

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

摘要

随着物联网的快速发展，工作在同一频段的异构无线设备数量在急剧增加，使得出现了严重的跨技术干扰现象。为了实现异构无线设备共存，相关研究人员提出了许多解决干扰的机制，但现有的机制都会在物理（PHY）层或媒体访问控制（MAC）层上造成很大的改动，使得它们难以部署在商业设备上。本文设计并实现了 SledZig，它可以通过提供更多传输机会并避免干扰来帮助低功耗设备做到跨技术共存，对于物理层和 MAC 层的标准它也是完全兼容的。具体来说，SledZig 通过使与低功率设备信道重叠的 WiFi 子载波中的星座点具有最低功率，以保持 WiFi 传输功率不变的情况下降低低功率设备信道上的 WiFi 信号功率，从而达到低功率设备不被干扰进行传输的目的，而这只需对 WiFi 有效负载进行编码即可实现。本人在硬件测试台上复现了 SledZig，并获得了实验结果，在频谱图上，可以明显观察到被降低的 WiFi 信号功率。

关键词：异构无线网络；共存；WiFi；ZigBee。

1 引言

1.1 选题背景和依据

物联网的繁荣发展使无线设备的数量呈指数级增长。无线设备采用异构无线技术，每种技术都有其各自的优点、缺点和合适的应用场景。在拥挤的 ISM（工业、科学和医疗）频段中，异构无线设备不可避免地工作在重叠的信道中，导致出现了严重的跨技术共存问题。

其中，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)，其现在和未来的市场都将稳定增长；而 ZigBee 则在为无线传感器网络提供低成本、低数据速率和低能耗特性方面发挥着重要作用。WiFi 和 ZigBee 具有不同的信道带宽：WiFi 有 13 个 20MHz 信道，信道间隔为 25MHz；ZigBee 有 16 个 2MHz 信道，信道间隔为 5MHz。因此，1 个 WiFi 信道与 4 个 ZigBee 信道重叠。在 WiFi 系统中，每个 20MHz 的 WiFi 信道被划分为 64 个子载波，其中包括 48 个数据子载波、4 个导频子载波和 12 个空子载波，在相同的模式中与 4 个 ZigBee 信道重叠，如图 1 所示。为了后文描述方便，这里将 4 个 ZigBee 通道简称为 CH1、CH2、CH3 和 CH4。可以看到 CH1-CH3 与导频子载波重叠，CH4 与空子载波重叠。

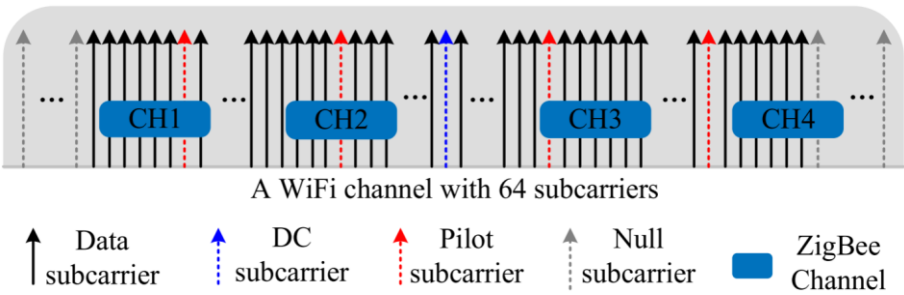


图 1 1 个 WiFi 信道与 4 个 ZigBee 信道重叠的图示

在上述的情况下，WiFi 的两个特征给 ZigBee 带来了严重的共存问题。一是 WiFi 和 ZigBee 的功率水平不对等：WiFi 信号的传输功率高达 100mW，目的是实现大覆盖范围；而 ZigBee 信号始终以低于 1mW 的功率传输来实现节能。较强的 WiFi 信号很容易干扰 ZigBee 传输。二是当两种设备竞争信道时，WiFi 由于其 MAC 层的设计而比 ZigBee 具有更高的优先级，总能赢得数据传输的信道。具体来说，WiFi 和 ZigBee 网络均采用 CSMA/CA 机制来竞争信道。详细的 CSMA/CA 机制如图 2 所示。当设备开始发送数据包时，它首先等待 DIFS 的时间。如果在 DIFS 的时间内信道空闲，则设备等待由多个退避时隙（backoff timeslot）组成的随机持续时间来竞争信道。当信道空闲了 1 个退避时隙时，退避定时器减 1；当信道繁忙时，退避定时器不变动；当退避定时器达到零，则设备最终可以发送数据包。在 DIFS 或每个退避时隙期间，设备应执行 CCA（信道空闲评估）以确定信道是否空闲：如果检测到的信号能量低于预定义的阈值，则确定信道是空闲的；否则，信道是繁忙的。而 WiFi 和 ZigBee 在竞争信道的过程中主要的区别在于，WiFi 的 DIFS 为 $28\mu\text{s}$ [2]，而 ZigBee 的 DIFS 为 $320\mu\text{s}$ [3]，且 WiFi 退避时隙为 9 或 $20\mu\text{s}$ ，而 ZigBee 退避时隙为 $320\mu\text{s}$ 。这个区别导致 WiFi 设备总能赢得传输信道。

总的来说，WiFi 设备或是禁止 ZigBee 设备的数据传输，或是干扰正在进行的 ZigBee 数据传输，从而给 ZigBee 设备带来了严重的跨技术共存问题。

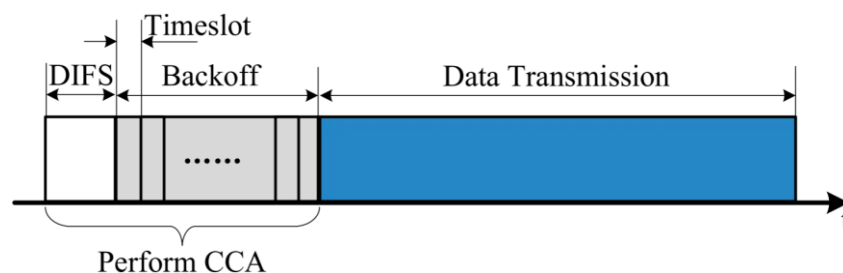


图 2 CSMA/CA 的工作图示

1.2 选题意义

从前文中可以看到，WiFi 与 ZigBee 的设备共存时，ZigBee 设备的数据传输将会被严重影响，主要有两个原因：一方面，WiFi 设备因为 MAC 层的设计可能会直接禁止 ZigBee 设备的数据传输；另一方面，WiFi 设备的传输功率远远高于 ZigBee 设备的传输功率，使得 WiFi 设备可能会干扰 ZigBee 设备的数据传输，这是 PHY 层的缘故。针对这个问题，相关研究人员提出了许多解决方法，但现有的方法在 PHY 层或 MAC 层上造成的改动很大，使得它们难以部署在商业设备上。因此本文提出了新的方法即 SledZig，在无需更改 PHY 层或 MAC 层相关机制并且不影响 WiFi 传输功率的基础上帮助 ZigBee 设备进行正常的信号发送与接收，该设计的具体方法亦可作为其它异构无线网络设备共存的参考方法。

