Wi-Lo++: WiFi-to-LoRa Cross-Technology Communication with Multi-Packet Emulation

Sascha Rösler, *Student Member, IEEE*, Anatolij Zubow, *Senior Member, IEEE*, Christos Laskos, *Student Member, IEEE*, Falko Dressler, *Fellow, IEEE*

Abstract*—LoRa was developed for long-range communication in internet of things (IoT) applications, also in the 2.4 GHz ISM band. However, native LoRa support remains limited in mobile devices like smartphones. Cross-technology communication (CTC) addresses this gap by enabling direct over-the-air communication among different communication technologies without requiring additional radio hardware. This work presents `Wi-Lo++`, a WiFi-to-LoRa CTC mechanism. In contrast to existing WiFi-to-LoRa CTC approaches, it removes the constraint of the maximum packet length of WiFi and therefore can emulate LoRa transmissions with higher spreading factors (SFs). This is achieved by emulating a single LoRa frame through a train of consecutive IEEE 802.11b frames, each emulating only a part of the overall LoRa waveform while preserving symbol-level continuity across packet boundaries. In order to mitigate waveform distortions caused by the variety of immutable WiFi preambles and headers, we shift these distortions into less critical regions of the LoRa waveform. Using a second WiFi interface, these distortions can be further reduced further using overlapping WiFi frames. Experiments using software-defined radio (SDR) for WiFi and COTS LoRa hardware reveal that `Wi-Lo++` achieves the same performance as classical LoRa. Finally, a prototype implemented using commercially available off-the-shelf (COTS) WiFi hardware demonstrates the practical feasibility of our approach, achieving an outdoor over-the-air communication range of up to 856 m towards an unmodified LoRa receiver. Our `Wi-Lo++` framework is released as open-source for SF<7.

***Index Terms*—Cross-technology communication, LoRa, WiFi, emulation**

I. Introduction

THE rapid growth of the internet of things (IoT) has increased the demand for low-power, long-range wireless connectivity across large and heterogeneous networks. LoRa has emerged as a key IoT technology due to its long communication range, making it particularly suitable for smart city applications [1]. With the introduction of the Semtech SX1280 chipset, LoRa is available worldwide on the same frequency band, the 2.4 GHz ISM band, which is also used by WiFi. However, native LoRa support is still limited in many devices like smartphones.

Cross-technology communication (CTC) provides a promising solution to overcome such limitations. It allows one wireless technology to emulate the waveform of another physical-layer, enabling direct over-the-air communication without hardware modifications [2], while also improving resilience in heterogeneous wireless networks [3].

S. Rösler, A. Zubow, C. Laskos and F. Dressler are with School of Electrical Engineering and Computer Science, TU Berlin, Germany (e-mail: {roesler,zubow,laskos,dressler}@tkn.tu-berlin.de).

This work was supported by the German Research Foundation (DFG) within the project ResCTC under grants DR 639/30-1 and ZU 235/4-1. We like to thank Lorenz Pusch for his support during the experiments.

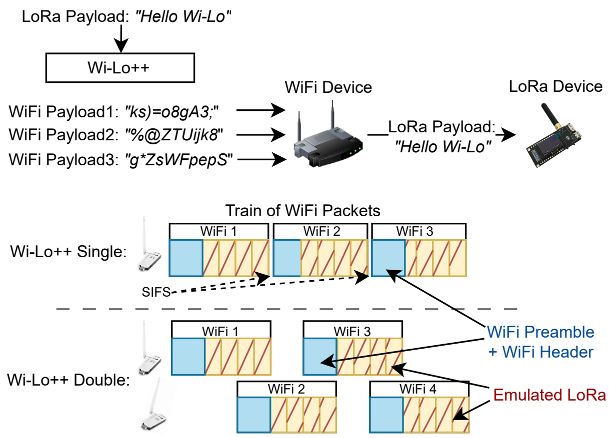


Figure 1. `Wi-Lo++` uses a train of multiple consecutive WiFi packets to emulate a single LoRa packet.

However, current CTC approaches like our WiFi-to-LoRa CTC Wi-Lo [4] are limited by the length of the emulating technology (e.g., WiFi). To overcome this limitation we introduce in this paper the idea of using a train of packets (see Figure 1) to emulate one packet of another technology (e.g., LoRa). Therefore, we enable long range communication on best-effort basis by building on the features of higher spreading factors (SFs). This can be used to achieve communication where no communication is possible otherwise without a guaranteed delivery.

We extend our prior work, Wi-Lo [4] with which we enabled direct WiFi to LoRa CTC by emulating narrow-band LoRa waveforms by broad-band IEEE 802.11b devices. In Wi-Lo we choose the complementary code keying (CCK) symbols that are closest to the upsampled LoRa waveform, which also becomes visible in the spectrum (chirps in Figure 5). Wi-Lo makes use of the band-pass filter of the narrow-band technology LoRa that smoothens the emulated signal as it is done for pulse-width modulation (PWM) signals. By using CCK, we avoid limitations of orthogonal frequency-division multiplexing (OFDM)-based approaches such as periodic distortions, uncontrollable cyclic prefix (CP) boundaries, unmodulated pilot tones, and randomized scrambling [2], [5].

To motivate our research in more detail, current CTC approaches remain constrained by the maximum packet size of the emulating technology (e.g., WiFi), which limits CTC with large bandwidth mismatches such as WiFi to LoRa. For example, a single LoRa transmission (SF 12, 203 kHz bandwidth, code rate $4/5$ and 24 Byte payload) can exceed 1 s, while the payload in IEEE 802.11b is limited by the maximum transmission unit (MTU) of 4095 Byte, which translates to 3 ms. The MTU for Linux is further limited to 2304 Byte including the RadioTap header. Thus, Wi-Lo was restricted to SF 5 and payload sizes of 32 Byte [4], also preventing larger communication ranges of higher SFs. This limit has been acknowledged but not resolved in the literature [6].

In this paper, we introduce `Wi-Lo++`, an extension of the Wi-Lo CTC approach, that enables the emulation of longer LoRa packets. This is achieved by sending a train of consecutive IEEE 802.11b packets (see Figure 1). However, the immutable WiFi preambles and headers raise issues as they distort the emulated LoRa waveform at various positions. We mitigate this impact by time shifting the emulated LoRa transmission within the train of IEEE 802.11b packets so that the distortions appear in the less critical regions of the emulated LoRa waveform. For LoRa configurations where this compensation is insufficient, we provide the `Wi-Lo++ Double` approach where two synchronized WiFi network interface cards (NICs) transmit by turns. Thus, the immutable part of the WiFi packet overlap with the mutable part of the previous WiFi packet (see Figure 1). This approach increases the effective signal to noise ratio (SNR) for the intervals where the immutable part of the WiFi packet is transmitted to approximately 0 dB. Finally, in order to guarantee the required strict transmission timing across consecutive WiFi packets, we utilize the concept of transmission opportunity (TXOP) introduced in IEEE 802.11e for deterministic channel access in IEEE 802.11b and switch off listen before talk (LBT) for the consecutive packets inside the TXOP. This is not supported by commercially available off-the-shelf (COTS) WiFi NICs and requires changes of the firmware. The spacing between consecutive WiFi packets is set to a short interframe space (SIFS), which makes our approach compliant to the WiFi channel access.

We demonstrate the feasibility of `Wi-Lo++` by presenting results from link-level simulations as well as from experiments using both software-defined radio (SDR) and COTS WiFi hardware. Our prototype consists of two synchronized WiFi NICs with modified firmware, while for short SF 5 packets one NIC is sufficient.

In addition, we release two main parts of the `Wi-Lo++` as open source. Hence, we release the full framework for SF < 7 and for SF ≥ 7 we release the user-level Matlab code used for simulations and calculation of the `Wi-Lo++` packets.

Contributions:

- We present `Wi-Lo++`, the first WiFi-to-LoRa CTC approach that uses multiple WiFi IEEE 802.11b packets to emulate a single LoRa transmission.
- `Wi-Lo++` allows emulation of larger airtime enabling high SF and large payload - a novelty for CTC with IEEE 802.11b.
- We show that time shifting the emulated LoRa waveform within the train of WiFi packets is beneficial as distortions from immutable parts of the WiFi frames are shifted into less critical regions of the LoRa waveform.
- `Wi-Lo++ Double` overcomes payload limitation for SFs ≥ 7, where two synchronized WiFi NICs transmit in turn with overlapping transmissions to reducing distortions from the immutable parts of the WiFi packets.
- Results from link-level simulations show impact of delay and jitter between consecutive WiFi packets within the train of packets.
- Our prototype shows proof-of-concept under real-world evaluation.

II. Related Work

Related work spans three major areas: First WiFi-to-LoRa CTC, second, recent CTC approaches, and, third, applications of CTC are discussed. Several works study CTC between WiFi and LoRa. In our previous work we presented Wi-Lo [4] that uses COTS IEEE 802.11b hardware to emulate short LoRa packets. We also showed that multiple LoRa transmissions on different radio channels can be emulated with a single IEEE 802.11b packet. Gao et al. [6] presented an approach using the IEEE 802.11g OFDM physical layer for emulation. Results using universal software radio peripheral (USRP) hardware were presented, but the limitation of short LoRa packets remains unresolved. WiRa [5], [7] used frame aggregation and resource units (RUs) from IEEE 802.11ax to emulate longer LoRa transmissions. However, the mandatory subframe headers distort LoRa symbols and require filling of additional LoRa symbols. Those symbols are removed by the modified LoRa receiver. The underlying OFDM modulation still suffers from CP boundaries, pilot tones, and scrambling effects. LoFi [8] explored the reverse direction, i.e., LoRa-to-WiFi CTC, using a technique termed as hitchhiking where the LoRa waveform is reconstructed from the WiFi bit error patterns.

Recent machine learning (ML)-based CTC approaches such as XiTuXi [9] and NNCTC [10] use neural network models and a transformer architecture to emulate different IoT waveforms and improve robustness without the need for full protocol knowledge. In XiTuXi [9] the useful features of the transmission are learned rather than mimicking the time-domain signal of the emulated waveform as close as possible. Gao et al. [11] used a transformer architecture also on the receiver side while He et al. [12] train a CTC transformer model by adding additional information via a Wingman token. However, these methods address symbol mapping, not the packet-length constraints imposed by the emulating technology.

Beyond LoRa-focused work, several recent CTC systems target other wireless technologies. Cho and Shin [13]–[15] demonstrate bidirectional CTC between IEEE 802.11b and Bluetooth low energy (BLE). They directly access radio registers and inject I/Q samples via optimized direct memory access (DMA) transfers and single instruction, multiple data (SIMD) instructions, enabling even real-time WiFi authentication on low-power BLE chips. WiBLE [16] achieves WiFi-to-BLE communication by introducing controlled phase shifts using IEEE 802.11b direct sequence spread spectrum (DSSS), leveraging the matching symbol durations of both technologies.

