RF-SIFTER: Sifting Signals at Layer-0.5 to Mitigate Wideband Cross-Technology Interference for IoT

Xiong Wang[ℙ], Jun Huang[ℂ✉], Bizhao Shi[ℙ], Zhe Ou[ℙ], Guojie Luo[ℙ]
Linghe Kong[𝕁], Daqing Zhang[ℙ𝕀𝕂], Shunmin Zhu[𝔸𝕋], Chenren Xu[ℙℤ𝕂✉*]

[ℙ]Peking University [ℂ]City University of Hong Kong [𝕁]Shanghai Jiao Tong University
[𝕀]Institut Polytechnique de Paris [𝔸]Alibaba Group [𝕋]Tsinghua University [ℤ]Zhongguancun Laboratory
[𝕂]Key Laboratory of High Confidence Software Technologies, Ministry of Education (PKU)

ABSTRACT

IoT uplink performance is crucial for a wide variety of IoT applications such as health sensing and industrial control, which demand reliable delivery of sensor data to the cloud. However, due to the limited transmission power budget imposed on many power-constrained IoT devices, IoT uplinks are highly susceptible to cross-technology interference (CTI) caused by coexisting networks. Previous approaches to mitigating CTI have relied on MAC/PHY designs. They suffer from poor performance and limited generality in the presence of wideband CTI sources such as Wi-Fi and RF jammer, which transmit aggressively on large spectrum chunks using diverse radio technologies.

This paper introduces RF-SIFTER, a general and highly-effective system that protects low-power IoT uplinks against intensive wideband CTI. RF-SIFTER enables technology-agnostic blind beamforming to sift signals based on bandwidth, allowing IoT signals to pass through while rejecting interference wider than the IoT band. RF-SIFTER is designed as a Layer-0.5 that is transparent to IoT MAC/PHY, ensuring general applicability and practical deployability across a wide range of coexistence scenarios. RF-SIFTER is implemented on an FPGA-based software radio platform. Extensive experiments show that RF-SIFTER can improve the SINR of IoT uplink signals by up to 29 dB and increase packet delivery ratio by 2× to 5× for ZigBee, BLE, and RFID in the presence of wideband CTI caused by 802.11ac networks and RF jamming.

CCS CONCEPTS

• **Networks** → **Network protocol design**.

KEYWORDS

Cross-Technology Interference, Jamming, IoT, Low-power wireless networks, Beamforming

ACM Reference Format:
Xiong Wang, Jun Huang, Bizhao Shi, Zhe Ou, Guojie Luo, Linghe Kong, Daqing Zhang, Shunmin Zhu, Chenren Xu. 2023. RF-SIFTER: Sifting Signals at Layer-0.5 to Mitigate Wideband Cross-Technology Interference for IoT. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3570361.3592513

1 INTRODUCTION

To meet the ever-growing demand for high data rates, wideband networks such as Wi-Fi are increasingly aggressive in utilizing large spectrum chunks, resulting in significant cross-technology interference (CTI) with Internet-of-Things (IoT). For instance, the 802.11ax standard [1] allows a single Wi-Fi link to occupy the entire 2.4 GHz ISM band, depleting spectrum resources available for coexisting IoT networks such as ZigBee and BLE. Given the often limited transmission power budget imposed on many IoT devices, the impact of CTI can be particularly severe on IoT uplinks, leading to substantial loss and delay of sensor data at the IoT gateway. This problem is especially critical for IoT applications such as health sensing and industrial control, which demand reliable delivery of sensor data to the cloud [2, 3].

To mitigate CTI for IoT, previous approaches have heavily relied on technology-specific MAC/PHY designs, resulting in poor performance and limited generality. Classic time-

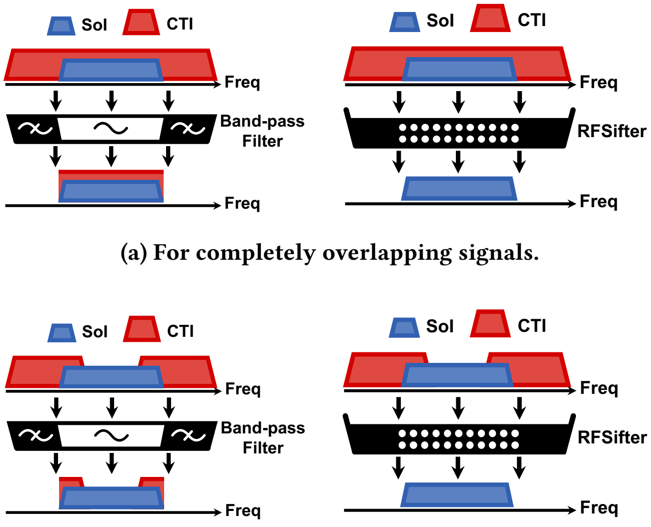
[*] ✉: jun.huang@cityu.edu.hk; chenren@pku.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3592513



(a) For completely overlapping signals.

(b) For partially overlapping signals.

Figure 1: Comparison between signal filter and sifter.

and frequency-domain schemes like WISE [4] and G-Bee [5] protect IoT packets by exploiting temporal and spectral gaps of interference. Their effectiveness depends on available spectrum resources left by coexisting networks, leading to significant performance degradation in the presence of aggressive wideband transmissions. Prior MIMO techniques like ZIMO [6] and SpaceHub [7] separate colliding signals of different radio technologies in the spatial-domain. However, they require demodulating the preambles of IoT and CTI packets, limiting their applicability to specific coexistence scenarios where the CTI modulation scheme is known and supported at the IoT receiver.

In this paper, we present RF-SIFTER, a *general* and *highly-effective* system that protects low-power IoT uplinks against intensive wideband CTI. Our key design is two-fold.

First, we propose a novel *signal sifting* algorithm that can extract IoT uplink signals from overwhelming wideband CTI. Our design is based on the observation that the signal bandwidth of power-constrained IoT is much narrower than that of coexisting wideband networks. As depicted in Fig. 1, acting like a sifter that separates wanted elements from other materials based on particle size, RF-SIFTER filters signals based on bandwidth, allowing IoT signals to pass through while rejecting all interference wider than the IoT band. In comparison, conventional frequency-based filters cannot separate signals overlapping in the same frequency band, making them ineffective in mitigating CTI. To sift signals, RF-SIFTER employs a cross-technology beamforming framework featuring a blind beam learning algorithm that leverages the bandwidth gap between IoT and coexisting wideband networks. By analyzing spatial signal covariance in the bandwidth gap, RF-SIFTER learns an optimal beam vectorto enable significant suppression of wideband CTI without prior knowledge of colliding signals. This is in contrast to conventional blind beamforming schemes that assume signal power levels or angle-of-arrivals (AoA) [8–10], which yield limited applicability in complex coexistence scenarios. To further enhance beamforming performance in the presence of frequency-selective fading, RF-SIFTER refines the beam vector by utilizing the spectral signature of IoT signals to guide a genetic beam search. The spectral signature enables accurate estimation of beam vector quality without demodulating IoT signals, enabling real-time signal processing at a significantly reduced overhead.

Second, we design a Layer-0.5 that decouples RF-SIFTER from the wireless architecture of IoT gateways, ensuring its general applicability across a wide range of coexistence scenarios featuring diverse radio technologies of IoT and wideband networks. As illustrated in Fig. 2, RF-SIFTER sifts signals received by the antenna array while preserving transparency to the traditional Layer-1, i.e., the IoT PHY. To achieve this, RF-SIFTER tackles two key challenges. Firstly, at the Layer-0.5, RF-SIFTER cannot rely on the PHY to detect the boundaries of colliding signals, which may yield misalignment of CTI and beamforming operation, resulting in performance degradation. Secondly, to mitigate multiple CTI transmissions, RF-SIFTER needs to apply different beam vectors on the same IoT packet, which may disrupt the processing of PHY-layer channel equalization, leading to demodulation errors. To address these issues, RF-SIFTER orchestrates two sliding receive windows to coordinate beamforming and beam learning, which effectively tolerates signal misalignment. We further develop a signal reshaping scheme to align RF-SIFTER outputs in the demodulation plane, which ensures consistent channel equalization at the IoT PHY.

