Bluebee：通过

物理仿真

                      江志萌尹若凤刘文超

    计算机科学系计算机科学系

工程，明尼苏达工程大学，明尼苏达工程大学，明尼苏达大学jiang832@umn.edu yinxx283@umn.edu liu4189@umn.edu

                           李志军宋敏金天河

    计算机科学系和计算机科学系

工程，明尼苏达大学乔治梅森大学工程，明尼苏达大学lixx5427@umn.edu song@gmu.edu tianhe@umn.edu

# 摘要

跨技术通信是近年来针对ISM频段异构无线技术共存问题提出的一种很有前途的解决方案。现有的工作只使用粗粒度的包级信息进行跨技术调制，吞吐量很低（例如，10bps）。我们的方法称为bluebee，通过使用蓝牙无线电模拟合法的zigbee帧，提出了一个新的方向。独特的是，Bluebee通过只选择蓝牙帧的有效载荷来实现双重标准遵从性和透明性，无需在蓝牙发送器和Zigbee接收器处更改硬件或固件。我们在usrp和商用设备上的实现表明，blueee的准确率可以达到99%以上，吞吐量比目前最先进的ctc报告快1000倍。

CCS概念

•网络→无线个人区域网络；

# 关键词

跨技术通信；信号仿真；蓝牙低能耗、Zigbee、物联网

ACM参考格式：

蒋文超，尹志萌，刘若凤，李志军，金宋民，和田和。2017。Bluebee：通过物理仿真实现的速度快1000倍的跨技术通信。第15届ACM嵌入式网络传感器系统会议记录（SENSYS&apos;17）。ACM，纽约，纽约，美国，13页。https://doi.org/http://dx.doi.org/10.1145/3131672.3131678



小精灵李志军是明尼苏达大学客座教授，正式隶属于中国哈尔滨工业大学。

γ李志军和田鹤是这部作品的对应作者。

|  |
| --- |
| 允许制作本作品的全部或部分数字或硬拷贝，供个人或课堂使用，不收取任何费用，但前提是复制品的制作或分发不是为了盈利或商业利益，且复制品上应附有本通知和第一页的完整引文。必须尊重ACM以外的其他人拥有的本作品组成部分的版权。允许赊账提取。否则，若要复制或重新发布、在服务器上发布或重新分发到列表，则需要事先获得特定许可和/或支付费用。从permissions@acm.org请求权限。  *Sensys&apos;172017年11月6日至8日，荷兰代尔夫特*  ？2017计算机械协会。ACM ISBN 978-1-4503-5459-2/17/11…$15.00 https://doi.org/http://dx.doi.org/10.1145/3131672.3131678 |

# 1简介

无线设备的主体在过去十年中经历了爆炸性的增长，在新兴的物联网（IoT）时代，预计到2020年将增长到200亿[12]。密集的部署导致高度共存的无线环境，长期以来一直被视为一个恶劣的环境，严重干扰。然而，最近的研究表明，共存提供了独特的机会——通过利用异构无线技术之间的特殊功能，协作使它们能够超越独立操作。例如，在zifi[44]中，在低功率zigbee无线电的帮助下，耗电量大的wifi接口的能源消耗显著减少，只有在附近发现接入点时，wifi才会打开。

传统的异构设备间通信方式是部署多个无线网关，存在硬件成本高、网络结构复杂、进出网关流量增加等缺点。为了解决这些问题，最新的文献介绍了跨技术通信（CTC）技术，该技术使用传统设备在物理层不兼容的异构无线设备之间实现直接通信。这种技术通常使用分组级调制，其中分组的定时[22]和持续时间[7]的组合传送数据。尽管它们有效，但是由于采用粗粒度“包”作为调制的基础（类似于典型数字通信中的“脉冲”），比特率本质上是有限的。例如，在现有技术中，ble到zigbee通信的比特率被限制在18bps[22]。这不仅限制了使用，而且表明如果用于传统的Zigbee和蓝牙，与250kbps和1Mbps相比，频谱效率低。

本文介绍了Bluebee，通过物理层仿真为实际的CTC系统铺平了道路。通过巧妙地选择蓝牙包中的有效载荷位，Bluebee有效地将Zigbee包封装在蓝牙包有效载荷中。这与传统的zigbee设备完全兼容，同时达到了250kbps的zigbee比特率上限。换言之，Bluebee不需要对蓝牙发射器或Zigbee接收器进行任何硬件或固件更改，提供与现有数十亿商品物联网设备、智能手机、PC和外围设备的完全兼容性（即作为应用程序实现）。

事实上，通过蓝牙传输的模拟zigbee包无法被zigbee接收器区分。这是令人惊讶的，尤其是当蓝牙（1兆赫）的带宽只有Zigbee（2兆赫）的一半时。Bluebee的设计源于两个关键的技术见解：（i）蓝牙和Zigbee调制技术的相似性；（ii）Zigbee解调（OQPSK/DSSS）的容错性。具体地说，这两种技术都使用样本之间的相位差（称为相移）来表示符号，这使得仿真成为可能。尽管Zigbee信号由于蓝牙带宽较窄而无法完全仿真，但Bluebee经过优化设计，使得不可避免的误差最小化，并保持在Zigbee的OQPSK/DSSS解调器（即，通过Zigbee的OQPSK/DSSS解调器成功地校正了该误差）的公差范围内。Bluebee只需在蓝牙数据包有效负载中放置特定的位模式，就可以轻松地在商品蓝牙设备上运行。它在90%的帧接收比（frr）下达到250kpbs，比最新技术[22]快10000倍。此外，Bluebee有效地利用了蓝牙的跳频特性，支持在不同信道上运行的设备之间的并发通信。最后，Bluebee在动态无线信道条件下提供可靠的通信。这项工作的贡献是三倍。

•我们设计了Bluebee，这是第一个在合法蓝牙数据包的有效载荷内模拟合法Zigbee帧的CTC技术。该设计不需要对发射机（蓝牙）或接收机（Zigbee）的硬件或固件进行任何修改，从而实现与数十亿现有商品设备的完全兼容。

•我们解决了信号仿真的几个独特挑战，包括（i）使用蓝牙信号的优化zigbee相移仿真，（ii）支持蓝牙跳频下的并发通信和低占空比操作，以及（iii）动态信道条件下的链路层可靠性。这些解决方案为异构设备之间的其他信号仿真提供了一般性的见解。

•我们在USRP平台和商品设备上设计和实现Bluebee。我们的大量实验表明，Bluebee在不同的环境和设置下建立了高吞吐量和可靠的通信。与最先进的CTC从蓝牙到Zigbee的18bps速率相比[22]，Bluebee 225kbps的可靠吞吐量表明性能提高了10000倍以上！

# 2动机

随着wifi、蓝牙、zigbee等无线技术的飞速发展，ism频段面临着跨技术干扰（cti）和信道低效率的问题[17、24、40]。这是因为共存于ISM频段的无线技术具有异构的物理层，无法直接通信，无法有效协调信道的使用。为了实现有效的信道利用，传统的方法是使用多信道网关。近年来，研究人员也提出了跨技术通信（CTC）技术作为一种很有前途的信道协调解决方案。然而，传统的网关方法和现有的ctc技术由于其固有的局限性，在信道协调方面都不能取得很好的效果。

•网关限制。多无线电网关是连接多技术通信的常用而直接的解决方案[13、14、20、26、29]。然而，网关不仅会带来额外的硬件成本，还会带来劳动密集型的部署成本，这对于移动和ad hoc环境来说是望而却步的。此外，双无线网关通过使ISM频带内的业务量加倍来增加业务开销，这进一步加剧了交叉技术干扰。

•分组级CTC的限制。近年来的跨技术通信旨在实现异构无线技术之间的直接通信，从而实现显式的信道协调。例如，异构设备可以以类似于802.11协议中的rts/cts的方式分配信道[1]，从而导致更好的信道效率。不幸的是，据我们所知，现有的ctc设计[7，22，43]依赖于稀疏的分组级信息，例如信标定时[22]和多分组序列模式[37]，引入了至少数百毫秒的延迟。这种延迟使得现有的解决方案无法实时有效地协调信道。

