공학박사 학위논문

MAC/PHY Strategies for Interference Resilient 2.4 GHz Wireless Connectivity Technologies

2.4 GHz 무선 연결 기술의 간섭 강인성 향상을 위한 MAC/PHY 기법

2017년 2월

서울대학교 대학원

전기·컴퓨터공학부

손 위 평

Abstract

Performance degradation due to ambient interference is emerging as a prominent problem for the wireless connectivity technologies at increasingly crowded 2.4 GHz unlicensed band, e.g., Wi-Fi, classic Bluetooth (BT), and Bluetooth Low Energy (BLE). They suffer from severe performance degradations due to both homogeneous (i.e., from the same type of technology) and heterogeneous (from different types of technologies) interference. How to efficiently and effectively manage the interference has been a major technical issue for the wireless connectivity technologies at 2.4 GHz unlicensed band.

In this dissertation, we will consider three research topics, i.e., 1) faster Wi-Fi direct device discovery, 2) better BT and Wi-Fi coexistence, and 3) robust BLE-based indoor localization, by focusing on how to deal with the ambient interferences in order to substantially improve the performance of the 2.4 GHz wireless connectivity technologies.

Firstly, Wi-Fi Direct, now part of Android smartphones, makes it possible for Wi-Fi devices to communicate directly without passing through an access point. Before starting actual data exchange, two devices should apparently find each other through a device discovery process, called *find phase*. We identify an inherent drawback of the legacy find phase in terms of the resilience towards the ambient interference, whereby the device discovery delay tends to become intolerable. Accordingly, we propose a simple and efficient scheme, called *Listen Channel Randomization* (LCR), in order to expedite the device discovery. Both the legacy find phase and LCR are evaluated with absorbing Markov chain model, NS-3 simulation, and prototype-based experiments to corroborate the delay reduction achieved by LCR.

Secondly, dense Wi-Fi and BT environments become increasingly common so that the coexistence issues between Wi-Fi and BT are imperative to solve. We propose

BlueCoDE, a coordination scheme for multiple neighboring BT piconets, to make them collision-free and less harmful to Wi-Fi. BlueCoDE does not require any modification of BT's existing PHY and MAC design, and is practically feasible. We implement a prototype of BlueCoDE on Ubertooth One platform, and corroborate the performance gain via analysis, NS-3 simulation, and prototype-based experiments. Our experimental results show that with merely 10 legacy BT piconets, neighboring Wi-Fi network becomes useless achieving zero throughput, while BlueCoDE makes the Wi-Fi throughput always remain above 12 Mb/s. We expect BlueCoDE to be a breakthrough solution for coexistence in dense Wi-Fi and BT environments.

Finally, BLE has recently attracted enormous attention for its usage in indoor localization system. Most BLE-based indoor localization systems utilize Received Signal Strength Indication (RSSI) of the received BLE packets to infer current location. We experimentally find out that, when there exist Wi-Fi networks in vicinity, the performance of BLE-based indoor localization system heavily degrades due to the ambient Wi-Fi interference causing BLE packet losses. To mitigate the performance degradation, we propose RESCUE, a robust BLE packet detection scheme for RSSI acquisition in indoor localization system. It exploits characteristics of the received signal's estimated Carrier Frequency Offset (CFO) values and timing information. We implement RESCUE on Ubertooth platform, and demonstrate its performance gain via real environment indoor localization experiments.

In summary, from Chapter 2 to Chapter 4, the aforementioned three pieces of the research work, i.e., LCR for faster Wi-Fi Direct device discovery, BlueCoDE for better Wi-Fi and BT coexistence, and RESCUE for robust BLE-based indoor localization system, will be presented, respectively.

keywords: 2.4 GHz, Wi-Fi, Bluetooth, BLE, device discovery, coexistence, localization

student number: 2010-24083

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Coexistence Issues at 2.4 GHz Unlicensed Band

Spectrum scarcity is the biggest obstacle in achieving high capacity wireless connectivity technologies. So far, spectrum sharing has been advocated as a key remedy for this problem, especially, for the Wireless Local Area Network (WLAN) and Wireless Personal Area Network (WPAN) at 2.4 GHz unlicensed band, e.g., Wi-Fi, classic Bluetooth (BT), and Bluetooth Low Energy (BLE).

However, along with widespread deployment of WLAN and WPAN at 2.4 GHz unlicensed band, severe performance degradation has been observed when heterogeneous devices share the same frequency band; such a coexistence problem is rooted in the entirely different Physical (PHY) and Medium Access Control (MAC) designs and the lack of coordination. Accordingly, how to efficiently and effectively alleviate the impact of heterogeneous interference has been a major technical issue for the coexistence problem.

Besides, due to the accelerated deployments and the growing popularity of the wireless connectivity technologies, highly-dense coexisting environments become increasingly common, e.g., increasing number of commuters enjoy audio streaming via BT headsets at subway station covered by dozens of Wi-Fi networks, and dozens of call center staffs make hands-free calls via BT headphones in office environment covered by several Wi-Fi networks. This trend intensifies the severity of the coexistence problem and makes it practically essential to deal with the coexistence problems in densely deployed environments.

In this dissertation, three technical issues, which are all focusing on how to deal with the ambient (homogeneous and/or heterogeneous) interference for the wireless connectivity technologies at 2.4 GHz unlicensed band, are addressed, respectively.

1) Wi-Fi Direct, now part of Android smartphones, makes it possible for Wi-Fi devices to communicate directly without passing through an Wi-Fi Access Point (AP). Before starting actual data exchange, two devices should apparently find each other through a device discovery process, called *find phase*. Wi-Fi Direct standard specifies that a device in find phase selects a *listen channel* among three *social channels* in 2.4 GHz unlicensed band, i.e., channels 1, 6, and 11, at the beginning, and periodically visits the selected listen channel to wait for searching messages, i.e., probe request frames from other devices. However, once it selects a social channel as the listen channel, it never changes the channel until the end of find phase such that it may suffer from severe performance degradation when it happens to work at a highly interfered channel. The device may face such situation unexpectedly since 2.4 GHz unlicensed band is open for public use and the networks at the band are typically unplanned.

2) Wi-Fi and BT, the two most widely used technologies at 2.4 GHz unlicensed band, share the same spectra but are built on top of entirely different PHY and MAC layer designs; Wi-Fi is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)-based static channel technology [1] while BT is Time Division Multiple Access (TDMA)-based frequency hopping technology [2]. How to make them coexist better has been a major technical issue [3]. So far, the severity of the heterogeneous interference between Wi-Fi and BT has been

largely mitigated by BT's Adaptive Frequency Hopping (AFH) capability, while the interference still remains intrusive in the following two cases. Firstly, when all the BT channels are covered by several ambient Wi-Fi networks bringing highly-loaded strong interference, AFH is hardly effective since there are no noticeable clues for desirable BT channels. Secondly, regarding the homogeneous interference among multiple BT devices, the respectively independent and pseudo-random properties of multiple BT devices' hopping patterns largely prevent packet collisions among them, while when only part of BT channels are occupied by one or more ambient Wi-Fi networks, AFH-enabled BT devices are supposed to share a narrower spectra untouched by the Wi-Fi networks, thus increasing packet collisions among them. It is a non-trivial problem in reality considering the fact that AFH is supported by most of the contemporary BT devices.

3) BLE has attracted enormous attention mainly for its usage in indoor localization system. There have been many papers dealing with the accuracy of BLE localization performance [4–9], while an often overlooked problem is the impact of the BLE packet loss on the performance of the localization system. Most BLE-based indoor localization systems utilize Received Signal Strength Indication (RSSI) of the received BLE packets to infer current location. When there are Wi-Fi networks in vicinity, the BLE-based indoor localization system tends to heavily degrade due to the lack of RSSI information caused by BLE packet losses. The lack of RSSI information often results in degradation of the localization performance, e.g., inaccurate location estimation, and delayed location tracking.

1.2 Overview of Existing Approaches

1.2.1 Faster Wi-Fi Direct Devicey Discovery

To the best of our knowledge, there has been no literature providing a comprehensive solution to deal with the delay encountered in the find phase. One of the earliest experimental researches on Wi-Fi Direct device discovery [10] provide several measurement statistics using Linux laptops and stated that find phase is a major source of the random lag in device discovery, while the authors do not analytically evaluate the performance. Besides, to reduce the overall group formation delay, in [11], the authors propose a new group formation algorithm, whereas they utilize the legacy find phase in device discovery.

Moreover, various Wi-Fi Direct-based applications have been proposed, e.g., for reducing the energy consumption of mobile devices [12, 13] and enabling the collaborative data transmissions among Wi-Fi Direct devices [14], which rely on the legacy find phase.

1.2.2 Better Coexistence Between Wi-Fi and BT

In [15], the performance degradations of Wi-Fi and BT in heterogeneous coexistence environment are analyzed without any performance enhancements. In [16–19], the performance degradations are mitigated via better spectra separation between Wi-Fi and BT. In [17, 18], time division approaches are proposed to mitigate the performance degradations. In [20], a coordination scheme for multiple BT piconets is proposed to control multiple piconets' hopping sequences by allocating multiple piconets an identical address and several delayed versions of clocks with different offsets to generate several delayed versions of hopping sequences, while non-zero channel differences among these delayed versions of hopping sequences are not guaranteed, thus still introducing collisions. The scheme in [21] divides 79 BT channels into several orthogonal sets, and lets multiple piconets randomly select a set to use as their used channels.

Apparently, the collision probability increases among the piconets selecting the same set, which is not suitable in dense environment.

1.2.3 Robust BLE-based Indoor Localization

There have been many papers dealing with the accuracy of BLE localization performance [4–9]. As a representative study, [8] presents the experimental results of the BLE-based indoor localization system utilizing RSSI fingerprinting method, demonstrating that it is practically feasible to construct a decent localization system based on BLE fingerprinting. However, the authors only show the susceptibility of the localization performance to the wireless channel imperfections, e.g., channel noise, fast fading, etc., and do not mention the impact of the BLE packet losses caused by ambient interference.

Besides, in [22], the authors emphasize the severity of the impact of ambient Wi-Fi interference on the performance of the indoor localization system constructed using Zigbee. They also pinpoint that the degradation in localization accuracy is mainly contributed by the loss of Zigbee packets rather than the variance of RSSI values. However, they reveal only the need of interference resilient indoor localization system without any performance enhancement.

1.3 Main Contributions

1.3.1 Faster Wi-Fi Direct Devicey Discovery

In order to tackle the problem caused by the static listen channel in legacy find phase, we propose a simple scheme referred to as *Listen Channel Randomization* (LCR). In the proposed scheme, a Wi-Fi Direct device is allowed to select an arbitrary social channel whenever it attempts to wait for other devices' probe requests. In this way, it can avoid highly interfered channel(s) statistically, thus mitigating the performance degradation incurred by the interfered channel(s).

We claim the following four major contributions in this research work.

- We have developed an elaborate absorbing Markov chain model in order to analyze the performances of LCR and the legacy find phase.
- We validate the accuracy of our Markov model by both simulations and empirical measurements.
- We propose LCR to expedite the device discovery procedure.
- We implement LCR in Galaxy Nexus phone, and empirically evaluate the performances of both LCR and the legacy find phase to demonstrate that LCR can reduce the device discovery delay by up to 72% in find phase.

1.3.2 Better Coexistence Between Wi-Fi and BT

In order to make BT and Wi-Fi coexist better in dense environment, we propose BlueCoDE, a coordination scheme for multiple nearby BT devices with the following features.

- It does not require the modifications of the BT's existing PHY and MAC designs.
- It improves Wi-Fi performance tremendously when all the BT channels covered by Wi-Fi signals by making the hopping patterns of multiple BT devices less harmful to Wi-Fi.
- It reduces collisions among multiple BT devices with almost zero collision probability.

In BlueCoDE, a device referred to as *coordinator* controls multiple nearby BT devices in terms of their native clocks and the device addresses, since we found out that the hopping pattern of a BT device can be deliberately manipulated by controlling its address and clock. This feature is exploited by BlueCoDE to make multiple BT devices almost collision-free and less harmful to Wi-Fi.

In summary, we claim the following four contributions in this work.

- We find out a simple way to manipulate hopping sequences of multiple BT devices to make them collision-free.
- We develop a method to make multiple BT devices less harmful to Wi-Fi via hopping sequence manipulation, and identify the rationale analytically.
- We propose BlueCoDE as a general framework to exploit aforementioned method.
- We corroborate the performance gains delivered by BlueCoDE via simulation and prototype-based experiments.

1.3.3 Robust BLE Packet Detection

In order to make BLE packet detection more interference-resilient, we propose RESCUE, a Carrier Frequency Offset (CFO)-based BLE packet detection scheme, where the uncorrupted part of partially corrupted BLE packet is detected and utilized to retrieve decent RSSI information. The packet detection is further processed to determine the transmitter of the packet based on the timing information of the detection.

In summary, we claim the following four contributions in this work.

- We find out that BLE packets can be detected by careful CFO inspections even when the packets are partially corrupted due to ambient interference.
- We propose RESCUE as a framework to utilize the CFO-based packet detection scheme and identify the detected packets by utilizing timing information of the detection.
- We corroborate the performance gain delivered by RESCUE via prototype-based indoor localization experiments.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows.

Chapter 2 presents an absorbing Markov chain-based statistical model, based on which we can analyze the stochastic behavior of the two Wi-Fi Direct devices in the legacy find phase. The effectiveness of the Markov model is validated via both NS-3 simulation and real device-based measurement. Then, LCR is proposed to randomize the listen channel during find phase to exploit frequency diversity, of which the significant performance gain is verified with absorbing Markov chain model and prototype-based experiments.

Chapter 3 presents how BlueCoDE is designed and why it can improve the coexistence performance substantially in dense Wi-Fi and BT coexistence environments. The performance gains of BlueCODE compared with the legacy scheme are corroborated via statistical analysis, NS-3 simulation, and prototype-based experiments.

Chapter 4 presents how RESCUE is designed and why it can detect BLE packets even when the packets are partially corrupted. The rationale behind the CFO-based BLE packet detection method is elaborated. The effectiveness of RESCUE in terms of packet detection capability and the robust RSSI acquisition effect in localization system are demonstrated via prototype-based experiments.

Finally, Chapter 5 concludes the dissertation with the summary of contributions and discussion on the future work.

Chapter 2

Faster Wi-Fi DirectDevice Discovery

2.1 Introduction

Wi-Fi Direct is the Wi-Fi based Peer-to-Peer (P2P) technology enabling direct wireless communications between Wi-Fi devices without passing through an Access Point (AP). It was introduced to enhance the user experiences especially for multimedia sharing. Now, it becomes a major built-in feature of smartphones, cameras, printers, PCs, displays, and gaming devices [23]. Accordingly, it is employed for various emerging industry-wide solutions as a key enabling technology. For example, Miracast [24] as a mirroring service allows a mobile device to duplicate its screen to an external wider display seamlessly via Wi-Fi Direct link.

Wi-Fi Direct inherits the concept of *P2P group* from the traditional Wi-Fi network architecture. In other words, the P2P group is functionally equivalent to the traditional Basic Service Set (BSS) of the Wi-Fi network architecture. In a P2P group, a Wi-Fi Direct device referred to as *P2P Group Owner* (P2P-GO) behaves like an AP and other Wi-Fi Direct device(s) takes the role of normal Wi-Fi station(s).

According to Wi-Fi Direct standard, P2P group formation procedure[1] consists of

[1] The autonomous and persistent P2P group formations are outside the scope of this work, which are optional and seldom-used features in current Wi-Fi Direct devices.

five steps, i.e., (1) device discovery, (2) service discovery, (3) GO negotiation, (4) Wi-Fi Protected Setup (WPS) provisioning, and (5) IP address configuration. Empirical measurement results in [10] show that device discovery accounts for the largest portion of the overall time required to complete P2P group formation procedure. From the actual measurement results, we can surprisingly notice that the device discovery may take over ten seconds in the worst case. It implies that reducing the device discovery time is the most critical issue for satisfactory service startup.

As detailed later, Wi-Fi Direct device discovery is divided into two phases, i.e., *802.11 scanning* and *find phase*. Actually, tremendous studies have been carried out to optimize 802.11 scanning so far since the scanning delay is a timeworn and well-known issue [25–27]. In contrast, we can find more rooms for improvement of the legacy find phase since it has not been sufficiently investigated yet due to the recent advent of Wi-Fi Direct technology. For this reason, we focus on the issue regarding how to reduce the delay for the legacy find phase.

Wi-Fi Direct standard specifies that a device in the legacy find phase selects a channel among three *social channels* in 2.4 GHz frequency band, i.e., channels 1, 6, and 11, at the beginning and periodically visits the selected channel to wait for searching messages, i.e., *probe request* frames from other devices. However, once it selects a social channel, it never changes the channel by the end of the legacy find phase such that it may suffer a severe performance degradation when it happens to work at a highly interfered channel. The device may face such situation unexpectedly since 2.4 GHz frequency band is open for public use and the networks on the band are typically unplanned.

In order to tackle the situation, we propose a simple scheme referred to as *Listen Channel Randomization* (LCR). In the proposed scheme, a Wi-Fi Direct device is allowed to select an arbitrary social channel whenever it attempts to wait for other devices' probe requests. In this way, it can avoid highly interfered channel(s) statistically, thus mitigating the performance degradation incurred by the interfered channel(s).

In this work, we claim the following four major contributions.

- We have developed an elaborate Markov model in order to analyze the performances of the proposed scheme and the legacy find phase.
- We validate the accuracy of our Markov model by both simulations and empirical measurements.
- We propose LCR to expedite the device discovery procedure.
- We implement LCR in Galaxy Nexus phone, and empirically evaluate the performances of both LCR and the legacy find phase to demonstrate that LCR can reduce the device discovery delay up to 72% in find phase.

The rest of this chapter is organized as follows: in Section 2.2, we provide preliminary knowledge and assumptions used in this work. In Section 2.3, we build an accurate Markov model for solid numerical analysis. In Section 2.4, our Markov model is validated. Thereafter, we propose LCR and evaluate its performance in Section 2.5. We give a brief summary of this chapter in Section 2.5.4.

2.2 Preliminary

2.2.1 Wi-Fi Direct Device Discovery

Wi-Fi Direct device discovery consists of 802.11 scanning and find phase. Specifically, a Wi-Fi Direct device can first conduct 802.11 scanning defined in 802.11 standard [1] in order to identify surrounding P2P groups. If it succeeds in finding an appropriate P2P-GO, it joins the group, otherwise it begins a find phase to discover other Wi-Fi Direct devices working in find phase. In find phase, the device alternates between *listen* and *search* states [28].

Search state: A device in search state conducts active scanning sequentially at three social channels by transmitting a probe request frame and staying at each social channel for a *dwell time*, expecting to receive *probe response* frames.

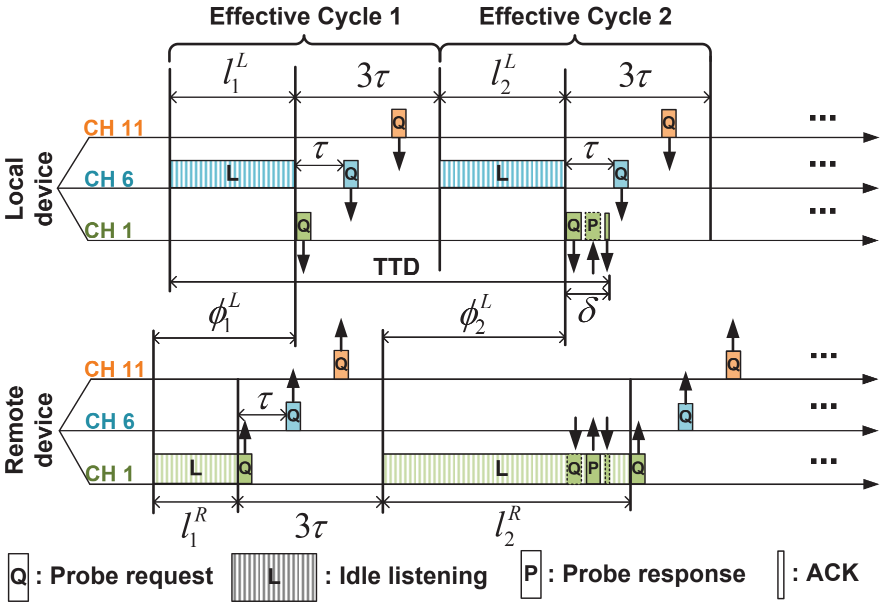


Figure 2.1: Illustration of find phase.

Listen state: In listen state, on the contrary, the device conducts idle listening at *listen channel* for a random duration, during which it replies to a probe request frame by transmitting a probe response frame. *The listen channel is randomly selected among social channels at the beginning of the legacy find phase and remains fixed until the end of the legacy find phase*. The duration of each listen state is a random number being an integer multiple of 100 Time Units (TUs), i.e., 102.4 ms.

2.2.2 Assumption

We assume that only a single probe request frame is transmitted per social channel in order of increasing channel number, and the dwell time at each social channel in search state is a constant value τ,[2] which are the typical cases according to real device measurement. Besides, the listen duration in each listen state is assumed to be selected among $\{100, 200, 300\}$ TUs with probability $\frac{1}{3}$, which is the typical case de-

[2] The detailed operations in search state are not specified in [28] and remain as implementation issues.

fined in [28].

Without loss of generality, during the analysis of the legacy find phase presented in Section 2.3, in order to facilitate explanation, we assume a remote device discovered by a local device always selects channel 1 as its listen channel.

2.2.3 Performance Metric

Fig. 2.1 illustrates the scenario that two Wi-Fi Direct devices attempt to find each other based on the legacy find phase, highlighting the performance metric of interest, i.e., *Time To Discovery* (TTD). It is measured from the start of the first *effective cycle* of the local device[3] to the moment when it finds the remote device based on the three-way handshake, i.e., sending a probe request, receiving a probe response, and replying with an acknowledgement (ACK) frame for confirmation. We define a listen state and the subsequent search state in combination as a *cycle*. A cycle of the local device is called an *effective cycle* if the probe request frame transmitted at the remote device's listen channel during the cycle overlaps with the remote device's find phase.