Compared to technology-specific MAC/PHY designs, RF-SIFTER offers three key advantages. Firstly, in contrast to classic time/frequency-domain CTI avoidance schemes, RF-SIFTER enables cross-technology beamforming in the spatial-domain, allowing low-power IoT uplinks to coexist with wideband devices aggressively transmitting on large spectrum chunks. Secondly, thanks to the technology-agnostic design of the signal sifting algorithm, RF-SIFTER provides resilient uplink protection for various IoT radio technologies against all wideband CTI sources, from interfering appliances and devices using unknown modulations (e.g., microwave and RF jamming), to present and future generations of Wi-Fi and beyond. Thirdly, designed as a Layer-0.5, RF-SIFTER can be seamlessly integrated with existing MAC/PHY, enabling practical deployment on IoT gateways. To this end, RF-SIFTER can be implemented as an independent ASIC sitting between IoT baseband chips and the analog-to-digital converter (ADC) connected to the antenna array front-end. This allows substantial reduction of deployment cost, especially for today's IoT gateways that host multiple radio technologies. Without a Layer-0.5, one would have to integrate the CTI mitigation scheme with each IoT module, which can result in substantial redundancy and hardware resource waste, and is often infeasible given the proprietary nature of most IoT baseband chips.

We have implemented RF-SIFTER on an FPGA-based software radio platform and conducted extensive experiments on a testbed consisting of IoT and wideband devices using different radio technologies. Our results obtained on a 4-antenna array show that RF-SIFTER improves the SINR of IoT uplink signals by up to 29 dB and increases packet delivery ratio by 2× to 5× for ZigBee, BLE, and RFID under the high-power wideband CTI of 802.11ac networks and a commodity RF jammer that employs a proprietary modulation scheme.

Contributions.

- We propose RF-SIFTER, a general and highly-effective system that protects low-power IoT uplinks against intensive wideband CTI.
- We design a technology-agnostic signal sifting algorithm that exploits the bandwidth gap between IoT and wideband signals to enable cross-technology beamforming without prior knowledge of colliding signals.
- We design a transparent Layer-0.5 that decouples RF-SIFTER from the wireless architecture, ensuring the general applicability and practical deployability cross a wide range of coexistence scenarios.
- We implement RF-SIFTER on an FPGA-based software radio platform and conduct extensive experiments to demonstrate the significant performance improvement for three popular IoT radio technologies under different wideband CTI.

2 A BEAMFORMING PRIMER

Conventional beamforming. Beamforming is a signal processing technique that uses an antenna array to increase the SINR for the signals of interest (SoI). The output of a beamforming receiver is the weighted sum of signals received by the antenna array, which can be expressed as,

$$\mathbf{x}(t) = \mathbf{w}^*(\mathbf{H}_s\mathbf{s}(t) + \mathbf{H}_i\mathbf{i}(t) + \mathbf{n}(t)), \qquad (1)$$

where $\mathbf{w}^*$ is the conjugate transpose of the beam vector; $\mathbf{s}(t)$, $\mathbf{i}$(t), and $\mathbf{n}(t)$ are the vectors of the signal-of-interest (SoI), interference, and noise; $\mathbf{H}_s$ and $\mathbf{H}_i$ are the channel matrices of SoI and interference. Receive beamforming aims to maximize the SINR for SoI by optimizing the beam vector. To this end, conventional beamforming algorithms rely on measuring $\mathbf{H}_s$ and $\mathbf{H}_i$ based on training signals. As a result, they cannot be applied in wireless coexistence scenarios, where the CTI cannot demodulated by the receiver.

Spectrum-based beamforming. In comparison, spectrum-based beamforming is a class of blind beamforming algorithms that do not require the receiver to demodulate SoI and interference. Rather than relying on training signals to measure $\mathbf{H}_s$ and $\mathbf{H}_i$, a spectrum-based beamforming receiver optimize weight vector based on the eigen-space of signal covariance matrices, which can be estimated as $\mathbf{R}_s = \mathbf{E}\{\mathbf{s}(t)\mathbf{s}^*(t)\}$ and $\mathbf{R}_i = \mathbf{E}\{\mathbf{i}(t)\mathbf{i}^*(t)\}$, respectively. Thanks to its independence of signal types, spectrum-based beamforming has been widely used for mitigating RF jamming against radars and GPS [8–10]. However, spectrum-based beamforming relies on clean measurements of SoI or interference to estimate $\mathbf{R}_s$ or $\mathbf{R}_i$, which is feasible for only radars and GPS, where the direction of the SoI is known (i.e., for radars) or the interference is significantly stronger such that the SoI can be neglected (i.e., in GPS applications where the power of satellite signals is only -125 dB when they reach earth surface). Unfortunately, these assumptions about the powers and directions of SoI and interference are invalid in the context of CTI. As a result, spectrum-based beamforming cannot be directly applied in CTI mitigation.

3 RF-SIFTER OVERVIEW

We introduce RF-SIFTER, a technology-agnostic approach to mitigating wideband CTI for IoT uplinks. As shown in Fig. 2, RF-SIFTER performs cross-technology receive beamforming at a transparent Layer-0.5 of IoT gateways. RF-SIFTER can serve different IoT technologies without modifying the MAC/PHY of IoT or itself except configuring only two input parameters, the IoT signal bandwidth and a spectral signature for guiding beam estimation. The spectral signature is consistent across IoT devices of the same wireless technology and can be profiled offline with a one-time effort. The output of RF-SIFTER is a narrowband signal flow with significantly reduced wideband interference. From the perspective of the IoT MAC/PHY, RF-SIFTER hides the complexity of performing cross-technology beamforming on the antenna array and exposes itself as a single antenna that is highly resistant to wideband interference.

In designing RF-SIFTER, we address four challenges. Firstly, existing signal-agnostic beamforming algorithms rely on assumptions about the powers and directions of SoI and interference, which are not applicable in the context of CTI. Secondly, wireless channels are susceptible to multi-path fading, which makes it challenging to accurately estimate a cross-technology beamforming weight vector for IoT and CTI signals that have different bandwidths. Thirdly, at Layer-0.5, RF-SIFTER cannot rely on PHY-layer signal processing to detect the boundaries of IoT and CTI packets. Finally, when

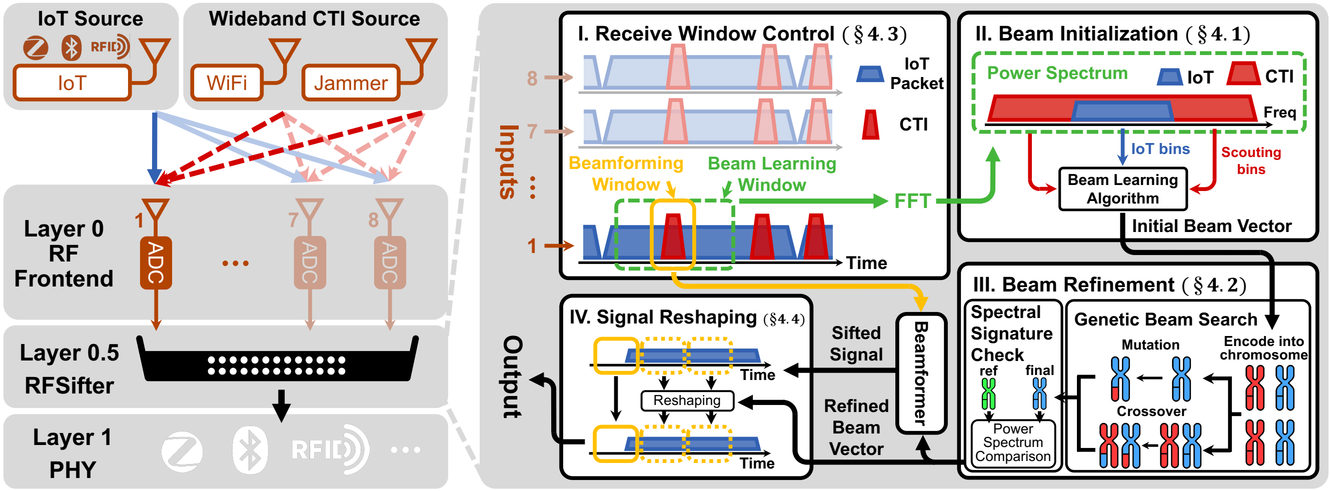


Figure 2: The system architecture of RF-SIFTER.

CTI source	Freq. band	Bandwidth
WiFi(802.11b,g,n,ac,ax)	2.4/5 GHz	10 to 80 MHz
Microwave ovens	2.45 GHz	20 to 80 MHz
Cordless phone	900 MHz	4 MHz
IoT technology		
Bluetooth/BLE	2.4 GHz	1 MHz
ZigBee	2.4 GHz	2 MHz
LoRa	433/868/915 MHz	125/500 KHz
RFID	433/900 MHz	≤ 500 KHz

Table 1: Bandwidths of IoT and wideband CTI.

an IoT packet collides with multiple CTI packets, applying different beafmorming weight vectors to the same IoT packet may lead to channel inconsistency in the PHY-layer channel equalizer, resulting in demodulation errors. To address these challenges, RF-SIFTER employs a signal processing workflow consisting of four stages.

