

Chitchat: Efficient and Reliable Decoding of Two-transmitter Superimposed Signals for IoT

Wen Cui, *Student Member, IEEE*, Chen Liu, and Lin Cai, *Fellow, IEEE*

Abstract—The need for wireless communication is growing at an unprecedented pace, making the wireless spectrum at a premium. To use the spectrum more efficiently, a promising solution was proposed to enable two concurrent users to transmit their signals in the same frequency at the same time, and then decode the superimposed signal. In current systems, the superimposed signal is decoded by Successive Interference Cancellation (SIC) that requires strict power control upon each individual user. However, this requirement is infeasible for many IoT devices that are heterogeneous and often low-cost. For other superimposed signal decoding technologies that require no power control, a reliable performance can be only achieved by introducing repetitive information in each transmission, which in turn reduces the spectrum efficiency. In this paper, we introduce Chitchat, a new solution to decode the superimposed signal from two concurrent transmitters without any power control nor repetitive transmissions. Chitchat presents a rotation-code based idea to provide both the diversity gain and the coding gain, so that it can achieve high reliability while preserving the spectrum efficiency. We implement Chitchat on a software-defined radio platform, and evaluate its performance in various scenarios.

Index Terms—Efficient Communications and Networking, Device-to-Device Communication, Wireless Communication, Spectrum Efficiency, Superimposed Signals.

I. INTRODUCTION

Wireless communications are growing at an unprecedented pace, considering the enormous interests of Internet-of-Things (IoT), with important applications including smart home, smart building, retail business, and machine to machine (M2M) communication. Since the wireless spectrum is at a premium, how to improve the spectrum efficiency becomes a major challenge. Traditionally, dedicated wireless resources, such as the time/frequency/space/code domain resource, are allocated orthogonally to each wireless user by using Carrier Sense Multiple Access (CSMA), Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), etc. [1] showed that the superposition coding reaches the capacity limit of two-transmitter Gaussian broadcast channel, and [2]

Manuscript received December 22, 2020; revised February 21, 2021 and March 26, 2021; accepted April 22, 2021. Date of publication XXX XX, 20XX; date of current version XXX XX, 20XX. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), Compute Canada, and China Scholarship Council (CSC). (*Corresponding author: Lin Cai.*)

Wen Cui, and Lin Cai are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada. Email: wencui@uvic.ca and cai@ece.uvic.ca.

Chen Liu is with the Department of Information Science and Technology, Northwest University, Xi'an, Shaanxi 710127, China, and also with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada. Email: liuchen@nwu.edu.cn.

Copyright (c) 2021 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

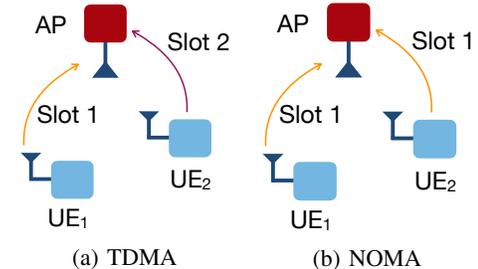


Fig. 1: Time-division multiple access v.s. Non-orthogonal multiple access.

proved that superposition coding can achieve a higher-rate region than orthogonal schemes. When we investigate the broadcast channel in a reciprocity manner, concurrent transmissions from two transmitters to one receiver, if designed appropriately, can also achieve higher-rate region than orthogonal schemes, which leads to the increasing interests in Non-Orthogonal Multiple Access (NOMA), by allowing more than one users to transmit signals in the same frequency at the same time, and then decode the superimposed signals at the receiver. As a toy example shown in Fig. 1, NOMA enables concurrent transmissions which outperforms TDMA that requires two time slots to complete the transmission of two UEs. Most NOMA implementations are for two-transmitter cases [3], [4], [5], which is also the focus of this paper.

Decoding the superimposed signal for NOMA can be mainly divided into two categories: signal-level approaches and modulation-level approaches. Both focus on decoding two-user superimposed signals. For the signal-level approaches, such as Successive Interference Cancellation (SIC) [6], [7], [8], they rely on dedicated infrastructure to guarantee strict power control, which may not be feasible for low-cost and heterogeneous IoT devices. The modulation-level approaches, such as SigMix [9] and NCMA [10], [11], [12], can decode the superimposed signals from the modulation domain without power control, desirable for IoT scenarios. Although promising, they are inherently limited, by design, in boosting the spectrum efficiency and network throughput: the modulation-level approaches need to either increase the diversity gain or the coding gain for a reliable performance by adding repetitive symbols in each transmission (e.g. SigMix requires to transmit each signal twice). However, the repetitive transmission in turn affects spectrum efficiency negatively. Moreover, their error recovering capability is constrained by the number of repetitive symbols [10], [11], [12]. When more errors happen, their error recovering performance would be severely impaired, resulting in poor reliability.

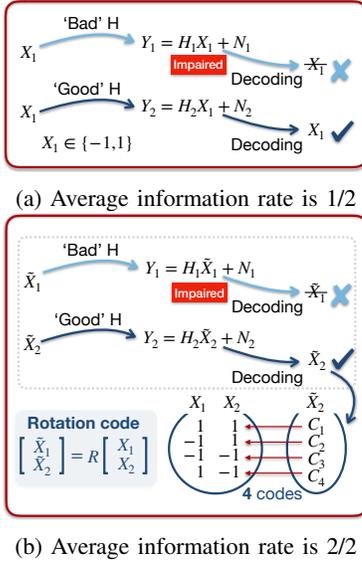


Fig. 2: Illustration of the rotation code.

In this work, we introduce Chitchat, an effective encoding/decoding system in modulation-level for concurrent transmissions from two transmitters. In Chitchat, rather than requiring adding repetitive symbols or signal copies, we enable decoding the superimposed signals from the original symbols directly. We are inspired by the *rotation code* [13] which has been used for the single transmitter communication. The intuition behind this is that the rotation code can exploit the degrees of freedom available in the wireless channel more effectively, i.e., a coding gain can be obtained beyond the diversity gain. Hence, both the reliability and spectrum efficiency can be achieved.

To see the basic idea of the rotation code clearly, we consider a single transmitter is transmitting signals to a receiver through two consecutive transmissions. As an illustrative example shown in Fig. 2. Using blind retransmission, the same symbol is transmitted twice to ensure high reliability, as shown in Fig. 2(a), so the throughput will be halved. In contrast, with the rotation code, the transmitter will transmit two different encoded symbols (i.e., \tilde{X}_1 and \tilde{X}_2) over two transmissions. The throughput can be maintained to two over two symbol durations, which can improve the spectrum efficiency compared to blindly retransmissions. More importantly, its decoding procedure jointly considers the two received symbols from two transmissions (Y_1 and Y_2) and employs a combining decoding approach. This combining decoding approach actually provides a more reliable performance, since each encoded symbol contains two wanted symbols, X_1 and X_2 . In this case, even if one of the received symbols, say Y_2 , is impaired, we can still successfully decode both wanted symbols X_1 and X_2 , as shown in Fig. 2(b) to maintain the throughput of two symbols per symbol duration. There is thus an opportunity to adopt the rotation code in concurrent transmissions to achieve a reliable decoding performance and preserve the spectrum efficiency.

However, the rotation code was invented for the single transmitter communication and cannot be adopted directly into the concurrent transmissions due to the following challenges.

- First, although the rotation code helps to improve the spectrum efficiency and reliability, **the code size and the number of constellation points are increased exponentially, making it difficult to decode superimposed signals.** Considering the previous example again, there are 4 codes and 4 corresponding constellation points for BPSK modulation after using the rotation code. However, when two transmitters transmitting signals concurrently, the code size and the number of constellation points will be increased to $4^2 = 16$. When the number of constellation points increases, the system will face fundamental difficulties in obtaining clearly distinguishable constellation points at the receiver side which are essential for demodulation [9].
- Second, for a concurrent transmission, we observe that the distinguishable constellation points are mainly depend on the phase difference γ between transmitters. Furthermore, when using the rotation code, there are γ_1 and γ_2 from the two consecutive concurrent transmissions. Then, the difference between γ_1 and γ_2 (denoted as $\Delta\gamma$) is the key factor to obtain distinguishable constellation points. However, due to the dynamic environment and hardware imperfection in practice, γ varies and is uncontrollable at each concurrent transmission. Therefore, **it is non-trivial to guarantee a $\Delta\gamma$ for a reliable decoding performance by directly manipulating γ_1 and γ_2 at each concurrent transmission, respectively.**

To deal with all the above challenges, we propose the following solutions:

- For the first time, we understand the potential of using the rotation code in the concurrent transmissions to substantially improve the spectrum efficiency and achieve a reliable performance without requiring any repetitive transmissions.
- To truly implement the rotation code for concurrent transmissions in practice, we analyze the relationship between the decoding performance and the phase differences γ_1 and γ_2 . Based on that, we derive the best value of the difference between γ_1 and γ_2 (i.e., $\Delta\gamma$) to achieve a lower decoding error rate. Hence, although the number of constellation points becomes larger (e.g., 16) when using the rotation code, Chitchat can still avoid indistinguishable constellation points through the $\Delta\gamma$ for a reliable performance.
- According to the analysis result, we propose a weighted rotation code (WRC) for concurrent transmissions by creating a ‘virtual’ $\Delta\gamma$ that can be easily manipulated in practice. Specifically, instead of using two consecutive packets, we leverage the stability of γ within one packet and move γ_1 and γ_2 to one superimposed packet. So we can guarantee a suitable $\Delta\gamma$ by adding a specific weight to the transmitted symbol from one transmitter.