2.3 Analytical Modeling for the Legacy Find Phase

We propose an absorbing Markov chain-based statistical model to analyze the legacy find phase-based device discovery procedure. Apparently, entering an absorbing state means that the local device finds the remote device.

2.3.1 Proof of Markov Property

We first prove that the find phase operation satisfies Markov (memoryless) property. As indicated in Fig. 2.1, the local device can find the remote device as long as it transmits a probe request frame at the remote device's listen channel just when the remote device

[3] We designate one of the two devices here as a local device, from whose perspective, TTD is calculated.

is in listen state and the residual listening time is greater than a certain threshold, δ, which indicates an adequate time for the two devices to handshake.

Let ϕ_n^L indicates the time difference between the starting time of the search state of the local device's nth effective cycle and the starting time of the remote device's most recent preceding listen state from the local device's perspective. If we denote the duration of the remote device's listen state as l_n^R, then

$$I_n = \begin{cases} 1, \ \phi_n^L \leq l_n^R - \delta, \\ \\ 0, \ \text{otherwise}, \end{cases} \tag{2.1}$$

is an indicator function, indicating whether the local device can find the remote device in the nth effective cycle. Then, if we denote the duration of the local device's listen state in the $(n+1)$th cycle as l_{n+1}^L, considering

$$\phi_{n+1}^L = \phi_n^L + (l_{n+1}^L - l_n^R), \tag{2.2}$$

I_{n+1} becomes

$$I_{n+1} = \begin{cases} 1, \ \phi_n^L + (l_{n+1}^L - l_n^R) \leq l_{n+1}^R - \delta, \\ \\ 0, \ \text{otherwise}. \end{cases} \tag{2.3}$$

Based on the fact that the duration of the listen state is an i.i.d. uniform random variable with the *probability mass function* (pmf)

$$\mathrm{P}(l = x) = \frac{1}{3}, \quad x = 100, 200, 300, \tag{2.4}$$

we can argue that the value of I_{n+1} depends only on ϕ_n^L, l_{n+1}^L, l_n^R, and l_{n+1}^R, and is conditionally independent of I_{n-1}, I_{n-2},..., I_1. Therefore, if the local device is in the nth cycle currently, whether it can find the remote device in the next cycle depends only on the status of the current cycle, reflecting the Markov property.

2.3.2 Basic Model

We start with a *basic model*, assuming ideal wireless channel environment, i.e., error-free and "clean" wireless environment without nearby contending devices except the local and the remote devices, after which a *generalized model* is developed to cope with more realistic wireless environments in the next subsection.

State Identification in Basic Model

Note that the valid range of ϕ^L is $(0, l^R + 3\tau]$, since it is always calculated using the remote device's most recent preceding listen state from the local device's perspective. Therefore, the condition for the local device being able to find the remote device can be expressed as

$$0 < \phi^L \leq l^R - \delta, \tag{2.5}$$

where

$$\delta = 2T_{\mathrm{DIFS}} + 2T_{\mathrm{SLOT}}N_{\mathrm{CW}} + T_{\mathrm{REQ}} + T_{\mathrm{RSP}} + T_{\mathrm{SIFS}} + T_{\mathrm{ACK}}, \tag{2.6}$$

indicating the minimum time needed for handshaking in ideal wireless channel environment. In (2.6), T_{SIFS}, T_{DIFS}, T_{SLOT}, and N_{CW} represent Short Inter Frame Space (SIFS), Distributed IFS (DIFS), slot time, and the random backoff counter value. T_{REQ}, T_{RSP}, and T_{ACK} represent the frame durations of probe request, probe response, and ACK, respectively.

Apparently, if we think of (2.5) as an absorbing condition for a Markov chain, the transient condition becomes

$$l^R - \delta < \phi^L \leq l^R + 3\tau. \tag{2.7}$$

Inspired by (2.5) and (2.7), we define a state of Markov chain using an ordered pair

$$\mathrm{s}_{(l,\phi)} \triangleq \{(l, \phi) | l^R = l,\ \phi^L \in (\phi, \phi + \Delta_\phi]\,\}, \tag{2.8}$$

for $l \in \{100,\ 200,\ 300\}$ and $\phi \in [0, l^R{+}3\tau{-}\Delta_\phi]$, where Δ_ϕ is defined as $\gcd(100,\ \delta,\ 3\tau)$.

The rationale behind the interval Δ_ϕ in (2.8) is that since the time difference ϕ^L can be any value among $(0, l^R + 3\tau]$, if we define a state with a specific value, there will be continuous state space and infinite number of states, which is unfavourable to straightforward statistical analysis. Hence, we divide the entire range, $(0, l^R+3\tau]$, into a finite number of intervals with a constant duration, Δ_ϕ, and map each state to each interval. Note that the interval should be as large as possible to reduce the total number of states so that the complexity incurred in TTD calculation is minimized. At the same time, its value should be small enough to ensure that all the ϕ^L's contained in a state have a common transition characteristic. We set Δ_ϕ as the $\gcd$ of 100 (the $\gcd$ of the ϕ^L's possible variations in a state transition, i.e., 0, ±100, and ±200 derived from (2.2)), δ, and 3τ to ensure that there are integer numbers of absorbing and transient states, among which the state transitions are performed in one-to-one correspondence. Note that the state is defined with respect to each effective cycle of the local device, whose transition occurs every cycle transition of the local device in time domain.

State Transition in Basic Model

A state $\mathrm{s}_{(l,\phi)}$ is defined as an absorbing state if it satisfies $\phi + \Delta_\phi \leq l - \delta$; otherwise, a transient state. Because the transition characteristic of the absorbing state is apparent, i.e., with probability 1 to themselves, we should figure out the transitions with respect to the transient states.

Firstly, if we assume the chain is in transient state $s_{(l_n,\phi_n)}$ in the nth effective cycle, the state in the $(n+1)$th effective cycle can be denoted as $s_{(l^R_{n+1},\phi_n+(l^L_{n+1}-l_n))}$ inferred from (2.2).

Type ①: If $s_{(l^R_{n+1},\phi_n+(l^L_{n+1}-l_n))}$ is a valid state, i.e., $\phi_n + (l^L_{n+1} - l_n)$ belongs to the valid range $(0, l^R_{n+1}+3\tau]$, we refer to this type of transition as Type ① transition, which results in 9 different cases: $s_{(100,\phi_n+(100-l_n))}$, $s_{(200,\phi_n+(100-l_n))}$, $s_{(300,\phi_n+(100-l_n))}$, $s_{(100,\phi_n+(200-l_n))}$, $s_{(200,\phi_n+(200-l_n))}$, $s_{(300,\phi_n+(200-l_n))}$, $s_{(100,\phi_n+(300-l_n))}$, $s_{(200,\phi_n+(300-l_n))}$, $s_{(300,\phi_n+(300-l_n))}$, induced by all the combinations of l^R_{n+1} and l^L_{n+1}, each with probability $\frac{1}{9}$ due to (2.4) and the mutual independence between l^R_{n+1} and l^L_{n+1}.

Type ②: However, $s_{(l^R_{n+1},\phi_n+(l^L_{n+1}-l_n))}$ can be an invalid state. Considering $s_{(l_n,\phi_n)}$

is a transient state so that (2.7) holds between l_n and ϕ_n, we know that

$$l^L_{n+1} - \delta < \phi_n + (l^L_{n+1} - l_n) \leq l^L_{n+1} + 3\tau. \tag{2.9}$$

Typically, in the ideal environment, δ is less than 100 TUs, and hence, the lower bound is greater than 0. It means that we just need to consider the invalid cases that $\phi_n + (l^L_{n+1} - l_n)$, which is less than $l^L_{n+1} + 3\tau$, is greater than $l^R_{n+1} + 3\tau$ here. The advent of an invalid state can be interpreted as the advent of *intermediate* state as illustrated in Fig. 2.2. Intuitively, $\phi_n + (l^L_{n+1} - l_n) > l^R_{n+1} + 3\tau$ means that intermediate state is skipped by the process since we always calculate the time difference with the most recent preceding listen state. Hence, we refine the $(n + 1)$th state $s_{(l_{n+1}, \phi_{n+1})}$, such that l_{n+1} and ϕ_{n+1} become l^R_{n+2} and $\phi_n + (l^L_{n+1} - l_n) - (l^R_{n+1} + 3\tau)$, respectively, as shown in Fig. 2.2. We refer to the transition involving one intermediate state as Type ② transition, where

$$\begin{aligned}
\mathrm{P}_{(l_n,\phi_n)(l_{n+1},\phi_{n+1})} &= p(l_{n+1},\, \phi_{n+1} | l_n,\, \phi_n) \\
&= p(l^R_{n+2},\, \phi_n + (l^L_{n+1} - l_n) \\
&\quad - (l^R_{n+1} + 3\tau) | l_n,\, \phi_n) \\
&= p(l^R_{n+1},\, l^R_{n+2},\, l^L_{n+1}) \\
&= p(l^R_{n+1}) p(l^R_{n+2}) p(l^L_{n+1}) = \frac{1}{27}
\end{aligned} \tag{2.10}$$

is the associated transition probability.

Type ③: If we consider the relationship between the minimum duration of the intermediate state, i.e., $100 + 3\tau$, and the maximum duration of $\phi_n + (l^L_{n+1} - l_n)$, i.e., $300 + 3\tau$ implied in (2.9), the maximum number of intermediate states possibly skipped by a single state transition is 2, since $2(100 + 3\tau) < 300 + 3\tau < 3(100 + 3\tau)$, when $3\tau < 100$. We refer to such state transition involving two consecutive intermediate states as Type ③ transition. In this case, l_{n+1} and ϕ_{n+1} of the $(n + 1)$th state $s_{(l_{n+1}, \phi_{n+1})}$ become l^R_{n+3} and $\phi_n + (l^L_{n+1} - l_n) - (l^R_{n+1} + 3\tau) - (l^R_{n+2} + 3\tau)$,

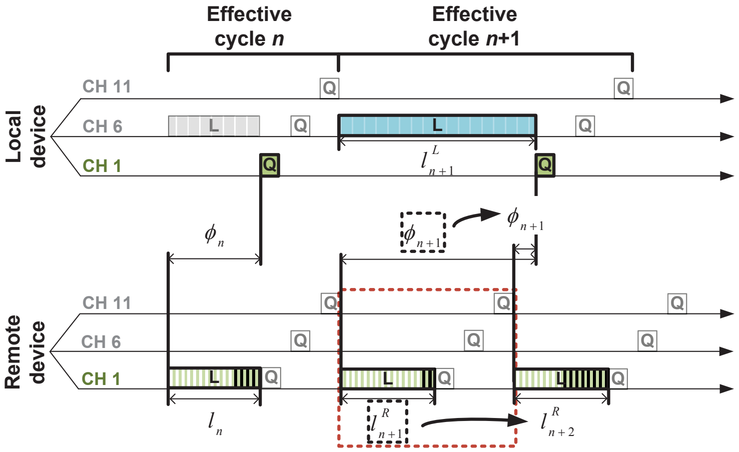


Figure 2.2: Intermediate state illustration.

respectively. The corresponding transition probability is

$$
\begin{aligned}
\mathrm{P}_{(l_n,\phi_n)(l_{n+1},\phi_{n+1})} &= p(l_{n+1},\, \phi_{n+1} | l_n,\, \phi_n) \\
&= p(l^R_{n+3},\, \phi_n + (l^L_{n+1} - l_n) \\
&\quad - (l^R_{n+1} + 3\tau) - (l^R_{n+2} + 3\tau) | l_n,\, \phi_n) \\
&= p(l^R_{n+1},\, l^R_{n+2},\, l^R_{n+3},\, l^L_{n+1}) \\
&= p(l^R_{n+1}) p(l^R_{n+2}) p(l^R_{n+3}) p(l^L_{n+1}) = \frac{1}{81}. \qquad (2.11)
\end{aligned}
$$

Fig. 2.3 illustrates the state distribution. The aforementioned three types of state transitions are emphasized using arrows marked with ①, ②, and ③, when current state is $s_{(100,100+3\tau-\Delta_\phi)}$.

The transition matrix is constructed by thoroughly examining the aforementioned three types of transitions and the corresponding transition probabilities for each transient state.

2.3.3 General Model

In general model, we introduce *busy medium preemption*, i.e., the channel access delay caused by busy channel status, and the transmission error caused by channel noise

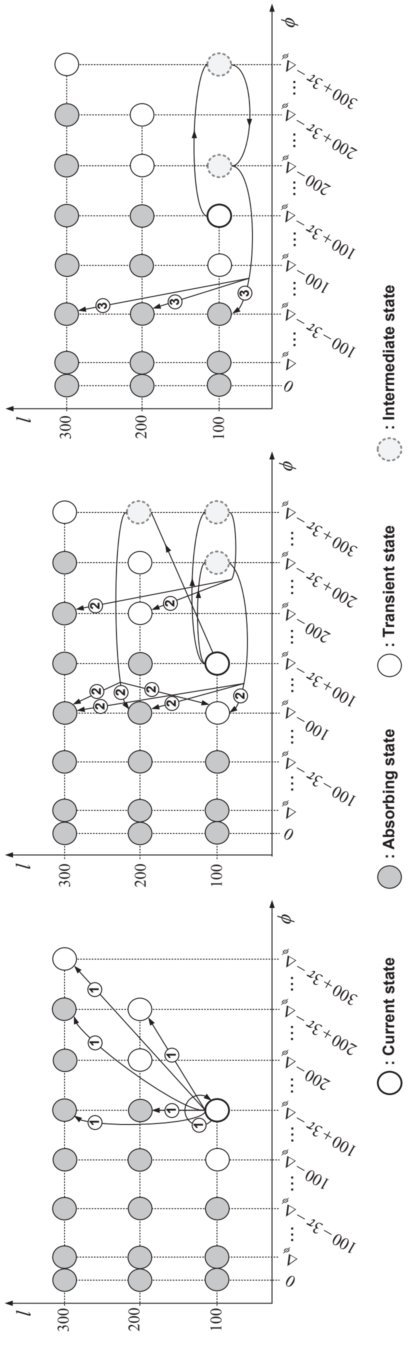


Figure 2.3: Illustration of the state transitions in basic model.

and/or co-channel interference.

State Identification in General Model

In general model, there is no such a threshold δ that can guarantee a successful handshake. Thus, the absorbing states previously defined in the basic model should be regarded as kind of transient states. In order to differentiate such "likely absorbing" transient states with the "pure" transient states used in the basic model, we call them *candidate* states. Besides, a *virtual absorbing* state is defined additionally to indicate a successful handshake. That is, when the local device in candidate state completes a handshake with the remote device, we say a virtual absorbing state is reached. Herein, although candidate states belong to transient states, we refer to only the "pure" transient state as transient state for convenience.

State Transition in General Model

The absorbing probability of a candidate state depends on the amount of the available time for handshake provided by the state. δ_m denotes the time consumed by a single probe request, m probe responses,[4] and an ACK, which is calculated as

$$\begin{aligned}\delta_m = (m+1)(T_{\mathrm{DIFS}} + \mathrm{E}[T_{\mathrm{BUSY}}]) &+ T_{\mathrm{SLOT}}\mathrm{E}[N_{\mathrm{CW}}^{(1)}] \\ + \sum_{i=1}^{m} T_{\mathrm{SLOT}}\mathrm{E}[N_{\mathrm{CW}}^{(i)}] &+ T_{\mathrm{REQ}} + mT_{\mathrm{RSP}} \\ +(m-1)T_{\mathrm{ATO}} &+ T_{\mathrm{SIFS}} + T_{\mathrm{ACK}}, \end{aligned} \tag{2.12}$$

where T_{ATO} indicates the ACK timeout value [1], and $N_{\mathrm{CW}}^{(i)}$ represents the random backoff counter value for the ith transmission attempt. Note that in (2.12), the duration of the busy medium preemption, T_{BUSY}, is taken as its expectation, which is generally a random value depending on the surrounding. If a certain candidate state, $s_{(l,\phi)}$,

[4] Retransmission is enabled for the probe response frame since it is a unicast frame, while there is no retransmission for the probe request frame, which is a broadcast frame.

satisfies

$$\delta_m \leq \min(\tau,\, l-\phi) < \delta_{m+1}, \tag{2.13}$$

meaning that the available time the candidate state provides for handshake can accommodate at most m probe response frames, the corresponding absorbing probability becomes

$$p_m = 1 - (f_{\text{REQ}} + (1 - f_{\text{REQ}}){f_{\text{RSP}}}^m), \tag{2.14}$$

where f_{REQ} and f_{RSP} denote the frame error rates (FERs) of probe request and probe response frames, respectively.

Note that the interval of the time difference, ϕ^L, covered by a single state, Δ_ϕ, is redefined here as $\gcd(100,\, 3\tau,\, \delta_1,\, \delta_2, ...,\, \delta_M)$, where M is the maximum possible number of (re)transmitted probe response frames, which is determined by τ and the retransmission limit.

Type ④: In the general model, the current state $s_{(l_n,\phi_n)}$ can be either transient state or candidate state. It means that besides the transitions from the transient states elaborated in the basic model, we should also deal with the transitions from the candidate states. It means the range of ϕ_n here becomes $(0,\, l_n + 3\tau]$, and hence, the range of $\phi_n + (l^L_{n+1} - l_n)$ becomes $(l^L_{n+1} - l_n,\, l^L_{n+1} + 3\tau]$. Therefore, if l^L_{n+1} is less than l_n, the lower bound is a negative number, thus resulting in the advent of another type of invalid state referred to as *negative* state. Fig. 2.4 illustrates how negative state is induced and refined to become valid. As shown in Fig. 2.4, negative state appears when two consecutive search states of the local device overlaps with the same cycle of the remote device. Accordingly, the nth and the $(n+1)$th search states should be compared with the same listen state of the remote device. In this manner, the first and the second entries of the $(n+1)$th state $s_{(l_{n+1},\phi_{n+1})}$ are modified to l_n and $\phi_n + (l^L_{n+1} - l_n) + l_n + 3\tau$, respectively. We refer to the state transition involving negative state as Type ④ transi-

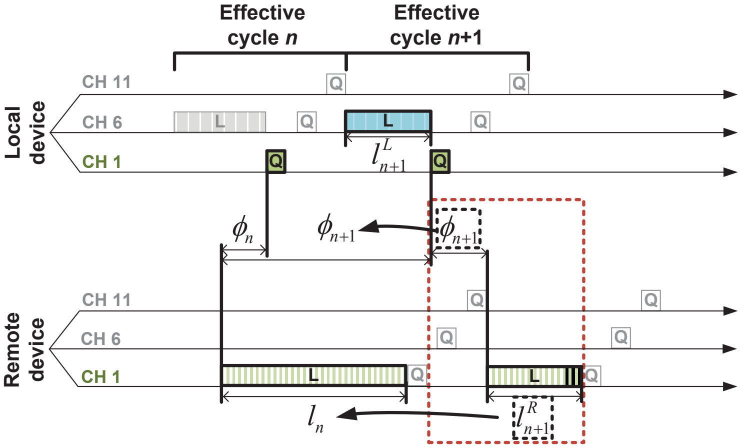


Figure 2.4: Negative state illustration.

tion. The corresponding transition probability is

$$\begin{aligned}
\mathrm{P}_{(l_n,\phi_n)(l_{n+1},\phi_{n+1})} &= p(l_{n+1},\, \phi_{n+1}|l_n,\, \phi_n) \\
&= p(l_n,\, \phi_n + (l_{n+1}^{L} - l_n) \\
&\quad + l_n + 3\tau|l_n,\, \phi_n) \\
&= p(l_{n+1}^{L}) = \frac{1}{3}.
\end{aligned} \tag{2.15}$$

Note that in the basic model, we do not encounter the negative state since it is only yielded from candidate states.

The transition matrix of the general model is similar to that of the basic model except that Type ④ transition and the absorbing probabilities of candidate states are additionally introduced.

2.3.4 TTD Calculation

After constructing the transition matrix, we renumber the states[5] so that transient and candidate states come first, to make the transition matrix in *canonical form*, from which

[5] Any renumbering rules are acceptable as long as the resulting transition matrices are in canonical form.

the *fundamental matrix*, $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-\mathbf{1}}$, can be obtained directly [29]. Then, each entry of the vector $\mathbf{v} = \boldsymbol{\pi}\mathbf{N}$ represents the expected times the process will visit each (transient or candidate) state before the absorption given the initial state distribution row vector $\boldsymbol{\pi}$.

After that, intuitively, we can multiply each entry of $\mathbf{v}$ with the corresponding *sojourn time*—the duration that a state will last once entered—of each state $\mathrm{s}_{(l,\phi)}$, i.e., $l + 3\tau$, and aggregate them to obtain TTD. However, TTD calculated in this way is not accurate. Note that l represents remote device's listen duration, and we cannot identify the omitted invalid states if TTD is calculated from the remote device's perspective.

Hence, TTD should be calculated from the local device's perspective. It becomes more evident in Figs. 2.2 and 2.4 that although there are invalid states involved, the process can be tracked continuously by sequentially aggregating the effective cycles of the local device, i.e., $\sum\limits_{i}(l_i^L + 3\tau)$. Then, the ij-entry e_{ij} of the matrix $\mathbf{E} = \mathbf{DQ}$, where $\mathbf{D}$ is a diagonal matrix whose diagonal entry d_{ii} is the ith entry v_i of vector $\mathbf{v}$, represents the expected number of occurrences of state i[6] to state j transition before absorption. TTD can be obtained by $\sum\limits_{i}\sum\limits_{j} e_{ij}c_{ij}$, where c_{ij} is the ij-entry of matrix $\mathbf{C}$, corresponding to the duration of local device's effective cycle, $l^L + 3\tau$, associated with the state i to state j transition. Matrix $\mathbf{C}$ can be constructed along with the construction of the transition matrix by recording the value of $l^L + 3\tau$ associated with each possible state transition.