(i) Bandwidth gap-based beam initialization. RF-SIFTER exploits the key observation that, due to the demand for high data rate, the frequency band used by CTI devices is typically much wider than that of low-power IoT signals, as compared in Tab. 1. RF-SIFTER leverages this opportunity to design a bandwidth gap-based beam learning framework, which can incorporate different beamforming algorithms and enable them to learn an initial beamforming vector without relying on demodulation or prior knowledge of IoT and CTI signals.

(ii) Genetic search-based beam refinement. Because of multipath fading, the initial beamforming weight vector measured based on bandwidth gap may be susceptible to estimation errors, resulting in a degradation of beamforming performance. RF-SIFTER addresses this issue by performing a genetic search around the initial beamforming weight vector. RF-SIFTER guides beam refinement by utilizing the spectral signature of the IoT signal, allowing accurate estimation of post-beamforming signal quality without demodulating the signal, which is critical in meeting the stringent requirement for real-time signal processing.

(iii) Receive window control. Instead of relying on the PHY-layer signal processing to detect packet boundaries, RF-SIFTER orchestrates two sliding receive windows at the Layer-0.5 to coordinate beam learning and beamforming, which effectively mitigate beamforming performance degradation caused by poor alignment between packet boundaries and beamforming operation.

(iv) Signal reshaping. RF-SIFTER employs a signal reshaping algorithm to align output signals beamformed using different weight vectors. From the perspective of the PHY-layer channel equalizer, RF-SIFTER exposes a single antenna where signals of the same IoT packet are received from a consistent wireless channel.

4 DESIGN OF RF-SIFTER

4.1 Beam Initialization

In the following, we first present a general cross-technology beam learning framework that enables different beamforming algorithms to estimate beamforming weight vector without relying on training signals and assumptions about signal powers and directions. We then integrate the Capon beamformer [11] into the framework to estimate an initial beamforming weight vector.

Exploiting bandwidth gap. Conventional beamforming algorithms rely on training signals or assumptions about the powers or directions of SoI and interference. To address this limitation, RF-SIFTER leverages the observation that the bandwidth of CTI is typically much wider than that of IoT signals. Motivated by this observation, RF-SIFTER extends the receive bandwidth of IoT gateway and performs FFT to slice the extended band into *scouting bins* that contain only wideband CTI, and *IoT bins* that contain mixed signals. Note that the scouting and IoT bins are defined in the baseband and thus are independent of the carrier frequency.

The beam learning framework of RF-SIFTER exploits the bandwidth gap to enable signal-agnostic spectrum-based beamforming introduced in section 2. To this end, RF-SIFTER estimates the covariance matrix of CTI in scouting bins. For simplicity of computation, RF-SIFTER estimates $\mathbf{R}_i$ based on two scouting bands as, $\hat{\mathbf{R}}_i = \mathbf{R}[\frac{K-B}{2} - \Delta] + \mathbf{R}[\frac{K+B}{2} + \Delta]$, where $\mathbf{R}(\cdot)$ is the signal covariance matrix in a given FFT bin, K is the size of FFT and thus $\frac{K}{2}$ stands for the center FFT bin corresponding to the frequency center of IoT signal, B is the bandwidth of IoT signal measured in number of FFT bins, and Δ is the size of a *guard band* reserved to prevent $\hat{\mathbf{R}}_\mathrm{i}$ from being polluted by IoT signal leaked in the transition band of finite impulse response filters. We set Δ to $B/2$ based on the empirical observation that the filter transition band of commodity IoT radios is typically narrower than the half of the signal bandwidth. Because IoT and CTI signals are uncorrelated, the covariance matrix of the IoT signal can be estimated as, $\hat{\mathbf{R}}_s = \mathbf{R}[\frac{K}{2}] - \hat{\mathbf{R}}_i$,

Integrating Capon beamforming. By exploiting bandwidth gap to estimate the covariance matrices of IoT and CTI signals, RF-SIFTER enables different spectrum-based algorithms to estimate the beamforming weight vector. In this work, we employ Capton beamforming [11], a computationally efficient spectrum-based beamforming algorithm widely used for anti-jamming in military applications.

The original formulation of Capon beamforming assumes the prior knowledge of the AoA of SoI. Specifically, Capon beamforming formulates the beamforming problem as a constrained optimization,

$$\min \mathbf{w}^*\mathbf{R}\mathbf{w} \quad \text{subject to} \quad \mathbf{w}\mathbf{a}^*(\theta_0) = 1.$$

where $\mathbf{R}$ is the covariance matrix of signal plus interference, $\Game(\theta)$ is the array manifold vector pointing the the direction of SoI, and $\mathbf{w}^*\mathbf{R}\mathbf{w}$ and $\mathbf{a}^*(\theta_0)$ compute the powers of beamforming output and SoI after beamforming, respectively.

To integrate Capon beamforming into RF-SIFTER, we replace the array manifold with the measured IoT signal covariance and re-formulate the optimization problem as,

$$\min \mathbf{w}^*\mathbf{R}_i\mathbf{w} \quad \text{subject to} \quad \mathbf{w}^*\mathbf{R}_s\mathbf{w} = 1.$$

By applying Lagrange multiplier, the optimal beamforming weight vector can be derived as,

$$\mathbf{w}_\text{opt} = \mathcal{P}\{\mathbf{R}_i^{-1}\mathbf{R}_s\}. \tag{2}$$

where $\mathcal{P}\{*\}$ stands for the principal eigenvector of a matrix.

4.2 Beam Refinement

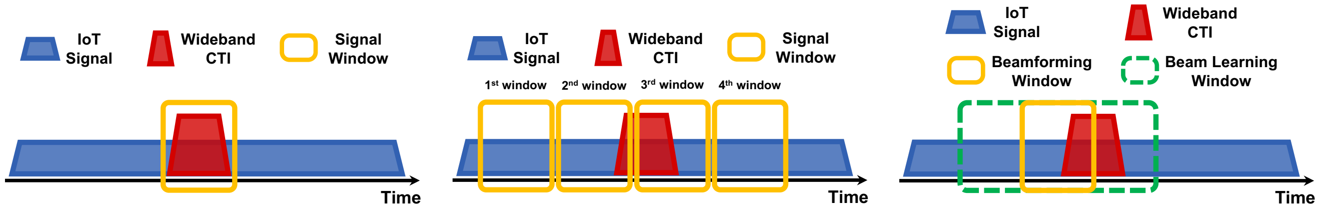
The beam learning algorithm described in §4.1 uses the CTI covariance measured in scouting bins to approximate that in IoT bins. However, due to multi-path, signal covariance may differ across frequencies, resulting in inaccurate estimation of beam vector. To address this problem, RF-SIFTER performs a genetic search around the initial beam vector to improve beamforming performance. In the following, we describe the genetic beam search algorithm and then discuss how to guide the search at Layer-0.5 without relying on signal demodulation to evaluate beamforming performance.

Genetic beam search. The genetic algorithm is an evolution-inspired metaheuristic that is particularly efficient in generating high-quality solutions for search problems [12, 13]. To enable genetic search for beam refinement, RF-SIFTER encodes a beam vector into a *chromosome* by discretizing beam weights. Specifically, RF-SIFTER represents the ranges of phase and the normalized magnitude of a beam weight using k_p and k_m bits, respectively, which transforms the beam vector of an n-antenna array into a chromosome of $n \times (k_p + k_m)$ bits. RF-SIFTER adapts the two operators of genetic algorithm, namely *mutation* and *crossover*, as follows. The crossover operator takes two beam weight vectors and computes the average beam. The mutation operator takes one beam weight vector and then randomly change the phase and amplitude of beam weights to generate a new beam weight vector.

To bootstrap the genetic beam search, RF-SIFTER generates an initial pool of parent chromosomes by mutating the chromosome corresponding to the beam vector learned based on bandwidth disparity. In each round of genetic search, RF-SIFTER produces a new generation of child chromosomes through mutation and crossover based on the current pool of parent chromosomes and then simulates natural selection by keeping 50% chromosomes that yield the best beamforming performance, which forms a new pool of parent chromosomes for the next round of search.

