Design and Implementation of a Throughput Improvement Method for Asynchronous Pulse Code Multiple Access

Atsushi Nakamura
Dept. of Elec. Eng.
Tokyo Univ. of Science
Tokyo, Japan
0000-0002-9530-639X

Ferdinand Peper
CiNet
NICT
Osaka, Japan
0000-0002-8576-7934

Kenji Leibnitz
CiNet
NICT
Osaka, Japan
0000-0002-3691-3675

Naoki Wakamiya
Grad. School of IST
Osaka University
Osaka, Japan
0000-0002-6195-6087

Mikio Hasegawa
Dept. of Elec. Eng.
Tokyo Univ. of Science
Tokyo, Japan
0000-0001-5638-8022

***Abstract*—APCMA is a pulse code-based communication method designed for massive IoT. This method is collision-resistant and suitable for high-density environments but has the disadvantage of long message lengths and low throughput. In this paper, we propose a method to improve throughput by using sub-slots to solve this drawback. In addition, we introduce two demodulation techniques: frequency offset detection and sliding window. We implemented the proposed method using USRP and GNU Radio and conducted experiments. Experimental results show that the proposed method has higher throughput than the conventional APCMA and a higher success rate and throughput than LoRa in high-density environments.**

***Index Terms*—LPWA, Massive IoT, APCMA**

I. Introduction

Recently, the Internet of Things (IoT) has spread over the world and many applications are using IoT devices. Accordingly, the number of IoT devices is rapidly increasing and will reach 30 billion in 2025. An environment with such a large density of IoT devices is referred to as Massive IoT. Several Low Power Wide Area (LPWA) methods, such as LoRa, ELTRES, and SigFox, are already used in practice. However, since these LPWAs are not tolerant to packet collisions, the communication quality is degraded in Massive IoT.

Asynchronous Pulse Code Multiple Access (APCMA) [1] was proposed as a new communication scheme suitable for Massive IoT. An APCMA message consists of several pulses and the payload is encoded into the intervals between the pulses. The message in which the data is encoded is called a codeword, and the codeword is designed to could be decoded even when the messages overlap (Fig. 1). This makes APCMA tolerant of packet collisions and, therefore, makes it a suitable communication scheme for massive IoT. It is shown in [2] that APCMA has a higher success probability than CSMA/CA, while [3] shows experiments of high-density communication using 500 APCMA transmitters. In [4], the pulses in the APCMA codewords are enhanced to be modulated by Chirp Spread Spectrum (CSS) to improve reception sensitivity and facilitate long-distance communication.

APCMA enables high-density communication by using sparse pulse trains, but this also leads to longer message length and lower bit rate. For example, in the experiment with 500 transmitters conducted in [3], parameters are used such that the total packets per second were 9, resulting in a bit rate of 200 bps for the entire network. To adapt APCMA to more IoT applications, it is necessary to increase its throughput.

In this paper, we propose a scheme to shorten the APCMA message length and increase the bit rate. To achieve this, we use two techniques: a frequency offset derived in the process of demodulating the Chirp Spread Spectrum (CSS) and a sliding window method considered for demodulation of LoRa modulation proposed in [5]. Furthermore, we implement this scheme using software-defined radio (SDR) and prove its effectiveness through experiments. Our first experiments confirm that the proposed method can archive a higher throughput than

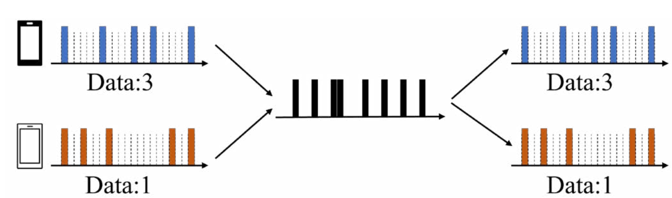


Fig. 1: Separable APCMA messages.

This research and development work is supported by MIC/SCOPE JP205007001.

conventional APCMA. The second experiments show that the proposed method has a higher success rate and throughput than LoRa in high-density environments.

