certain duty cycle of the transmissions [6]. Despite these regulations, ISM-bands are still getting increasingly congested due to the growing traffic generated by the deployment of new devices [7]. If the resultant interference is not managed correctly and accounted for, future wireless IoT services will be adversely impacted. For instance, the coverage of future LPWAN technologies operating in the ISM-band, such as LoRa, will inevitably suffer. This is only exacerbated when these technologies are deployed in urban environments, where the transmissions encounter significant path-loss due to the density of buildings and structures in these environments [8]. Therefore, new IoT network deployments require a careful assessment of incumbent technologies' existing interference levels.

Traditional mobile operators typically deploy network design and planning processes given an allocated spectrum band. However, this is not necessarily the case for emerging IoT networks where a startup business can establish a local council or even by the end-users themselves [9]. Hence, it is crucial that network design is undertaken with utmost attention to the existing spectrum usage. Such unique impairments require different network design approaches from typical licensed-spectrum wireless services, where the spectrum usage is tightly controlled [6]. This paper presents a novel framework that targets the wireless coverage design of IoT networks operating using LoRa technology. At the heart of the framework is an open-source Matlab LoRa emulator [10], capable of accurately simulating an actual LoRa physical layer, is implemented. The emulator consists of a transmitter and receiver block that adopt the LoRa modulation and coding scheme equivalent to a real LoRa transceiver. Furthermore, the emulator also accounts for the channel effects by simulating the radio channel propagation via ray-tracing. The emulator allows the users to inject actual interference measurements recorded via a software defined radio (SDR). The design framework utilizes this emulator to appraise the performance of a LoRa physical link and to provide the expected network-level coverage and data rate. In order to validate the proposed framework, it was implemented in a practical setting as part of a large-scale IoT deployment, incorporating 48 gateways and 297 wireless IoT sensors [11]. The project won the 2020 IoT Alliance Australia award and 2020 Municipal Association of Victoria award.

In this respect, the contributions of this paper are summarized as follows,

- It presents a novel open-source emulator design for the modulation and coding scheme deployed by LoRa as per the physical layer utilized in typical LoRa communication modules.
- It provides an LPWAN network design framework that accounts for the unique features in the shared spectrum bands for IoT applications.
- It develops a methodology for incorporating real radio spectrum measurements in order to account for the temporal behavior of the shared spectrum interference.
- The framework allows the integration with built-in ray-tracing, thus enabling a high-fidelity replica of the propagation environment.

II. BACKGROUND AND RELATED WORK

IoT access points or gateways provide *access* to IoT end-devices located within their coverage. The locations and heights of these gateways, the spectrum interference, and antennas govern the provided coverage area for a given propagation environment. Network design and planning rely on coverage modeling to predict the optimal physical locations for these gateways in order to meet the network's connectivity and capacity requirements. The coverage obtained relies on the radio environment of the network, captured via propagation modeling which is obtained through empirical measurements, ray-tracing simulations, or a hybrid of both. Many empirical models are derived based on real measurements in specific rural [12], suburban [13], and urban [14] density propagation environments for wireless networks. These models are subsequently generalized to provide an estimate for similar density propagation environments based on their distinct parameters [8]. While empirical models provide simple propagation estimations, they fail to capture the environment's unique features. Alternatively, ray-tracing approximates the propagation into straight lines similar to optical rays. This relies on the unique geometric features of the propagation environment, hence, providing a more specific model that effectively captures the distinct geographic features. However, implementing ray-tracing for large-scale networks can be computationally exhaustive. Hybrid propagation modeling combines both empirical models and ray-tracing to achieve a middle point of complexity and reliability.

To predict the performance of a digital communication link, packet error rate (PER) curves are typically obtained. The simplest PER curve for LoRa is that for a channel with additive white Gaussian noise (AWGN). Threshold signal-to-noise ratios (SNR) are provided by LoRa transceiver supplier Semtech in their receiver data-sheet [15]. Moreover, many efforts were invested in deriving the entire curve through measurements [5], empirical representations [5], [16], [17], and closed-form estimations assuming an AWGN or fading channels [18], providing a benchmark for researchers. Nevertheless, these models do not account for the effect of interference. Therefore, other tools such as stochastic geometry were utilized to derive mathematical models to predict the performance of LoRa, while assuming the interference is solely generated from the network [16], [19]–[22]. Wireless network design platforms, typically deployed in mobile broadband, are also migrating to include IoT networks and rely on ray-tracing or hybrid techniques to obtain the error curves through extensive simulations [21]. All the aforementioned models assume that interference behaves similarly to shot-noise, and the average interference power is linearly summed to the average noise power in order to obtain the signal-to-noise-and-interference-ratio (SINR) as an equivalent to SNR [19]–[24]. While these techniques may suffice for mobile broadband or licensed IoT technologies, the same does not apply to shared spectrum IoT technologies such as LoRa due to the significant variations in interference levels that exist in the spectrum. The interference present in the IoT spectrum has been shown to have a temporal correlation, rendering its average insufficient [25] and often an inaccurate representation of a realistic deployment [9].

Other open-source implementations utilized LoRa's modulation and coding scheme for GNUradio applications to enable SDRs to replace more expensive LoRa receivers [26], [27]. Nevertheless, these implementations do not provide a platform for network design purposes. As a result, this paper presents a LoRa network design framework that utilizes an open-source emulator built using Matlab scripting. The emulator simulates LoRa signals based on LoRa's modulation and coding scheme. Moreover, the emulator utilizes ray-tracing to incorporate the terrain and building footprints, to more accurately estimate the radio propagation of the signals. Furthermore, the emulator utilizes measurements taken from an SDR to incorporate the spectrum interference. Finally, the emulator is also capable of demodulating and decoding the signals, in a similar manner to a typical LoRa receiver. Ultimately, we are capable of simulating an entire traffic profile that takes into account the effect of the coexistence of multiple LoRa transmitters and receivers in order to evaluate the performance of the network. Thus, the framework provides a more reliable and comprehensive network design tool to attain an accurate representation of the performance, prior to physical deployment.

III. LoRa EMULATOR

The constructed LoRa emulator is divided into two main components: (i) LoRa emulator transmitter (LE-Tx), which is capable of encoding and modulating the data into a LoRa frame waveform, and (ii) LoRa emulator receiver (LE-Rx), which is responsible for the demodulation and decoding of the frame to extract the payload data.

