

SledZig: Boosting Cross Technology Coexistence for Low-Power Wireless Devices

刘朝阳

摘要

随着物联网的快速发展,在同一频段内工作的异构无线设备数量急剧增加,导致设备之间存在严重的相互干扰。为了实现设备共存,研究人员提出了大量的管理干扰的机制。然而,这些现有的机制对物理层或MAC介质访问控制层上都进行了过度的修改,使得它们很难部署在商业设备上。

本次论文复现工作,拟设计并实现SledZig,在对高功率设备(wifi)传输吞吐量影响最小化的前提下,降低其信号功率,为低功率设备(zigbee)提供更多的传输机会,来促进设备同步共存。

关键词: 异构无线设备; 设备共存

1 引言

物联网(IoT)的蓬勃发展使无线设备的数量成指数级增长。无线设备采用异构无线技术[1],每种技术由于其优缺点,都有其合适的应用场景。然而在ISM频带[2]中,异构无线设备不可避免地在重叠信道中工作,产生严重的跨技术共存问题。

WiFi和ZigBee是物联网中最常见的两种无线技术。WiFi用于无线局域网(WLAN),其市场非常广阔。专家预测,2023年WiFi热点的数量将达到6.28亿个。同时ZigBee在为无线传感器网络提供低成本、低数据率、低能耗等特性方面发挥着重要作用。ZigBee市场也在稳步增长。2018年其价值达到28.1亿美元,预计到2026年将达到53.8亿美元。在功率方面,ZigBee信号凭借传输频率始终小于1 mW来实现节能,而WiFi信号传输频率为100 mW进行大覆盖。于是,由于功率上的不对等,以及MAC层设计的差异,当设备进行竞争信道时,WiFi比ZigBee具有更高的优先级,总是可以赢得数据传输信道。因此,WiFi设备通过干扰ZigBee数据传输,对ZigBee设备造成严重的共存问题。

共存问题近年来引起了广泛的研究兴趣,相关工作可分为跨技术抗干扰和抗干扰两类[3]。实现干扰避免的机制总是通过设计PHY或MAC层协议来减轻跨技术干扰(CTI)[4]。例如,EmBee[5]让WiFi设备识别ZigBee信号的信道,然后通过设计空子载波预留相应的信道用于ZigBee的传输。抗干扰机制试图通过PHY层设计来恢复碰撞信号,如CrossZig[6],它利用数据包合并和自适应前向纠错(FEC)[7]编码在CTI下恢复数据包。这两种机制都需要对MAC层或PHY层进行修改,因此不能直接应用于当前的商业设备。

提出了一种WiFi节能机制SledZig来维护ZigBee传输。SledZig完全兼容标准的PHY和MAC层进程,并且不需要修改商业WiFi和ZigBee设备。通过利用WiFi中的QAM调制特性,来降低ZigBee通道上的WiFi信号能量,同时保持WiFi传输功率不变。QAM是相位和振幅调制方法的组合,使得不同QAM星座点具有不同的功率水平。通过向原始WiFi数据位中插入额外的位来让与ZigBee通道重叠的子载波中的

QAM点的功率最低，而在ZigBee通道外的QAM点保持不变，使得ZigBee通道上的能量减少14 dB左右。随着能量的减少，ZigBee网络的性能可以通过获得更多的传输机会和避免干扰来显著提高。

使用方法上，SledZig非常简单。使用原始数据位，WiFi发射器首先插入额外的位来生成传输位。当传输位通过标准WiFi传输过程时，ZigBee信道上的信号能量可以自动降低，从而提高ZigBee传输。同时，WiFi接收器可以通过从接收到的位中去除额外的位，获得原始数据位。

2 相关工作

此部分将介绍SledZig设计的背景知识。

2.1 WIFI传输

标准的WiFi传输过程：数据位首先通过信道编码模块来实现抗干扰，QAM调制后转换为复杂符号；QAM点映射到S/P（串行转并行）模块后的OFDM（正交频分复用）子载波，经过IFFT（快速傅里叶逆变换）和P/S（并行转串行）处理后输出为时域OFDM符号；每个OFDM符号都插入了CP（循环前缀）来消除符号间干扰，信号最终在射频前端传输。