II. Chirp Spread Spectrum (CSS)-APCMA

A. Asynchronous Pulse Code Multiple Access (APCMA) [1]

APCMA is based on the Communication through Silence (CtS) [6] scheme, which encodes data at intervals between pulses. The common advantage of CtS and APCMA is a low power consumption due to the pulses being very short and sparse. However, CtS has the disadvantage of being vulnerable to packet collisions because the pulse intervals are interrupted when messages overlap. To solve this problem, APCMA uses separable codewords with four or more pulses to become robust against packet collisions. Several codewords, such as 4-pulse codes, 5-pulse codes, and 6-pulse codes have been designed to decode them with high probability in [2], [7]. This paper uses a 5-pulse code, specifically consisting of four symmetrically arranged pulses and one central pulse.

The transmitter converts the transmitted data into the corresponding codeword and sends its pulses at the intervals determined by the codeword. The time axis is divided into equally spaced time-slots, and the pulses have a duration that exactly fits into one of the time slots like fig. 2a.

The receiving process is roughly divided into two processes, the first is pulse detection and the second is decoding the messages from the pulse train. The receiver first detects whether there is a pulse in each time slot and converts the received signal into a pulse train, assigning a "1" to a detected slot and a "0" to an undetected slot. There are two decoding schemes from pulse trains to received data: one is spike automaton [8] and the other is using a shift register [1], [9], which is the algorithm used in this paper. The shift register is a method where decoding is performed by shifting the pulse train by one slot at each clock cycle and comparing it with all available codewords.

B. Chirp Spread Spectrum (CSS) [10]

Chirp Spread Spectrum (CSS) is a wireless communication modulation technique that spreads a signal over a wider bandwidth than the original signal by using a chirp signal that increases its frequency over time. CSS is used in several types of LPWA systems, such as LoRa and ELTRES, because it has a high tolerance to noise and interference, and can achieve long-distance communication. In [11], [12] and in this paper, we use a chirp pulse for each pulse to improve the reception sensitivity of APCMA, which we call CSS-APCMA.

C. Current Issue of APCMA

In APCMA, data is represented by codewords, and the lengths of the codewords are proportional to the time slot width. Since the time slot width is generally equal to the pulse width, an increase in pulse width results in longer message lengths and causes a decrease in bit rate. In CSS modulation, the relationship between noise tolerance and pulse width is determined by a parameter called the spreading factor (SF) and when the SF value is increased by one, the pulse width becomes twice as long. This causes the messages to become extremely long for large SF values.

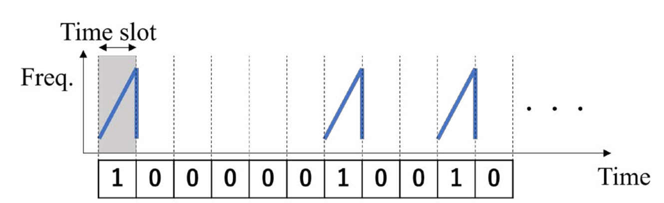
III. Throughput Improvement of APCMA

To make the APCMA scheme more suitable for more IoT applications, it is necessary to shorten codeword lengths and improve throughput. In this section, we propose a method to shorten the message length by improving the resolution on the time-axis using sub-slots. Additionally, two techniques, frequency offset detection, and sliding window, are introduced into the CSS demodulation process to accurately detect which sub-slots have the rising edge of the chirp pulse.

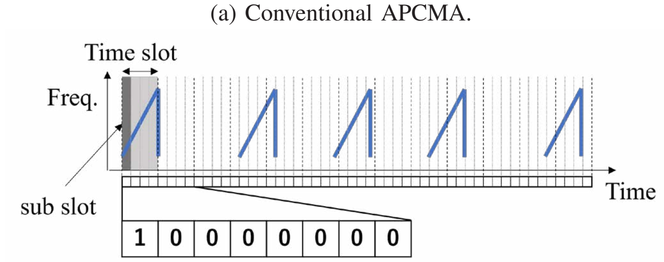
A. Shortening the Length of APCMA Messages using Sub-Slots

Fig. 2 shows the difference between the conventional APCMA and the proposed method, where each blue triangle is a pulse represented by a chirp pulse. In Fig. 2a, the pulse width and the time slot width are the same, and the codeword is represented by assigning "1" to the slot containing a pulse, and a "0" to the other slots. In the proposed method shown in Fig. 2b, a time slot is divided into even finer sub-slots and a codeword is represented by assigning a "1" to the sub-slot containing the rising edge of the pulse and a "0" to the others.