除此之外，本文涉及了许多与无线网络技术尤其是 WiFi 相关的基础知识，本人在学习并复现文章的过程中，不但可以接触到很多新的知识，还能复习已有的相关知识，以为后续的研究生涯奠定坚实的基础。

2 相关工作

2.1 跨技术共存

WiFi 和 ZigBee 的技术共存问题在过去的几年激发了人们广泛的研究兴趣，相关工作可分为两类：跨技术干扰避免和干扰抵抗机制。

干扰避免机制始终通过设计 PHY 层或 MAC 层协议来减轻跨技术干扰（CTI）。已有一些方法通过在异构设备之间交换协调信息进行协议设计来避免 CTI。例如，CBT[7]、Weeble[8]和 WiCop[9]通过使 ZigBee 设备传输专门设计的信号来提高 ZigBee 对 WiFi 的可见性，从而使 WiFi 设备在 ZigBee 传输的过程中保持沉默；Gsense[10]使 WiFi 设备通过定制的前导码向 ZigBee 设备传输协调信息，从而调度它们的传输。近年来，一些方法利用新兴的跨技术通信（CTC）[11]-[14]，通过实现异构设备之间的显式协调来实现干扰的管理[15]-[20]。例如，ECC[15]使 WiFi AP 协调所有 WiFi 和 ZigBee 设备的数据传输以避免干扰，从而实现高网络吞吐量；ECT[16]为 CTC 设计网络层，并让服务器调度 ZigBee 的传输；Chiron[17]设计了一种定制的网关，可以实现同一频段内 WiFi 和 ZigBee 数据流的并发传输，以减少传输延迟；BiCord[20]利用异构设备之间的双向协调来实现高效的射频信道分配。这些已有的方法总是会引起额外的数据包传输，并且需要对标准进行大量的修改。

干扰抵抗机制利用 PHY 层解决方案来对抗 CTI。例如，BuzzBuzz[4]设计了具有更多冗余的新 ZigBee 数据包来减轻 WiFi 干扰；ZIMO[5]利用 MIMO（多输入多输出）和干扰消除技术将 WiFi 和 Zigbee 信号分离成不同的数据流；CrossZig[1]和 PolarScout[6]让 ZigBee 设备检测损坏的数据包中 CTI 的存在情况，然后恢复该数据包。这些方案总是需要修改硬件，甚至需要设计新的收发器，从而使得它们无法应用于当前的商业设备。

还有其他一些方法是通过使异构设备工作在不同频段来避免 CTI[21]-[23]的。例如，G-Bee[22]首先让 ZigBee 设备识别 802.11b WiFi 通道，然后在 WiFi 传输的保护带上传输自己的数据包以避免 CTI，它要求所有的 WiFi 设备都要工作在非重叠的信道上，这在拥挤的 ISM 频段中是很难满足的；EmBee[23]通过设计空子载波让 WiFi 设备预留给 ZigBee 传输的信道，但它需要硬件修改，因为该过程与标准 WiFi 传输过程不兼容。

2.2 WiFi 有效负载编码

近年来出现了一些通过对 WiFi 有效负载编码来设计信号以进行 CTC 数据传输的工作。例如，WEBee[11]设计了 WiFi 有效负载，让 WiFi 信号模拟 ZigBee 信号，从而使该信号能够被标准 ZigBee 接收器正确检测到；BlueFi[12]将类似的想法扩展到 WiFi 到蓝牙的场景，使 WiFi 设备可以通过周密设计的 WiFi 有效负载来传输标准蓝牙信号。虽然这些方法可以使具有 20MHz 或更高信道的 WiFi 模拟成分别仅为 2MHz 和 1MHz 的 ZigBee 和蓝牙信道，但它们对编码的信号有严格的要求，因此需要所有的 WiFi 有效负载都被用于 CTC 数据传输。SLEM[24]和 OfdmFi[25]通过在原始 WiFi 数据比特中插入额外的比特来实现符号级能量调制以传递 CTC 信息。然而，这些方法有其局限性：QAM 点不能总是是指定的最低或最高点，这可能会显著地影响其性能。

3 本文方法

3.1 本文方法概述

本文提出了 SledZig——一种在 WiFi 上的子载波级能量降低机制,以帮助 ZigBee 的传输。SledZig 完全兼容标准的 PHY 和 MAC 层流程,无需对商用 WiFi 和 ZigBee 设备进行任何改动。它利用 WiFi 中 QAM (正交调幅) 调制的特点,在保持 WiFi 发射功率不变的情况下,降低 ZigBee 信道上的 WiFi 信号能量。QAM 是相位和幅度调制方法的结合,使得 QAM 星座点具有不同的功率水平。通过在原始 WiFi 数据比特中插入额外的比特,让与 ZigBee 信道重叠的 WiFi 子载波中的 QAM 点具有最低的功率,而 ZigBee 通道外的 QAM 点保持不变,从而使得 ZigBee 通道上的 WiFi 能量降低高达 14dB。随着能量的减少,ZigBee 网络的性能可以通过提供更多传输机会和避免干扰来显著提高。

SledZig 使用起来也相当简单。对于原始数据位,WiFi 发送器首先通过本文的方法插入额外的比特以生成发送比特。当传输比特通过标准 WiFi 传输过程时,ZigBee 通道上的 WiFi 信号能量会自动降低,从而增强 ZigBee 传输。同时,WiFi 接收器可以通过从接收到的比特中去除多余的比特来轻松获得原始数据比特。