RF-SIFTER terminates genetic beam search when the estimated beamforming performance reaches a threshold, or a maximum number of rounds has been reached. We note that, due to the narrowband nature of IoT signals, the scouting bins employed for beam learning are typically close to the IoT bins in the frequency domain, which limits the effect of frequency-selective fading and therefore assures fast convergence of genetic beam search.

Spectral signature-based search guidance. Guiding genetic beam search at Layer-0.5 is challenging because of the lack of a metric to evaluate the fitness of a beam vector, *i.e.,* the beamforming performance. PHY-layer metrics like SINR



(a) Ideal window alignment, which however requires PHY-layer support for determining the boundaries of CTI.

(b) Naive window control. The 2nd window is poorly aligned where only a small number of dirty signals are included.

(c) The window control scheme of RF-SIFTER.

Figure 3: Comparison of different window control schemes.

require demodulating beamformed signals, which introduces significant redundant overhead.

To address this problem, RF-SIFTER leverages the observation that the modulation scheme of a wireless technology typically yields unique power spectrum characteristics [14], which can be used as a reference spectral signature to evaluate beamforming performance at Layer-0.5. To profile a reference spectral signature, RF-SIFTER measures the power spectrum of IoT signals offline under high SINR, and then normalizes it with respect to the highest spectral peak. To evaluate beam vector fitness during the genetic search, RF-SIFTER compares the normalized post-beamforming power spectrum in the IoT bins with the reference spectral signature. Intuitively, the comparison should exhibit a close match when the beam vector yields high post-beamforming SINR.

RF-SIFTER measures the distance between the reference spectral signature and the normalized post-beamforming power spectrum using the Kullback-Leibler divergence [15], a statistic distance widely used for quantifying the relative entropy between two probability distributions. Denote the reference spectral signature as P and the post-beamforming power spectrum as Q. The Kullback-Leibler distance between P and Q can be computed as,

$$D_{KL}(P||Q) = \sum_{i=0}^{B} P(x) \log \left(\frac{P(i)}{Q(i)} \right), \tag{3}$$

where $P(i)$ and $Q(i)$ are the normalized powers in the i-th IoT bin, and B is the number of bins in the IoT band.

4.3 Receive Window Control

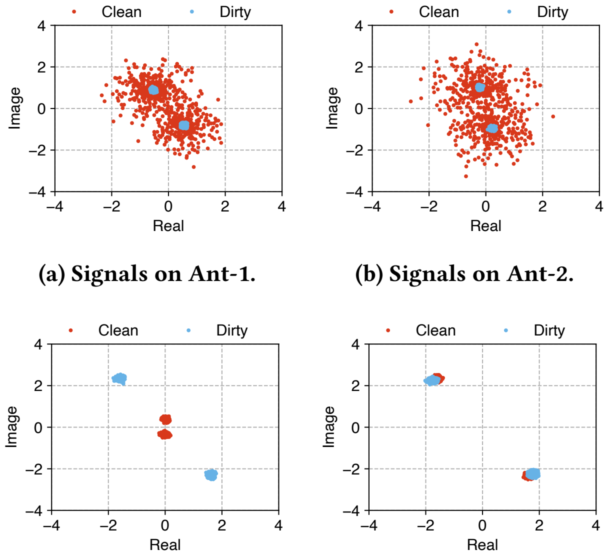
To improve the accuracy of beam learning, the signal covariance matrices need to be averaged over a window of signals to suppress noise. Ideally, the window should be aligned with a the signal block that contains both IoT and CTI signals, as illustrated in Fig. 3a. Unfortunately, operating at Layer-0.5, RF-SIFTER cannot determine the boundaries of dirty signal blocks due to the lack of support from PHY-layer packet detection schemes. On the other hand, simply dividing the received signals into discrete windows may yield poor performance. As illustrated in Fig. 3b, a poorly aligned window may include an insufficient number of dirty signal samples, resulting in inaccurate beam learning that substantially degrades beamforming performance.

RF-SIFTER addresses this problem using a sliding window-based beamforming controller. As illustrated in Fig. 3c, RF-SIFTER controls two concentric windows, namely a *beam learning window* denoted as W_L, and a *beamforming window* W_B containing a subset of signals in W_L. RF-SIFTER runs the beam learning algorithm over W_L and then applies the learned beam vector for beamforming over the signals in W_B. Then, both W_L and W_B are moved forward by $\frac{L(W_B)}{2}$, where $L(W_B)$ denotes the length of W_B. Using this method, RF-SIFTER assures that when performing beamforming over a W_B that contains dirty signals, at least $\min\left(S_d, \frac{L(W_L)}{2}\right)$ dirty signal samples will be included in beam learning, where S_d is the size of the dirty signal block. This allows RF-SIFTER to circumvent beamforming performance degradation caused by poor alignment of beam learning window, without resorting to PHY-layer for detecting interference and IoT signal boundaries.

Ideally, the window sizes should be adapted based on the length of IoT packets and interference. However, implementing window adaption at layer-0.5 may introduce extra hardware complexity. To address this problem, RF-SIFTER employs fixed W_L and W_B, which are set to 819.2 μs and 409.6 μs, respectively. On our prototype operated in 2.4 GHz band under 10 MHz sample rate, we empirically observed that the chosen window sizes allow RF-SIFTER to capture a sufficient number of signal samples for accurate beam learning, while maintaining responsiveness to channel variation.

4.4 Signal Reshaping

As IoT packets transmitted at low data-rate are typically longer than that of wideband CTI at much faster data-rate [16], RF-SIFTER may need to apply different beam vectors



(a) Signals on Ant-1.

(b) Signals on Ant-2.

(c) Beamforming outputs.

(d) After reshaping.

Figure 4: The effect of signal reshaping.

to mitigate interference bursts within the same IoT packet, which will result in channel inconsistency at the virtualized antenna. For example, consider the case shown in Fig. 4 where a 2-antenna array performs beamforming on two consecutive receive windows, including a clean one that contains only SoI, and a dirty one containing both SoI and interference. Without loss of generality, we modulate the SoI using BPSK. As shown in Fig. 4c, the beamforming outputs exhibit inconsistent channel coefficients, which may confuse the PHY. This is because the two receive windows are beamformed with different beam vectors. Specifically, denote the beam vectors of the clean and dirty receive windows as $w^c = w_1^c ... w_N^c$ and $w^d = w_1^c ... w_N^c$, respectively. From the point of view of the PHY, the SoI in these blocks appears to be received from two different channels, i.e.,

$$h^c = \sum_n w_n^c h_n, \quad \text{and} \quad h^d = \sum_n w_n^d h_n, \tag{4}$$

where h_n is the channel coefficient between the sender's antenna and the n-th antenna of the array.

To overcome this problem, RF-SIFTER reshapes beamforming outputs before passing them to the PHY. To this end, RF-SIFTER detects the presence of IoT signals based on Kullback-Leibler distance observed during beam refinement. If IoT signals are present, RF-SIFTER reshapes the beamforming outputs of all subsequent receive windows of the same IoT packet until the Kullback-Leibler distance falls below a predefined threshold. Specifically, RF-SIFTER reshapes beamforming outputs by multiplying output signals with a reshaping ratio $\gamma_k = \frac{h^1}{h^k}$, where h^1 and h^k are the channel coefficients for the first and k-th receive window of the IoT packet after beamforming. In this way, the output of the k-th receive window appears to have a channel coefficient of h^1.

The key challenge in signal reshaping at Layer-0.5 is to obtain the reshaping ratio without the support of PHY to measure channel coefficient, which typically requires preamble demodulation. To address this problem, RF-SIFTER computes the reshaping ratio γ_k for the k-th block of an IoT packet as follows,

$$\gamma_k = \frac{h^1}{h^k} = \frac{\sum_i w_i^1 h_i}{\sum_j w_j^k h_j} = \sum_i \frac{w_i^k}{\sum_j w_j^1 \frac{h_j}{h_i}}. \tag{5}$$

The reshaping ratio can be derived as long as RF-SIFTER knows the *channel ratios*, i.e., $\frac{h_j}{h_i}$, between every antenna pair i and j. To compute this reshaping ratio, RF-SIFTER computes the correlation between the beamforming output and the pre-beamforming signals received from the antenna array. Under a high post-beamforming SINR, the correlation results approximates to a coefficient $\frac{h_j}{\sum_i w_i h_i}$, where $w_0, ..., w_{N-1}$ are the beam vector. Then, RF-SIFTER computes the ratio between these coefficients to obtain channel ratios. We note that the estimated reshaping ratio can be inaccurate under low post-beamforming SINR. However, it suffices to assure that IoT packets that can be correctly decoded after beamforming will not corrupt by channel inconsistency at the PHY. Fig. 4d shows the effect of reshaping. As shown in the figure, the signals of beamforming outputs are closely aligned on the constellation map after reshaping, exhibiting a coherent channel coefficient.