2.4 Model Validation

2.4.1 Simulation

In order to validate the effectiveness of the proposed model, we implement the legacy find phase in ns-3 simulator [30], and conduct simulations under various effects of channel conditions and different values of dwell time τ. We set the length of both

[6] State i corresponds to the ith state after state renumbering.

probe request and probe response frames to 250 bytes, where both frames are transmitted with IEEE 802.11g 6 Mb/s Physical (PHY) rate [28]. The maximum retransmission limit is set to 7. The busy medium preemption, T_{BUSY}, is assumed from 28 μs, representing the Null Data Packet (NDP) duration [1], to 2,166 μs, representing the time to transmit a 1,536-byte MAC layer Protocol Data Unit (MPDU) (corresponding to 1,500-byte Ethernet Maximum Transmit Unit (MTU) as payload) with IEEE 802.11g 6 Mb/s PHY rate followed by T_{SIFS}, T_{ACK}, and T_{DIFS}. We consider a uniformly distributed T_{BUSY} with the aforementioned upper and lower bounds as a representative case. That is, we simulate the scenario where there exists an extra contending Wi-Fi device ceaselessly attempting to transmit frames with uniformly distributed frame lengths, causing the busy medium preemption approximately with probability $\frac{1}{2}$ due to the nature of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism.

Fig. 2.5 shows the comparison between the simulation (indicated as ‘sim’) and the analysis (indicated as ‘ana’) obtained by utilizing the proposed Markov chain-based model. We here consider three scenarios: (1) three social channels are ideal channels, (2) the three channels induce FERs of 0.3, and (3) the three channels induce FERs of 0.5. In the ideal channel environment, simulation and analysis results are perfectly matched with nearly 0% difference. Besides, as we can expect, TTD increases along with the increase of τ, since the extra dwell time beyond the minimum essential time, δ (about 1 ms), has no effect on device discovery due to the ideal channel assumption, thus resulting in longer cycle and delaying the device discovery.

However, when the busy medium preemption and the frame errors are introduced, the tradeoff starts appearing since longer τ enables more (re)transmissions of probe response and can boost the robustness, while, as mentioned above, it lengthens the duration of the cycle and inevitably incurs extra delay. Fig. 2.5 unfolds the tradeoff relationship as we can observe that the minimum TTD is driven by τ equal to about 5 ms. Note that the results obtained by uniformly distributed T_{BUSY} coincides with

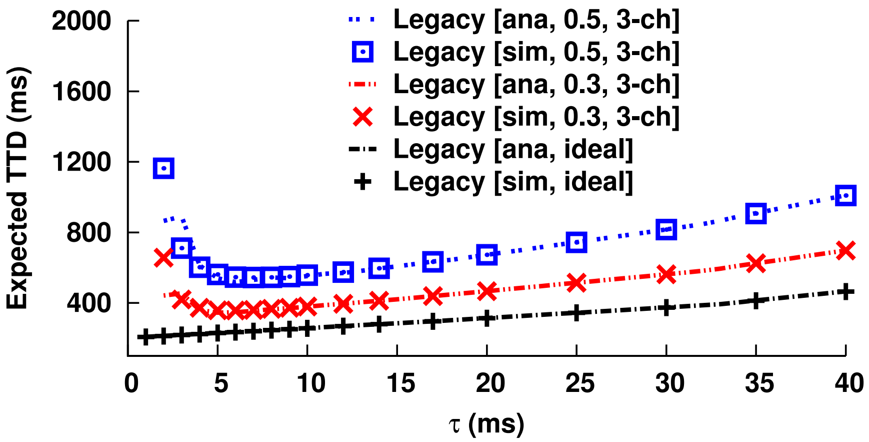


Figure 2.5: Simulation-based model validation for three cases that f_{REQ} and f_{RSP} are both set to 0 (ideal), 0.3, and 0.5, respectively.

the analysis very well although gaps can be observed in the cases of relatively short τ. It can be explained by rethinking the calculation of the absorbing probability of a candidate state in (2.14), which necessitates the premise that if a candidate state satisfies (2.13), it can always accommodate up to m probe response (re)transmissions, which is not always the case if T_{BUSY} is a random variable. Even so, we can see that in the most cases the proposed model estimates TTD with high accuracy.

2.4.2 Experiments

We also conduct real device-based measurement study using two Galaxy Nexus smartphones in a relatively clean wireless environment to practically validate the proposed model. Two aspects need to be additionally considered for modeling the legacy find phase in real devices.

(i) The overall operation of the legacy find phase is implemented as a user space application as part of *wpa supplicant* in Android smartphone, while the basic listening and searching functions are kernel space Wi-Fi Network Interface Card (NIC) driver's tasks. We observe that there are random internal delays incurred when the device performs state changes between listen and search states due mainly to the hardware imperfection and the internal network stack's latency. We confirm it by comparing the timing information of the state changes recorded in wpa supplicant's log file and that of the packets captured over the air. We collect a large number of internal delay samples and calculate the average T_{DELAY}, which is about 100 ms for each state change in the case of Galaxy Nexus. We take into account the internal delays in TTD calculation by introducing extra $\frac{2T_{\text{DELAY}}}{\Delta_\phi}$ transient states to the general model.

(ii) We found that the duration of the listen state is selected among $\{200, 300\}$ TUs each with probability $\frac{1}{2}$ in Android 4.2.2 used in Galaxy Nexus. Therefore, we should eliminate the states, whose first entry l is 100 TUs, and the ϕ's possible variations during state transition become 0 and ± 100. Besides, the transition probabilities of Type ①, Type ②, and Type ④ transitions become $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{2}$, respectively. Moreover, the maximum number of intermediate state skipped during a single state transition is 1, since

$$\begin{aligned} 200 + 3\tau + 2T_{\text{DELAY}} &< 300 + 3\tau + 2T_{\text{DELAY}} \\ &< 2(200 + 3\tau + 2T_{\text{DELAY}}). \end{aligned} \tag{2.16}$$

That is, Type ③ transition is out of scope.

We measure TTD with different values of τ ranging from 10 ms to 60 ms by modifying bcmdhd driver code included in kernel source, and collect about 10,000 TTD samples for each τ. Fig. 2.6 compares the average of the TTD samples obtained from the experiments (indicated as 'expr') and the analysis results. The largest gap is about 5.2% corresponding to the case that τ is 20 ms, and the average gap is 1.9%. We argue that the analysis is quite close to the reality even when the internal delay generates extra randomness.

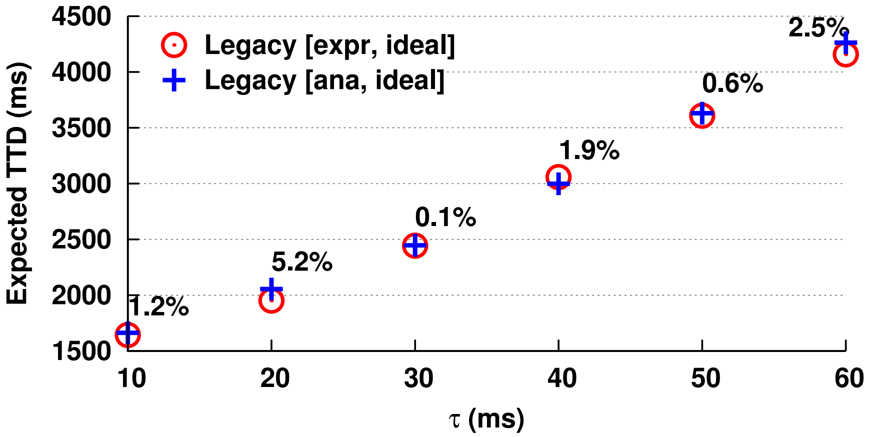


Figure 2.6: Model validation for the legacy find phase by experiments using two Galaxy Nexus phones.

2.5 Listen Channel Randomization

As explained in Section 2.2.1, listen channel is of significant importance since a device transmits probe response frames to other surrounding devices at the listen channel to inform its existence. In the legacy find phase, the listen channel is selected randomly at the beginning and remain fixed during the entire find phase. Hence, if the selected listen channel is a highly interfered channel, the device discovery will suffer tremendous disturbance.

Therefore, we propose *Listen Channel Randomization* (LCR), in which the listen channel is randomized among three social channels in each listen state to exploit the channel diversity gain. We define each social channel is selected with probability $\frac{1}{3}$ in LCR as a baseline approach.

2.5.1 Markov Chain Construction

The stochastic behavior of LCR can be modeled utilizing the proposed Markov chain-based model applying following modifications.

(i) First, we individually construct three Markov chains used to analyze the legacy find phase based on the channel status of the three social channels, respectively.

(ii) Then, the state indices of the Markov chains corresponding to channels 6 and 11 are cyclically shifted forward by $\frac{\tau}{\Delta_\phi}$ and $\frac{2\tau}{\Delta_\phi}$, respectively, to reflect the fact that, at each cycle, the time difference ϕ's of channels 6 and 11 are always larger than that of channel 1 by τ and 2τ, respectively, as indicated in Fig. 2.1.

(iii) Last, we construct a new *synthesized* Markov chain to analyze LCR, whose transition matrix is similar to that of the channel 1's Markov chain except that the absorbing probability of each candidate state is the average of that of the three pre-defined Markov chains.

2.5.2 Evaluation

The performance of LCR is evaluated by ns-3 simulations and prototype implementation-based experiments.

Simulation-Based Evaluation

We implement LCR in ns-3 simulator and simulate the following three scenarios, namely, (1) three social channels are all ideal channels (indicated as 'ideal'), (2) one of the social channels induces FER of 0.3 and the same uniformly distributed busy medium preemption adopted in Section 2.4.1 (indicated as '1-ch'), and (3) two of the social channels induce FERs of 0.5 and the same busy medium preemption as in the previous scenario (indicated as '2-ch'). We also assume ideal devices without internal delays since the measured delays are specific only to Galaxy Nexus phones.

Fig. 2.7 shows the simulation-based validation results in terms of the effectiveness of the synthesized Markov chain for LCR analysis. We observe that the simulation results coincide very well with the analysis results for most cases.

We then evaluate the performance of the legacy find phase and LCR by numerical

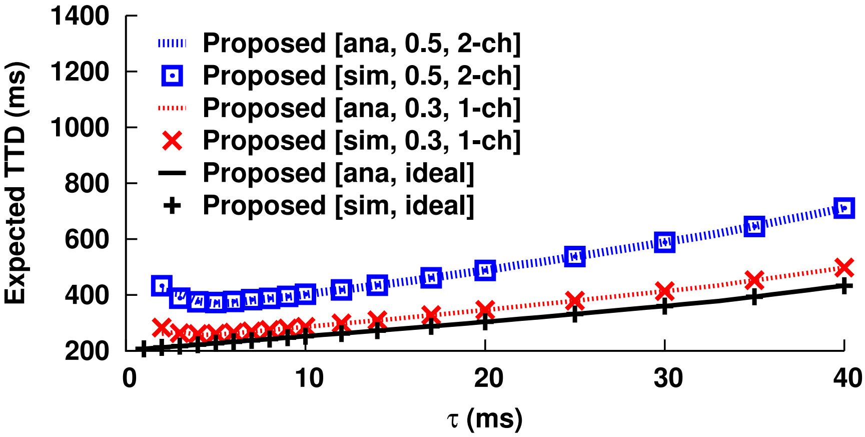


Figure 2.7: Model validation for LCR.

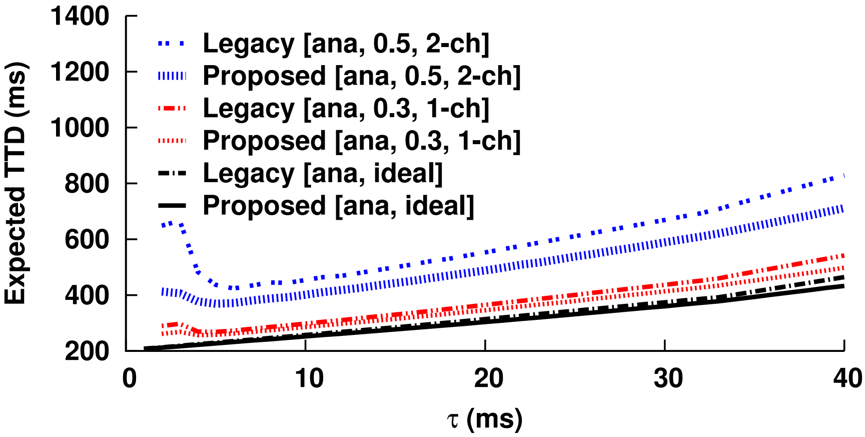


Figure 2.8: Performance comparison between legacy find phase and LCR by analysis.

analysis, as presented in Fig. 2.8. We observe that in the ideal case, LCR performs similar to the legacy find phase for relatively short τ, and gradually outperforms it along with the increase of τ.

The improvement can be interpreted into the time diversity gain. As shown in Fig. 2.9, the absorbing probability distribution of LCR is flatter than that of the legacy find phase, which is more favourable to absorption due to the diversity gain in the time domain (reflected by the cyclic shift operation in LCR Markov chain construction).

For the other cases, LCR always outperforms the legacy find phase and the gain increases as the overall channel status of the three social channels becomes worse. It can be interpreted into the diversity gains in both time and frequency domains (reflected by the cyclic shift operation and the averaging operation of the absorbing probabilities in LCR Markov chain construction).

Note that, at first, the listen channel of the legacy find phase is randomly selected among the clean (or non-interfered) social channel(s) and the interfered channel(s). Thus, the TTD reduction effect of LCR implies that the gain obtained by LCR, compared with the case of the legacy scheme with bad channel(s), is much larger than the loss compared with the case of the legacy scheme with good channel(s).

Experimental Evaluation

We also implement LCR in the Galaxy Nexus by modifying wpa supplicant source code. We conduct experiments under the topology with four Wi-Fi devices shown in Fig. 2.10, where L, R, I_1, and I_2 indicate the local device, the remote device, the interferer at channel 11, and the interferer at channel 6, respectively. In order to make channels 11 and 6 interfered, we arrange the four Wi-Fi devices such that I_1 and I_2 are almost hidden from the local device, and make them ceaselessly generate UDP traffic to the associated APs as shown in Fig. 2.10. The lengths of the MPDUs transmitted by these two interferers are 1,536 bytes, and the PHY rates are dynamically controlled by the inherent link adaptation algorithm.

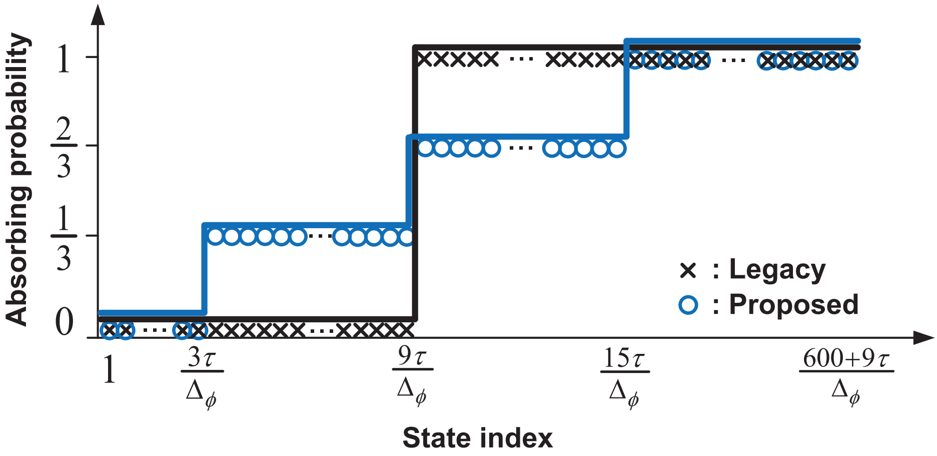


Figure 2.9: Comparison of absorbing probability distributions of legacy find phase and LCR in ideal environment.

Fig. 2.11 shows the average TTD obtained under three scenarios, i.e., (1) ideal environment, (2) I_1 is on and I_2 is off (indicated as '1-ch'), and (3) both I_1 and I_2 are on (indicated as '2-ch'). We observe LCR performs similarly to the legacy find phase in the ideal environment while the TTD reduction effect of LCR increases with the increase of τ due to the time diversity gain as discussed above. Besides, when the overall channel status becomes worse, the TTD reduction effect of LCR increases, thus yielding as high as 72% reduction compared with the legacy find phase in the 2-ch case.

Figs. 2.12, 2.13, and 2.14 show the empirical Cumulative Distribution Functions (ECDFs) of the TTD samples obtained in the experiments. We observe that LCR effectively cuts the long tails of the ECDF curves of the legacy find phase such that TTD can be confined within much narrower range, e.g., less than 20 seconds in the case of τ is 20 ms.

Although the experimental results cannot represent all the possible cases in reality, we argue that the performance gain of LCR is likely to reside between the two extreme cases, i.e., ideal (TTD reduction of 12.7% in average) and 2-ch (TTD reduction of 72%

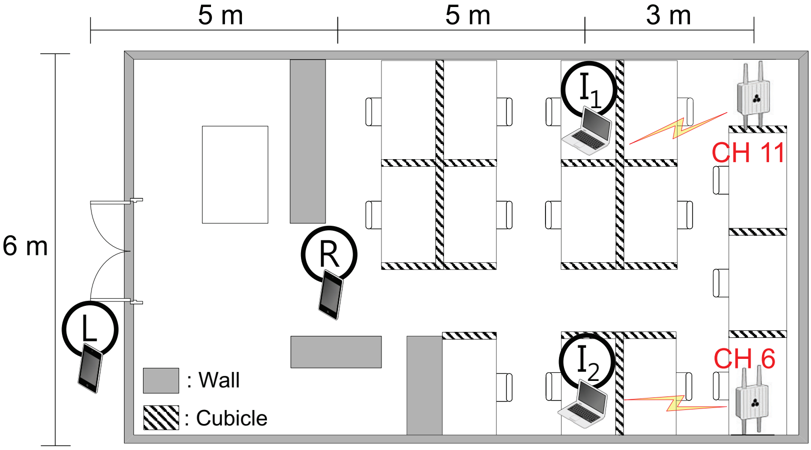


Figure 2.10: Experimental venue.

in average) in general.

2.5.3 Remarks

From the experiments, we found that among the total TTD samples, the percentages that the device discovery occurs at clean (non-interfered) channel(s) in LCR for the cases of 1-ch and 2-ch are 96.3% and 86%, respectively, while, in the legacy find phase, the corresponding percentages are 66.6% and 33.3%, respectively. It means LCR exploits better channel(s) more often for device discovery.

Considering the fact that service discovery and GO negotiation are also defined to be conducted at the listen channel after device discovery [28], we can imagine that LCR can be used by a device to statistically discern between interfered and clean channels during the device discovery phase and one of the channels judged as clean can be used in service discovery and GO negotiation to accelerate the entire P2P group formation.

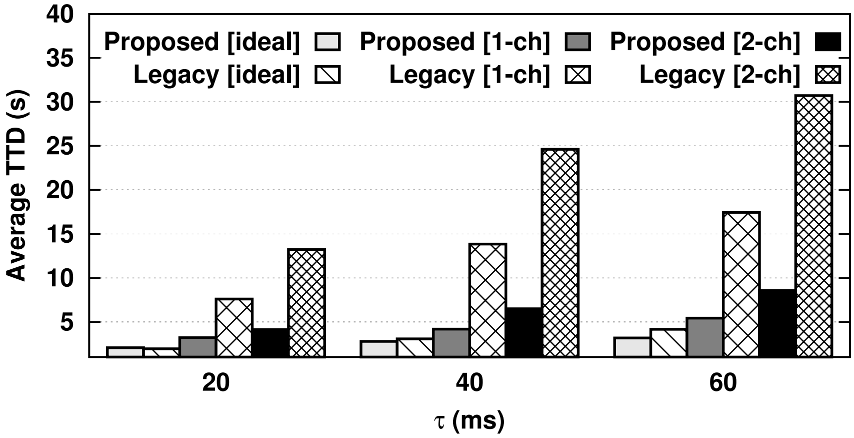


Figure 2.11: Experimental results: average TTD.