值得注意的是，OFDM使一个设备传输多个正交的子载波信号 [8]，并紧密间隔，以并行携带数据。在WiFi系统中，每个20MHz的WiFi通道分为64个子载波，包括48个数据子载波、4个导频子载波和12个空子载波。

2.2 WIFI与ZIGBEE的差异

WiFi和zigbee的差异主要为如下两点：

1，在每个WiFi通道中，会有4个zigbee通道与之重叠，且WiFi因为DIFS（退避时间） [9]较短，更容易抢占信道。2，WiFi功率更大，会覆盖zigbee，对其传输造成影响。

图1为信道分布情况图：

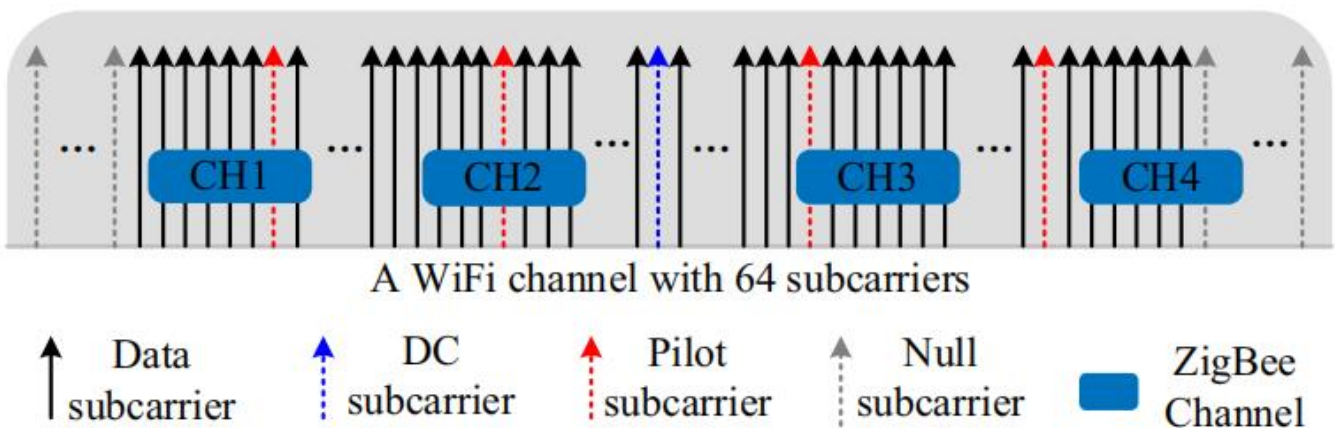


图 1: WiFi与zigbee信道分布情况

3 本文方法

3.1 本文方法概述

SledZig通过向payload插入额外的比特，生成发射位；当发射位通过标准WiFi传输过程时，使重叠信道的子载波对应振幅最低星座点，即图2右红点区域。于是，在信道内不断发送低振幅序列（本实验使用256QAM，因此序列为X100X100X100...）来降低重叠信道中的WiFi信号功率。

本次复现使用1/2卷积编码，256QAM调制，对于发射位的构建是一个逆向推导过程：1、对QAM调制前的数据，即 $\{X_{100}\}$ 序列，由解交织函数得到其交织前显著位100中0和1的位置；2、依据显著位的位置确定原始数据需要插入的额外比特的位置，具体的做法是除以2，同时序列中每6个已知值外加1个未知数（额外比特）进行1次反卷积运算并解方程，便可确定1个额外比特的值；3、最后在对应位置插入额外比特。