5 IMPLEMENTATION

We have implemented RF-SIFTER based on FPGA and deployed it as a layer-0.5 on a software radiop platform consisting of four USRPs synchronized using a 10 MHz external clock. Each USRP is equipped with two receiving chains, allowing an antenna array of up to 8 antennas. The antenna array receives on a 10 MHz channel, which are further divided into 32 bins by performing FFT. To prevent IoT signal leakage from polluting the measurement of CTI covariance, we set a guard band of $\frac{B}{2}$ and select the bins centered at $\pm B$ as the scout bands, where B is the signal bandwidth of the IoT technology. The prototype of RF-SIFTER sits below open-source software PHYs of three IoT technologies, i.e., ZigBee [17], BLE [18], and RFID [19], which are widely used in a broad range of IoT applications, such as home/building automation, industrial control, and low-power media streaming. We run RF-SIFTER under these PHYs without modifying their implementations. To integrate RF-SIFTER with different IoT PHY, we configure RF-SIFTER by adjusting two inputs,

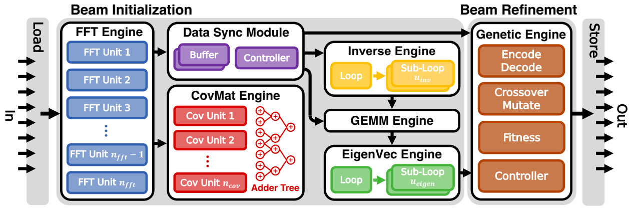
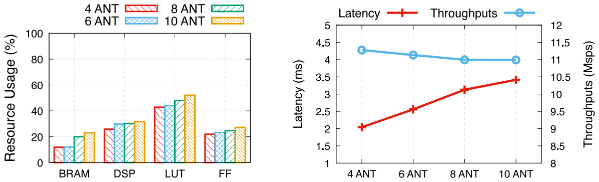


Figure 5: FPGA-based implementation of RF-SIFTER.

the IoT signal bandwidth and the spectrum signature used for beam refinement.

We implement the compute-intensive beam initialization and refinement of RF-SIFTER on Xilinx ZCU104 [20], an FPGA platform widely used for edge-based signal processing and computer vision. Fig. 5 shows the structure of the FPGA-based implementation. We exploit the parallelism and pipelining of FPGA, and organize signal processing stages in a coarse-grained dataflow with streaming and ping-pong buffer interfaces to maximize system throughput. Specifically, the FFT engine transforms signals received from the antenna array into the frequency domain in parallel. The CovMat engine computes signal covariance matrices on scout bins and the center bin. The Inverse engine, GEMM engine, and EigenVec engine perform eigen-decomposition and return the principal eigenvector as an initial beam. The Genetic engine refines the initial beam by performing a maximum of 4 rounds of search over a pool of 32 genes. The refined beam is then passed to a software module that performs beamforming and signal reshaping.

Fig. 6 shows the resource usage and signal processing efficiency of RF-SIFTER on Xilinx ZCU104. Specifically, for a 4-element array, RF-SIFTER uses only 11.9% BRAM, 25.8% DSP, 21.9% FF, and 42.8% LUT, and maintains signal processing latency under 2.041 ms with a clock frequency of 150 MHz, which can be further reduced by aggressively tuning the trade off between execution frequency and power consumption at the IoT gateway. Despite ms-level signal processing latency, the pipelined design allows RF-SIFTER to achieve a high signal processing throughput of up to 11.28 Msps for the 4-element array, which is very abundant for supporting IoT technologies with narrow receive bandwidth (i.e., typically smaller than 2 MHz as shown in Table 1). Moreover, we observe that the increase of resource usage and the degradation of signal processing efficiency are much slower than the increase of array size. For example, when the array size grows to 8 from 4, the use of DSP and LUT only slightly increases by 4.4% and 5.1%, respectively, and the signal processing throughput only drops by 2.5%. The results suggest the potential for supporting a large antenna array to further boost beamforming performance.



(a) Breakdown of resource usage on Xilinx MPSoC ZCU104 for antenna arrays of different size.

(b) Signal processing throughput and latency of beam initialization and beam refinement for antenna arrays of different size.

Figure 6: Evaluation of FPGA-based implementation.

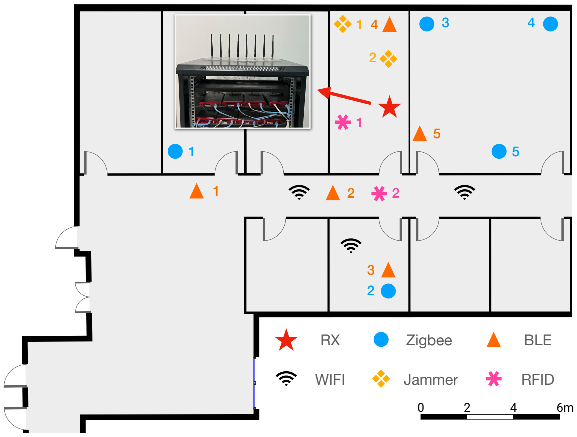


Figure 7: The deployment of IoT and interfering nodes.

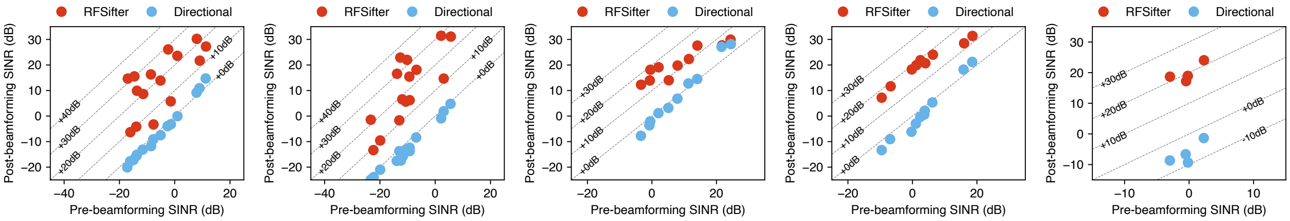
6 EVALUATION

6.1 Experiment Setup

Testbed. Our experiments are conducted in a testbed consisting of IoT nodes and wideband CTI devices deployed in an indoor office environment, as shown in Fig. 7.

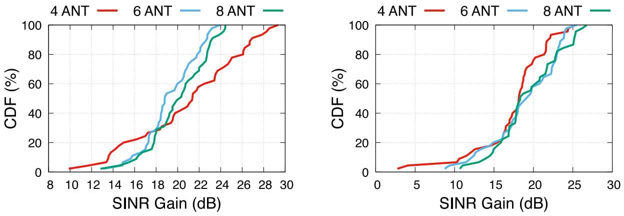
We employ commodity off-the-shelf Zigbee, BLE, and RFID devices to transmit packets to our software radio-based gateway. Due to the large channel switching delay of software radios, we disable frequency hopping of BLE devices and configure them to transmit on a fixed channel. However, we note that RF-SIFTER is designed as a layer-0.5 and therefore its operation is independent to carrier frequency. RF-SIFTER can support frequency hopping as long as the radio front end can switch carrier frequency in real-time.

In our experiments, we employ two representative CTI sources, including a 2.4 GHz 802.11ac-based Wi-Fi network that interferes with ZigBee and BLE devices, and a wideband RF jammer that not only covers the 2.4 GHz band but also



(a) Zigbee w/ Wi-Fi **(b) BLE w/ Wi-Fi** **(c) Zigbee w/ jammer** **(d) BLE w/ jammer** **(e) RFID w/ jammer**

Figure 8: Pre- and post-beamforming SINR of RF-SIFTER and directional beamforming for different IoT and CTI technologies.



(a) Under Wi-Fi interference. **(b) Under RF jamming.**

Figure 9: Post-beamforming SINR gains of RF-SIFTER with antenna arrays of different sizes.

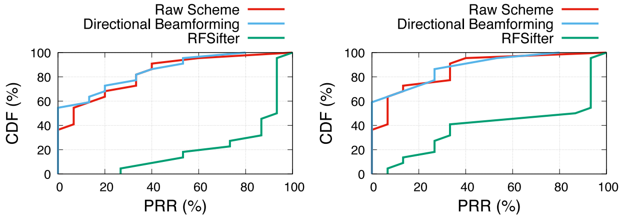
interferes with the RFID node operating in the 915 MHz band. Specifically, the RF jammer is a ZY-002J3 which continuously jams a wide spectrum using a chirp-based proprietary modulation. The Wi-Fi network consists of a 4-antenna TP-Link WDR7500 access point and a 4-antenna RTL8814-based USB adaptor. To generate Wi-Fi interference, we use the client to ping the access point using 1000-byte packets. In addition, we study RF-SIFTER performance in the presence of interference of real Wi-Fi applications, including video streaming and large file downloading.