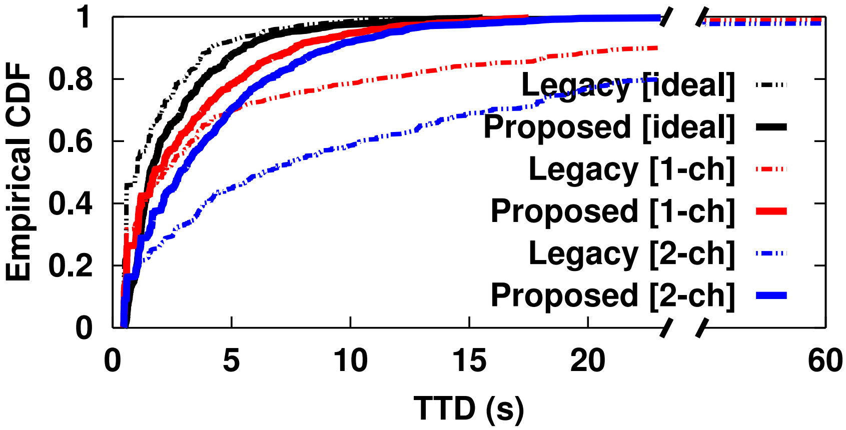


Figure 2.12: Empirical CDF of experimental results: $\tau = 20$ ms

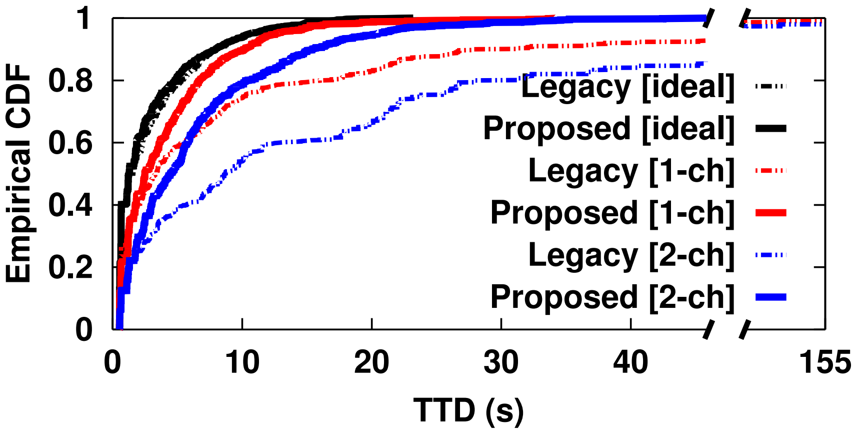


Figure 2.13: Empirical CDF of experimental results: $\tau = 40$ ms

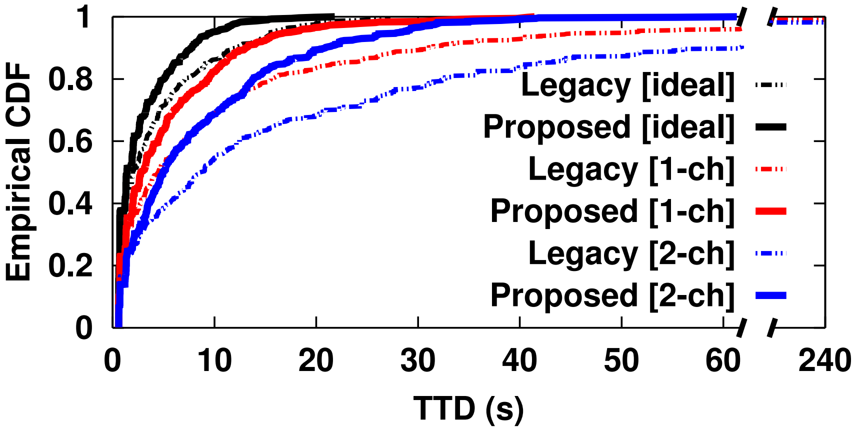


Figure 2.14: Empirical CDF of experimental results: $\tau = 60$ ms

2.5.4 Summary

In this chapter, we analyze the stochastic behavior of the two devices in the legacy find phase of Wi-Fi Direct, based on the Markov chain-based statistical model. We validate the effectiveness of the Markov model via both simulation and measurement study. Moreover, we improve the legacy find phase by proposing Listen Channel Randomization (LCR), of which the significant performance gain is verified theoretically and practically.

Chapter 3

BlueCoDE: Bluetooth Coordination in Dense Environment for Better Coexistence

3.1 Introduction

Cross-technology interference is emerging as a prominent problem at increasingly crowded 2.4 GHz Industrial, Science, and Medical (ISM) band. Wi-Fi and Bluetooth[1] (BT), the two most widely used technologies at 2.4 GHz ISM band, share the same spectra but are built on top of entirely different Physical (PHY) and Medium Access Control (MAC) layer designs; Wi-Fi is Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)-based static channel technology [1] while BT is Time Division Multiple Access (TDMA)-based frequency hopping technology [2]. How to make them coexist better has been a major technical challenge [3].

Due to the accelerated deployments of Wi-Fi hotspots and the growing popularity of BT-enabled peripherals, e.g., headset, speaker, etc., highly-dense Wi-Fi and BT co-existing environments become increasingly common, e.g., increasing number of commuters enjoy audio streaming via BT headsets at a subway station covered by dozens of Wi-Fi networks. Such trend intensifies the severity of the coexistence problem stem-

[1] In this chapter, Bluetooth means classic Bluetooth, which is widely used today for hands-free, speaker, mouse, etc., not Bluetooth Low Energy (BLE).

ming from the *heterogeneous* interference between Wi-Fi and BT devices and the *homogeneous* interference among BT devices. Accordingly, it is practically essential to deal with the coexistence problems in dense environments [31].

Wideband heterogeneous environment: The severity of the heterogeneous interference between Wi-Fi and BT has been largely mitigated by BT's Adaptive Frequency Hopping (AFH) capability, which renders the heterogeneous interference harmless by dividing their respective frequencies, i.e., BT hops within the BT channels free from Wi-Fi interference [2]. However, the interference still remains intrusive when all the BT channels are covered by several ambient Wi-Fi networks bringing highly-loaded strong interference; AFH is hardly effective, since there are no noticeable clues for desirable BT channels. In such a case, Wi-Fi and BT become mutually unavoidably harmful because a complete spectra separation is hardly feasible. We refer to this scenario as *Wideband Heterogeneous* (WBH) environment.

Narrowband homogeneous environment: Regarding the homogeneous interference among multiple BT devices, the respectively independent and pseudo-random properties of multiple BT devices' hopping patterns largely prevent packet collisions among them. However, when part of BT channels are occupied by one or more ambient Wi-Fi networks, AFH-enabled BT devices are supposed to share narrower spectrum untouched by the Wi-Fi networks, thus increasing packet collisions among them. It is a non-trivial problem in reality considering the fact that AFH is supported by most of the contemporary BT devices [32]. We refer to this scenario as *Narrowband Homogeneous* (NBH) environment.

In this chapter, we propose BlueCoDE, a coordination scheme for multiple nearby BT devices with the following features.

- It does not require the modifications of BT's existing PHY and MAC designs.
- It improves Wi-Fi performance tremendously in WBH environment by making the hopping patterns of multiple BT devices less harmful to Wi-Fi.

- It reduces collisions among multiple BT devices in WBH and NBH environments with almost zero collision probability.

In BlueCoDE, device referred to as *coordinator* controls multiple nearby BT devices in terms of their native clocks and the device addresses. The hopping sequence[2] of a BT device can be deliberately manipulated by controlling its address and clock; they are used as the inputs of hopping sequence generation function defined in the BT standard [2]. This feature is exploited by BlueCoDE to make multiple BT devices' hopping sequences always keep a non-zero channel offset from each other, thus becoming collision-free in WBH and NBH environments. Besides, in BlueCoDE, multiple piconets' hopping channels are closely bonded to reduce the number of interfered Wi-Fi channels. This effort drastically improves Wi-Fi performance in WBH environment.

In summary, we claim the following four contributions.

- We find out a simple way to manipulate hopping sequences of multiple BT devices to make them collision-free.

- We develop a method to make multiple BT devices less harmful to Wi-Fi via hopping sequence manipulation, and identify the rationale analytically.

- We propose BlueCoDE as a general framework to exploit aforementioned method in NBH and WBH environments.

- We corroborate the performance gain delivered by BlueCoDE via simulation and prototype-based experiments.

The rest of this chapter is organized as follows: in Section 3.2, we provide a brief introduction to the legacy BT operations and the coexistence problems addressed in this work. In Section 3.3, we introduce the method used in hopping sequence manipulation, and in Section 3.4, we give an overview of BlueCoDE. Two practical issues regarding BlueCoDE are addressed in Section 3.5. Thereafter, the performance gain of

[2] Hopping sequence is a sequence of hopping channel numbers over time.

BlueCoDE is evaluated via analysis, prototype-based experiments, and simulation in Sections 3.6, 3.7.1, and 3.7.2, respectively. Finally we conclude this chapter in Section 3.8.

3.2 Preliminary

3.2.1 Bluetooth Basics

Modulation: Two modulation modes are defined in the BT standard; a mandatory mode referred to as Basic Rate (BR) using Gaussian Frequency Shift Keying (GFSK), and an optional mode referred to as Enhanced Data Rate (EDR) using Phase Shift Keying (PSK) with two variants, i.e., $\pi/4$-Differential Quadrature PSK ($\pi/4$-DQPSK), and 8-Differential PSK (8DPSK).

Piconet: A piconet is a group of BT devices, consisting of a *master* and up to seven *slaves*. All the operations performed in a piconet are controlled by the master of the piconet.

Time slot: A time slot is 625 μs long, and a BT packet can only be transmitted at the start of a time slot. A BT packet can occupy one, three, or five time slots. In this work, we only consider one-slot packet, and the solutions for longer packets remain as future work.

Frequency hopping: There are 79 BT channels defined in 2.4 GHz ISM band (from 2.4 GHz to 2.483 GHz). The channels are spaced 1 MHz apart and ordered from number 0 to 78. Each BT packet is transmitted at a certain hopping channel. AFH enables interference avoidance by classifying 79 BT channels based on channel status and allows only good channels to hop. The channels used in AFH are referred to as *used channels* while the rest are called *unused channels*.

Clock and address: Each BT device has its own clock implemented as 28-bit binary counter (CLK) that increases by one every 312.5 μs (i.e., increment by two every BT time slot). In a piconet, CLKs of all the slaves are synchronized with that of the master, and slot transitions occur whenever CLK_1[3] of the master toggles between 0 and 1. Besides, each BT device has a unique 48-bit device address (ADDR) allocated by manufacturer. The master's ADDR is shared among all the slaves in the same piconet. In this chapter, unless stated otherwise, CLK and ADDR of a piconet correspond to those of the master of the piconet.

Hop selection kernel: Hopping sequence is determined by *hop selection kernel* defined in the BT standard, consisting of a series of binary operations—addition (ADD), exclusive or (XOR), 5-bit permutation (PERM5), and register selection—as shown in Fig. 3.1. Specifically, r_4 is used as an index by a register selection operation to select a hopping channel number from *Basic Channel Table* (BCT). BCT has a unified format; it contains all the 79 channel numbers, where the upper half contains the even numbered channels and the lower half contains the odd numbered channels, both of which are in the order of increasing channel number. When AFH is enabled, if the channel obtained from BCT (indicated as CH_b in Fig. 3.1) is an unused channel, channel re-mapping takes place, where r_4 and BCT are replaced with r_5 and *Used Channel Table* (UCT), respectively, to select a used channel CH_u. UCT contains only the used channel numbers, following an ordering rule similar to that of BCT. UCTs in different piconets can vary due to different channel status or different methods in channel classification. Note that in connection state,[4] the inputs of hop selection kernel, i.e., X, Y1, Y2, A–F, and F′, are derived from CLK, ADDR, and the number of used channels (indicated as Ω) of a piconet as shown in Table 3.1.

[3] CLK_i indicates the ith bit of CLK consisting of CLK_{0-27}.

[4] In other states such as inquiry state, the inputs of the hop selection kernel are different from those in connection state.

Table 3.1: Inputs of hop selection kernel in connection state.

Input	**Value**	**Input**	**Value**
X	CLK_{6-2}	**Y1**	CLK_{1}
Y2	$32 \times \mathrm{CLK}_{1}$	**A**	$\mathrm{ADDR}_{27-23} \oplus \mathrm{CLK}_{25-21}$
B	ADDR_{22-19}	**C**	$\mathrm{ADDR}_{8,6,4,2,0} \oplus \mathrm{CLK}_{20-16}$
D	$\mathrm{ADDR}_{18-10} \oplus \mathrm{CLK}_{15-7}$	**E**	$\mathrm{ADDR}_{13,11,9,7,5,3,1}$
F	$16 \times \mathrm{CLK}_{27-7} \bmod 79$	**F′**	$16 \times \mathrm{CLK}_{27-7} \bmod \Omega$

3.2.2 Problems of Interest

If we classify coexistence problems by the locations of the coexisting entities, there are *collocated* and *non-collocated* coexistence problems, depending on whether the entities are equipped within a single device or not. *In this work, we consider only non-collocated problems encountered among Wi-Fi and BT devices*; compared with the counterpart, where coexisting entities can exchange collaborative information internally, the *non-collocated* problem is more imperative to settle.

Besides, coexistence problems can be also categorized by the diversity of the coexisting technologies, i.e., *homogeneous* and *heterogeneous* coexistence problems, depending on whether the coexisting entities are utilizing the same type of technologies or not. *Considering the fact that, to some extent, homogeneous coexistence problem of Wi-Fi is addressed by CSMA/CA MAC protocol, we consider only the homogeneous coexistence problem of BT and the heterogeneous coexistence problem between Wi-Fi and BT.*

In summary, we focus on the following two cases.

- WBH environment: Several Wi-Fi networks occupy entire BT channels and there is no noticeably good BT channel, and hence, collisions between BT and Wi-Fi packets inevitably happen. The performances of both Wi-Fi and BT are evaluated. In this case, all the 79 BT channels are supposed to be used evenly in the long-term, even when AFH is exploited, due to the condition that there is

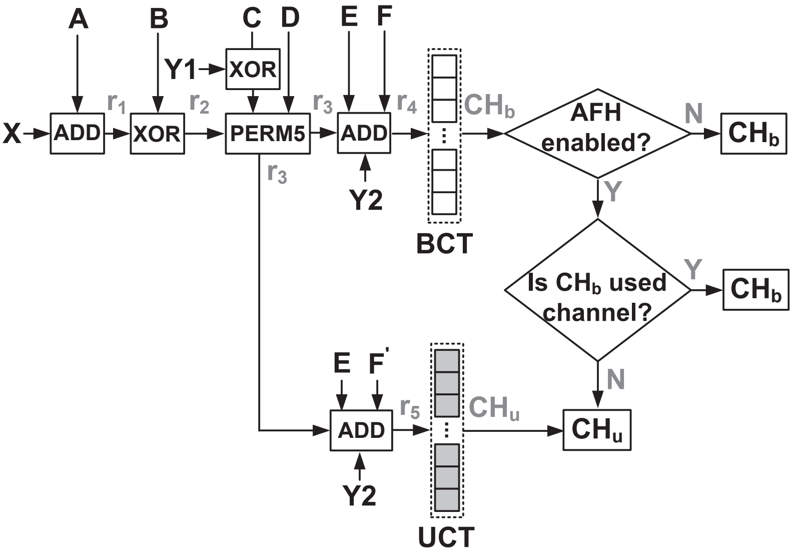


Figure 3.1: Hop selection kernel.

no distinctive quality among the 79 BT channels suffering from similarly strong Wi-Fi interference.

- NBH environment: Only part of BT channels are used due to Wi-Fi interference at the other BT channels, and hence, multiple piconets[5] coexist within the used channels, thus increasing the collisions among them; the fewer used channels, the higher collision probability. The performances of multiple coexisting piconets are evaluated. We do not evaluate Wi-Fi performance in this case, because a perfect spectra separation between Wi-Fi and BT is achieved.

[5] We assume all the piconets support AFH in this work.

3.3 Parallel Hopping Sequences for Collision-Free BT Coexistence

In this section, we present the principle and method used in hopping sequence manipulation.

3.3.1 Definition

We define that piconet a's hopping sequence S^a and piconet b's hopping sequence S^b are *parallel* if and only if the elements of S^a and S^b satisfy the condition that $(s_i^a - s_i^b) \mod \Omega$ is a non-zero 'constant' value regardless of the BT time slot index i. When AFH is not enabled, Ω, the number of used channels, becomes 79.

3.3.2 Conditions for Parallel Hopping Sequences

Hopping sequence is determined by the inputs of the hop selection kernel, i.e., X, Y1, Y2, A–F, and F′, when BCT and UCT are given. The inputs are derived from CLK and ADDR as shown in Table 3.1. It means that we can manipulate hopping sequence by controlling CLK and ADDR.

AFH-disabled case: In this case, multiple piconets generate hopping sequences only from their BCTs. That is, as long as the inputs, X, Y1, Y2, and A–F, are identical among different piconets, the respective hopping sequences are identical. In order to make multiple piconets' hopping sequences parallel, we need to differentiate the inputs, and make sure that, for each pair of piconets, the offset values between their hopping channel numbers are congruent modulo 79 over time. It is achieved by differentiating only the input E, the time-invariant value used in the last ADD operation. The conditions for parallel hopping sequences are listed as follows:

1) CLKs of multiple piconets should be synchronized.

2) $\mathrm{ADDR}_{27-0} \backslash \mathrm{ADDR}_{9,7,5,3,1}$ should be identical and $\mathrm{ADDR}_{9,7,5,3,1}$ should be

different for multiple piconets.

Note that $\mathrm{ADDR}_{13,11}$, which are also included in E, are excluded in differentiation as they are also included in the derivation of input D.

AFH-enabled case: In this case, hopping channel can be generated either from BCT or UCT. This makes the situation more complicated, and consequently, there are two additional conditions appended.

3) A common UCT should be used by multiple piconets.

4) In each BT time slot, all the piconets should use either BCT or UCT at the same time.

The necessity of the last condition arises from the fact that there is no guaranteed non-zero constant difference between the channels of each pair of piconets generated from BCT and UCT, respectively, even if all the first three conditions are satisfied. Note that the last condition cannot be satisfied all the time unless all the piconets are mandated to use only UCTs in AFH.[6] Accordingly, we define two different modes of BlueCoDE: 1) a Fully Standard-Compliant (FSC) mode, satisfying only the first three conditions and following the legacy hop selection kernel, and 2) an Almost Standard-Compliant (ASC) mode, whose operations are the same as those of FSC except that the use of BCT in AFH is prohibited to satisfy all of the four conditions.

In summary, if all the four conditions are satisfied, there are as many as $\min(2^5, \Omega)$ mutually parallel hopping sequences constrained by the number of different $\mathrm{ADDR}_{9,7,5,3,1}$ values and the number of used channels, Ω. Fig. 3.2 shows the comparison between legacy hopping sequences and proposed parallel hopping sequences, emphasizing the collision-free property of the three mutually parallel hopping sequences generated by piconets P1, P2, and P3. Note that, in the case of parallel hopping sequences, the offsets between the channel numbers of each pair of piconets are congruent modulo Ω

[6] The role of BCT in AFH is to facilitate the slaves, which do not support AFH, to remain synchronized with the AFH-enabled master.

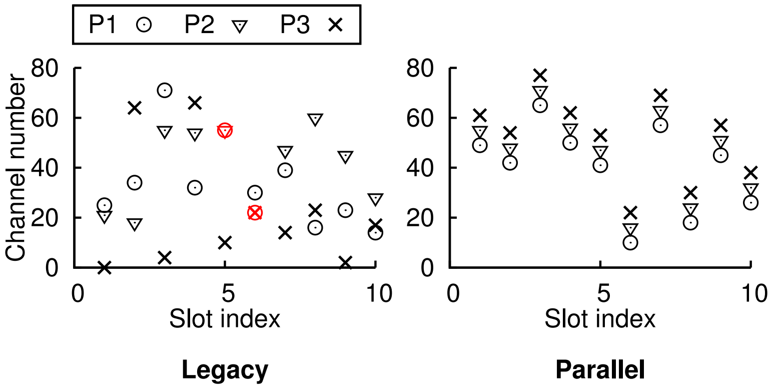


Figure 3.2: Hopping sequence example.

(79 in this case). This property is exploited by BlueCoDE to bond multiple piconets' hopping channels so that the resultant BT interference is less harmful to Wi-Fi compared with the legacy BT operation in WBH environment, as explained in detail in Section 3.6.

3.4 BlueCoDE Overview

We propose BlueCoDE with the following considerations.

- PHY and MAC of BT should not be modified.
- BlueCoDE is a general framework, in which the coordinator can be implemented using various types of devices.
- BlueCoDE should be practically feasible.

3.4.1 Coordination

As mentioned above, BlueCoDE coordinates multiple surrounding piconets to make them satisfy the first three (in FSC mode) or four (in ASC mode) conditions for parallel hopping sequences. Accordingly, there should be a coordinator to control the CLKs, ADDRs, and UCTs of multiple piconets. The coordination procedures consist of the following four steps.

Step 1. A master device supporting BlueCoDE will inform the coordinator of its identity information, i.e., its original device address and an indication bit to indicate the demand for coordination.

Step 2. Then, the coordinator replies with a list of control messages, i.e., a 28-bit temporal CLK (T_CLK), a 5-bit temporal $\text{ADDR}_{9,7,5,3,1}$ (T_ADDR),[7] a UCT expressed as a 10-byte bitmap, where value 1 (0) indicates a used (unused) channel, and a 9-bit time offset (t_o)[8] to indicate the time difference between the starting time of T_CLK and the moment the control messages are received.

Step 3. The master shares T_CLK and T_ADDR with the slave(s) in the same piconet and use them to perform normal BT operations t_o later.

Step 4. Repeat Steps 1 to 3 periodically with period τ to correct the synchronization errors due to the relative clock drift between the master and the coordinator (discussed in Section 3.5.2) and to confirm piconet's departure.

Besides, the coordinator has to detect Wi-Fi signals to determine whether it is in WBH or NBH environments. In a WBH environment, it will indicate all the BT channels as used channels in UCT, while in an NBH environment, it will deliver a common UCT containing only the BT channels free from Wi-Fi interference to multiple nearby piconets. Fig. 3.3 illustrates the coordination procedures, where Steps 1 and 2 are denoted as *acquisition* and Step 3 is denoted as *adjustment*, which are detailed in the following.

[7] We assume that the other bits of ADDR have been shared among BT devices supporting BlueCoDE beforehand.

[8] t_o is always less than the CLK resolution, i.e., 312.5 μs, such that 9-bit is sufficient to express it in unit of μs.

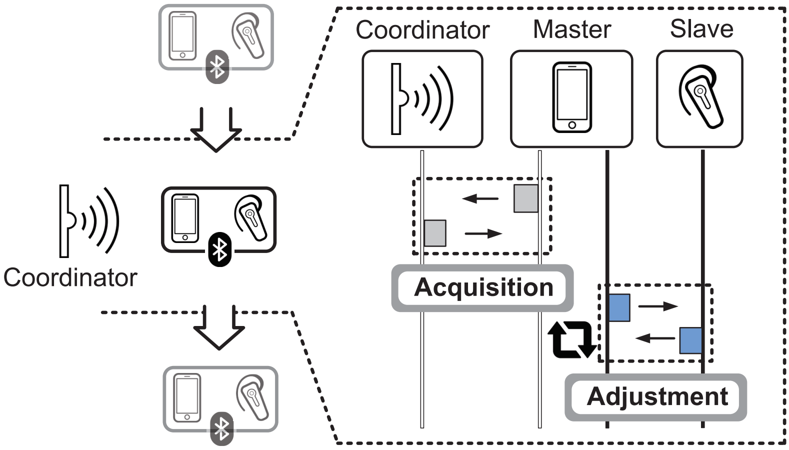


Figure 3.3: Overview of BlueCoDE operations.