A. EMULATOR MODULATION

The symbol is encoded into a cyclic chirp signal, where the starting position of the chirp (the shift) encodes the value of the symbol. The chirp rate can be controlled by a parameter called the spreading factor (SF), where higher SF means a longer chirp which enables the encoding of more bits per symbol. The LoRa chirp is designed to represent $M = 2^{\mathrm{SF}}$ symbol values, where the chirp (symbol) duration is given by $T_\mathrm{s} = M/B$, and B is the utilized bandwidth. A LoRa modulated waveform is therefore analytically represented as follows,

$$s(t) = \exp\left(j2\pi \int_0^t \left[(\beta x + \gamma_n)_{\mathrm{mod}_B} - \frac{B}{2}\right] \mathrm{d}x\right), \qquad (1)$$

where γ_n is the frequency shift obtained as follows,

$$\gamma_n = m_n \Delta_\mathrm{f} = \frac{m_n}{T_\mathrm{s}}, \qquad (2)$$

where m_n is the data symbol value $m_n \in \{0,1,\ldots,M-1\}$, n is the symbol's index, and Δ_f represents the frequency step between the shifts. The frequency step in LoRa's case is selected to be equal to the symbol rate itself B/M. Moreover, β represents the chirp slope which is a function of the bandwidth and symbol time as follows,

$$\beta = \frac{f_\mathrm{high} - f_\mathrm{low}}{T_\mathrm{s}} = \frac{B}{T_\mathrm{s}}, \qquad (3)$$

where f_low and f_high are the lower and upper frequency bounds of the chirp respectively. A representation of two LoRa symbols is illustrated in Fig. 1, showing the advantage of deploying LoRa modulation that spreads the signal energy over larger channel bandwidth, thus enhancing its resilience against narrow-band interference. When selecting a higher SF, the receiver sensitivity is enhanced due to increased energy per symbol. However, this comes at the cost of slower data rate and higher energy consumption [5]. A LoRa frame is composed of a train of symbols as follows, $x(t) = \sum_{n=1}^{N_\mathrm{m}} s_n(t - nT_\mathrm{s})$ where N_m is the total number of symbols per frame.

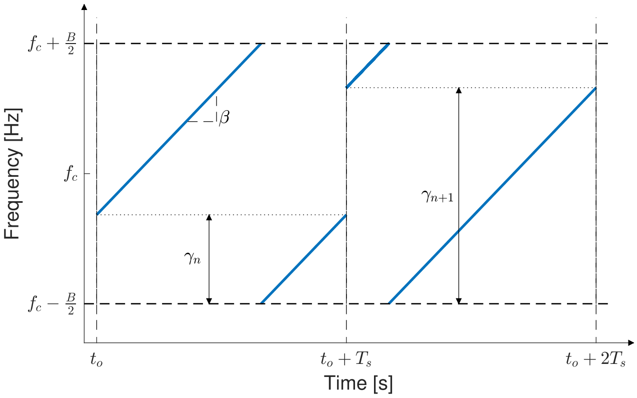

FIGURE 1. An illustration of the spectrogram of an upchirp LoRa modulated symbol.

B. EMULATOR DEMODULATION

The proposed LoRa receiver emulator utilizes a two-step process where the received signal undergoes dechirping first and then demodulation based on Multiple-Frequency-Shift-Keying (MFSK). In order to perform the dechirping, the received signal is mixed with a train of inverted chirps

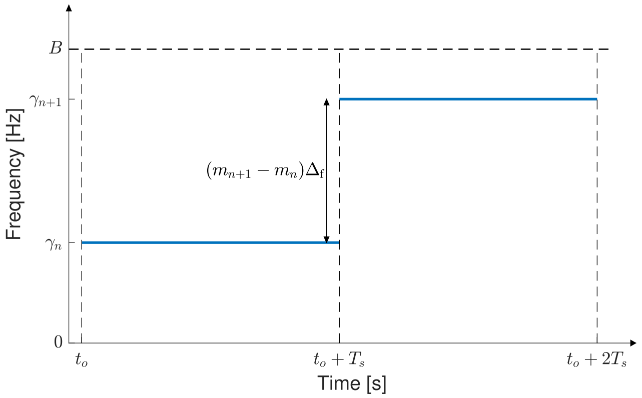

FIGURE 2. A snapshot illustration of the spectrogram of the MFSK signal (dechirped signal of the signal in Fig. 1).

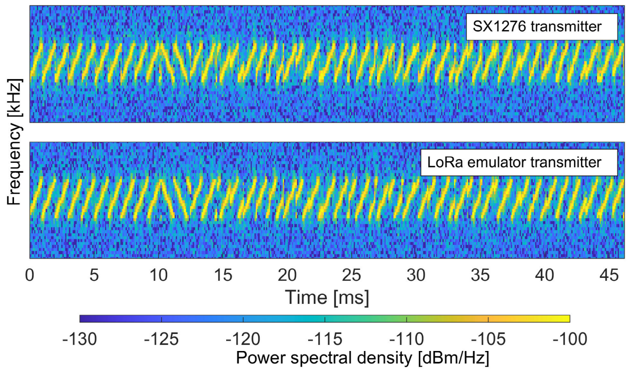

FIGURE 3. Transmission of a LoRa frame with a 13 byte payload, generated from an actual SX1276 chipset (top) versus the proposed LE-Tx (bottom) wth an identical SNR of 20 dB.

with no frequency offset. The dechirping signal is represented as follows,

$$s^{\star}(t)=\sum_{n=1}^{N_{\mathrm{m}}}\exp\left(j\pi Bt-j2\pi\beta t^{2}\right)\circledast\delta(t-nT_s). \tag{4}$$

The dechirping signal is then mixed with the received signal as follows,

$$\begin{aligned} y(t) &= [g\{x(t)\}+i(t)+n(t)]\, s^{\star}(t) \\ &= r(t)+\widetilde{i}(t)+\widetilde{n}(t), \end{aligned} \tag{5}$$

where $g\{.\}$ represents the channel function, $i(t)$ is the radio spectrum interference, and $n(t)$ is the additive white Gaussian noise. While both the interference and noise are multiplied with the dechirping signal, they still maintain their distribution and are represented here as $\widetilde{i}(t)$ and $\widetilde{n}(t)$ respectively. In an ideal channel, the channel function has no effect on the received signal where $g\{x(t)\}=x(t)$ and the outcome of the mixing is an MFSK signal with a frequency offset equivalent to the encoded symbol. The received signal simplifies to,

$$r(t)=\sum_{n=1}^{N_m}\exp\left(j2\pi\gamma_n t\right)\circledast\delta(t-nT_s), \tag{6}$$

where γ_n is the received LoRa waveform frequency offset. Fig. 2 illustrates the spectrogram of the dechirped signal which looks identical to the spectrogram of an MFSK signal. The chirp slope of both the dechirping signal and received LoRa signal needs to be identically opposite in order to achieve accurate dechirping. Thus, symbol synchronization is crucial for ensuring the accuracy of the dechirping. In some applications, these two parameters are pre-configured at the receiver. Also, it is essential to identify the SF and B of the received LoRa signal prior to the dechirping process. Once the signal is dechirped, the obtained MFSK signal can be demodulated using the well-known coherent or non-coherent demodulators [28]. While conventional hardware receivers utilize non-coherent demodulation, the proposed open-source LE-Rx allows both coherent and non-coherent MFSK signal demodulation. This is due to the higher computational power available at the hosting machines when compared to simple IoT gateways that usually rely on the simpler non-coherent detectors.