The authors extend their work to enable WiFi to BLE audio transmission and solve scheduling issues by disabling carrier sense multiple access with collision avoidance (CSMA/CA) in their kernel level CTC implementation [17]. For WiFi-to-ZigBee communication, Cheng et al. [18] provide a detailed insight in how to design a CTC based on CCK spreading. With their WibZig approach, they increase the packet reception rate at longer distances of WiFi-to-ZigBee CTCs. TransFi [19] uses the higher degrees of freedom in a MIMO system to emulate ZigBee waveforms, while PCTC [20] minimizes phase distortion to support parallel ZigBee receivers. Other works exploit hitchhiking effects, such as Gao et al. [21], who receive BLE with WiFi by encoding information in repeated symbol patterns, or use a continuous phase signal as acknowledgment [22], [23]. Finally, in our previous work we presented CTC schemes for communication between LTE-U/LAA and WiFi [24], [25].

Side-channel based CTC is still of interest in research. c-Chirp [26] addresses the asymmetric communication range in bidirectional WiFi-ZigBee CTC. It increases the communication range of ZigBee by sending LoRa-like chirps as ZigBee perform channel hopping. This cross-technology interference (CTI) can be seen in the channel state information (CSI) measurements on a WiFi receiver. SLEM [27] enables faster WiFi-to-ZigBee communication by modulating the transmit power using quadrature amplitude modulation (QAM), which the ZigBee receiver decodes via energy detection. Although SLEM requires a high SNR, it avoids the pilot-tone and center-frequency alignment issues that limit other OFDM-based WiFi-to-ZigBee CTC.

Higher-layer aspects of CTC have also been explored. The CTC WeBee [28] is used to distribute periodic synchronization messages across ZigBee networks, extending coordination range. KEP [29] introduces location-based key generation to enable encrypted bidirectional CTC. ML approaches address security, such as detecting and mitigating jamming [30], identifying spoofing and sniffing attacks [31]. They can also be used to classifying relay or injection attacks through error-pattern analysis [32].

To the best of our knowledge, no prior CTC schemes support multi-packet LoRa emulation on COTS WiFi and unmodified LoRa hardware and analyzes the resulting timing and distortion limits.

III. Background

We begin by giving a short introduction into the IEEE 802.11b waveform and channel access, followed by the presentation of the LoRa physical layer.

A. IEEE 802.11b Waveform

Even though the IEEE 802.11b standard [33] was published already in 2000, it is still supported by actual hardware for compatibility. It defines WiFi transmission in the 2.4 GHz ISM band and includes a 5.5 Mbit/s and a 11 Mbit/s version. The physical layer uses CCK as spreading technique for the PLCP service data unit (PSDU) while the physical layer convergence procedure (PLCP) header is DBPSK-modulated and code spreaded (DSSS). In the 11 Mbit/s variant, blocks of 8 bits are selected and modulated pairwise. The first pair of bits of each 8 bit block are Gray-coded and differential quadrature phase shift keying (DQPSK) modulated in reference to the modulated first pair of the block before. For the first PSDU symbol, the reference is taken from a DQPSK modulation of the header. The phase of every odd symbol gets an extra 180°. The other 6 bits of each block are quadrature phase shift keying (QPSK) modulated and are non-Gray coded. At the end of the modulation, the four phases are spread over 8 chips [33]:

$$c = \Big\{e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, -e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, -e^{j(\phi_1+\phi_2)}, e^{j\phi_1}\Big\} \quad (1)$$

All bits of the physical layer protocol data unit (PPDU) are scrambled using a scrambler polynomial of $z^{-7} + z^{-4} + 1$ with an initial state of $[1101100]$ [33].

B. IEEE 802.11b Channel Access

A WiFi device using IEEE 802.11b relies on the distributed coordination function (DCF) for channel access. Whenever a WiFi device has a packet to transmit, it first has to perform carrier sensing to determine if the channel is idle. In case the channel is busy, the device has to wait until it becomes empty and has to perform the channel access routine afterwards. Therefore, it has to wait for a DCF interframe space (DIFS) in case it wants to transmit a data packet, while an acknowledgment (ACK) can be sent after a SIFS. After the DIFS, the device has to wait for a random backoff. The duration of the backoff is calculated as

$$T_{\text{Backoff}} = U\big[\text{CW}_{\min}, \text{CW}_{\max}\big] \cdot T_{\text{slot}} \quad (2)$$

where U denotes a uniform random distribution between the two parameters $\text{CW}_{\min}$ and $\text{CW}_{\max}$ and T_{slot} the duration of a slot. In numbers, T_{slot} = 20 µs, T_{SIFS} = 10 µs and T_{DIFS} = 50 µs. During DIFS and each slot of the backoff, the device has to check for an idle channel and has to stop the procedure if the channel becomes busy [33].

In IEEE 802.11e [34], quality of service (QoS) is introduced for WiFi including the enhanced distributed channel access (EDCA). Besides adding prioritization for specific traffic by introducing variable arbitrary interframe space (AIFS) instead of DIFS and changes to the contention window parameter ($\text{CW}_{\min}$ and $\text{CW}_{\max}$), TXOP is introduced. Instead of requiring a WiFi device to run the complete channel access procedure for each packet, it can send multiple packets after each other with only one SIFS in between if no ACK is required. DCF is only performed for the first packet of a train of packets [34].

C. LoRa Waveform

LoRa is a proprietary wireless technology [35], [36], which achieves long transmission ranges due to its chirp spread spectrum (CSS) technique. With the Semtech SX1280 chip, LoRa is also available in the 2.4 GHz band and shares the spectrum with technologies like WiFi. A LoRa symbol consists of one chirp (up or down chirp) for which the starting frequency (f_0) of a chirp defines its symbol value. The chirps are divided

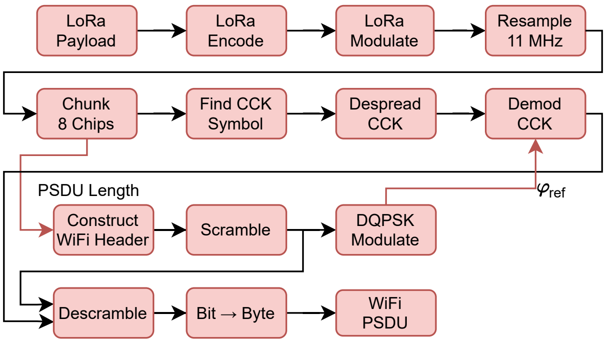


Figure 2. Wi-Lo processing chain [4]

into 2^{SF} chips with a duration of $\frac{1}{\mathrm{BW}}$. The frequency stays constant during the duration of a chip and changes between consecutive chips. In case of an upchirp, the frequency increases with each chip of the LoRa symbol starting from f_0 until it reaches the bound of the defined bandwidth $\mathrm{BW}/2$. Afterwards, the chirp continues from $-\mathrm{BW}/2$ until it concludes one period. There are 2^{SF} different starting frequencies allowing LoRa to modulate SF-many bits into one symbol. Demodulation of LoRa symbols happens via multiplication of an inverted base chirp (mathematically speaking, the complex conjugated base chirp) and performing a fast Fourier transformation (FFT) on the result. The symbol value can be conducted from the peak of this FFT. In case of a symbol value $\neq 0$, the chirp is spit into two parts (one part ending at $\mathrm{BW}/2$ and the next starting at $-\mathrm{BW}/2$). Therefore, two peaks become visible in the spectrum; here merging improves demodulation [37], [38].

IV. Wi-Lo in a Nutshell

This work is based on Wi-Lo [4], which we introduce in the following. The key idea of Wi-Lo is the usage of IEEE 802.11b CCK WiFi packets to mimic LoRa signals. It therefore chooses CCK symbols that have smallest Euclidean distance to the upsampled LoRa waveform. So, a 11 MHz WiFi signal is used to emulate a 1.625 MHz LoRa signal. Due to the filtering and lower sample rate of LoRa only the spectrum around the LoRa center frequency is demodulated. All in all, Wi-Lo transmits a WiFi payload that is not meaningful for WiFi but forms a waveform that LoRa can demodulate.

As depicted in Figure 2, the core step of Wi-Lo is the selection of the CCK symbol out of 256 possible symbols, which emulates a specific chunk of the LoRa waveform best. As preparation we resample the LoRa baseband signal to 11 MHz, such that it has the same sample rate as WiFi IEEE 802.11b to emulate a LoRa signal on the same center frequency as WiFi. Indeed, the LoRa signal can also be shifted within the 11 MHz of WiFi. The upsampled signal is chunked into blocks of 8 samples, such that each chunk can be reconstructed by one CCK symbol consisting of 8 chips. Our first attempt was to find the closest CCK symbol by the smallest Euclidean distance (L2-Norm). However, using the complex cross-correlation instead reduces the execution time by a factor of 3.7 when using a host with AMD Ryzen 7 CPU. Even though the different approaches select different CCK symbols, they achieve the same performance of packet delivery ratio (PDR) in the additive white Gaussian noise (AWGN) channel as they preserve the features of the LoRa signal. Additionally, due to the bandwidth asymmetry there are multiple possible emulated signals. Cross-correlation performs well, as each CCK chip is $\in \{1, i, -1, -i\}$ so real and imaginary part are $\in \{\pm 1, 0\}$. Intermediate values are formed by the band-pass filter of the LoRa receiver, thus we elaborate the lower bandwidth of LoRa and use WiFi as PWM (see Figure 3). This lower bandwidth of LoRa is achieved by digital-to-analog conversion at the WiFi transmitter and analog-to-digital conversion in the LoRa hardware in a real-world implementation. For simulations down-sampling in the digital domain is used. Thus, the LoRa waveform is formed at the receiver from the generated pulses. The low-pass filter has two important tasks: first, removal of out-of-LoRa-band frequencies from the WiFi waveform, second, smoothing the PWM behavior of the CCK signal.

As the payload is sent via a COTS WiFi card, the effect of the scrambler, which uses a static initialization [33], and the influence of the PLCP header bits of WiFi are included in the payload calculation (see Figure 2). Both are important as the 11 Mbit variant of IEEE 802.11b uses DBPSK or even DQPSK and, therefore, takes the modulation of the header as a reference (φ_{ref}). Therefore, the scrambled header is needed to demodulate the WiFi payload, which is the LoRa signal in our case. The header bits are also needed to descramble the required WiFi bits. For these two reasons, it is important to estimate the correct WiFi PLCP header the WiFi NIC will generate later on. This includes the selection of the data rate, preamble setting, and the correct estimation of the payload length. The generation of the PLCP header can not be switched off or changed as it is done by hardware.