相对于网关方法和现有的ctc方法的局限性，bluebee能够在几毫秒内直接从蓝牙无线电发送zigbee包，这是第一次使信道协调成为可能。在本文中，虽然我们的描述将基于特定的蓝牙协议bluetooth low energy（bluetooth bluetooth bluetooth，bluetooth low energy，blue low energy，bluetooth low energy，bluetooth low energy，bluetooth low

蓝牙协议，如Bluetooth Classic（在第节中讨论

7.1）。

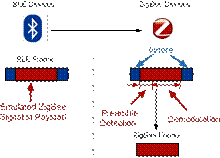


图1:Bluebee的系统架构。

# 3个蓝莓

概述。BlueBee是一种从BLE到ZigBee的高速CTC通信，同时兼容ZigBee和BLE协议。Bluebee的基本思想如图1所示——Bluebee通过仔细选择有效负载字节，将合法的Zigbee帧封装在合法的BLE帧的有效负载内。在phy层，所选择的有效载荷类似于（即，模拟）合法zigbee帧的信号。当Bluebee模拟的ble包到达Zigbee设备时，检测有效负载部分

（通过前导）和解调，就像来自zigbee发送者的任何其他zigbee包一样。我们注意到，ble帧的头和尾与zigbee不兼容，并且自然地被忽略，或等效地被视为噪声。事实上，这样的设计使得bluebee透明；在发送方，ble设备无法区分它是普通的ble包还是包含模拟的zigbee帧，因为它仅仅是有效载荷中的字节模式。相反，在接收器处，由于phy层波形不可区分，zigbee设备无法判断帧是来自zigbee设备还是由ble设备模拟。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 成本 | 光谱  效率 | 向前推进 | 多通道  CTC |
| 网关 | 培养基 | 培养基 | 高 | 不支持 |
| 埃森斯[ 7 ] | 低 | 低 | 低 | 不支持 |
| 免费蜜蜂〔22〕 | 低 | 培养基 | 低 | 不支持 |
| *乙*2*W*2 〔11〕 | 低 | 培养基 | 低 | 支持 |
| 蓝蜂 | 低 | 高 | 高 | 支持 |

表1:Bluebee和现有CTC解决方案的比较

独特的特点。在表1中，我们说明了Bluebee作为第一物理层CTC与网关方法和最先进的包级CTC方法相比的技术优势。Bluebee通过提供异构设备之间的直接通信，克服了现有网关方法的不足。与网关相反，Bluebee不会产生部署成本或额外流量。同时，它提供的通信吞吐量和传输延迟比目前的CTC要高得多。此外，blueee通过ble通信中固有的跳频实现了多信道并发ctc。

Bluebee在兼容性方面也有一些创新和独特的特性：首先，它是从ble到Zigbee的第一个CTC设计，既不需要硬件也不需要固件更改。其他设计要求接收器至少进行固件更改[7、22、37]。第二，bluebe是“双重标准符合性”，即zigbee和ble接收机都可以接收和解调帧。

# 4蓝莓设计

本节详细说明Bluebee设计。

## 4.1背景

我们首先简要介绍了一个ble发射机和一个zigbee接收机如何与blueee相关工作，然后介绍了信号仿真的可行性。

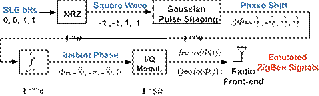


图2:ble作为带gfsk调制的发射机。

无线电发射机。ble使用高斯频移键控

（gfsk）调制，通常通过随时间的相移来实现。图2示出了从有效载荷比特到从步骤（i）到（iv）的对应无线电波的整个过程。在（i）布尔比特中，首先经过一个不归零（nrz）模块，该模块将布尔比特序列调制成振幅为-1或1的平方波序列。由于每一个波的长度为1微秒，并且只携带一个比特，这就导致了BLE的1Mbps比特率。（ii）该波通过高斯低通滤波器，该滤波器将波塑造成带限信号。基带信号与载波相乘时的相移为±∏/2。（i i i）将一系列波的积分取t，得到相对于时间的相位（即瞬时相位）。这本质上是上一步累积相移的时域表示。（iv）同相和正交（I/Q）信号分别通过瞬时相位的余弦和正弦进行计算，余弦和正弦分别乘以载波并通过BLE射频前端推入空气中。[〔1〕](" \l "_ftn1" \o ")

Bluebee的目标是构造可由商品Zigbee接收器解调的时域波形。换句话说，在BLE上模拟Zigbee信号。为此，我们假设包含我们选择的数据的zigbee信号从ble rf前端发射，并相应地反向工程步骤（iv）到（i）。在步骤（iv）中，以ble采样率（1msps）对空气中的zigbee信号进行采样。从采样的i/q信号中，得到相应的瞬时相位。反转步骤（iii）产生连续ble样本之间的相移，其中通过反转步骤（ii）找到对应的方波序列。最后，这些波被映射到可自由设置的ble分组有效载荷处的数据位，指示仅通过使用正确的位设置ble分组有效载荷来模拟目标zigbee信号。

这种方法使得商品zigbee无线电能够无缝地将仿真波形解调为合法的zigbee分组，而不必对ble的gfsk调制器进行任何改变。然而，由于各种限制，例如ble（1MHz）比zigbee（2MHz）的带宽更窄（将在本节的后面部分讨论），因此这种仿真并不简单。



图3:Zigbee作为OQPSK解调的接收器。

Zigbee接收器。如图3所示，bluebee使得能够通过标准偏移正交相移键控（oqpsk）解调过程发送可由任何商品zigbee设备解调的模拟zigbee包。这由步骤（a）开始，其中zigbee通过模数转换器（adc）捕获重叠的2.4ghz ism上的ble信号，以获得i/q采样。一对i/q样本通常被称为复合样本s（n）=i（n）+jq（n）。在步骤（b）中，从arctan（s（n）×s（n1））计算连续复样品之间的相移，其中s（n1）是s（n1）的共轭。在步骤（c）中，正相移和负相移量化为1和-1，对应于zigbee芯片1和0。小精灵小精灵

最后，在（d）中，通过查找dsss中预定义的符号到芯片映射表（表2），将32个zigbee芯片映射到zigbee符号。共有16个不同的符号，每个符号表示loü16=4位。我们注意到在噪声/干扰面前2

|  |  |
| --- | --- |
| 符号（4位） | 芯片序列（32位） |
| 0 0 0 0 | 11011001110000110101001000101110 |
| 0 0 0 1 | 11101101100111000011010100100010 |
| ... | ... |
| 1 1 1 1 | 11001001011000000111011110111000 |

### 表2:Zigbee中的符号到芯片映射（802.15.4）

相位可能会出现错误（+∏-），这会导致反向切屑（1∏0）。在这种情况下，选择汉明距离最小的最近符号。

## 4.2仿真的机遇与挑战

概念上，由于两个关键的技术见解，通过ble模拟zigbee信号是可能的。首先是ble和zigbee调制技术的相似性。也就是说，ble的gfsk和zigbee的oqpsk通常利用连续样本之间的相移来指示符号（zigbee的芯片）。此外，zigbee只考虑相位的符号（+或-）而不考虑特定的相位值，这在仿真中提供了很大的灵活性。然而，挑战来自于ble（1MHz）的带宽仅为zigbee（2MHz）的一半。这从根本上限制了ble的相移速率。换言之，ble中的相移不够快，无法表示所有zigbee芯片，从而导致仿真中不可避免的错误。Bluebee仿真的第二个关键组件（即Zigbee中的DSSS）弥补了这一不足。

DSSS将32位芯片序列映射到4位符号（表2），为抗噪声和干扰的稳健性留有公差。由于这一裕度，如果接收到的和理想芯片序列之间的汉明距离在12的阈值范围内（可以调整到20[24]），则可以正确地解码zigbee符号。这种公差裕度可以用来从带宽不对称造成的不可避免的误差中恢复。在下面的章节中，我们将详细说明这两个见解，以及Bluebee是如何设计来有效地探索它们以使CTC成为可能的。