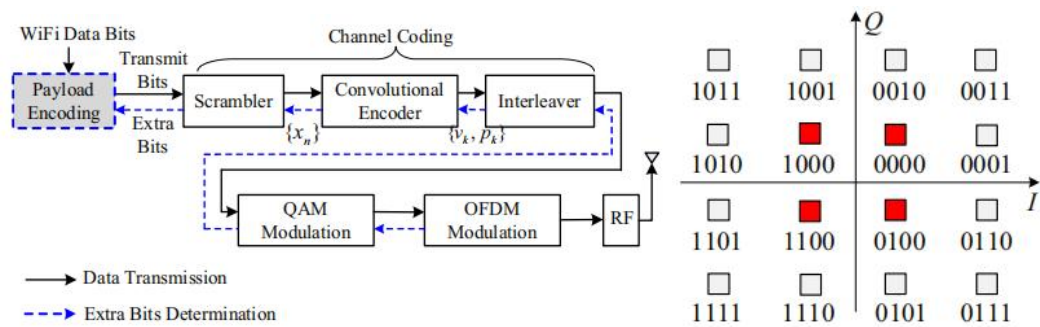


图 2: 方法概述

3.2 星座点分析

系统运行的第一个步骤为星座点分析。根据设计，重叠子载波中的QAM点应为功率最低的四个点。对于256QAM，每个点携带8位，只有6位对使功率最低是重要的。我们称它们为显著位，如表i中所示的阴影位。类似地，每个16QAM和64QAM点分别有2个和4个显著位。额外的位应该插入只使重要的位是指定的位，而QAM点中的其他位可以是任意的。

3.3 解交织模块

信道编码过程包括扰码器、交织器和卷积编码器，此步骤为系统运行的第二个步骤，应用了解交织模块。在无线通信系统中使用交织器来减少解码错误，这里的解交织模块设计是通过QAM调制前的位进行解交织来生成重要位。如图2左所示，我们将解交织后的显著位表示为 $\{v, p\}$ ，其中 v 和 p 表示当前显著位的值和位置。值得一提的是，这个过程为SledZig带来了额外的好处：在解交织之前聚集在一起的重要位被分散到相距较远的不同位置，有利于卷积编码中额外位的确定。

扰码器用于避免具有相同值的长位序列。该模块的SledZig设计是根据乱码的传输位 $\{x_n\}$ 获取传输位。由于这两个模块都是从输入位到输出位的逐个映射，所以对SledZig的反向处理非常简单。

3.4 卷积编码模块

此模块设计的主要目标是根据显著位 $\{v, p\}$ 来确定插入WiFi数据位所需的额外位，如图2左所示。这个过程具有挑战性，因为卷积编码器向数据位添加了冗余位 [10]，并且这些位不是任意的。我们通过分析卷积编码过程，对其特征进行分析，来确定额外的位。

在802.11标准下，在每个QAM调制下使用几种编码速率，从而会有不同的WiFi数据速率 [11]。1/2速率编码是卷积编码的基本过程，其一个输入位产生两个输出位。其他的编码速率，如2/3、3/4和5/6是在1/2速率的编码比特的基础上来实现的：一些编码比特被省略以增加编码速率 [12]。本设计关注的是1/2速率的编码，其他编码速率的过程也是相似的。

1/2速率的卷积编码过程如图3所示。使用了两个二进制生成多项式 $g_0 = \{1011011\}$ 和 $g_1 = \{1111001\}$ 。一个输入位 x_n 触发两个编码位 y_{2n-1} 和 y_{2n} 。输出编码位不仅由当前的输入位 x_n 决定，而且还由从 x_{n-1} 到 x_{n-6}

的这几个先前的位决定。为了便于描述，我们让 $X'_n = \{X_{n-1}X_{n-2}X_{n-3}X_{n-4}X_{n-5}X_{n-6}\}$ 。然后，这个生成两个输出位的一步编码过程可以表述为：

$$\begin{aligned} g_0 \times GR_2 X_n &= y_{2n-1}, \\ g_1 \times GR_2 X_n &= y_{2n}, \end{aligned} \quad (1)$$

在解交织后得到的 $\{v, p\}$ 中有重要的位额外位，那么 $\{x_n\}$ 可以通过等式来一个接一个地确定。以得到的 $[1, 15][0, 16]$ 为例子，对原始序列 X'_n 进行扫描时，扫描到 x_8 （ $n=8$ ）时，发现 $2n-1$ 和 $2n$ 分别为15和16，都是显著位，即双显著位情况，设当前已知序列中附近5位是 $\{10110\}$ ，则插入2个未知数 a 和 b ，其变为 $\{1a011b0\}$ ，反转后为 $\{0b110a1\}$ ，分别与 $\{1011011\}$ 和 $\{1111001\}$ 点乘，并对2取模得到二元一次方程组： $(0+0+1+1+0+a+1)$ 对2取模=1； $(0+b+1+1+0+0+1)$ 对2取模=0。容易解得 $a=0$ ， $b=1$ ，于是插入额外比特后的序列为 $\{1001110\}$ ，以上便是一次模拟的插入过程。依次类推，对于所有得到的 $[v, p]$ ，遍历完原始序列 X'_n 后，即可完成整个插入流程，得到输入序列 X_n 。由这个输入序列进行256QAM调制，必能映射于低振幅星座点位置，从而实现低功率传输。