V. Problem Statement

Due to their lower data rate, narrow-band technologies like LoRa use much longer air time than broadband technologies like WiFi. In case of LoRa, the duration of a single symbol is

$$T_{sym} = \frac{2^{\mathrm{SF}}}{\mathrm{BW}} \tag{3}$$

where SF is the spreading factor and BW the used bandwidth. For example, a SF 12 symbol has a transmission time of 2.5 ms for $\mathrm{BW} = 1625\,\mathrm{kHz}$. Already a LoRa header consists of 12.25 symbols and its preamble consists normally of 8 symbols resulting in 51 ms of airtime. However, a CTC system is limited by the maximum packet size, and hence airtime, of its emulating technology. Thus, in case of Wi-Lo we are limited by the maximum MTU of IEEE 802.11b of 4095 Byte. As Wi-Lo uses 11 Mbit/s the resulting airtime is only 2.978 ms limiting the LoRa configurations which can be emulated. Thus, Wi-Lo [4] is limited to SF 5 and 64 Byte. The resulting research

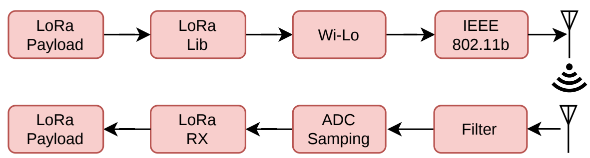


Figure 3. Wi-Lo-to-LoRa transmission chain

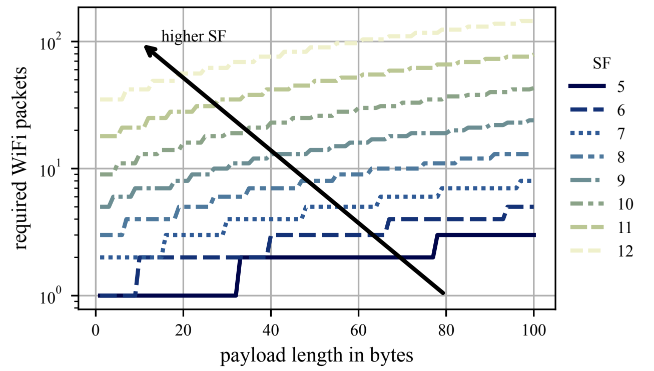


Figure 4. Number of WiFi packets needed to emulate a single LoRa packet (CR=4/5, BW = 1625 kHz)

question is: *Can we design a WiFi-to-LoRa CTC which can emulate LoRa packets with a large airtime?*

VI. Wi-Lo++ Multi-Packet Emulation

In this paper, we introduce Wi-Lo++ which overcomes the highlighted limitations of its predecessor.

A. Key Idea

The key idea of Wi-Lo++ is to emulate a single LoRa packet with the help of a train of consecutive IEEE 802.11b WiFi packets. Thus, we chunk the LoRa packet into consecutive parts and apply Wi-Lo for each of these parts separately under the assumption that the WiFi packet is sent in time. We therefore assume fixed timing of the Wi-Lo packet transmission.

As Wi-Lo++ uses multiple WiFi packets to emulate a single LoRa packet, we analyze the required number of WiFi packets in Figure 4 for MTU size of 2304 Byte (maximal length on Linux) for different payload lengths and SFs for BW = 1625 kHz. For example, more than 140 WiFi packets are needed to emulate a single 100 Byte SF 12 LoRa packet while already a 1 Byte SF 7 LoRa packet requires two WiFi packets. Please note that using a smaller bandwidth further increases the airtime of a LoRa packet and hence the number of WiFi packets required for emulation even further.

As the transmission of consecutive WiFi packets results in gaps between the packets due to the random access based on DCF, we perform DCF for channel access only for the first packet of the train. All subsequent packets within the train are sent contention-free after SIFS as suggested in IEEE 802.11e with TXOP [34].

However, the additional immutable WiFi header and preamble of each intermediate WiFi packet within the train creates distortions in the emulated LoRa waveform. We solve this problem by providing two alternative solutions: Wi-Lo++ Single and Wi-Lo++ Double. First, Wi-Lo++ Single uses only a single WiFi NIC to send the train of WiFi packets (see Figure 1). Distortions by the WiFi preambles and header are placed at rather harmless positions within the emulated LoRa waveform. This is achieved by slightly time shifting the beginning of the emulation within the first packet of the train of packets. However, this solution works only for some LoRa settings and has limits in the payload length. The Wi-Lo++ Double approach uses two synchronized WiFi cards such that the second NIC starts its transmission already during the transmission of the first NIC. Hence, the immutable parts of WiFi overlaps with the preceding packet (see Figure 1) and the stream of emulated LoRa is continuous. Wi-Lo++ Double works for all payload lengths, however, it requires a second WiFi NIC and synchronization of both NICs.

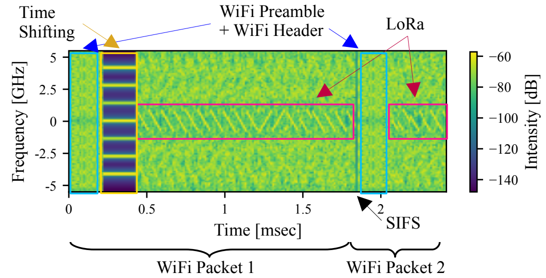


Figure 5. A LoRa waveform emulated using a train of two WiFi packets

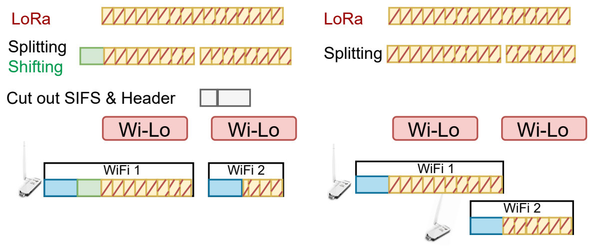
B. Detailed Description

The key idea of Wi-Lo++ is to emulate a single LoRa packet with multiple consecutive WiFi packets. Figure 5 shows an example where two WiFi packets were used for emulation. We can clearly see that packets have to sent in time and the immutable parts, i.e., PHY preamble and header, of the second WiFi packet create a distortion in the emulation starting at $t = 1.84$ ms. Moreover, we can observe that a part of the first WiFi packet was not used for emulation, i.e., the emulation started with an offset of $t = 0.21$ ms. This time shifting was needed in order to shift the distortions from the immutable parts of the second packet to less important regions in the LoRa waveform. It becomes apparent that the gap by the SIFS is quit small and barley visible in Figure 5.

Wi-Lo++ computes the number of IEEE 802.11b packets and their payload bytes needed to emulate a single LoRa packet. This is the difference between the two versions. In case of the Wi-Lo++ Single approach, we take into account the fact that the WiFi preamble and header are created by the WiFi NIC. Hence, their known durations are removed from the to be emulated LoRa packet to keep the emulated LoRa waveform in sync as the WiFi NIC will send the WiFi preamble and header instead (Figure 6a). Moreover, we also account for the gap, i.e., SIFS, between WiFi packets.

Note that even though the usage of TXOP Wi-Lo++ deviates from legacy IEEE 802.11b, it does not break the coordination of WiFi. This is because each WiFi packet inside the train has it own PHY preamble and header (see Figure 5) making it recognizable by other WiFi devices by means of carrier sensing. Furthermore, the WiFi header makes the Wi-Lo++ transmission recognizable by other by other WiFi devices and prevents them from collisions, which is not the case for legacy LoRa transmissions. Thus, even though Wi-Lo++ consumes more bandwidth than legacy LoRa it also prevents from CTI.

The distortions by the additional WiFi preambles and headers are handled by two different approaches. First, in the Wi-Lo++ Single approach (Figure 6a), we are time shifting the LoRa waveform within the train of WiFi packets to make the decoding of the LoRa frame possible. This is needed to make sure that



(a) `Wi-Lo++ Single` (b) `Wi-Lo++ Double`

Figure 6. From LoRa to `Wi-Lo++` by splitting the LoRa signal and applying Wi-Lo

the immutable parts of WiFi interfere with the least important parts of the LoRa waveform. We achieve the offset by adding zeros in front of the LoRa waveform. However, the `Wi-Lo++ Single` approach works only for some LoRa configurations (see Section VII-B).

For this reason, we developed the second approach, `Wi-Lo++ Double`, which requires the usage of two WiFi NICs (Figure 6b). Here the WiFi packets of a train are sent alternately by two synchronized WiFi cards. The key idea is to start the transmission of a WiFi packet on the second NIC already during the transmission of the previous WiFi packet on the first NIC. The packets are aligned in a way that the previous packet ends at the same time when the payload of the next packet starts as shown in Figure 6b. So for the LoRa receiver instead of just receiving noise (the WiFi header), there is a meaningful signal mixed with the WiFi header. Mathematically speaking the signal to interference ratio (SIR) of the distorted periods is increased from $-\infty$ to ≈ 0 dB. Hence, this approach allows the emulation of all LoRa configurations.

C. Software Prototype

The predecessor Wi-Lo worked only with precomputed LoRa waveforms. In this work, we extend it to full-chain solution that sends any given LoRa payload from a WiFi device using our `Wi-Lo++` approach (see Figure 2). Our implementation uses MATLAB and the LoRa library by Xu et al. [37]. We extended the library to includes the generation of SF 5 and SF 6 packets. For testing, we use legacy Atheros WiFi NICs for which we modified the firmware to achieve TXOP for the multi-packet emulation. Note, that for the single-packet version of `Wi-Lo++` our prototype works without changes to the WiFi firmware. We realized that the WiFi NIC requires a valid MAC header at the start of each injected WiFi packet which we take into account when calculating the required WiFi payload. Additionally, our prototype is limited to the usage of the long IEEE 802.11b preamble. Our single-packet version of `Wi-Lo++` uses a Python script that starts the calculation in MATLAB and injects the packets using a raw socket into WiFi NIC afterwards. This Python application is listening on a Linux-TUN device for LoRa payload to be transmitted. We made our software publicly available on GitHub.[1]

[1]https://github.com/tkn-tub/wi-lo.git

VII. Link-Level Simulation

In this section, we evaluate `Wi-Lo++` by means of link-level simulations. The goal is to validate the feasibility of multi-packet LoRa emulation, quantify its limits, and study the impact of timing imperfections, payload size, bandwidth, coding rate, and noise.