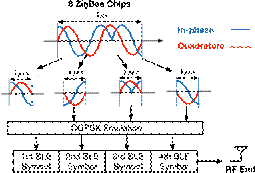


图4：用BLE模拟Zigbee

## 4.3 OQPSK仿真

在这里，我们演示了用ble模拟zigbee的oqpsk调制，这是一个非常重要的问题，因为ble的带宽比zigbee窄（1兆赫比2兆赫）。图4以8个zigbee芯片为例说明了仿真过程，首先将序列切割成两个持续时间为1微秒的芯片片（一个同相芯片和一个正交芯片）。然后将每个芯片片仿真为ble符号，我们将在下一节中详细讨论。我们注意到，引入的技术只涉及设置ble包有效负载的位模式，而不强制对硬件或固件进行任何更改。

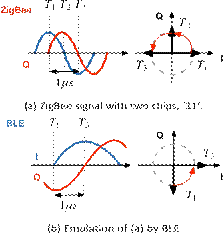
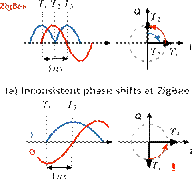
**我**

图5:（a）Zigbee信号，表示两个芯片，“11”，从TTOT到TTOT的相移均为正（π/2）。（b）是（a）的模拟信号，由ble（实际上是ble符号“1”）。当输入Zigbee接收器时，该信号在T、T和T处采样，给出两个连续的正（π/4）相移。这将产生“11”的Zigbee芯片，这表明模拟成功。1 22 3 123

现在让我们来看看如何在图4中划分的两片芯片上执行仿真。回想一下，oqpsk（即zigbee）观察连续样本之间的相移，其符号被转换为-1和1的芯片（图3中的步骤（a）和（b））。图5a中的左图描绘了包含两个“11”芯片的zigbee信号（未模拟），其中每0.5μs对三个连续样本的计时t-tare，zigbee采样率。在右侧，用箭头绘制对应时间点的样本星座图。tand-tisπ/2箭头间的相移。因为是正值，所以它被转换成“1”的芯片。下一个芯片在samplestandt之间进行类似的计算，这也会产生一个“1”芯片。1 3 1 2 2 3

现在，我们证明上述zigbee信号可以被ble成功地模拟，如图5b的左边所示。尽管该信号看起来不同于zigbee信号（图5a的左边），但它仍然向zigbee接收器传送相同的“11”芯片。这里的关键是只考虑相移的迹象（而不是数量）。为了理解这一点，我们首先注意到，图5b中的左边反映了ble的带宽仅为zigbee的一半，即，表示i/q信号的正弦曲线具有一半的频率，或者等效地是周期的两倍。当该信号馈入zigbee接收器并在t-t采样时，所得星座如图5b中所示。从图中可以看出，t and t isπ/4（即正）之间的相移，其产生“1”芯片。这同样适用于tand t之间的相移。这表示图5b）中左侧的ble信号确实在zigbee接收器处产生与图5a中左侧的zigbee信号相同的芯片序列‘11’。换句话说，zigbee信号被ble成功模拟。1 31 2 2 3

事实上，从ble的角度来看，图5b的左边的信号仅仅是表示π/2的相移的ble信号。这是因为ble的采样率是zigbee的一半，这是由于带宽差和相应的nyquist采样率。具体来说，ble样品和twhose样品的相位差为π/2。相反地，通过让ble发送对应于π/2相移的比特，ble设备能够将“11”的芯片序列发送到zigbee接收器。这是Bluebee的关键使能器，在Bluebee中，Zigbee包仅通过有效负载位模式封装在一个BLE包中。1 3

**我**

**Q**

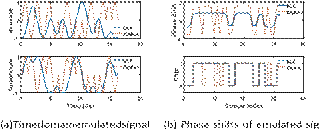
**布尔**

（b）不完善的信号模拟

### 图6：不一致的Zigbee相移的影响

从图5b中的示例中，我们发现，ble中的单相移位被解释为zigbee中根据带宽差的两个相位移位。也就是说，ble具有较低的自由度，在这里它可以每1微秒改变一次相移（－∏+），而zigbee为0.5微秒。因此，虽然zigbee芯片序列是“11”或“00”（“一致相位”在后面，因为相移保持一致在+或-）可以被完美地模拟，但序列“01”或“10”的情况并非如此。图6a示出了“10”的zigbee芯片序列。如图6b所示，ble将其模拟为“11”（在图中）或“00”，在两种情况下都会产生1个芯片错误。虽然由于ble的带宽较窄，这种芯片错误是不可避免的，但有趣的是，它对解码比特的影响可以根据ble相移显著降低。也就是说，通过巧妙地模拟芯片序列“01”到“11”或“00”（与“10”相同），我们能够最大化dsss将接收到的芯片序列映射到正确符号并输出正确比特的概率。我们将在下一节详细讨论这个问题。

作为概念验证示例，我们模拟了来自ble的32芯片zigbee符号“0”（即表2中的“0000”）。在图7a中，对zigbee和ble的时域i/q信号进行了比较，它们由于不同的脉冲形状而非常不同，即ble的高斯脉冲和zigbee的半正弦脉冲。如前所述，图7b的上部所示的相移证明，对于ble，每0.5μs的位移为±π/4，其中对于zigbee为±π/2。此外，在zigbee的相移不一致的地方观察到一些错误，这也反映在芯片中（在图7b中较低的位置），我们在模拟dsss时考虑这些芯片，以便最小化解码比特中的错误。下一节将对此进行详细说明。



纳尔

图7:BLE模拟信号和所需Zigbee信号之间的比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 传说 |  | 仿真符号 |  | 理想符号 |

！

                   0001           0000           **0010** 0011

0101 **0100** 0110 0111 1001  1000 1010 1011

                   1101           1100  1111

图8：优化仿真示例

## 4.4最优DSSS仿真

在本节中，我们将讨论Bluebee如何通过DSSS将OQPSK仿真中不可避免的芯片错误带来的影响最小化。首先，让我们看一个简化的遍历示例：图8示出了4位hamming空间中的仿真（从zigbee dsss中的32简化）。在这个hamming空间中，有三个理想符号，需要使用第4.3节中介绍的方法进行仿真。由于blue的能力有限，bluebe只能生成有限数量的仿真符号，这些符号在图中用虚线矩形标记。这个汉明空间中的其他符号不能用蓝白来表示。让sdeenote the iideal symbol，eto表示iedeal symbol。然后，我们定义了两个符号（hamming）距离，如下所示：*我钍我钍*

*定义4.1。*符号内距离dist（e，s）是从仿真符号到理想符号s的汉明距离。*我我我我*

*定义4.2。*符号间距离dist（e，s）是从仿真符号到理想符号s的汉明距离，其中j，i。*我J我J*

以图8为例。要模拟理想符号“1110”，

Bluebee可以生成两个可仿真的符号“1100”和“1111”，它们具有相同的符号内距离1。在此之后，Bluebee考虑从这些仿真符号到其他两个理想符号的符号间距离。对于仿真符号“1100”，它与理想符号“0100”和“0010”的符号间距离分别为1和3。类似地，对于仿真符号“1111”，其符号间距离分别为3和3。因此，Bluebee选择“1111”作为模拟选项，因为它具有最小符号间距离的最大值（即最大边距）。

前面的例子说明了在4位hamming空间中dsss仿真的思想。现在我们将讨论Bluebee如何在标准Zigbee符号空间中优化DSSS仿真，遵循相同的原则。

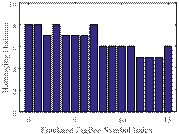


图9：模拟和理想Zigbee符号之间的符号内汉明距离

符号内距离。每个4位zigbee符号映射到32个芯片。将32个芯片分为16个连续的芯片对，并计数“01”或“10”，得到zigbee仿真中的芯片错误数乘以ble或等效的dist（e，s）（即，符号内汉明距离）。对于给定符号，此值是常量，因为模拟“01”或“10”始终会导致1个芯片错误，而不管模拟为“00”或“11”。例如，在图9中，我们绘制了所有可能的zigbee符号的内hamming距离。我们发现最大的内汉明距离是8，例如zigbee符号“0000”的内汉明距离。注意，符号内汉明距离不能被优化，因为在模拟不一致的zigbee相移时，无论选择哪个比特，总是会有一个芯片错误。*我我*