To summarize, we first present the benefit of using rotation code in decoding superimposed signals; second, we reveal the underlying reasons of decoding errors—indistinguishable constellation points; third, we design a newly proposed rotation code specifically for superimposed signals; last, we implement Chitchat on a software-defined radio platform and evaluate its performance across various scenarios. Our extensive experimental results demonstrate that Chitchat outperforms the state-

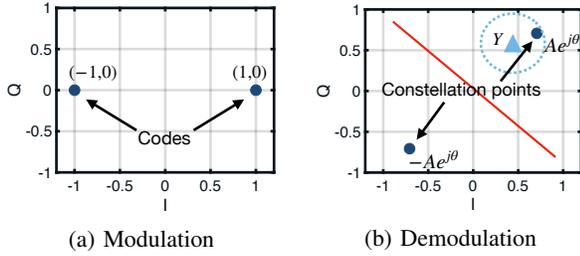


Fig. 3: The modulation and demodulation of a single transmitter communication with BPSK as an example.

of-the-art concurrent transmission scheme with lower BERs and a higher throughput gain in static and dynamic channel environments.

II. BACKGROUND

A. A Single Transmitter Communication

The wireless signal is a complex and discrete function of time. For a single transmitter, at each discrete time, the received symbol Y can be represented as

$$Y = HX + N, \quad (1)$$

where H is the wireless channel coefficient, X the transmitted symbol, and N the Gaussian noise with a zero-mean.

Specifically, if we have a binary digit S that needs to be transmitted, where $\forall S \in \{1, 0\}$, we first need to *modulate* it to a specific complex number X from the *code set* \mathbf{C} , say $\mathbf{C} = \{(1, 0), (-1, 0)\}$ for BPSK modulation. So, we will have $X = (1, 0)$ for “1” and $X = (-1, 0)$ for “0” as shown in Fig. 3(a). Then, this symbol X will transverse the wireless channel and this channel can cause signal variations on X . The signal variations can be represented by the channel coefficient H which is a complex number $Ae^{j\alpha}$. Here, A is the signal strength attenuation and α is the signal shifting in the phase part. The received symbol Y is the varied X , which will be either $Ae^{j\alpha} + N$ or $-Ae^{j\alpha} + N$ based on Eq. 1. Here, we define HC as *constellation points*. For instance, $Ae^{j\alpha}$ and $-Ae^{j\alpha}$ are constellation points for codes $(1, 0)$ and $(-1, 0)$, respectively. Note that the constellation points in the constellation map are actually the codes shifted by the channel coefficient (see as shown in Fig. 3(b)). *Demodulating* Y is to identify Y belonging to which code that the constellation point refers to. This can be achieved by calculating the shortest Euclidean distance to these constellation points as shown in Fig. 3(b).

B. The Concurrent Communication

Similar to the single transmitter case, the received superimposed signal for a concurrent communication is still a complex function at each discrete time. The major difference is that the symbols from different transmitters will transverse different wireless channels. For simplicity, we use two concurrent transmitters Alice (\mathcal{A}) and Bob (\mathcal{B}) as an example, hence, we have

$$\begin{aligned} Y &= H_{\mathcal{A}}X_{\mathcal{A}} + H_{\mathcal{B}}X_{\mathcal{B}} + N \\ &= A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}}X_{\mathcal{A}} + A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}}X_{\mathcal{B}} + N, \end{aligned} \quad (2)$$

where $H_{\mathcal{A}}$ and $H_{\mathcal{B}}$ are the channel coefficients for $X_{\mathcal{A}}$ from Alice and $X_{\mathcal{B}}$ from Bob, respectively. For the transmitter side, the modulation requires no change at this moment, but the demodulation part at the receiver has two notable changes as shown below.

- The size of the constellation points have grown from **2 to 4**, i.e., $A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}} + A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}}$, $-A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}} + A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}}$, $-A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}} - A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}}$, and $A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}} - A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}}$. As a result, to demodulate the received symbol Y , the receiver needs to calculate the Euclidean distances to **4** constellation points instead of **2**. **Generally, if we have M codes and n concurrent transmitters, we will have M^n constellation points.**
- Some of the constellation points may be *indistinguishable* to each other. For example, if $H_{\mathcal{A}} = H_{\mathcal{B}}$, then two of the constellation points are identical to each other, i.e., $-A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}} + A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}} = A_{\mathcal{A}}e^{j\alpha_{\mathcal{A}}} - A_{\mathcal{B}}e^{j\alpha_{\mathcal{B}}}$. Hence, the receiver cannot demodulate the symbol Y correctly as there is no difference between the two constellation points. **This implies that the distinguishable constellation points are essential for demodulation.**

Note that **the constellation points mainly depends on the difference between the channel coefficients (denoted as channel difference for short)**. This provides us an opportunity to obtain distinguishable constellation points by manipulating the channel difference.

III. THE DESIGN OF CHITCHAT

The design of Chitchat starts with a question: *how can we ensure a ‘good’¹ channel difference, so that the constellation points are distinguishable?* Intuitively, each transmitter can transmit the same packet multiple times, so there is a high chance to have at least one ‘good’ channel difference among all transmitted packets. Then, the packets with the ‘good’ channel difference can be used for decoding, while other packets are discarded. Apparently, this solution offers a diversity gain to achieve reliable decoding performance, but at the cost of a lower spectrum efficiency.

Chitchat aims to obtain ‘good’ channel difference without sacrificing the spectrum efficiency. We are inspired by the *rotation code* [13] used for the single transmitter communication. The purpose of the rotation code is to exploit the degrees of freedom available in the wireless channel effectively. Specifically, the rotation code can not only offer a diversity gain, but also a coding gain to ensure both the reliability and spectrum efficiency. In the traditional rotation code, the diversity gain is 2 as each symbol has been transmitted twice in two consecutive packets, and the coding gain is 3.5 dB for a Rayleigh fading channel condition. Chitchat shares the same design principle of the traditional rotation code, hence we see it is clear that both the diversity gain and coding gain can be obtained. For the diversity gain, which is also 2 for Chitchat, as we use the same diverse transmission scheme. For the coding gain, given the much higher complexity in analyzing the error

¹We use the words ‘good’ and ‘bad’ to vividly describe the situations of distinguishable constellation points and indistinguishable constellation points, and we keep the definition consistent in the following context.

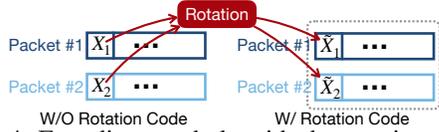
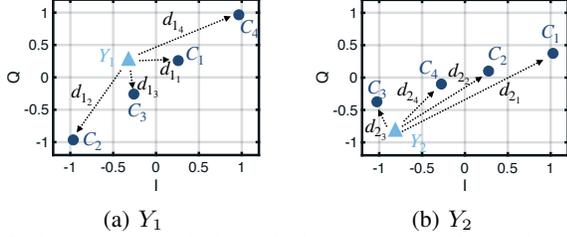


Fig. 4: Encoding symbols with the rotation code.



(a) Y_1

(b) Y_2

Fig. 5: Decoding symbols with the rotation code. In the above example, the demodulation result is C_3 , as the summation $d_{1_3} + d_{2_3}$ is the smallest.

probability boundary of a superimposed signal [9], we leave it as an important further research issue.

A. The Rotation Code in a Single Transmitter Communication

The rotation code has been used in the single transmitter communication as follows.

Encoding. Suppose X_1 and X_2 are two symbols from two consecutive packets, respectively. Without the rotation code, they will directly transverse different channels represented by H_1 and H_2 . When applying the rotation code, as shown in Fig. 4, these two symbols will be encoded to two new symbols \tilde{X}_1 , \tilde{X}_2 with a rotation matrix \mathbf{R} ,

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (3)$$

where θ is the rotation angle that belongs to $[0, 360^\circ)$. Mathematically, this encoding procedure is given by

$$\begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \cos \theta X_1 - \sin \theta X_2 \\ \sin \theta X_1 + \cos \theta X_2 \end{bmatrix}. \quad (4)$$

Then, the transmitter transmits the new symbols \tilde{X}_1 and \tilde{X}_2 through different channels H_1 and H_2 . The received symbols Y_1 and Y_2 can be represented as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix}. \quad (5)$$

For simplicity, we omit the Gaussian noise in the above equation.