The pulse interval of the proposed method consists of one guard-slot of one time-slot width and the same number of sub-slots as the pulse interval of conventional APCMA. The guard slots are provided to prevent the pulse interval from becoming too short, ensuring that it is always greater than or equal to the time-slot width. For example, in Fig. 2a, one time slot is divided into four sub-slots, and the interval between the first and second pulses is nine sub-slots. This is derived from the sum of the guard slot, which has a width of one time slot



(a) Conventional APCMA.



(b) Proposed method.

Fig. 2: Comparison of conventional APCMA and the proposed method.

(equal to 4 sub-slots), and the 5 sub-slots, which is the interval between the 1st and 2nd pulses in the conventional APCMA.

Therefore, the length of the message in the proposed method is determined by the sum of three factors: the total length of the pulse, the total length of the guard slot, and the same number of sub-slots as the sum of pulse intervals for conventional APCMA. If the number of pulses is N_p and the width of each pulse is W_p, the total length of the pulses is simply expressed as $N_p \times W_p$. Since the guard slot is placed after every pulse except the last pulse of the message, and it has the same length as the pulse width, the total length of the guard slots is $(N_p - 1) \times W_p$. The third factor can be calculated as C-Np[sub-slot], where C is the message length of conventional APCMA. Therefore, if the sub-slot width is W_{ss}, the entire message length is expressed as follows:

$$L_{msg} = (2N_p - 1) \times W_p + (C - N_p) \times W_{ss} \tag{1}$$

In the conventional APCMA, one parameter combination allowed encoding 12-bit data using a 5-pulse code with C=12293. Using this code, the transmission bit rate of the two schemes can be calculated as shown in Fig. 3. Assuming that the bandwidth is 250 kHz, the parameters are set as $W_{ss} = 8\ \mu$s and $W_p =$ SF/250k. The figure shows that the proposed method can transmit at bit rates up to 820 times higher than conventional APCMA and that the bit rate does not decrease significantly as SF increases.

B. Frequency Offset Detection [13]

Conventional APCMA only needs to detect which time slot contains the pulse, i.e., whether the power of the chirp signal is above a threshold in the time slot. On the other hand, the proposed method needs to detect more precisely which sub-slot contains the rising edge of the chirp pulse. To achieve this, we introduce a frequency offset detection technique into the CSS demodulation process. Fig. 4 shows the frequency offset of the chirp signal with the blue arrow. In the proposed method, pulses may be received at positions shifted from the time slot as shown in the figure. By detecting the frequency offset, it is possible to detect how much the pulse is shifted with respect to the time slot and to determine which sub-slot the pulse's rising edge is in.

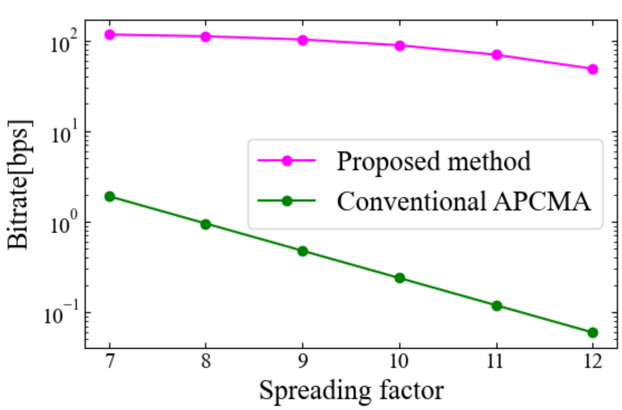


Fig. 3: Comparison of the theoretical bit rates between conventional APCMA and proposed method.

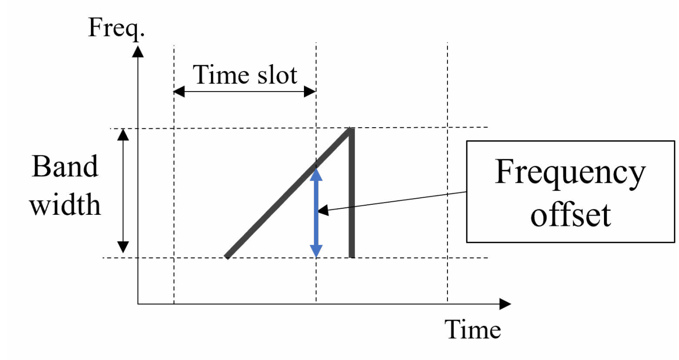


Fig. 4: Frequency offset.