符号间距离。虽然每个符号的符号内距离是固定的，但Bluebee试图增加符号间距离以提高可靠性。这是因为符号间hamming距离dist（e，s），i，j，取决于如何模拟“01”或“10”。例如，“01”可以通过“00”或“11”模拟。因此，zigbee符号可以在2个不同的序列中模拟，其中bluebe选择具有最大最小符号间hamming距离的模拟符号。这种优化可以用下面的方程式来描述：*我J迪特*（e）*我*的S*我*）

                        argmax{距离（e，s），i，j}（1）*e最小值我J*

我们注意到，在有限的搜索空间为0≤i，j≤15的情况下，计算量较小。此外，这只需要计算一次，因此可以在运行Bluebee之前预先计算并加载到设备上。

## 4.5处理BLE数据白化

**数据**



**白化数据**

**（符号）**

### 图10：通过LFSR的BLE数据白化

出于安全考虑，ble发送的符号不是有效载荷的纯消息。相反，在ble有效载荷上采用了一种称为数据白化的置乱技术，以随机化有效载荷字节和空中传输字节之间的匹配。因此，克服ble上的数据白化问题是通过ble有效载荷控制发送信号的关键。

事实上，最近的文献已经表明ble的lfsr电路是可逆的[19，35]；技术上，ble使用7位线性反馈移位寄存器（lfsr）电路和多项式x+x+1，如图10所示。该电路用于产生一个位序列，通过异或运算使输入数据变白。lfsr电路的初始状态是在ble规范中定义的二进制表示中的当前信道号（即从0到39）。7 4

Bluebee逆向工程的白化过程，根据精心选择的字节生成BLE负载进行仿真。

这使得Bluebee完全兼容可商品化的设备，并在SEC的商品化设备上通过广泛的测试平台实现和评估进行了验证。8。

# 5并发通信

ble的一个特点是跳频，这有助于ble设备避免其他ism波段无线电占用的繁忙信道。在Bluebee中，此功能允许一个BLE设备在2.4GHz频带之间跳跃，并在不同信道上与多个Zigbee设备通信。此外，我们可以控制ble跳频序列，同时仍然遵循ble跳频协议。在这一节中，我们将首先简要介绍ble跳频协议，然后设计两个blueee信道调度方案。

## 5.1 BLE跳频

BLE有40个2兆赫宽的通道，标记为通道0到通道39。其中37、38、39频道为广告频道，其余为数据频道。一旦在广告频道上建立连接，两个成对的设备将在数据频道之间跳跃。

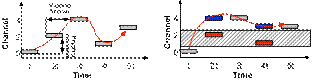


图11:BLE正常频率-图12:BLE自适应跳频

在ble中，使用一个简单而有效的跳频协议来确定下一个要跳频的信道。第一个信道总是“0”，并且在跳变间隔的一段时间之后，ble设备将随着跳变增量的增加跳到下一个信道。

公式中

*丙下一个*=c+hoppinüinc（mod37），（2）其中分别代表下一个和当前信道，37是ble数据信道的总数，hoppinginc是跳变增量。在图11中，我们示出了5个信道（即信道“0”到信道“4”）上的跳频序列，跳频增量为2，跳频间隔为t。*现在的下一个现在的*

为了避免与同一ISM频段上的其他无线电台发生冲突，当某些信道上的包接收比较低时，BLE采用自适应跳频（AFH）。在ble afh中，37位信道映射用于保持信道链路质量，其中“0”表示坏信道，“1”表示好信道。让我们用沙子分别表示好的和坏的通道集。当下一个通道是坏通道时，它将被S中的另一个通道替换。更具体地说，将通过*奥德坏的奥德*

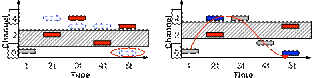
*重新索引*=C S，（3）和C将替换为S（remapindex）。例如，在图12中，信道1和信道2是坏信道。因此，每当ble设备跳到这两个信道时，它们将在eq之后分别被重新映射到信道3和信道4。三。*下一个模型奥德下一个奥德*

## 5.2蓝莓频道调度

使用afh，访问不同频道的频率变得不均匀。例如，在图12中，在信道3和4的一半频率的一个跳变周期（即，在示例中为5跳）中，信道0将仅被访问一次。在真实的网络环境中，afh会在ble-zigbee重叠信道（即2410、2420，…）上对zigbee节点造成不公平的服务。2480MHz）。也就是说，Bluebee的QoS无法得到保证。为了解决这个问题，我们希望以无中断的方式平衡ble访问重叠频道的频率。

为了实现这一点，我们可以利用ble中的37位信道位图。如前所述，如果启用afh，则使用信道位映射来计算下一个要跳的信道。另外，现有的ble协议支持在正常传输过程中对信道位图进行更新，以适应快速变化的网络环境。因此，我们可以通过更新信道比特图来控制ble的跳变行为。针对应用场景中不同的优化目标，我们提出了两个并发的bluebe解决方案。

最大吞吐量解决方案。通过更新信道位图，我们可以控制ble设备可以跳到的信道集。为了最大限度地提高并发bluebee的吞吐量，如果zigbee重叠信道在原始信道比特图（即，s）中标记为空闲，则可以将它们留在信道比特图中，同时将非重叠信道列入黑名单。在连接初始化阶段，通道位图只需设置一次，因此网络开销非常低。注意，我们所做的只是从空闲信道中选择一个子信道，因此我们不会中断ble信道跳跃的原始功能，这是为了避免信道冲突。此外，ble标准通过主机（即用户）级命令（如hci\_set\_afh\_channel\_classif ication[34]）支持这种更改。负载平衡解决方案。在某些情况下，调度集中的公平性更为重要，如多信道同步问题。最大吞吐量设计可能无法保证在不同信道上的负载平衡CTC。本文提出了一种简单而有效的启发式方法来平衡ble在重叠信道上的跳频，同时仍然遵循ble的afh协议。更具体地说，我们可以通过稍微修改通道位图中的沙子来平衡蓝蜂流量。其基本思想是，对于每个不均衡信道C（即，访问小于其他重叠信道），我们发现另一个信道CIN，其重新映射的信道将是C。然后，我们将CAS标记为信道比特映射中的坏信道，以便每当BLE设备跳到C时，信道将被重新映射到C。当然，我们需要保证S不变，所以THA。t剩余索引不变。为此，我们选择在通道位映射中将一个坏通道标记为好，这样s仍然保持不变。*奥德奥德坏的*’*奥德*’’*奥德奥德*



（a）在s中选择一个频道（b）在sto中添加一个频道*面向对象坏的*

其重新映射的频道将是目标频道*面向对象*

### 图13:Bluebee频道调度的步骤

在图13a和图13b中，我们说明了我们的调度算法。在本例中，我们尝试重新平衡通道0和通道4。我们发现0频道的访问量小于4频道，因此我们希望将跳频重定向到0频道。我们首先假设所有通道都需要重新映射（标记为红色），除了通道1。然后我们找到其重新映射的信道将是信道0，即信道3，如图13a所示。我们添加通道3来替换s中的一个通道，即通道1，这样s不会如图13b所示发生变化。最后我们重新平衡了通道0和通道4。不可否认，它是一种尽力而为的调度方法，因为有时由于坏信道太多而无法平衡所有重叠信道。在这种情况下，我们不会破坏很多好的渠道来实现再平衡的目标。*坏的坏的面向对象*