A. Methodology

We conduct link-level baseband simulations using the MATLAB WLAN Toolbox together with the LoRa physical-layer implementation [37]. Except where explicitly stated otherwise, we assume an ideal channel without noise. Moreover, unless noted, we assume that consecutive WiFi packets forming a train are transmitted without gaps; thus, the mandatory SIFS spacing under TXOP is ignored in the baseline analysis.

Our default LoRa configuration uses a payload length of 10 Byte, a CR of 4/5, and a bandwidth of 1.625 MHz. Under these settings, single-packet emulation is possible for SF 5 and SF 6; therefore, these SFs are excluded from experiments that explicitly study multi-packet emulation effects.

We systematically analyze the influence of (i) time offset for `Wi-Lo++ Single` (ii) payload length, (iii) controlled and uncontrolled inter-packet delays, (iv) jitter between packets, (v) LoRa bandwidth and coding rate, and (vi) AWGN.

B. Influence of Time Offset

We first evaluate the feasibility of `Wi-Lo++ Single`, which relies on a controlled time offset applied to the emulated LoRa waveform. This offset shifts distortions introduced by immutable WiFi preambles and headers into less critical regions of the LoRa frame. Without such a shift, LoRa packets using SF 7 or SF 8 cannot be reliably decoded, as the distortions overlap with essential synchronization and payload regions.

Figure 7 shows the PDR as a function of the applied offset for different SFs and payload lengths. For short payloads and SF $\in 7, 8$, an appropriate offset enables reliable communication. However, for payloads of 50 Byte, no offset yields successful reception for SFs $\leq$ 8, as the number of distortions becomes too large to be placed harmlessly. For SFs $\geq$ 10, no offset is required, since the LoRa symbol duration ($\geq$ 630 µs) exceeds the distortion duration ($\sim$ 192 µs), allowing successful demodulation despite partial symbol corruption. For SFs $\leq$ 6, only a limited range of offsets allows successful decoding, and this range shrinks as the payload length increases. Consequently, multi-packet emulation with `Wi-Lo++ Single` is infeasible for SF $\leq$ 6.

Take away: Controlled time shifting enables `Wi-Lo++ Single` for SFs $\in \{7, 8\}$ for 1625 kHz and short payloads, but its applicability is fundamentally limited for large payloads.

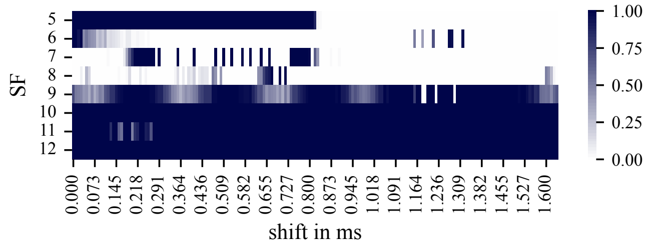
C. Impact of Payload Length

Next, we analyze how the LoRa payload length affects PDR. Figure 8a shows that for `Wi-Lo++ Single` and a bandwidth of 1.625 MHz, smaller SFs strongly limit the maximum payload size. SF 5 supports payloads of up to 32 Byte, whereas SF 6

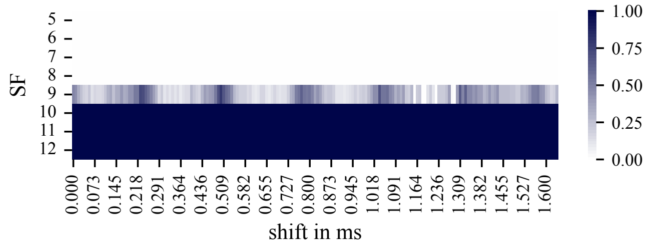
Table I
COMPARISON OF WI-LO WITH THE TWO PROPOSED `Wi-Lo++` APPROACHES

Property	**Wi-Lo**	**`Wi-Lo++ Single`**	**`Wi-Lo++ Double`**
Number of WiFi NICs	Single WiFi NIC	Single WiFi NIC	Two synchronized WiFi NICs
Supported LoRa SFs	SF 5, SF 6	Unlimited	Unlimited
Maximum payload length	Strictly limited (64 or 18 Byte)	Limited (SF- and BW-dependent)	Partially limited (SF $<$ 7)
Timing sensitivity	Low	Very high	Very high
Tolerance to WiFi preamble/header distortions	High	Low	High (overlapping transmissions)
Required WiFi firmware modifications	No	Yes (TXOP)	Yes (parallel transmission)
Implementation complexity	Low	High	Higher

(a) LoRa payload length = 1 Byte

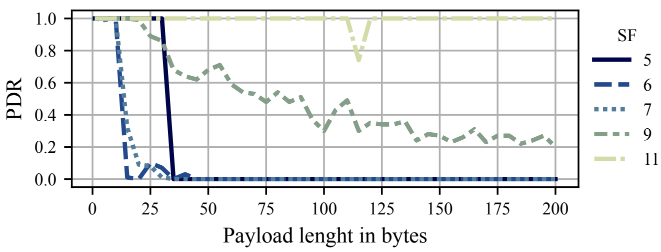


(b) LoRa payload length = 10 Byte

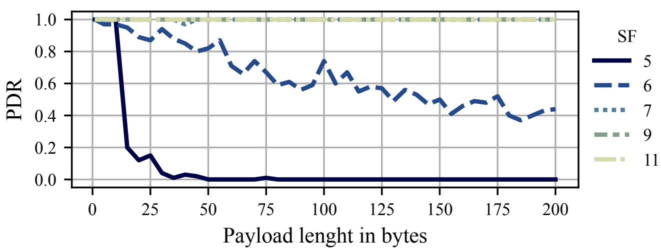


(c) LoRa payload length = 50 Byte

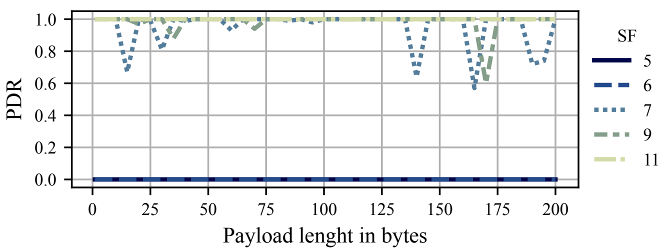
Figure 7. `Wi-Lo++ Single`: influence of time shifting the LoRa signal inside the train of WiFi packets on PDR for different SF and payload length



(a) `Wi-Lo++ Single` (BW = 1625 kHz)



(b) `Wi-Lo++ Single` (BW = 203 kHz)



(c) `Wi-Lo++ Double` (BW = 1625 kHz)

Figure 8. Influence of payload length on the PDR (CR=4/5)

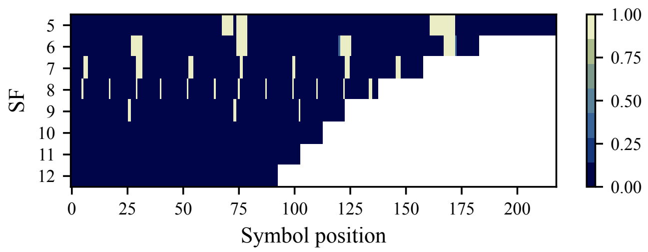


Figure 9. The position of WiFi header is clearly visible from the probability, that a LoRa symbol at a specific position is erroneous (CR=4/5, BW = 1625 kHz, Payload = 100 Byte)

and SF 7 are limited to around 9 Byte. For SF $\geq$ 10, the PDR remains close to 1 for all tested payload lengths.

This behavior is explained by distorted LoRa symbols by the WiFi preambles. As shown in Figure 9, increasing SF leads to fewer, longer LoRa symbols, which reduces the number of symbols affected by WiFi-induced distortions. The figure shows which symbol positions are prone to errors, however, this might not translate into bit errors due to coding. Additionally, the interleaver can spread symbol errors to multiple bytes. Symbols can be resolved if they overlap with the header only partially, which is visible in the irregular error pattern for SF 9 and the lower drop in PDR. For SF $\geq$ 10, no symbol errors are observed.

Reducing the LoRa bandwidth to 203 kHz significantly improves robustness (Figure 8b), enabling longer payloads for SF $\geq$ 6. In contrast, `Wi-Lo++ Double` largely removes

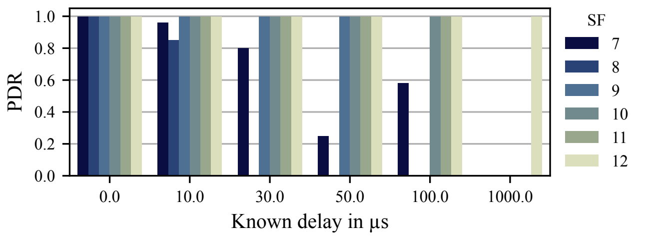


Figure 10. Influence of known delay between packets on PDR for `Wi-Lo++ Single` (CR=4/5, BW = 1625 kHz).

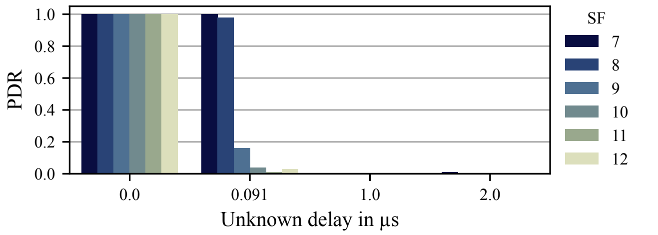


Figure 11. Influence of unknown delay between packets on packet delivery ratio (PDR) for CR=4/5, BW = 1625 kHz

payload length limitations for all SFs $\geq$ 7 (Figure 8c), as overlapping transmissions increase the effective SNR during distorted intervals. However, an additional header for SF$<$ 7 distorts even the `Wi-Lo++ Double` reception.

Take away: Payload length limitations in `Wi-Lo++ Single` primarily affect up to SF 9 and can be mitigated by reducing bandwidth or by using `Wi-Lo++ Double`.

D. Influence of Known Delays

We now study the effect of fixed, known inter-packet delays, which model deterministic gaps such as SIFS. In the simulations, a silent period is added between the WiFi packets and the known delay is compensated by removing the corresponding duration from the LoRa waveform.

As shown in Figure 10, small delays of up to 10 µs (SIFS) are tolerable for most SFs. Delays of 30 µs already prevent successful decoding for SF 8, corresponding to roughly 20% of its symbol duration. For DIFS (50 µs) only SFs $\geq$ 9 are decodable. The robustness of SF 7 is most likely due to the lower number of required WiFi packets, while the higher SFs profit from longer symbol durations. For SF 9, delays of 100 µs leads to complete failure, while SF 10 and SF 11 tolerate delays up to 100 µs. With the exception of SF 7, packet transmissions for a delay of up to 100 µs fail due to symbol errors, while the preamble detection works perfect. `Wi-Lo++ Double` is more robust, maintaining a PDR of 1 for SF$>$ 7 and delays of up to 100 µs. Lower LoRa bandwidths further improve tolerance, allowing reliable reception for delays of up to 100 µs for SFs$>$ 7 and 203 kHz bandwidth.