Before explaining the frequency offset detection method, we explain the demodulation process of CSS. The demodulation process consists of the following three steps: Dechirp, FFT, and Compare with a threshold.

1) Dechirp: The received signal in one time-slot is multiplied by the conjugate of the chirp signal.
2) FFT: The dechirped signal is converted to the frequency domain by FFT.
3) Compare with threshold: The maximum power of the frequency domain signal is compared with the threshold.

The frequency offset can be determined in the third step as the index at which the power peak appears. A pulse shifted with respect to a time slot, as shown in the Fig. 4, will show the same frequency offset in two consecutive time slots. If two pulses are received in one time-slot, not perfectly but slightly misaligned, multiple power peaks will appear in the FFT results of Step 2. This allows the transmission bit rate to be increased while maintaining the advantage of collision tolerance of the APCMA method.

C. Sliding Window

LoRa uses a modulation scheme derived from CSS that communicates using the frequency offset as information, but it

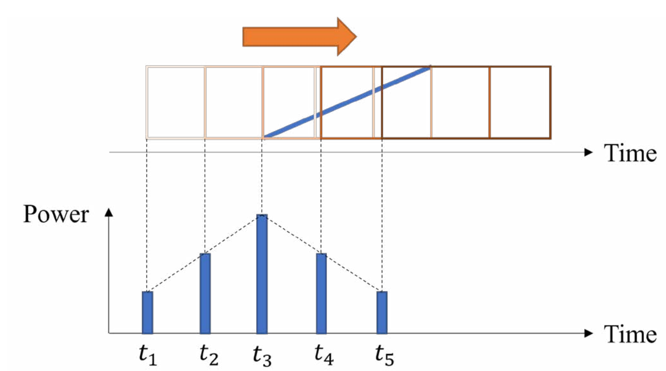


Fig. 5: Sliding window.

has the disadvantage of being vulnerable to packet collisions. To solve this problem, [5] introduces a sliding window in the LoRa demodulation process to decode collided packets. Normally, in the demodulation process of LoRa and CSS modulation, Dechirp and FFT are performed for each time slot, but with the sliding window, the FFT window is shifted by a length smaller than the width of the time slot. Fig. 5 illustrates how the sliding window works. The upper half of the figure shows how the FFT window shifts along the time axis, and the lower half shows the magnitude of the power detected in each FFT window. As shown in the figure, the magnitude of power detected in each FFT window varies, and there is a time when the highest power can be detected.

This feature is very useful in APCMA, which is asynchronous in communication and allows efficient power detection for chirp pulses that were received misaligned to the time slot. In the proposed method, the sliding window helps the demodulation process because sufficient power must be present in the FFT window to accurately detect the frequency offset.

IV. Implementation and Experiment

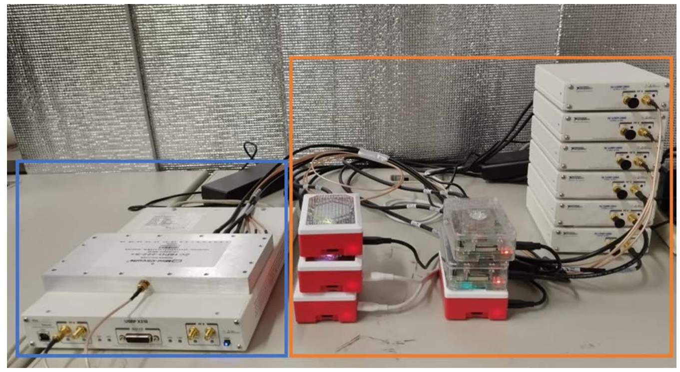
A. Implementation

We implemented the receiver and the transmitters using the proposed method on software-defined radio. Fig. 6a shows our experiment environment. We implemented six transmitters and one receiver. The receiver consists of a USRP X310 and a Linux PC(Fig. 6b), and each of the six transmitters consist of a USRP 2900 and Raspberry Pi 4(Fig. 6c). Due to the radio laws in Japan, all devices are connected via coaxial cables, and the outputs of the transmitters are mixed by a mixer before being input to the receiver.