3.4.2 Acquisition

Due to the fact that master device is normally equipped with multiple Radio Access Technologies (RATs), e.g., Global Positioning System (GPS), LTE, Wi-Fi, BT, Bluetooth Low Energy (BLE), etc., there are various candidate technologies to implement acquisition. Instead of restricting the device type and the RAT of the coordinator to a particular case, the main purpose of this work is to provide fundamental elements in building a coordinator, since it is more desirable to propose BlueCoDE as a general framework that can be easily adopted with various RATs suitable for different scenarios. Nonetheless, in order to deliver a deeper understanding of BlueCoDE in a more concrete context, we illustrate an example of the acquisition process under the assumption that the coordinator is a Wi-Fi AP and the master device is a Wi-Fi station equipped with Wi-Fi and BT combo chip, e.g., smartphone.[9]

Acquisition through Wi-Fi scanning: Typically, Wi-Fi station (i.e., the master device in this example) is supposed to periodically conduct active scanning, i.e., exchange of probe request and probe response frames, to discover neighboring Wi-Fi APs [33]. The frame exchange herein can be used as a side channel to perform acquisition by embed-

[9] This is the most common scenario, where BlueCoDE can be applied.

ding the identity information (involved in Step 1) and the control messages (involved in Step 2) into the vendor-specific fields[10] of probe request and probe response frames, respectively. In particular, Wi-Fi AP (i.e., the coordinator) can disseminate T_ADDR and UCT to multiple neighboring Wi-Fi stations by simply embedding them in the probe response frames. On the other hand, the CLK synchronization is a non-trivial task, and hence, we need to additionally incorporate the following two factors.

1) In most Wi-Fi and BT combo chips, such components as antenna (for 2.4 GHz radio), Low Noise Amplifier (LNA), and Power Amplifier (PA) are shared by Wi-Fi (at 2.4 GHz) and BT modules, thus resulting in various TDMA-like solutions for the collocated coexistence problem, so that Wi-Fi and BT modules can alternately work [34–36]. These solutions are enabled by a dedicated internal communication link, e.g., shared register, bidirectional bus, etc., between them to share current status and timing information [34–36]. In particular, in the case of Qualcomm QCA6234 combo chip [35], the clock source of the BT is provided internally from the collocated Wi-Fi. Such trends will continue to make products more compact and cost-effective. Therefore, we argue that as long as multiple master devices' Wi-Fi clocks are time-synchronized, we easily achieve CLK synchronization by deriving the T_CLK and t_o internally based on collocated Wi-Fi clocks.

2) In practice, simply embedding T_CLK and t_o into vendor-specific field of probe response frame may cause unpredictable CLK synchronization errors among multiple piconets since the delay between the time that the vendor-specific field is embedded and the time that the probe response frame is transmitted over the air is random due to the nature of CSMA. To achieve stringent (microsecond-

[10] Many Wi-Fi AP vendors implement proprietary features with vendor-specific Information Element (IE), which is defined in IEEE 802.11 standard to provide flexibility to the vendors for proprietary services. It can be easily embedded in probe request, probe response, and beacon frames by modifying driver codes of off-the-shelf Wi-Fi devices.

level accuracy) time synchronization, we can re-use Wi-Fi's *Timing Synchronization Function* (TSF) defined in the Wi-Fi standard as a mandatory feature [1]. It defines that each Wi-Fi AP should set the value of the *timestamp* of probe response frame (or beacon frame) to the value of its TSF timer—a 64-bit binary counter based on a 1-MHz clock with microsecond resolution—at the moment that the first bit of the timestamp is transmitted over the air, and each Wi-Fi station should update its local TSF timer based on the timestamp coming from the associated Wi-Fi AP in order to achieve time synchronization; the received timestamp value should be adjusted by adding the Wi-Fi station's local processing delay passed since the first bit of the timestamp was received in order to achieve accurate time synchronization.[11] So, CLK synchronization can be achieved by deliberately letting multiple Wi-Fi stations associate with the coordinator or just overhear the timestamp value coming from the coordinator without association. Then, T_CLK and t_o are derived by $\mathrm{T\_CLK} = \left\lceil \frac{\mathrm{CLK_{tsf}}}{312.5} \right\rceil$ and t_o= $\lfloor \mathrm{T\_CLK} \cdot 312.5 - \mathrm{CLK_{tsf}} \rfloor$, respectively, where $\mathrm{CLK_{tsf}}$ is the timestamp value.

3.4.3 Adjustment

After acquisition, master needs to update its CLK, ADDR, and UCT to the new values. The UCT and CLK can be updated by *channel map update procedure* and *coarse clock adjustment procedure*, respectively, which are existing features in the BT standard. Regarding the ADDR, we have two options: 1) introduce a new protocol to enable ADDR adjustment, or 2) remove all the paired slaves and re-establish the piconet. Apparently, the first option introduces less control overhead but necessitates a newly defined protocol, while the second option can be immediately supported by off-the-shelf BT products with extra control overhead. Which option is better should be

[11] On a commercial level, industry vendors assume the Wi-Fi TSF's synchronization to be within 25 μs [37].

determined by the specific system requirements. Besides, if a piconet has not been established yet, master can carry out normal pairing procedure to set up a piconet based on the control messages obtained during the acquisition.

3.4.4 Piconet Management

In BlueCoDE, the coordinator carefully selects a proper $\mathrm{ADDR}_{9,7,5,3,1}$ value for each piconet to make the overall BT interference less harmful to Wi-Fi in WBH environment. We conclude that when the hopping channels of the multiple piconets are more closely located in frequency domain in each BT time slot, the resultant BT signals are less harmful to Wi-Fi, as further explained in Section 3.6. We also conclude that the channel offset between each pair of piconets should be at least 2 MHz to overcome the adjacent channel interference caused by in-band emission as explained in Section 3.5.1. As a result, in BlueCoDE, the coordinator allocates several consecutive $\mathrm{ADDR}_{9,7,5,3,1}$ values to multiple piconets starting from all 0 sequence, and consequently, the hopping channels of the multiple piconets becoming an arithmetical progression with 2 MHz offset in each time slot due to the ordering rules of BCT and UCT described in Section 3.2.1. Note that in NBH and WBH environments, the maximum number of piconets, which BlueCoDE can coordinate simultaneously, are $\min\left(2^5, \left\lfloor \frac{\Omega}{2} \right\rfloor\right)$ and $\min\left(2^5, \left\lfloor \frac{79}{2} \right\rfloor\right)$, respectively, due to the 2 MHz offset.[12]

In order to maintain the aforementioned hopping channel relations among nearby piconets, the coordinator should periodically inspect whether there is a new piconet that has not been coordinated or a piconet that has left the coordinator's coverage. When a new piconet joins, the next consecutive $\mathrm{ADDR}_{9,7,5,3,1}$ is assigned to it. When a piconet has left, its $\mathrm{ADDR}_{9,7,5,3,1}$ will be reused by the piconet[13] currently with the largest $\mathrm{ADDR}_{9,7,5,3,1}$ value so that no further ADDR adjustment is needed among

[12] We did not consider the case when the number of piconets exceeds the maximum supportable piconet number in both NBH and WBH environments, which remains as future work.

[13] The piconet will be informed of the new $\mathrm{ADDR}_{9,7,5,3,1}$ through the next control messages received afterwards.

other piconets.

3.5 Practical Issues

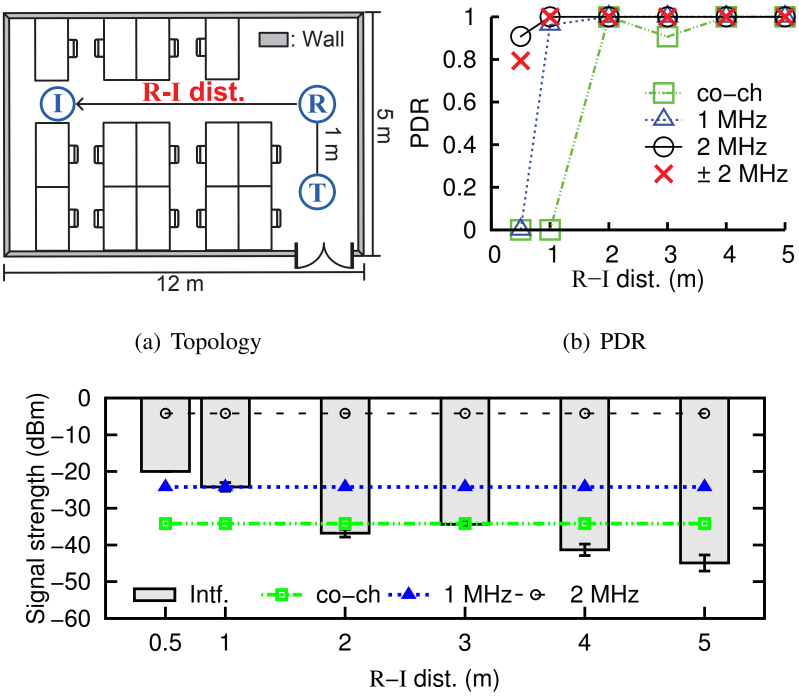
3.5.1 In-band Emission

Two BT signals at different channels can severely interfere with each other due to the in-band spurious emission [2]. The BT standard stipulates the interference rejection capabilities for each modulation type with respect to various types of in-band interference as shown in Tables 3.2. Each value (in dB) indicates the minimum acceptable Signal-to-Interference Ratio (SIR) to meet the Bit Error Rate (BER) requirement (i.e., BER under 0.001) defined in the BT standard.

In the cases of co-channel and 1 MHz offset interference, if the output power levels of a transmitter and an interferer are similar, the distance between the interferer and the receiver should be larger than that between the transmitter and the receiver to make SIR larger than 0 dB. However, this condition cannot be always satisfied in dense environment. *Consequently, in* BlueCoDE*, we advocate the channel offset between each pair of piconets should be at least* 2 *MHz.*

We conducted a simple experiment using Ubertooth [38]—an open source BT platform equipped with cc2400 transceiver—to inspect the impacts of various types of in-band interference in real world. The right-most column of Table 3.2 shows the minimum acceptable SIRs for the case of Ubertooth, which supports GFSK only.

As illustrated in Fig. 3.4(a), the experiment was conducted with three roles of Ubertooth devices, i.e., transmitter (indicated as '**T**'), receiver (indicated as '**R**'), and interferer (indicated as '**I**'), in an office environment. The distance between **R** and **T** is fixed at 1 m, analogous to the general case of BT usage, and the distance between **R** and **I** (R-I distance) varies from 0.5 m to 5 m. **T** transmits 1,000 BT packets to **R** with 0 dBm output power (i.e., the nominal output power of BT) at BT channel 39, (i.e., the channel in the midst, centered at 2.441 GHz), and Packet Delivery Rate (PDR) is



(a) Topology

(b) PDR

(c) Interference strength

Figure 3.4: In-band emission experiment topology and results.

measured at **R**. Each packet is 366-bit long without Forward Error Correction (FEC), analogous to a HV-3 packet format [2]. A packet is successfully delivered if all the bits of the packet are received correctly.

Interferer **I** generates GFSK-modulated pseudo-random binary sequence ceaselessly with 0 dBm output power at channels 37 (-2 MHz offset), 38 (-1 MHz offset), 39 (co-channel), 40 ($+1$ MHz offset), and 41 ($+2$ MHz offset), respectively, and the measured PDRs are averaged for each absolute offset value. We also consider the case when two **I**'s located at the same position operate at channels 37 and 41 simultaneously (indicated as ± 2 MHz) to verify the performance of the piconet in BlueCoDE with two neighboring piconets always operating at the channels with -2 MHz and $+2$ MHz offsets, respectively.

Fig. 3.4(b) shows the PDR results. As expected, when **I** operates at co-channel and

Table 3.2: Interference rejection capabilities.

Interference Type	**GFSK**	$\pi/4$-**DQPSK**	**8DPSK**	**Ubertooth**
Co-Channel	11 dB	13 dB	21 dB	10 dB
1 MHz offset	0 dB	0 dB	5 dB	0 dB
2 MHz offset	-30 dB	-30 dB	-25 dB	-20 dB
$\geq$ 3 MHz offset	≤ -40 dB	≤ -40 dB	≤ -33 dB	≤ -41 dB

1 MHz offset channel, R-I distances should be at least 2 m and 1 m for satisfactory PDR, respectively. Instead, if the channel offset becomes 2 MHz, even when there are two **I**'s, PDRs remain always fairly high regardless of the R-I distance, even though Ubertooth is supposed to perform much worse than off-the-shelf BT devices in this case.[14] In order to deliver deeper understanding of the PDR results, Fig. 3.4(c) shows the received signal strengths at **R** of the signals from **I** when **R** and **I** operate at the same channel with various R-I distances. Fig. 3.4(c) also compares the received signal strength with the maximum allowable interference strength obtained by subtracting the minimum acceptable SIRs of Ubertooth (shown in Table 3.2) from the wanted signal strength (from **T** to **R**) on logarithmic scale. The comparison proves the validity of the PDR results, since when the interference strength is lower than the maximum allowable value, the resulting PDR should be satisfactorily high.

3.5.2 Clock Drift

Each BT device has its own clock supposed to advance with a clock rate of 3.2 kHz as described in Section 3.2.1. However, due to the imperfection of the hardware generating signal source such as crystal oscillator, and the impact of the surrounding environments, e.g., temperature, humidity, etc., the clock rate fluctuates over time with a confined drift rate. Similarly, the coordinator's clock also has its own drift rate. Considering the fact that the maximum clock synchronization errors between each pair

[14] The minimum acceptable SIR of Ubertooth is higher than the three SIRs defined in the BT standard for 2 MHz offset case as shown in Table 3.2.

of piconets should be less than the frequency hopping guard time I_{S}[15] (illustrated in Fig. 3.7) at the end of each BT time slot, all the masters should limit their clock synchronization errors under $\frac{I_{\mathrm{S}}}{2}$ with respect to the coordinator's clock. A simple algorithm is proposed to cope with the clock drift problem by adjusting the coordination period elaborated as follows.

1) Master m first uses the initial coordination period τ_0 for the first T coordination intervals.

2) When it receives control messages from the coordinator in the ith coordination interval, the relative clock drift rate between them during the ith interval is calculated as

$$r_d^{m,i} = \frac{\left| t_{\mathrm{CLK}} \times (\mathrm{T\_CLK}^i - \mathrm{CLK}^{m,i}) - (t_o^i + \Delta^i) \right|}{\tau_0} \tag{3.1}$$

where t_{CLK}, $\mathrm{CLK}^{m,i}$, and Δ^i indicate CLK cycle (312.5 μs), the CLK value of master m when receiving the control messages in the ith interval, and the actual elapsed time since the moment that the master's clock becomes $\mathrm{CLK}^{m,i}$, respectively.

3) After T intervals, the master calculates the average of the preceding T clock drift rates, $\overline{r_d^m}$, and the next coordination period is calculated as $\tau = \frac{I_{\mathrm{S}}}{2\overline{r_d^m}}$.

4) After that, the coordination period τ is updated whenever the master receives control messages based on the relative clock drift rate during the previous interval.

Note that the coordination period should be always upper bounded by the average sojourn time of a piconet considering the mobility pattern in a specific venue.

[15] The guard time is at least 259 μs in the case of one-slot packet.

3.6 Analytical Discussion

In this section, we analyze the performance gain delivered by BlueCoDE compared with legacy scheme, i.e., the operation of each piconet is independent of the other piconets, in terms of the Wi-Fi performance in WBH environment and the BT performance in NBH environment.[16]

In the following analysis, for the sake of simplicity, we assume that 1) each piconet always transmits one-slot packet every BT time slot, and 2) a Wi-Fi device in idle state detects an on-going BT transmission with probability 1 as long as the hopping channel of the BT transmission overlaps with the 20 MHz operating channel of the Wi-Fi device in frequency domain.

3.6.1 Wi-Fi Performance in WBH Environment

The performance of Wi-Fi in WBH environment is mainly determined by two aspects: 1) Packet Error Rate (PER), representing how reliably an on-going Wi-Fi packet can be received by its receiver without error, and 2) Channel Access Probability (CAP), representing how frequently a Wi-Fi device can grab the wireless medium to transmit a packet.

PER: We here assume that a Wi-Fi packet is received in error as long as at least one BT packet is collided with it in both time and frequency domains. Since, for a given Wi-Fi packet with a certain PHY rate, there will be a fixed number of BT time slots overlapping with it in time domain, we focus on the frequency domain performance, i.e., the distribution of the number of overlapping BT packets in frequency domain during a BT time slot. With N piconets, the distribution with BlueCoDE is derived by counting the number of overlapping BT packets for the 79 possible hopping patterns as illustrated in Fig. 3.5, which can be expressed with the following Probability Mass

[16] Since the BT performance in WBH environment is influenced by both Wi-Fi and BT devices in the neighborhood, it is difficult to derive a general analytical model for the analysis. We will evaluate it via experiments and simulation in Section 3.7.

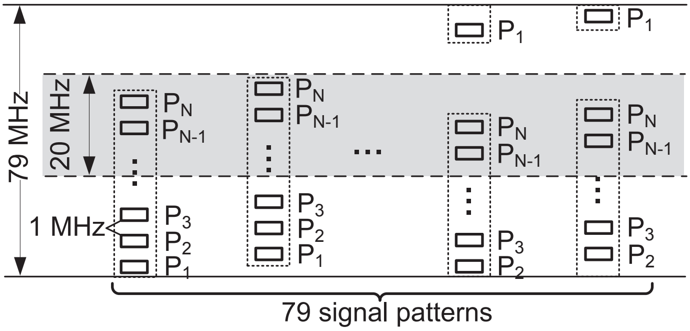


Figure 3.5: BT and Wi-Fi overlapping patterns in BlueCoDE.

Function (PMF).

$$f_p(o_\mathrm{B}) = \begin{cases} \frac{61-2N}{79}, & o_\mathrm{B} = 0, \\ \frac{4}{79}, & o_\mathrm{B} = 1, 2, ..., N-1, \\ \frac{22-2N}{79}, & o_\mathrm{B} = N, \end{cases} \tag{3.2}$$

for $0 < N < 11$, and

$$f_p(o_B) = \begin{cases} \max\left(\frac{61-2N}{79}, 0\right), & o_\mathrm{B} = 0, \\ \frac{4}{79}, & o_\mathrm{B} = 1, 2, ..., 9, \\ \min\left(\frac{2N-18}{79}, \frac{43}{79}\right), & o_\mathrm{B} = 10, \\ 0, & \text{otherwise}, \end{cases} \tag{3.3}$$

for $11 \leq N \leq 32$,[17] where o_B indicates the number of overlapping BT packets in a BT time slot. Then, the distribution with legacy scheme has the PMF

$$f_l\left(o_\mathrm{B}\right) = \binom{N}{o_\mathrm{B}} \left(\frac{20}{79}\right)^{o_\mathrm{B}} \left(\frac{59}{79}\right)^{N-o_\mathrm{B}}, o_\mathrm{B} = 0, 1, ..., N, \tag{3.4}$$

following a binomial distribution due to the independence among multiple piconets' hopping channels.

Fig. 3.6 shows the Cumulative Distribution Functions (CDFs) of o_B for N equal to 10 and 20, respectively. In both cases, BlueCoDE yields clean 20 MHz spectra (i.e.,

[17] The maximum number of piconets BlueCoDE can coordinate simultaneously in WBH environment is 32 as explained in Section 3.4.4.

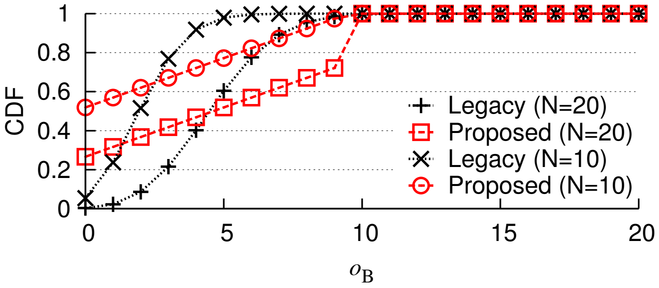


Figure 3.6: CDF of the number of overlapping BT packets in a time slot.

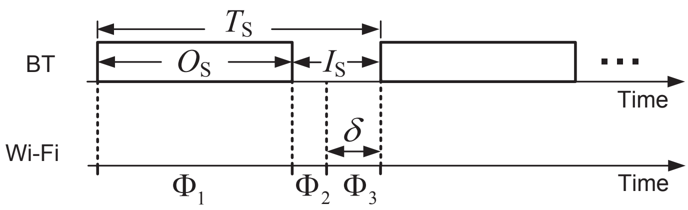
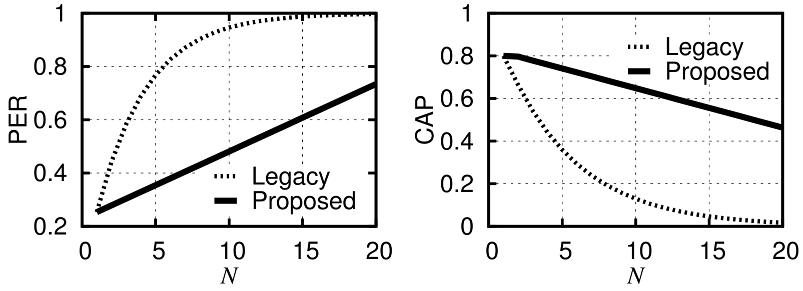


Figure 3.7: The intervals for Wi-Fi's channel access within a BT time slot.