In the proposed emulator, the non-coherent demodulation is attained by implementing energy detection on the MFSK signal using fast Fourier transform (FFT), where the dominant component of the power spectral density (PSD), i.e. its peak, is extracted. This dominant frequency component represents the original shift that encodes the value of the symbol. Accordingly, the LoRa symbol is retrieved as follows,

$$\hat{m}=\left\lfloor \frac{1}{\Delta_{\mathrm{f}}}\operatorname*{argmax}_{f}[Y(f)]\right\rceil, \tag{7}$$

where $Y(f)$ is the FFT transform, $Y(f)=\mathrm{FFT}\{y(t)\}$, $\lfloor.\rceil$ is the rounding function representing the hard-decision process and $\hat{m}\in\{0,1,\ldots,M-1\}$ is the retrieved symbol value.

Alternatively, LE-Rx can also deploy coherent demodulation which relies on identifying the frequency shift by deploying the maximum likelihood. In other words, the frequency shift waveform that maximizes the convolution magnitude represents the estimated symbol as follows,

$$\hat{m}=\operatorname*{argmax}_{k}\left[\int_{0}^{\infty}y(t)z_{\mathrm{k}}(t-\tau)d\tau\right], \tag{8}$$

where $z_k(t)$ is the MFSK modulated signal with frequency shift $k\Delta_{\mathrm{f}}$, represented as $z_{\mathrm{k}}(t)=\exp\left(j2\pi k\Delta_{\mathrm{f}}t\right)$ and k is an integer representing all symbol possibilities $k=\{0,1,\ldots,M-1\}$.

C. EMULATOR CODING AND DECODING

The LE-Tx generates a frame identical to an actual LoRa transmitter as shown in Fig. 3. The figure illustrates a comparison between (i) the spectrogram of an actual frame emitted by a LoRa SX1276 [15] chipset, and (ii) the emulated transmission generated from the LoRa emulator proposed in this paper. We note from the figure that both transmissions are identical, demonstrating the ability of the proposed LE-Tx to simulate an actual transmission. The LE-Rx can demodulate

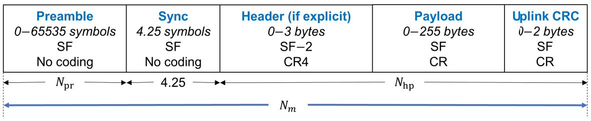

FIGURE 4. An illustration of the LoRa frame, including the length, spreading factor, and coding rate used to encode the symbols.

and decode both actual and emulated LoRa frames. The generated frame consists of three parts; the preamble, header, and payload [29], summarized in Fig. 4. The payload and header undergo coding in order to protect the information they hold, whereas the preamble does not.

1) PREAMBLE

The preamble represents padded symbols that do not correspond to any information but are utilized for synchronization purposes. Note that the synchronization between the transmitter and receiver is necessary to dechirp and demodulate the signal accurately.

2) PAYLOAD AND HEADER

The payload is where the message is encoded and can be up to 255 bytes in length [29]. The payload is followed by an uplink cyclic redundancy check, which is only available in the uplink frame and is up to 2 bytes in length. The header is used to incorporate information about the payload length in bytes N_b, the payload coding rate, whether the payload cyclic redundancy check exists, and the header cyclic redundancy check.

The header is always 3 bytes in length and exists when the transmission is in the explicit mode. The total number of symbols encoded in the header and payload combined depends on the coding rate, cyclic check flag, implicit header flag, and optimization control and is given by [15],

$$N_\mathrm{hp} = 8 + \max\left\{\left\lceil \frac{8N_b - 4\mathrm{SF} + 68 + 16\mathrm{CRC} - 20\mathrm{H}}{4\mathrm{SF} - 8\mathrm{OP}} \right\rceil \times (4 + \mathrm{CR}), 0\right\}, \qquad (9)$$

where CR is the LoRa payload coding rate parameter, CRC is the cyclic rate code flag; and is 1 when the cyclic code rate is utilized whereas 0 if not. $\mathrm{H} = \{0, 1\}$ is the implicit header flag where it is 0 when the header is incorporated in the frame and 1 when it is not. Finally $\mathrm{OP} = \{0, 1\}$ is the optimization mode flag, where it is 1 when the adaptive rate is active and 0 when it is off. The header and payload are encoded prior to modulation. The encoding process is carried out in the following four main steps:

1) *Hamming encoding:* LoRa uses a 4/(4+CR) hamming code where the data word length is 4 and is converted into $4 + \mathrm{CR}$ code word length. The coding rate in LoRa is hence restricted to $\mathrm{CR} = \{1,2,3,4\}$ [15], where a code-word length of 5 and 6 includes one bit for error detection only, whereas a code-word length of 7 or 8 includes one bit for error detection and another bit for error correction. The payload can be encoded using any of the aforementioned combinations of hamming codes, whereas the header is always encoded using a hamming code of (4,8). This is to ensure both error detection and correction of the header, which carries crucial information about the payload. Moreover, a fixed coding rate ensures that the receiver can always decode the header without the need for preliminary information about its coding rate. Once the header is decoded, the coding rate parameter of the payload (CR) is extracted from the header's second byte.
2) *Whitening:* LoRa deploys whitening to reduce the correlation of the redundancy bits added by the channel encoder. Whitening in LoRa is achieved by adding a known sequence to the payload only, and the sequence is dependent on the payload CR parameter. Note that no whitening is deployed to the header [15], [27], [29].
3) *Interleaving:* LoRa implements a (k×r) interleaver that imposes transposition and digit shift on the symbols. This means that every k code-words are transposed and its digits shifted to the left by $r_{\mathrm{mod\ SF}}$ digits, where r is the row number. The header is interleaved by $(\mathrm{SF} - 2 \times 8)$ and the payload by $(\mathrm{SF} \times 4 + \mathrm{CR})$ [15], [29].
4) *Graying:* LoRa implements gray coding on the interleaved symbols to reduce errors of adjacent bits [15], [27].

Note that the encoding and decoding are one-to-one inverted operations.

A typical LoRa frame composes of a preamble of $(N_\mathrm{pr} + 2)$ up-chirps, followed by 2.25 down-chirps employed in order for a LoRa receiver to identify the start of the header/payload symbols. The total number of symbols per frame as per the LoRa PHY standard is obtained as follows,

$$N_m = N_\mathrm{hp} + N_\mathrm{pr} + 4.25. \qquad (10)$$

Accordingly, the total time-on-air of the frame is therefore, $T_\mathrm{m} = N_\mathrm{m} T_\mathrm{s}$. A block diagram for the LE-Tx and LE-Rx are illustrated in Fig. 5.