Note that given $X_1, X_2 \in \{(1, 0), (-1, 0)\}$ for BPSK modulation, if we substitute X_1 and X_2 into Eq. 4, then there are 4 unique codes for each encoded symbol, \tilde{X}_1 or \tilde{X}_2 , i.e.,

$$\begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \in \left\{ \underbrace{\begin{bmatrix} (\cos \theta - \sin \theta, 0) \\ (\sin \theta + \cos \theta, 0) \end{bmatrix}}_{c_1}, \underbrace{\begin{bmatrix} (-\cos \theta - \sin \theta, 0) \\ (-\sin \theta + \cos \theta, 0) \end{bmatrix}}_{c_2}, \right. \\ \left. \underbrace{\begin{bmatrix} (-\cos \theta + \sin \theta, 0) \\ (-\sin \theta - \cos \theta, 0) \end{bmatrix}}_{c_3}, \underbrace{\begin{bmatrix} (\cos \theta + \sin \theta, 0) \\ (\sin \theta - \cos \theta, 0) \end{bmatrix}}_{c_4} \right\} \quad (6)$$

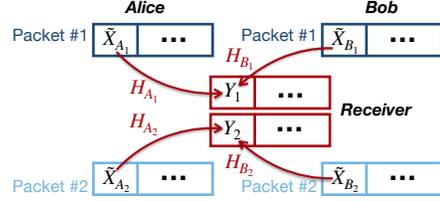


Fig. 6: The rotation code in concurrent transmissions.

Decoding. From the receiver side, these 4 codes refer to 4 constellation points in the constellation map. As shown in Fig. 5, the two received symbols Y_1 and Y_2 have different constellation maps. According to Section II, \tilde{X}_1 can be demodulated by calculating the shortest Euclidean distance from the received symbol Y_1 to these 4 constellation points. \tilde{X}_2 can be also demodulated in the same way. However, due to the dynamic environment in practice, either Y_1 or Y_2 may suffer from the ‘bad’ channel difference, resulting in an unreliable decoding performance. To solve this problem, in the design of the rotation code, the decoding procedure is to jointly consider both Y_1 and Y_2 and employ the combining approach. This combining approach is to minimize the summation of d_{1_i} and d_{2_i} ($i = 1, 2, 3, 4$) as shown in Fig. 5, which can be calculated by the maximum likelihood (ML) as

$$\hat{\tilde{\mathbf{X}}} = \arg \min_{\tilde{\mathbf{X}} = [\tilde{X}_1, \tilde{X}_2]^T} \|\mathbf{Y} - \mathbf{H}\tilde{\mathbf{X}}\|^2. \quad (7)$$

Apparently, once $\tilde{\mathbf{X}}$ can be demodulated correctly, both wanted symbols X_1 and X_2 can be decoded. Similarly, other symbols in packet #1 and packet #2 can be pairwise encoded and decoded.

Benefits. With rotation code, the two transmissions contain two different symbols (\tilde{X}_1 and \tilde{X}_2). So the average information rate is $2/2$, preserving the spectrum efficiency. More importantly, the combining decoding approach provides a higher reliability. For example, even if one of the received symbols, say Y_2 , is impaired, we can still successfully decode both wanted symbols X_1 and X_2 . We just need to **avoid the situation when both Y_1 and Y_2 are impaired.**

The above analysis reveals that for a single transmitter, **by using the rotation code, we can achieve the reliable performance and preserve the spectrum efficiency simultaneously** without transmitting the same packets multiple times.

B. The Rotation Code in the Concurrent Communication

In Chitchat, we adopt the rotation code in the concurrent communication to ensure a high reliability without sacrificing efficiency.

Consider two transmitters, Alice and Bob, transmitting signals simultaneously to one receiver. Each transmitter encodes its symbols with the rotation code described in Section III-A. Hence, as shown in Fig. 6, the received symbol Y_1 and Y_2 are superimposed symbols which can be represented as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_{A_1} & H_{B_1} & 0 & 0 \\ 0 & 0 & H_{A_2} & H_{B_2} \end{bmatrix} \begin{bmatrix} \tilde{X}_{A_1} \\ \tilde{X}_{B_1} \\ \tilde{X}_{A_2} \\ \tilde{X}_{B_2} \end{bmatrix}. \quad (8)$$

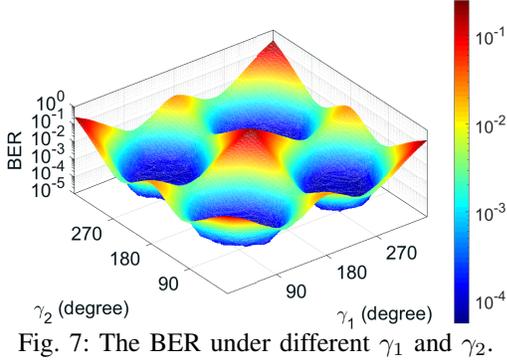
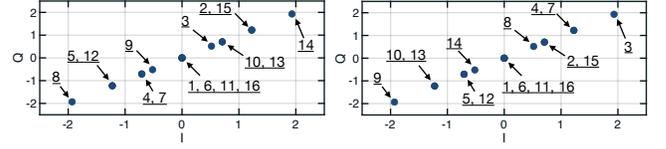


Fig. 7: The BER under different γ_1 and γ_2 .

Similar to a single transmitter, when adopting the rotation code for concurrent transmissions, it can offer a diversity gain for a reliable performance and also a coding gain to preserve the spectrum efficiency. However, recall that in the design of the rotation code, we need to avoid the situation when both Y_1 and Y_2 are failed to be demodulated. Hence, a question is raised naturally: *how can we avoid this situation happening in concurrent transmissions?*

Recall that for the concurrent transmissions, obtaining the distinguishable constellation points depends on the ‘good’ channel difference, which is essential for the demodulation. Therefore, to answer the above question, we need to understand the relationship between the decoding performance and the channel difference. Specifically, which values of the channel difference are ‘good’ for decoding and which are ‘bad’.

Understanding the errors. The channel difference contains two parts: the attenuation ratio ($\eta = \frac{|H_A|}{|H_B|} = \frac{A_A}{A_B}$) and the phase difference ($\gamma = |\angle H_A - \angle H_B| = |\alpha_A - \alpha_B|$). Manipulating the attenuation ratio requires dedicated devices for a strict power control is infeasible for low-cost IoT devices. Therefore, in Chitchat, we focus on manipulating the phase difference γ to control the channel difference. We use the bit-error-rate (BER) as the metric for decoding performance. For each received symbol, Y_1 or Y_2 , there is a corresponding phase difference, i.e., γ_1 or γ_2 . Thus, we start by understanding the relationship between BER and the phase difference (γ_1 and γ_2). Specifically, we conducted Monte-Carlo simulation to emulate all the possible values of γ_1 and γ_2 within $[0^\circ, 360^\circ]$. The rotation angle θ is fixed as 30° which has been proved optimal [14]. Here, we use one of the simulation results as an example shown in Fig. 7 to explain our observations, where the SNR for Alice and Bob are 10 dB and 15 dB, respectively. Clearly, the error peaks (i.e., the highest BER) appear at certain values of γ_1 and γ_2 , such as $\gamma_1 = 180^\circ, \gamma_2 = 180^\circ$, and $\gamma_1 = 0^\circ (360^\circ), \gamma_2 = 360^\circ (0^\circ)$. This is the ‘bad’ situation that we need to avoid in our design. One thing worth noting here is that we also tested different values of SNR for Alice and Bob, and we notice that the varying pattern of the BER is the same with arbitrary SNRs. One key feature is revealed from this observation: the BER varying pattern is dominated by the γ_1 and γ_2 , and it is independent of the SNRs. Therefore, we can ignore the influence of the SNR which cannot easily be controlled, and focus on manipulating the phase difference γ .



(a) For received symbol Y_1 .

(b) For received symbol Y_2 .

Fig. 8: The distribution of constellation points when $\gamma_1 = 180^\circ, \gamma_2 = 180^\circ$.

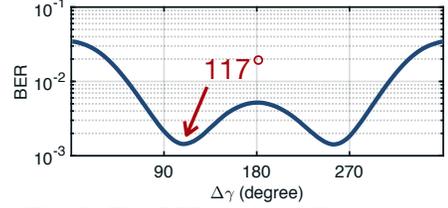


Fig. 9: The BER under different $\Delta\gamma$.

To further understand why these errors happen, we zoom into the constellation points of the received symbols. Note that since each transmitted symbol (either from Alice or Bob) has **4** possible codes after using the rotation code, the received superimposed symbol will have **16** possible codes corresponding to **16** constellation points. Taking $\gamma_1 = 180^\circ, \gamma_2 = 180^\circ$ as an example, we plot the **16** constellation points for Y_1 and Y_2 as shown in Fig. 8(a) and Fig. 8(b), respectively. Our observations are as below. The similar observation can be obtained in other cases, such as $\gamma_1 = 0^\circ (360^\circ), \gamma_2 = 360^\circ (0^\circ)$.

- **Distinguishable constellation points:** some constellation points are clearly distinguishable, such as points 8 and 14 for Y_1 , and 3 and 9 for Y_2 .
- **Close constellation points:** some constellation points are too close to the nearby constellation points, such as points 3 and 9 for Y_1 , and 8 and 14 for Y_2 .
- **Indistinguishable constellation points:** some constellation points are overlapped and merge to one point, resulting in indistinguishable cases, such as points 1/6/11/16, 2/15, 4/7, 5/12 and 10/13 for both Y_1 and Y_2 .

The above observations reveal two properties: (1) *complementary property*: the close constellation points for Y_1 are actually distinguishable for Y_2 , and vice versa; (2) *consistent property*: the overlapped constellation points are the same for both Y_1 and Y_2 . The complementary property can help the combining decoding approach, so that the close constellation points problem can be solved. But the consistent property makes the overlapped constellation points unsolvable and causes decoding errors, such as the BER peak shown in Fig. 7.