3.2 本文方法详述

(1) QAM 调制提供的机会

在标准 WiFi 传输过程中,WiFi 设备会在 OFDM 模块之前进行 QAM 调制。QAM 调制是相位调制和幅度调制的组合,图 3(a)显示了 QAM-16 的星座图,其中每个星座点包含 4 个数据比特。在 16 个点中,红点的能量最低。当重叠在 ZigBee 信道上的 WiFi 子载波中的 QAM 点全都为红点时,ZigBee 信道上的 WiFi 信号功率会显著降低,如图 3(b)所示。由于这样做可以保持 WiFi 发射功率不变,因此对 WiFi 性能的影响有限。

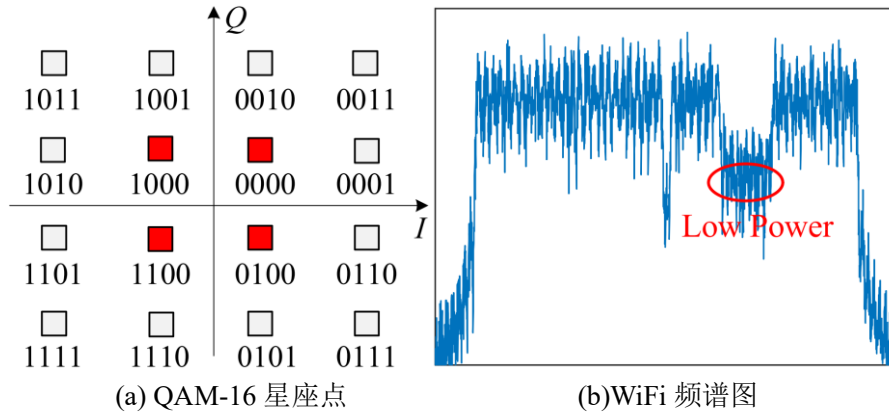


图 3 (a)QAM-16 最低星座点示例; (b)当所有重叠子载波被最低星座点填充时的频谱图

根据上述设计,重叠的 WiFi 子载波中的 QAM 点应该是功率最低的 4 个红点。对于 QAM-16 而言,每个红点携带 4 个比特,且只有 2 个比特对使重叠的 WiFi 子载波功率降低是有用的,本文将这种比特称为显著比特。类似地,每个 QAM-64 和 QAM-256 红色星座点分别具有 4 个和 6 个显著比特,如表 1 所示的阴影位。插入额外比特的目的是使 QAM 点中显著比特的

存在在发送比特中成为事实，即插入额外比特后的发送比特在经过信道编码后那些额外比特将成为显著比特，而 QAM 点中的其他比特可以是任意比特。

表 1 显著比特展示

	QAM-16	QAM-64	QAM-256
Bits	0 0 0 0	0 1 0 0 1 0 0	0 1 0 0 0 1 0 0
	0 1 0 0	0 1 0 1 1 0 0	0 1 0 0 1 1 0 0
	1 0 0 0	1 1 0 0 1 0 1	1 0 0 0 1 0 0
	1 1 0 0	1 1 0 1 1 0 1	1 0 0 1 1 0 0

在清楚插入的额外比特将成为 QAM 点中显著比特的情况后，最关键的问题也就诞生了：应该在原始 WiFi 数据比特中的哪些位置插入什么内容的额外比特？接下来的内容将从 OFDM 子载波中的 QAM 星座点开始，逐步遵循反向的标准 WiFi 传输过程来回答此关键问题。

(2) 显著比特值与位置的确定

原始 WiFi 数据在进入标准 WiFi 传输过程中后首先需要进行信道编码 (Channel Coding)，信道编码的过程包括扰码器、卷积编码器和交织器，如图 4 所示。其中交织器在无线通信系统中被用于减少解码错误，在 SledZig 中，交织的反向过程即解交织被用于确定反卷积编码前显著比特的值和位置。具体来说，前文中已经说明需要将与 ZigBee 信道重叠的 WiFi 子载波中的 QAM 星座点都设为低功率星座点才能达到 ZigBee 信道上的 WiFi 信号功率显著降低的目的，其中低功率星座点是可以确定其值和位置的，因此也就可以确定解交织前的显著比特的值和位置，例如在 QAM-256 中解交织前的显著比特类似于“?100?100” (“?”代表任意值) 中“100”这样的一串比特。解交织后，即可确定反卷积编码前显著比特的值和位置，本文将反卷积编码前的显著比特表示为 $\{v_k, p_k\} (k \in [1, K])$ ，其中 v_k 和 p_k 分别表示第 k 个显著比特的值和位置，例如在 QAM-256 中反卷积编码前的显著比特类似于“??10?0?1??00?”中除“?”以外的比特，而 v_k 和 p_k 则分别代表这些显著比特值和位置。可见，解交织这一操作使得在解交织前聚在一起的显著比特在解交织后被分散到远处的不同位置，为反卷积编码中额外比特值和位置的确定带来了便利。

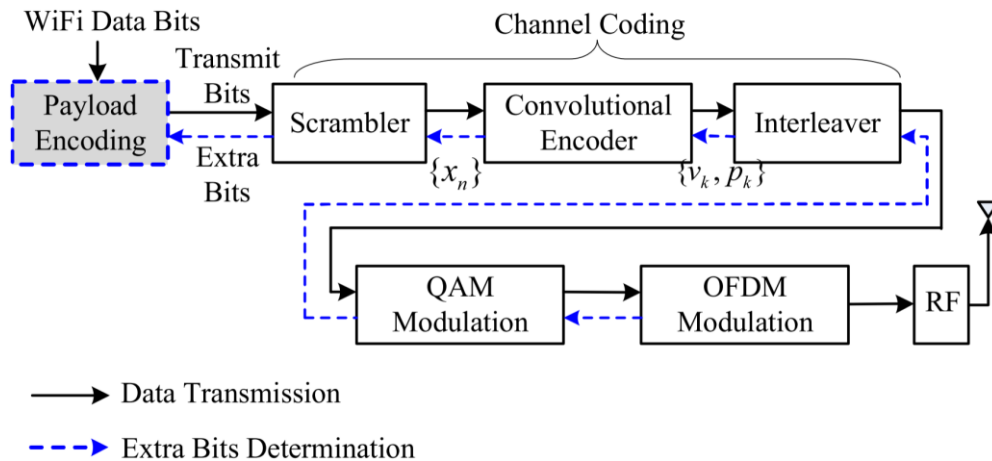


图 4 SledZig 的概况

(3) 额外比特值与位置的确定

在反卷积编码前已经确定了显著比特的值和位置，现在，则要在这个基础上通过反卷积编

码来确定额外比特的值和位置。具体地说，上一部分已经求得了反卷积编码前显著比特的值与位置 $\{v_k, p_k\}$ ，这一部分则要根据 $\{v_k, p_k\}$ 来对经过解交织的数据进行反卷积编码以确定需要插入 WiFi 数据位的额外比特。这个过程的关键是反卷积编码，即卷积编码的反向操作，因此必须先清楚卷积编码的操作，才能知道反卷积编码的具体操作。