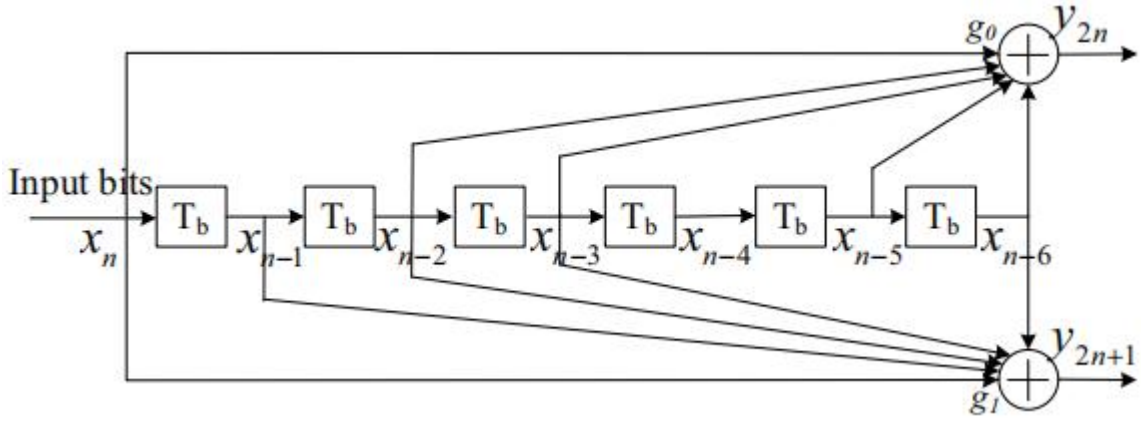


图 3: 1/2卷积编码过程

4 复现细节

4.1 与已有开源代码对比

本次复现过程引用了insertBit以及deinterleave函数，并对其进行优化：原始代码中仅给出了16QAM调制的情况，我们在此基础上通过一个if语句，添加了256QAM的情况，同时借鉴了16QAM情况下所引用的sim_consts.unitBitsBeforeConvQAM16函数，补充添加了sim_consts.unitBitsBeforeConvQAM256函数，此函数的作用是，在配置为256QAM的环境下，将解交织得到的重要位数据作为insertBit的输入。

insertBit代码模块的输入为解交织得到的重要位 $\{v, p\}$ ，以及原始数据 $\{x'_i\}$ 。输出为用于发射的数据 $\{x_i\}$ 。

其具体的步骤为：初始化额外位的值为0。对原始数据的每一位进行遍历， i 表示当前位的序号， N 为序号的上界。如果检测到 $2n-1$ 或者 $2n$ 的值等于 p ，则为单额外位的情况，重新构建当前的6位序列，在第1位前插入额外位。将 v 的值赋值给 y ，即卷积编码的输出，对其由公式1进行反向计算，得到输入 x ，并赋值给当前的这个额外位。如果检测到 $2n-1$ 和 $2n$ 的值同时等于 p ，则位双额外位的情况，重新构建当前的5位序列，分别在第2位和第5位前插入额外位，将 v 的值赋值给 y ，同理得到两个输入，分别赋值给这两个额外位。其他情况下，则仅仅将原始序列的各个位搬运到发射序列即可。如此反复，直到遍历完原始序列，即可得到发射序列。

Procedure 1 Transmit bits generation process.

Input: Data bits $\{x'_i\}$, $i \in [1, N']$; Significant bits $\{v_k, p_k\}$, $k \in [1, K]$.

Output: Transmit Bits $\{x_n\}$, $n \in [1, N]$.

$n \leftarrow 1$; $etr_0 \leftarrow 0$; $etr_1 \leftarrow 0$; $tmp \leftarrow 0$.

for $i = 1 : N'$ **do**

if $(2n - 1) == p_k$ or $2n == p_k$ **then**

$X_n = [etr_0 \ x_{n-1} \ x_{n-2} \ x_{n-3} \ x_{n-4} \ x_{n-5} \ x_{n-6}]'$;

if $(2n - 1) == p_k$ **then**

$y_{2n-1} \leftarrow v_k$.

else

$y_{2n} \leftarrow v_k$.