$o_{\mathrm{B}} = 0$) with higher probability compared with the legacy scheme. The cross points between the CDF curves with BlueCoDE and with the legacy scheme explain the rationale behind the superiority of BlueCoDE; *it makes an already occupied Wi-Fi channel collided with more BT packets simultaneously, and hence, the other Wi-Fi channels can be unoccupied so that overall more unoccupied Wi-Fi channels can be yielded compared with the legacy scheme.* Finally, the PERs with BlueCoDE and with the legacy scheme are $1 - f_p(0)$ and $1 - f_l(0)$, respectively.

CAP: We define CAP as the probability of a Wi-Fi channel being idle for at least δ since the arrival of a Wi-Fi packet, whose arrival time is uniformly distributed, where δ is the expectation of the Wi-Fi's channel access delay.[18] For the legacy scheme, CAP

[18] For 802.11 Distributed Coordination Function (DCF), δ is 95.5 μs, i.e., Distributed Inter Frame Space (DIFS) (28 μs) plus the expected initial random backoff time calculated as Wi-Fi slot time (9 μs) multiplied by the expected initial backoff counter (7.5).



(a) Wi-Fi PER

(b) Wi-Fi CAP

Figure 3.8: Wi-Fi performance analysis in WBH environment.

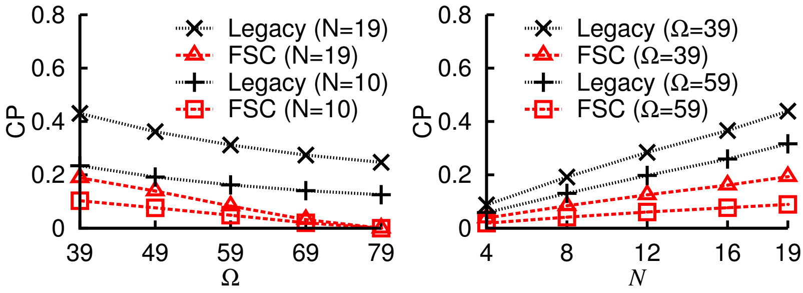
can be expressed as

$$CAP_l = \left(\frac{I_{\mathrm{S}} - \delta}{T_{\mathrm{S}}} + \frac{O_{\mathrm{S}} + \delta}{T_{\mathrm{S}}} \cdot \frac{59}{79}\right)^N, \tag{3.5}$$

where N, T_{S}, O_{S}, and I_{S} indicate the number of piconets, the duration of a BT time slot (625 μs), the actual transmission time of a one-slot BT packet, and the frequency hopping guard time at the end of each BT time slot, respectively. The base in (3.5) corresponds to the CAP when coexisting with a single piconet. The first term of the base indicates the probability that a Wi-Fi packet arrives during the time interval Φ_2 shown in Fig. 3.7, such that the corresponding Wi-Fi device can transmit the packet after δ with probability 1. Similarly, the second term is the probability that a Wi-Fi packet arrives during Φ_1 or Φ_3, such that the corresponding Wi-Fi device can transmit the packet after δ with probability $\frac{59}{79}$. There is an exponent N, since the operations of N piconets are independent of each other. Similarly, with BlueCoDE, the CAP becomes

$$CAP_p = \frac{I_{\mathrm{S}} - \delta}{T_{\mathrm{S}}} + \frac{O_{\mathrm{S}} + \delta}{T_{\mathrm{S}}} \cdot f_p(0). \tag{3.6}$$

Fig. 3.8 shows PER and CAP of Wi-Fi when coexisting with N piconets. We observe that BlueCoDE yields lower PER and higher CAP compared with the legacy scheme; the PER (CAP) with the legacy scheme grows (decays) rapidly and saturates, while it grows (decays) gradually with BlueCoDE, thus resulting in the expanding gaps between them until N becomes approximately 10.



(a) CP vs. Ω (b) CP vs. N

Figure 3.9: CP comparison between BlueCoDE (in FSC mode) and legacy scheme in NBH environment.

While the analysis is rather simplistic, it essentially captures the rationale behind the superiority of BlueCoDE, capable of explaining the performance gains shown in Section 3.7.

3.6.2 BT Performance in NBH Environment

As described in Section 3.3.2, in ASC mode, BlueCoDE eliminates collisions among multiple piconets completely in NBH environment. In FSC mode, the Collision Probability (CP), i.e., the probability that a piconet collides with one or more nearby piconets by selecting the same hopping channel,[19] with the number Ω of used channels is

$$CP_{\mathrm{FSC}} = 1 - \frac{\Omega}{79} \cdot \prod_{i=1}^{N-1} \left(1 - \max\left(\frac{80-\Omega-i}{(79-i)\,\Omega}, 0\right)\right) - \left(\frac{79-\Omega}{79}\right) \cdot \prod_{i=1}^{N-1} \left(1 - \max\left(\frac{\Omega-i+1}{(79-i)\,\Omega}, 0\right)\right). \tag{3.7}$$

We also derive the CP with the legacy scheme by utilizing the statistical model proposed in [39]. Fig. 3.9 shows the CPs with the number Ω of used channels (Fig. 3.9(a)) and the number N of piconets (Fig. 3.9(b)), respectively, where the CPs with FSC mode are always much lower than those with the legacy scheme, due mainly to the

[19] In analysis, we only consider the co-channel collisions.

considerable collision-free cases in FSC mode when all the piconets use either UCT or BCT at the same time.

3.7 Evaluation

We developed a prototype of BlueCoDE on Ubertooth platform [38] and conducted experiments to corroborate its effectiveness in real wireless environment. We made a Ubertooth device take the coordinator role,[20] and at the beginning of each experiment, the other Ubertooth devices resided at a predefined BT channel (channel 39 in our case) and waited for a *trigger frame* transmitted by the coordinator, based on which the channel hopping operations of these Ubertooth devices were triggered simultaneously, thus achieving CLK synchronization as illustrated in Fig. 3.10, where $f(\cdot)$, $\mathrm{CLK}(i)$, and $\mathrm{T\_ADDR}\{j\}$ indicate hopping sequence generation function, CLK value at the ith BT time slot since the trigger frame was received with initial value of $\mathrm{T\_CLK}$, and $\mathrm{T\_ADDR}$ for the jth Ubertooth device, respectively. The control messages, i.e., $\mathrm{T\_CLK}$, $\mathrm{T\_ADDR}$, and UCT, are stored in advance in the firmware of each Ubertooth device.

We also implemented BT module in NS-3 simulator [30], where the mutual interference between Wi-Fi and BT is reflected in both time and frequency domains based on the model proposed in [40].

In both experiment and simulation, we only consider the downlink traffic scenario of Wi-Fi network, which accounts for the majority of Wi-Fi traffic in reality. Besides, Ubertooth devices taking the roles of transmitter and interferer are fully loaded with one-slot BT packets in HV-3 format.

[20] Since the firmwares of Wi-Fi and BT combo chips are proprietary, we are not authorized to modify them to achieve CLK synchronization in the way described in Section 3.4.2.

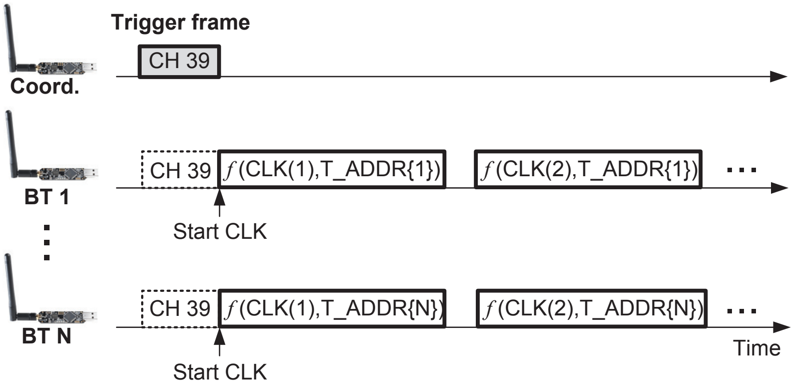


Figure 3.10: Prototype methodology.

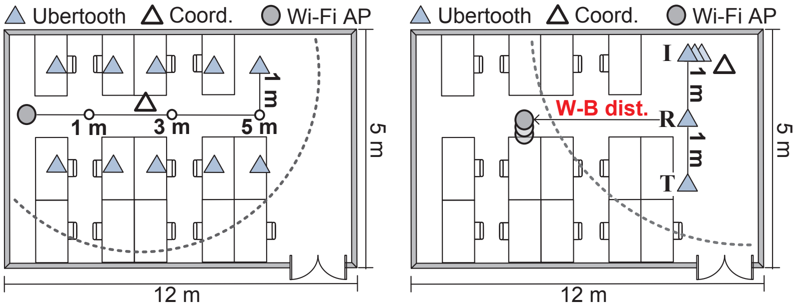
3.7.1 Prototype-based Experiments

WBH experiment for Wi-Fi performance: The experiment was conducted under the topology shown in Fig. 3.11(a). A Wi-Fi station (Galaxy Nexus smartphone) was receiving downlink traffic from the associated Wi-Fi AP (Buffalo WZR-HP-G300NH2),[21] which was fully loaded with 1470-byte UDP datagrams[22] destined to the Wi-Fi station with the internal rate-adaptation algorithm enabled.

Fig. 3.12(a) shows the application layer throughput of the Wi-Fi station, when it is located 1 m, 3 m, and 5 m away from the AP, respectively. Without BT interference, the throughput is about 30 Mb/s. When there is BT interference, we observe that the throughput performance with BlueCoDE is much better than that with the legacy scheme. Note that when the distance is larger than 1 m, the throughput is almost 0 Mb/s with the legacy scheme, while BlueCoDE always provides throughput above 12 Mb/s. Fig. 3.12(b) shows the empirical CDF of the Modulation and Coding Scheme (MCS)

[21] Although WBH environment needs several Wi-Fi networks to cover the entire BT channels, we only deploy a single Wi-Fi AP operating at Wi-Fi channel 6, since only the Wi-Fi performance is of interest in this experiment, which is independent of Wi-Fi networks operating at other non-overlapping Wi-Fi channels.

[22] The UDP datagrams were generated from iPerf application running on the Wi-Fi AP supporting OpenWrt [41].



(a) WBH for Wi-Fi

(b) WBH for BT

Figure 3.11: Experiment topologies.

indices[23] of the Wi-Fi packets captured during the experiment.[24] It unfolds that with the legacy scheme, the AP uses lower ordered MCS indices more frequently to deal with packet losses, thus degrading the performance, since the lengthened Wi-Fi packet duration tends to incur more collisions with BT packets.

WBH experiment for BT performance: We employed another topology[25] for the evaluation of the BT performance in WBH environment as illustrated in Fig. 3.11(b), consisting of an Ubertooth transmitter (indicated as '**T**'), an Ubertooth receiver (indicated as '**R**') placed 1 m away from **T**, several collocated[26] Ubertooth interferers (indicated as '**I**') placed 1 m away from **R**, and three collocated fully loaded Wi-Fi APs at Wi-Fi channels 1, 6, and 11,[27] respectively.

Fig. 3.12(c) shows the PER of **R**, when the distance between the Wi-Fi APs and **R** (W-B distance) varies from 1 m to 4 m, with five ($N = 6$ including T) and nine ($N = 10$) **I**'s, respectively. When W-B distance is 1 m, the Wi-Fi interference dominates the PER, and hence, the PER with BlueCoDE is higher than that with the legacy scheme due mainly to the fact that the higher CAP yielded by BlueCoDE (shown in Fig. 3.8(b))

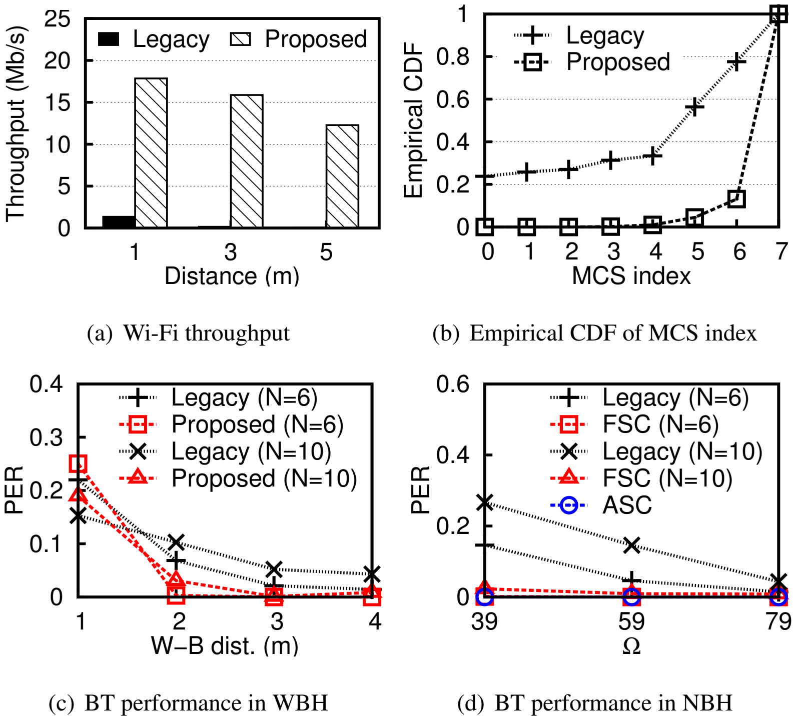
[23] Higher MCS index means higher but less reliable PHY rate.

[24] Wi-Fi packets were captured by a laptop placed beside the AP.

[25] We select a more densely deployed topology, where collisions between **I**'s and **T** are exposed more explicitly.

[26] In this section, 'collocated' means being placed at the same position.

[27] The three non-overlapping Wi-Fi channels cover almost the entire BT channels.



(a) Wi-Fi throughput

(b) Empirical CDF of MCS index

(c) BT performance in WBH

(d) BT performance in NBH

Figure 3.12: Experimental results.

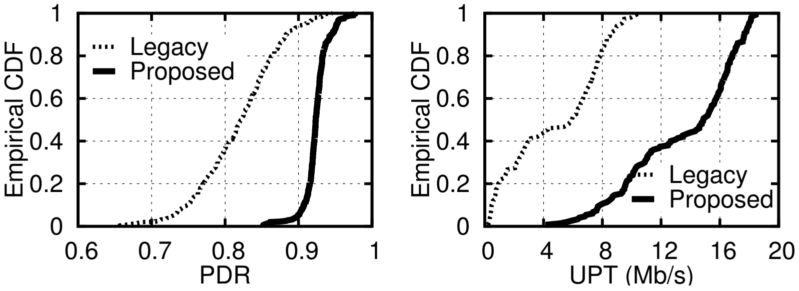
allows more Wi-Fi packets to be transmitted, and hence, severer Wi-Fi interference is imposed to **R**. When W-B distance is larger than 1 m, the PERs with BlueCoDE become nearly zero while with the legacy scheme, PERs are still noticeable due to the collisions between **T** and **I**'s. *Note that if the APs are not always fully loaded, but have finite and fixed amount of data to transmit, e.g., stations download a certain number of files through the APs, **R** is affected less by Wi-Fi with* BlueCoDE *than with the legacy scheme regardless of W-B distances, since the legacy scheme will induce more Wi-Fi packet errors followed by several packet retransmissions with lower PHY rates,*[28] *thus incurring longer Wi-Fi channel occupancy time compared with* BlueCoDE*, as verified in Section 3.7.2.*

NBH experiment for BT performance: The experiment was conducted under the topology similar to Fig. 3.11(b) except that there was no involved Wi-Fi AP, and the numbers of used BT channels, Ω, equal to 39, 59, and 79 were considered, respectively. The PERs of **R** are shown in Fig. 3.12(d). As expected, BlueCoDE in FSC (ASC) reduces (eliminates) the collisions between **T** and **I**'s substantially (completely), thus always achieving much lower PERs than the legacy scheme.

3.7.2 NS-3 Simulation

To eliminate the correlation between performance and specific topology, we conducted NS-3 simulations with 100 random topologies to evaluate Wi-Fi and BT performances in WBH environment. In each topology, 10 BT piconets, three Wi-Fi APs at channels 1, 6, 11, respectively, and three Wi-Fi stations associated with the three Wi-Fi APs, respectively, were randomly deployed within a 20 m by 20 m rectangular region. Each piconet consists of a master and a slave located 1 m apart. During each simulation run, we let each Wi-Fi AP transfer a 0.5 MB file to its Wi-Fi station with Minstrel [42]

[28] In Wi-Fi, a packet error will be followed by several packet retransmissions with more robust but lower PHY rates until the packet is received correctly due to Automatic Repeat Request (ARQ) and rate-adaptation mechanisms.



(a) BT performance

(b) Wi-Fi performance

Figure 3.13: NS-3 simulation results.

rate-adaptation algorithm enabled, and made each piconet fully loaded with one-slot BT packets. Figs. 3.13(a) and 3.13(b) show the empirical CDFs of the BT's PDR[29] and Wi-Fi's User Perceived Throughput (UPT) defined as the size of the file divided by the time to transfer it, respectively. The PDR CDF curve with BlueCoDE is densely concentrated—most piconets achieve PDR larger than 0.9, while the curve with the legacy scheme is relatively flat, ranging from 0.7 to 0.9, verifying that BT is less affected by Wi-Fi with BlueCoDE when the amount of Wi-Fi data is fixed. As for the UPT, BlueCoDE outperforms the legacy scheme as expected.

3.8 Summary

In this chapter, we propose BlueCoDE to mitigate the performance degradation of Wi-Fi and BT in WBH and NBH environments, respectively. The performance enhancements of BlueCoDE compared with the legacy scheme are evaluated through analysis, simulation and prototype-based experiments.

It should be noted that Wi-Fi's vulnerability to neighboring BT piconets is much severer than expected; from both the analysis and experimental results, we learn that merely 10 BT piconets can completely destroy the neighboring Wi-Fi networks. That reminds us of the necessity of the solution to mitigate the severity of the problem. We

[29] A PDR of a piconet during a simulation run becomes a PDR sample.

expect BlueCoDE to be a breakthrough solution for coexistence in dense Wi-Fi and BT environments.

Chapter 4

Robust BLE Packet Detection for RSSI Acquisition in Indoor Localization System

4.1 Introduction

The growth of wireless and mobile communication technologies offers new possibilities for location-aware information systems [43]. Contemporary mobile phones are armed with several radio-frequency technologies, e.g., Global Positioning System (GPS), Wi-Fi, and Bluetooth. GPS has enabled accurate, ubiquitous outdoor positioning, while the inability of the satellite signals to penetrate buildings necessitates other techniques that can be used for accurate indoor positioning.

Bluetooth 4.0 has introduced a newly designed Bluetooth subsystem, referred to as Bluetooth Low Energy (BLE) [2], which is designed for battery-powered portable devices running for several years with built-in coin-cell battery. BLE has since attracted enormous attention mainly for its usage in indoor location-aware applications from both academy and industry. Specifically, Apple has announced a new protocol called iBeacon [44] on top of advertising and scanning modes of BLE for industry-wide indoor location-aware services. Afterwards, various vendors have made iBeacon-compatible BLE transmitters—typically called BLE beacons—to help developers release their own location-aware services. It already has comparable market penetration due to the support of wide range of contemporary user devices.

Most indoor localization systems employ Received Signal Strength Indication (RSSI) fingerprinting method, exploiting temporal stability of RSSI of wireless signals. In particular, at every known location, the RSSI samples collected from a set of pre-deployed beacons form an RSSI signature of the target device, which is compared with those of surveyed sites to infer corresponding location with the most similar RSSI signature.

There have been many papers dealing with the accuracy of BLE localization performance [4–9]. However, an often overlooked problem is the impact of BLE packet loss on the performance of the localization system. As a technology at 2.4 GHz ISM band, BLE can be seriously affected by other technologies utilizing the same frequency band, e.g., Wi-Fi, microwave oven, and Zigbee, thus resulting in insufficient RSSI samples required in localization algorithm.

In this chapter, we propose RESCUE, a Carrier Frequency Offset (CFO)-based BLE packet detection scheme, where the unaffected part of an partially corrupted BLE packet is detected and utilized to retrieve the corresponding RSSI information. The packet detection is further processed to determine the transmitter of the packet based on the timing information of the detection.

In summary, we claim the following four contributions.

- We find out BLE packets can be detected by CFO inspection even when the packets are partially corrupted due to ambient interference.

- We propose RESCUE as a framework to utilize the CFO-based packet detection scheme and identify the detected packets utilizing timing information of the detection.

- We corroborate the performance gain delivered by RESCUE via experiments with an indoor localization system prototype implementation.

The rest of this chapter is organized as follows: in Section 4.2, we provide preliminary knowledge. In Section 4.3, we present an overview of RESCUE and introduce the main functional blocks of RESCUE. We present the experimental results in Section 4.4, and finally conclude this chapter in Section 4.5.

4.2 Preliminaries

4.2.1 Channelization

There are 40 BLE channels defined in 2.4 GHz ISM band ordered from number 0 to 39 with 2 MHz spacing. The open 2.4 GHz ISM band that is used by BLE is filled with many other wireless technologies, e.g, Wi-Fi, Zigbee, and microwave ovens. These radio activities may interfere with BLE. The BLE channels used in localization, referred to as *advertising channels* (i.e., channels 37, 38, and 39), have been chosen not to collide with the three most commonly used Wi-Fi channels, i.e., channels 1, 6, and 11, as shown in Fig. 4.1. However, in many countries, e.g., South Korea, the most commonly used Wi-Fi channels are channels 1, 5, 9, and 13, such that the three BLE advertising channels can still be interfered by the Wi-Fi signals at the four commonly used Wi-Fi channels.

4.2.2 Advertising and Scanning

The BLE standard defines two types of operations, i.e., 1) broadcasting advertisement messages without connection and 2) connection-based data communications. BLE-based localization system is based on the former type that there is a device, referred to as *advertiser*, broadcasting advertising packets periodically at advertising channels and one or more devices, referred to as *scanners*, scan advertising channels periodically to capture the advertising packets. BLE advertisers are often called BLE beacons, which are usually installed in public area to enable localization.

