공학박사 학위논문

Environment-Aware Resource Management Strategies in IoT Protocols

IoT 프로토콜에서의 상황 인식 자원 관리 기법에 대한 연구

2022년 2월

서울대학교 대학원

전기 · 컴퓨터공학부

박 준 현

Environment-Aware Resource Management Strategies in IoT Protocols

IoT 프로토콜에서의 상황 인식
기반 자원 관리 기법에 대한 연구

지도교수 권 태 경

이 논문을 공학박사 학위논문으로 제출함
2021년 11월

서울대학교 대학원
전기 · 컴퓨터공학부
박 준 현

박준현의 공학박사 학위논문을 인준함
2021년 12월

위 원 장: 전 화 숙 (인)

부위원장: 권 태 경 (인)

위 원: 박 세 웅 (인)

위 원: 심 병 효 (인)

위 원: 최 재 혁 (인)

Abstract

Environment-Aware Resource Management Strategies in IoT Protocols

Junhyun Park
Department of Electrical Engineering & Computer Science
The Graduate School
Seoul National University

To bring a hyper-connected society, where a large number of devices are connected to each other, into reality, various Internet of Things (IoT) protocols have emerged. In particular, unlicensed band protocols such as LoRa/LoRaWAN, Wi-Fi, and Bluetooth are widespread from city-scale to local area services, as they enable impromptu establishment of public and private networks at low cost. However, the devices that communicate with heterogeneous IoT protocols presents research tasks of efficiently managing energy, computation, and radio resources while ensuring reliable connectivity. The various possible combinations of radio parameters, thereby induced complex trade-offs in link performance, and constantly changing wireless environments are the major factors making such tasks challenging. It often requires expert knowledge on wireless networks to accurately diagnose seemingly random radio properties and devise effective resource management strategies.

In this dissertation, we cover three topics, i.e., 1) transmission parameter tuning in LoRa/LoRaWAN, 2) Wi-Fi scan offloading via a collocated low-power radio, and 3) collaborative Wi-Fi and Bluetooth coexistence, where we efficiently manage device resources by exploiting useful information extracted from radio signals.

First, LoRa, which is one of the most promising low-power wide-area protocols operating in unlicensed bands, presents a set of transmission parameters that have to be properly regulated to manage radio resources and link performance. In this work, we present EARN, an enhanced Adaptive Data Rate (ADR) mechanism with Coding Rate (CR) adaptation, to optimize the trade-off between delivery ratio and energy consumption. In EARN design, we theoretically model the link performance of LoRaWAN to find the best parameter set and leverage the capture effect to increase the survival rate of colliding signals. We validate the feasibility of the CR adaptation with an empirical study, and large-scale simulations reveal that EARN outperforms the conventional schemes.

Secondly, Wi-Fi scanning, which is a fundamental feature to provide high-quality and seamless wireless connectivity in modern smart devices, induces performance overheads while unnecessarily probing on channels where no access points (APs) exist. We present C-SCAN, which exploits a collocated low-power wireless interface to offload the Wi-Fi scanning overheads. C-SCAN inspects the shared spectrum with a Bluetooth radio and identifies which Wi-Fi channels are in use prior to the actual Wi-Fi scanning. By excluding the channels determined to be empty, the Wi-Fi can perform scanning only on available Wi-Fi channels. We implement a prototype of C-SCAN using a Bluetooth-compliant wireless transceiver and demonstrate its efficiency. Experimental results show that C-SCAN achieves high detection accuracy with low latency and energy even in dense Wi-Fi environments.

Lastly, Wi-Fi and Bluetooth are collocated in modern smart devices and suffer from cross-technology interference (CTI) internally and externally, as they share a single antenna and spectrum. In this work, we present D-SCAN, a novel collaborative coexistence mechanism. D-SCAN infers nearby Wi-Fi information efficiently with a

Bluetooth radio, thereby offsetting the overhead of key Wi-Fi functions and preventing collisions between Wi-Fi and Bluetooth. To this end, D-SCAN adopts a data-driven approach that captures the unique temporal and spectral features of Wi-Fi signals from Bluetooth spectrum measurements by leveraging deep neural networks. D-SCAN prototype in real-world experiments reduces the latency and energy consumption of legacy Wi-Fi scanning, and it also promotes the agile interference avoidance of Bluetooth.

In summary, from Chapters 2 to 4, the aforementioned three research tasks, i.e., EARN for LoRa parameter distribution, C-SCAN for Wi-Fi scan offloading, and D-SCAN for Wi-Fi and Bluetooth coexistence, will be presented, respectively.

Keywords: Unlicensed bands, LoRa/LoRaWAN, Wi-Fi, Bluetooth, resource management, coexistence.

Student Number: 2013-23115

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Limited Resources of IoT Protocols at ISM Bands

The advances in information and communication technology (ICT) have brought the IoT paradigm, where a massive number of things connect to each other. To make the hyper-connected society a reality, various but distinct wireless technologies such as Cellular, Wi-Fi, Bluetooth, and LPWA are devised and sophisticated, complementing each other's technology gaps (e.g., bandwidth, coverage, etc.). In particular, IoT services based on protocols operating in the industrial, scientific, and medical (ISM) bands are prevalent in our daily lives from city-scale to local area due to the convenience in network establishment and management.

However, devices that communicate with these IoT protocols are often resource-constrained. For example, mobile smart devices run on batteries, as is the case with sensor devices that collect contextual data for environmental monitoring. The computational power of IoT devices is limited by network stakeholders to reduce the deployment costs. In addition, the radio spectrums shared by thousands of devices

and a different set of protocols should be managed wisely to offer reliable connectivity. To sum up, one of the major issues in IoT protocols is to efficiently manage communication system resources and hold balance between device lifespan and maintenance costs, while boasting high degree of sustainability.

There are several non-trivial challenges to be resolved. First, radio parameters, which is to modify link configuration, are complexly correlated with each other. As different possible combinations of the parameters generate the trade-offs in the performance, it is difficult to select one suitable configuration, which best fits a given link condition. Furthermore, the link condition constantly changes in wireless environments due to the network dynamics. Therefore, it often requires expert knowledge on wireless networks to accurately diagnose seemingly random radio properties and devise effective resource management strategies.

In this dissertation, we address resource management issues of three IoT protocols: LoRa/LoRaWAN, Wi-Fi, and Bluetooth, and present environment-aware strategies that diagnose and exploit complex radio properties.

1) Low Power Wide Area (LPWA) networks have emerged as a power-efficient and cost-effective choice of technologies for city-wide IoT. LoRa, one of the most promising unlicensed band techniques, is capable of providing reliable communications under harsh link conditions and, therefore, a higher chance of getting the combination of worthy performance and price. LoRa, however, cannot fully offer its efficiency and scalability in typical smart city deployments, as it is vulnerable to collisions due to the adoption of pure ALOHA mechanism and the lack of proper resource management strategies in LoRaWAN. LoRa/LoRaWAN presents a set of tunable transmission parameters, along with an Adaptive Data Rate (ADR) mechanism, to manage

radio resources and promote the best performance under the variable link state. But the performance of ADR, whose design neglects the complex correlation between such parameters, is yet to be practical.

2) Wi-Fi channel scanning—the task of searching for available channels at a given location—is a fundamental feature to maintain always-available and high-quality wireless connectivity in today's mobile devices. To ensure seamless connectivity, frequent Wi-Fi scanning on the full set of channels should be performed owing to user mobility and the short transmission range of Wi-Fi networks, which in turn leads to excessive battery drain. Furthermore, the halt of data transmission incurred by Wi-Fi scanning directly affects communication performance. The concept of selective Wi-Fi scanning, which scans only subset of channels where APs' presence is confirmed, can save significant amount of scanning latency and energy by reducing the active duration of the radio. However, it is a challenging task to design an intelligent scanning algorithm that can discover available APs in a short time period.

3) As the demand for efficient and versatile wireless connectivity is ever-growing, mobile IoT devices equipped with multiple heterogeneous radios have become prevalent. The most representative case would be combo-module solutions which integrate Wi-Fi and Bluetooth interfaces into a single SoC (System on Chip) circuit, due to their cost-effectiveness in terms of form factor. To alleviate the problem of Wi-Fi and Bluetooth interfering with each other internally, TDM (Time-Division Multiplexing) based medium sharing approaches have been applied, where Wi-Fi and Bluetooth take turns to access the 2.4 GHz ISM band. Unfortunately, however, significant performance degradation has been reported as collocated Wi-Fi and Bluetooth share an

antenna while operating in the same 2.4 GHz ISM band, generating a new interference pattern. The reason is mainly that the external and internal CTI mitigation schemes inadvertently impede each other and lead to the failure of both. More specifically, both protocols fail to deal with nearby interferers and their collision rates mutually increase, when they are used simultaneously.

1.2 Overview of Existing Approaches

1.2.1 Coding Rate Adaptation in LoRa/LoRaWAN

The study of LoRa parameter allocation mainly focuses on the distribution of spreading factors (SFs) in ADR improvements. Many have taken approaches to properly disperse the high demands for the lowest SFs to other SFs [1-5]. EXPLoRa [1] defined a fixed SF distribution ratio and assigned SFs to LoRa End-devices (EDs) in order of distance to the gateways (GWs) so that all available SFs have the same air occupancy time; in the meanwhile, only the maximum transmission power (TP) was employed. And others suggested controlling TP together with SF [3-5]. FADR [5] used genetic algorithms to find the optimal SF distribution ratio that equalizes the collision probability among all available SFs. It then adjusted TP to fit in a given SNR_{req} with a fixed signal-to-noise-ratio (SNR) margin. But unfortunately, none of the prior parameter allocation schemes or ADR proposals have noted the impact of CR, and the capture effect has also been neglected or misused.

1.2.2 Wi-Fi Scan Offloading via Bluetooth Radio

In 802.11 channel discovery process, researchers have attempted to make the best

use of a single Wi-Fi radio [6-11]. They searched for the optimal values for the scanning parameters, such as scanning intervals [6], min/max channel time [7], etc. However, since scanning parameter-based methods do not fit well for all users and environments, the concept of selective scanning has become popular [12]. In this method, Wi-Fi scans only a subset of channels with high probability of finding an AP based on previous scanning results, reducing the active duration of radio.

To identify available Wi-Fi channels in advance and avoid the inherently expensive nature of Wi-Fi by offloading parts of scan procedure, several approaches utilizing secondary coexisting radios, such as Bluetooth and ZigBee, have been proposed [13-18]. Zi-Fi [13] and BlueScan [14] search for beacon frames solely by analyzing statistics and periodicity of RSSI (Received Signal Strength Indicator) samples. However, it is challenging to grab the periodicity of beacon frames in a short period of time, and furthermore, energy signature created by beacons is susceptible to noise, making it a harder task.

1.2.3 Better Coexistence Between Wi-Fi and Bluetooth

Many mechanisms have been proposed over the past several years to resolve the network-level external CTI issue. These include the detection of heterogeneous interferers and prevention of their collisions [19-23]. Then, to alleviate the problem of Wi-Fi and Bluetooth interfering with each other internally within a device, TDM based medium sharing approaches have been proposed [24-26].

To improve the performance of collocated radios in a collaborative manner, there have been several attempts to utilize a low-power secondary radio to acquire Wi-Fi information. Wake-on-WLAN [17], Esense [13], and S-WOW [27] allowed cross-technology communications between ZigBee and Wi-Fi using special codes to

exchange their channel information. Others have suggested Wi-Fi and Bluetooth combo APs embed the Wi-Fi channel information into Bluetooth broadcast packets [28, 29]. To the best of our knowledge, however, these prior studies require significant amount of hardware/software modifications, fail to adapt to network dynamics, and cannot provide detailed Wi-Fi information.

1.3 Main Contributions

1.3.1 EARN: Enhanced ADR with Coding Rate Adaptation in LoRaWAN

We leverage several facts that prior researches have overlooked to devise an enhanced ADR scheme, EARN. First, there is no provision for proper CR assignment and its trade-offs. And the capture effect has also been neglected or misused, even though it can increase the survival rate of colliding frames. When the link performance of devices, taking these into account, is closely approximated, we are capable of adapting the parameters for a given link condition.

The main contributions of this work can be summarized as follows:

- We theoretically model the link performance of Class A unconfirmed mode LoRa/LoRaWAN, focusing on the impact of CR.
- We present EARN, an enhanced ADR mechanism that uses the developed models to predict link performances and assigns the best parameter set.
- We demonstrate that EARN outperforms the conventional methods in all aspects in the thorough evaluations based on large-scale simulations.

1.3.2 C-SCAN: Wi-Fi Scan Offloading via Collocated Low-Power Radios

Unlike the existing solutions, our proposed scheme C-SCAN is not only to predict the availability of Wi-Fi networks but also to pinpoint the available Wi-Fi channels by using a collocated Bluetooth radio.

The main contributions of this work can be summarized as follows:

- We present a novel approach, called C-SCAN, that harnesses a low-power radio collocated in a mobile device to identify available Wi-Fi channels. C-SCAN uses a Bluetooth radio, which is readily available in most modern mobile devices, and enables low-delay and energy-efficient Wi-Fi scanning.
- We conduct rigorous measurement studies and design an intelligent channel identification algorithm that pinpoints the Wi-Fi channel numbers based on the RSSI values measured over several narrowband Bluetooth channels.
- We demonstrate the feasibility and effectiveness of C-SCAN by implementing a prototype using an open source Bluetooth-compliant board, on an Android-based platform. The experiments demonstrate that C-SCAN detect available Wi-Fi channels with high accuracy in real-world environments.

1.3.3 D-SCAN: Toward Collaborative Multi-Radio Coexistence in Mobile Devices via Deep Learning

A large number of possible outcomes hinder the accurate classification by deep neural networks (DNNs) due to the high complexity. We overcome this challenge using a divide-and-conquer approach. D-SCAN divides the 2.4 GHz spectrum into

tractable narrow frequency bands to render training and decision-making manageable. We introduce a simple yet effective data representation method, termed *edge projection*. The edge projection preserves temporal and spectral correlations of Wi-Fi signals in 2-D matrices, and thus can leverage the CNN (Convolutional Neural Network) model effectively. We generalize D-SCAN, which uses a low-power radio to learn the distinguished features of wideband and narrowband signals, and demonstrates its utility in IoT devices over different frequency bands.