 Calculate etr_0 through Eq. 1;

$x_n \leftarrow etr_0$;

$n \leftarrow n + 1$; $k \leftarrow k + 1$;

$x_n \leftarrow x'_i$;

$n \leftarrow n + 1$.

else if $(2n - 1) == p_k$ and $2n == p_{k+1}$ **then**

$X_n = [x_{n-2} \ etr_0 \ x_{n-3} \ x_{n-4} \ x_{n-5} \ etr_1 \ x_{n-6}]'$;

$y_{2n-1} \leftarrow v_k$, $y_{2n} \leftarrow v_{k+1}$;

 Calculate etr_0 and etr_1 through Eq. 1;

$tmp \leftarrow x_{n-1}$;

$x_n \leftarrow x_{n-2}$;

$x_{n-1} \leftarrow etr_0$;

$x_{n-2} \leftarrow x_{n-3}$;

$x_{n-3} \leftarrow x_{n-4}$;

$x_{n-4} \leftarrow x_{n-5}$;

$x_{n-5} \leftarrow etr_1$;

$n \leftarrow n + 2$; $k \leftarrow k + 2$;

$x_{n-1} \leftarrow tmp$;

$x_n \leftarrow x'_i$;

$n \leftarrow n + 1$.

else

$x_n \leftarrow x'_i$;

$n \leftarrow n + 1$.

end

4.2 实验环境搭建

本实验环境的搭建在GNU上进行。操作系统使用的是Ubuntu 18.04，gnuradio版本使用v3.7.13.4。

USRP在接通电源后，可以通过其两根天线实现WiFi信号的自发自收。zigbee设备则包括发射机和接收机两台。

在GUN端，TX文件用于生成基带调制数据用于USRP发射，也可以在连接了USRP的同时，一边调制一边发射数据。由于目前发射的每个数据包内容都相同，为了减轻计算机的负载，本实验选择的是先生成基带调制数据，然后再连接USRP，在TX_sample_from_file文件上进行发射。