We conclude that *the main reason for decoding errors is the overlapped or close constellation points, and they are caused by a ‘bad’ difference between γ_1 and γ_2 , i.e., a ‘bad’ $\Delta\gamma$, where $\Delta\gamma = |\gamma_1 - \gamma_2|$. Thus, we need to obtain a ‘good’ $\Delta\gamma$ and avoid the ‘bad’ $\Delta\gamma$.*

The value of ‘good’ $\Delta\gamma$. According to the relationship between the BER and γ_1, γ_2 shown in Fig. 7, the BER can be considered as a function of γ_1, γ_2 , notated as $P(\gamma_1, \gamma_2)$. For simplicity, we rewrite it as $P(\Delta\gamma)$. Thus, the ‘good’ $\Delta\gamma$ that

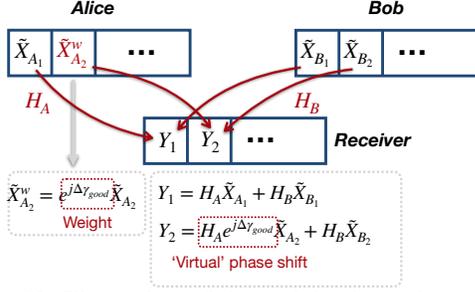


Fig. 10: Illustration of the weighted rotation code.

refers to a lower BER can be calculated by

$$\Delta\gamma_{good} = \arg \min_{\Delta\gamma \in [0^\circ, 360^\circ]} P(\Delta\gamma). \quad (9)$$

To obtain the ‘good’ $\Delta\gamma$, we further plot the BER curve under different $\Delta\gamma$ in Fig. 9, which is the statistical result from the above Monte-Carlo simulations (Fig. 7). Note that different values of γ_1 and γ_2 may result in the same $\Delta\gamma$. Hence, the BER under a particular $\Delta\gamma$ shown in Fig. 9 is an average result. Apparently, there are two valleys when $\Delta\gamma = 117^\circ$ and $\Delta\gamma = 243^\circ$, and they are symmetrical with $\Delta\gamma = 180^\circ$. Thus, the best choice of a ‘good’ $\Delta\gamma$ is centered around 117° or 243° . We choose $\Delta\gamma = 117^\circ$ in our following experiments and this value performs well. Also note that the ‘good’ value of $\Delta\gamma$ may be slightly different given different hardware and channel environments.

C. Put it together

Next, given the ‘good’ value of $\Delta\gamma$, *how can we guarantee this value in the practical system?*

In practice, the phase difference γ varies at each concurrent transmission due to the dynamic environments caused by the moving objects [15], [16], [17] and the system level noises caused by the hardware imperfections [18], [19], [20]. Recall that γ_1 and γ_2 are originally from two consecutive concurrent transmissions, so it is very challenging to guarantee a ‘good’ $\Delta\gamma$ by directly manipulating γ_1 and γ_2 , respectively.

Weighted encoding. To solve this problem, we observe that although γ is different among different superimposed packets, it is stable and measurable within a packet transmission time. Therefore, when γ_1 and γ_2 are from the same superimposed packet, we can add a weight (i.e., a phase shift) to the transmitted symbol to create a ‘virtual’ $\Delta\gamma$. Apparently, this virtual $\Delta\gamma$ can be easily manipulated to the ‘good’ value. To this end, we propose the Weighted Rotation Code (WRC) for concurrent transmissions as follows. For each transmitter, instead of encoding two symbols from two consecutive packets, we encode two consecutive symbols within one packet. For example, with the typical rotation code, Alice encodes two symbols \tilde{X}_{A_1} and \tilde{X}_{A_2} from packet #1 and packet #2 (see Fig. 6). But when employing our proposed weighted rotation code, Alice will encode two consecutive symbols \tilde{X}_{A_1} and \tilde{X}_{A_2} from the same packet as shown in Fig. 10. Bob will encode the symbols in the same way. Then, Alice adds a weight $e^{j\Delta\gamma_{good}}$ to its second symbol, i.e., $\tilde{X}_{A_2}^w = e^{j\Delta\gamma_{good}} \tilde{X}_{A_2}$.

For the receiver side, Y_1 and Y_2 are two consecutive symbols in the same superimposed packet. Thus, we have $\gamma_1 = \gamma_2 = |\angle H_A - \angle H_B|$. However, since the second encoded symbol from Alice is a weighted symbol, this weight can be considered as a ‘virtual’ phase shift to the channel coefficient H_A , so we will have a new channel coefficient $e^{j\Delta\gamma_{good}} H_A$. Therefore, γ_1 and γ_2 can be rewritten as

$$\begin{aligned} \gamma_1 &= |\angle H_A - \angle H_B| \\ \gamma_2 &= |(\angle H_A + \Delta\gamma_{good}) - \angle H_B|. \end{aligned} \quad (10)$$

Apparently, by doing so, we will have a ‘virtual’ $\Delta\gamma = |\gamma_1 - \gamma_2| = \Delta\gamma_{good}$. That is to say, we can now guarantee a ‘good’ $\Delta\gamma$ by adding a specific weight ($\Delta\gamma_{good}$) to the second symbol from one transmitter (Alice). Note that the remaining symbols within the same packet from Alice and Bob can be pairwise encoded in the same way, such as the third and fourth symbols, and the fifth and sixth symbols, etc.

Smart decoding scheme. The decoding scheme in Chitchat is similar to the decoding procedure in the rotation code introduced in Section III-A. Suppose there are $2n$ symbols in each packet, any two consecutive symbols from Alice and Bob, i.e., $\tilde{X}_{A_i}, \tilde{X}_{A_{i+1}}, \tilde{X}_{B_i}, \tilde{X}_{B_{i+1}}$, where $i \in [1, 2n]$ can be demodulated by the combining approach as below. Note that if there are odd number symbols, we will pad a 0 at the end of the packet.

$$\begin{aligned} \arg \min_{\tilde{X}_{A_i}, \tilde{X}_{A_{i+1}}, \tilde{X}_{B_i}, \tilde{X}_{B_{i+1}}} & [Y_i - (H_A \tilde{X}_{A_i} + H_B \tilde{X}_{B_i})] \\ & + [Y_{i+1} - (e^{j\Delta\gamma_{good}} H_A \tilde{X}_{A_{i+1}} + H_B \tilde{X}_{B_{i+1}})]. \end{aligned} \quad (11)$$

Note that the channel coefficient H_A (H_B) slightly varies within each packet, but the two consecutive symbols have the similar channel coefficients. Thus, it is safe to use H_A (H_B) for both \tilde{X}_{A_i} (\tilde{X}_{B_i}) and $\tilde{X}_{A_{i+1}}$ ($\tilde{X}_{B_{i+1}}$) in Eq. 11. Now, we have the demodulation results $\tilde{X}_{A_i}, \tilde{X}_{A_{i+1}}, \tilde{X}_{B_i}, \tilde{X}_{B_{i+1}}$, and then the original wanted symbols $X_{A_i}, X_{A_{i+1}}, X_{B_i}$ and $X_{B_{i+1}}$ can be decoded according to the possible **16** codes.

To conclude, the understandings of the superimposed signals motivate our proposed solutions. In particular, first we understand the underlying reason for decoding superimposed signals in the constellation map—indistinguishable constellation points; second, we study the phase relations between the two concurrent transmission signals, and propose to use the rotation code to avoid the indistinguishable constellation points; last, according to our understandings, we design a weighted encoding and smart decoding scheme for superimposed signals.

IV. THE EXTENSION AND PERFORMANCE ANALYSIS

A. Beyond Two Transmitters

Our design has focused on two concurrent transmitters. When more transmitters are involved, we can divide all the transmitters into several pairs and each pair contains two transmitters. Then, each pair transmits signals simultaneously, and different pairs transmit at different time slots. For example, if an Access-Point (AP) can schedule four users into two pairs by making Alice and Bob transmit concurrently first, and then Carol and David. Based on that, these two pairs of users can share the spectrum via alternately transmitting in the time

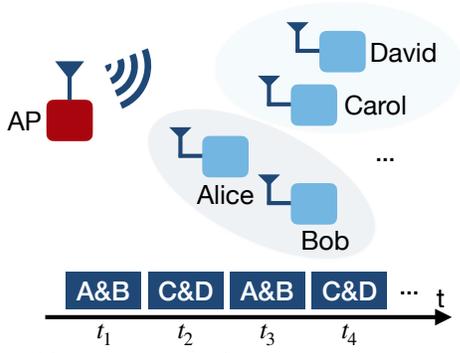


Fig. 11: An example for more than two users.

domain. To implement this idea, we need a lightweight MAC layer protocol to coordinate the users. The protocol can be similar to existing time-domain multiple access protocols, and we believe it is feasible to off-the-shelf devices. It is worth noting that the scheduling of the transmission can be implemented in a distributed or centralized manner. Specifically, a distributed scheme can be applied by having a master node that can call up a concurrent transmission from two neighbor nodes, and those who replied to the master node will be transmitting at the next time slot [21]. A centralized scheme can be applied by having a controller, such as a sink node in wireless sensor networks [22], to broadcast the controlling messages and enable two suitable end nodes to join in the transmission, and the selection process may depend on the link quality or application needs. As it is out of the scope of this paper, we leave a detailed discussion as our future work. Note that the existing modulation-level approaches for NOMA all focused on implementing 2 concurrent transmitters, such as SigMix and NCMA. While even with fine-designed devices that are capable of strict power control, the signal-level approaches, e.g., SIC, are limited to 2 transmitters due to the challenge of maintain an obvious power gap between users [3], [4], [5], [23], [24], [25], [26], [27]. Normally, the number of concurrent transmitters can be controlled in advance via grouping strategies. This is much different from the collision scenario where dozens of transmitters may naturally collide when a massive number of devices are deployed.