The main contributions are summarized as follows:

- We propose D-SCAN mechanism, which exploits a Bluetooth radio to efficiently obtain Wi-Fi channel information including Wi-Fi channel usage, signal strength, and channel utilization.
- We introduce a novel 2-D RSSI data representation and effectively extend deep learning models to learn temporal and spectral features of Wi-Fi signals.
- We implement a prototype of D-SCAN for three use cases, where the obtained Wi-Fi information is used to improve the efficiency of Wi-Fi scanning and handover, and to help Bluetooth promptly adapt to Wi-Fi interference.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows.

Chapter 2 presents EARN, an enhanced ADR mechanism with CR adaptation, to optimize the trade-off between delivery ratio and energy consumption. In EARN design, we leverage the capture effect to increase the survival rate of colliding signals We validate the feasibility of the CR adaptation with an empirical study, and large-scale simulations reveal that EARN outperforms the conventional schemes.

Chapter 3 presents C-SCAN, that exploits a low-power wireless interface integrated into the device to offload Wi-Fi scanning overhead. C-SCAN inspects channel information with a Bluetooth radio and identifies which Wi-Fi channels are in use, prior to the actual channel scanning with a Wi-Fi interface. By excluding the channels determined to be empty, the Wi-Fi can perform scanning only on available Wi-Fi channels. Experimental results show that C-SCAN achieves high detection accuracy with low latency and energy, even in dense Wi-Fi environments.

Chapter 4 presents D-SCAN, a novel Wi-Fi and Bluetooth collaborative coexistence mechanism. D-SCAN infers nearby Wi-Fi information with a collocated Bluetooth radio, thereby offsetting the overheads in key Wi-Fi functions and preventing collisions between Wi-Fi and Bluetooth. D-SCAN captures the unique temporal and spectral features of Wi-Fi signals by leveraging deep neural networks. Then, we evaluate D-SCAN prototype in terms of latency, energy, and robustness.

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

Adapted from Junhyun Park, Kunho Park, Hyeongho Bae, and Chong-Kwon Kim "EARN: Enhanced ADR with coding rate adaptation in LoRaWAN" IEEE Internet of Things Journal 7.12 (2020)

Chapter 2

EARN: Enhanced ADR with Coding Rate Adaptation in LoRaWAN

2.1 Introduction

The advances in information and communication technology have brought the Internet of Things (IoT) paradigm, where a massive number of things connect to each other. Recently, IoT has been expanding its application from small scale to city-wide IoT, such as smart metering and smart environmental monitoring. To allow power-efficient and cost-effective wide area connectivity, different LPWA protocols have emerged as promising solutions. The most representative protocols, LTE-M, NB-IoT, LoRa, and Sigfox, have the common characteristics of connecting thousands of devices at low data rates, but they also have their own capabilities to meet specific application needs. Licensed LPWA technologies, such as LTE-M and NB-IoT, present reliable communications at relatively higher data rates owing to cellular infrastructures. However, the protocols to grant accesses only to authenticated users are often so complicated that they may bring about 2,000

retransmissions and energy overheads. LPWA protocols on the unlicensed bands, such as LoRa and Sigfox[1], require sophisticated interference control, yet, enable the impromptu establishment of private and public networks at a much lower cost. In this work, we focus on LoRa/LoRaWAN, whose simple use of protocols and high availability have attracted immense attention from both academia and industry.

LoRa, with its robust modulation scheme, is capable of providing reliable communications under harsh link conditions and, therefore, a higher chance of getting the combination of worthy performance and price. And in efforts to develop practical solutions by embracing these benefits, many researchers have started questioning the scalability that LoRa can serve in real-world environments [30-32]. An experiment has revealed that LoRaWAN adopting pure ALOHA is vulnerable to collisions in typical smart city deployments, and only 64 devices can achieve a transmission success rate of 0.9 or higher [30]. Some studies [1, 4, 5] have mitigated collisions by distributing the traffic load of devices on parallel demodulation paths but also reduced the network coverage to less than half of the maximum communication range claimed by LoRaWAN specification. As such, highly constrained scalability of current LoRaWAN is only suitable for low-rate sparse applications. In this work, we aim to improve coverage and capacity of LoRaWAN, along with the efficient use of energy for battery-powered devices, so that it can meet the various needs of emerging future IoT applications.

In an environment where one base station serves a massive number of IoT devices, channel and radio resources should be utilized efficiently. To this end, LoRa presents various transmission parameters that manage the resources to regulate the link performance. Proper parameter configuration is arguably essential to provide stable

[1] Sigfox networks are operated by Sigfox and its regional partners.

link quality under different application scenarios, and fail to do so significantly drains device power and degrades network scalability. Typically, LoRa devices can be configured to use different carrier frequencies (CF), bandwidth (BW), spreading factors (SF), transmission powers (TP), and coding rates (CR). These parameters, rather than being independent, are complexly correlated with each other to affect link performance. Due to this nature, it is challenging to select one suitable configuration, which best fits a given link condition.

As a mean to adjust some parameters such as SF and TP, LoRaWAN exploits the Adaptive Data Rate (ADR) mechanism. However, it assigns transmission parameters to devices with a very conservative policy in a way that the constrained medium they share gets quickly saturated. Other ADR proposals [1, 5] have attempted to distribute the load evenly over different SFs, whose orthogonality is known to prevent inter-SF collisions, in efforts to optimize fairness between devices. But the fair distribution of SF does not account for energy efficiency, leaving edge devices to constantly suffer from short lifespan regardless of cell size and link condition. We have also found that they are only practical in small cells where all SFs are available. After all, we leverage several facts that prior researches have overlooked to devise our own enhanced ADR scheme, EARN. First, there is no provision for proper CR assignment and its trade-offs. And the *capture effect* has also been neglected or misused, even though it can increase the survival rate of colliding frames. When the link performance of devices, taking these into account, is closely approximated, we are capable of adapting the parameters for a given link condition.

The main contributions of this paper can be summarized as follows: we theoretically model the link performance of Class A unconfirmed mode LoRaWAN in terms of frame delivery ratio (FDR), goodput, and energy efficiency. In the meantime, we observe the tradeoffs of transmission parameters focusing on CR,

which has been overlooked in prior works; the adoption of non-default CRs leads to longer frame length but increases reliability to afford extended communication range. Then, we present EARN, an enhanced ADR mechanism that uses the developed models to predict link performances and assigns the best parameter set. In EARN design, a LoRaWAN server maintains an aggregated load status for each SF and SNR to estimate collision probabilities within the impact range of collision and the capture effect. Then by extension, EARN employs a concept of adaptive SNR margin to endure link variations, as a tight estimation of the link can severely degrade the performance in harsh environments. In the end, the thorough evaluations based on large-scale simulations, which we justify with our empirical pilot experiment, demonstrate that the proposed algorithm completely outperforms the conventional methods in all aspects.

The rest part of this chapter is organized as follows. Section 2.2 provides preliminaries and research trends on LoRa/LoRaWAN. In Section 2.3, we formulate the link performance models and analyze the impact of CR in detail. Then we explain EARN algorithm along with its extension in Section 2.4. Section 2.5 describes the simulation setup and compares EARN with other ADR schemes. Finally, we summarize this chapter in Section 2.6.

2.2 Preliminaries

2.2.1 LoRa/LoRaWAN

LoRa is a PHY layer developed by Semtech, which adopts Chirp Spread Spectrum (CSS) modulation for robust and long-range communications. LoRaWAN, an open standard maintained by the LoRa Alliance, operates on top of LoRa as the MAC

layer protocol [33]. A LoRaWAN network uses a star-of-stars topology consisting of three entities: network servers, gateways (GW), and end-devices (ED). A network server can have multiple gateways, each of which is connected via the IP network to the server and covers up to thousands of end-devices with LoRa wireless links.

In LoRaWAN, there are three types of end-devices: Class A, B, and C. Note that Class A devices receive downlinks only after uplink transmissions. In contrast, Class B and Class C devices schedule extra downlink opportunities; Class B exploits beacon-based synchronous receive windows, and Class C continuously listens for downlinks. In each class, a *confirmed* message must be acknowledged by the receiver, while an *unconfirmed* message does not require an acknowledgment.

2.2.2 Transmission Parameters

LoRaWAN EDs have several adjustable transmission parameters to adapt and best facilitate link performances; CFs are the central frequencies of LoRaWAN channels spread along wide range of spectrum from 137 MHz to 1,020 MHz abided by regional regulations. BW is the bandwidth around CF and is directly proportional to the transmission capacity. BW of 125 kHz is typically used out of three configurable options, 125, 250, and 500 kHz. SF, ranging from 7 to 12, denotes the number of raw bits that can be encoded in a symbol. It also affects the symbol rate of LoRa, $BW/2^{SF}$, where a symbol is modulated as a chirp signal of 2^{SF} chips and transmitted at a given chip rate (i.e. BW). As can be inferred here, the data rate is nearly halved as SF increases. However, the higher SF makes the frame more robust to interference and introduces the lower SNR_{req} [34], the minimum SNR needed to demodulate a signal (Table 2.1).

Here, note that LoRa ensures orthogonality between different SFs, enabling the

Table 2.1: Required SNR to demodulate a signal on each SF.

Spreading Factor	7	8	9	10	11	12
$\boldsymbol{SNR_{req}}$ $\boldsymbol{(dB)}$	-7.5	-10	-12.5	-15	-17.5	-20

simultaneous receptions of the frames. TP is the transmission power ranging from −4 dBm to 20 dBm, but available power options for data rate adaptation are limited to a few by the regional regulations [35]. Here, the use of TP over 17 dBm is restricted by a 1% duty cycle. Using a higher TP extends the transmission range while consuming more energy. CR refers to the proportion of useful payload to total data bits, including the additional parity bits for forward error correction (FEC). LoRa adopts Hamming FEC method and employs CRs of 4/5, 4/6, 4/7, and 4/8. We thoroughly discuss the trade-off of CR in Section 2.3.

2.2.3 Adaptive Data Rate

The LoRaWAN employs the link-based Adaptive Data Rate (ADR) mechanism, which dynamically adjusts the transmission parameters of an ED to link states. The main purpose of ADR is to optimize data rates, but it also affects the energy consumption and capacity of the network. It consists of two different concurrently running algorithms, each on server-side [36] and ED-side [37].

The ED-side mechanism, as complicated operations may drain a limited amount of energy, plays a one-sided role of simply lowering data rates to maintain or restore the connectivity to a GW. An ED maintains an uplink counter for ADR, and if it does not receive a downlink from a GW within a certain number of uplinks defined by *AdrAckLimit* and *AdrAckDelay*, TP or SF is increased by one step. Either case where

a downlink arrives from the GW or the counter meets the limit, the above procedure repeats after initializing the counter back to zero. Here, note that SF is increased after TP is first raised to its maximum possible option.

The server-side ADR starts by measuring the link budget from uplink messages from each ED. SNR_{margin} defines the link budget and is obtained by subtracting SNR_{req} and *deviceMargin* of 10 dB from the measured SNR. A positive link budget implies that the data rate should be lowered, and a negative value implies the opposite. A server also uses the margin to minimize SF first and then adjusts TP. When the configuration by the server is passed to an ED as a downlink control message called *LinkADRReq*, the ED can decide whether to accept it.

2.2.4 Capture Effect

The capture effect is a phenomenon between two colliding signals at the receiver, where only the stronger survives. When two signals are nearly equal in strength, both are lost. So it is a better strategy to keep one of two colliding signals alive by giving one enough power to suppress the other. Bor *et al*. [30] have demonstrated the capture effect in LoRa experimentally and noted its potential. Some studies [1-5, 38] to disentangle collisions in Section 2.2.5 can also take advantage of this for better performance.

2.2.5 Related Works

In the early stage of LoRa studies, on-site measurements were prevalent to verify the coverage of the protocol. Although the transmission range has been found to be largely dependent on field characteristics, such as topography and temperature [39,

40], typical LoRaWAN deployments are confirmed to have wide transmission ranges of around 3 km and 7 km in urban and suburban scenarios, respectively. However, a number of researches have since been introduced to question the applicability of the protocol to real-world use cases. The authors in [30, 31] condemns the collision, whose rate increases exponentially as the number of EDs grows, as a dominant cause of scalability constraints. It is shown in [30] that only 64 EDs can achieve a transmission success rate of 0.9 or higher in typical smart city deployments. Recently in [32], the authors have defined a metric, bit flux, to quantify the different LPWANs in terms of throughput over a coverage area, and evaluated LoRaWAN as suitable only for the low-rate sensing applications.

In this regard, many LoRaWAN studies nowadays aim to improve network connectivity with respect to coverage and density while ensuring a certain quality of service. Many researchers have tried to overcome the limited capabilities of LoRa hardware. The authors in [41] have implemented Choir on GW to leverage radio imperfections in frequency, time, and phase to decode a large number of transmissions simultaneously. They have also managed EDs to deliver messages in a collaborative manner to enable communication beyond their reach. Charm [42] uses coherent combining, which pools signals from multiple GWs and decodes them in the cloud, to increase the resiliency of weak signals. From an energy perspective, backscattering techniques combined with LoRa significantly reduce power consumption [43-45]. However, the above methods require modifications to hardware, thereby increasing the deployment cost.

Next, there are studies to improve performance by efficiently utilizing the channel and radio resources. The optimal number of message replication to exploit the time diversity and enhance the reception rate is explored in [38], and different channel access mechanisms, such as slotted ALOHA [46] and CCA-based CSMA [47], have

been put on evaluations. However, these approaches [38, 46, 47] essentially increase uplink or downlink traffic and burden EDs with additional operations (e.g., time synchronization). Instead, the researches on transmission parameter allocation, to which our study also belongs, have become popular due to its low cost and high effectiveness. In particular, we note that the parameter allocation is most fundamental in LoRa transmissions, and therefore, can be used in conjunction with many other aforementioned approaches to achieve synergistic performance gain.