Take away: Known delays in the order of SIFS are acceptable, but larger deterministic delays quickly degrade performance, especially for small SFs.

E. Influence of Unknown Delays

We next consider unknown, uncompensated delays between WiFi packets emulating the LoRa packet, which cause loss

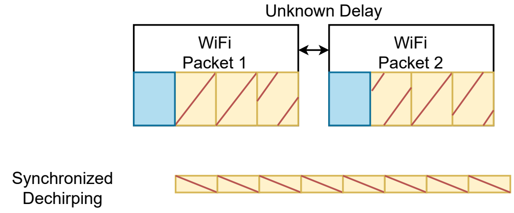
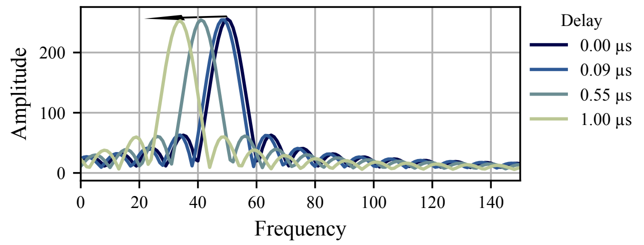
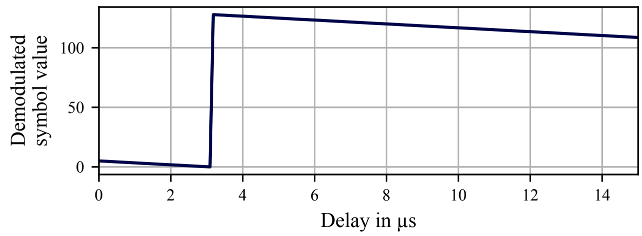


Figure 12. Demodulation error due to lost synchronization caused by unknown delay. The dechirping by multiplication with a downchirp is done beyond the boundaries of a symbol.



(a) Spectrogram of a delayed LoRa symbol

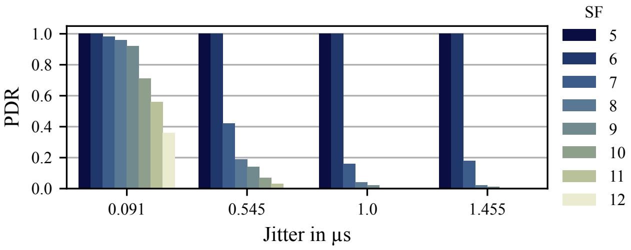


(b) Demodulated LoRa symbol value in case the symbol is delayed for SF 7 and BW=1.625 MHz

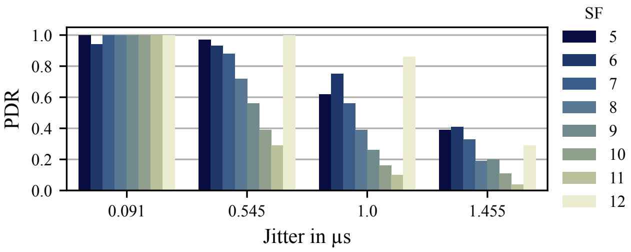
Figure 13. LoRa packet demodulation is affected by unknown delays

of synchronization between LoRa symbols and the dechirping process. Even a delay of a single WiFi sample (90.9 ns) prevents communication for SF $\geq$ 9, as shown in Figure 11. The root cause is the loss of synchronization during demodulation which results in a misaligned despreading (Figure 12). The frequency modulation (FM) demodulation by a FFT of the misaligned chirp results in a left shift of the peaks in the spectrum as illustrated in Figure 13a. For example, in the case of a delay of 1 µs, the spectrum is shifted by 20 Hz. The detected symbol value for a symbol value of 5 for SF 7 in Figure 13b shows how the symbol value decreases by one every 600 ns of delay. Thus, LoRa is highly sensitive to loss of synchronization. The coding in LoRa is designed to correct bit errors, however, systematic bit errors cannot be corrected. Higher SFs are particularly vulnerable, as symbol spacing in the frequency domain decreases with increasing SF. Additionally, the demand for more WiFi packets to be used leads to a summation of the unpredicted delay. Reducing the bandwidth to 203 kHz slightly improves robustness, allowing successful decoding for some higher SFs. However, unknown delays remain a fundamental limitation for multi-packet emulation.

Take away: `Wi-Lo++` is extremely sensitive to unknown delays; even sub-microsecond timing errors can completely prevent decoding.



(a) LoRa BW = 1625 kHz



(b) LoRa BW = 203 kHz

Figure 14. Influence of jitter on PDR (CR=4/5)

F. Influence of Jitter

Jitter between consecutive WiFi packets is modeled as a Gaussian random variable. As shown in Figure 14b, jitter with a standard deviation of only one WiFi sample (90.9 ns) already degrades the PDR for larger SFs. At 545 ns, multi-packet emulation fails for nearly all SFs at 1.625 MHz bandwidth. Lower bandwidth significantly improves resilience: at 203 kHz, jitter of 90.9 ns is tolerable for all SFs, and even larger jitter values are averaged out for SF 12.

Take away: Jitter must be kept well below 500 ns for reliable multi-packet emulation; lower LoRa bandwidths substantially increase robustness.

G. Influence of Noise

Next, we compare `Wi-Lo++` against legacy LoRa under an AWGN channel. Figure 15a shows that `Wi-Lo++ Single` achieves nearly identical performance to LoRa when equal power spectral density (PSD) is assumed, with at most a 5 dB loss for SF 7. This degradation is mainly caused by residual distortions from WiFi headers, which become critical for smaller SFs when additional noise is present. Indeed, SF 6 is very sensitive to noise, most probably due to distortion by the second WiFi header, which is resolvable unless more noise adds to it. For SF 9 the steepness of the curve not as steep as symbols are partially recovered as explained in Section VII-C. This effect is more relevant for larger packets (see Figure 15b).

`Wi-Lo++ Double` slightly outperforms `Wi-Lo++ Single` due to its higher effective SNR during overlapping transmissions resulting in a gain of $\frac{1648\,\mu\mathrm{s}}{1648\,\mu\mathrm{s}-192\,\mu\mathrm{s}} = 1.06\,\mathrm{dB}$. Indeed, there is also a PDR close to 1 for lower SFs where the `Wi-Lo++ Single` approach fails at all for 100 Byte.

Take away: Under noise, `Wi-Lo++` achieves LoRa-like robustness, with `Wi-Lo++ Double` providing a modest SNR gain.

H. Influence of Bandwidth

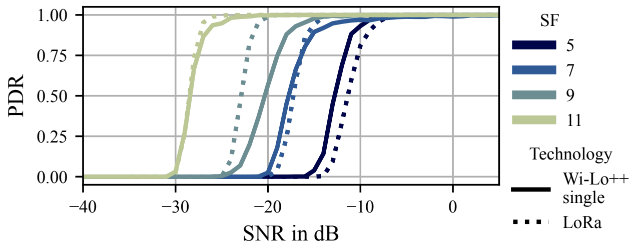
We next analyze the impact of the LoRa bandwidth on `Wi-Lo++` performance. Although reducing the bandwidth increases the overall airtime of a LoRa packet and thus the number of WiFi packets required for emulation, it substantially increases the symbol duration. Longer symbols improve robustness against known delays. Our results show that a bandwidth of 203 kHz marginally improves tolerance to both unknown delays and jitter when compared to 1.625 MHz.

In particular, multi-packet emulation becomes feasible for a wider range of SFs, and the payload-length limitations observed for `Wi-Lo++ Single` at large bandwidths are partially mitigated. Under AWGN, the reduced bandwidth yields an SNR gain of approximately 6 dB, consistent with the behavior of native LoRa. An exception occurs for small SFs, where the increased airtime forces multi-packet emulation and can therefore reintroduce sensitivity to timing errors, even under otherwise favorable channel conditions.

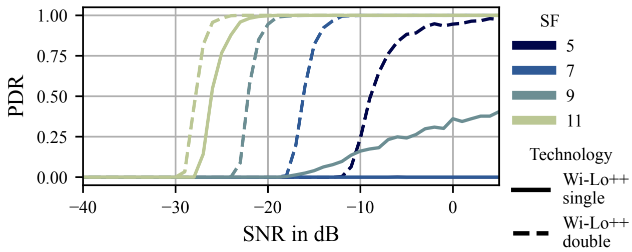
Take away: Reducing the LoRa bandwidth increases robustness to known delay, and noise, but negatively affect small SFs due to the need for multi-packet emulation.

I. Influence of Coding Rate

We finally investigate the influence of the LoRa coding rate. Using a higher coding rate of 4/8 increases redundancy and improves resilience against noise at the symbol level, but it also lengthens the LoRa packet while keeping the symbol duration unchanged. This makes the overall transmission more vulnerable to cumulative distortions introduced by WiFi headers and timing imperfections. Our simulations show that higher coding rates often reduce the PDR in the presence of known or unknown delays and jitter, particularly for small and moderate SFs. While gains from stronger forward error correction are visible in pure noise-limited scenarios, they are outweighed by the increased sensitivity to timing-related distortions in most multi-packet emulation settings.

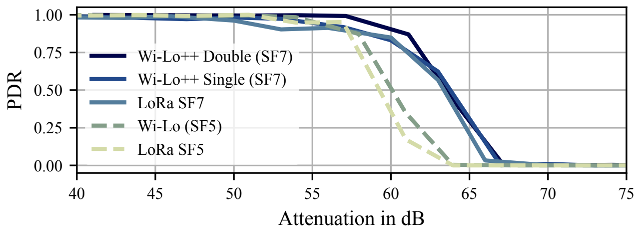


(a) LoRa payload length = 10 Byte

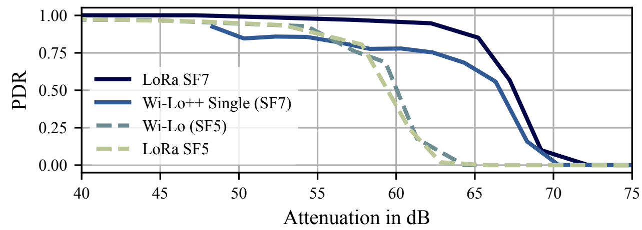


(b) LoRa payload length = 100 Byte

Figure 15. PDR of `Wi-Lo++` and LoRa under different SNR



(a) USRP WiFi transmitter



(b) COTS WiFi transmitter

Figure 16. Influence of attenuation on the PDR of `Wi-Lo++`, Wi-Lo, WiFi and LoRa in a wired setup

Take away: Although a higher coding rate improves noise robustness, it is generally disadvantageous for `Wi-Lo++` due to the increased packet duration and accumulated distortions.