802.11 标准在每种 QAM 调制下都推荐了一些编码率，从而导致产生了不同的 WiFi 数据率。1/2 编码率是卷积编码中的基本过程，其中一个输入比特会生成两个输出比特。1/2 卷积编码率过程如图 5 所示。它使用两个生成多项式 $g_0 = (1011011)_2$ 和 $g_1 = (1111001)_2$ 。一个输入比特 x_n 触发两个编码比特 y_{2n-1} 和 y_{2n} 。需要注意的是，输出编码比特不仅由当前输入比特 x_n 决定，还由从 x_{n-1} 到 x_{n-6} 的少量先前比特决定，为了便于描述，这里令 $X_n = [x_n \ x_{n-1} \ x_{n-2} \ x_{n-3} \ x_{n-4} \ x_{n-5} \ x_{n-6}]'$ 。那么这一步生成两个输出比特的编码过程可以被表述为：

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

其中 $GF(2)$ 表示计算是在伽罗瓦域 $GF(2)$ 中的。

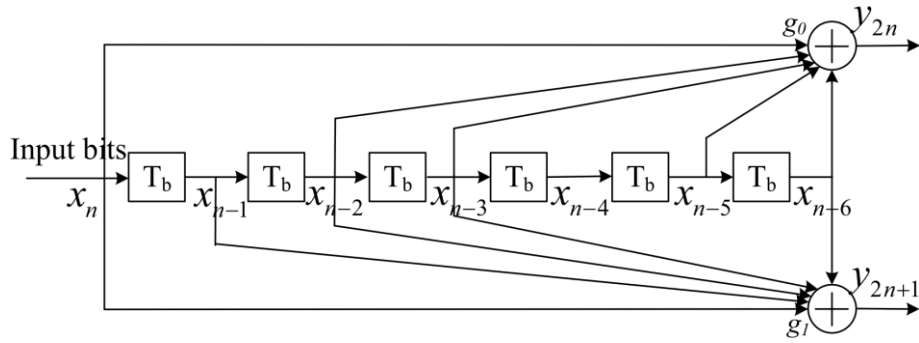


图 5 1/2 卷积编码率的过程

前文已经得到了反卷积编码前显著比特的值与位置 $\{v_k, p_k\}$ ，那么额外比特的确定则可以通过式（1）一一达成。为了便于描述，在表 2 中列出了第一个 OFDM 符号中的显著比特的示例，其中采用了 QAM-16 的调制，且 ZigBee 信道为 CH2。可见，总共有 14 个显著比特，在显著比特中有两种情况，对于后面的分析非常重要：一，给定 n （代表原始数据比特中的第 n 个比特），显著比特是式（1）中 y_{2n-1} 或 y_{2n} 中的一个，另一个则是任意位，例如表 2 中 $k = 9$ 的情况，其中 $n = 63$ 且 $p_k = 2n - 1 = 125$ ，即第 63 个原始数据比特对应着位置为 125 的显著比特，这种比特被称为单显著比特；二， y_{2n-1} 和 y_{2n} 这两个比特都是显著比特，例如 $k = 1$ 和 $k = 2$ 的情况，其中 $n = 15$ ，这种比特被称为双显著比特。接下来就根据这两种情况分别阐述其额外比特求解方式。

表 2 在第一个 OFDM 符号中显著比特的示例

k	1	2	3	4	5	6	7
p_k	29	30	41	42	77	78	89
n	15	15	21	21	39	39	45
k	8	9	10	11	12	13	14
p_k	90	125	138	172	173	183	186
n	45	63	69	86	87	92	93

对于单显著比特的情况，使 X_n 中的 x_n 为额外比特，插入这个额外比特以使式（1）成立，其余 x_{n-6} 到 x_{n-1} 的比特可能是加扰的 WiFi 数据比特或是前面步骤中已经确定的额外比特，它们在当前步骤中都是已知的。于是，通过求解式（1）中相应的方程可以很轻松地得到额外比特 x_n 。

对于双显著比特的情况， X_n 中需要有两个额外比特是未知的才能使式（1）成立，这里让 x_{n-1} 和 x_{n-5} 为待求的额外比特，而 X_n 中其余的比特则可能是加扰的 WiFi 数据比特或是前面步骤中已经确定的额外比特，它们在当前步骤中都是已知的，因此两个额外比特可以通过求解式

（1）来确定。从理论上来说，此时会发现一个坏情况，即比特 x_{n-5} 也用于计算从 $y_{2(n-5)-1}$ 到 $y_{2(n-1)}$ 的先前编码比特，一旦其中存在双显著比特，则式（1）可能没有解，因为总共将会有大约十个方程但未知数却只有三个。然而事实上这种情况在整个额外比特确定的过程中都不会发生，因为无论在 QAM 调制和 ZigBee 信道的哪种组合中，解交织过程都已经将显著比特分散得足够远，足以避免这种情况发生。

接下来总结一下整体的额外比特求解过程。令 WiFi 数据比特为 $\{x'_i\}$ ，插入额外比特生成的发送比特为 $\{x_n\}(n \in [1, N])$ ，需注意，此时的 $\{x'_i\}$ 和 $\{x_n\}$ 都是还未解扰的加扰比特，需要通过解扰 $\{x_n\}$ 才能得到最终的发送比特。从 $\{x'_i\}$ 的第一位开始，设备判断是否触发显著比特。如果是，则根据上述的单/双显著比特的情况采取相应的策略确定额外比特，将求解的额外比特放入 $\{x_n\}$ 中；如果不是，则只将当前的 x'_i 分配给 x_n 。持续进行这个过程直至遍历完所有 WiFi 数据比特 $\{x'_i\}$ 。

（4）解扰

在信道编码过程中加扰器用于避免具有相同值的长比特序列。该模块的 SledZig 设计是根据加扰后的发送比特 $\{x_n\}$ 对其进行解扰来获取最终的发送比特。由于两个模块都是从输入比特到输出比特的一对一映射，因此 SledZig 的解扰过程非常简单。