The study of parameter allocation mainly focuses on the distribution of SF or, in addition, the ADR improvements. Many have taken approaches to properly disperse the high demands for the lowest SF to other SFs [1, 2]. EXPLoRa [1] defines a fixed SF distribution ratio and assigned SFs to EDs in order of distance to the GW so that all available SFs have the same air occupancy time; in the meanwhile, only the maximum TP is employed. And others have suggested controlling TP together with SF [3-5]. FADR [5] applies genetic algorithms to find the optimal SF distribution ratio that equalizes the collision probability among all available SFs. It then adjusts TP to fit in a given SNR_{req} with a fixed SNR margin. But unfortunately, none of the prior parameter allocation schemes or ADR proposals have noted the impact of CR, and the capture effect has also been neglected or misused.

Meanwhile, our ADR scheme, EARN, leverages those overlooked factors and fine-tune the link performance of EDs. In Section 2.5, we compare and evaluate EARN with the legacy ADR and FADR.

2.3 Link Performance Modeling

In this section, we model the link performance of a single gateway LoRaWAN. The network consists of Class A EDs that transmit unconfirmed messages, assuming

the most common application scenario. Unless otherwise stated, we assume the following parameters settings for the validation and analysis of the formulated models; 125 kHz BW, SF 7, CR 4/5, and (or) PHY payload size of 12 bytes, which is minimal due to the constraint on MAC message formats. The derived models are later used for link performance predictions to assign optimal parameters.

2.3.1 Collision Probability

As the first building block of the models, we derive the probability of having no collisions (P_{nc}). LoRaWAN adopts pure ALOHA as the medium access control mechanism and the probability is given as e^{-2G}, where G is the offered load to the network. G is obtained by multiplying *time-on-air* (T_{frame}), the amount of time it takes to send a frame, by the *arrival rate* (λ). Then we get

$$P_{nc} \approx e^{-2\lambda T_{frame}},$$

which simply implies that the greater the number of frames arriving on the network is or the larger the payload is, the higher the likelihood of collisions becomes.

T_{frame} consists of the time taken for the preamble ($T_{preamble}$) and payload ($T_{payload}$) transmissions. Given $2^{SF}/BW$ as the symbol duration (T_{symbol}), T_{frame} equals the total number of symbols that make up the preamble and payload multiplied by T_{symbol}. The number of symbols in the preamble is $n_{preamble} + 4.25$, where $n_{preamble}$ is 8 in most regions. The number of symbols needed to transmit the PHY payload with an explicit header, $n_{payload}$, is as follows [34];

$$n_{payload} = 8 + \max\left(\left\lceil \frac{8PL - 4SF + 28 + 16C}{4(SF - 2DE)} \right\rceil (cr + 4), 0\right).$$

Here, PL is the number of payload bytes, $C = 1$ indicates the presence of the CRC payload, and DE is a data rate optimization option which is set to 1 when SF 11 and

12 are employed. The lowercase cr indicates a CR index ranging from 1 to 4, each of which corresponds to a rate from 4/5 to 4/8.

Obviously, CR changes the airtime of a modulated LoRa signal, thereby affecting P_{nc}. LoRa employs Hamming FEC as the coding scheme, and CRs of 4/5 and 4/6 each exploits 1 and 2 parity bit(s) for error detection. These rates are vulnerable to interference as they can only detect errors. A single-bit error, on the other hand, can be corrected by putting 3 or 4 additional bits at a CR of 4/7 or 4/8. LoRa partitions PHY payload into bit chunks corresponding to 4 symbols and appends parity bits to form cr additional symbols. Fig. 2.1 illustrates CR impact on the collision probability $(1 - P_{nc})$ over different SFs when $\lambda = 0.1$. The probability at SF 12 increases by nearly 22% as CR changes from 4/5 to 4/8 to contain additional parity bits. Employing a higher cr seems to be a bad choice as it always increases the collisions. However, higher cr has a significant benefit in that it prevents potential error under harsh link conditions.

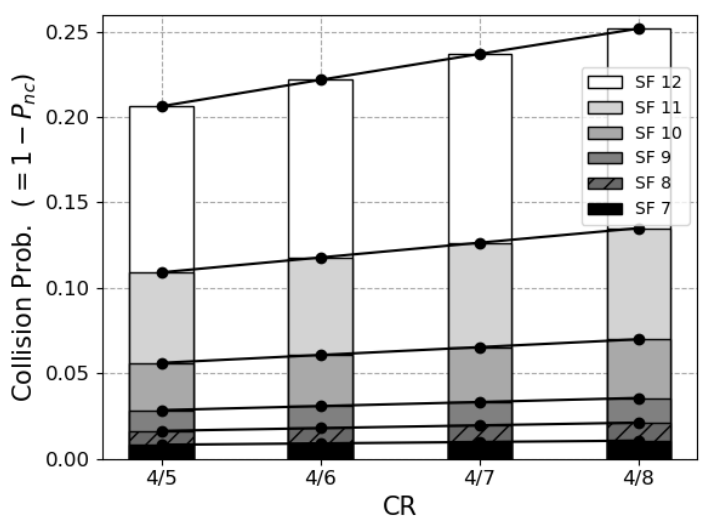


Figure 2.1: CR impact on collision probability.

2.3.2 BER and FER

The analytical expression for the bit error rate (BER) of CSS modulation can be derived from E_b/N_0, the energy per bit to noise power spectral density ratio, and the Q-function [48] as

$$BER = Q\left(\frac{\log_{12}(SF)}{\sqrt{2}} \cdot \frac{E_b}{N_0}\right).$$

Then, we express E_b/N_0 as a function of SNR,

$$\frac{E_b}{N_0} = SNR - 10\log\left(\frac{R_s \cdot SF \cdot CR}{BW}\right),$$

to examine the BER change in terms of more familiar parameters.

In the previous part, we have found the total number of symbols in a modulated signal. Since each symbol encodes SF bits, a frame contains $SF \cdot (n_{preamble} + 4.25 + n_{payload})$ bits. Then the probability of not having an error, P_{ne}, for a k-bit frame can be expressed as follows, reflecting the error detecting or correcting capability of each CR;

$$P_{ne} = \begin{cases} (1-BER)^k & if\ \ cr = 1,2 \\ (1-BER)^k + k \cdot BER \cdot (1-BER)^{k-1}, & if\ \ cr = 3,4 \end{cases}.$$

Fig. 2.2 shows the frame error rates ($FER = 1 - P_{ne}$) of different CRs as functions of SNR. As stated, a frame with 12 bytes-long PHY payload using SF 7 is taken into account. Note that each CR has a different tolerance for SNR but shows a common pattern in which FER sharply increases as SNR drops to and beyond SNR_{req}. The impact of CR on FER becomes noticeable near -7.5 dB, SNR_{req} of SF 7; FERs of CR 4/5 and 4/6 are over 0.65 while CR 4/7 and 4/8 manage to keep the rates sufficiently low thanks to single-bit error correction. For the SNR_{req} that we theoretically derive for each CR to keep FER under 0.01, CR 4/8 endures the toughest link conditions up to nearly -8 dB.

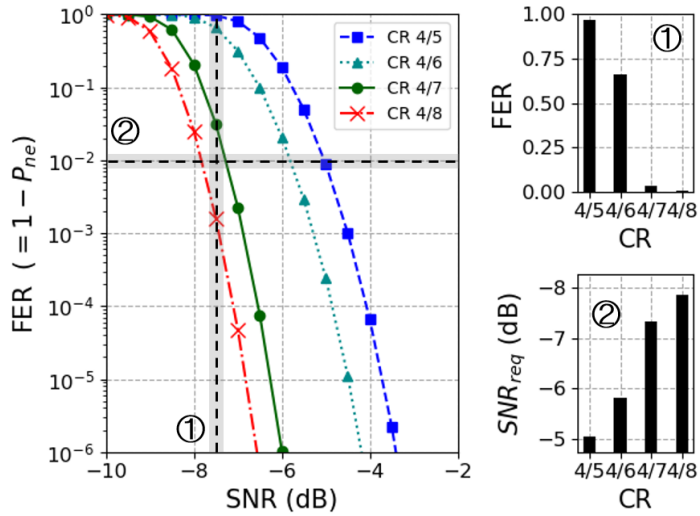


Figure 2.2: CR impact on frame error rate.

2.3.3 Link Performance

We now model the link performance of LoRa on different criteria; FDR, goodput, and energy efficiency. First, we approximate FDR by combining the previously obtained P_{nc} and P_{ne} and it inherits the characteristics of the both ($FDR \approx P_{nc} \cdot P_{ne}$). The goodput of an ED is obtained straight-forwardly by multiplying FDR, physical payload length, and λ;

$$Goodput = FDR \cdot PL \cdot \lambda.$$

Another important performance metric for battery-operated LoRaWAN EDs that are deployed over a wide area is energy efficiency. We define *energy-per-frame* (EPF), the energy consumed to successfully transfer a frame, and formulate it as follows;

$$EPF = \frac{V_{tx} \cdot I_{tx} \cdot T_{frame}}{FDR}$$

LoRa's supply voltage, V_{tx}, is 3.3 V and current, I_{tx}, whose maximum value is 125 mA, is decided by TP [34]. Which metric to use depends on the application purpose,

but EPF is what we use in our algorithm to capture the trade-off between success rate and the energy consumed.

To observe the CR impact on the link performance, we plot the theoretical EPF of an ED in relation to SNR under the load of $\lambda = 0.1$ in Fig. 2.3. In case of high SNR, it is better to employ CR 4/5 without having to endure overheads incurred by additional parity bits, since bit errors are less likely to occur. If SNR falls to or beyond SNR_{req}, the loss at P_{ne} dominates the gain in P_{nc} by low cr. Therefore, EDs should sequentially choose a higher cr as SNR deteriorates. Down until SNR -5.8 dB, CR 4/5 achieves the best EPF. Then it is followed by 4/6 until -6.4 dB, 4/7 until -7.8 dB, and 4/8 afterward. In the same context, one can also find the trade-off of CRs in the λ domain; a high CR, such as 4/5 or 4/6, is preferred as a medium gets saturated. Therefore, it is often hard to determine which CR is the best, especially near the SNR_{req}.

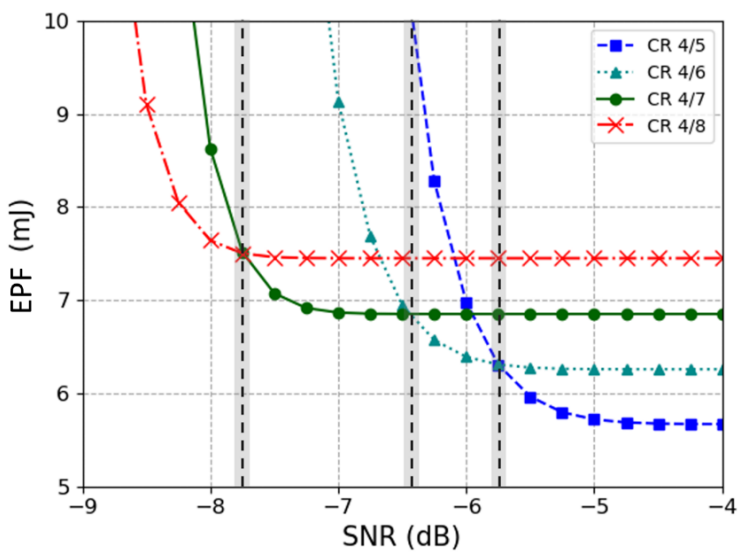


Figure 2.3: EPF vs. SNR.

2.3.4 Pilot Experiment on the Impact of CR

To suggest the necessity and effectiveness of CR adaptation on the ADR mechanism, while at the same time demonstrating that our theoretical models are also valid in real-world environments, we design and conduct a pilot experiment as in Fig. 2.4. We construct a small LoRaWAN cell with a GW connected to a server and 20 EDs. Here, one target ED is chosen to measure the performances of each CR under different link conditions, and the other EDs operate as interferers to transmit noise signals. Fig. 2.5 shows the commercially available LoRa devices we use.

We build a GW using a RAK831 board attached to a Raspberry Pi 3, which also takes the role as a server. We implement EDs with SX1272MB2DAS communication modules attached to NUCLEO-L073RZ boards. The GW is located on the third floor of an office building, and the EDs are placed at different positions on the 6th and 7th-floor corridors of the same building about 50 m apart from the GW with many walls in-between. Only the position of the target ED is relocated depending on the SNR level we try to emulate, and interferers are fixed in their place to generate noise of constant strength. And the payload length and message rates of the interferers are controlled depending on the λ level we try to emulate, while the target ED maintains its settings; 180 messages of 12 bytes payload on each 15 minutes long link condition. As impacts from different SF are inherently minimal due to their orthogonality, we assume single SF scenarios where all EDs are configured to use SF 7. Then, we fix TP to the default value of 14 dBm throughout the whole experiment and change CR on each scenario. In each round, the server records the FDR and EPF of the target node based on the number of successfully received frames. And we conduct multiple runs of the test at late nights where the impacts of human obstacles are minimized.

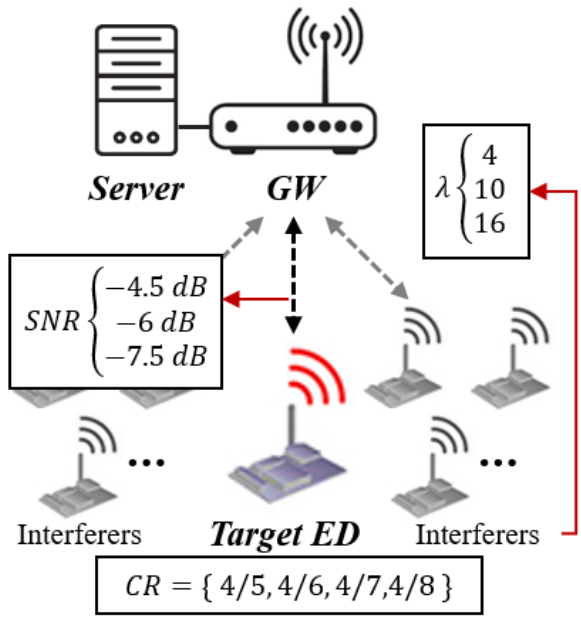


Figure 2.4: Pilot experiment setup.