IV. CHANNEL AND INTERFERENCE EMULATOR

As part of the developed LoRa emulator, we provide an integrated channel and interference emulator such that the generated LoRa frames are subjected to realistic radio channel impairments. This is essential to replicate practical radio environments in order to get accurate prediction results within the proposed design framework. A zoomed-in block diagram explaining the proposed radio channel emulator is illustrated in Fig. 6, many of these blocks utilize the built-in Matlab propagation tools and thus are well-tested and validated. Further to the typical AWGN and fading impairments, the developed emulator has two important additions: (i) the network designer can inject realistic interference measurements collected using an SDR, and (ii) ray-tracing simulation

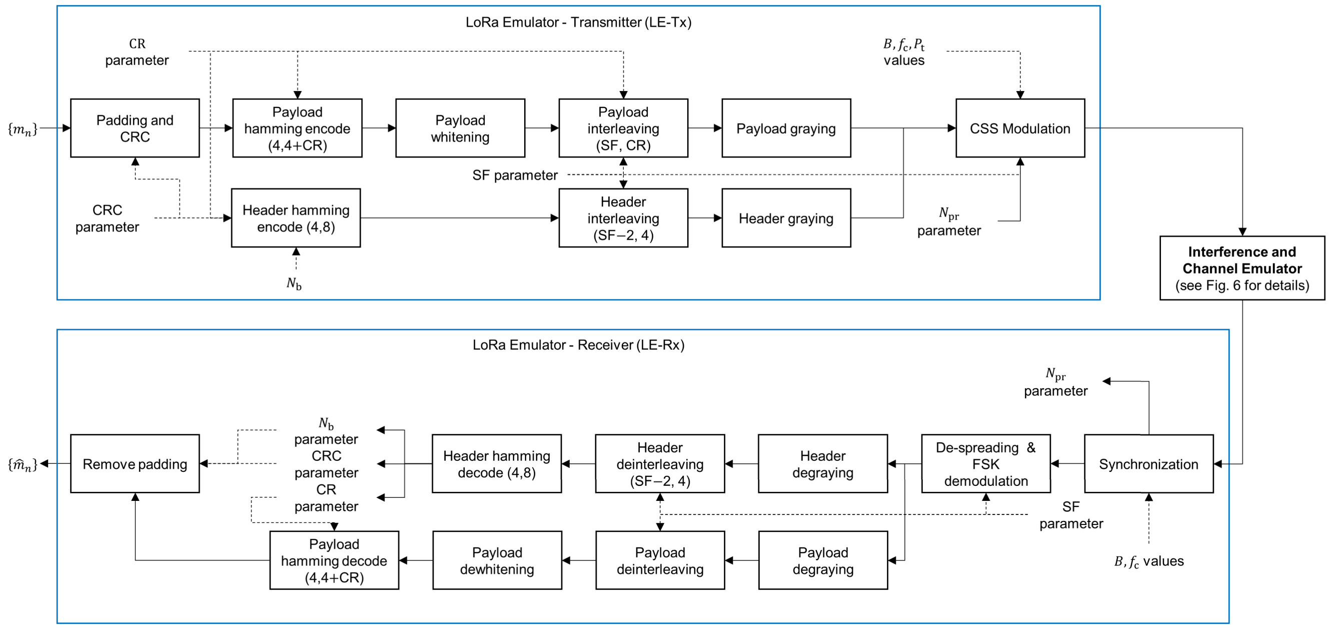

FIGURE 5. LoRa modulation and coding scheme block diagram for both the transmitter and the receiver. The header and payload follow small variations in operations and are then combined before modulation in the transmitter and are similarly separated after demodulation in the receiver.

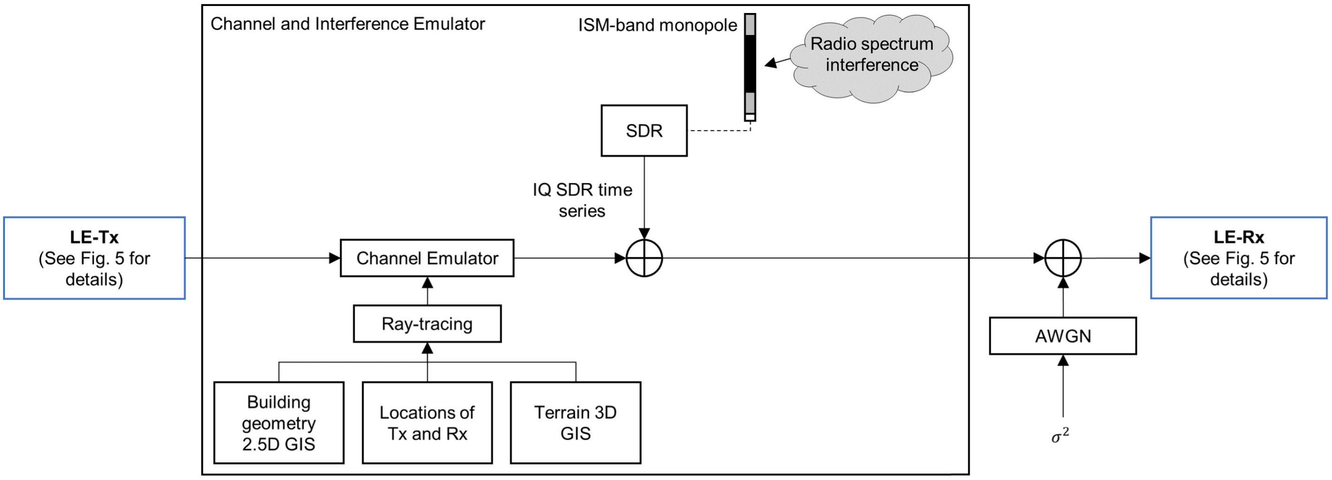
Channel and Interference Emulator
ISM-band monopole
Radio spectrum interference
SDR
IQ SDR time series
LE-Tx (See Fig. 5 for details)
Channel Emulator
Ray-tracing
Building geometry 2.5D GIS
Locations of Tx and Rx
Terrain 3D GIS
AWGN
σ^2
LE-Rx (See Fig. 5 for details)

FIGURE 6. Block diagram of LoRa network design framework proposed in this paper to estimate LoRa's performance in a real practical environment prior to deployment.

is deployed to accurately estimate the path-loss and shadowing effects.