RX文件用于接收此发射的数据文件，由于本实验需要观测的是WiFi频谱图下的功率变化情况，因此除了需要设置文件的存储位置外，其余基本流程与TX类似。

图4为GNU上的TX数据流图：

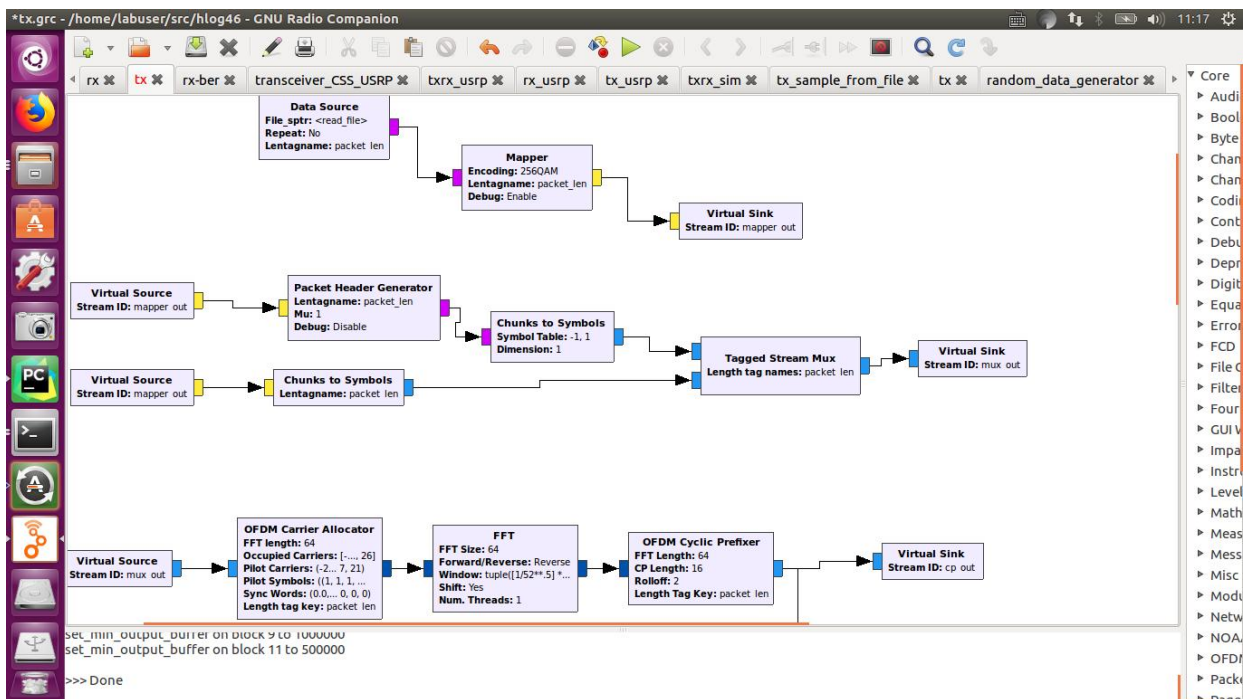


图 4: TX数据流图

4.3 界面分析与使用说明

Data source该模块的作用是使用Read File Object对象读取到的文件数据,输出到输出端口。

其中, 参数File_sptr表示填入Read File Object模块的ID, 已配置好,一般无需修改; 参数Repeat表示是否重复输出文件数据。LenTagName位GNU内部用于区分数据包的标签名字,用户不需要修改。总体来说, 输入分为两种方式, 一种是随机数生成, 另一种是文件导入生成。本实验使用文件导入方式。

在mapper模块中选择256QAM。该模块的作用是将输入数据包按照调制方式对应的1个子载波包括的bit数对数据进行分割,输出的每个item对应1个星座图上的1个点的数据。

在packet header generator模块, 系统根据数据包长度和调制方式生成物理包头, 并进行交织和1/2卷积编码。在随后的chunks to symbols中。对包头进行BPSK, 同时对payload进行星座点映射。为了降低误比特率, 采用格雷编码。

OFDM carrier allocator将输入的星座点映射到各个符号对应的位置, 输出为OFDM调制频域数据。子载波数量为64, 其中有48个用于数据传输, 其余为空子载波, 导频4个。

然后, 在FFT模块中进行快速傅里叶逆变换, 将频域数据转变为时域波形数据, 再在OFDM cyclic prefixer中添加循环前缀。

如此一来, 生成的基带波形文件便可用于USRP发射。

4.4 创新点

本实验添加了256QAM调制情况下插入额外比特的模块, 这个模块由if语句扩展了分支, 同时借鉴了sim_consts.unitBitsBeforeConvQAM16函数, 添加了sim_consts.unitBitsBeforeConvQAM256函数。如图5所示, 在函数InsertBit中, 我们模拟了用于数据传输的1-48个信道, 其中30-38信道为zigbee和WiFi的重叠信道。由图5可以看出在第30-38个信道中, 比特流中的第2, 3, 4, 6, 7, 8位都分别是1, 0, 0, 1, 0, 0。于是可以保证该数据在后续的调制中映射于低功率星座点位置。