Figure 2.5: LoRa devices to demonstrate the necessity of CR adaptation.

The top of Fig. 2.6 shows the 9 different link conditions experienced by the target ED; -7.5, -6.0, and -4.5 dB for low, medium, and high SNR and arrival rates (λ) of 4, 10, and 16 under low, medium, and high traffic load, on average. The center of Fig. 2.6 shows the averaged FDR of the target ED for each CRs under different link conditions. For a clear and concise interpretation of the experimental results, we compare the three-link conditions in the front, which have low traffic loads, and the three from the back, which has heavy traffic loads. A common characteristic observed in all cases is that, naturally, FDR degrades with SNR. But an important thing to note is that the degree of performance degradation depends on CR the target ED uses. When the channel is lightly loaded, and therefore, the impact of collisions is small, SNR is the dominant factor that affects FDR. Under stable link conditions with SNR of -4.5 and 6 dB, all CRs guarantee a relatively high FDR. However, as SNR drops to -7.5 dB, FDRs of high CRs (4/5 and 4/6) is significantly reduced. In

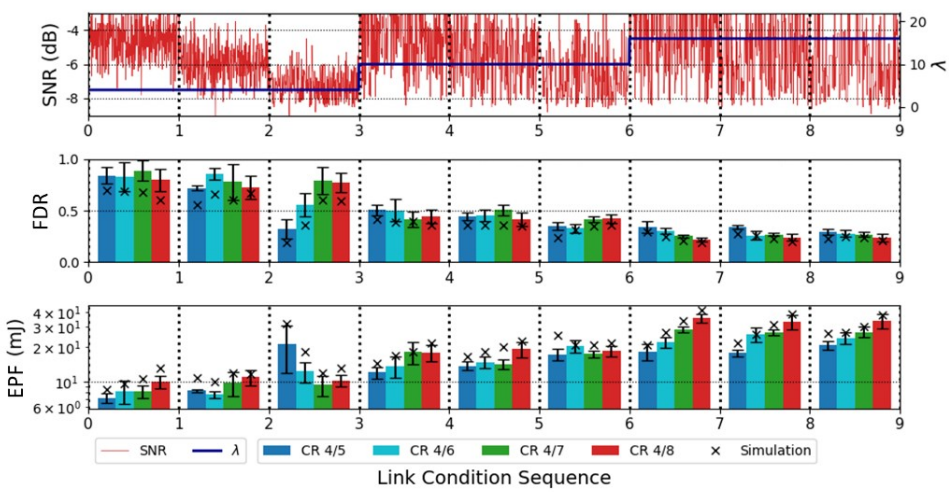


Figure 2.6: LoRaWAN link performances of different CRs under a varying link condition (SNR and λ).

the meanwhile, low CRs (4/7 and 4/8) manage to keep their FDRs, confirming that the error correction of the coding scheme actually benefits the frame delivery on the bad link. The situation is slightly different when the traffic load is high. Low CRs no longer benefit from error correction under the bad link condition as before but rather worsen the FDR. This is because the low CR introduces a higher chance of collisions with the enlarged frame length under heavy traffic scenarios, where the collision rate act as a dominant factor affecting FDR.

Another interesting observation can be derived from the bottom of Fig. 2.6 depicting EPF measurements of each CR. Even the CR with the highest FDR may not be the best option in terms of energy efficiency. As shown in the first link condition, CR 4/5 has the lowest EPF, while CR 4/7 has the highest FDR. And this implies that the CR 4/5, when energy is taken into account together, is the most efficient option. In Fig. 2.6, we also put the simulation results, which are driven based on the performance models we formulated above, to justify the evaluations that follow in Section 2.5. Although there are small performance gaps between the two, the trade-off behavior of CR depending on the link condition is similar; the best CR option in each link condition generally coincides with each other.

The above results confirm the impact of CR on target metric optimization and inspire us to design an ADR algorithm that adaptively manages CR. In general, a high CR allows EDs to utilize energy and network capacity efficiently. Conversely, given the assumption that the link is not congested, a low CR improves the network coverage by helping EDs that are located in the distance or have poor link conditions to reach a GW. The trade-off is similar to that of SF, but the scale is smaller, allowing precise calibration of the link performance.

2.4 EARN: Proposed Algorithm

In this section, we explain our enhanced ADR algorithm, EARN (Fig. 2.7), to optimize the tradeoff between success rate and energy consumption. EARN takes into account an additional transmission parameter, CR, upon data rate assignments. It also leverages the capture effect to distribute the TP of EDs. The LoRaWAN server infers configurations and path loss of all EDs from uplink messages without requiring them to explicitly hand over extra data. Upon every uplink arrival from EDs, EARN predicts link performances by substituting all possible combinations of transmission parameters into the models we designed in section 2.2 and chooses the best among them. Our objective function aims to minimize EPF, the energy efficiency metric, and is as follows;

$$\underset{x}{\text{minimize}}\, EPF(x),$$

where x is the set of target parameters, $\{SF, TP, CR\}$. The server sends a downlink control message in a receive slot of an ED in response to the uplink, only when the computed setting is better than that of the ED. In EARN, the server takes over all the complex computations, and therefore, does not risk the lifespan of EDs. Rather, the parameters, which are chosen to minimize EPF, improve the energy efficiency of EDs to help LoRaWAN keep its promise of years-long battery life.

The input to EARN, as in Algorithm 1, includes the aggregated load status of a LoRaWAN, ED_i's uplink frame, and ED_i's $\{SF_i, SNR_i, G_i\}$ measurements at the previous EARN operation. These are configured SF, estimated SNR and load which are kept by the server for ED_i, and the default value of G_i is zero. A LoRaWAN server that operates EARN maintains an aggregated load status for each SF and SNR to estimate collision probability within the impact range of collision and the capture effect. To be specific, the server can tell the network demand of all EDs whose SF is

Algorithm 1 EARN Algorithm

Input: Uplink info and the last $\{SF_i,\ SNR_i,\ G_i\}$ of ED_i and $AggregatedLoad$

Output: $\{SF,\ TP,\ CR\}$ configuration for ED_i

```
    # Initialize and load the previous config info
1:  EPF_best ← ∅
2:  Conf ← {∅, ∅, ∅}
3:  {SF_i, SNR_i, G_i} ← Last {SF_i, SNR_i, G_i} of ED_i
    # Retrieve the load portion of ED_i
4:  AggregatedLoad[SF_i][SNR_i] −= G_i
    # Update the arrival rate and path loss info of ED_i
5:  λ_i ← avg(FCntDiff/TimeGap of 20 recent frames)
6:  PathLoss_i ← avg(PathLoss of 20 recent frames)
7:  for sf = 7,8,…,12 do
8:      for tp = 2,5,…,14 do
9:          for cr = 1,2,…,4 do
10:             T_frame ← getTframe(PL_i, BW_i, sf, cr)
11:             G_temp ← T_frame · λ_i
12:             SNR_temp ← getSNR(PathLoss_i, tp)
13:             G_col ← G_temp
                # Sum up the load that may collide with ED_i
14:             for j = (SNR_temp − 6) … SNR_max do
15:                 G_col += AggregatedLoad[sf][j]
16:             end for
17:             P_nc ← PncModel(G_col)
18:             P_ne ← PneModel(PL_i, BW_i, sf, cr, SNR_temp)
19:             FDR ← P_nc · P_ne
20:             Energy ← getEnergy(T_frame, tp)
21:             EPF_temp ← Energy/FDR
                # Keep the best config for ED_i
22:             if EPF_best == ∅ or EPF_best > EPF_temp then
23:                 EPF_best ← EPF_temp
24:                 Conf ← {sf, tp, cr}
25:                 {SF_i, SNR_i, G_i} ← {SF, SNR_temp, G_temp}
26:             end if
27:         end for
28:     end for
29: end for
    # Update the aggregated load info
30: AggregatedLoad[SF_i][SNR_i] += G_i
31: return Conf
```

Figure 2.7: EARN Algorithm.

7 and SNR at the GW is, on average, around -4 dB. After initialization, it subtracts from aggregated load the portion of ED_i, if there is any ED_i's contribution from the past operation (line 4). Then, EARN can obtain parameter settings, arrival time, frame count, and SNR from the uplink frame, which are then translated into the input for EARN to run a set of thorough substitutions to the performance models. The message rate (λ) is computed by dividing the frame count difference of two successive frames from an ED by arrival time gap (line 5). The path loss of an ED is inferred from SNR measured at the GW (line 6). The λ and path loss estimation may not be accurate and fluctuate from time to time. So, EARN takes the averaged values of up to 20 latest measurements to mitigate the error. After all the ingredients are ready, EARN predicts link performance by try feeding all SF, TP, and CR combinations to the model sequentially. For each iteration, we first compute T_{frame} whose length varies by SF and CR, and multiply it with λ_i to get G_{temp}, the load of ED_i (line 10-11). Then, we also estimate SNR for the given TP (line 12). These are used to consolidate the loads (G_{col}) within the SNR range by which ED_i is affected (line 13-16). Then the rest procedures to estimate EPF are straight forward. P_{nc}, the probability that no collision occurs, can be estimated by substituting G_{col} to the P_{nc} model. P_{ne}, the probability that no error occurs, can be estimated by substituting PL, BW, SF, CR, and SNR_{temp} to the P_{ne} model. T_{frame} and TP are used to compute the energy consumed on transmitting a frame. EARN assigns to ED_i the SF, TP, and CR values with the lowest expected EPF (line 22-25). Finally, the corresponding SF, SNR, and load estimates for ED_i are recorded to the latest network load status of the server (line 30).

The parameters assigned by EARN to satisfy SNR_{req} and achieve minimal EPF work well in ideal link conditions. However, the tight parameter match to a predicted

link condition is not always the best choice, as noise and multipath effects adversely influence the EARN's link estimation in the real-world scenarios; there is a higher chance of losing frames that have lower SNRs than the expected ones. So we make up for the weaknesses that allow EARN to adapt to dynamic channel conditions. Unlike the legacy ADR that assigns parameters by placing a huge fixed-size margin of 10 dB on a given link budget, we let EARN employ an *adaptive margin* of variable size. In EARN, the size of the margin to adjust the link budget measurement is decided by averaging the SNR standard deviations from up to 20 latest frames. The above revision can be easily reflected to the original EARN by subtracting the marginal value from the estimated SNR which is passed as an input to P_{ne} model in line 20. We named the algorithm EARN-AM (Adaptive Margin), and it brings a protective margin against channel noise, just like the legacy ADR, but alters the size in response to network conditions to make better use of communication resources.

2.5 Evaluations

In this section, we compare the performance of EARN with legacy ADR and FADR [5]. FADR is a well-established recent ADR proposal whose mechanism is described in our preliminaries. In all schemes, the ED-side ADR mechanism operates as before, but FADR and EARN replace the server-side ADR operation with their own. We evaluate the performances in terms of goodput, EPF, and fairness under three different scenarios, each of which varies the noise level (i.e., the standard deviation in the path loss model), cell radius, and traffic load. Each scenario averages the results of 30 episodes of 24 hours long simulation whose network topology is regenerated every time.

2.5.1 Simulation Setup

We implement a large-scale LoRaWAN simulation testbed using Simpy [49], a discrete event simulator on Python. The simulation environment imitates the real-world behavior of LoRaWAN as much as it can. All factors that can possibly bring about a collision, such as co/inter-SF interference, the capture effect, and overlap durations of all up/downlink frames as well as transmission parameters, are taken into account.

Table 2.2 shows the parameter settings we use on the experiments, where the values with asterisks (*) are set by default; 1,000 EDs transmitting 12 bytes of payload once every 20 minutes ($\lambda = 0.83$) are randomly scattered within the range of 4.5 km around a GW. We exploit the log-distance path loss model with the following parameters as in [39]; 128.95 dB as the mean path loss of 1 km distance, 2.32 as the path loss exponent, and 3.0 as the standard deviation of the path loss. Although the maximum transmission range of a LoRaWAN can be as far as 12 km with SF 12 in our configuration that reflects a suburban environment, we limit the cell radius to 4.5 km by default, which is the communication range of an ED with SF 8, for the fair comparison with conventional schemes. FADR requires EDs to be placed close enough to a GW so that the fixed SF distribution ratio can be kept. In LoRaWAN's basic operation, multiple channels and bandwidths can be employed. However, for the sake of simplicity, only 125 kHz wide single-channel communication is tested, as the use of EARN and other ADR schemes can be easily extended to multi-channel and multi-band scenarios.

2.5.2 Parameter Distribution

We first observe the different behaviors of ADR schemes in terms of SF, TP, and

Table 2.2: Simulation parameters.

Parameters	***Values***
Carrier frequency (CF)	922.0 MHz
Bandwidth (BW)	125 kHz
Spreading factor (SF)	7, 8, 9, 10, 11, 12*
Transmission power (TP)	2, 5, 8, 11, 14* dBm
Coding rate (CR)	4/5*, 4/6, 4/7, 4/8
Number of nodes (N)	1,000
PHY payload (PL)	12 bytes
Traffic load (λ)	0.28, 0.56, 0.83*, …, 2.22
Cell radius (R)	3.5, 4.5*, 5.7, 7.3, 9.4, 12 km
Path loss Std Dev. (σ)	0, 1, 2, 3*, 4

CR distribution. Fig. 2.8 shows an example of a simulation environment with 1,000 EDs randomly placed within a range of 4.5 km around a GW. The circled borderlines present the maximum communication ranges defined by SNR_{req} for each SF. For different ADR schemes, Fig. 2.9, 2.10, and 2.11 show the distributions of SF, TP, and CR, where each point corresponds to an ED, and Fig. 2.12, 2.13, and 2.14 show the distribution ratios of SF, TP, and CR.