A. INTERFERENCE EMULATOR

One of the significant channel impairments in the unlicensed spectrum is interference. This is because various technologies that are not necessarily limited to LoRa or even IoT compete for these scarce resources. Radio spectrum congestion is shown to correlate with the population density, and thus it is important to quantify it, particularly in urban environments [7]. It has also been shown that the availability of the spectrum varies not only spatially but also temporally [25]. Accordingly, the impact of the interference cannot be simply abstracted into a single parameter such as the average, but rather needs to be statistically modeled or could be regenerated based on empirical measurements. To incorporate the spatio-temporal interference behavior, the proposed framework allows for the injection of real spectrum measurements as pre-recorded by an SDR. As such, an SDR can be configured to capture the radio spectrum near the expected

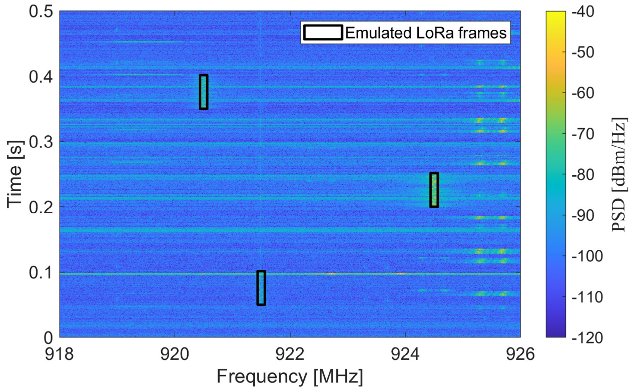

FIGURE 7. Injected emulated LoRa transmissions on top of the collected real spectrogram measurements in the ISM-band radio spectrum.

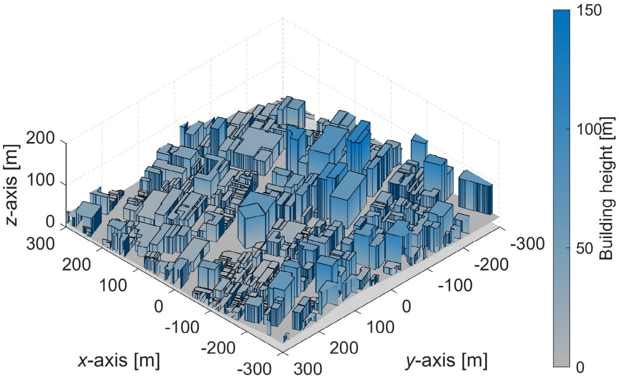

FIGURE 8. Geometric representation of buildings in Melbourne CBD in 2.5D. Geodata source is from the Australian Urban Research Infrastructure Network database [32].

location of the gateway. The emulated LoRa frames then are linearly summed with the collected measurements as indicated in (5). In Fig. 7 we illustrate a snapshot of the recorded power spectral density (PSD) in a dense urban environment, where the emulated LoRa frames are linearly superimposed to the measurements. The resulting signal is then passed to the LE-Rx in order to demodulate and decode the LoRa signal. If there is a significant interference variability across the day, it is important to collect interference measurements and sample this variability at different periods of time and then run the emulator over these periods. On that account, the emulator is capable of providing not only the average performance but also the variations over an extended period of time. For instance, in our large-scale network deployment, we have noticed that the PER of some devices was higher at certain hours of the day [9].

B. CHANNEL EMULATOR

Radio signal propagation is naturally affected by the power loss due to electromagnetic wave-front expansion, where the signal in free-space loses power proportionally to the square of the distance. In built-up environments, radio waves encounter the additional losses caused by different obstacles such as buildings, vegetation, and other structures. There are many good radio propagation models that abstract the channel behavior into simplified equations, well-known examples can be found in the WINNER-II model and by 3GPP [30], [31]. These standard models could be used for performance bench-marking, however, for region-specific coverage prediction, it is important to either tune these models or utilize the more computationally-intensive ray-tracing. The integrated channel emulator is capable of incorporating both approaches, providing the designer with the flexibility of choice when balancing the trade-off between computational complexity and accuracy. In order to efficiently utilize ray-tracing, the designer requires access to the terrain elevation map and the footprint of the buildings. An example of ray-tracing input is shown in Fig. 8, for part of Melbourne's Central Business District (CBD) with many high-rise buildings.

V. EMULATOR PERFORMANCE VALIDATION

The presented LoRa emulator is validated via two steps. In the first step, we start validating the performance of the emulator under an AWGN only scenario, by using a physical LoRa module in a controlled lab environment. We connect a LoRa transmitter (SX1276 LoRa module [15]) via a coaxial cable to a variable attenuator in order to control the path-loss. The signal passes through the variable attenuator to a LoRa receiver (another SX1276 LoRa module). Moreover, the attenuator and the receiver are shielded inside a metallic box acting as a small Faraday cage to ensure that leakage of electromagnetic radiation from the transmitter is suppressed. Both the transmitter and receiver are controlled by a computer, where the transmitter generates a known number of packets, and the receiver reports the successful packets to the computer which calculates the corresponding PER. The received signal power is controlled by two means: (i) by pragmatically adjusting the RF transmit power, and (ii) by manually adjusting the variable attenuator. Accordingly, several PER vs. SNR curves are recorded for varying spreading factors. Note that the average noise power level is assumed constant at the receiver.

On the other hand, to obtain the AWGN performance of the emulator, we directly inject the LE-Tx output with an additive complex Gaussian random variable and then feed their sum to the LE-Rx. Accordingly, (5) simplifies to the following for an AWGN channel,

$$y(t) = x(t) + \mathcal{CN}(0, \sigma^2), \tag{11}$$

where $\mathcal{CN}(0, \sigma^2)$ is the complex AWGN noise and σ^2 is the average noise of the receiver channel. Similarly, we obtain the PER curves by calculating the complement of the successful to total packet ratio for different spreading factors. The PER curves of the emulator and actual hardware are compared versus analytical estimations obtained for non-coherent MFSK demodulation [18] along with cutoff SNR values reported in the data-sheet [33]. The obtained emulator PER curves agree with the PER obtained for actual hardware, analytical MFSK demodulation curves, and data-sheet cut-off values, as illustrated in Fig 9.

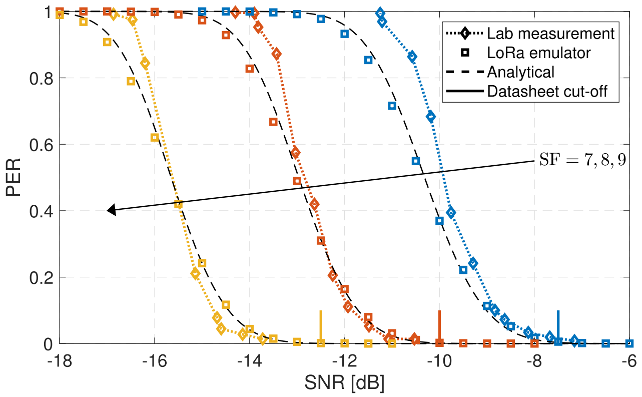

FIGURE 9. PER curve for an AWGN channel for SF7, 8, and 9 for a payload of 17 bytes and bandwidth of 125 kHz.