# 6链路层保护

在这一部分中，我们介绍了blubee的链路层保护方法，即多个前置码、链路层编码和基于ble链路统计的自适应保护。

## 6.1帧重传

为了提高传输的可靠性，bluebee可以多次发送相同的蓝牙数据包来模拟zigbee数据包，以防在接收端丢弃一些模拟的zigbee数据包。如果正确接收到同一zigbee分组的至少一个副本，即该分组按照802.15.4标准[18]的规定通过crc校验和，则zigbee接收器能够接收正确的信息。帧重传技术在接收端与zigbee协议自然兼容。这是因为，如果zigbee接收器已经根据802.15.4标准接收到重传的zigbee帧，它将自动忽略重传的zigbee帧。

帧重传的次数与帧接收比（frr）有关。假设每个模拟的zigbee帧的接收独立于其他帧，在发送k个副本之后，我们将成功地接收概率为1-（1-frr）的至少一个zigbee帧。实验表明，在不同的信噪比情况下，Bluebee包的成功接收是不同的。假设我们有70%的frr，那么经过6次重传，最终的成功接收率超过99.9%，这表明bluebe可以通过简单地重传模拟帧来获得很高的frr。请注意，重新传输不会对信道效率造成显著的开销，因为CTC通常用于控制目的，而总流量需求很少。）*K*

## 6.2重复序言

除了上述帧重传技术外，bluebe还利用重复前导码技术进一步提高可靠性。在商品zigbee芯片中，可能的zigbee分组的解调首先通过搜索由八个“0”符号组成的特定前导码开始，接着是符号“a7”，这是开始帧定界符（sfd）。由于该前导码检测是在zigbee可以接收任何帧之前完成的，因此不能使用上层编码对其进行保护。为了提高分组接收速率，bluebe发送多个重复的前码，如图14所示。如果成功地接收到第一个前导码，则zigbee接收器将丢弃上层解码中的剩余前导码。否则，zigbee还有第二次机会检测到前导码。

**普通Zigbee包**

|  |  |  |
| --- | --- | --- |
|  |  | |
| **0** | **0** | **Zigbee有效载荷** |

                           ““啊！”！“”$！”“啊！”“啊”

**重复正常**

**序言**

图14：重复序言的可靠反恐委员会

# 7讨论

## 7.1与Bluetooth Classic的兼容性

BLE is defined in Bluetooth core specification 4.0 [34]. Another well known Bluetooth technique is Bluetooth Classic, defined in Bluetooth core specification 1.0. There are some connection and distinctive difference between these two techniques. First, in modulation, although both adopts GFSK, Bluetooth Classic&apos;s modulation index is 0.35 while BLE&apos;s modulation index is between 0.45 and 0.55. The difference in modulation index affects the shape of the final signal. As mentioned earlier, the phase shift error brought by pulse shape can be mediated through phase shift quantization at ZigBee receiver, which means BlueBee can still be used in Bluetooth Classic. Second, Bluetooth Classic has 79 channels distributed from 2402MHz to 2480MHz spaced 1MHz apart. So it can cover all ZigBee channels. Third, on the frequency hopping, Bluetooth Classic will hop among all 79 channels following a frequency hopping pattern calculated through master device&apos;s MAC address and clock. Its hopping interval will always be 625µs. The hopping interval is long enough to transmit a Bluetooth emulated ZigBee packets. Although the channel scheduling methods will be different, the same heuristic method can be used to find a channel scheduling solution.

## 7.2 Feasibility of Reverse Communication

Although in this paper, we focus on the communication from BLE to ZigBee, the reverse communication (e.g., CTC from ZigBee to BLE) might be needed to provide the feedback (e.g., ACKs for BLE to ZigBee packets) from ZigBee. The reverse communication from ZigBee to BLE is also possible through the phase shift emulation.

More specifically, due to the similarity in (de)modulation, a BLE receiver can get the information about the phase shifts of a ZigBee symbol in the air, but only in coarse grain (i.e., 1Mbps BLE data rate compared to the 2Mbps ZigBee chips) restricted to its limited bandwidth. However, a BLE receiver is still able to derive the corresponding ZigBee symbols from the detected phase shift information because ZigBee chips are redundant. We will make the communication from ZigBee to BLE and its compatibility with commodity devices our future work.

# 8 EVALUATION

In this section, we evaluate the performance of BlueBee across various domains, such as CTC performance comparison, communication reliability, support in mobility and low-duty cycle, and the example application of coexistence between ZigBee and BLE.

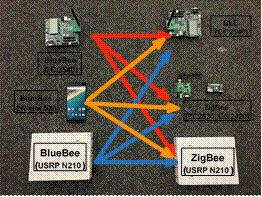


Figure 15: Experiment Setting for BlueBee

## 8.1 Platform Setting

Fig. 15 demonstrates the evaluation platform of BlueBee. We have implemented BlueBee as a sender on (i) a GNU radio BLE implementation called scapy radio [3] with a USRP-N210 platform, (ii) a commodity BLE CC2540 development kit [2], and (iii) a commodity smartphone Nexus 5X. Note that, we use USRP here only for its convenience to change parameters in the experiments. Our design is compatible with the widely used BLE 4.0 chips, such as CC2540, as well as smartphones with the latest BLE 4.2 protocol, such as Nexus 5X, which is back compatible to BLE 4.0 and supports the long BLE frame up to 257 bytes.

As for the receiver side, we have tested the BlueBee on the following platforms: 1) A commodity BLE receiver (i.e., CC2540 development kit); 2) Commodity ZigBee receivers including CC2530 and CC2420 (i.e., MICAz and TelosB); and 3) 802.15.4 implementation on USRP N210 to provide detailed examination of the PHY level emulation performance. The arrows from BlueBee to three receivers indicate that a broadcast frame from BlueBee (either USRP or commodity devices) can be decoded by both commodity ZigBee receivers and commodity BLE receivers simultaneously, indicating the emulated frames are both BLE-compliant and ZigBee-compliant.

## 8.2 CTC Throughput

To evaluate the CTC throughput of BlueBee, we compare its throughput with the state-of-the-art packet level CTC methods.

*8.2.1 Compare with FreeBee.* The only state-of-the-art CTC work on BLE to ZigBee communication is FreeBee [22]. FreeBee&apos;s throughput is 17bps with a single CTC transmitter, while the throughput of BlueBee is 225kbps, 13, 000× the throughput of FreeBee. Admittedly, FreeBee has its unique advantage of a free channel design, which differentiates it from those CTC designs that saturate the channel for high throughput. BlueBee can also beat existing packet-level CTC technologies that can saturate the channel for high throughput.

*8.2.2 Compare with Other Packet-Level CTC.* Here we compare BlueBee with other state-of-the-art packet-level CTC technologies, including Esense ( WiFi → ZigBee), and B(BLE → WiFi) in throughput. Note that, these CTC techniques have a highbandwidth radio (i.e., 20MHz WiFi radio) either at the sender or at the receiver. From Fig. 16, we can see that BlueBee can surpass the state-of-the-art packet-level CTC by 70 × −100×. It indicates the intrinsic advantage of PHY-layer CTC over packet-level CTC.2*W*2

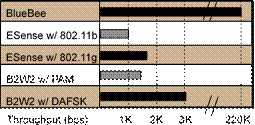
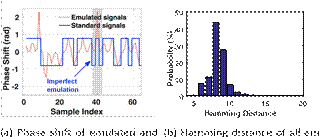


Figure 16: Comparison with the state of the arts

## 8.3 Emulation Reliability

Here we evaluate the emulation reliability of BlueBee, including PHY-layer reliability (i.e., phase shift and hamming distance) and link-layer reliability (i.e., frame reception ratio). To provide the details, we test these experiments under various situations, including different transmission power, distances, scenarios, and different packet duration.



standard symbols ulated symbols

### Figure 17: Performance of phase shift emulation

*8.3.1 Emulated Signals.* Since BlueBee&apos;s BLE sender emulates the phase shifts in legitimate ZigBee frames, we first examine the performance of signal emulation.