The default settings for both the transmitters and receiver are as follows.

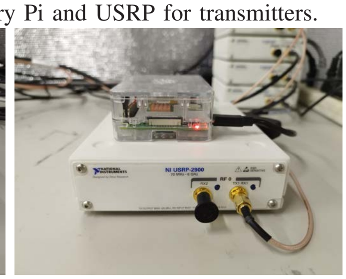
- Center frequency: 920 MHz
- Bandwidth: 250 kHz
- SF: 7
- Payload: 8 bit
- Number of Tx: 6
- $W_p = 2^{\mathrm{SF}}/\mathrm{Bandwidth} = 512\mu\mathrm{s}$

Fig. 7 and Fig. 8 show the signals of the conventional APCMA and the proposed method, respectively, observed with a real-time spectrum analyzer. The horizontal axis represents time, while the vertical axis represents frequency. It should be noted that, in this particular case, the payload is 4 bits and the transmitted data is 0b0100. From the figures, it can be observed that the waveform of each chirp pulse follows a pattern of linearly increasing frequency. Furthermore, the message length of the conventional APCMA is 27.1ms, while the proposed method achieves a message length of 10.3ms, indicating a reduction in message length is possible with the proposed method.



(a) Experiment environment. Left: USRP X310 for receiver and mixer. Right: 6 pairs of Raspberry Pi and USRP for transmitters.

(b) Receiver(Connected to PC).



(c) Transmitter.

Fig. 6: Implemented devices.

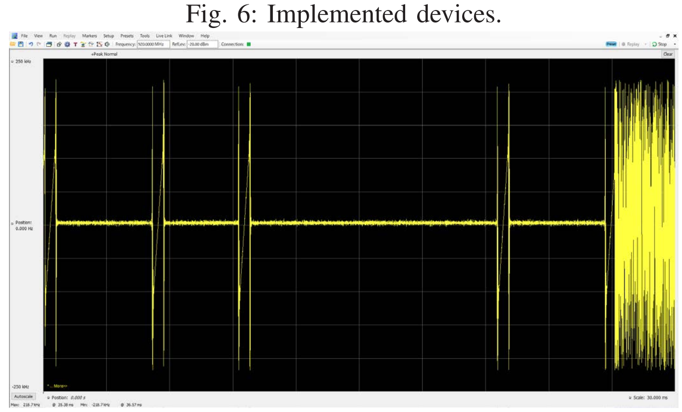


Fig. 7: The signal of the conventional APCMA. (50 kHz/div, 3 ms/div)

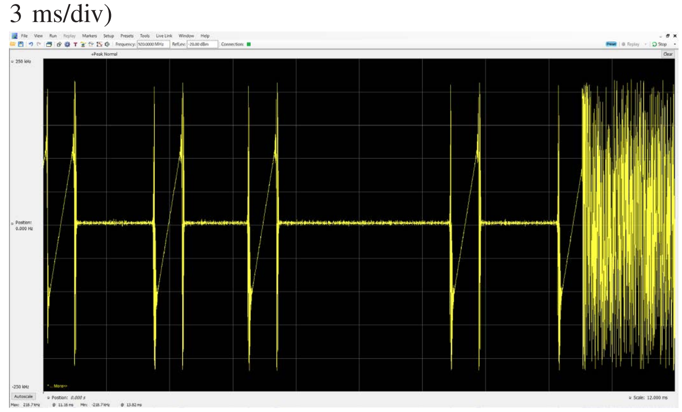


Fig. 8: The signal of the proposed method. (50 kHz/div, 1.2 ms/div)

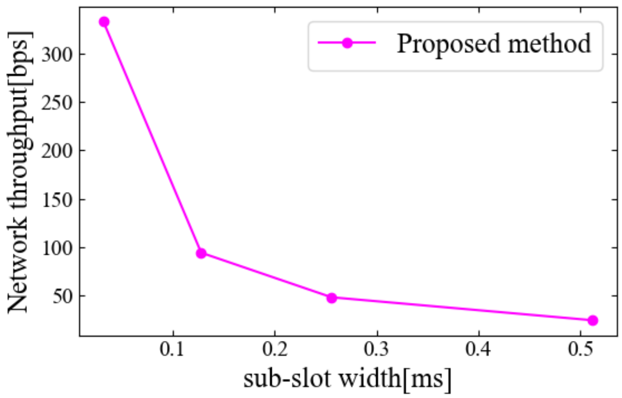


Fig. 9: Network throughput of the proposed method with fixed message duty cycle at 20%.