Fig. 4.2 illustrates how advertiser and scanner work. A BLE advertiser broadcasts an advertising packet at each advertising channel sequentially during each *advertising*

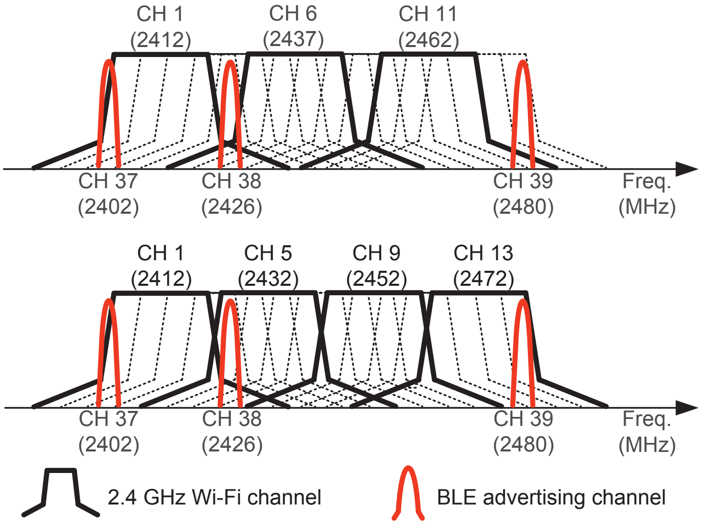


Figure 4.1: Overlapping pattern of 2.4 GHz Wi-Fi channels and BLE advertising channels.

event. In Fig. 4.2, I_{A} indicates advertising interval, i.e., the time between the starts of two consecutive advertising events, and it is the sum of a constant value C_{A} and a random value R_{A} ranging from 0 ms to 10 ms generated for each advertising event. Besides, a scanner scans the advertising channels sequentially with *scan interval* I_{S}. During each scan interval, the scanner continuously conducts idle listening at a certain advertising channel for the duration of *scan window* W_{S}. When the scanning operation is conducted at the same time and the same channel as those of an advertising packet, the scanner can receive the advertising packet.

4.2.3 GFSK Modulation

BLE employs Gaussian Frequency Shift Keying (GFSK) modulation, where the bandwidth of the transmitted FSK signal is deliberately reduced by Gaussian filter. The functional blocks of a typical GFSK modulator is shown in Fig. 4.3. A passband BLE

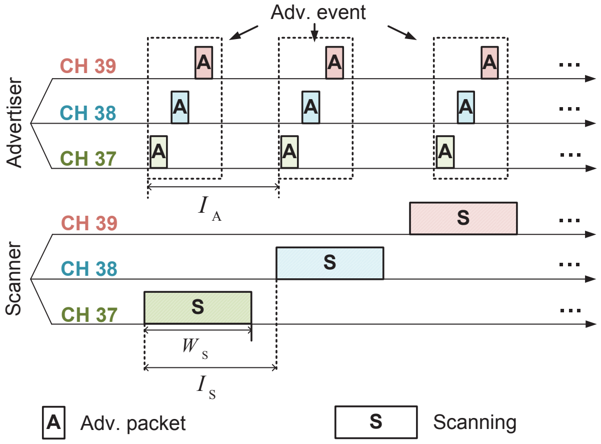


Figure 4.2: Illustration of advertising and scanning.

signal can be represented as

$$s(t) = \sqrt{\frac{2E}{T}} \cos[2\pi f_c t + \theta(t) + \theta_0], \tag{4.1}$$

where E, T, f_c, and θ_0 indicate symbol energy, symbol period, carrier frequency, and arbitrary initial phase, respectively. The phase deviation $\theta(t)$ is determined by the input data sequence $x[t]$ with $x[i] \in \{\pm 1\}$, which is expressed as

$$\theta(t) = \frac{\pi h}{T} \int_{-\infty}^{t} \sum_{n=-\infty}^{\infty} x[n] r\left(\tau - nT\right) d\tau, \tag{4.2}$$

where h[1] and $r(\cdot)$ indicate modulation index and Gaussian pulse shaping function, respectively. The pulse shaping function $r(\cdot)$ is expressed as

$$\begin{aligned} r(t) = Q\left(\sqrt{2}\alpha T\left(-\frac{1}{2} - \frac{t}{T}\right)\right) \\ - Q\left(\sqrt{2}\alpha T\left(\frac{1}{2} - \frac{t}{T}\right)\right), \end{aligned} \tag{4.3}$$

where $Q(\cdot)$ represents Q-function, and α is $5.336B$, where B is the 3 dB bandwidth of the Gaussian filter. Fig. 4.4 illustrates the pulse shaping function as a function of t/T

[1] The BLE standard defines that modulation index is between 0.45 and 0.55.

for several values of BT product.[2] One observation from these curves is that the Inter Symbol Interference (ISI) introduced by the Gaussian pulse shaping filter extends to one adjacent (on each side) symbol in the case of BLE using $BT = 0.5$.

A natural solution of GFSK demodulation is to utilize phase detector and frequency discriminator as shown in Fig. 4.3. We denote the ISI version of $x(t)$ as

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} x[k]r(t - kT). \tag{4.4}$$

Then, the baseband in-phase (I) and quadrature (Q) components can be expressed as

$$I(t) = \cos\left(\frac{\pi h}{T}\int_{-\infty}^{t} \tilde{x}(\tau)d\tau + \theta_0\right), \tag{4.5}$$

and

$$Q(t) = \sin\left(\frac{\pi h}{T}\int_{-\infty}^{t} \tilde{x}(\tau)d\tau + \theta_0\right), \tag{4.6}$$

respectively. The output of the frequency discriminator can be expressed as

$$\begin{aligned}\tilde{f}(t) &= \frac{d}{dt}\left(\tan^{-1}\frac{Q(t)}{I(t)}\right)\\ &= \frac{\pi h}{T}\sum_{k=-\infty}^{\infty} x[k]r\left(t - kT\right),\end{aligned} \tag{4.7}$$

which can be used to determine transmitted data $x[n]$ by comparing the samples of $\tilde{f}(t)$ with the DC level in the middle of each symbol.

4.2.4 CFO Estimation

The output of the comparator of the GFSK demodulator can be affected by the CFO between the transmitter and the receiver. Correspondingly, contemporary GFSK transceivers are equipped with Automatic Frequency Control (AFC) module that is capable of measuring CFO based on the incoming signal to automatically compensate the CFO for better demodulation [45].

[2] BT product is the product of the 3-dB bandwidth *B* and the symbol period *T*, and it is the main parameter to describe GFSK modulation. For example, *BT* 0.3 is used for GSM and *BT* 0.5 is used for classic Bluetooth and BLE. The lower the *BT* value is, the narrower the signal bandwidth is.

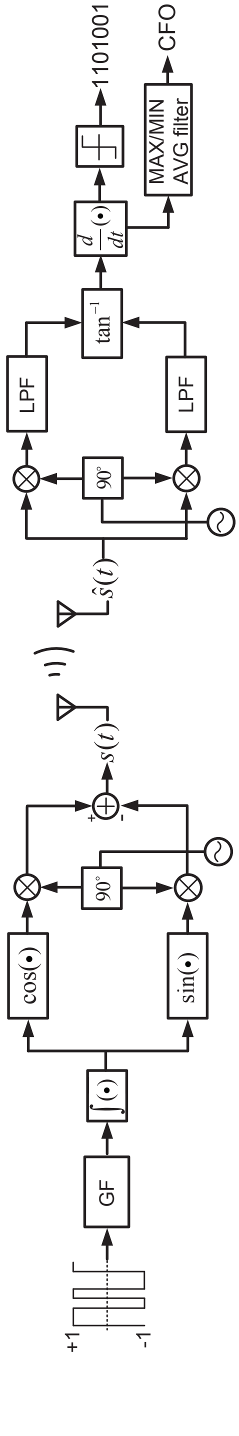


Figure 4.3: Modulation and demodulation procedure of GFSK signal.

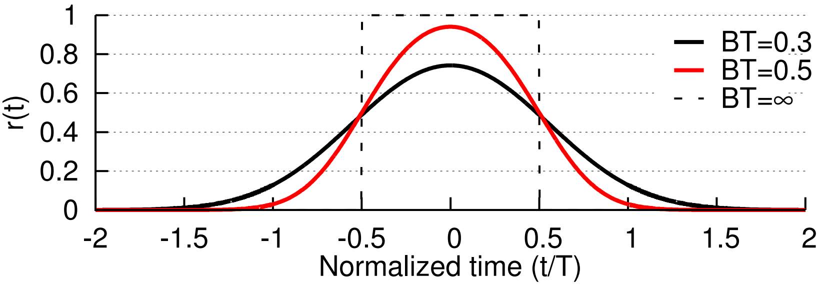


Figure 4.4: Gaussian pulse shaping function.

In conventional Bluetooth receiver, the CFO is estimated by taking the average of the detected maximum and minimum levels of the incoming baseband signal, which is referred to as *MAX/MIN algorithm* [46, 47].

4.2.5 Data Whitening

The BLE standard [2] defines that BLE transmitter should perform *whitening* before transmitting a BLE packet over the air in order to avoid long sequences of zeros or ones. Accordingly, *de-whitening* is performed at the receiver to recover the original packet payload.

The whitening and de-whitening functions are defined in the same way that a 7-bit linear feedback shift register with the polynomial x^7+x^4+1 is used. Before whitening and de-whitening, the shift register is initialized with a sequence that is derived from the index of the channel used to transmit the packet.

4.3 Proposed Scheme

We propose RESCUE with the following considerations.

- It does not require the modifications of the functional blocks of the conventional BLE transceiver.

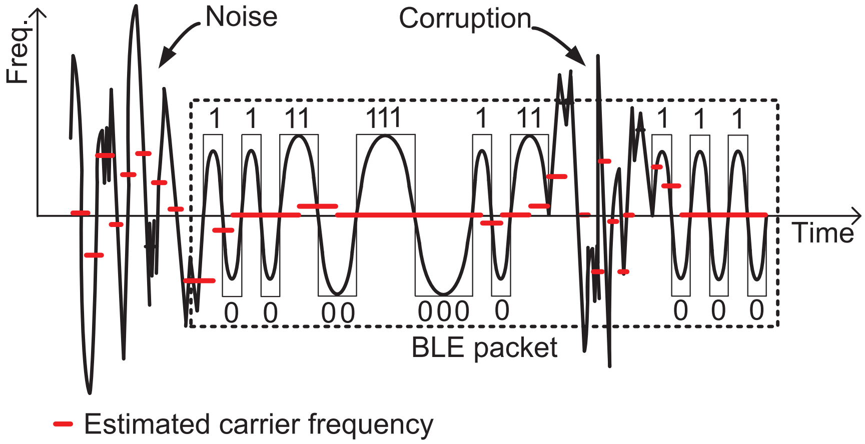


Figure 4.5: Estimated carrier frequency illustration.

- BLE packet detection and RSSI acquisition are attained even with partially corrupted BLE packets.
- Its performance gain in indoor localization system should be corroborated in real wireless environment.

4.3.1 Repetition Payload for Stable CFO

In RESCUE, we propose to detect a BLE packet by observing CFO values measured from the incoming signal. Since the noise and interfered part of the packet tend to produce random CFO values, we can detect the unaffected part of a BLE packet based on several consecutive CFO values confined within a certain range, as illustrated in Fig. 4.5

Note that, in MAX/MIN algorithm, the detected maximum and minimum levels of the GFSK signal, are strongly affected by the contents of the GFSK signal due to the ISI introduced by the Gaussian filter. For example, the maximum level produced by 11 is higher than the maximum level produced by a single 1, and the minimum level produced by 00 is lower than the minimum level produced by a single 0. To

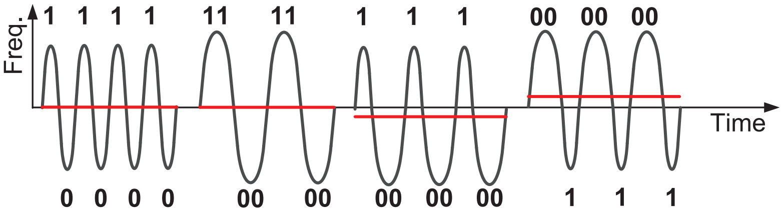


Figure 4.6: Bit repetition pattern leads stable CFO values.

obtain stable CFO values, we need *bit repetition pattern*, where the maximum and the minimum levels of the GFSK signal are constant based on the repetition of the consecutive 1 and 0 patterns as illustrated in Fig. 4.6.

In order to make CFO values more stable, we propose to include *repetition payload* in BLE advertising packet. Repetition payload is defined as the payload which results in bit repetition pattern after whitening. In most beacon packet formats, there are fields that are allowed to be modified by the developer to customize the packet for their own services. For example, in the case of iBeacon, there are three fields, i.e., *proximity UUID*, *major*, and *minor*, allowed to be modified. These fields are modified as repetition payload as illustrated in Fig. 4.7.

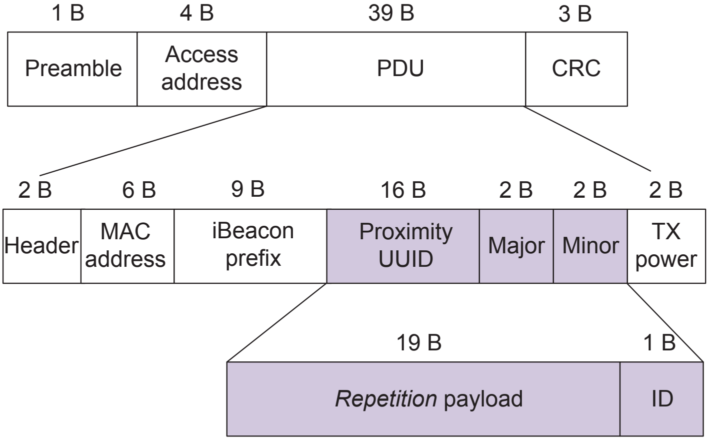
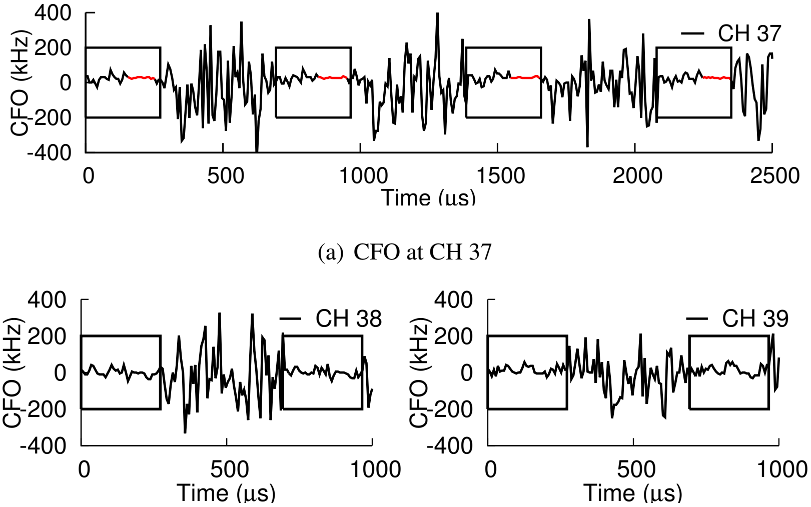


Figure 4.7: Proposed iBeacon format.



(a) CFO at CH 37

(b) CFO at CH 38

(c) CFO at CH 39

Figure 4.8: Estimated CFO at BLE channels 37, 38, and 39.

4.3.2 Dedicated Channel

On the other hand, the BLE standard [2] defines that the channel index is used to initialize the shift register used in whitening and de-whitening. That is, a certain repetition payload can be used to generate bit repetition pattern at only one advertising channel.

Fig. 4.8 illustrates the CFO values estimated at three BLE advertising channels, while the repetition payload is designed to generate bit string 1010...10 after whitening at channel 37. In the case of channel 37, the stable CFO values resulted from repetition payload are highlighted with red color while in the cases of channels 38 and 39, the CFO values fluctuate even during the BLE packet reception as highlighted with black boxes. We define the BLE advertising channel dedicated by repetition payload as *dedicated channel*.

4.3.3 BLE Packet Detection and RSSI Acquisition

CFO is attributed to two important factors: frequency mismatch in the transmitter and the receiver oscillators; and the Doppler effect due to the mobility of the transmitter

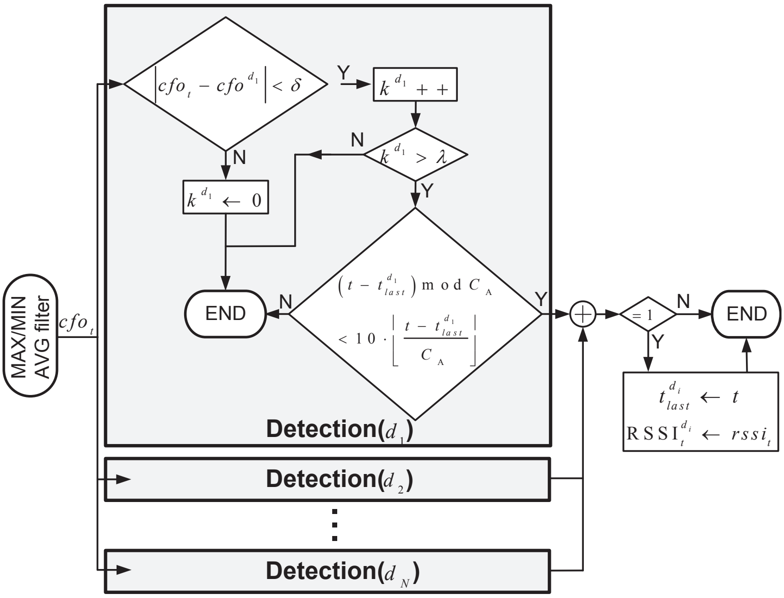

MAX/MIN AVG filter
cfo_t
$\left|cfo_t - cfo^{d_1}\right| < \delta$
Y
$k^{d_1} + +$
N
$k^{d_1} > \lambda$
N
Y
$k^{d_1} \leftarrow 0$
END
N
$\left(t - t_{last}^{d_1}\right) \bmod C_A < 10 \cdot \left\lfloor \frac{t - t_{last}^{d_1}}{C_A} \right\rfloor$
Y
$= 1$
N
END
Y
$t_{last}^{d_i} \leftarrow t$
$\mathrm{RSSI}_t^{d_i} \leftarrow rssi_t$
Detection(d_1)
Detection(d_2)
Detection(d_N)

Figure 4.9: Proposed packet detection algorithm.

and/or the receiver. In indoor environment, the CFO resulted from Doppler effect is relatively minor.

On the other hand, the CFO caused by the frequency mismatch between oscillators in the transmitter and receiver is relatively stable [48–50]. RESCUE exploits the stability of the CFO that when the number of consecutive CFO values having confined difference, i.e., $(-\delta, +\delta)$, with that of a previously known BLE beacon is greater than a certain threshold λ, it judges that there may be a BLE packet from the BLE beacon. Then, since it is not sufficient simply to detect a packet when there are more than one neighboring BLE beacon, RESCUE exploits the periodic nature of the advertising packets such that when a BLE packet is detected, it calculates the time elapsed since the last detected time for each neighboring beacon. If the elapsed time is within the allowed time interval for only one beacon, RESCUE will notify the detection result and corresponding RSSI value to the localization system. Fig. 4.9 shows the logic flow of the detection algorithm used in RESCUE.

4.4 Performance Evaluation

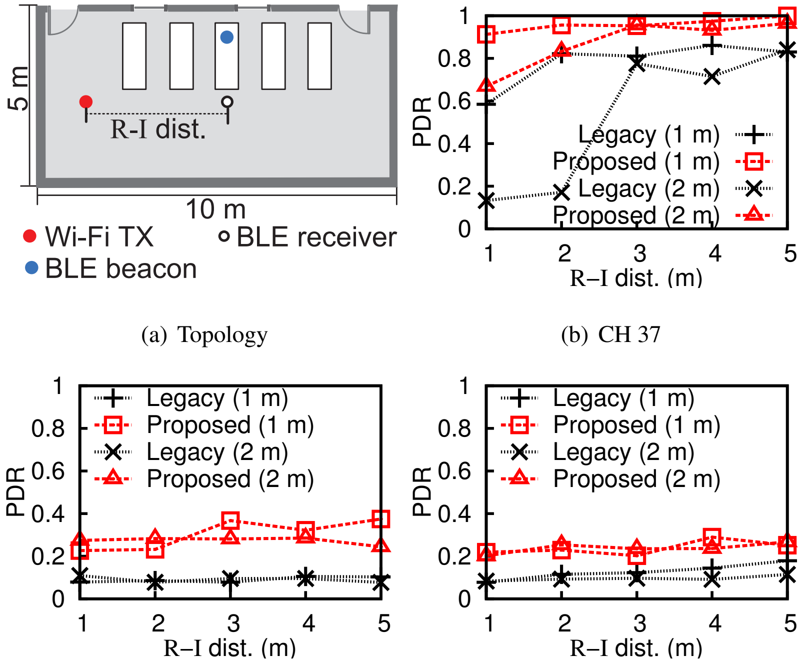
We developed a prototype of RESCUE receiver on Ubertooth One platform [38]. We chose Estimote [51] as BLE beacon and modified the configurable fields as repetition payload for advertising channel 37. We conducted experiments to corroborate the gain of RESCUE in terms of BLE packet detection. A BLE packet is detected as long as the existence of a BLE packet is detected and accurate RSSI information of the packet is retrieved at the receiver.

The experiment was conducted under the topology shown in Fig. 4.10(a). There are three Wi-Fi stations (Galaxy Nexus smartphones) receiving fully loaded downlink traffic from the associated Wi-Fi APs (ipTIME A2004NS-R) operating at Wi-Fi channels 1, 5, and 13, respectively. The position of the three collocated[3] Wi-Fi APs is indicated as a red circle, and a BLE receiver attempts to detect the BLE packets from a BLE beacon placed 1 m or 2 m away.