Recall that in the Section IV, ZigBee&apos;s OQPSK demodulation is based on the phase shifts, whose positive and negative sign will be further decoded as BLE symbol &apos;1&apos; and &apos;0&apos;. In Fig. 17a, we plot the phase shift of received ZigBee symbol and an ideal ZigBee symbol. We find that BLE can emulate consistent phase shifts (i.e., slowly changing phase shifts) while failing to emulate inconsistent phase shifts (i.e., fast changing phase shifts) due to its limited bandwidth. Note that the 64 samples for a ZigBee symbol is due to the oversampling of commodity ZigBee devices. The 64 samples will then be decimated to 32 chips for decoding. In Fig. 17b, the distribution of the Hamming distances of decoded ZigBee symbols is plotted. We find that most Hamming distances are in the range of [6, 10] especially in [8, 9], showing that the number of error chips caused by inconsistent phase shifts is small and within the tolerance of ZigBee.

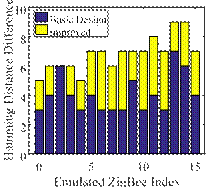
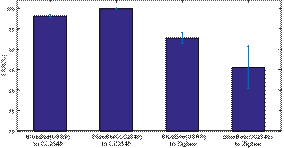
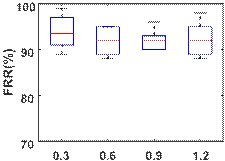


Figure 18: Hamming distance improvement of DSSS emulation

Since the ZigBee&apos;s OQPSK demodulation needs to consider the closest hamming distance, the inter-symbol hamming distance also affects the accuracy of emulation. In Fig. 18, we illustrate the performance gain when BlueBee considers the intra symbol hamming distance. For example, after the optimization, the hamming distance improvement is in Figure 18. In the basic design, the hamming distance difference ranges from 3 to 7, while the hamming distance difference of 3 suggests very little protection from the background noises. With the optimization of BlueBee, we manage to increase this hamming distance difference for the emulated ZigBee symbols, as shown in Figure 18. This means that BlueBee can tolerate more background noises than the basic design, leading to a better reliability.



### Figure 19: FRR comparison under BLE and ZigBee

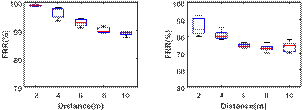


**Duration of Frame(ms)**

### Figure 21: FRR with different frame duration

*8.3.2 Dual-standard Compliance.* In BlueBee, a legitimate ZigBee packet is embedded in a legitimate BLE frame. To verify and evaluate such embedding, we have implemented BlueBee on various hardwares, including 1) software defined radio, i.e., USRP N210 and 2) commodity BLE devices, i.e., CC2540 development kit. At the receiver side, we use both commodity BLE receiver (i.e., CC2540) and commodity ZigBee receiver (i.e., MICAz). As shown in Fig. 19, BlueBee, either the USRP implementation or commodity device implementation, can achieve over 99% frame reception ratio (FRR) at commodity BLE receiver, showing that it is a BLE compliant design. In addition, BlueBee&apos;s USRP and commodity device implementations can achieve an over 90% FRR and an over 85% FRR at commodity a ZigBee receiver, showing that it is also a ZigBee compliant design.

The characteristic of dual-standard compliance indicates BlueBee can achieve cross-technology broadcast. That means we can construct a dual-standard frame where part of it is a ZigBee frame and part of it is a BLE frame. Each technology can identify their parts by detecting legitimate preamble and header while regarding the rest as noise.



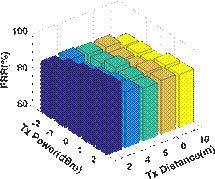
(a) FRR with distance on USRP (b) FRR with distance on com-

(lab) modity devices (lab)

### Figure 20: FRR with distance

*8.3.3 Impact of Distance.* We also evaluate the frame reception ratio (FRR) where the BLE sender sends out emulated ZigBee frames on both USRP and commodity CC2540 development kit. Fig. 20a depicts the FRR when USRP N210 emulates the ZigBee frames with the transmission power of 0dBm, the maximum energy level allowed in BLE standard [34]. In all the experiments, the average FRR is within 92% to 86%, demonstrating the reliability of BlueBee, at a transmission distance of 10m (the usual communication range between two BLE devices) Note that the FRR slightly decreases with the increasing of distance, due to the lower SNR. Even so, in all the experiments, the FRRs are all above 85%. The experiments on commodity CC2540 development kit have similar trend. During these experiments, the FRR is above 73% for all the different transmission distances.

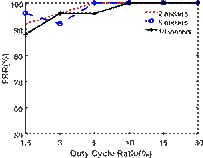
*8.3.4 Impact of Frame Duration.* In BLE specification 4.2 [34], the maximum payload for BLE has been extended from 39bytes to 257bytes, which means the frame duration will grow from around 0.3ms to over 2ms. So we here study the impact of frame duration on BlueBee&apos;s performance. In Fig. 21, we study the FRR with frame duration ranging from 0.3ms to 1.2ms, following the latest standard. We find that the increase in frame duration will slightly decreases FRR, about 2% decrease. That is because a longer frame is usually more vulnerable to environment noise and interference [33]. Even so, over 90% FRR shows BlueBee&apos;s resistance to the impact of long frame.



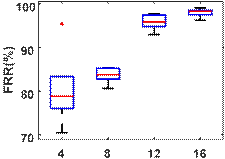
### Figure 22: FRR with Tx power and distances

*8.3.5 Impact of Tx Power and Tx Distance.* In Fig. 22 we study the frame reception ratio (FRR) of BlueBee with impact of various Tx power and distance from a USRP to a commodity CC2530 ZigBee device for its convenience to control transmission power. We find that when Tx power increases from −2dBm to 2dBm , most FRR also increases from 85% to 90%. Then we fix the Tx power, and study the FRR of BlueBee with different distances. We find that when the distance is as far as 10m , the FRR is still over 80%. Note that the transmission power of a typical BLE device is 0dBm and the typical transmission range is 10m. That means BlueBee can work well with typical BLE setting.

*8.3.6 Protection in the link layer-multiple header.* In Fig. 23 we study the performance of our link layer protection by repeated preambles. Typical preamble length in ZigBee is 8 ZigBee symbols &apos;0&apos;.The number of &apos;0&apos;s can be changed with at least four &apos;0&apos;s. We change the length of ZigBee preamble from 4 symbols to 16 symbols which doubles the length of preamble. We can see from the figure, with a typical preamble length of 8 symbols, FRR is about 84%, When we increase the preamble length to 12 symbols, the FRR jumps to about 95%, a 13% improvement. The experiments prove the effectiveness of our multiple preamble technique. Even when we reduce the preamble length to 4 symbols, we find that the average FRR is still about 78%, which shows the robustness of BlueBee.



### Figure 25: BluBee&apos;s support for low duty cycle network

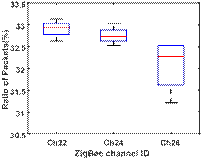


**Preamble Length(symbol)**

Figure 23: FRR with different preamble length out door

## 8.4 BlueBee Channel Scheduling

In this section, we evaluate the performance of the BlueBee scheduling algorithm to evenly distribute the BlueBee emulation frames. Three TelosB nodes are set to ZigBee channel 22, 24, and 26, which have the same central frequencies with BLE channel 27, 32, and 39 respectively. The BLE sender is implemented on the USRP N210platform, with a total number of 999 emulation frames. To test the performance, BlueBee adopts out traffic adaptive algorithm to evenly distribute the CTC traffic among ZigBee channels, i.e. 333 frames at each ZigBee channel (i.e., 33% of all packets).



### Figure 24: Concurrent CTC on three ZigBee channels

Fig. 24 depicts the number of successful receptions at each ZigBee channel. It is obvious that these ZigBee nodes working at different ZigBee channels are able to receive the similar number of frames (only 1% difference), demonstrating the efficiency of the traffic adaption method based on the existing BLE channel bitmap. Note that the ratio of received packet will be slightly lower in ZigBee channel 26 because it overlaps with BLE advertising channel 39, which is very busy.