Baselines. We compare RF-SIFTER with two baselines. The *raw scheme* simply selects the first antenna, whereas the *directional beamforming* scheme selects one optimal receiving direction. To implement directional beamforming, we calibrate the USRPs by taking an additional reference signal to correct the phase offsets between receive chains. The antenna array is structured as a uniform linear array. We measure the AoA spectrum using MUSIC, identify all peaks, and then decode the signals along all peak directions to find the one that results in best receiving performance for IoT.

6.2 SINR Improvement

Under Wi-Fi interference. We first compare the pre- and post-beaforming SINR of RF-SIFTER and directional beamforming for ZigBee and BLE uplinks under Wi-Fi interference. In this experiment, both RF-SIFTER and directional beamforming employ a 4-antenna array. As shown in Fig. 8a and Fig. 8b, RF-SIFTER offers substantial SINR improvements after performing cross-technology beamforming. For example, on all BLE links, the post-beamforming SINR of RF-SIFTER is at least 10 dB higher than that before beamforming. Notably, on approximately 56% of the links analyzed, the SINR gain is over 20 dB and can reach as high as about 38 dB. In comparison, owing to the multi-path condition in the indoor environment, the efficacy of directional beamforming is considerably limited on most of the studied links under all SINR conditions. Moreover, due to the limited transmission power budget of IoT nodes, IoT uplink signals are susceptible to fading and noise, making it difficult to accurately measure signal AoA at the gateway, which further degrades the performance of directional beamforming.

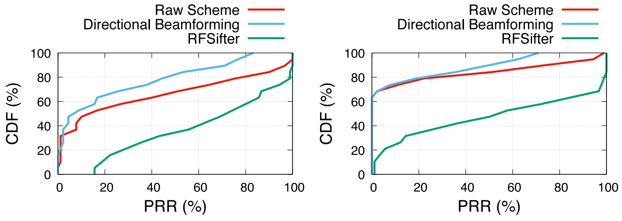
Under RF Jamming. We then study the performance of RF-SIFTER in the presence of RF jamming, which uses a proprietary chirp-based modulation that differs substantially from the 802.11ac-based Wi-Fi network. Similar to the results obtained under Wi-Fi interference, we observe that RF-SIFTER significantly improves SINR for ZigBee and BLE uplinks under RF jamming. In comparison, directional beamforming performs poorly for both ZigBee and BLE uplinks under all SINR conditions. In particular, RF-SIFTER demonstrates notably higher SINR gains under low pre-beamforming SINR conditions. For instance, on ZigBee uplinks, the SINR gains of RF-SIFTER are approximately 5 dB and 17 dB when the pre-beamforming SINR is over 20 dB and around 0 dB, respectively. This is because RF-SIFTER achieves more precise estimation of the CTI covariance when the CTI on the scouting bands overwhelms the noise floor, which makes RF-SIFTER particularly effective on IoT uplinks that experience higher levels of wideband interference. We then further evaluate RF-SIFTER for RFID nodes under the interference of 915 MHz RF jamming. As shown in Fig. 8e, RF-SIFTER brings 16 dB to 26 dB SINR improvement, which demonstrates the



(a) Video downloading

(b) File downloading

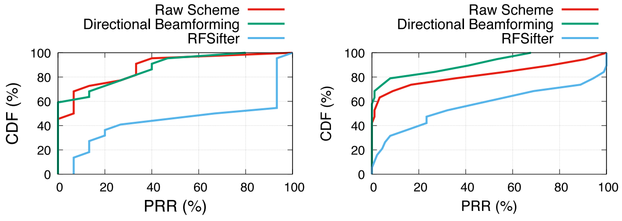
Figure 10: The PRRs of ZigBee uplinks under the interference of real Wi-Fi traffics.



(a) Video downloading

(b) File downloading

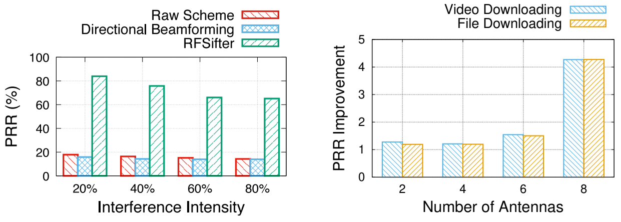
Figure 11: The PRRs of BLE uplinks under the interference of real Wi-Fi traffics.



(a) Zigbee

(b) BLE

Figure 12: The PRRs of ZigBee and BLE uplinks under RF jamming.



(a) Channel utilization ratio.

(b) Antenna array Size.

Figure 13: The impacts of CTI channel utilization ratio and antenna array size on PRR.

efficacy of RF-SIFTER when operating at different frequency bands.

Effects of array size. Fig. 9 examines the effectiveness of RF-SIFTER for ZigBee uplinks with antenna arrays of different sizes. As expected, we observe that the SINR gain of RF-SIFTER increases with the number of antennas. This is due to the fact that with more antennas, there is a higher degree of freedom available for cross-technology beamforming, thereby enhancing RF-SIFTER's performance.

6.3 PRR Improvement

Under Wi-Fi interference. We evaluate RF-SIFTER's impact on link-layer performance by examining the packet reception ratio (PRR) of IoT uplinks. We first conduct experiments using the 4-antenna array under the Wi-Fi interference produced by two real applications, namely live video streaming and large file downloading. Fig. 10a compares the PRRs of the raw scheme, directional beamforming, and RF-SIFTER for ZigBee uplinks. We observe that RF-SIFTER outperforms the other schemes by a great margin under the interference of live video downloading. Specifically, the average PRRs of the raw scheme, directional beamforming, and RF-SIFTER are 19.09%, 16.35%, and 81.5%. RF-SIFTER boosts PRR by 3.3× and 4×, respectively. As shown in Fig. 10b, similar comparison results can be observed under the interference of file downloading, despite that the PRR is relatively lower than that under the interference of live video streaming. This is because file downloading produces intensive interference with a high channel utilization ratio, while the traffics of video streaming typically have larger time intervals between consecutive video frames. To further study the impact of CTI channel utilization ratio, we conduct experiments by generating Wi-Fi interference with controlled packet intervals. Specifically, we use the Wi-Fi client to ping the access point and adjust the time interval of ping packets. As shown in Fig. 13a, we observe that the PRR of RF-SIFTER slightly decreases as the Wi-Fi channel utilization ratio increases. However, even under intensive Wi-Fi interference with a channel utilization ratio of 80%, RF-SIFTER can maintain the PRR above 60%, while the PRRs of the raw scheme and directional beamforming drop to below 15%. We note that the PRR can be further increased by integrating RF-SIFTER with MAC layer schemes, such as retransmission and forward error correcting coding. We have repeated the above experiments for BLE uplinks. Similar results can be observed in Fig. 11a.

Under RF jamming. We then evaluate RF-SIFTER's improvement of PRR under the interference of RF jamming. Different from Wi-Fi, the RF jammer transmits interfering signals continuously. As shown in Fig. 12a, under RF jamming, RF-SIFTER achieves a PRR of 57.86%, which is an improvement of 3.1× and 3.1× compared to the raw scheme and directional beamforming, respectively. Similar results can be observed for BLE links, as shown in Fig. 12b. The significant PRR improvement demonstrates the RF-SIFTER's ability in exploiting spatial diversity to mitigate intensive interference that is saturated in time-domain.

Effects of array size. Fig. 13b shows the PRR improvement ratio of RF-SIFTER over the raw scheme with different numbers of antennas. The performance of RF-SIFTER improves as the size of antenna array increases, which is expected because a larger antenna array enables RF-SIFTER to better exploit spatial diversity. In particular, we observe that the improvement ratio enjoys a significant boost with an 8-antenna array. The gain is mostly likely the result when the number of antennas overwhelms the total number of interference signals, including their multi-path copies. In comparison, a smaller antenna may limit RF-SIFTER ability to separate signals of different spatial properties. In theory, an array of $N + 1$ antennas allows RF-SIFTER to mitigate N interference signals of different spatial properties. We note that, as shown in section 5, the hardware cost of RF-SIFTER grows slowly with array size, which suggests the feasibility of incorporating a large array in practical deployments to achieve the significant performance improvement demonstrated by the 8-antenna array shown in Fig. 13b.