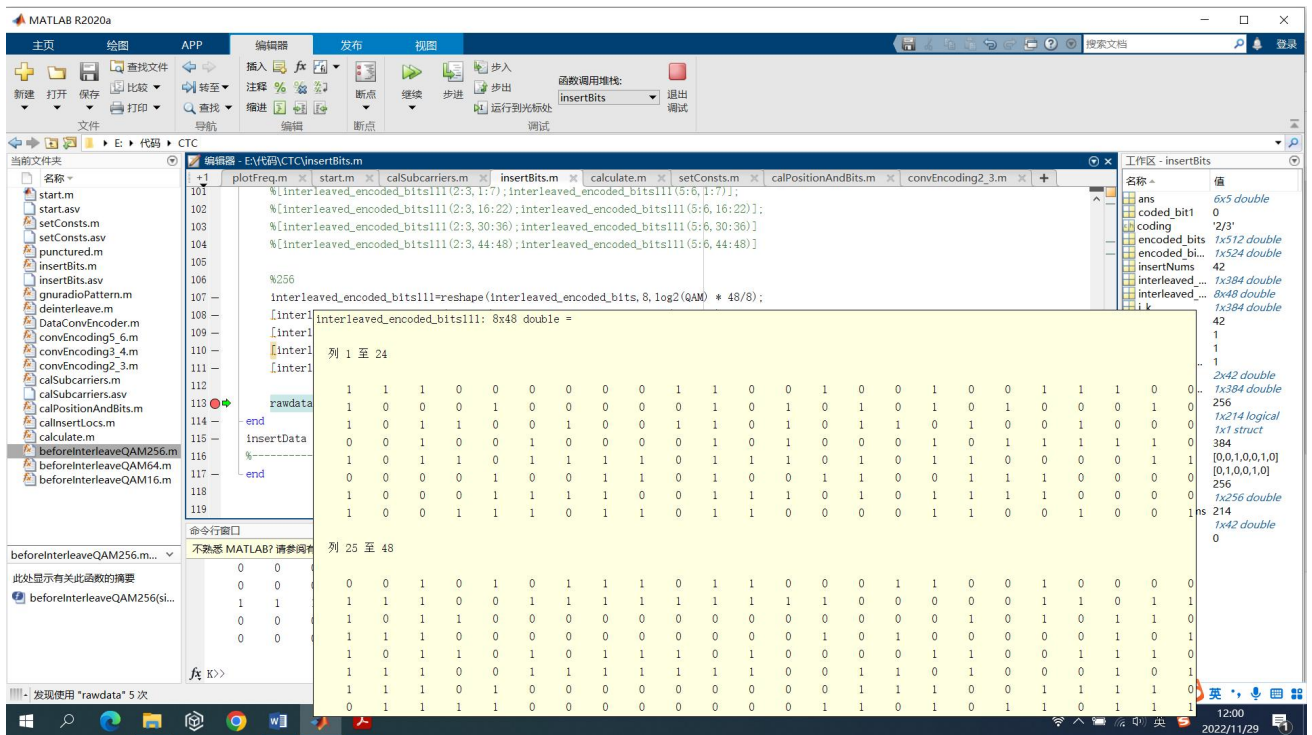


图 5: MATLAB上的模拟信道分布情况

5 实验结果分析

表1统计了在3种情况下，zigbee的吞吐量计数：

1、我们首先让USRP不发送WiFi信号，zigbee发射机向zigbee接收机发送zigbee信号，同时统计其传输吞吐量，观测到其吞吐量计数值大概在56左右；

2、接着，我们在GNU上导入生成的256QAM基带波形文件，让USRP开始发送WIFI数据，WiFi频谱下各个信道的WiFi相对增益较高。于是在WiFi信号的干扰下，观测到zigbee吞吐量几乎为0，接收到的包显示为bad packet；

3、最后，我们运行SledZig程序进行额外比特的插入，在WiFi频谱下显示CH3信道WiFi的相对增益显著降低，而zigbee此时的吞吐量恢复到了56左右，与原先保持一致。

这说明在SledZig的作用下，WiFi的相对增益有所降低，从而减小了其对zigbee传输的干扰，实现了WiFi设备与zigbee设备的共存。

CASE	ZIGBEE吞吐量计数
无WiFi干扰	56左右
存在WiFi干扰	0
在SledZig下存在WiFi干扰	56左右

表 1: 不同情况下zigbee的吞吐量计数

图6为USRP运行后，在GNU界面上的WiFi频谱图。左图为生成的原始随机数的WiFi信号频谱，WiFi相对增益较高。右图为应用了SledZig的WiFi信号频谱，其中zigbee所在的信道CH3，WiFi的相对增益明显降低。