Generally, the farther away an ED is from the GW, the higher SF and TP are adopted. ADR estimates the link budget with a generous margin of 10 dB to SNR_{req}, and therefore, allocates especially higher SF and TP than those of which an actual communication requires. FADR allocates SF to EDs based on fixed ratios with the purpose of equally balancing the frame airtime or collision probability among

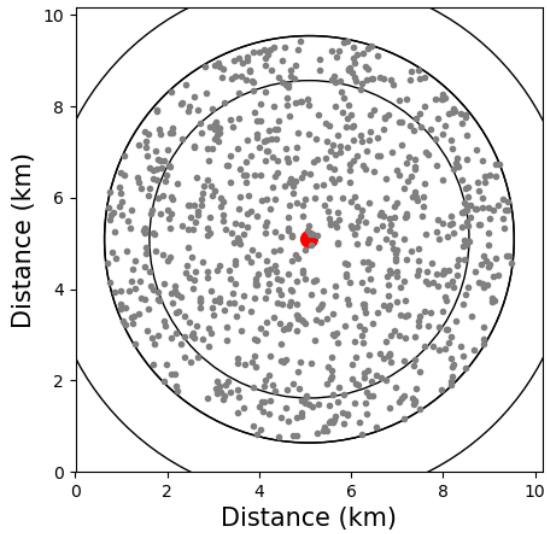


Figure 2.8: GW and ED deployment.

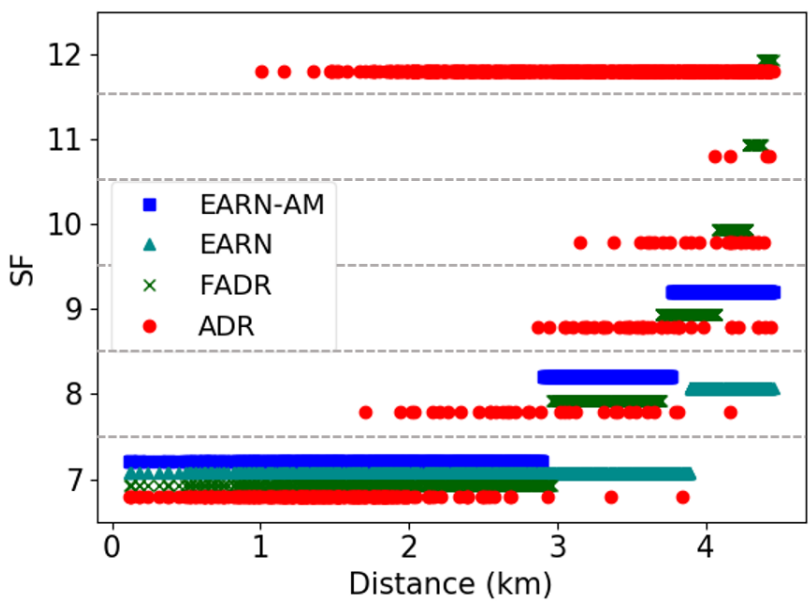


Figure 2.9: SF distribution.

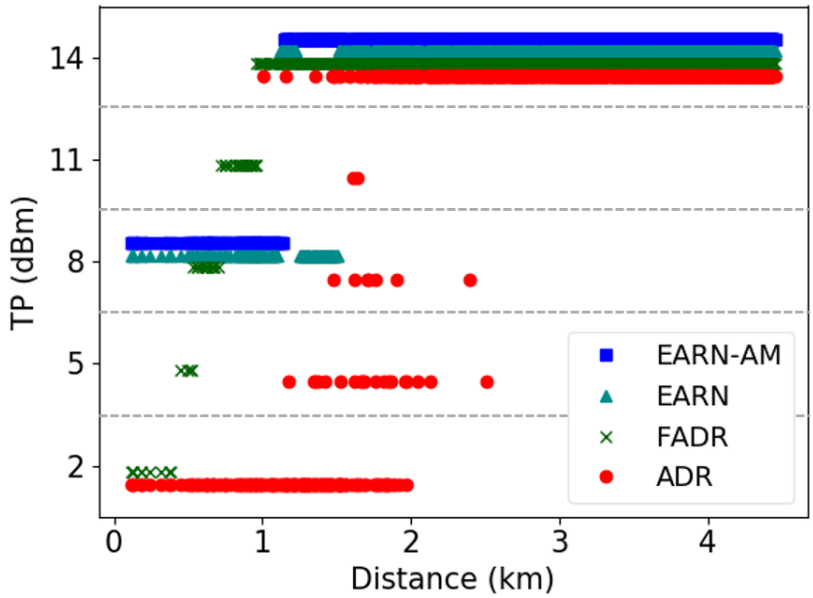


Figure 2.10: TP distribution.

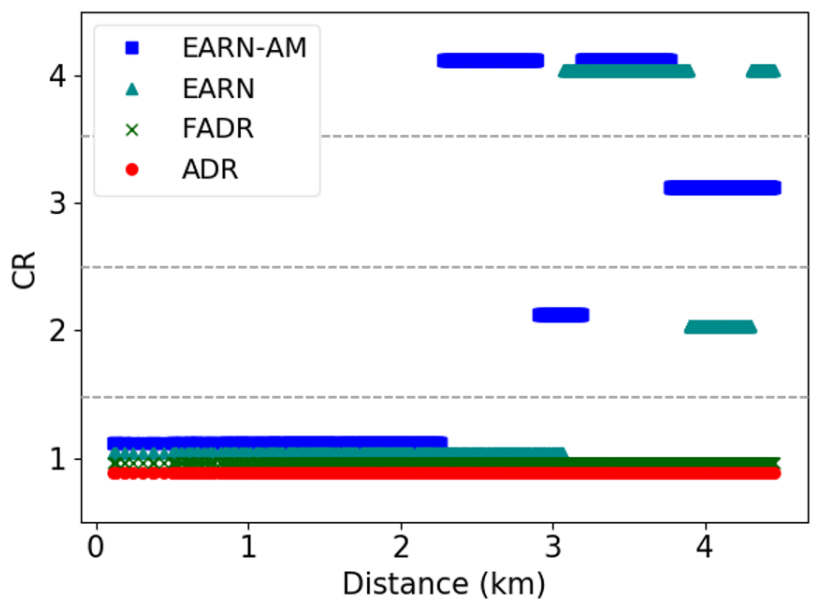


Figure 2.11: CR distribution.

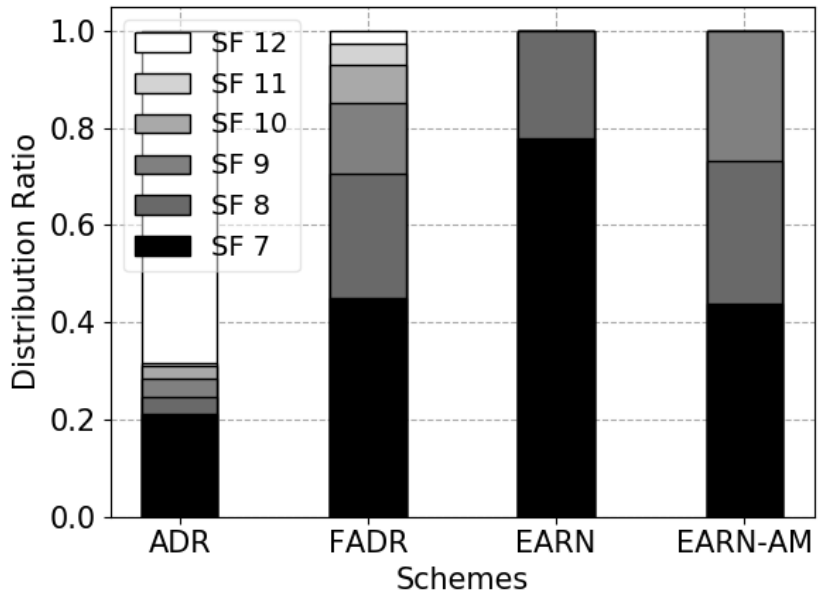


Figure 2.12: SF distribution ratio.

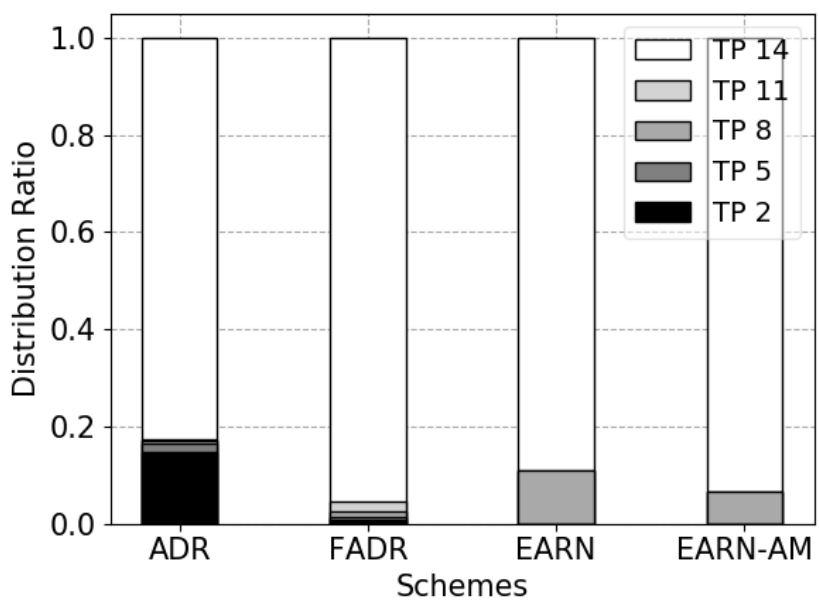


Figure 2.13: TP distribution ratio.

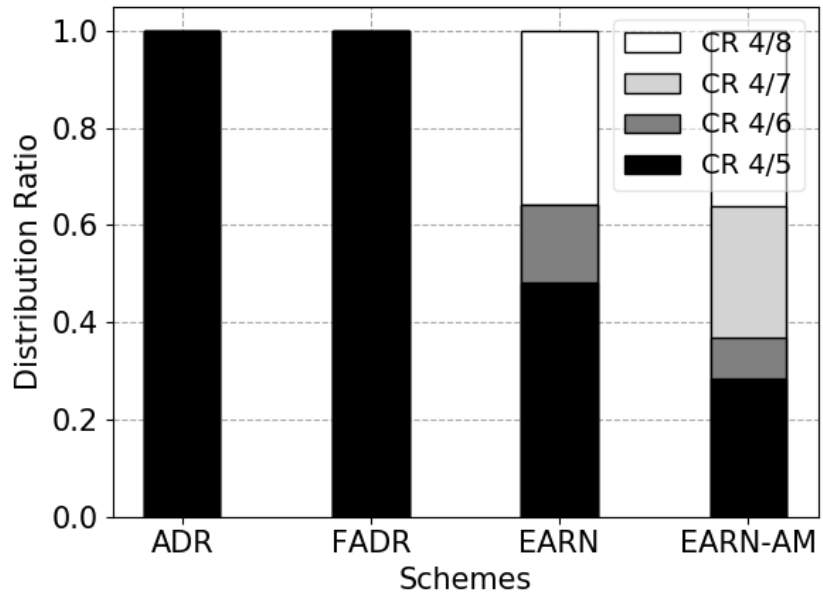


Figure 2.14: CR distribution ratio.

different SFs; the ratio of each SF from 7 to 12 is {0.449, 0.257, 0.144, 0.08, 0.044, 0.024}. EDs out of those ratios are eventually allotted SF 12, aggravating the load in the lowest data rate. FADR can work intact on LoRaWAN with the cell radius of up to 3.5 km, but they may not function properly on larger cells depending on the ED deployments. The SF distribution of EARN and EARN-AM, on the other hand, is not tied to any ratio and consists only of SF 7, 8, and 9. Since the cell is small and the default network load can be sufficiently accommodated with these high data rates, SF 10, 11 and 12, which consume unnecessarily much energy, are not employed at all. The rate of allocation of SF 7 in EARN, without taking path loss deviation into account, is significantly higher than in EARN-AM.

Unlike other methods that sequentially assign higher TPs based on the distance of an ED to a GW, EARN and EARN-AM assign TP of 8 dBm to EDs near the GW and TP of 14 dBm to the others. This is to evenly distribute the collision probability on

the SNR domain among EDs with the same SF. If TPs of some EDs, no matter how close they are to the GW, are equally configured to SNR_{req}, their messages will all be lost upon collision due to the capture effect. The power chosen by EARN, 8 dBm, is neither too weak to lose in competition, nor too strong to waste energy as in Fig. 2.15.

Moreover, in Fig. 2.9, it is important to note that the wider communication ranges to each SF by EARN and EARN-AM is the benefits from the suitable CR values. Our methods, unlike the other ADR methods, are capable of reducing the error rates of EDs with lower CRs, especially near the SNR_{req}. The lower CR constructs a slightly larger frame, but it obtains an improved tolerance to the path loss and noise. If we observe the EARN's CR allocation in Fig. 2.11, EDs close to a GW put priority on efficiency and exploit the highest CR of 4/5, and EDs far from the GW start to adopt a lower CR to prioritize stability. The CRs allocated by distance are sequentially 4/5,

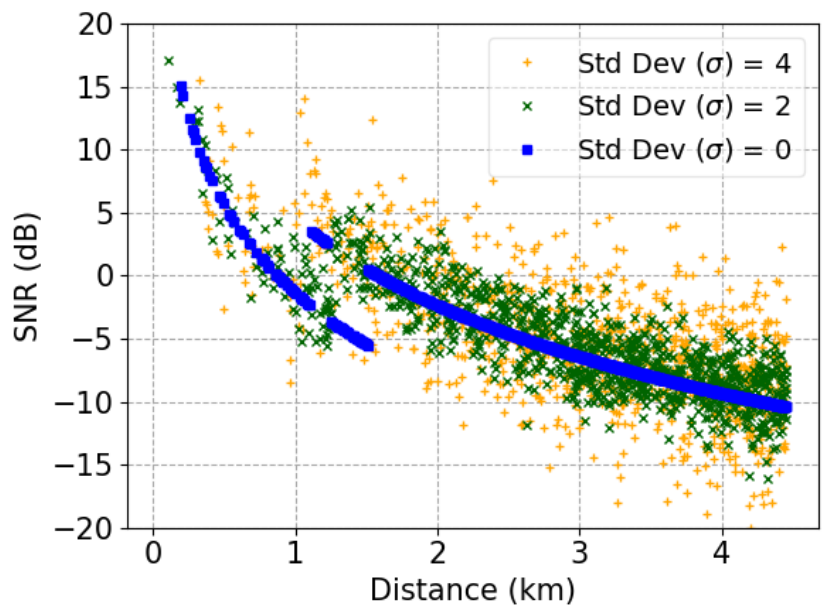


Figure 2.15: SNR at the GW with and without fluctuation in EARN-AM.