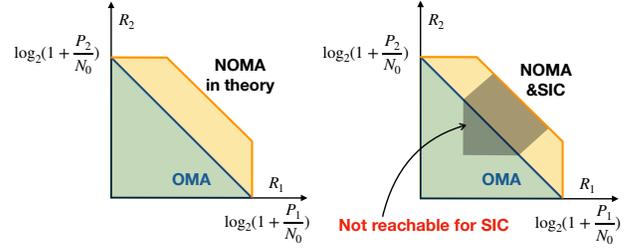
B. Analysis

Capacity regions for NOMA with Chitchat and NOMA with SIC.

The capacity region of OMA and NOMA has been studied extensively in existing work [13]. In particular, for AWGN channels and using two users as an example, the communication rate (bit per second per Hz) constrains R_1 and R_2 for user 1 and user 2 are shown below.

$$\begin{cases} R_1 \leq \log_2(1 + \frac{P_1}{N_0}), \\ R_2 \leq \log_2(1 + \frac{P_2}{N_0}), \end{cases} \quad (12)$$

where N_0 refers to the power spectral density of the white Gaussian noise, and P_1 and P_2 are the received power for user 1 and user 2, separately. Considering existing OMA solutions are sharing the spectrum with different users, we can see the



(a) NOMA with Chitchat (b) NOMA with SIC
Fig. 12: Capacity region for NOMA.

capacity region marked as the green color for OMA as shown in Fig. 12.

When it comes to NOMA, it aims at enabling concurrent transmissions and then decode the superimposed signal, which would offer a higher capacity gain than OMA,

$$R_1 + R_2 \leq \log_2(1 + \frac{P_1 + P_2}{N_0}), \quad (13)$$

also as shown in the orange region of Fig. 12. Here, we can use an numerical example to describe it more clearly. Given $\frac{P_1}{N_0} = 10\text{dB}$ and $\frac{P_2}{N_0} = 5\text{dB}$, and then we would have

$$\text{OMA} \begin{cases} R_1 = a \log_2(1 + \frac{P_1}{N_0}) = 3.46a, \\ R_2 = (1 - a) \log_2(1 + \frac{P_2}{N_0}) = 2.06(1 - a), \end{cases} \quad (14)$$

$$\text{NOMA with SIC} \begin{cases} R_1 = \log_2(1 + \frac{P_1}{N_0 + P_2}) = 1.77, \\ R_2 = \log_2(1 + \frac{P_2}{N_0}) = 2.06, \end{cases} \quad (15)$$

where $0 \leq a \leq 1$, and the sum rate of NOMA with SIC is greater than the OMA schemes.

However, the implementation of SIC in practice has a strong assumption that the concurrent signals should be separable in the power domain, and when it is not, the SNR of one user would be decreased significantly, leading to an undesirable communication performance. For example, given $\frac{P_1}{N_0} = \frac{P_2}{N_0} = 5\text{dB}$, and then we would have

$$\text{OMA} \begin{cases} R_1 = a \log_2(1 + \frac{P_1}{N_0}) = 2.06a, \\ R_2 = (1 - a) \log_2(1 + \frac{P_2}{N_0}) = 2.06(1 - a), \end{cases} \quad (16)$$

$$\text{NOMA with SIC} \begin{cases} R_1 = \log_2(1 + \frac{P_1}{N_0 + P_2}) = 0.815, \\ R_2 = \log_2(1 + \frac{P_2}{N_0}) = 2.06. \end{cases} \quad (17)$$

Note that although NOMA with SIC still has a better capacity gain in theory, but the SNR for user 1, $\frac{P_1}{N_0 + P_2}$, would be decreased to -1.1 dB. Most of communication systems cannot maintain reliable communications in such a low SNR. Therefore the dark region marked in the Fig. 12b is not reachable for SIC in practice. To conclude, NOMA schemes aim to approach the capacity boundary of the two-user case, $R_1 + R_2 \leq \log_2(1 + \frac{P_1 + P_2}{N_0})$, however, given the strong power assumption of SIC, traditional NOMA schemes can hardly reach to that boundary when the power level of two users are close to each other. But for a modulation-level superimposed signal decoding scheme, Chitchat, it requires no power control, and it has the potential to approach the theoretical capacity

region of NOMA, resulting in a better performance for low-cost devices when compared with SIC.

Spectrum efficiency/throughput. Current superimposed signal decoding approaches in modulation-level (e.g., SigMix [9] and NCMA [10], [11], [12]) either require to add one repetitive signal copy or several repetitive symbols to achieve a reliable decoding performance. Hence, their throughput upper bound is equal or similar to that of sequential transmissions. Unlike the state-of-the-art modulation-level approaches, Chitchat can decode the superimposed signal without requiring any repetitive transmissions. The theoretical throughput of Chitchat is actually doubled compared to sequential transmissions. Hence, Chitchat can truly enable concurrent transmissions in the same frequency at the same time and substantially boost the spectrum efficiency.

Complexity. At the transmitter side, Chitchat introduces a weighted encoding scheme. This scheme can be seen as a patch of the existing modulation block, and its overhead is growing linearly and limited by the number of transmitted symbols. That is to say the use of Chitchat will cause no notable extra energy consumption on the transmitter, making it friendly to IoT devices. Similarly, the receiver decodes the superimposed signal by using a maximum-likelihood solution as the existing WiFi decoder does, which causes no notable extra complexity energy consumption. The experiment of energy consumption is out of the scope of this work and we leave it as future work.

Influence by the dynamic fading channel conditions. In contrast to SIC that strictly requires a power gap between the two concurrent signals, which can be achieved by frequent channel condition inquires, Chitchat does not require any channel feedback for prior knowledge, because we use the diversity gain to avoid the indistinguishable constellation points under varies channel conditions. We have conducted experiments under different channel conditions, and we present the results in the following experiment section. In particular, the results reveal that the increased power to noise ratio does reduce BER, but there is no correlation between channel conditions of the two concurrent signals and the BER, which represents a robust performance of Chitchat to the dynamic fading channel conditions.

V. IMPLEMENTATION

Note that the modulation-level superimposed signal decoding implementation is not trivial, given the frequency and phase offsets in practical systems. Next, We implement Chitchat by using three USRPs embedded with XCVR2450 daughter boards. For each USRP, we employ an ECOM9-5500 mag-mount antenna with 9 dBi gain or an HG2458RD-SM omni-directional antenna with 3 dBi gain, and the selection of the antenna depends on the deployment requirement as the mag-mount antenna is preferred to be placed on the ground or on metal surfaces. Chitchat is built upon a recent project programmed in GNU-radio [28], which follows IEEE 802.11p standard with 5.9 GHz carrier frequency and 10 MHz bandwidth. The program is running on PCs operating on Ubuntu 16.04, and each PC processes the data streams

coming from the USRP. To focus on our design, we remove other schemes along with the Wi-Fi protocol for simplicity, such as scrambling, channel coding, etc.

Practical issues. Decoding the superimposed signals for NOMA demands symbol-level time synchronization, frequency synchronization and phase synchronization [29], which can guarantee different transmitters transmit signals in the same frequency at the same time. For the time synchronization, fortunately, there have been extensive studies recently to provide the symbol-level time synchronization with low-cost solutions, such as by using GPS clocks in outdoor scenarios [30], and by using WiFi devices in indoor scenarios [21], [31]. For simplicity, we use a central clock as a solution in this paper for time synchronization, and focus on the other unsolved synchronization problems in the frequency and phase domains. In practice, due to the differences between oscillators and signal detection failures [20], three types of signal offsets exist, including the Carrier Frequency Offset (CFO), the Sampling Frequency Offset (SFO) and the Sampling Time Offset (STO). These offsets will influence the required synchronization accuracy negatively. In Chitchat, we employ and implement a precise offsets tracking scheme at the receiver to correct these offsets. Specifically, this mechanism introduces orthogonal preambles in the packet header to track the offsets of each transmitted signal. After that, the detected offsets can be added up to the channel coefficient for further demodulation.

The transmitters. Two USRP N210s connect to a PC through a Gigabit Ethernet router as two transmitters, Alice and Bob. For the time synchronization, each USRP is connected to a central clock (i.e., NI CDA-2990) via SMA cables. Each transmitted packet has a total length of 1500 bytes payload and a preamble, which follows the Wi-Fi standard. Specifically, after the BPSK modulation, every two consecutive symbols within the packet from Alice are encoded by the proposed weighted rotation code, and Bob follows the rotation code without weighting. We then transform the frequency domain BPSK symbols into the time domain samples via the IFFT. The time domain samples are ordered by the parallel-to-serial converter and passed through a DAC converter, resulting in the baseband OFDM signal which is then upconverted to the carrier frequency.