6.4 Micro Benchmark

Effects of beam refinement. To improve beamforming performance under multi-path conditions, RF-SIFTER performs an iterative search to refine the initial beamforming weight vector estimated based on bandwidth gap. Fig. 14 shows the SINR improvement brought by RF-SIFTER compared to the raw scheme under the interference of the Wi-Fi network and RF jammer, respectively. We observe that the performance of RF-SIFTER using the initial beam vector is comparable to that with beam refinement under only high SINR conditions (i.e., 5 dB). However, when SINR decreases, RF-SIFTER with beam refinement can offer a further beamforming gain of about 4 dB and 6 dB under the interference of Wi-Fi and RF jammer, respectively. The reason is that under low SINR conditions, RF-SIFTER is more susceptible to frequency selective fading, which renders inaccurate initial beam vector estimations. As shown in Fig. 14, RF-SIFTER with the beam refinement can improve PRR by 30% and 40% compared to that without beam refinement under the interference of Wi-Fi and RF jammer, respectively. The results demonstrate the efficacy of beam refinement, especially under intensive wideband CTI.

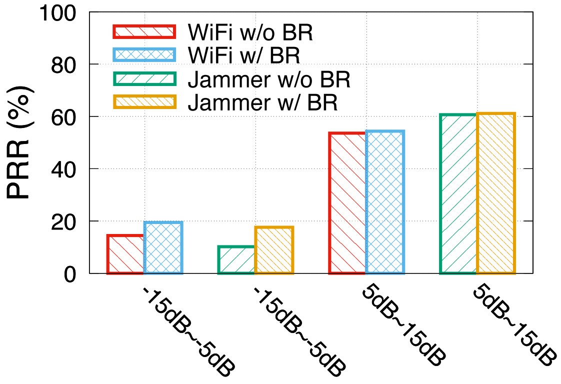


Figure 14: The effects of beam refinement.

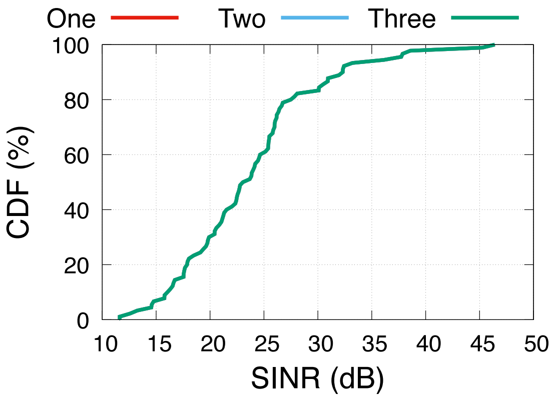
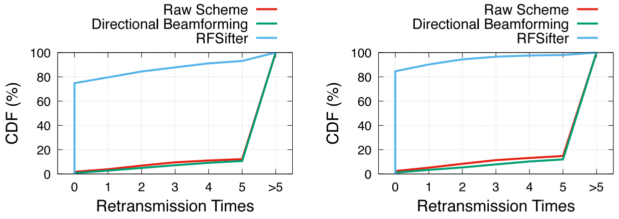


Figure 15: The impact of the CTI transmitter's antenna number.

Impacts of the CTI transmitter's antenna number. We then evaluate the impacts of the CTI transmitter's antenna number on the performance of RF-SIFTER. In this experiment, we run RF-SIFTER on an 8-antenna array. It can be observed from Fig. 15 that the SINR improvement of RF-SIFTER remains unaffected despite the increase of the CTI transmitter's antenna number. The result demonstrates that RF-SIFTER is able to maintain efficacy when there is sufficient degree of freedom for performing cross-technology beamforming.

Effects of retransmission. Next, we study the effect of retranmission on RF-SIFTER's performance. We conduct experiments using ZigBee under the Wi-Fi interference generated by live video streaming and large file downloading. As shown in Fig. 16, without retransmissions, RF-SIFTER achieves a PRR of about 80%, which is significantly higher than that of the raw scheme and directional beamforming, which are only around 2%. As expected, the PRRs of all schemes increase with the number of retransmission. However, due to the poor resistance to CTI, the performance of the raw scheme and directional beamforming increase slowly with



(a) File downloading

(b) Video downloading

Figure 16: The impacts of packet retranmission on PRR.

the number of retranmissions. For example, under the interference of file downloading, the PRRs of the raw scheme and directional beamforming are only 17% and 16% after five retransmissions, respectively. In contrast, after only four retransmissions, the PRR of RF-SIFTER increases to 95% and 98% under the interference of file downloading and video streaming, respectively. To support IoT applications that have strict requirement on reliability, one can further integrate RF-SIFTER with MAC/PHY-layer schemes such as error correcting coding.

7 DISCUSSION

Deployment and Implementation Cost. To exploit the bandwidth gap between IoT signals and wideband CTI, RF-SIFTER needs to operate on a wider receiving band, which requires ADCs of higher sampling rate. However, since IoT technologies typically operate on narrow bandwidths, the sampling rate required by RF-SIFTER is moderate (i.e., 10 Msps in current implementation) and much lower than that required by wideband receivers.

When integrated into the IoT gateway, RF-SIFTER can be implemented as an independent ASIC that sits between the baseband chip of IoT and the ADC. Thanks to the Layer-0.5 design, the integration of RF-SIFTER does not require any changes to the baseband chip of IoT, which not only ensures practical deployability but also significantly reduces integration cost, especially for modern IoT gateways that host multiple IoT technologies. Without a Layer-0.5 design, one would have to integrate CTI mitigation with each IoT module, resulting in substantial redundancy and hardware resource waste.

Technology-aware Beamforming. Technology-aware beamforming directly estimates beamforming weight vectors using training signals, which may result in a more accurate estimation than RF-SIFTER. However, technology-aware beamforming requires the receiver to demodulate IoT packet preambles, which is impossible in the event of signal collision and is not applicable to CTI with unknown/non-sense modulations, such as RF jamming and microwave interference. Moreover, implementing technology-aware beamforming requires the receiver to integrate and modify the PHY layers of both IoT and CTI, which can incur significant cost given the vast diversity of wireless technologies.

CTI on Downlink ACKs. Wideband CTI may also interfere with downlink ACKs transmitted by the IoT gateway to acknowledge uplink packets. However, the severity of this issue is significantly lower compared to that of CTI on IoT uplinks. Firstly, the transmission power budget of IoT gateways is much higher than that of IoT nodes. Secondly, ACK packets are much shorter than IoT uplink packets carrying data, which greatly reduces the likelihood of collision. These two factors make downlink ACKs significantly more resilient against CTI than IoT unplink packets, rendering the latter a prominent bottleneck for IoT applications.

Narrowband CTI. While this work focuses on mitigating wideband CTI for IoT devices operating on narrowband, the design of RF-SIFTER can be easily extended to mitigate narrowband CTI on wideband networks. Specifically, RF-SIFTER can exploit the bandwidth gap to collect clean measurements of wideband signals, enabling beamforming to suppress narrowband CTI. We leave this to future work.

Security Implications. RF-SIFTER is agnostic to the modulation of interference and therefore can be applied in a wide range of anti-jamming applications. Conventional frequency hopping schemes evade jamming by randomly switching communication channels, which can be defeated by a wideband jammer that can interfere with the entire spectrum. To address this limitation, RF-SIFTER can enable a *bandwidth hopping* paradigm where the sender transmit on pseudo-randomly changing bandwidth, allowing the receiver to exploit the bandwidth gap between the sender and the jammer to mitigate jamming using beamforming. We leave this to future work.

8 RELATED WORK

CTI Mitigation. Previous work on CTI mitigation can be categorized into PHY-layer, MAC-layer, and spatial-domain schemes. A comparison between RF-SIFTER and existing approaches is summarized in Tab. 2.