4/8, 4/6, 4/8, 4/7, and 4/8. The same pattern is exhibited in all subsequent experiments. The CR adaptation of EARN-AM follows this pattern with a distance imposed by the adaptive margin.

2.5.3 Noise Level

The path loss and resulting SNR can be varied due to factors like shadowing and noise. We analyze the impact of link fluctuations on performance in Fig. 2.16 and 2.17, to justify EARN-AM, the extended version of EARN. In Fig. 2.16, the goodput of all ADR schemes decreases as the standard deviation of the path loss model increases from 0 to 4. This is because the transmission parameters were assigned to satisfy the link budget estimated by the SNR measurements. If the actual SNR of an uplink is lower than the predicted due to the fluctuations, the message will likely be lost. As mentioned earlier, ADR employs a generous SNR margin of 10 dB to respond to such deviation. Although it results in the worst goodput and EPF, no further performance degradation will occur unless the SNR variation is greater than 10 dB. FADR shows the highest goodput in the smallest cell, as collision probabilities are evenly distributed to different SFs. Still, there is also a fast performance degradation as the deviation in the path loss model becomes more substantial. EARN, without noise, achieves great baseline performance in terms of goodput but is susceptible to link fluctuations. This is because the EARN makes the best use of the estimated link budget by tightly adjusting the SF, TP, and CR to fine-tune the performance. The parameters become impractical when the actual link condition does not match the prediction, and the penalty in the performance is the largest among the ADR schemes. Nonetheless, it provides the best energy efficiency with EARN-AM. EARN-AM, in particular, uses statistics on channel conditions to

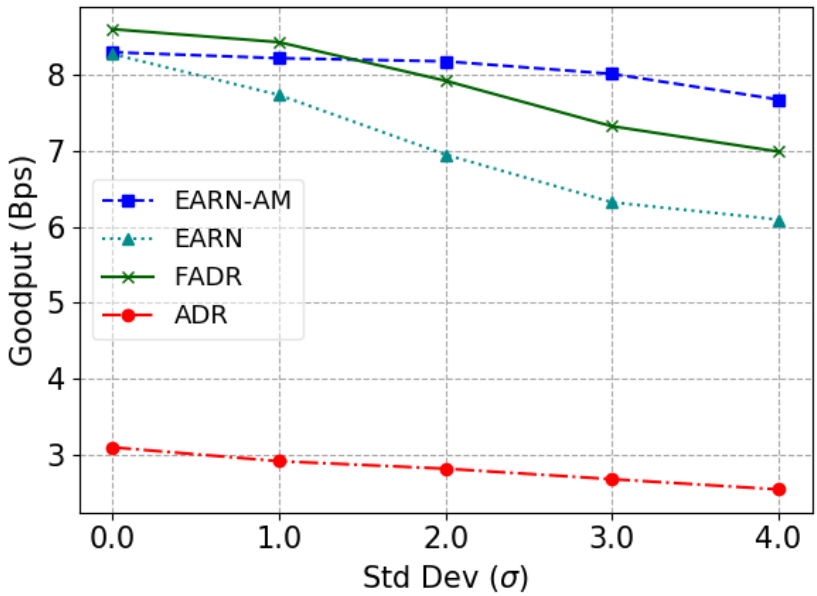


Figure 2.16: Goodput vs. Std dev. in the path loss model.

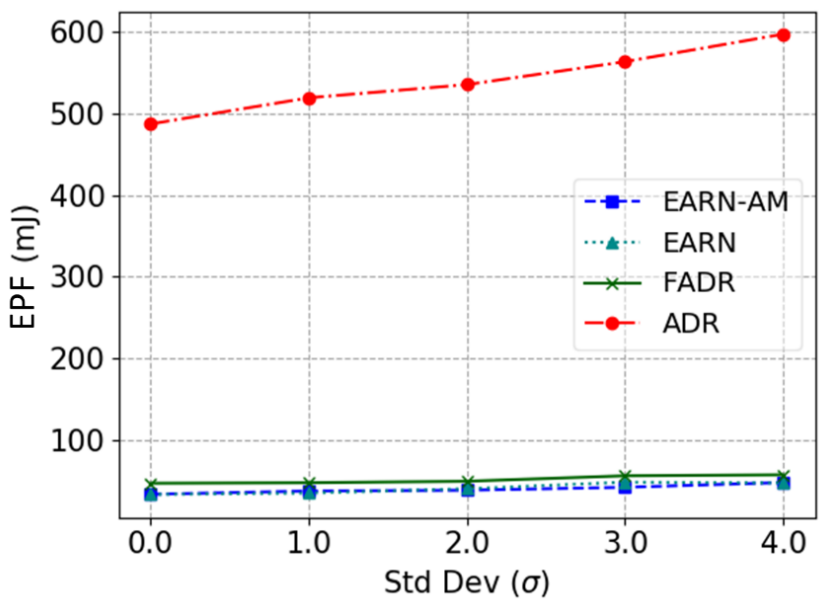


Figure 2.17: EPF vs. Std dev. in the path loss model.

respond to link fluctuations. As shown in Fig. 2.16, EARN-AM performs as good as EARN when the deviation in the path loss model is zero. The employment of the *adaptive margin* has notably relieved the performance penalty by severe link fluctuations, and EPF of EARN-AM is also the lowest of all.

2.5.4 Cell Radius

In Fig. 2.18 and 2.19, we observe the performance change due to the cell radius, the maximum distance from the GW where an ED can be placed. The tested distances are the maximum communication ranges of each SF based on SNR_{req} with default parameter settings; respectively, 3.5, 4.5, 5.7, 7.3, 9.4, and 12 km. In the figures, ADR seems to be resistant to distance changes. However, the performances are so bad that ADR stands far behind other methods in terms of both efficiency and scalability. The conservative strategy of ADR always assigns high SF and TP to most EDs except for the few adjacent to the GW, easily saturating the medium in low data rates. FADR shows high goodput and low EPF in narrow cells, but performance degrades rapidly as the cell radius broadens. The fixed SF distribution ratio of FADR, which evenly portion out the collision probability to SF, can be complied with only in small cells. The goodput of EARN-AM in the smallest cell is slightly lower than that of FADR about 2.2%, but the performance reduction over radius change is modest. The goodput of EARN constantly suffers from the path loss deviation. In the meanwhile, it suffices for its objective function and achieves the best EPF in all cases. It is not tied to a certain SF distribution ratio, but rather it immediately assigns parameters that are most suitable for an ED in a given network condition. Besides, the stability achieved by CR adaptation significantly enhances the capacity of all SFs by lowering SNR_{req}. FADR, in particular, is imperfect in its design in that it only

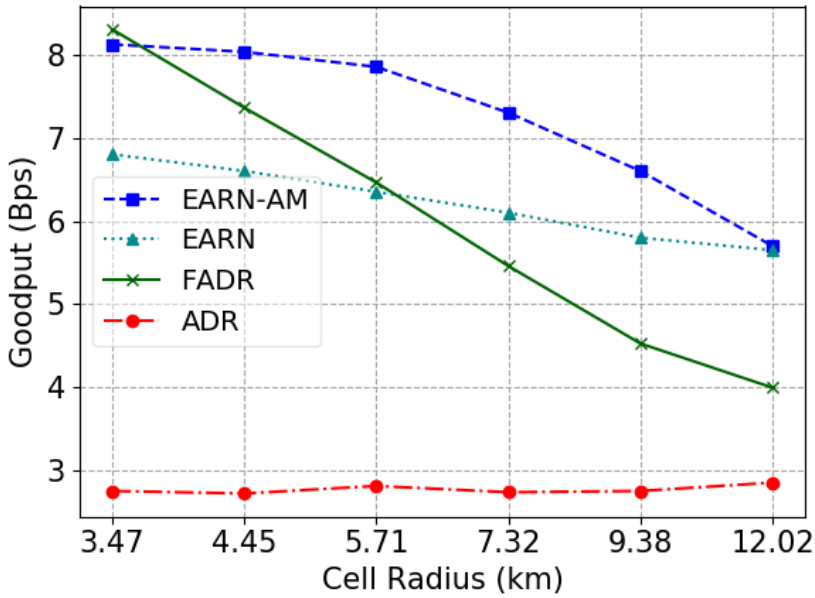


Figure 2.18: Goodput vs. Cell radius.

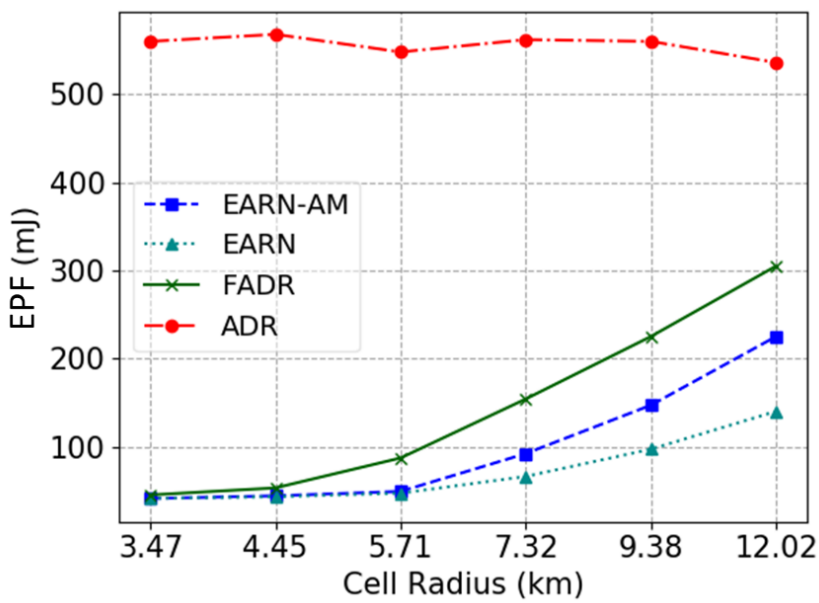


Figure 2.19: EPF vs. Cell radius.

assumes the narrow range cells where all SFs are available. They do not properly reflect low-power and long-range characteristics of LoRaWAN. EARN-AM assigns link-adaptive tight SF, TP, and CR exhibiting a reliable performance over a wide range of environments and application scenarios.

2.5.5 Traffic Load

We observe performance changes by varying traffic loads in Fig. 2.20 and 2.21, as the traffic capacity of a network is a metric that is closely related to the scalability. The evaluation assumes general application scenarios of LoRaWAN and starts from the initial load of about arrival rate (λ) 0.28, which is the load of 1,000 EDs transmitting messages once an hour. Then the traffic is gradually increased by shortening the message period to 30, 20, 15 minutes, and so on. The goodput of all ADR methods in Fig. 2.20 show a pattern similar to that of the famous ALOHA performance function; goodput increases to a certain level and then begins to decrease as the collision rate grows exponentially. The legacy ADR is the first to reach its maximum goodput near $\lambda = 0.56$ and enters the decreasing phase. This is followed by EARN, FADR, and EARN-AM near $\lambda = 1.39$. EARN-AM is ahead of other methods regardless of the load degree, showing about 7.5% better performance than FADR, the second-best ADR method, at their maximum goodput. We can refer that EARN-AM makes the best use of channel capacity to sustain the highest load with the greatest stability. However, we notice that collisions between uplink frames are not the sole factor that degrades the performance. When we track down the cause of frame losses, the processing capacity of a GW acts as a bottleneck under heavy traffic loads. A LoRaWAN GW, which can simultaneously decode up to eight frames, is incapable of sending downlink control messages, which contains the new parameter

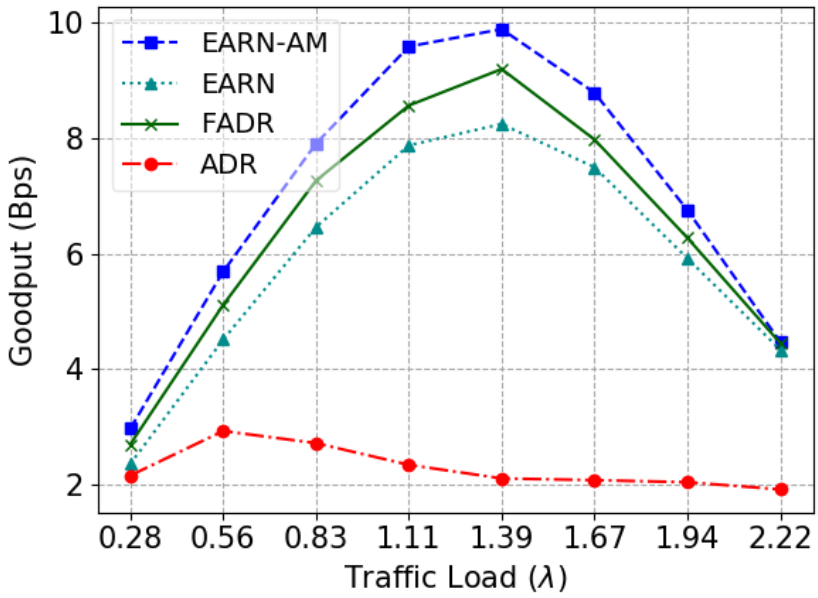


Figure 2.20: Goodput vs. Traffic load (λ).