The receiver. One USRP N210 connects to another PC through Ethernet cable as the receiver. The received superimposed signal is downconverted to baseband and filtered to remove the high-frequency components. Then, the ADC converter samples the signal. The receiver detects the beginning of a superimposed packet by extracting the long training sequence, and tracks CFO, STO and SFO. These time samples are serial-to-parallel converted and passed through the FFT. The FFT output is passed through the demodulator. Then, the proposed smart decoding scheme is implemented to decode the original data. Note that we omit some detailed descriptions, e.g., the cyclic prefix, due to space limit. The block diagram is shown in Fig. 13.

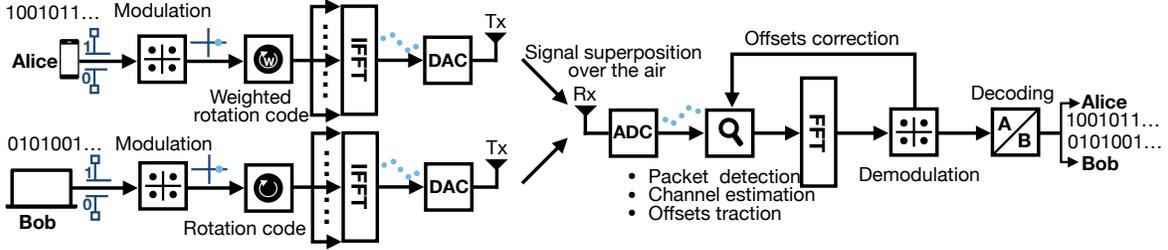


Fig. 13: The block diagram of Chitchat.

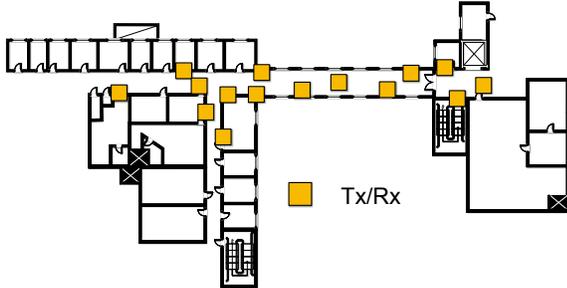


Fig. 14: The deployment layout.

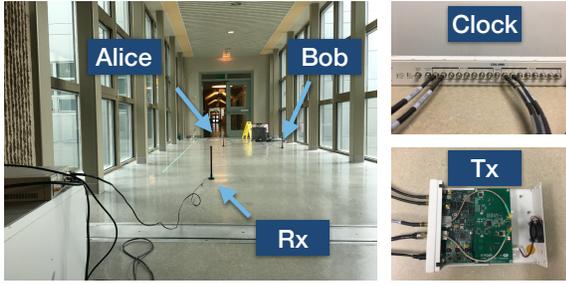


Fig. 15: Experimental setup.

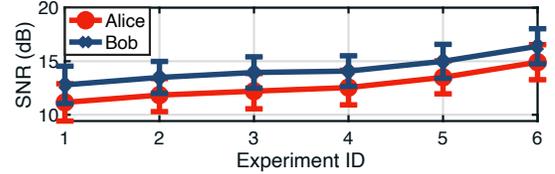
VI. EVALUATION

We conduct experiments in an office building including labs, offices and corridors as shown in Fig. 14. The transmitted signal may transverse a metal/glass door which contributes to a Non-Line-of-Sight (NLoS) a communication scenario. The distances among the three nodes vary from 1 m to 8 m according to the size of each region. Fig. 15 shows the setup in one of the corridors. The sending rate is 1 pkt/s and we collect 500 concurrent transmissions in each region.

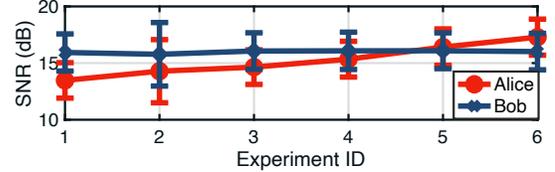
Compared schemes. We compare the performance of Chitchat with two state-of-the-art NOMA schemes.

- **PhyCode.** PhyCode [32] requires no change at the transmitter side, assuming the constellation points are distinguishable when decoding the superimposed signals.
- **SigMix.** SigMix [9] requires transmitting two same signal copies within each packet and adds a phase shift between the two copies. It then decodes the superimposed signal by selecting a good copy and discards the other for a reliable decoding performance. Note that SigMix uses a ‘rotation code’ to add this phase shift, but the ‘rotation code’ is not the same rotation code as in our design.

Note that we use the same offset tracking mechanism as SigMix and PhyCode to address the signal offset problem,



(a) Different SNRs.



(b) Different SNR ratios.

Fig. 16: The setting of the signal strength.

i.e., CFO, SFO, and STO. Compared with existing NCMA schemes [10], [11], [12] that focused on homogeneous devices, the offset tracking mechanism we are using can handle the diverse offsets from heterogeneous devices, which is more desirable for IoT devices.

Metrics. We employ the following metrics:

- **Bit Error Rate (BER):** The BER refers to the raw BER without considering channel error coding.
- **Packet Reception Rate (PRR):** The ratio of successfully received packets. We use the bit error information from all the packets received by the USRP and then apply the CRC-32 algorithm along with convolutional codes at 1/2 code rates to calculate the PRR.
- **Throughput gain:** the ratio of throughput in Chitchat to the throughput in the existing scheme (i.e., SigMix or Phycode).

A. Impact of the Signal Strength

To demonstrate that Chitchat can perform well without power control, we evaluate the impact of both the SNR and the SNR gap of two transmitted signals. To focus on the influence of SNR and SNR gap, we use 30 dB attenuators to connect two transmitters to the receiver, which can emulate a stable wireless channel and avoid the impact of the dynamic environment and uncontrollable wireless. This experiment was conducted in an ordinary office. We collect 500 packets for each setting.

Impact of the SNR. As shown in Fig. 16(a), we manually tune the SNR of Alice and Bob from 9.5 dB to 18 dB. We present the BER results in Fig. 17. Clearly, for different SNRs setting, Chitchat outperforms PhyCode significantly. Chitchat achieves

comparable performance with SigMix in most cases. As the SNR increases, the performance of Chitchat is much higher than PhyCode and even better than SigMix. For example, as shown in Fig. 17(c), when the SNR is around 15 dB, the BER results of Chitchat, SigMix and PhyCode are 4.34×10^{-5} , 8.68×10^{-5} and 2.6×10^{-4} , respectively. This experiment result reveals that Chitchat performs well within the typical SNR range.

Impact of the SNR gap. Recall that in current NOMA technique—SIC, in order to decode the superimposed signal, it requires a clear power domain gap between the transmitted signals, i.e., SNRs with a large difference, which may limit the accessibility of many IoT devices. In contrast, Chitchat is designed without such requirement. Here, we use SNR ratio as a metric to represent the SNR gap between transmitters. To evaluate the influence of a small SNR gap, we keep the ratio close to 1. Specifically, SNR_B is around 16 dB and SNR_A is gradually changed as shown in Fig. 16(b). We plot the BER in Fig. 18. Apparently, Chitchat still outperforms PhyCode and has the comparable performance of SigMix with the varying SNR ratio. The results reveal that Chitchat can perform well under a small SNR gap between transmitters, so Chitchat does not need the power control.

PRR. We then evaluate the impact of the signal strength from the packet level. We calculate the PRR according to the bit error information from the above two experiments. As plotted in Fig. 19(a), Chitchat can achieve a comparable PRR of SigMix with a median value of 1, while PhyCode only has a median value of 0.77. Hence, Chitchat can achieve a reliable performance under the typical signal strength.

Throughput gain. Fig. 19(b) shows that Chitchat achieves a median $1.3\times$ higher throughput gain than PhyCode. Importantly, although Chitchat and SigMix have the comparable reliability performance, Chitchat achieves a median $2\times$ higher throughput gain than SigMix. This is because SigMix requires transmitting two signal copies within one packet and only use one copy for decoding. The amount of information is therefore reduced to half. In contrast, Chitchat can achieve reliable performance without repetitive symbols, which represents a significant improvement of the spectrum efficiency.

B. Performance in Dynamic Environment

We next evaluate Chitchat in dynamic environments with 5 test regions as shown in Fig. 14. During the experiments, people in the environment worked as usual (e.g., walking around), which contributes to a dynamic channel condition. We plot the SNR of two concurrent transmitted signals in Fig. 20. As we can see, although the average SNR can be achieved to 14.97 dB and 17.12 dB for Alice and Bob, respectively, the variance of the SNR is very high as some outliers are below 0 dB, resulting in a highly dynamic channel condition. The overall performance is as follows.

BER and PRR. From Tabel. I, we can see that Chitchat obtains a better average BER (6.6×10^{-3}) when compared with PhyCode (1.19×10^{-2}). Further, Chitchat obtains a lower standard deviation (3.34×10^{-2}) when compared with PhyCode (2.67×10^{-2}). The results contribute to a higher median PRR

TABLE I: Comparison of BER.