（5）WiFi 接收器的处理过程

WiFi 接收器在接收到最终发送比特后的处理过程非常简单：接收器首先进行标准的 WiFi 接收过程来获取最终发送比特，然后去除额外比特以得到原始的 WiFi 数据比特。额外比特的位置在最终发送比特中是固定的，它们由三种信息确定：ZigBee 信道、QAM 调制和编码率。后两个信息可以直接从 WiFi 数据包的 PLCP（物理层汇聚协议）头部中获得[2]，这里的关键问题是获取 ZigBee 信道。有了最终发送比特，WiFi 接收器就可以进行信道编码和调制，然后观察 QAM 点并确定 ZigBee 信道：与 ZigBee 信道重叠的 WiFi 子载波中的 QAM 点都是最低的。这个过程完全兼容 802.11 标准。

4 复现细节

4.1 实验设置

要实现 SledZig 实验原型，需要同时发送 WiFi 和 ZigBee 信号。本人使用一个 USRP 作为 WiFi 发射器（WiFi Tx）来发送遵循 IEEE 802.11 标准的 WiFi 信号，然后使用两个 TelosB 设备

分别作为 ZigBee Tx 和 Rx 来测试 ZigBee 的性能，如图 6 所示。



图 6 发送 WiFi 信号的 USRP 和收发 ZigBee 信号的 TelosB 设备

4.2 性能测试过程

在上述基本的实验设置下，本人通过 GNURadio 平台来控制 WiFi Tx 所发送的 WiFi 信号，并分别通过三种不同设置的对比来进行后续的实验结果分析：一，不发送 WiFi 信号，只发送 ZigBee 信号，在 ZigBee Rx 处观察 ZigBee 信号的吞吐量；二，利用 GNURadio 平台控制 WiFi Tx 发送随机的但未经 SledZig 设计处理的数据信号，同时发送 ZigBee 信号，在 ZigBee Rx 处观察 ZigBee 信号的吞吐量；三，利用 GNURadio 平台控制 WiFi Tx 发送随机的但经过 SledZig 设计处理的数据信号，同时发送 ZigBee 信号，在 ZigBee Rx 处观察 ZigBee 信号的吞吐量。GNURadio 平台具体设置如图 7 和图 8 所示。