J. Summary

Our link-level simulations demonstrate that both of our proposed versions are feasible. `Wi-Lo++ Single` is limited in payload length for SF$\leq$9 at large bandwidths but benefits from reduced bandwidth. `Wi-Lo++ Double` removes payload limitations for SFs $\geq$ 7 at the cost of additional hardware. Across all configurations, precise and deterministic timing is essential, as `Wi-Lo++` is highly sensitive to unknown delays and jitter. Under ideal synchronization, `Wi-Lo++` achieves robustness comparable to native LoRa.

VIII. Experimental Evaluation

This section evaluates `Wi-Lo++` using both controlled laboratory experiments and real-world measurements. We first assess signal robustness in a wired setup using a SDR as a WiFi transmitter. We then evaluate the practical performance of `Wi-Lo++` with COTS WiFi hardware in both wired and outdoor environments. In both cases an unmodified COTS iM282A LoRa receiver with a SX1280 chip was used.

A. SDR-based Evaluation

We begin with a controlled, cable-based experiment in which an USRP acts as the WiFi transmitter and is directly connected to a COTS LoRa receiver through a programmable attenuator. All `Wi-Lo++` waveforms are precomputed in MATLAB using the WLAN Toolbox and the LoRa PHY implementation by Xu et al. [37]. For `Wi-Lo++ Double`, the two WiFi waveforms are generated independently and superimposed prior to transmission, emulating perfectly synchronized NICs.

Figure 16a shows the PDR as a function of attenuation. Both `Wi-Lo++ Single` and `Wi-Lo++ Double` achieve a PDR of 1 and slightly outperform native LoRa under equalized PSD.

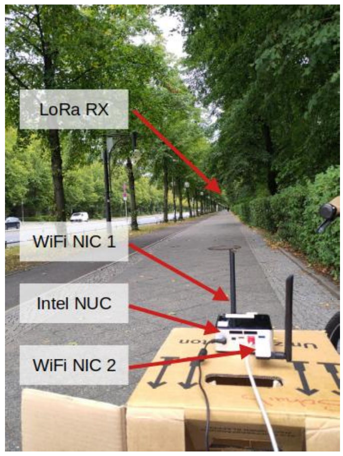


Figure 17. `Wi-Lo++ Single` prototype used in outdoor experiment

This gain can be caused by noise during the PSD measurement for power normalization. As expected, `Wi-Lo++ Double` outperforms LoRa and `Wi-Lo++ Single` slightly due to the increased effective SNR during intervals affected by WiFi preambles and headers.

Take away: In a fully synchronized and noise-controlled environment, `Wi-Lo++` matches the robustness of native LoRa, validating the feasibility of multi-packet emulation.

B. Evaluation with COTS WiFi Hardware

We next evaluate `Wi-Lo++` using COTS hardware to assess real-world feasibility. Unless stated otherwise, we focus on the `Wi-Lo++ Single` approach, as USRP-based results indicate similar trends for both variants.

We compare four configurations: (i) `Wi-Lo++ Single` emulating SF 7 LoRa packets, (ii) native LoRa SF 7, (iii) native LoRa SF 5, and (iv) single-packet Wi-Lo emulating SF 5. For SF 7, `Wi-Lo++` requires two consecutive WiFi packets to emulate a single LoRa frame.

Although `Wi-Lo++ Single` conceptually requires only one NIC, our prototype uses two synchronized NICs because we were not able to enqueue packets fast enough. Hence, our `Wi-Lo++` prototype for SF 7 (Figure 17) uses two Atheros AR9271 USB WiFi NICs with a modified firmware and an unmodified LoRa receiver. The firmware changes are required to allow for TXOP, which is not part of the IEEE 802.11b standard, and to achieve synchronization between the NICs.

To achieve the strict timing required for TXOP-based transmission, we disable contention for the second WiFi packet and preload it onto the NIC. Synchronization between the WiFi NICs works over the air. Therefore, we modified the firmware of a WiFi jammer by Vanhoef and Piessens [39] to trigger the second transmission immediately after the first one. In more detail, the second WiFi NIC acts as receiver and waits for the first NIC to start. After receiving the valid first WiFi packet it waits for a calibrated duration before switching into transmission mode send the second packet. This method works as the first WiFi packet is always of full length in case a second one is needed. While switching into transmit mode takes time the transition is scheduled prior to the required start of the second packet to achieve a continuous calibrated stream from

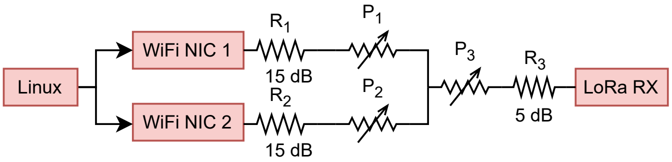


Figure 18. Wired setup with attenuators

both WiFi NICs. Our prototype is limited to Atheros WiFi NICs up to IEEE 802.11n. Both sides have one dipole antenna.

For LoRa waveforms that can be emulated by one WiFi packet (e.g. SF 5 and payload $\leq$ 32 byte), our prototype works with one WiFi card. This comes with the legacy firmware as there is no synchronization nor TXOP needed in this case.

Take away: Our prototype requires two WiFi NICs due to too slow enqueue of packets. Hence, the firmware is modified to allow synchronization and TXOP.

C. Wired Experiments with COTS Hardware

We first evaluate the prototype in a wired setup (Figure 18) that enables controlled SNR measurements and asymmetric power experiments. In the SNR experiment, we adjust the attenuation by P_3. The results in Figure 16b show how Wi-Lo and native LoRa achieve comparable performance for SF 5. However, `Wi-Lo++ Single` exhibits a slightly reduced PDR of approximately 0.9 compared to native LoRa SF 7. This degradation is primarily attributed to imperfect synchronization between consecutive WiFi packets.

To further investigate synchronization effects, we vary the relative transmit power of the two WiFi NICs by adjusting P_1 and P_2. For the overall system, we count P_1 negative; thus, a negative power ratio relates to the signal of the first WiFi NIC being weaker than the signal of the second WiFi NIC, while a positive power relates to a stronger signal from the first WiFi NIC. Figure 19 shows that an increase of P_1 or P_2 degrades performance which comes as side effect as the attenuation between the WiFi cards is increased. This increases jitter and harms the synchronization. Interestingly, `Wi-Lo++ Single` is more resilient when the first packet is attenuated than when the second packet is weakened. This asymmetry is explained by the LoRa receiver's reliance on the preamble for gain control: an overly strong preamble is harmful when the second packet will be too weak. However, the PDR is stable for -60 dB to 30 dB allowing antenna placements within 30 cm in a free-space channel model.

Take away: With COTS hardware, `Wi-Lo++ Single` remains robust under symmetric conditions with a slight loss in PDR and tolerate a power asymmetry of 30 dB, but it is sensitive higher attenuation between the WiFi NICs.

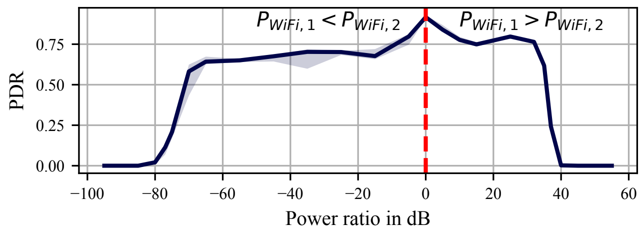


Figure 19. Impact of power asymmetry on PDR of `Wi-Lo++ Single`

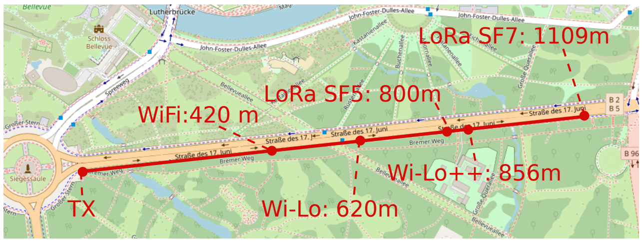


Figure 20. Maximum communication range of `Wi-Lo++ Single`, Wi-Lo, WiFi and LoRa along an urban street in a park area

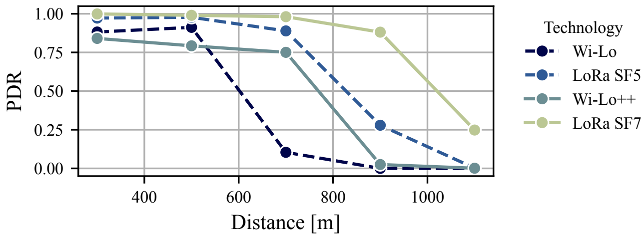


Figure 21. PDR of `Wi-Lo++ Single`, Wi-Lo and LoRa along an urban street in a park area

D. Outdoor Experiments

Finally, we evaluate `Wi-Lo++ Single` in an outdoor scenario along an urban road in a park area (Figure 20), such that line-of-sight (LOS) was achieved. We measure both maximum communication range and PDR and compare `Wi-Lo++` (SF 7), Wi-Lo (SF 5), native LoRa, and standard WiFi. We configure WiFi to transmit with 20 dBm while LoRa uses 12.5 dBm emitting similar power per bandwidth. For both technologies, we use the same $\lambda/2$ Dipole antennas with a theoretical gain of 2.15 dB.

The maximum communication range of `Wi-Lo++` is evaluated first. As shown in Figure 20, `Wi-Lo++ Single` significantly extends WiFi maximum communication range from 414 m to 856 m, outperforming single-packet Wi-Lo (620 m). This improvement aligns with the higher effective SF emulated by `Wi-Lo++`. However, native LoRa still achieves longer ranges, reaching approximately 800 m for SF 5 and 1109 m for SF 7, highlighting residual imperfections in waveform emulation.

Figure 21 presents the PDR as a function of distance. `Wi-Lo++ Single` achieves about 84% PDR at 300 m and maintains a relatively stable PDR up to 700 m, after which performance degrades. The loss in PDR aligns with the loss by the higher attenuation between the WiFi cards in our lab. In contrast, native LoRa consistently outperforms both Wi-Lo and `Wi-Lo++` across all distances.