Metric	Scheme		
	PhyCode	SigMix	Chitchat
Average BER	1.19×10^{-2}	2.8×10^{-3}	6.6×10^{-3}
Standard deviation	3.34×10^{-2}	2.47×10^{-2}	2.67×10^{-2}

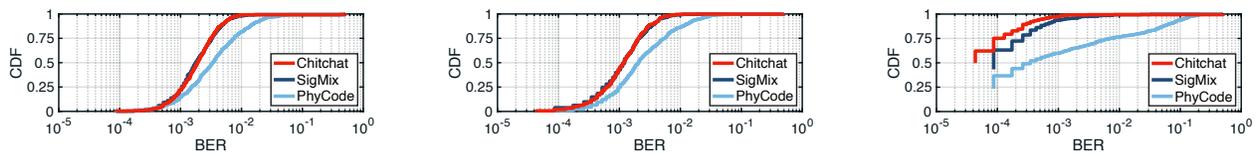
as shown in Fig. 21(a), 0.94 for Chitchat compared with 0.8 for PhyCode in the dynamic environment. Note that both the BER and PRR of Chitchat are not as good as that of SigMix. This is reasonable since the SNRs vary below 0 dB in the dynamic environment SigMix actually transmitted the same symbol twice within each packet for a reliable performance, but at a cost of a much lower throughput. In contrast, Chitchat can decode the superimposed signal without repetitive symbols.

Throughput gain. Fig. 21(b) shows that Chitchat achieves a median $1.88\times$ and $1.18\times$ higher throughput gain than SigMix and PhyCode, respectively. This is expected as Chitchat can achieve a reliable performance without introducing repetitive transmissions. Note that these results are slightly lower than the above experiments in the static environment. The reason is that the dynamic environment will cause more signal variations and SNR fluctuations. Overall, Chitchat can still outperform the state-of-the-art schemes even in the dynamic environment.

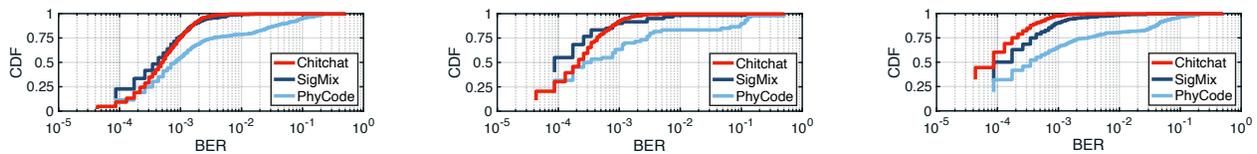
VII. RELATED WORK

Decoding the superimposed signals has been applied to a wide range of wireless communication scenarios including NOMA, two-way relay networks and general collisions. Although different scenarios are designated for different applications, they all benefit from the use of superimposed signals, and these three categories are the most commonly seen scenarios in the literature. We describe the related work separately in the following context.

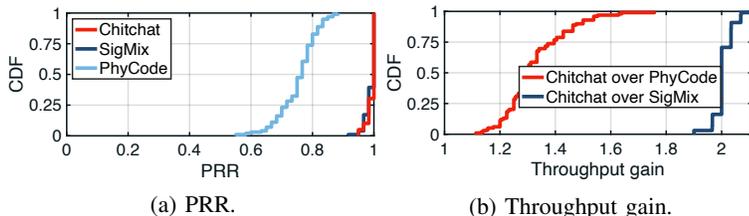
(a) NOMA: Successive Interference Cancellation (SIC) is a major NOMA technology, which requires strict power control to guarantee the power differences among transmitted signals. Thus SIC can decode the strongest signal first while treating others as noise, and then cancel the decoded signal out. By repeating this procedure, other signals can be decoded separately [33], [8]. Although SIC is promising, it relies on dedicated infrastructure and channel feedback for the strict power control [13], [34], which may not be desirable for low-cost and heterogeneous IoT devices [18] and may cause extra delay [35], [36]. Instead of strict power control, other works were proposed to either increase the diversity gain or coding gain to decode the superimposed signals. For the diversity gain, SigMix [9] transmits each signal twice back-to-back to guarantee a low error rate. Although SigMix provides high reliability, its throughput upper bound is equal to that of sequential transmissions, which limits its gain in the network throughput. NCMA [10], [11], [12] employs error correction code by adding repetitive transmissions to recover the decoding error in the MAC layer. However, the spectrum efficiency cannot be fully utilized due to the repetitive transmissions. More importantly, their error recovering performance is severely impaired when the raw error rate is high, resulting in a poor reliability. In contrast to the previous works, Chitchat does not need power control and can obtain both diversity



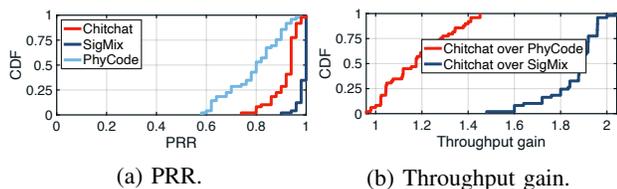
(a) $\text{SNR}_A = 11.2$ dB, $\text{SNR}_B = 12.8$ dB. (b) $\text{SNR}_A = 11.8$ dB, $\text{SNR}_B = 13.5$ dB. (c) $\text{SNR}_A = 14.9$ dB, $\text{SNR}_B = 16.4$ dB.
Fig. 17: BER under different SNR settings.



(a) SNR ratio $\sigma = \text{SNR}_B/\text{SNR}_A = 1.18$ (b) $\sigma = 1.11$ (c) $\sigma = 1.09$
Fig. 18: BER under different SNR gaps.



(a) PRR. (b) Throughput gain.
Fig. 19: The performance comparisons in static environments.



(a) PRR. (b) Throughput gain.
Fig. 21: The performance comparisons in dynamic environments.

gain and coding gain without any repetitive transmissions. The theoretical throughput of Chitchat is doubled compared to sequential transmissions without sacrificing reliability can be improved significantly.

(b) Two-way relay networks: In two-way relay networks, two end nodes are scheduled to transmit signals concurrently, so signals are superimposed at the relay, and then the relay node broadcasts this superimposed signal back to the end nodes. Each end node can decode the signal transmitted from the other end node by canceling out its own signal [37], [38]. Several systems have been proposed to implement this idea including Physical-layer Network Coding (PNC) [29], [39], [40], Analog Network Coding (ANC) [41] and full duplex [42], [43], [44], [45]. More recent works extended these systems to be more robust [40] and scalable to a long-hop scenario [46], [47]. However, these works were designed for relay networks and required knowing one of the two concurrently transmitted packets in decoding, making them incompatible to decode the superimposed signals for multiple access. In contrast, Chitchat can be applied for multiple access scenarios where the superimposed signals can be decoded without knowing one of the concurrently transmitted packets.

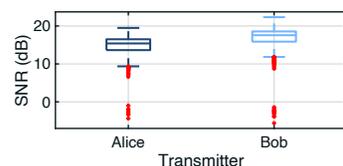


Fig. 20: The SNR distributions in dynamic environments.

(c) General collisions: Due to the hidden terminals problem, multiple users may transmit packets to the same receiver simultaneously, which creates interference at the receiver. Unlike NOMA and bidirectional relay networks, this concurrent transmission collides naturally without any synchronization scheme. Past work leveraged time offsets among collisions to separate multiple packets which contain two copies of each signal (e.g., ZigZag [48]). Given the redundancy of two copies per signal, ZigZag's throughput upper bound is limited to that of sequential transmissions. Decoding collisions in other IoT techniques, such as LoRa [49], [50], RFID [51], [52], [35], [53], [54] and ZigBee [55], has attracted many research interests. These approaches leverage some notable features, often associated with a lower rate, are not spectrum efficient. Motivated to achieve high efficiency and reliability for two-transmitter concurrent transmissions, we design Chitchat, which is applicable for high-rate wireless systems such as Wi-Fi with OFDM. Note that diverse wireless technologies have been proposed for these IoT applications including Bluetooth, ZigBee, LoRa, narrow band-IoT, etc. Among these technologies, Wi-Fi based systems are attractive thanks to the pervasive availability of Wi-Fi devices and their high data rate property [56].

VIII. DISCUSSION

Other modulation schemes Currently, our proposed scheme has been implemented with BPSK modulated signals that provide a sufficient data rate for many IoT applications. Compared with traditional low-data-rate IoT communication schemes, such as LoRa and RFID, our system owns a superior advantage in offering a higher data rate. Furthermore, to meet an even higher communication demand, our scheme is

compatible with the extension of other modulation schemes, e.g., QAM. However, such a high data rate may increase the decoding complexity, which should be studied in future work.

A larger-scale implementation To reveal more performance details of our system, a larger scale of implementation is needed. In particular, more communication units are required to construct a network that contains diverse and practical communication features. We see this as a promising future work to verify our performance and enhance our system through real-world feedback.

Compatibility with related techniques The growth of IoT is drawing so much attention recently. The compatibility of our system to related techniques is crucial for truly applying Chitchat in practice. Fortunately, we have witnessed a rapid development of Cross Technology Communication (CTC) [57], [58] that enables popular technologies, e.g., Wi-Fi and ZigBee, to communicate freely. Because Chitchat is based on Wi-Fi, it can also be used for CTC which would boost the compatibility of Chitchat to many related techniques.

Promising applications Chitchat aims to obtain the wireless transmission reliability while increasing the efficiency. By doing this, many wireless applications can use Chitchat to enhance their performance. For example, the wireless sensor network can reduce data uploading time so that the energy can be saved for both the transmitter and the receiver; vehicle networks can update critical information more frequently to achieve a safer and robust driving environment; abundant smart home devices can communicate with AP with fewer signal collisions and then the user would experience a stable connection.