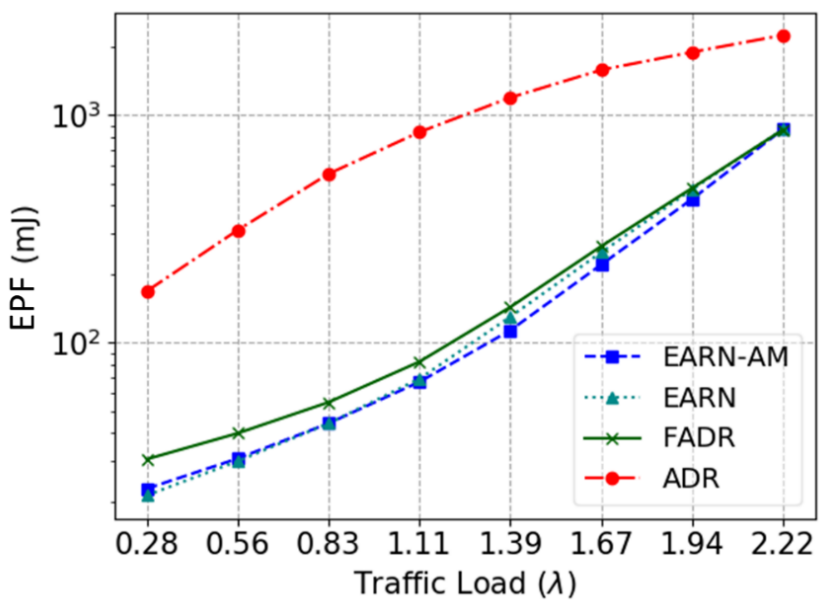


Figure 2.21: EPF vs. Traffic load (λ).

settings from the server, back to EDs when there are too many frames arriving. EDs, failing to receive any downlinks from the GW, gradually lower their data rates and intensify the performance degradation.

In the meantime, EARN and EARN-AM maintain the lowest EPF under any loads as shown in Fig. 2.21, indicating that the link prediction and subsequent parameter assignment are successful. The adoption of different CRs lowers SNR_{req} and increases the number of EDs that a SF can sustain, which in turn results in better P_{ne} and P_{nc}. If our models additionally take into account the processing power of a GW and the impact of downlink messages, or if we find an effective way to secure the downlink control messages, further improvements are expected in link performance prediction and parameter assignment.

2.5.6 Fairness

Finally, we compare the fairness of different ADR schemes. The fairness that we use in our evaluation is the fairness in results (i.e., FDR), as the transmission opportunities of EDs are already properly controlled by the existing duty cycle limit of LoRaWAN. Fig. 2.22, 2.23, and 2.24 show the FDRs of EDs when ADR, FADR, and EARN-AM are applied, respectively. In the figures, each point corresponds to an ED and demonstrates the distance from a GW. In the legacy ADR, most EDs that are not close enough to the GW are easily congested with one another and exhibit low FDR overall with wide variations. FADR, with an improved SF and TP allocation strategy, shows a better FDR distribution compared to ADR. However, it does not adapt CR and therefore is susceptible to frame error induced by path loss. The CR impact is evident where an ED is about 3km away from the GW near SNR_{req}, and in Fig. 2.23, we observe the huge gaps between FDRs of EDs placed at the

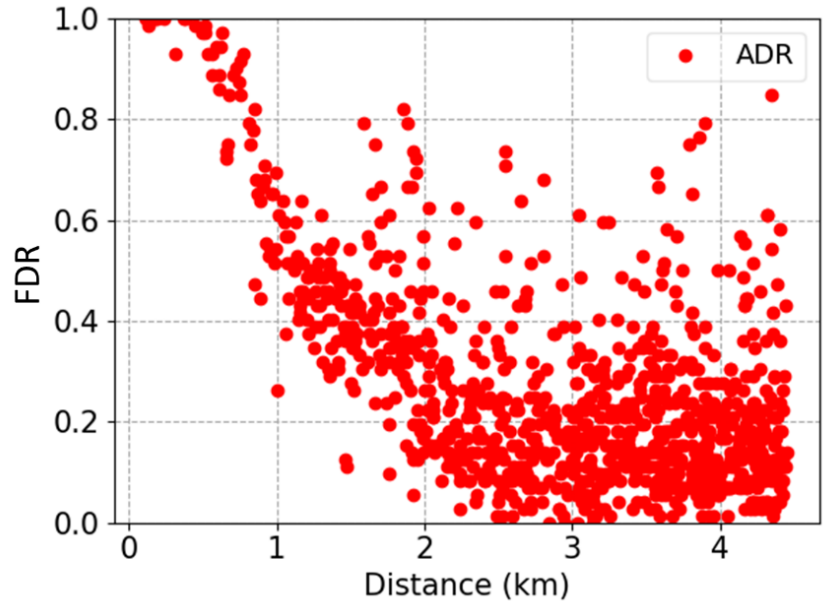


Figure 2.22: FDR vs. Radius of ADR.

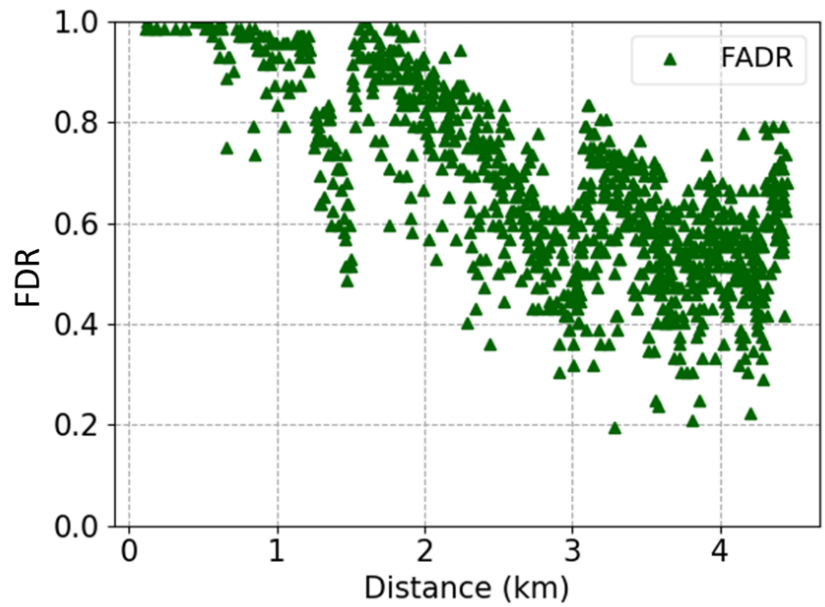


Figure 2.23: FDR vs. Radius of FADR.

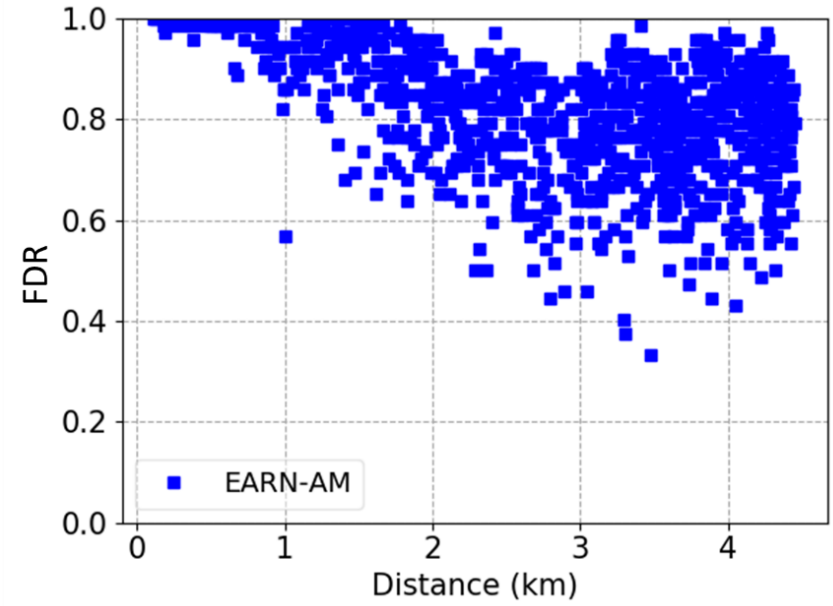


Figure 2.24: FDR vs. Radius of EARN-AM.

transmission range boundaries limited by each SF. Furthermore, the low FDRs from EDs 1.5 km away from the GW is due to the bad TP allocation practice. They are in the middle of near and far EDs in SF 7, and their frames are lost in collisions. EARN-AM, on the other hand, fine-tunes the link performance of EDs by adapting CR as well as SF and TP. As a result, the decrease in FDR with distance is moderate, and the deviation is also not significant as in Fig. 2.24.

We also evaluate the fairness of each method with Jain's fairness index based on FDR as in [5], and the formula is as follows;

$$\zeta = \frac{\left(\sum_{i=1}^{N} FDR_i\right)^2}{N \sum_{i=1}^{N} FDR_i^2}.$$

Here, N is the number of EDs located within a LoRaWAN cell, and FDR_i denotes the FDR of an ED_i. The fairness index has a value ranging from 0 and 1, where the higher value indicates the higher FDR fairness between the EDs. Fig. 2.25 shows the

fairness index of each method under different cell sizes. The result is truly meaningful in that EARN-AM, which optimizes EPF, outperforms FADR which prioritizes fairness. FADR shows good fairness between most EDs in the smallest cell. However, as the cell size increases, the fixed SF distribution ratios cannot be complied with and thus, the fairness is lost. In EARN-AM, all EDs independently obtain high FDR by estimating the link condition and searching for the best parameter set. This, in turn, ensures overall high FDR throughout the network, even to the EDs far away from the GW. The fairness of EARN-AM is similar to FADR in the smallest cell and surpasses the others in the larger cell scenarios.

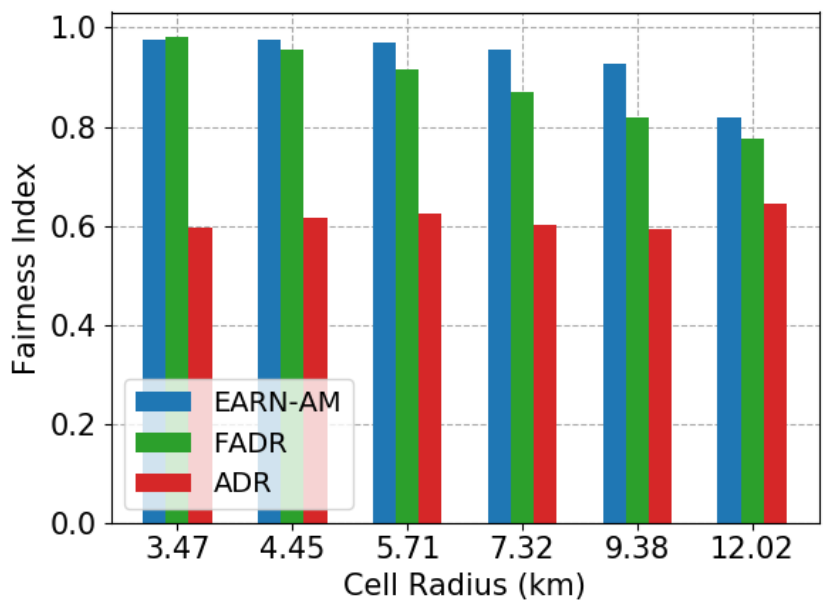


Figure 2.25: Fairness index vs. Cell radius.

2.6 Summary

In this work, we formulated LoRaWAN link performance models that reflect the complex correlation between transmission parameters, emphasizing the need for the adaptive employment of CR in ADR mechanism. Then, we proposed an enhanced ADR called EARN, which takes into account CR and the capture effect, to fine-tune the link performance of EDs. In the end-to-end LoRaWAN simulator we developed, EARN outperformed the conventional schemes in terms of goodput, EPF, and fairness, refining LoRa/LoRaWAN as a scalable and practical IoT solution.

Adapted from Jonghwan Chung, Junhyun Park, Chong-Kwon Kim, and Jaehyuk Choi "C-SCAN: Wi-Fi scan offloading via collocated low-power radios" IEEE Internet of Things Journal 5.2 (2018)

Chapter 3

C-SCAN: Wi-Fi Scan Offloading via Collocated Low-Power Radios

3.1 Introduction

With the increasing demand for high-speed and cost-effective Internet access, Wi-Fi has become the predominant wireless technology in modern mobile devices. According to Cisco's latest forecast [50], mobile data traffic has grown 18 times over the past five years and 60% of the total mobile data traffic in 2016 was offloaded onto the fixed network through Wi-Fi and/or femtocell. In fact, Wi-Fi connectivity has become a crucial requirement for wireless consumers to use data-hungry applications without worrying about their cellular data usage.

The first step in delivering the benefits of using a Wi-Fi connection on mobile devices is to discover available Wi-Fi APs in the vicinity by *Wi-Fi scanning*. Wi-Fi scanning is the task of searching for available APs and their operation channels at a given location. It is a fundamental feature to provide adequate quality of experience requirements for various mobile applications, such as virtual reality and mobile

media, where the demand for ensuring low-delay and seamless connectivity is important. To meet these requirements, however, frequent Wi-Fi scanning should be performed owing to user mobility and the short transmission range of Wi-Fi networks [9], which will lead to excessive battery drain.

An ideal scanning algorithm seeks to discover the maximum number of APs in the shortest period of time [11]. However, it is not an easy task to design such an optimal Wi-Fi scanning algorithm because the scanning station has no prior knowledge on the channel information of neighboring APs. Fig. 3.1 and 3.2 show the empirical probability distribution that at least one AP is deployed in a given channel, measured in various places such as subway stations, cafes, restaurants, and houses. From the result, we can observe high variations in the AP deployments across channels. In order to obtain the AP and channel information, traditional scanning algorithms scan the full set of channels, including channels where no APs exist, spending a certain amount of time on each of them. As a result, this approach often leads to high scanning latency and/or energy consumption due to unnecessary scanning on the empty channels.