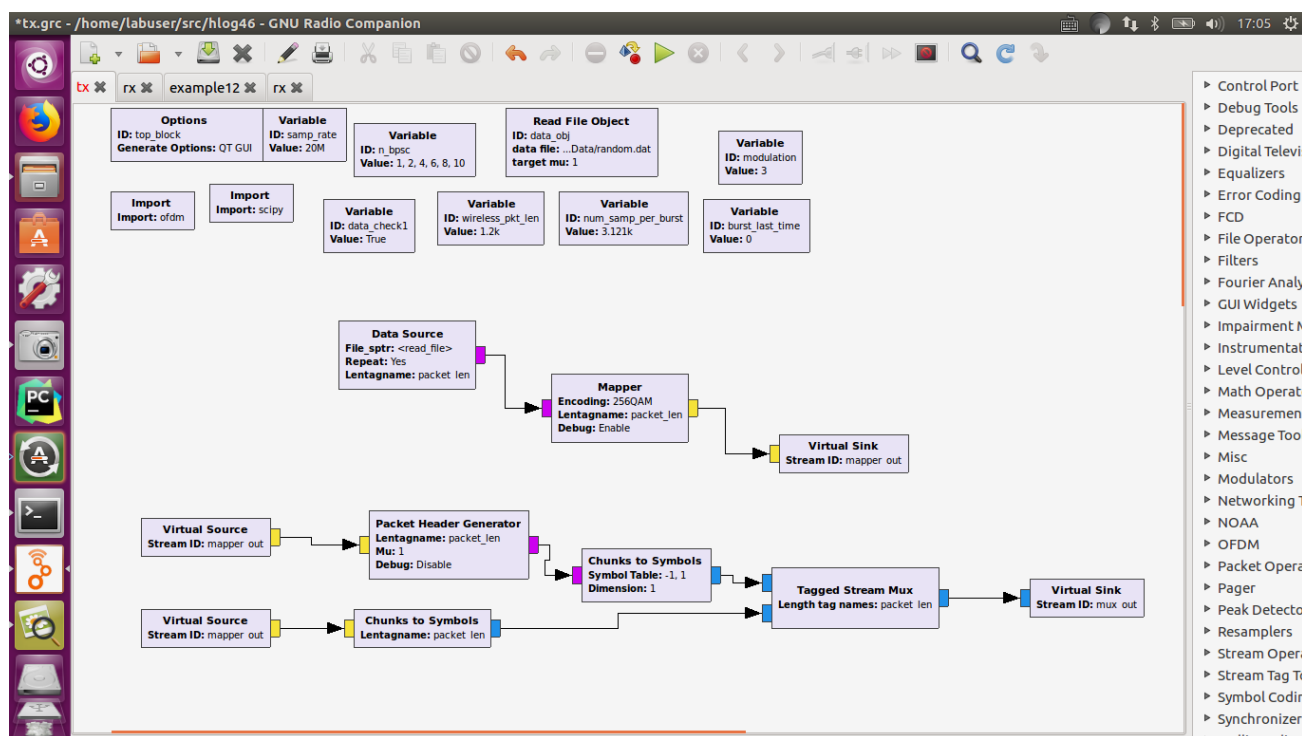


图 7 GNURadio 平台控制 WiFi Tx 发送 WiFi 信号的具体设置

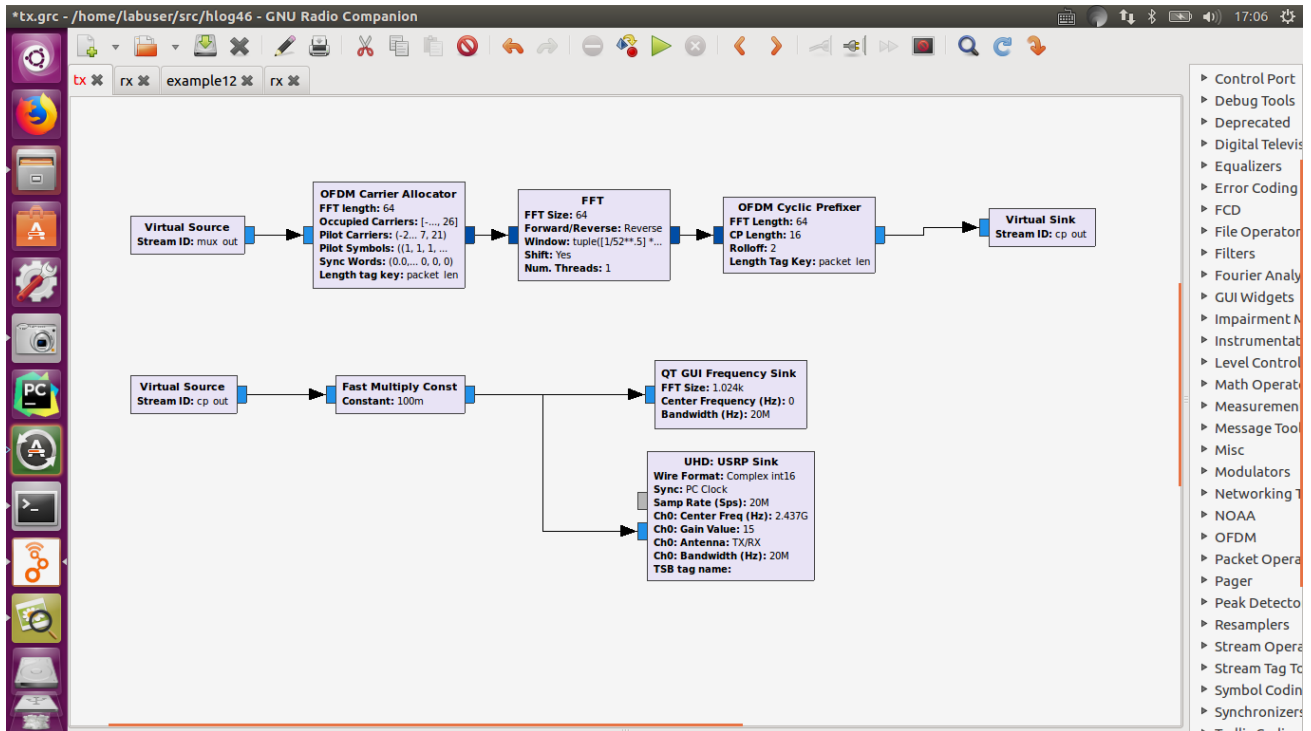


图 8 GNURadio 平台控制 WiFi Tx 发送 WiFi 信号的具体设置（续图 7）

5 实验结果分析

1、不发送 WiFi 信号，只发送 ZigBee 信号，在 ZigBee Rx 处观察 ZigBee 信号的吞吐量。实验结果如图 9 所示。可见，此时 ZigBee 的信号传输与传播没有受到任何影响，其吞吐量较高、延迟时间较低，且 ZigBee Rx 设备处的指示灯也一直保持亮灯状态。

```
labuser@LabComputer: ~/src/hlog46/MyBaseStation
time:3549,throughput:51.39475908706678k
time:3226,throughput:56.54060756354619k
time:3315,throughput:55.022624434389144k
time:3228,throughput:56.50557620817844k
time:4049,throughput:45.04816003951593k
time:3828,throughput:47.64890282131662k
time:4648,throughput:39.24268502581756k
time:3578,throughput:50.978200117943k
time:4359,throughput:41.844459738472125k
time:4984,throughput:36.597110754414125k
time:3530,throughput:51.671388101983005k
time:3330,throughput:54.77477477477478k
^Z
[13]+  Stopped                  java net.tinyos.tools.MyListen -comm serial@/dev/
ttyUSB0:telosb
labuser@LabComputer:~/src/hlog46/MyBaseStation$ java net.tinyos.tools.MyListen -
comm serial@/dev/ttyUSB0:telosb
serial@/dev/ttyUSB0:115200: resynchronising
serial@/dev/ttyUSB0:115200: bad packet
serial@/dev/ttyUSB0:115200: bad packet
time:3395,throughput:53.726067746686304k
time:3220,throughput:56.64596273291925k
time:3227,throughput:56.52308645801054k
```



图 9 实验 1: ZigBee 信号的吞吐量展示和 TelosB 设备的工作状态

2、利用 GNURadio 平台控制 WiFi Tx 发送随机的但未经 SledZig 设计处理的数据信号，同时发送 ZigBee 信号，在 ZigBee Rx 处观察 ZigBee 信号的吞吐量。实验结果如图 10 所示。可见，此时 ZigBee 的信号受到了极大的影响，其吞吐量较低，而延迟时间较高，且 ZigBee Rx 设