IX. CONCLUSION

Guided by the seminar work of Cover [1], both academia and industry are inspired to apply superimposed signals to achieve a higher spectrum efficiency, and SIC-type decoding has been proposed as a candidate radio access technology for 5G cellular systems. As SIC is limited to deal with the situations that two concurrent transmitted signals have a significant SNR difference, in this paper, we propose and implement Chitchat, a rotation based encoding and decoding system for two-transmitter concurrent transmissions, without the SNR difference constraint. It is inherently challenging to decode superimposed signals given the uncontrollable and time-varying channels which may make the constellation maps of the superimposed signals difficult and sometimes even infeasible to decode. By addressing the design and implementation challenges, this work nevertheless demonstrates the feasibility in this promising direction. It is anticipated that the modulation-level superimposed signal decoding will lead to a wide range of applications, from radio access for infrastructure-based wireless systems to two-way relay for multi-hop wireless mesh networks, and even for collision resolution in random-access networks, which can be seen as further research directions.

REFERENCES

[1] T. Cover, "Broadcast channels," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 2–14, 1972.

[2] P. Bergmans and T. Cover, "Cooperative broadcasting," *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 317–324, 1974.

[3] S. R. Islam, M. Zeng, O. A. Dobre, and K.-S. Kwak, "Nonorthogonal multiple access (NOMA): How it meets 5G and beyond," *Wiley 5G Ref: The Essential 5G Reference Online*, pp. 1–28, 2019.

[4] Y. Liu, Z. Qin, M. ElKashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Non-orthogonal multiple access for 5G and beyond," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2347–2381, 2017.

[5] X. Xiong, W. Xiang, K. Zheng, H. Shen, and X. Wei, "An open source SDR-based NOMA system for 5G networks," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 24–32, 2015.

[6] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: A back-of-the-envelope perspective," in *ACM SIGCOMM Workshop*, 2010.

[7] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *ACM SIGCOMM*, 2009.

[8] T. Das, L. Chen, R. Kundu, A. Bakshi, P. Sinha, K. Srinivasan, G. Bansal, and T. Shimizu, "CoReCast: Collision resilient broadcasting in vehicular networks," in *ACM MobiSys*, 2018.

[9] W. Cui, C. Liu, H. Mosavat-Jahromi, and L. Cai, "SigMix: Decoding superimposed signals for IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3026–3040, 2020.

[10] H. Pan, L. Lu, and S. C. Liew, "Practical power-balanced non-orthogonal multiple access," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2312–2327, 2017.

[11] L. You, S. C. Liew, and L. Lu, "Network-coded multiple access II: Toward real-time operation with improved performance," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 2, pp. 264–280, 2014.

[12] L. Lu, L. You, and S. C. Liew, "Network-coded multiple access," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2853–2869, 2014.

[13] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[14] A. Z. Ghanavati and D. C. Lee, "Tight bound on the error probability of rotation code in rayleigh fading channels," in *IEEE VTC2014-Fall*, 2014.

[15] C.-Y. Hsu, R. Hristov, G.-H. Lee, M. Zhao, and D. Katabi, "Enabling identification and behavioral sensing in homes using radio reflections," in *ACM CHI*, 2019.

[16] C. Liu, J. Xiong, L. Cai, L. Feng, X. Chen, and D. Fang, "Beyond respiration: Contactless sleep sound-activity recognition using RF signals," *ACM UbiComp*, 2019.

[17] C. Liu, D. Fang, Z. Yang, H. Jiang, X. Chen, W. Wang, T. Xing, and L. Cai, "RSS distribution-based passive localization and its application in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2883–2895, 2015.

[18] T. H. Lee and A. Hajimiri, "Oscillator phase noise: A tutorial," *IEEE journal of solid-state circuits*, vol. 35, no. 3, pp. 326–336, 2000.

[19] V. Syrjala, M. Valkama, N. N. Tchamov, and J. Rinne, "Phase noise modelling and mitigation techniques in OFDM communications systems," in *Wireless Telecommunications Symposium*. IEEE, 2009, pp. 1–7.

[20] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*. John Wiley & Sons, Inc., 1997.

[21] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: A distributed wireless architecture for exploiting sender diversity," *ACM SIGCOMM*, 2011.

[22] I. F. Akyildiz and M. C. Vuran, *Wireless sensor networks*. John Wiley & Sons, 2010, vol. 4.

[23] H. Xing, Y. Liu, A. Nallanathan, and Z. Ding, "Sum-rate maximization guaranteeing user fairness for noma in fading channels," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[24] X. Wei, H. Liu, Z. Geng, K. Zheng, R. Xu, Y. Liu, and P. Chen, "Software defined radio implementation of a non-orthogonal multiple access system towards 5g," *IEEE Access*, vol. 4, pp. 9604–9613, 2016.

[25] S. Abeywickrama, L. Liu, Y. Chi, and C. Yuen, "Over-the-air implementation of uplink noma," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[26] E. Khorov, A. Kureev, and I. Levitsky, "Noma testbed on wi-fi," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 1153–1154.

[27] E. Khorov, A. Kureev, I. Levitsky, and I. F. Akyildiz, "Prototyping and experimental study of non-orthogonal multiple access in wi-fi networks," *IEEE Network*, vol. 34, no. 4, pp. 210–217, 2020.

- [28] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "Performance assessment of IEEE 802.11 p with an open source SDR-based prototype," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1162–1175, 2017.
- [29] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *ACM MobiCom*, 2006.
- [30] J. Schmitz, F. Bartsch, M. Hernández, and R. Mathar, "Distributed software defined radio testbed for real-time emitter localization and tracking," in *IEEE ICC Workshops*, 2017.
- [31] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *USENIX NSDI*, 2016.
- [32] W. Cui, C. Liu, L. Cai, and J. Pan, "PhyCode: A practical wireless communication system exploiting superimposed signals," in *IEEE ICC*, 2019.
- [33] W. Zhou, T. Das, L. Chen, K. Srinivasan, and P. Sinha, "BASIC: Backbone-assisted successive interference cancellation," in *ACM MobiCom*, 2016.
- [34] A. Y.-C. Peng, S. Yousefi, and I.-M. Kim, "On error analysis and distributed phase steering for wireless network coding over fading channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, pp. 5639–5649, 2009.
- [35] P. Hu, P. Zhang, and D. Ganesan, "Laissez-faire: Fully asymmetric backscatter communication," *ACM SIGCOMM*, 2015.
- [36] Z. Song, L. Shangguan, and K. Jamieson, "Wi-Fi goes to town: Rapid picocell switching for wireless transit networks," in *ACM SIGCOMM*, 2017.
- [37] C. E. Shannon, "Two-way communication channels," in *4th Berkeley Symposium on Math. Stat. and Prob.*, 1961.
- [38] E. C. Van Der Meulen, "Three-terminal communication channels," *Advances in applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.
- [39] L. Lu, T. Wang, S. C. Liew, and S. Zhang, "Implementation of physical-layer network coding," *Physical Communication*, vol. 6, pp. 74–87, 2013.
- [40] L. You, S. C. Liew, and L. Lu, "Reliable physical-layer network coding supporting real applications," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2334–2350, 2016.
- [41] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *ACM SIGCOMM*, 2007.
- [42] D. Bharadia and S. Katti, "FastForward: Fast and constructive full duplex relays," *ACM SIGCOMM*, 2015.
- [43] B. Chen, Y. Qiao, O. Zhang, and K. Srinivasan, "AirExpress: Enabling seamless in-band wireless multi-hop transmission," in *ACM MobiCom*, 2015.
- [44] L. Chen, F. Liu, and K. Srinivasan, "Verification: Constructive and destructive full duplex relays," in *ACM MobiCom*, 2019.
- [45] Y. Ma, N. Selby, and F. Adib, "Drone relays for battery-free networks," in *ACM SIGCOMM*, 2017.
- [46] H. Zhang and L. Cai, "Bi-directional multi-hop wireless pipeline using physical-layer network coding," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7950–7965, 2017.
- [47] L. Chen, F. Wu, J. Xu, K. Srinivasan, and N. Shroff, "BiPass: Enabling end-to-end full duplex," in *ACM MobiCom*, 2017.
- [48] S. Gollakota and D. Katabi, "ZigZag decoding: Combating hidden terminals in wireless networks," in *ACM SIGCOMM*, 2008.
- [49] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *ACM SIGCOMM*, 2017.
- [50] M. Hesar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *USENIX NSDI*, 2019.
- [51] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," *ACM SIGCOMM*, 2012.
- [52] M. Jin, Y. He, X. Meng, Y. Zheng, D. Fang, and X. Chen, "Fliptracer: Practical parallel decoding for backscatter communication," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 330–343, 2019.
- [53] M. Jin, Y. He, X. Meng, D. Fang, and X. Chen, "Parallel backscatter in the wild: When burstiness and randomness play with you," in *ACM MobiCom*, 2018.
- [54] T. Das and P. Sinha, "ADS: Accurate decoding of RFID tags at scale," in *ACM CoNEXT*, 2019.
- [55] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in ZigBee," in *ACM MobiCom*, 2015.
- [56] C. V. Forecast *et al.*, "Cisco visual networking index: Global mobile data traffic forecast update 2016–2021," *Cisco Public Information*, 2017.
- [57] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for iot devices," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 204–216.
- [58] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 2–14.