## 8.5 Low Duty Cycle Support

In this section, we study the BLE&apos;s support to the low duty cycle network due to the fact that ZigBee devices are usually work on low duty cycle mode to save energy. Low duty cycle scenario becomes even critical due to the fact that the BLE transmitter will do frequency hopping. In the experiment, we transmit BLE frame from USRP to MICAz, a commodity ZigBee device. The BLE transmitter&apos;s hopping interval is set to be 10ms, within the range of available hopping interval in the standard. From Sec. 5.1 we know that BLE will always return to the start channel after 37 hops, which means the transmission interval of BLE to a ZigBee node at a specific channel will be 370ms. To make successful CTC from BLE to ZigBee in low duty cycle mode, BLE will retransmit each frames 20 times. As shown in Fig. 25, FRR increases when ZigBee&apos;s duty cycle becomes larger. When the duty cycle is larger than 10%, 100% FRR is reached. However, even when BLE&apos;s duty cycle is only 1.5%, a FRR of at least 88% still can be reached. This experiment indicate that BlueBee has the potential to be used in a low duty cycle as a long lasting coordinator.

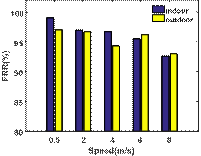


Figure 26: BluBee&apos;s support for mobile scenario

## 8.6 Mobility Support

In this part, we study the impact of mobility on the performance of BlueBee because BLE radios are widely used in mobile scenario such as in smart wristbands. In the experiment, a USRP with BlueBee sender is put on a table broadcasting emulated ZigBee frames on a fixed channel. A person carrying a commodity ZigBee node, i.e., MICAz node, is walking, jogging, and running with different speed at about 10m away. As shown in Fig. 26, there is only a slightly decrease in FRR when the speed increases. Even when the person is running at speed 8m/s, we can still achieve about 90% FRR. Both indoor and outdoor environment show similar results.

## 8.7 Application

|  |
| --- |
| direct control of ZigBee devices from a BLE radio without a ZigBee-  BLE gateway [16] and any hardware modification at either side. BlueBee can be easily generalized to other IoT control scenario. It is a key enabler for other IoT cross-technology control design under commodity ZigBee and BLE devices.  9 RELATED WORK |

Application I: Channel Coordination In this section, we demonstrate one possible application built upon BlueBee, i.e. the channel coordination between incompatible ZigBee and BLE. Note that BlueBee enables many possible benefits as stated in Section II, and Figure 27: Channel coordination between ZigBee and BLE

we only introduce its channel coordination due to the limitation of space as shown in the left part of Fig. 27.

In this experiment, the two ZigBee TelosB devices are communicating on ZigBee channel 26, to avoid other possible ISM band interference. One BLE sender is transmitting its advertising frames on frameBLE channel 39, which overlaps with the ZigBee channel 26. Since BLE does not perform CSMA before transmission, the BLE frames might corrupt the ZigBee frames when they collide into each other.

To evaluate the performance, we conduct experiments on different coordination methods, such as no CSMA, with CSMA, and our channel scheduling method. In our channel scheduling, when the BLE wants to transmit the BLE frames, it first broadcasts the scheduling frame using BlueBee, which contains the future channel usage of BLE. After successfully receiving these frames, the ZigBee transmitter will coordinate the timing of the transmitted frames accordingly.

In the right part of Fig. 27 shows the experimental results. Compared with no CSMA, and with CSMA, BlueBee successfully improves the frame reception ratio to 98%, clearly demonstrating the channel efficiency of BlueBee&apos;s coordination. This implies that effective radio coordination can be achieved through CTC, which opens a door for cross-technology MAC design in the future.

**Smart Bulb**

**(ZigBee)**

**Smartphone**

**(BlueBee)**

### Figure 28: BlueBee smart light bulb control

Application II: Smart Light Control BlueBee can be easily deployed on commodity smartphones with BLE support, e.g., Nexus 5X smartphone, and benefit the smart home devices in real life. In Fig. 28, we implement BlueBee on a Nexus 5X smartphone to control ZigBee light bulbs at one of the overlapped channels, i.e., 2.48GHz. Available commands including the on/off status, the color, the intensity, and which light bulb to control. BlueBee achieves With the rapid development of various wireless technologies, the

ISM band suffers from significant cross-technology interference

(CTI) [4, 5, 8–10, 15, 20, 25, 27, 30]. To alleviate this this, there has been numerous researches on alleviating the CTI by detecting and avoiding the interference, or recovering the corrupted signal from the interference[31–33, 36, 38, 39, 41, 42, 44–46]. However, this line of methods force the receivers to adapt to the interference pattern, resulting in the unfairness between various technologies.

To address this issue, researchers propose cross-technology communication (CTC) which directly builds the communication between heterogeneous devices [7, 11, 22, 42, 43]. The core idea of these CTC methods is that the sender creates special energy patterns by sending out legacy packets, while the receivers detect these patterns by either the received signal strength (RSS) sampling, or the channel state information (CSI), which are supported by the existing hardware. However, the existing CTC technologies commonly use coarse packet-level information, thus suffer from the significant low throughput and long transmission delay.

In contrast to these packet-level CTC methods, BlueBee is the first approach to achieve the PHY-level CTC to the best of our knowledge. The core idea of signal emulation in the BlueBee is inspired by several recent works studying the signal manipulation[6,

19, 21, 23, 28]. In addition, in the LTE system, Ultran [6] emulates the WiFi packets via a LTE transmitter to coordinate between LTE and WiFi, while it requires the modification of existing LTE standard. Different from these approaches, BlueBee does not require andy modification to existing hardware, and is fully compatible with existing commodify Bluetooth and ZigBee hardware.

In summary, BlueBee is the first PHY-level CTC which does not require any hardware modification. It is fully compatible with existing Bluetooth and ZigBee hardware, and achieves high CTC throughput with little transmission delay.

# 10 CONCLUSION

In this work we present BlueBee, a new PHY layer cross-technology communication technique proposing a direction of emulating legitimate ZigBee frames using BLE radio. BlueBee paves the road to practical CTC by offering over 10, 000× the throughput compared to the state-of-the-art CTC designs that rely on coarse-grained packetlevel information. The emulation is achieved simply by selecting the payload bytes of BLE frames to provide unique dual-standard compliance and transparency where neither hardware nor firmware changes are required at the BLE senders and ZigBee receivers. BlueBee includes advanced features such as multi-channel concurrent CTC via adaptive frequency hopping in BLE operation. Comprehensive testbed evaluation on both USRP and commodity ZigBee/BLE devices show that BlueBee achieves 99% accuracy, while providing reliability under mobile and duty cycled scenarios.

# ACKNOWLEDGMENTS

This work was supported in part by the NSF CNS-1444021, NSF CNS-1718456, NSF CNS-1717059, and NSF China 61672196. We sincerely thank our shepherd Abusayeed Saifullah and anonymous reviewers for their valuable comments and feedback.

# REFERENCES

[1] 1999. IEEE 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification (1999).

[2] 2013. TI cc2540 development kit. http://www.ti.com/tool/cc2540dk/. (2013).

[3] 2016. Scapy radio. https://github.com/BastilleResearch/scapy-radio. (2016).

[4] Fadel Adib, Swarun Kumar, Omid Aryan, Shyamnath Gollakota, and Dina Katabi. 2013. Interference alignment by motion. In MobiCom &apos;13. ACM, 279–290.

[5] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. 2009. White space networking with wi-fi like connectivity. ACM SIGCOMM Computer Communication Review 39, 4 (2009), 27–38.

[6] Eugene Chai, Karthik Sundaresan, Mohammad A Khojastepour, and Sampath Rangarajan. 2016. LTE in unlicensed spectrum: are we there yet?. In MobiCom &apos;16. ACM, 135–148.

[7] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: communication through energy sensing. In Proceedings of the 15th annual international conference on Mobile computing and networking. ACM, 85–96.

[8] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. 2015. Airexpress: Enabling seamless in-band wireless multi-hop transmission. In MobiCom &apos;15. ACM, 566–577.