Figs. 4.10(b), 4.10(c), and 4.10(d) show the Packet Detection Rate (PDR) results of the BLE receiver, when the BLE beacon and the BLE receiver operating at channels 37, 38, and 39, respectively. The PDR results are shown with the distance between Wi-Fi AP and the BLE receiver varies from 1 m to 5 m. As expected, RESCUE significantly improves PDR, while the overall PDR results are much worse in the cases of channels 38 and 39 due to the closer center frequencies between BLE channel and interfering Wi-Fi channel. We advocate that when RESCUE is adopted, the BLE receiver collects much more RSSI samples from neighboring BLE beacons.

Finally, in order to evaluate actual gain delivered by RESCUE in indoor localization system, we constructed an indoor localization system in office environment with five BLE beacons installed randomly as shown in Fig. 4.11. We divided the office room into nine equal cells and constructed an RSSI map for each survey site located at the center of each cell. We installed an Wi-Fi AP at Wi-Fi channel 1 with fully loaded downlink traffic to generate interference to BLE channel 37, which is chosen as the

[3] 'Collocated' means being placed at the same position.



(a) Topology

(b) CH 37

(c) CH 38

(d) CH 39

Figure 4.10: PDR experimental results.

dedicated channel of RESCUE.

We let five BLE beacons transmit advertising packets at three advertising channels periodically with the constant part of the advertising interval C_{A} equal to 2 seconds. We let a BLE receiver collect RSSI samples from neighboring BLE beacons and update the location every 2 seconds with the RSSI samples collected during the last 2 seconds. The location is estimated based on RSSI fingerprinting method, i.e., the receiver selects the closest survey site among the nine survey sites in RSSI domain. The localization performance is evaluated in terms of *localization error* defined as the distance between the estimated survey site and the ground truth.

We conducted experiments at each survey site for about 30 minutes. During each location update window, we let the BLE scanner estimate its location based on the RSSI samples collected from neighboring BLE beacons.

Fig. 4.12 shows the empirical CDFs of the localization errors resulted from the nine

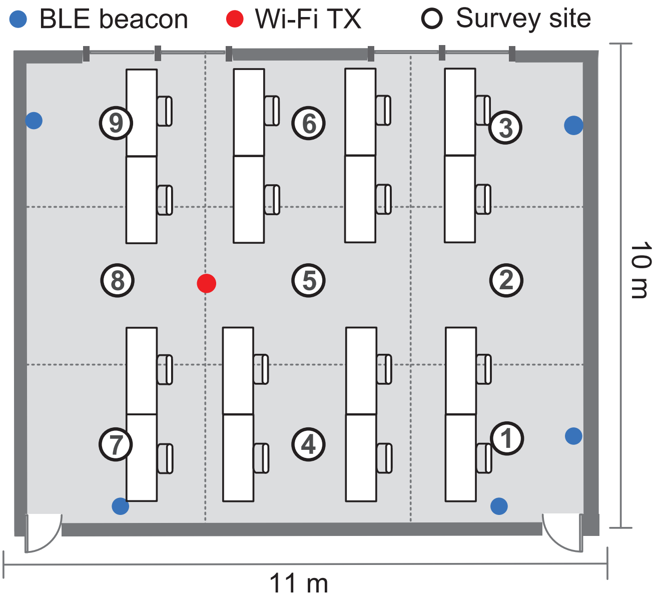
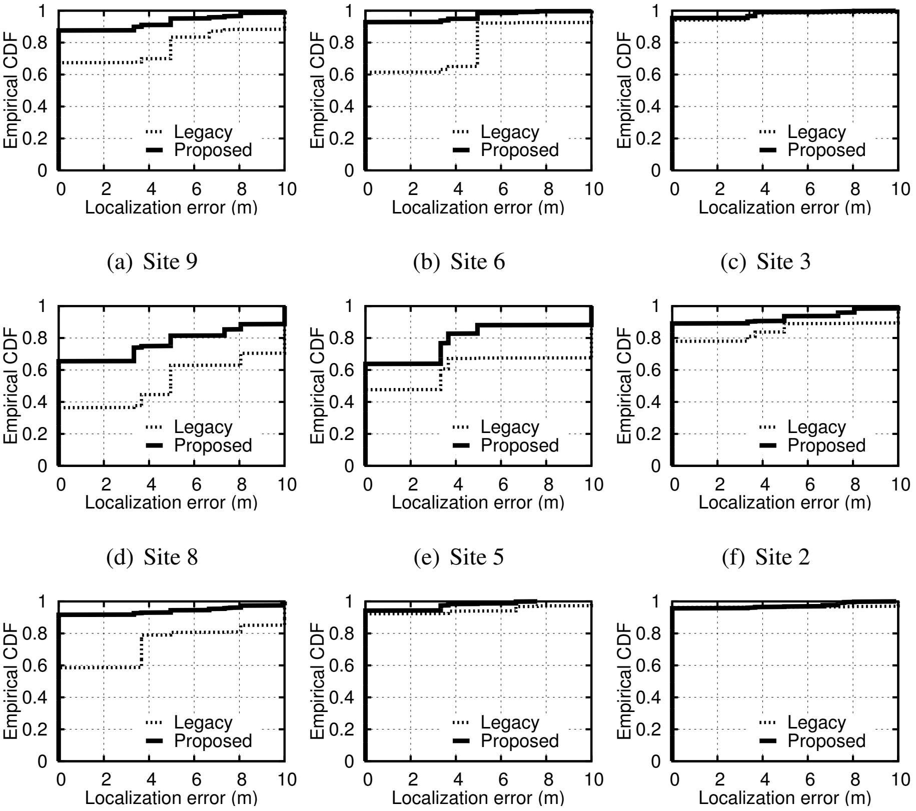


Figure 4.11: Localization experiment topology.

survey sites. We observe that at the sites near the Wi-Fi AP, e.g., sites 5, 6, 7, 8, and 9, the probability that the BLE receiver can accurately detect its location, i.e., localization error is 0 m, decreases abruptly compared to the other sites far away from the Wi-Fi AP, e.g., site 1, due to the lack of RSSI samples incurred by Wi-Fi interference. RESCUE improves the localization performance compared with the legacy scheme at these sites since it retrieves more RSSI samples from partially corrupted BLE packets.

In order to clarify the relationship between the number of the RSSI samples used in localization and the accuracy of the estimated location, location update windows of the traces collected at the nine survey sites are classified based on the number of available RSSI samples as shown in Fig. 4.13(a). The corresponding average localization error and the variance shown in Fig. 4.13(a) indicate that fewer RSSI samples result in increasing localization error and variation. A location estimation would be very imprecise if it is only based on zero or one RSSI sample due mainly to the fact that several survey sites share similar RSSI maps for a single beacon. The insufficient



(a) Site 9 (b) Site 6 (c) Site 3

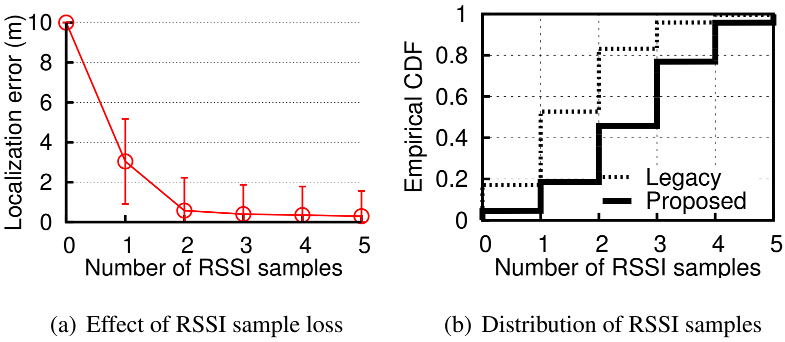
(d) Site 8 (e) Site 5 (f) Site 2

(g) Site 7 (h) Site 4 (i) Site 1

Figure 4.12: Experimental results in terms of localization error (with 5 BLE beacons).

RSSI information due to the RSSI sample losses would cause larger localization errors.

Fig. 4.13(b) shows the empirical CDF of the number of RSSI samples collected by the BLE receiver in location update windows in the cases of the legacy scheme and RESCUE, respectively. Fig. 4.13(b) indicates that the BLE receiver collects much more RSSI samples in each location update window with RESCUE compared with the legacy scheme. Therefore, with RESCUE, more accurate location estimation is achieved.



(a) Effect of RSSI sample loss (b) Distribution of RSSI samples

Figure 4.13: Impact of RSSI samples on localization performance.

4.5 Summary

In this chapter, we have proposed RESCUE to efficiently utilize unaffected part of partially interfered BLE packets to retrieve the RSSI information used in BLE-based indoor localization system. The performance gain is corroborated via prototype-based experiments. As future work, we plan to introduce an algorithm to evenly distribute the advertising packet transmission times of multiple neighboring BLE beacons to make multiple beacons more differentiable in time domain.

Chapter 5

Conclusion

5.1 Research Contributions

In this dissertation, we dealt with the performance degradation of wireless connectivity technologies caused mainly by ambient homogeneous and/or heterogeneous interference at the 2.4 GHz unlicensed band.

In Chapter 2, we analyze the stochastic behavior of the two Wi-Fi Direct devices in the legacy find phase, based on the absorbing Markov chain-based statistical model. We validate the effectiveness of the Markov model via both simulation and measurement. Moreover, we improve the legacy find phase by proposing Listen Channel Randomization (LCR), of which the significant performance gain is verified theoretically and practically.

In Chapter 3, we propose BlueCoDE to mitigate the performance degradation of Wi-Fi and BT in both WBH and NBH environments. The performance gain of BlueCoDE compared with the legacy scheme is demonstrated via analysis, simulation, and prototype-based experiments.

In Chapter 4, we propose RESCUE to utilize uncorrupted part of partially interfered BLE packets to retrieve the RSSI information in order to make the BLE-based indoor localization system more interference-resilient. The performance gain delivered by BLE packet detection and RSSI acquisition capabilities in localization system are corroborated via prototype-based experiments.

5.2 Future Research Directions

Based on the results of this dissertation, there are several new research directions which require further investigation. We highlight some of them as follows.

First, regarding the faster Wi-Fi Direct device discovery, as future work, we plan to devise an adaptive listen channel selection scheme by dynamically adjusting the selection probability of each social channel according to the surrounding channel status. We envision further substantial performance improvement via such adaptive scheme.

Second, regarding the Bluetooth coordination for better coexistence, we plan to consider multi-slot BT packets and multi-coordinator environments. Besides, a topology-aware hopping sequence manipulation will be considered to exploit spatial reuse effect to utilize wireless resources more efficiently for further improvements.

Finally, regarding the robust BLE packet detection, we plan to introduce an algorithm to evenly distribute the transmission times of advertising packets from multiple ambient BLE beacons so that multiple beacons can be differentiated more easily in time domain.

Bibliography

[1] I. std., *IEEE 802.11-2012, Part 11: Wireless LAN medium access control (MAC) and physical Layer (PHY) specifications*, 2012.

[2] B. SIG, *Specification of the Bluetooth system version 4.2*, 2014.

[3] J. Lansford, A. Stephens, and R. Nevo, "Wi-Fi (802.11b) and Bluetooth: enabling coexistence," *IEEE Network*, vol. 15, no. 5, pp. 20–27, 2001.

[4] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does btle measure up against wifi? a comparison of indoor location performance," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of.* VDE, 2014, pp. 1–6.

[5] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin, "Ble localization using rssi measurements and iringla," in *Industrial Technology (ICIT), 2015 IEEE International Conference on.* IEEE, 2015, pp. 2178–2183.

[6] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, no. 5, p. 596, 2016.

[7] J.-W. Qiu, C.-P. Lin, and Y.-C. Tseng, "Ble-based collaborative indoor localization with adaptive multi-lateration and mobile encountering," in *Wireless Communications and Networking Conference (WCNC), 2016 IEEE.* IEEE, 2016, pp. 1–7.

[8] R. Faragher and R. Harle, “Location fingerprinting with bluetooth low energy beacons,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[9] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, “Alps: A bluetooth and ultrasound platform for mapping and localization,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 73–84.

[10] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device-to-device communications with Wi-Fi Direct: overview and experimentation,” *Wireless Communications, IEEE*, vol. 20, no. 3, pp. 96–104, 2013.

[11] H. Zhang, Y. Wang, and C. C. Tan, “WD2: an improved WiFi-direct group formation protocol,” in *Proc. ACM MobiCom*, 2014.

[12] S. Trifunovic, A. Picu, T. Hossmann, and K. A. Hummel, “Slicing the battery pie: fair and efficient energy usage in device-to-device communication via role switching,” in *Proc. ACM MobiCom’13*, 2013.

[13] D. Camps-Mur, X. Pérez-Costa, and S. Sallent-Ribes, “Designing energy efficient access points with Wi-Fi Direct,” *Elsevier ComNet*, vol. 55, no. 13, pp. 2838–2855, 2011.

[14] H. Yoon and J. W. Kim, “Collaborative streaming-based media content sharing in WiFi-enabled home networks,” *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2193–2200, 2010.

[15] I. Ashraf, K. Voulgaris, A. Gkelias, M. Dohler, and A. H. Aghvami, “Impact of interfering Bluetooth piconets on a collocated-persistent CSMA-based WLAN,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 4962–4975, 2009.

[16] C. Chiasserini and R. Rao, "Coexistence mechanisms for interference mitigation between IEEE 802.11 WLANs and Bluetooth," in *Proc. IEEE INFOCOM*, 2002.

[17] I. std., *Part 15.2: Coexistence of wireless personal area networks with other wireless devices operating in unlicensed frequency bands*, Aug. 2003.

[18] A. Hsu, D. Wei, C. Kuo, N. Shiratori, and C. Chang, "Enhanced adaptive frequency hopping for wireless personal area networks in a coexistence environment," in *Proc. IEEE GLOBECOM*, 2007.

[19] J. Li and X. Liu, "A frequency diversity technique for interference mitigation in coexisting Bluetooth and WLAN," in *Proc. IEEE ICC*, 2007.

[20] N. Amanquah and J. Dunlop, "Improved throughput by interference avoidance in co-located Bluetooth networks," in *Proc. IET European Personal Mobile Communications Conference*, 2003.

[21] Z. Jiang, V. C. Leung, and V. W. Wong, "Reducing collisions between Bluetooth piconets by orthogonal hop set partitioning," in *Proc. IEEE RAWCON*, 2003.

[22] S.-Y. Lau, T.-H. Lin, T.-Y. Huang, I.-H. Ng, and P. Huang, "A measurement study of zigbee-based indoor localization systems under rf interference," in *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization*. ACM, 2009, pp. 35–42.

[23] "Xbox One: A Modern, Connected Device," http://news.xbox.com/2013/06/connected.

[24] *Wi-Fi Display Technical Speficication v1.2*, Wi-Fi Alliance, 2011.

[25] S. J. *et al.*, "Fast scanning schemes for IEEE 802.11 WLANs in virtual AP environments," *Elsevier ComNet*, vol. 55, no. 10, pp. 2520–2533, 2011.

[26] S. Jin and S. Choi, "A seamless handoff with multiple radios in IEEE 802.11 WLANs," *IEEE Trans. Veh. Technol.*, vol. 63, no. 3, pp. 1408–1418, 2014.

[27] I. Ramani and S. Savage, “SyncScan: practical fast handoff for 802.11 infrastructure networks,” in *Proc. IEEE INFOCOM*, 2005.

[28] *Wi-Fi Peer-to-Peer (P2P) Technical Specification, ver. 1.1*, Wi-Fi Alliance, P2P Technical Group, Oct. 2010.

[29] C. M. Grinstead and J. L. Snell, *Introduction to probability*. American Mathematical Soc., 2012.

[30] NS-3 Consortium, “The NS-3 network simulator,” https://www.nsnam.org, 2015.

[31] Plantronics, “A new look at Bluetooth density for the office: expanded simulation results deliver real-world opportunities for audio deployments,” *White Paper*, 2013.

[32] R. Chokshi, “Yes! Wi-Fi and Bluetooth can coexist in handheld devices,” *Marvell White paper*, 2010.

[33] X. Hu, L. Song, D. Van Bruggen, and A. Striegel, “Is there wifi yet?: How aggressive probe requests deteriorate energy and throughput,” in *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*. ACM, 2015, pp. 317–323.

[34] Texas Instruments, “WiLink 8 solutions,” http://processors.wiki.ti.com/images/1/13/WiLink8_-_TI_Coexistense_Tutorial_1.pdf, 2013.

[35] Qualcomm, “QCA6234 integrated dual-band 2x2 802.11n + Bluetooth 4.0,” 2016.

[36] Broadcom, “BCM4325 Bluetooth and WLAN Coexistence,” *White paper*, 2008.

[37] Wikipedia, “Timing Synchronization Function,” https://en.wikipedia.org/wiki/Timing_Synchronization_Function, 2017.

[38] D. Spill, “Project Ubertooth,” http://ubertooth.sourceforge.net/, 2015.

[39] T. Lin, Y. Liu, and Y. Tseng, "An improved packet collision analysis for multi-Bluetooth piconets considering frequency-hopping guard time effect," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 10, pp. 2087–2094, 2004.

[40] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 159–170, 2010.

[41] O. Project, "OpenWrt," https://openwrt.org/, 2016.

[42] L. Wireless, "Minstrel rate control algorithm," https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel, 2016.

[43] R. Mautz, "Indoor positioning technologies," 2012.

[44] "iBeacon for Developers," https://developer.apple.com/ibeacon/.

[45] T. Instruments, "CC2400 2.4 GHz low-power RF transceiver," 2007.

[46] A. W. Payne, "Dc offset estimation," Nov. 15 2006, uS Patent App. 12/093,991.

[47] R. E. Ryter, "Apparatus for determining a frequency offset error and receiver based thereon," Jun. 8 2010, uS Patent 7,733,991.

[48] A. Pásztor and D. Veitch, "Pc based precision timing without gps," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1. ACM, 2002, pp. 1–10.

[49] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Transactions on Mobile Computing*, vol. 9, no. 3, pp. 449–462, 2010.

[50] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," 2016.

[51] "Estimote," http://estimote.com/.

초록

2.4 GHz 비면허대역은 정부에서 정한 최대 허용 출력 등 기술기준을 만족하면 누구나 사용할 수 있다. 비면허대역은 주로 저출력 근거리 무선 통신에 활용되며, 대표적인 기술들로는 Wi-Fi, Classic Bluetooth (BT), Bluetooth Low Energy (BLE) 등이 있다. 이러한 기술들은 주파수 독점이 아니라 주파수 공유 기반으로 동작하도록 설계되었다. 하지만 이종 기술들 간 동작 방식의 차이로 인하여 서로 호환이 불가능하고 간섭으로 인한 성능저하가 발생한다. 본 논문에서는 2.4 GHz 대역의 Wi-Fi, BT, BLE 기술들이 주변 간섭으로 인하여 생기는 성능저하를 대폭적으로 줄일 수 있는 해결책을 제시한다. 크게 아래와 같은 세 가지 연구 주제에 대해서 연구를 진행하였다.

최근 Wi-Fi 공유기 없이 Wi-Fi 기기들 간 직접 연결하여 통신할 수 있는 Wi-Fi Direct 기술이 등장했다. 이 기술의 상용화와 보편화를 결정하는데 있어서 중요한 이슈 중 하나가 기기 연결 시간인데 대부분의 연결 시간은 기기 발견 절차 (device discovery)에서 초래 된다. 우리는 Wi-Fi Direct 기기 발견 시간을 분석할 수 있는 Markov Chain 모델 및 주변 간섭이 존재하는 환경에서 기기 발견 시간을 효과적으로 줄일 수 있는 Listen Channel Randomization (LCR) 기법을 제안하였다. 제안 기법의 성능에 대해서 이론적 분석, NS-3 기반 시뮬레이션, 그리고 상용 스마트폰 기반의 구현 및 실측 결과를 통하여 실증 분석하였다.

한편, Wi-Fi와 BT의 공존 성능은 이 두 기술의 상용화에 있어서 중요한 이슈 중 하나이다. 해당 이슈의 가장 범용적인 해결책인 BT의 Adaptive Frequency Hopping (AFH) 기법은 BT가 Wi-Fi 간섭이 없는 채널에서만 동작하도록 하는 주파수 분리

기법이다. 하지만 모든 BT 채널에 Wi-Fi 간섭이 미치는 경우 및 여러 대의 BT 기기들이 Wi-Fi 간섭이 없는 일부 채널에서만 동작하는 상황에서 BT 기기들 사이의 충돌이 심해지는 경우, AFH만으로는 만족스러운 성능을 기대하기 어렵다. 우리는 주변의 BT 기기들의 frequency hopping 패턴을 조절하여 서로 간의 간섭을 제거하고, 주변의 Wi-Fi에 미치는 간섭을 최소화하는 BlueCoDE 기법을 제안하였다. 제안 기법의 성능은 이론적 분석, 시뮬레이션, 그리고 Ubertooth 오픈 소스 플랫폼 기반의 Prototype 구현 및 실험을 통하여 실증 분석하였다.

최근 대두되고 있는 BLE는 주로 실내측위 서비스에 많이 사용된다. BLE 기반의 실내측위는 BLE 수신 기기가 주변의 BLE 송신 기기들로부터 BLE 패킷을 정확하게 수신하여 수신 신호 세기를 활용할 수 있어야 가능하다. 하지만 주변의 Wi-Fi 간섭으로 BLE 패킷 손실이 발생하면 수신 신호 세기에 대한 정보 손실도 같이 발생하게 된다. 우리는 주변의 Wi-Fi 간섭에 강인한 BLE 패킷 검출 기법인 RESCUE를 제안하였고, 제안 기법을 Ubertooth 플랫폼에 구현하였다. 또한, 실 환경에서의 실내측위실험을 통하여 Wi-Fi 간섭이 존재하는 환경에서 제안 기법의 성능이 기존 성능보다 대폭 향상되었음을 검증하였다.

주요어: 2.4 GHz 비면허대역, 무선연결기술, 간섭 강인성

학번: 2010-24083