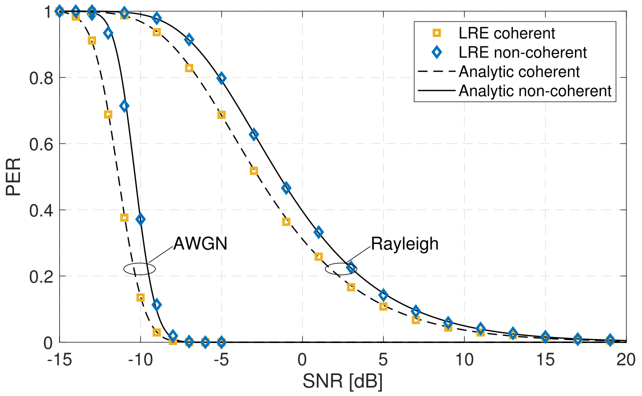

FIGURE 10. Performance of a SF7 LoRa transmission in a Rayleigh channel for a payload of 17 bytes and bandwidth of 125 kHz.

For the second validation step, we test the emulator performance in a multi-path fading setting, particularity a channel with Rayleigh fading. In this case, we only compare the emulator performance with analytical representations that are available in the literature. This is because it is difficult to suppress other factors such as interference and shadowing along with reduced control over path-loss for an actual hardware implementation. Similar to the AWGN simulation, we inject the LE-Tx generated signal into a non-frequency-selective Rayleigh fading channel and into the LE-Rx with additive complex Gaussian noise. Accordingly, (5) simplifies to the following,

$$y(t) = \alpha(t)e^{-j\phi(t)}x(t) + \mathcal{CN}(0, \sigma^2), \tag{12}$$

where the channel is modeled as a zero-mean complex Gaussian random variable with $\alpha(t)$ and $\phi(t)$ being Rayleigh and uniform distributed respectively. The same is repeated utilizing both non-coherent and coherent MFSK demodulation. Similar to AWGN, the performance of coherent and non-coherent MFSK demodulation models are well-documented in the literature [18], [28]. The SF7 PER curves using both the emulator and analytical representation are illustrated in Fig. 10.

VI. USING THE EMULATOR WITHIN THE DESIGN FRAMEWORK

The objective of an LPWAN network design phase is to determine suitable locations for the IoT gateways [9]. The suitability is determined based on two contributing factors, namely, the coverage and the capacity of the gateways. Therefore, a gateway is ideally located where it is capable of serving a cluster of sensors. To obtain the expected coverage of a gateway, we deploy a regular grid of points surrounding the gateway $x \in \mathbb{R}^2$. We utilize the emulator for each grid point and obtain the ratio of the number of successful transmissions N_s to the total number of transmissions N_T. The expected packet success rate (PSR) for each grid point is therefore, $p_s(x) = N_\mathrm{s}/N_\mathrm{T}$. We define a certain PSR cut-off, where a grid point is covered if its PSR is above this threshold and vice-versa if not. Accordingly, the coverage is represented as follows,

$$\mathcal{C}(x) = \begin{cases} 1, & p_\mathrm{s}(x) \geq \eta \\ 0, & p_\mathrm{s}(x) < \eta, \end{cases} \tag{13}$$

where η is the cut-off PSR. Note that the interference injected is measured at the gateway.

The obtained coverage incorporates a more accurate representation of the interference profile and multi-path fading. Nevertheless, other practical factors that affect the coverage must also be taken into account, such as the antenna gains and the actual EIRP of the devices. Therefore, we introduce a tuning parameter that takes into account the discrepancy between the emulated and measured coverage. In order to obtain the tuning parameter, we calculate the accuracy as follows,

$$\mathcal{A} = \frac{1}{2}\sum_{i=0}^{1}\left[\frac{\mathbb{1}_{\mathcal{C}(x)=i \text{ AND } C_\mathrm{p}(x)=i}}{\mathbb{1}_{C_\mathrm{p}(x)=i}}\right], \tag{14}$$

where $C_\mathrm{p}(x)$ represents the measured coverage and $\mathbb{1}$ is the logical operator. Ideally, we expect the emulated and measured coverage to be identical providing an accuracy of 1. However, due to practical impairments that does not occur, therefore, we tune a scalar parameter to obtain the maximum accuracy as follows,

$$\theta^\star = \underset{\theta}{\mathrm{argmax}}[\mathcal{A}], \tag{15}$$

where θ is bounded between [0, 1]. The tuning parameter is a multiplicative factor that scales the obtained path-loss for the grid points. The tuning parameter is constant once it is obtained since it does not vary in magnitude for different propagation environments and only relies on the utilized transmitter and receiver hardware.

VII. PRACTICAL RESULTS AND DISCUSSION

In this section, we explain how we utilized the proposed framework in one of the largest open-access LoRa network deployments in Australia, involving 48 LoRa gateways serving around 294 IoT devices. The network deployment is part of the Northern Melbourne Smart Cities Network project which includes other related aspects such as network management and data analytics [11]. Different IoT sensor types were chosen to fulfill particular use-cases that the local councils were interested in. These use-cases included (i) air quality, (ii) waste management, (iii) water level monitoring, (iv) asset tracking, and (v) people counting in locations where the councils provide services to the community. Furthermore, the aim of the project is to allow the local councils to provide better services based on the collected data from the sensors. The aim is to adopt proactive service approaches based on the projected events and asset utilization. The project spans over a large geographic area in both metropolitan and rural Melbourne. Different propagation environments provide varying signal propagation conditions and significantly different spectrum interference and traffic profiles. The design phase starts with identifying the locations for the IoT sensors based on the use-case scenarios. These sensors are divided into different clusters based on their geographic dispersion. Each cluster of sensors is referred to as a site and requires at least one gateway to *serve* this cluster. Furthermore, several site-surveys and measurement campaigns are required to assist the gateway allocation process. The network design phase and the rest of the deployment phases for the NMSCN are further elaborated in [9]. Once some preliminary gateway locations are proposed, we employ the framework presented in this paper in order to obtain the performance of the communication links, the gateway coverage, and the traffic profile of the network for each site or multiple sites as per the previous section. In the following, we detail the measurement setup used to collect the radio spectrum interference. Then we compare the coverage obtained by the framework to the actual coverage derived from real measurements conducted for the deployed LoRa gateway.

TABLE 1. Measurement setup parameters.