To better understand these losses, we analyze timing jitter

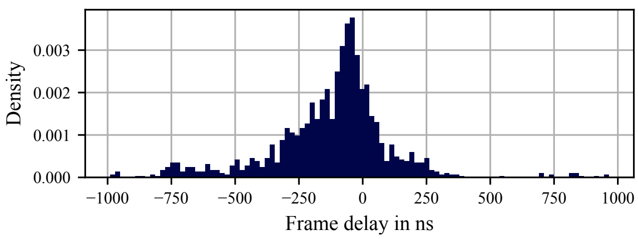


Figure 22. Measured jitter of the `Wi-Lo++` prototype

by recording Wi-Lo++ transmissions in a cable setup with an USRP. The histogram in Figure 22 shows a median inter-packet delay of approximately −85 ns, with a spread between −300 ns and 250 ns and a standard deviation of 240 ns. As shown in Section VII, jitter on this order is sufficient to noticeably degrade PDR, explaining the observed performance gap in outdoor experiments. Therefore, Wi-Lo++ achieves a reduces PDR.

Take away: Wi-Lo++ substantially extends WiFi range in real-world environments but remains sensitive to sub-microsecond timing jitter, which limits its achievable PDR compared to native LoRa.

E. Summary

Across all experiments, Wi-Lo++ demonstrates the feasibility of multi-packet LoRa emulation on unmodified LoRa receivers. In controlled settings, it achieves LoRa-like robustness, while in practical deployments it is outperformed by LoRa. Nevertheless, Wi-Lo++ significantly outperforms WiFi in terms of range. The main limiting factor is precise inter-packet timing, especially when using COTS hardware, motivating future work on tighter synchronization and NIC-level optimizations.

IX. Discussion

Our evaluation demonstrates that by transmitting a train of IEEE 802.11b packets Wi-Lo++ enables direct WiFi-to-LoRa communication at LoRa-like performance under controlled and well-synchronized conditions. Experiments with COTS hardware show proof-of concept. However, the results reveal a clear performance gap compared to native LoRa, manifested as a reduced PDR and lower effective SNR with our prototype, particularly in outdoor deployments.

A. Timing

A central observation is the high sensitivity of Wi-Lo++ to timing imperfections between consecutive WiFi packets. Even sub-microsecond jitter and small synchronization errors can noticeably degrade decoding performance, as predicted by our simulation results. Setups of Wi-Lo++ must therefore achieve an ultra-fine synchronization, where 240 ns seams to be the upper limit. While such timing precision can be achieved in laboratory settings, it is difficult to guarantee on COTS WiFi hardware that was not designed for deterministic, fine-grained inter-packet timing. Therefore, the proposed synchronizing via WiFi preambles works at the cost of reduced PDR. This largely explains the discrepancy between simulations SDR-based experiments and real-world measurements. A better synchronization mechanism would improve our prototype and, hence, lead to a higher PDR. Nevertheless, retransmissions of packets can help to achieve a higher PDR at the cost of reduced throughput.

B. Comparison Wi-Lo++ Single and Wi-Lo++ Double

The Wi-Lo++ Single approach represents a pragmatic trade-off between system complexity and capability. It extends the original Wi-Lo design by supporting higher SFs without requiring additional hardware in principle, but it remains constrained in payload length for SFs ≤ 9. These constraints arise from the accumulation of waveform distortions introduced by repeated WiFi preambles and headers. The Wi-Lo++ Double approach overcomes these limitations by overlapping transmissions from two synchronized WiFi interfaces, effectively increasing robustness against such distortions at the cost of additional hardware and synchronization complexity.

C. Efficiency

As Wi-Lo, Wi-Lo++ comes at a cost of longer airtime due to the additional PLCP header of WiFi of the first packet in a train of packets, which adds 192 µs. Shifting within a packet also adds up to 660 µs additional airtime for Wi-Lo++ Single, while there is no additional airtime needed for Wi-Lo++ Double. While small LoRa packets fit into one WiFi packet, longer LoRa packets require a train of multiple WiFi packets. As the header of this additional packets are sent during the ongoing LoRa transmission, they are sent instead of an emulated LoRa and therefore do not consume additional airtime. Additionally, packets are sent directly after each other with only SIFS in between (Wi-Lo++ Single) or even start while another WiFi packet is sent (Wi-Lo++ Double). As the SIFS fall into the duration of the LoRa transmission, they also do not consume more airtime. All in all, there is no additional use of airtime cunsumed as legacy LoRa except the first PLCP header and shifting in case of Wi-Lo++ Single.

Another cost of Wi-Lo++ is a reduced spectral efficiency, since WiFi with its 11 MHz of bandwidth is used for a Wi-Lo transmission compared to LoRa with a bandwidth of 1.625 MHz. However, this can be seen as an advantage as it improves coexistence between WiFi and LoRa as the additional WiFi headers make the ongoing emulated LoRa transmission recognizable to other WiFi stations. With Wi-Lo++ there is a legacy PLCP WiFi header at the beginning of each transmission, using the complete bandwidth of WiFi. Therefore, WiFi stations can detect the start of the transmission as any other WiFi transmission. As the emulated LoRa is sent as a WiFi packet it consumes the complete WiFi channel and makes the transmission detectable by other WiFi stations and avoids CTI which can happen otherwise cased by the bandwidth asymmetry. The intermediate WiFi header are again decodable for Wi-Lo++ Single and notify other WiFi stations of the ongoing transmission. Therefore, CTI is mitigated as a side effect. Despite this, an explicit cross-technology coordination can developed on top of Wi-Lo++

D. Comparison to Other Approaches

We compare Wi-Lo++ to our previous Wi-Lo approach [4], legacy LoRa, WiRa [7], and WiLo (Gao et al. [6]) in Table II. While LoRa has no overhead, emulation technologies require an additional WiFi preamble (192 µs for IEEE 802.11b, 24 µs for IEEE 802.11g and 43.2 µs for IEEE 802.11ac) which is highest for Wi-Lo++. Due to the higher bandwidth of WiFi (11 MHz for IEEE 802.11b and 20 MHz for OFDM WiFi) emulation comes with higher energy costs which can be reduced by

Table II
COMPARISON TO STATE OF THE ART APPROACHES

	LoRa	**Wi-Lo++**	**Wi-Lo [4]**	**WiRa [7]**	**Gao et al. [6]**
Airtime overhead	0 µs	192 µs	192 µs	43.2 µs	24 µs
Spectral energy costs	1×	6.8×	6.8× / 2.3× [5]	12.5×	12.5× / 6.3× [4]
PHY throughput (SF 5)	193 kbit/s	132 kbit/s	132 kbit/s / 396 kbit/s [5]	169 kbit/s	169 kbit/s / 507 kbit/s [4]
Range	1050 m	830 m	600 m	> 300 m	> 500 m [2]
PDR	1	0.9	1	1	0.98
Hardware complexity	1 board	1-2 NICs	1 NIC	1 USRP	1 USRP
TX modifications	no	firmware, synchronization	no		
RX modifications	no	no	no	yes [3]	no

the emulation of parallel LoRa streams. The preamble creates additional energy costs. The throughput of emulated LoRa is highest for SF 5, even though this comes with only single packet emulation and its payload length limitation to the length of one WiFi packet for all approaches except WiRa. Compared to LoRa the throughput is additionally reduced by the additional WiFi preamble and, in case of WiRa, by additional filling symbols (reduction by 14% [7]). However, WiRa is evaluated under SF 8 and BW = 500 kHz. Nevertheless, the throughput of 40 kbit/s for WiRa and 1.3 kbit/s for WiLo is lower than calculations. This can also affect the measured range for which WiRa and Gao et al. [6] not fully show the limits. We also just use SF 7 while the range for Wi-Lo is reduced as it is limited to SF 5. We expect all approaches and LoRa to achieve higher ranges for SF 12. The achieved PDR is reduced for Gao et al. [6] and Wi-Lo++. Even though Wi-Lo++ requires modified firmware, only our approaches show feasibility with COTS hardware. WiRa and [6] use USRPs and WiRa rfurtehr equires changes to the receiver.

E. Summary

For all of our experiments we use legacy LoRa receiver with all its features like the capture effect. However, we did not look into coexistence between Wi-Lo++ and LoRa, where Wi-Lo++ has to perform DCF as channel access and therefore might have less access to the channel.

Despite these limitations, Wi-Lo++ expands the design space of CTC by demonstrating that long-duration, low-rate waveforms can be emulated using packet trains on existing WiFi hardware. Thus, new WiFi-to-LoRa CTC combinations are enabled. This capability enables new forms of interoperability and resilience in heterogeneous wireless systems, particularly in scenarios where LoRa connectivity is needed but dedicated LoRa transmitters are unavailable. By extending the communication range beyond WiFi, Wi-Lo++ enables connectivity in scenarios where WiFi alone fails, such as sparse deployments, or infrastructure-limited environments without the requirement for additional hardware. In mesh networks Wi-Lo++ enables wider connectivity and, therefore, increases resilience against failing intermediate nodes. In such cases, even a moderate PDR can be sufficient to enable communication, positioning Wi-Lo++ as a complementary technology rather than a replacement for native LoRa.

[2] with SF 8, BW = 500 kHz

[3] WiRa fills in symbols which can be corrupted during transmission of subframe headers. Those symbols are removed at the receiver.

[4] two LoRa streams

[5] three LoRa streams

X. CONCLUSION

We presented Wi-Lo++, a WiFi-to-LoRa CTC system that enables direct over-the-air communication from commodity IEEE 802.11b devices to unmodified LoRa receivers in the 2.4 GHz ISM band. Wi-Lo++ overcomes the fundamental packet-length limitation of prior solutions by emulating a single LoRa transmission using a train of consecutive WiFi packets, thereby supporting LoRa frames with large airtime. We introduced two complementary designs. Wi-Lo++ Single mitigates waveform distortions caused by immutable WiFi preambles and headers through controlled time shifting, enabling multi-packet emulation for selected SFs and payload sizes using a single WiFi transmitter. Wi-Lo++ Double generalizes this approach by overlapping transmissions from two synchronized WiFi interfaces, allowing the emulation of arbitrary LoRa configurations at the cost of additional hardware. The performance evaluation shows that, under precise and deterministic timing, Wi-Lo++ achieves robustness comparable to native LoRa. In real-world deployments, Wi-Lo++ substantially extends the communication range of WiFi – up to 856 m in our outdoor experiments – demonstrating its practical value despite residual performance gaps caused by timing jitter and synchronization limitations on commodity hardware.