[9] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. Interference alignment using shadow channel. In INFOCOM 2015. IEEE, 2128–2136.

[10] Lin Chen, Ruolin Fan, Kaigui Bian, Mario Gerla, Tao Wang, and Xiaoming Li. 2015. On heterogeneous neighbor discovery in wireless sensor networks. In INFOCOM &apos;15. IEEE, 693–701.

[11] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. 2016. B2W2: N-Way Concurrent Communication for IoT Devices. In Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM. ACM, 245–258.

[12] Inc Gartner. 2016. Gartner Report. Available at urlhttp://cloudtimes.org/2013/12/20/gartner-theinternet-of-things-willgrow-30-times-to-26-billion-by-2020/.

[13] Minkeun Ha, Seong Hoon Kim, Hyungseok Kim, Kiwoong Kwon, Nam Giang, and Daeyoung Kim. 2012. SNAIL gateway: Dual-mode wireless access points for WiFi and IP-based wireless sensor networks in the internet of things. In 2012 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, January 14-17, 2012.

[14] Minkeun Ha, Kiwoong Kwon, Daeyoung Kim, and Peng-Yong Kong. 2014. Dynamic and Distributed Load Balancing Scheme in Multi-gateway Based 6LoWPAN. In 2014 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, and IEEE Cyber, Physical and Social Computing, iThings/GreenCom/CPSCom 2014, Taipei, Taiwan, September 1-3, 2014.

[15] Tian Hao, Ruogu Zhou, Guoliang Xing, Matt W Mutka, and Jiming Chen. 2014. Wizsync: Exploiting wi-fi infrastructure for clock synchronization in wireless sensor networks. IEEE Transactions on mobile computing 13, 6 (2014), 1379–1392.

[16] Mayur Hawelikar and Sunil Tamhankar. 2015. A design of Linux based ZigBee and Bluetooth low energy wireless gateway for remote parameter monitoring. In Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on. IEEE, 1–4.

[17] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. 2010. Beyond coexistence: Exploiting WiFi white space for ZigBee performance assurance. In Network Protocols (ICNP), 2010 18th IEEE International Conference on. 305–314.

[18] Ieee802.org. 2012. IEEE 802.15.4. (2012).

[19] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. 2016. Inter-technology backscatter: Towards internet connectivity for implanted devices. In Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference. ACM, 356–369.

[20] Tao Jin, Guevara Noubir, and Bo Sheng. 2011. WiZi-Cloud: Applicationtransparent dual ZigBee-WiFi radios for low power internet access. In INFOCOM. 1593–1601.

[21] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R Smith. 2016. Passive wi-fi: Bringing low power to wi-fi transmissions. In NSDI &apos;16. USENIX Association, 151–164.

[22] Song Min Kim and Tian He. 2015. FreeBee: Crosstechnology Communication via Free Sidechannel. In MOBICOM, 2013 Proceedings ACM.

[23] Zhenjiang Li, Yaxiong Xie, Mo Li, and Kyle Jamieson. 2015. Recitation: Rehearsing wireless packet reception in software. In MobiCom &apos;15. ACM, 291–303.

[24] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. 2010. Surviving Wi-fi Interference in Low Power ZigBee Networks. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys &apos;10).

[25] Rajesh Mahindra, Hari Viswanathan, Karthik Sundaresan, Mustafa Y Arslan, and Sampath Rangarajan. 2014. A practical traffic management system for integrated LTE-WiFi networks. In Proceedings of the 20th annual international conference on Mobile computing and networking. ACM, 189–200.

[26] Stefan Nastic, Hong Linh Truong, and Schahram Dustdar. 2015. SDG-Pro: a programming framework for software-defined IoT cloud gateways. J. Internet Services and Applications 6, 1 (2015), 21:1–21:17.

[27] Georgios Nikolaidis, Mark Handley, Kyle Jamieson, and Brad Karp. 2015. COPA: cooperative power allocation for interfering wireless networks. In Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies. ACM, 18.

[28] Jiajue Ou, Yuanqing Zheng, and Mo Li. 2014. MISC: Merging incorrect symbols using constellation diversity for 802.11 retransmission. In INFOCOM 2014. IEEE, 2472–2480.

[29] Soheil Qanbari, Negar Behinaein, Rabee Rahimzadeh, and Schahram Dustdar.

2015. Gatica: Linked Sensed Data Enrichment and Analytics Middleware for IoT Gateways. In 3rd International Conference on Future Internet of Things and Cloud, FiCloud 2015, Rome, Italy, August 24-26, 2015. 38–43.

[30] Saravana Rathinakumar, Bozidar Radunovic, and Mahesh K Marina. 2016. CPRecycle: Recycling Cyclic Prefix for Versatile Interference Mitigation in OFDM based Wireless Systems. In Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies. ACM, 67–81.

[31] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. 2016. SNOW: Sensor network over white spaces. In Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys).

[32] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. 2011. No time to countdown: Migrating backoff to the frequency domain. In MobiCom &apos;11. ACM, 241–252.

[33] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. 2013. Successive interference cancellation: Carving out MAC layer opportunities. IEEE Transactions on Mobile Computing 12, 2 (2013), 346–357.

[34] Bluetooth specification. 2011. Bluetooth technology website. (2011). http: //www.bluetooth.com/

[35] Dominic Spill and Andrea Bittau. 2007. BlueSniff: Eve Meets Alice and Bluetooth. WOOT 7 (2007), 1–10.

[36] Karthikeyan Sundaresan, Srikanth V Krishnamurthy, Xinyu Zhang, Amir Khojastepour, Sampath Rangarajan, et al. 2015. TRINITY: A Practical Transmitter Cooperation Framework to Handle Heterogeneous User Profiles in Wireless Networks. In MobiHoc &apos;15. ACM, 297–306.

[37] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. [n. d.]. C-Morse: Crosstechnology Communication with Transparent Morse Coding. In Proceedings of INFOCOM 2017.

[38] Sangki Yun and Lili Qiu. 2015. Supporting WiFi and LTE co-existence. In Computer Communications (INFOCOM), 2015 IEEE Conference on. IEEE, 810–818.

[39] Xinyu Zhang and Kang G Shin. 2011. Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi. In MobiHoc &apos;11. ACM, 6.

[40] Xinyu Zhang and Kang G. Shin. 2013. Cooperative Carrier Signaling: Harmonizing Coexisting WPAN and WLAN Devices. IEEE/ACM Trans. Netw. 21, 2 (April

2013).

[41] Xinyu Zhang and Kang G Shin. 2013. Cooperative carrier signaling: Harmonizing coexisting WPAN and WLAN devices. Networking, IEEE/ACM Transactions on 21, 2 (2013), 426–439.

[42] Xinyu Zhang and Kang G Shin. 2013. Gap sense: Lightweight coordination of heterogeneous wireless devices. In INFOCOM, 2013 Proceedings IEEE. IEEE, 3094–3101.

[43] Yifan Zhang and Qun Li. 2013. HoWiES: A holistic approach to ZigBee assisted WiFi energy savings in mobile devices. In INFOCOM, 2013 Proceedings IEEE. IEEE, 1366–1374.

[44] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. 2010. ZiFi: wireless LAN disc overy via ZigBee interference signatures. In Proceedings of the sixteenth annual international conference on Mobile computing and networking. ACM, 49–60.

[45] Wenjie Zhou, Tarun Bansal, Prasun Sinha, and Kannan Srinivasan. 2014. Bbn: throughput scaling in dense enterprise wlans with bind beamforming and nulling. In MobiCom &apos;14. ACM, 165–176.

[46] Wenjie Zhou, Tanmoy Das, Lu Chen, Kannan Srinivasan, and Prasun Sinha. 2016. BASIC: backbone-assisted successive interference cancellation. In MobiCom &apos;16.

ACM, 149–161.

[[1]](" \l "_ftnref1" \o ") Note that a frequency shift keying (t) = Acos(2π(f ± ∆f )t) is equivalent to a phase shift keying of (t) = Acos(2πf t ±Φ(t)), where Φ(t) =2π∆f t.*ss*