Parameter	Value	Unit
Frequency band	915 - 928	MHz
Antenna gain @921 MHz	6	dBi
SDR sampling rate	20	MHz
SDR measured noise figure	2.5	dB
LoRa gateway measured noise figure	6	dB
SDR average noise σ_s^2@125 kHz	-121	dBm
LoRa gateway average noise σ_g^2@125 kHz	-117	dBm

A. MEASUREMENT SETUP

We utilize a commercially available SDR with a very low noise figure (SDRPlay RSP2 [34]) connected to a 921 MHz monopole antenna as illustrated in Fig. 6. The SDR collects the samples as an I/Q time series based on the selected measurement parameters as summarized in Table 1. It is worth noting that the captured signal incorporates the SDR receiver additive noise which is much lower than a typical LoRa gateway. Thus, we compensate the difference by injecting additional Gaussian white noise in the emulator on top of the collected I/Q samples, more details on this process can be found in [35].

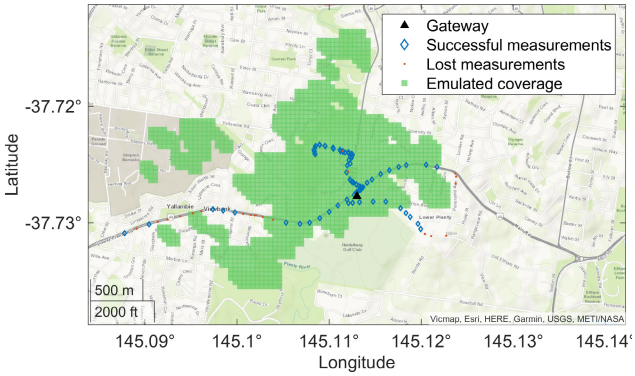

FIGURE 11. Coverage representation with cut-off PER of 0.25 using ray-tracing simulation versus actual site coverage measurements. The dark green represents the coverage obtained from the framework whereas the light green represents the coverage obtained from the SINR representation. All coverage contours are derived based on SF12 for a packet of 17 bytes and bandwidth of 125 kHz.

B. GATEWAY COVERAGE

Using the spectrum measurements obtained from the SDR, we inject the interference signal to the emulator in order to obtain the PER for each grid point. To derive the coverage, we select a QoS threshold level of $\eta = 0.75$, i.e. the region is considered as *covered* if 75% of the radio frames are successfully received. Furthermore, a drive test is conducted around the gateway locations to obtain their measured coverage. A packet tester is developed as part of the project and is configured to continuously transmit LoRaWAN frames. The packet testing is fixed inside a car that is driven on a selected route to account for various distances around the gateway [9]. Fig 11 illustrates the measured coverage and the emulated coverage for SF12, showing a representation of one of the 48 gateways installed in the Northern Melbourne Smart Cities Network.

VIII. CONCLUSION

In this paper, we present a design framework for the deployment of large-scale LoRa networks. At its core, the framework utilizes an open-source LoRa emulator capable of behaving like a typical LoRa transmitter or receiver based on the LoRa modulation and coding scheme. The framework also employs a channel and interference emulator which deploys ray-tracing to accurately depict the effect of multipath fading in urban environments integrated with external spectrum monitoring tools that capture the interference temporal variations. As a result, the framework is capable of incorporating the spontaneous variations in the link performance as opposed to the conventional methods which rely on depicting the interference as an average value only. The framework deploys a tuning methodology to further enhance the accuracy of the emulated coverage by utilizing practical coverage measurements. The proposed framework can be utilized for any propagation environment and can serve to provide network designers with more accurate tools to depict the performance of the network prior to its physical deployment.

REFERENCES

[1] Ericsson. (Nov. 2016). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[3] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, 2nd Quart., 2017.

[4] LoRa Alliance. (2018). *Specification, LoRaWAN*. Accessed: Oct. 5, 2020. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf

[5] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *Proc. Symp. Commun. Veh. Technol. (SCVT)*, Nov. 2016, pp. 1–5.

[6] *Radiocommunications (Low Interference Potential Devices) Class Licence 2015*, Austral. Commun. Media Authority, Sydney, NSW, Australia, 2015.

[7] B. Al Homssi, A. Al-Hourani, R. J. Evans, K. G. Chavez, S. Kandeepan, W. S. T. Rowe, and M. Loney, "Free spectrum for IoT: How much can it take?" in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[8] A. Al-Hourani, "On the probability of line-of-sight in urban environments," *IEEE Wireless Commun. Lett.*, vol. 9, no. 8, pp. 1178–1181, Aug. 2020.

[9] B. A. Homssi, A. A.-Hourani, K. Magowe, J. Delaney, N. Tom, J. Ying, H. Wolf, S. Maselli, S. Kandeepan, K. Wang, and K. M. Gomez, "A framework for the design and deployment of large-scale LPWAN network for smart cities applications," *IEEE Internet Things Mag.*, vol. 4, no. 1, pp. 53–59, Mar. 2021.

[10] B. A. Homssi. (2020). *LoRa MATLAB Emulator Data Retrieved From AURIN*. [Online]. Available: https://au.mathworks.com/matlabcentral/fileexchange/81046-loramatlab

[11] Australian Government, Department of Infrastructure, Regional Development and Cities. *Northern Melbourne Smart Cities Network, Enabling Data to Drive Change*. Accessed: Mar. 1, 2021. [Online]. Available: https://www.infrastructure.gov.au/cities/smart-cities/collaboration-platform/northern-melbourne-smart-cities-network.aspx

[12] G. R. MacCartney and T. S. Rappaport, "Rural macrocell path loss models for millimeter wave wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1663–1677, Jul. 2017.

[13] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 7, pp. 1205–1211, Jul. 1999.

[14] A. Al-Hourani, S. Chandrasekharan, and S. Kandeepan, "Path loss study for millimeter wave device-to-device communications in urban environment," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2014, pp. 102–107.

[15] SEMTECH. (2020). *SX1276/77/78/79: 137 MHz to 1020 MHz Low Power Long Range Transceiver*. Accessed: Jan. 11, 2021. [Online]. Available: https://www.semtech.com/

[16] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.

[17] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017.

[18] T. Elshabrawy and J. Robert, "Closed-form approximation of LoRa modulation BER performance," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1778–1781, Sep. 2018.

[19] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[20] B. Al Homssi, A. Al-Hourani, S. Chandrasekharan, K. M. Gomez, and S. Kandeepan, "On the bound of energy consumption in cellular IoT networks," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 355–364, Jun. 2020.

[21] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan, and M. Gidlund, "Scalability analysis of a LoRa network under imperfect orthogonality," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1425–1436, Mar. 2019.

[22] L. Beltramelli, A. Mahmood, M. Gidlund, P. Österberg, and U. Jennehag, "Interference modelling in a multi-cell LoRa system," in *Proc. 14th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2018, pp. 1–8.