Our results suggest several directions for future work. First, improving synchronization accuracy on COTS WiFi hardware - potentially through tighter firmware integration or hardware timestamping - could significantly enhance robustness. Second, extending the current prototype to support all LoRa SFs and payload configurations remains an important goal. Third, eliminating the need for multiple WiFi NICs in the Wi-Lo++ Single prototype by leveraging a deeper understanding of NIC internals would further improve practicality and deployability. Finally, coexistence and interference between legacy LoRa and Wi-Lo++ is an interesting topic to continue the research.

REFERENCES

[1] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *IEEE International Conference on Communications (ICC 2017)*, Paris, France: IEEE, May 2017, pp. 1–7.

[2] Z. Li and T. He, "WEBee: Physical-Layer Cross-Technology Communication via Emulation," in *23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017)*, Snowbird, UT: ACM, Oct. 2017, pp. 2–14.

[3] A. Zubow, I. von Stebut, S. Rösler, and F. Dressler, "ResCTC: Resilience in Wireless Networks through Cross-Technology Communication," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2024)*, Valencia, Spain: IEEE, Sep. 2024, pp. 1–6.

[4] P. Gawłowicz, A. Zubow, and F. Dressler, "Wi-Lo: Emulation of LoRa using Commodity 802.11b WiFi Devices," in *IEEE International Conference on Communications (ICC 2022)*, Seoul, South Korea: IEEE, May 2022, pp. 4414–4419.

[5] D. Xia and X. Zheng, "Towards Symmetric Cross-technology Communication among Heterogeneous IoT Devices," in *ACM Turing Award Celebration Conference - China (ACM TURC 2023)*, Wuhan, China: ACM, Jun. 2023, pp. 37–38.

[6] D. Gao, H. Wang, S. Wang, W. Wang, Z. Yin, S. Mumtaz, X. Li, V. Frascolla, and A. Nallanathan, "WiLo: Long-Range Cross-Technology Communication from Wi-Fi to LoRa," *IEEE Transactions on Communications*, pp. 1–15, Sep. 2024.

[7] D. Xia, X. Zheng, F. Yu, L. Liu, and H. Ma, "WiRa: Enabling Cross-Technology Communication from Wifi to Lora with IEEEE 802.11ax," in *41st IEEE International Conference on Computer Communications (INFOCOM 2022)*, Virtual Conference: IEEE, May 2022.

[8] D. Gao, W. Jiang, R. Liu, W. Wang, Z. Zhang, and Y. Liu, "LoFi: Physical-layer CTC from LoRa to WiFi with IEEE 802.11ax," in *44th IEEE International Conference on Computer Communications (INFOCOM 2025)*, London, United Kingdom: IEEE, May 2025, pp. 1–10.

[9] S. Liao, Z. An, Q. Pan, X. Zhao, J. Tong, and L. Yang, "XiTuXi: Sealing the Gaps in Cross-Technology Communication by Neural Machine Translation," in *21st ACM Conference on Embedded Networked Sensor Systems (SenSys 2023)*, Istanbul, Turkey: ACM, Nov. 2023, pp. 70–82.

[10] H. Wang, J. Wang, W. Jiang, S. Wang, and D. Gao, "Physical Layer Cross-Technology Communication Via Explainable Neural Networks," *IEEE Transactions on Mobile Computing*, pp. 1–17, Oct. 2024.

[11] D. Gao, W. Jiang, R. Liu, Y. Liu, T. He, S. Wang, and Y. Wang, "Poster Abstract: Neural Network-based OFDM/QAM Modulation for Wi-Fi-to-X Communication," in *23rd ACM Conference on Embedded Networked Sensor Systems (SenSys 2025)*, Irvine, CA: ACM, May 2025, pp. 654–655.

[12] H. He, Q. Wu, S. Yuan, Z. Gu, Z. Yu, and Q. Zhu, "Heterogeneous Packet Translation for Cross-Technology Communication," in *50th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2025)*, Hyderabad, India: IEEE, Apr. 2025, pp. 1–5.

[13] H.-W. Cho and K. G. Shin, "FLEW: Fully Emulated Wifi," in *28th ACM International Conference on Mobile Computing and Networking (MobiCom 2022)*, Sydney, Australia: ACM, Oct. 2022, pp. 29–41.

[14] H.-W. Cho and K. G. Shin, "Unify: Turning BLE/FSK SoC into WiFi SoC," in *29th ACM International Conference on Mobile Computing and Networking (MobiCom 2023)*, Madrid, Spain: ACM, Oct. 2023.

[15] H.-W. Cho and K. G. Shin, "DREW: Double-Throughput Emulated WiFi," in *30th ACM International Conference on Mobile Computing and Networking (MobiCom 2024)*, Washington, D.C.: ACM, Nov. 2024, pp. 663–678.

[16] L. Li, Y. Chen, and Z. Li, "WiBle: Physical-Layer Cross-Technology Communication with Symbol Transition Mapping," in *18th IEEE International Conference on Sensing, Communication and Networking (SECON 2021)*, Virtual Conference: IEEE, Jul. 2021, pp. 1–9.

[17] L. Li, Z. Liu, Y. Zhang, S. Peng, Z. Liu, Y. Chen, and Z. Li, "WiLE-Audio: Wide-Coverage Low-Energy Audio via WiFi-BLE Cross-Technology Communication," in *31st ACM International Conference on Mobile Computing and Networking (MobiCom 2025)*, Hong Kong, China: Association for Computing Machinery (ACM), Nov. 2025.

[18] T. Cheng, S. Li, W. Dang, Q. Wen, and Y. Pan, "Enabling Reliable and Commodity-Device Compatible PHY-CTC From WiFi to ZigBee via Chip Emulation in Phase," *IEEE Sensors Journal*, vol. 24, no. 11, pp. 18 444–18 456, Jun. 2024.

[19] R. Chen and W. Gao, "TransFi: Emulating Custom Wireless Physical Layer from Commodity Wifi," in *20th ACM International Conference on Mobile Systems, Applications and Services (MobiSys 2022)*, Portland, OR: ACM, Jun. 2022, pp. 357–370.

[20] J. Yan, S. Cheng, Z. Li, and J. Liu, "PCTC: Parallel Cross Technology Communication in Heterogeneous wireless systems," in *21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2022)*, Virtual Conference: IEEE, May 2022, pp. 67–78.

[21] D. Gao, L. Ou, Y. Chen, X. Guo, R. Liu, Y. Liu, and T. He, "Physical-Layer CTC From BLE to Wi-Fi With IEEE 802.11ax," *IEEE Transactions on Mobile Computing*, vol. 24, no. 1, pp. 338–351, Jan. 2025.

[22] H. Guo, L. Feng, Y. Xiang, and Z. Ouyang, "Design and Application of Physical Layer ACK frame for WiFi to ZigBee Cross Technology Communication," in *34th IEEE Wireless and Optical Communications Conference (WOCC 2025)*, Taipa, Macao: IEEE, May 2025, pp. 235–239.

[23] S. Yao, Q. Zhao, M. Zhou, L. Feng, P. Zhang, X. Xu, and H. Chen, "Offset-Resistant Physical-Layer Cross-Technology Acknowledgment: A General Approach and Its Case Studies," *IEEE Wireless Communications*, pp. 1–9, May 2025.

[24] P. Gawłowicz, A. Zubow, and A. Wolisz, "Enabling Cross-technology Communication between LTE Unlicensed and WiFi," in *37th IEEE Conference on Computer Communications (INFOCOM 2018)*, Honolulu, HI: IEEE, Apr. 2018.

[25] P. Gawłowicz, A. Zubow, S. Bayhan, and A. Wolisz, "Punched Cards over the Air: Cross-Technology Communication Between LTE-U/LAA and WiFi," in *21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2020)*, Virtual Conference: IEEE, Sep. 2020, pp. 297–306.

[26] D. Xia, X. Zheng, L. Liu, C. Wang, and H. Ma, "c-Chirp: Towards Symmetric Cross-Technology Communication Over Asymmetric Channels," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1169–1182, Jun. 2021.

[27] J. Yao, X. Zheng, R. Xie, and K. Wu, "Cross-Technology Communication for Heterogeneous Wireless Devices Through Symbol-Level Energy Modulation," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 3926–3940, Nov. 2022.

[28] D. Gao, Y. Liu, H. Bin, L. Wang, W. Chen, Y. Chen, and T. He, "Time Synchronization Based on Cross-Technology Communication for IoT Networks," *IEEE Internet of Things Journal*, vol. 10, no. 22, pp. 19 753–19 764, Nov. 2023.

[29] W. Wang, X. Liu, Z. Chi, S. Ray, and T. Zhu, "Key Establishment for Secure Asymmetric Cross-Technology Communication," in *19th ACM Asia Conference on Computer and Communications Security (ASIA CCS 2024)*, Singapore, Singapore: ACM, Jul. 2024, pp. 412–422.

[30] X. Zhang, Y. Wu, A. Zhang, and G. Zhang, "Jamming Attack and Defense Based on Deep Learning in Cross-Technology Communication," in *2nd 2024 IEEE International Conference on Cloud and Network Computing (ICCNC 2024)*, Jinhua, China: IEEE, Jun. 2024, pp. 126–133.

[31] D. Gao, L. Ou, Y. Liu, Q. Yang, and H. Wang, "DeepSpoof: Deep Reinforcement Learning-Based Spoofing Attack in Cross-Technology Multimedia Communication," *IEEE Transactions on Multimedia*, vol. 26, pp. 10 879–10 891, Jun. 2024.

[32] W. Wang, D. Niyato, Z. Xiong, and Z. Yin, "Detection and Authentication for Cross-Technology Communication," *IEEE Transactions on Vehicular Technology*, pp. 1–15, Sep. 2024.

[33] IEEE, "802.11b-1999 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band," IEEE, Std 802.11b-1999, Jan. 2000.

[34] IEEE, "802.11e-2005 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," IEEE, Std 802.11e-2005, Nov. 2005.

[35] O. B. A. Seller and N. Sornin, "Low Power Long Range Transmitter," US Patent Office, US Patent US9252834B2, Feb. 2016.

[36] J. Haxhibeqiri, E. D. Poorter, I. Moerman, and J. Hoebeke, "A Survey of LoRaWAN for IoT: From Technology to Application," *MDPI Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018.

[37] Z. Xu, S. Tong, P. Xie, and J. Wang, "From Demodulation to Decoding: Toward Complete LoRa PHY Understanding and Implementation," *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–27, Jan. 2023.

[38] 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 Transactions on Sensor Networks*, vol. 15, no. 2, pp. 1–35, May 2019.

[39] M. Vanhoef and F. Piessens, "Advanced Wi-Fi Attacks Using Commodity Hardware," in *30th Annual Computer Security Applications Conference (ACSAC 2014)*, New Orleans, LA: ACM, Dec. 2014, pp. 256–265.