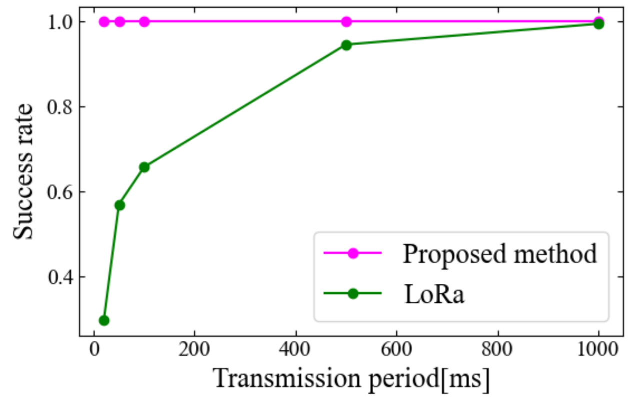
B. Confirmation of Throughput Improvement

We conducted an experiment to confirm the throughput improvement of the proposed method. In this experiment, the message duty cycle was fixed at 20% and we measured the network throughput. The result is shown in Fig. 9. The measurement point on the far right indicates the performance of conventional APCMA since the sub-slot width is equal to W_p at 512 μs. As can be seen from the figure, network throughput increases exponentially with decreasing sub-slot width, with a maximum improvement of about 14 times that of the conventional APCMA. This is because the proposed method shortens the message length, which allows the transmission interval to be shortened when the message duty cycle stays the same.

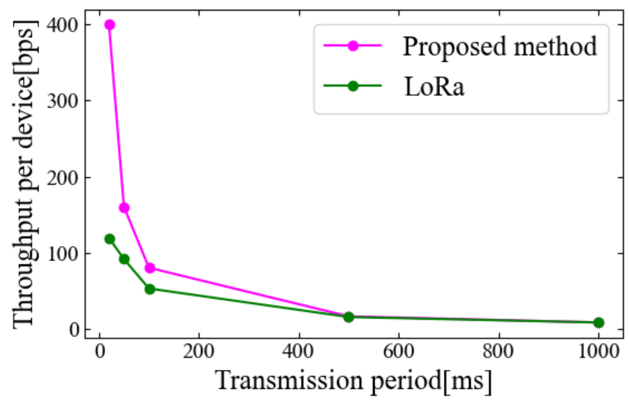
C. Comparison with other LPWA

In this section, we compare the performance of the proposed method and LoRa, using the program introduced in [14] as an implementation of LoRa. Fig. 10a shows the percentage of packets received out of those sent by the proposed method and LoRa in a high-density environment. The transmission period on the horizontal axis is varied to change the load on the network. At the point where the transmission period is 20 ms, the total packets per second is 300, which makes this experiment a very dense environment compared to the experiment in [3]. As can be seen from the figure, the proposed method maintains a 100% success rate even though the network is busy, while the success rate of LoRa decreases as the transmission period decreases from right to left on the graph and the environment becomes denser. Reference [12] shows that the packet error rate of conventional APCMA is lower than LoRa, and this experiment shows the same features in the results. This shows that we were able to achieve our APCMA throughput improvement while maintaining collision tolerance.

We also measure the effective throughput per device, and the result is shown in Fig. 10b. The throughput of the proposed method is about 3.3 times higher than that of LoRa at maximum. This indicates that the proposed method can perform better than LoRa in a Massive IoT environment.



(a) Comparison of success rate.



(b) Comparison of effective throughput per device.

Fig. 10: Comparison proposed method with LoRa in a high-density environment.

The air time, which means the time when the radio wave is actually emitted, was observed to be 10.37 ms for LoRa and 2.56 ms for the proposed method. Thus, it can be also confirmed that the proposed method is superior to LoRa both in terms of radio resource utilization efficiency and low power consumption.