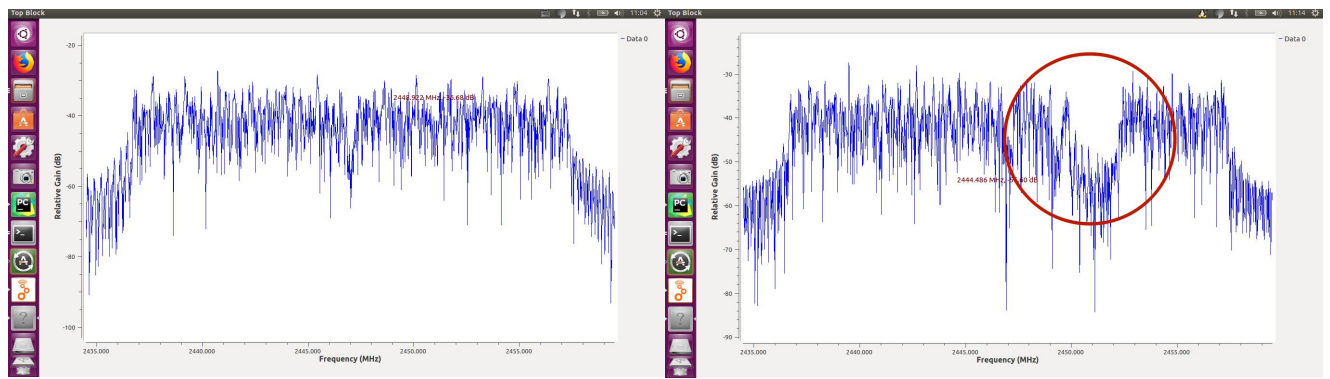


图 6: GUN界面上的WiFi频谱图

由于这种方法不是直接降低WiFi传输功率，而是仅改变映射位置来降低幅度，从而降低相对增益，实现低功率传输，因此对WiFi实际性能的影响很小。

6 总结与展望

本次工作复现了一种WiFi节能机制SledZig来降低WiFi的干扰，从而提高ZigBee在传输时的吞吐量。SledZig能够完全兼容标准的PHY和MAC层进程，并且无需对当前商业WiFi和ZigBee设备进行修改。通过对WiFi传输过程进行逆向计算，从星座点分析到解交织，再到卷积编码运算，实现了对原始数据的额外比特插入，在功率最低的重叠子载波中创建星座点，降低了CH3信道中的WiFi相对增益，随着其能量的减少，为ZigBee的传输提供了更多机会，性能得以提升。在硬件测试台上实现并对其进行评估，实验结果表明，本次复现的SledZig可以有效地提高ZigBee传输，提高WiFi通道的性能，WiFi吞吐量损失低至7%左右。

然而尽管如此，由于冗余信息的增加，WiFi的7%吞吐量降低仍然有改善空间，在为zigbee传输寻求更多机会的同时，还需寻求一种降低冗余度的方案，来实现WiFi更好的传输性能。

参考文献

- [1] R. Chen and W. Gao, "Enabling cross-technology coexistence for extremely weak wireless devices," in Proc. of the IEEE INFOCOM, 2019.
- [2] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy, "CrossZig: combating cross-technology interference in low-power wireless net works," in Proc. of the ACM/IEEE IPSN, 2016.
- [3] IEEE Computer Society. 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," 2009.
- [4] IEEE Computer Society. 802.15.4, "IEEE Standard for low-rate wireless networks," 2016.
- [5] Y. Yan, P. Yang, X. Li, Y. Tao, L. Zhang, and L. You, "ZIMO: building cross-technology MIMO to harmonize zigbee smog with WiFi flflash without intervention," in Proc. of the ACM MobiCom, 2013.
- [6] C. Shao, H. Park, H. Roh, W. Lee, and H. Kim, "PolarScout: WiFi interference-resilient ZigBee communication via shell-shaping," IEEE/ACM Transactions on Networking, vol. 28, no. 4, pp. 1587–1600, Aug. 2020.

- [7] R. Boidar, R. Chandra, and D. Gunawardena, “Weeble: Enabling low-power nodes to coexist with high-power nodes in white space networks,” in Proc. of the ACM CoNEXT, 2012.
- [8] X. Zheng, D. Xia, X. Guo, L. Liu, Y. He, and H. Ma, “Portal: transparent cross-technology opportunistic forwarding for low-power wireless networks,” in Proc. of the ACM Mobihoc, 2020.
- [9] W. Chen, Z. Yin, and T. He, “Global cooperation for heterogeneous networks,” in Proc. of the IEEE INFOCOM, 2020.
- [10] Y. Chae, S. Wang, and S. M. Kim, “Exploiting WiFi guard band for safeguarded ZigBee,” in Proc. of the ACM SenSys, 2018.
- [11] G. Chen, W. Dong, and J. Lv, “LoFi: enabling 2.4GHz LoRa and WiFi coexistence by detecting extremely weak signals,” in Proc. of the IEEE INFOCOM, 2021.
- [12] P. Gawowicz, A. Zubow, S. Bayhan, and A. Wolisz, “Punched cards over the air: cross-technology communication between LTE-U/LAA and WiFi,” in Proc. of the IEEE WoWMoM, 2020.