备处的指示灯多数时间保持着灭灯状态，表示接收 ZigBee 信号较困难。

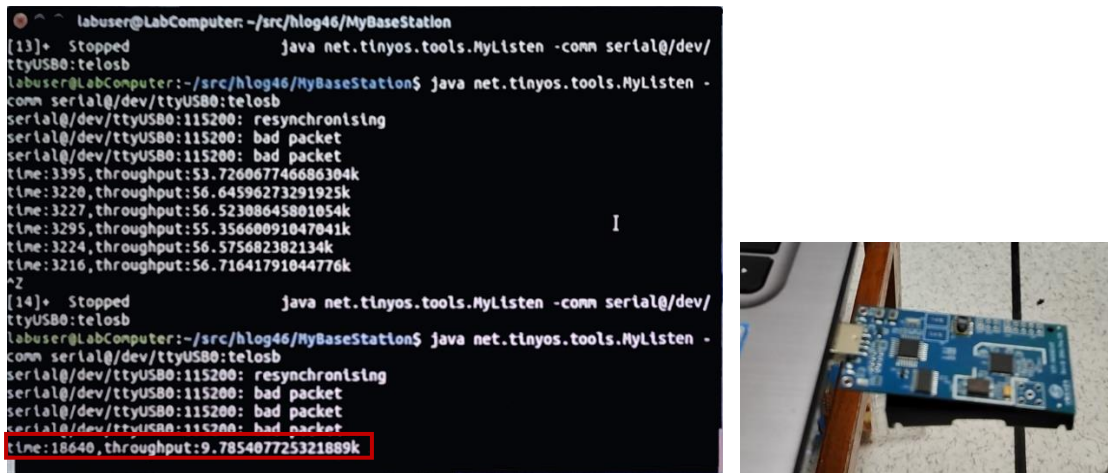


图 10 实验 2: ZigBee 信号的吞吐量展示和 TelosB 设备的工作状态

此时 WiFi 信号的频谱图如图 11 所示。WiFi 信号功率稳定，但严重影响了 ZigBee 信号的收发。

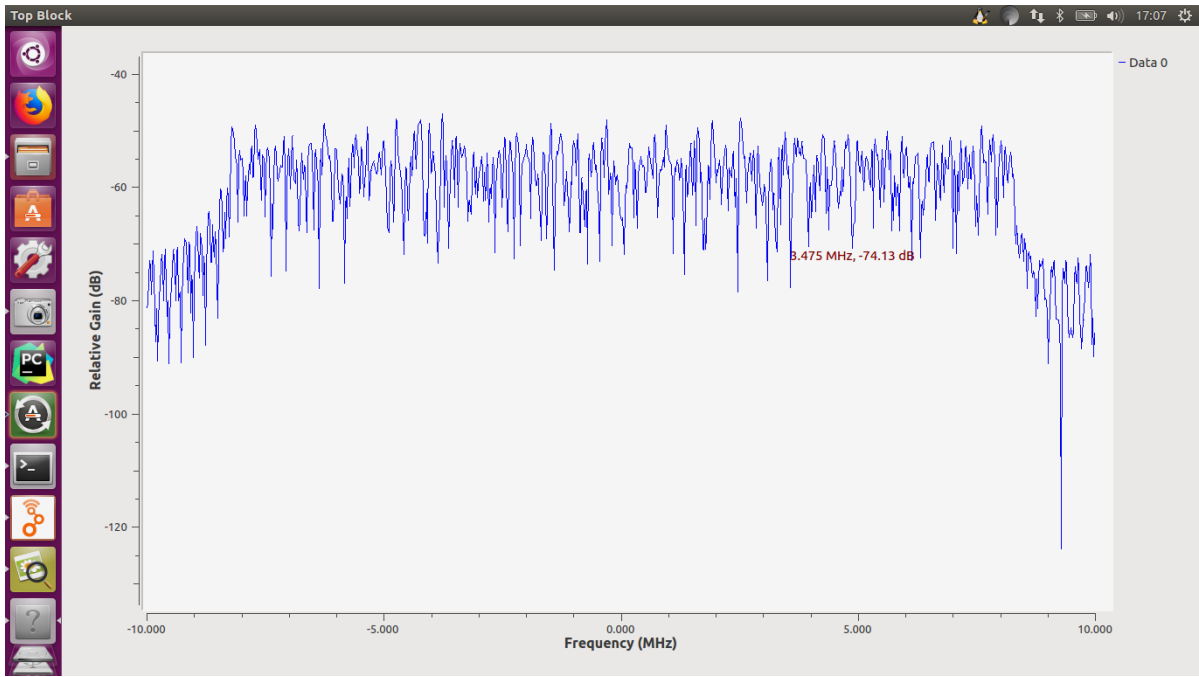


图 11 WiFi Tx 发送的未经处理的 WiFi 信号频谱图

3、利用 GNURadio 平台控制 WiFi Tx 发送随机的但经过 SledZig 设计处理的数据信号，同时发送 ZigBee 信号，在 ZigBee Rx 处观察 ZigBee 信号的吞吐量。实验结果如图 12 所示。可见，在 SledZig 的帮助下此时 ZigBee 的信号并没有受到 WiFi 信号影响，其吞吐量和延迟时间恢复到了正常状态，且 ZigBee Rx 设备处的指示灯也一直保持亮灯状态，可以做到正常的信号接收。此时 WiFi 信号的频谱图如图 13 所示。可以看到，其中有一段的功率较低，这一段正是 WiFi 和 ZigBee 重叠的信道区域。因此，本文提出的 SledZig 可以在不影响 WiFi 传输功率的基础上有效解决 WiFi 和 ZigBee 的跨技术共存问题。