To avoid CTI, early MAC/PHY-layer schemes focused on scheduling low-power wireless transmissions in the time- and frequency-domain. For example, WISE [4] schedules ZigBee transmissions during the intervals of Wi-Fi packets. ZiSense [24] detects ZigBee signal patterns to wake up ZigBee receiver only when there is no interference. G-Bee [5] hides IoT signals in the guard band of 802.11b. Channel hopping schemes evade interference by randomly switching

	Technology dependency		Architecture modification	
	CTI source	Protected signal	TX	RX
BUZZBUZZ[21]	WiFi	Zigbee	Physical layer	Physical layer
WISE[4]	WiFi	Zigbee	MAC layer	None
TIMO[22]	Narrowband signals with one antenna	WiFi	None	Physical layer
ZIMO[6]	WiFi	Zigbee	None	Physical layer
ECC[23]	WiFi	Zigbee	MAC layer	None
RFSifter	All wideband signals	All IoT signals with narrowband	None	Layer-0.5

Table 2: Comparison between RF-SIFTER and existing CTI solutions.

the communication channel. However, time- and frequency-domain schemes perform poorly in today's crowded spectrum. For instance, the 802.11ax released in 2020 [1] allows a single Wi-Fi access point to transmit on an 80 MHz channel, which exhausts the spectrum of the 2.4 GHz band. Moreover, many throughput demanding applications, such as HD video streaming, require extensive channel utilization, which severely limits the opportunity for scheduling IoT transmissions in time-domain.

Previous studies have utilized forward error correction (FEC) and retranmission to combat CTI. For example, BuzzBuzz [21] utilizes redundant preambles and Hamming coding to protect ZigBee packets against Wi-Fi interference. However, FEC offers limited gain under high power interference, and its efficacy heavily relies on coding rate. The maximum gain of coding under a certain level of SINR is fundamentally limited by theory [25]. In comparison, RF-SIFTER exploits spatial diversity and therefore enables significant gain even under extremely low SINR conditions. For example, the widely used Reed-Solomon (15, 11) code provides a gain of only 2 dB under an SINR of 6 dB [26], whereas RF-SIFTER can delivery a gain of 20 dB to 40 dB under an SINR of -10 dB (Fig. 8).

Several studies have proposed to coordinate the transmission of wireless networks to avoid CTI. CBT [27] requires ZigBee node to transmit a busy tone to improve its visibility to Wi-Fi. Cross-technology communication protocols [23, 28] leverage explicit channel coordination between coexisting wireless networks. However, explicit coexistence coordination requires modifications to both IoT and CTI devices.

MIMO and Interference Nulling. Several studies have proposed using cross-technology MIMO to mitigate CTI [6, 22]. TIMO [22] uses MIMO to mitigate narrowband CTI generated by single-antenna devices for Wi-Fi. ZIMO [6] requires a ZigBee receiver to demodulate the packet preambles of ZigBee or Wi-Fi to perform MIMO. However, both TIMO and ZIMO are technology-dependent, limiting their general applicability and practical deployability.

Interference nulling can suppress interference by forming a null towards the direction of interfering signals [9, 29, 30]. However, to apply interference nulling in CTI mitigation, the receiver must decode signals from all directions to identify the SoI, which makes this solution technology-dependent. Moreover, examining signals from all directions can incur high complexity, especially in multi-path environments. SpaceHub [7] leverages an antenna array as a relay node to separate the signals of different technologies coming from distinct directions. The relay node then performs directional beamforming to forward the signals to different receivers. However, SpaceHub is technology-dependent. In particular, it requires the relay node to integrate the PHY of all coexisting networks, which can incur significant implementation cost.

9 CONCLUSION

We presented RF-SIFTER, a general and highly-effective system that protects low-power IoT uplinks against intensive wideband CTI. By sifting colliding signals at a transparent Layer-0.5, RF-SIFTER achieves significant suppression of wideband CTI while ensuring general applicability and practical deployability on IoT gateways. The design of RF-SIFTER can be extended to facilitate a wide range of wireless coexistence and security scenarios, such as protecting wideband networks from narrowband interference and mitigating jamming by utilizing bandwidth gaps.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their constructive critique and valuable comments. We also thank Zhen Tan and Tian Liu for their contributions to an early version of this work. This work is supported in part by the CityU Start-up (Project No. 9610579), GRF grant from the Research Grants Council of Hong Kong (Project No. 11204722), National Key Research and Development Plan, China (Grant No. 2020YFB1710900) and National Natural Science Foundation of China (Grant No. 62022005, 62272010 and 62061146001). Jun Huang and Chenren Xu are the corresponding authors.

REFERENCES

[1] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. A tutorial on IEEE 802.11 ax high efficiency WLANs. *IEEE Communications Surveys & Tutorials*, 21(1):197–216, 2018.

[2] Avik Ghose, Priyanka Sinha, Chirabrata Bhaumik, Aniruddha Sinha, Amit Agrawal, and Anirban Dutta Choudhury. UbiHeld: ubiquitous healthcare monitoring system for elderly and chronic patients. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1255–1264, 2013.

[3] C Arcadius Tokognon, Bin Gao, Gui Yun Tian, and Yan Yan. Structural health monitoring framework based on internet of things: A survey. *IEEE Internet of Things Journal*, 4(3):619–635, 2017.

[4] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. Beyond co-existence: Exploiting WiFi white space for ZigBee performance assurance. In *The 18th IEEE International Conference on Network Protocols*, pages 305–314. IEEE, 2010.

[5] Yoon Chae, Shuai Wang, and Song Min Kim. Exploiting WiFi guard band for safeguarded Zigbee. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 172–184, 2018.

[6] Yan Yubo, Yang Panlong, Li Xiangyang, Tao Yue, Zhang Lan, and You Lizhao. Zimo: Building cross-technology MIMO to harmonize ZigBee smog with WiFi flash without intervention. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 465–476, 2013.

[7] Meng Meng, Lizhao You, Kun Tan, Jiansong Zhang, and Wenjie Wang. SpaceHub: A smart relay system for smart home. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, pages 1–7, 2015.

[8] Ömer Can Dabak, Fatih Erdem, Tolga Sönmez, Lale Alatan, and S Sencer Koç. Interference suppression in a GPS receiver with 4 element array design and implementation of beamforming algorithms. In *IEEE/ION Position, Location and Navigation Symposium*, pages 645–652. IEEE, 2016.

[9] Dan Lu, Renbiao Wu, and Wenyi Wang. Robust widenull anti-jamming algorithm for high dynamic gps. In *2012 IEEE 11th International Conference on Signal Processing*, 2012.

[10] David S De Lorenzo et al. *Navigation accuracy and interference rejection for GPS adaptive antenna arrays*. Stanford University, 2007.

[11] Jack Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8):1408–1418, 1969.

[12] SN Sivanandam and SN Deepa. Genetic algorithms. In *Introduction to genetic algorithms*, pages 15–37. Springer, 2008.

[13] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[14] Shravan Rayanchu, Ashish Patro, and Suman Banerjee. Airshark: detecting non-WiFi RF devices using commodity WiFi hardware. In *Proceedings of the ACM SIGCOMM conference on Internet measurement conference*, pages 137–154, 2011.

[15] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[16] P. Yang, Y. Yan, X. Y. Li, Y. Zhang, T. Yue, and L. You. Taming cross-technology interference for WiFi and ZigBee coexistence networks. *IEEE Transactions on Mobile Computing*, 15(4):1009–1021, 2016.

[17] gr-ieee802-15-4. https://github.com/bastibl/gr-ieee802-15-4.

[18] https://github.com/hhornbacher/gr-ble.

[19] Gen2-uhf-rfid-reader. https://github.com/nkargas/Gen2-UHF-RFID-Reader.

[20] https://www.xilinx.com/support/documents/boards_and_kits/zcu104/ug1267-zcu104-eval-bd.pdf.

[21] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving WiFi interference in low power ZigBee networks. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 309–322, 2010.

[22] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. In *Proceedings of the ACM SIGCOMM Conference*, pages 170–181, 2011.

[23] Zhimeng Yin, Zhijun Li, Song Min Kim, and Tian He. Explicit channel coordination via cross-technology communication. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 178–190, 2018.

[24] Xiaolong Zheng, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. Zisense: towards interference resilient duty cycling in wireless sensor networks. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 119–133, 2014.

[25] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2001.

[26] Kan Yu, Filip Baracć, Mikael Gidlund, Johan Åkerberg, and Mats Björkman. A flexible error correction scheme for ieee 802.15.4-based industrial wireless sensor networks. In *2012 IEEE International Symposium on Industrial Electronics*, 2012.

[27] Xinyu Zhang and Kang G Shin. Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–11, 2011.

[28] Zhijun Li and Tian He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 2–14, 2017.

[29] Kenneth Mills, Fauzia Ahmad, Moeness G. Amin, and Braham Himed. Fast iterative interpolated beamforming for interference doa estimation in gnss receivers using fully augmentable arrays. In *2019 IEEE Radar Conference (RadarConf)*, 2019.

[30] Bo Qiu, Wei Liu, and Renbiao Wu. Blind interference suppression for satellite navigation signals based on antenna arrays. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013.