V. Conclusion

This paper proposed a method to improve the throughput of APCMA, implement transmitters and a receiver, and evaluate the performance of the proposed method by experiment. According to the experiment, the proposed method can improve the throughput of APCMA by about 14 times compared to conventional APCMA. Furthermore, we confirmed that the proposed method can achieve a higher success rate than LoRa in a high-density environment, and has advantages in terms of radio resource efficiency and power consumption. In the future, we will conduct experiments using more transmitters to evaluate the performance of this method in higher-density environments and in long-distance communications.

REFERENCES

[1] F. Peper, K. Leibnitz, C. Tanaka, K. Honda, M. Hasegawa, K. Theofilis, A. Li, and N. Wakamiya, "High-Density Resource-Restricted Pulse-Based IoT Networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 1856–1868, Dec. 2021.

[2] K. Leibnitz, F. Peper, K. Theofilis, M. Hasegawa, and N. Wakamiya, "Evaluating Multiple-Access Protocols: Asynchronous Pulse Coding vs. Carrier-Sense with Collision Avoidance," in *Mobile and Ubiquitous Systems: Computing, Networking and Services*, T. Hara and H. Yamaguchi, Eds. Cham: Springer International Publishing, 2022, pp. 693–706.

[3] K. Honda, A. Nakamura, F. Peper, K. Leibnitz, N. Wakamiya, and M. Hasegawa, "Performance Evaluation of CSS-APCMA by Experiments Using 500 Devices for Massive IoT," in *International Symposium on Nonlinear Theory and Its Applications*, 2022, pp. 343–346.

[4] K. Honda, F. Peper, A. Nakamura, A. Li, Y. Ito, K. Leibnitz, K. Theofilis, N. Wakamiya, and M. Hasegawa, "Design and Implementation of Pulse-Based Protocol with Chirp Spread Spectrum for Massive IoT," in *2021 20th International Symposium on Communications and Information Technologies (ISCIT)*, Oct. 2021, pp. 172–175.

[5] Z. Xu, P. Xie, and J. Wang, "Pyramid: Real-Time LoRa Collision Decoding with Peak Tracking," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–9.

[6] Z. Yujie and R. Sivakumar, "Challenges: Communication through silence in wireless sensor networks," in *Annual International Conference on Mobile Computing and Networking, MOBICOM*, Aug. 2005, pp. 140–147.

[7] N. Wakamiya, K. Leibnitz, F. Peper, and M. Hasegawa, "Evaluation and optimization of asynchronous pulse code multiple access," *Nonlinear Theory and Its Applications, IEICE*, vol. 13, no. 2, pp. 318–323, 2022.

[8] C. Tanaka, F. Peper, and M. Hasegawa, "Application of APCMA protocol to power packet networks for multiplexing power packet transmissions," *Nonlinear Theory and Its Applications, IEICE*, vol. 11, no. 4, pp. 433–445, 2020.

[9] C. Tanaka, K. Honda, A. Li, F. Peper, K. Leibnitz, K. Theofilis, N. Wakamiya, and M. Hasegawa, "Performance evaluation of pulse-based multiplexing protocol implemented on massive IoT devices," *Nonlinear Theory and Its Applications, IEICE*, vol. 12, no. 4, pp. 726–737, 2021.

[10] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, 2016, pp. 1–5.

[11] K. Honda, A. Nakamura, F. Peper, K. Leibnitz, K. Theofilis, N. Wakamiya, and M. Hasegawa, "Performance Evaluation of CSS-APCMA by Experiments using 500 Transmitters," in *The 2022 IEICE General Conference*, Mar. 2022.

[12] A. Nakamura, K. Honda, F. Peper, K. Leibnitz, N. Wakamiya, and M. Hasegawa, "Performance evaluation of CSS-APCMA for massive IoT using GNU Radio/USRP," *IEICE Communications Express*, vol. 12, no. 4, pp. 120–125, 2023.

[13] N. Zhao, D. Zhao, Y. Wang, X. Tang, X. Zhao, Y. Zhang, and Y. Zhao, "Frequency Offset Estimation of Chirp Modulation," in *2020 International Conference on Wireless Communications and Smart Grid (ICWCSG)*, 2020, pp. 76–80.

[14] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, "An Open-Source LoRa Physical Layer Prototype on GNU Radio," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, May. 2020, pp. 1–5.