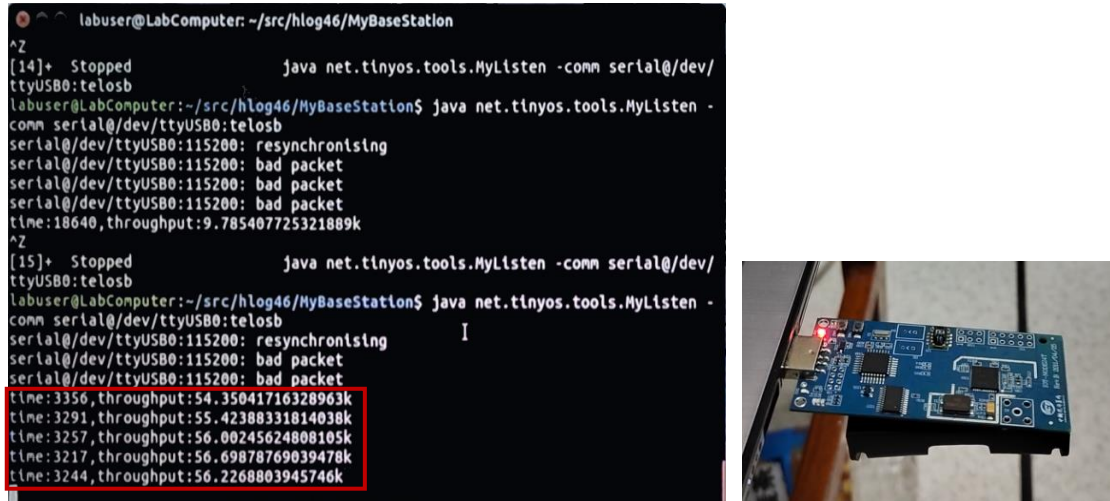


图 12 实验 3: ZigBee 信号的吞吐量展示和 TelosB 设备的工作状态

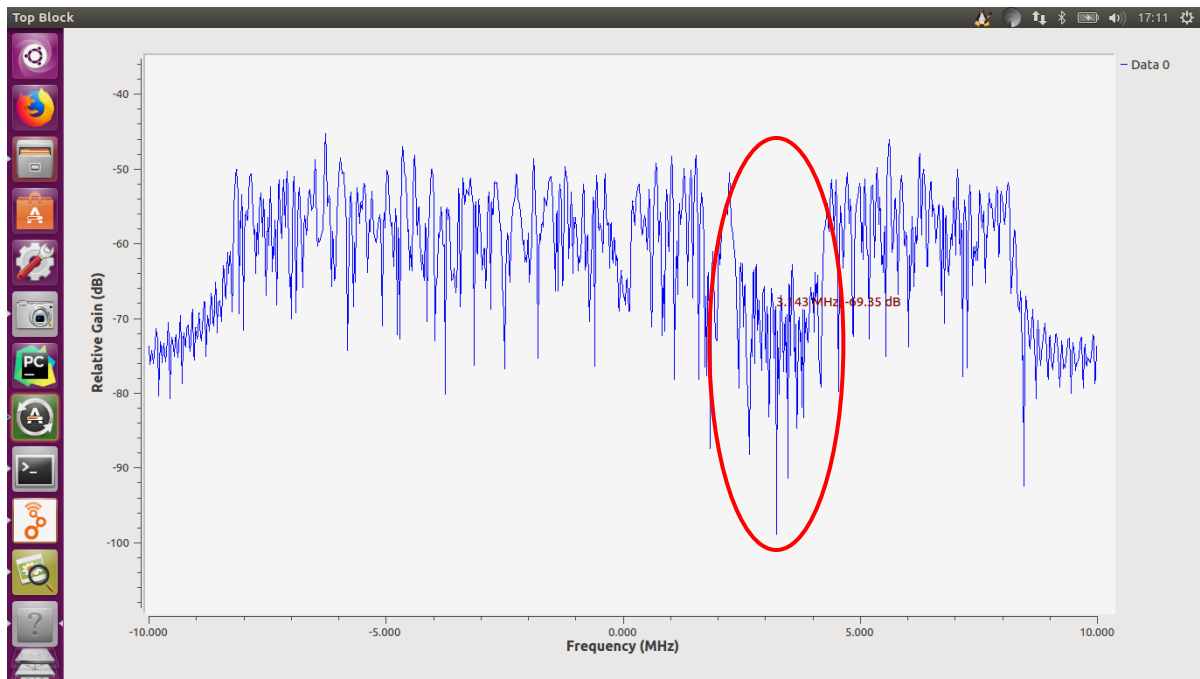


图 13 WiFi Tx 发送的经过处理的 WiFi 信号频谱图

6 总结与展望

本文提出的 SledZig 的设计可以帮助 WiFi 和 ZigBee 网络解决跨技术共存的问题，且效果较好，但在应用了这样的设计后却不可避免地使得冗余信息增加了，因此在传播过程中 WiFi 信号的吞吐量有些降低（约 10%）。在本次的复现工作中，本人完成了实验场景的搭建并进行了实验，在已有源码的基础上补充了 QAM-1024 场景下的解交织方法。后续的工作主要是对 USRP 平台中传输的参数进行调试，比如卷积编码率、调制方式等，同时通过对吞吐量进行量化来寻求提升方法，不断调试以达到更好的效果。

参考文献

- [1] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy, “CrossZig: combating cross-technology interference in low-power wireless networks,” in Proc. of the ACM/IEEE IPSN, 2016.
- [2] IEEE Computer Society. 802.11, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput,” 2009.
- [3] IEEE Computer Society. 802.15.4, “IEEE Standard for low-rate wireless networks,” 2016.
- [4] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, “Surviving wifi interference in low power ZigBee networks,” in Proc. of the ACM SenSys, 2010.
- [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 flash 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] X. Zhang and G. S. Kang, “Enabling coexistence of heterogeneous wireless systems: case for ZigBee and WiFi,” in Proc. of the ACM MobiHoc, 2011.
- [8] 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.
- [9] Y. Wang, Q. Wang, Z. Zeng, G. Zheng, and R. Zheng, “Wicop: Engineering wifi temporal white-spaces for safe operations of wireless body area networks in medical applications,” in Proc. of the IEEE RTSS, 2011.
- [10] X. Zhang and K. G. Shin, “Gap sense: lightweight coordination of heterogeneous wireless devices,” in Proc. of the IEEE INFOCOM, 2013.
- [11] Z. Li and T. He, “WEBee: physical-layer cross-technology communication via emulation,” in Proc. of the ACM MobiCom, 2017.
- [12] H.-W. Cho and K. G. Shin, “BlueFi: Bluetooth over WiFi,” in Proc. of the ACM SIGCOMM, 2021.
- [13] D. Xia, X. Zheng, L. Liu, C. Wang, and H. Ma, “c-Chirp: towards symmetric cross-technology communication over asymmetric channels,” ACM Transactions on Sensor Networks, vol. 29, no. 3, pp. 1169–1182, Jun. 2021.
- [14] 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.
- [15] Z. Yin, Z. Li, S. M. Kim, and T. He, “Explicit channel coordination via cross-technology communication,” in Proc. of the ACM MobiSys, 2018.
- [16] W. Wang, T. Xie, X. Liu, and T. Zhu, “ECT: Exploiting Cross-Technology Transmission for Reducing Packet Delivery Delay in IoT Networks,” ACM Transactions on Sensor Networks, vol. 15, no. 2, Feb. 2019.
- [17] Y. Li, Z. Chi, X. Liu, and T. Zhu, “Chiron: concurrent high throughput communication for iot

devices.” in Proc. of the ACM MobiSys, 2018.

- [18] Z. Chi, Y. Li, Z. Huang, H. Sun, and T. Zhu, “Simultaneous bidirectional communications and data forwarding using a single ZigBee data stream,” in Proc. of the IEEE INFOCOM, 2019.
- [19] W. Chen, Z. Yin, and T. He, “Global cooperation for heterogeneous networks,” in Proc. of the IEEE INFOCOM, 2020.
- [20] Z. Yu, P. Li, C. A. Boano, Y. He, M. Jin, X. Guo, and X. Zheng, “BiCord: bidirectional coordination among coexisting wireless devices,” in Proc. of the IEEE ICDCS, 2021.
- [21] Y. Wang, X. Zheng, L. Liu, and H. Ma, “CoHop: quantitative correlation-based channel hopping for low-power wireless networks,” ACM Transactions on Sensor Networks, vol. 17, no. 2, pp. 1–29, Jun. 2021.
- [22] Y. Chae, S. Wang, and S. M. Kim, “Exploiting WiFi guard band for safeguarded ZigBee,” in Proc. of the ACM SenSys, 2018.
- [23] R. Chen and W. Gao, “Enabling cross-technology coexistence for extremely weak wireless devices,” in Proc. of the IEEE INFOCOM, 2019.
- [24] J. Yao, X. Zheng, R. Xie, and K. Wu, “Cross-technology communication for heterogeneous wireless devices through symbol-level energy modulation,” IEEE Transactions on Mobile Computing, Mar. 2021. DOI: 10.1109/TMC.2021.3065998.
- [25] 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.