[23] J. M. de Souza Sant'Ana, A. Hoeller, R. D. Souza, H. Alves, and S. Montejo-Sánchez, "LoRa performance analysis with superposed signal decoding," *IEEE Wireless Commun. Lett.*, vol. 9, no. 11, pp. 1865–1868, Nov. 2020.

[24] L. Beltramelli, A. Mahmood, P. Osterberg, and M. Gidlund, "LoRa beyond ALOHA: An investigation of alternative random access protocols," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3544–3554, May 2021.

[25] B. A. Homssi, A. Al-Hourani, Z. Krusevac, and W. S. T. Rowe, "Machine learning framework for sensing and modeling interference in IoT frequency bands," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4461–4471, Mar. 2021.

[26] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, "A multi-channel software decoder for the LoRa modulation scheme," in *Proc. 3rd Int. Conf. Internet Things, Big Data Secur. (IoTBDS)*, vol. 1. Setúbal, Portugal: SciTePress, 2018, pp. 41–51.

[27] M. Knight and B. Seeber, "Decoding LoRa: Realizing a modern LPWAN with SDR," in *Proc. GNU Radio Conf.*, 2016, vol. 1, no. 1, pp. 1–5.

[28] M. K. Simon and M.-S. Alouini, *Digital Communication Over Fading Channels*, vol. 95. Hoboken, NJ, USA: Wiley, 2005.

[29] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," *ACM Trans. Sensor Netw.*, vol. 15, no. 2, pp. 1–35, Apr. 2019, doi: 10.1145/3293534.

[30] P. Kyosti, J. Meinila, L. Hentila, X. Zhao, T. Jamsa, C. Schneider, M. Narandzic, M. Milojevic, A. Hong, J. Ylitalo, V. Holappa, M. Alatossava, R. Bultitude, Y. Jong, and T. Rautiainen. (Apr. 2008). WINNER II channel models. Information Society Technologies. Accessed: Jun. 29, 2020. [Online]. Available: https://cept.org/

[31] *5G: Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document TR 38.901, 3GPP, Version 14.3.0, Release 14, 2018.

[32] Melbourne Water Corporation. (2019). *VIC MWC—Flood Mapping—Building Footprints Data Retrieved From AURIN*. Accessed: Jan. 1, 2021. [Online]. Available: https://data.aurin.org.au/dataset/mwc-mw-building-footprints-feb-2019-na

[33] SEMTECH. (2015). *LoRa Modulation Basics*. Accessed: Jan. 11, 2021. [Online]. Available: https://www.semtech.com/

[34] *Radio Spectrum Processor 2 & Radio Spectrum Processor 2Pro*, SDRPlay, Princes Risborough, U.K., 2017.

[35] B. Manzoor, A. Al-Hourani, B. A. Homssi, K. Magowe, S. Kandeepan, and K. G. Chavez, "Evaluating coverage performance of NB-IoT in the ISM-band," in *Proc. 27th Int. Conf. Telecommun. (ICT)*, Oct. 2020, pp. 1–5.

BASSEL AL HOMSSI (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the American University of Sharjah, United Arab Emirates, in 2014 and 2016, respectively, the M.Eng. degree in telecommunications from the University of Melbourne, in 2017, and the Ph.D. degree from the School of Engineering, RMIT University, Melbourne, Australia, in 2020. He was the Lead Network Designer of the Northern Melbourne Smart Cities Network (NMSCN), resulting in the deployment of the largest open-access IoT network in Australia. He is currently a Postdoctoral Research Assistant with the RMIT University. His research interests include wireless sensor networks, the Internet of Things, the practical deployment of large-scale IoT networks, and green IoT networks.

KOSTA DAKIC (Graduate Student Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from RMIT University, Melbourne, Australia, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include wireless sensor networks, the Internet-of-Things, and the application of deep learning in wireless communications.

SIMON MASELLI is currently the Founder and the CEO of Minnovation Australia. His Australian-based company manufactures IoT and telemetry products that have been deployed across Australia and New Zealand, South East Asia, and Europe for dozens of global brands.

HANS WOLF received the Masters of Business degree in information technology. He is currently the Chief Information Officer with the City of Whittlesea and a Sponsor of the Northern Melbourne Smart Cities Project. He has led technology teams in various public and private sector organizations ensuring stakeholder value by leveraging technology.

SITHAMPARANATHAN KANDEEPAN (Senior Member, IEEE) is currently the Discipline Leader for telecommunications and photonics with the School of Engineering, RMIT University. He has authored a scholarly book *Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation and Localization* (New York: Artech House, 2012). He has published over 135 peer-reviewed scientific journals and conference papers. His research interests include communication engineering, signal processing, and networks applied in wireless, satellite, and mobile systems. He has extensive experience in working with the industry for bringing research to practice and conducting fundamental research. He currently works on several research projects, such as defense science technology on intelligent spectrum management communications, satellite-aerial terrestrial communication systems, cognitive (satellite) radios, and the IoT communications. He is also engages with the ITU as an Academic Member. He had received the RMIT Research Excellence Award in 2019 for his research work with the industry. He was the Chair of IEEE Comsoc VIC Chapter. He was also the Vice-Chair of the IEEE Technical Committee on Cognitive Networks.

AKRAM AL-HOURANI (Senior Member, IEEE) received the Ph.D. degree from RMIT University, Melbourne, Australia, in 2016. He is currently a Senior Lecturer and the Program Manager for the Master of Engineering in telecommunication and networks with the School of Engineering, RMIT University. He published more than 60 journal articles and conference proceedings, including three book chapters. His current research interests include UAV communication systems, automotive and mmWave radars, energy efficiency in wireless networks, and the Internet-of-Things over satellite. In 2020, he has won the IEEE Sensors Council Paper Award for his contribution to hand-gesture recognition using neural networks. He has extensive industry/government engagement as a Chief Investigator in multiple research projects related to the Internet-of-Things (IoT), smart cities, satellite/wireless communications. As a Lead Chief Investigator, he oversaw the design and deployment of the largest open IoT network in Australia in collaboration with five local governments "Northern Melbourne Smart Cities Network," this project has won the 2020 "IoT Awards," the official awards program of IoT Alliance Australia. Prior to his academic career, from 2006 to 2013, he had extensively worked in the ICT industry sector as an Research and Development Engineer, a Radio Network Planning Engineer, and an ICT Program Manager for several projects spanning over different technologies, including mobile networks deployment, satellite networks, and railway ICT systems. He is serving as an Associate Editor for *Frontiers in Space Technologies* and *Frontiers in Communications and Networks*, and a Guest Editor for the special issue "Satellite Communication" in MDPI Remote Sensing.