In this work, we present a novel approach for Wi-Fi channel scanning on mobile devices, called C-SCAN, that discovers available channels of nearby Wi-Fi APs in a fast and energy-efficient way. Unlike traditional approaches using a Wi-Fi interface, C-SCAN uses a low-power collocated wireless personal area network (WPAN) radio, particularly Bluetooth or BLE, coexisting in the scanning device for Wi-Fi channel scanning. C-SCAN exploits the fact that a Bluetooth radio can sense Wi-Fi signals in a frequency band (i.e., a set of consecutive Bluetooth channels) that overlaps with the Wi-Fi channel. By using the Bluetooth radio, C-SCAN identifies which Wi-Fi channels are used or not prior to the actual channel scanning with a Wi-Fi interface. By excluding the channels determined to be empty, Wi-Fi scanning manager then

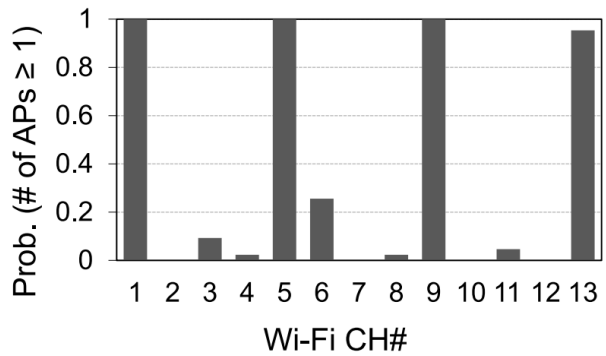


Figure 3.1: Channel assignment status of managed Wi-Fi deployments.

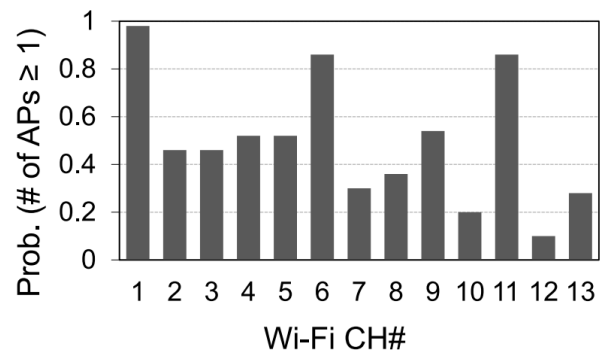


Figure 3.2: Channel assignment status of un-managed Wi-Fi deployments.

can perform *selective* scanning only on available Wi-Fi channels, thereby obtain significant performance gain in terms of delay and energy.

Although there have been many efforts [13-18] to assist Wi-Fi discovery by utilizing secondary coexisting WPAN radios, such as Bluetooth or ZigBee, none of them has the ability to provide channel information of available APs, in particular, their operating channels. As a result, they have to scan the full set of channels including the channels where no APs exist even after the availability of Wi-Fi systems is detected, thus spending a significant amount of time. Unlike the existing solutions, our proposed scheme C-SCAN is not only to predict the availability of Wi-

Fi networks but also to pinpoint the available Wi-Fi channels by using collocated Bluetooth radios.

However, there are several challenges to achieve this goal because of the following factors.

1) *Large Search Space:* To scan the entire *wideband* Wi-Fi channel using narrowband Bluetooth radio, the entire search space increases (i.e., from 13 to 79 channels), which may induce significant scanning latency.

2) *Measurement Accuracy:* A practical challenge in the design of C-SCAN is the Bluetooth radio's channel sensing accuracy. Since the Bluetooth radio cannot decode Wi-Fi frames, we should rely only on the RSSI measurement. However, RSSI values are very noisy and are easily changed by multipath effects [51].

3) *Computational Cost:* The solution should be lightweight to be easily implemented and operated in resource-constrained devices, such as IoT devices, without considerable computational cost or high energy consumption [52].

To tackle these challenges, we present a lightweight RSSI sampling method for C-SCAN to minimize the number of sampling channels. In addition, we employ a min–max-based sample normalization and similarity analysis to investigate the correlation between RSSI samples, thus, making C-SCAN robust to RSSI variations. Further, we develop a simple binary string matching and scoring algorithm to make an accurate decision on the presence of APs on target channels.

The main contributions of this work can be summarized as follows.

- Introduction of a new energy-efficient approach, called C-SCAN, that harnesses a low-power Bluetooth interface collocated in a mobile device to identify available Wi-Fi channels. Unlike traditional schemes using Wi-Fi

interfaces, our scheme uses a Bluetooth radio, which is readily available in most modern mobile devices and enables low-delay and energy-efficient Wi-Fi scanning (Section 3.4).

- Design of an intelligent channel identification algorithm that pinpoints the Wi-Fi channel numbers. Inspired by the observation obtained from a rigorous measurement study, we design the baseline to identify a target Wi-Fi channel based on the RSSI values measured over several narrowband Bluetooth channels (Section 3.4). In addition, by using min–max normalization and circular sampling methods, we enhance the baseline algorithm to minimize the number of sampling channels, thereby improving the scanning performance in terms of delay and energy consumption (Section 3.5)

- Implementation and evaluation of a prototype of CSCAN algorithm. We demonstrate the feasibility and effectiveness of our approach by implementing the proposed C-SCAN using Ubertooth [53], an open source Bluetooth-compliant board, on an Android-based platform. The extensive experiment-based evaluation demonstrates that C-SCAN can accurately detect channels with high accuracy in realistic wireless environments (Sections 3.6 and 3.7).

The remainder of this chapter is organized as follows. We summarize related research work in Section 3.2. Section 3.3 describes the system overview of our approach. We explain the baseline design of C-SCAN in Section 3.4. In Section 3.5, our baseline algorithm is optimized. Further, we discuss the implementation of our algorithm in Section 3.6. Section 3.7 presents the evaluation results, and Section 3.8 summarizes this chapter.

3.2 Related Work

Many articles and researches have reported lack of efficiency in Wi-Fi operations, which induce significant performance overhead and high energy consumption [7, 54-59]. Our measurement results shown in Fig. 3.3 confirmed that the energy trace of Wi-Fi soars as active scanning is performed. Even when connected to an AP, periodic channel scanning is triggered in the range of 1, 5, and 10 min to ensure stable link quality [10]. This is not only a main culprit of energy consumption but also acts as an interruption to data transmission, directly affecting communication performance.

Most of traditional solutions consider a single type of interface; they use active (i.e., in use) Wi-Fi interfaces for probing Wi-Fi channels. Thus, while scanning candidate channels, Wi-Fi throughput degrades due to the halt of data transmission and overheads incurred by Wi-Fi scanning. From a measurement study, we have found periodic outage intervals in received TCP sequence numbers during Wi-Fi scanning, as depicted in Fig. 3.4. In the same context, Hu *et al.* [60] have pointed out that increasing the number of probe messages to coarsely discover neighboring APs

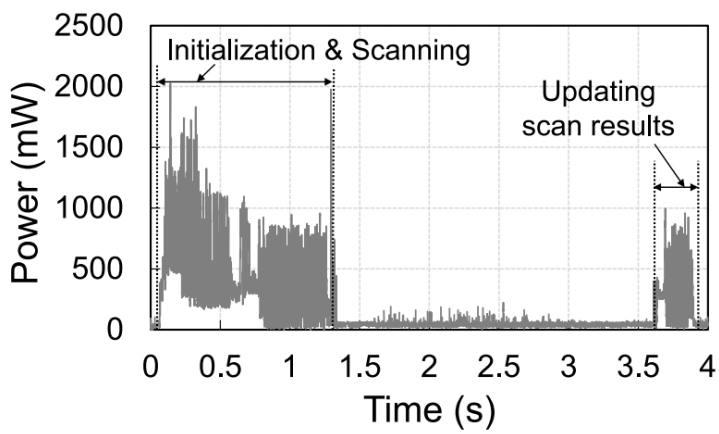


Figure 3.3: Energy trace of active Wi-Fi scanning.

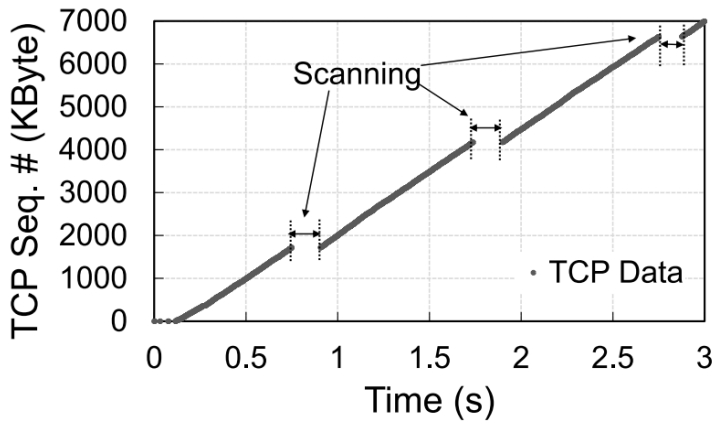


Figure 3.4: TCP sequence number diagrams for periodic active Wi-Fi scanning.

causes significant overheads that hinder channel utilization in a static environment. Likewise, various methods have been studied to eliminate the overhead or to reduce unnecessary scanning trials.

3.2.1 Wi-Fi Channel Discovery

In 802.11 channel discovery process, related works fall into the following broad categories: 1) efficient use of Wi-Fi radios and 2) offloading[2] Wi-Fi scan procedure in an attempt to optimize performance and device power consumption.

The first category usually takes the approach of adjusting the Wi-Fi's scanning parameters. For an instance, the scanning intervals are adaptively set based on measured AP interarrival time, AP density, and user velocity [9, 61]. Velayos and Karlsson [62] have theoretically explored the optimal values for MinCT and MaxCT. Recently, Wu *et al.* [63] and Xu *et al.* [64] have suggested higher values for those

[2] In this work, the term offloading refers to a computation offloading, i.e., reducing the burden of the standard Wi-Fi scanning's task, which distinguishes itself from data traffic offloading in cellular networks.

timers. Furthermore, a method has been developed to set adaptive timer values other than fixed ones [11]. However, since scanning parameter-based methods do not fit well for all environments and users, the concept of selective scanning has become popular [12]. In this method, Wi-Fi scans only a subset of channels with high probability of finding an AP based on experiences from previous scanning results, therefore, reducing the active duration of radio. Another strategy reported by Eriksson *et al.* [65], consists in a precomputed and stored probability of an AP operating in a channel to determine the scanning sequence. However in selective scanning, if any assumption for APs' presence is found to be wrong, it leads to full scan failure incurring additional costs. Additionally, efficient AP discovery methods aided by context information (e.g., sensory data [9], GPS [8, 66, 67], and cellular signals [68]) have been introduced.

The second category takes an approach that offloads parts of scan procedure to avoid the inherently expensive nature of Wi-Fi. There are prior works which offload partial upperlayer protocol to hardware; TCP/IP stack [69, 70] and ARP and ICMP [71] have been of major concerns to boost performance and save energy, respectively. Li *et al.* [72] have identified that the main processor typically consumes 1–2 time more energy than the Wi-Fi radio during the scan procedure, and achieved power gain by offloading the tasks of the processor. These designs require a secondary processor to perform the offloaded tasks.

3.2.2 Multi-Radio Cooperation Technology

There have been many efforts to utilize cooperation techniques between multiple same-type radios. In multiple-input multiple-output system, diverse cooperative multiple antenna techniques have been proposed to maximize the spectrum gain [73-

78]. The capability of spatial collaboration of the distributed antennas is investigated in [73-75]. To leverage the spatial diversity, various network coding scheme for cooperative communication have been introduced: space-frequency code [76], space-time code [77], and rateless code [78]. In cognitive ratio network, relaying strategies are adapted for cooperation with other nodes in the vicinity [79-82].

Different from cooperation with the same type of radio, several approaches utilizing secondary coexisting radios, such as Bluetooth and ZigBee, have been proposed [13-18]. The idea of using a secondary low-power radio for saving device power consumption is first proposed in [15]. Wake-on-WLAN [17] and Essense [18] allow Zigbee and Wi-Fi radios to communicate with special codes and obtain channel information. To support direct communication between two different protocols, they require substantial amount of software and/or hardware modifications. Zi-Fi [13], Wake-On-Wireless [83], and Turducken [84] promote a hands-off listening approach to obtain networks information. Zi-Fi is designed to search for beacon frames solely by analyzing statistics and periodicity of RSSI samples. However, it is challenging to grab time periodicity of beacon frames in a short period of time and furthermore, energy signature created by beacons is susceptible to noise, making it a harder task. Choi [52] proposed a lightweight algorithm so called WidthSense to detect Wi-Fi signals via correlation analysis between the RSSI measurements obtained on two Bluetooth channels.

However, none of these existing approaches using collocated low-power WPAN radios provides the detailed channel information of available APs in the vicinity, especially operating Wi-Fi channels. Our proposed scheme C-SCAN is also one of the kind that exploits secondary coexisting radio. Unlike these solutions, C-SCAN is not only to discover available Wi-Fi networks but also to provide the details of available Wi-Fi channels. Our approach may be viewed as advancing the solution discovering

the availability of Wi-Fi by leveraging coexisting WPAN radios. It offloads scanning procedure of Wi-Fi and reliably detects channels with multiple active APs.

Furthermore, we expect that our approach can be jointly combined with solutions of low-power Wi-Fi, such as *EMiLi* [85] and *Sampleless Wi-Fi* [86], to optimize Wi-Fi performance in terms of energy consumption. One example includes cooperation with *Sampleless Wi-Fi*. By saving energy during channel-sensing (with C-SCAN) and energy during transmission (with *Sampleless Wi-Fi*), a significant reduction of Wi-Fi interface power is expected.

3.3 C-SCAN Overview

In this section, we outline the problem of Wi-Fi channel scanning and present an overview of our approach toward the problem.

3.3.1 Problem Statement and Overview

Fig. 3.5 illustrates the high-level overview of our approach. C-SCAN aims to identify available APs on target Wi-Fi channels by using a low-power Bluetooth interface—or an 802.15 WPAN-compliant radio, such as a CC2420 RF transceiver

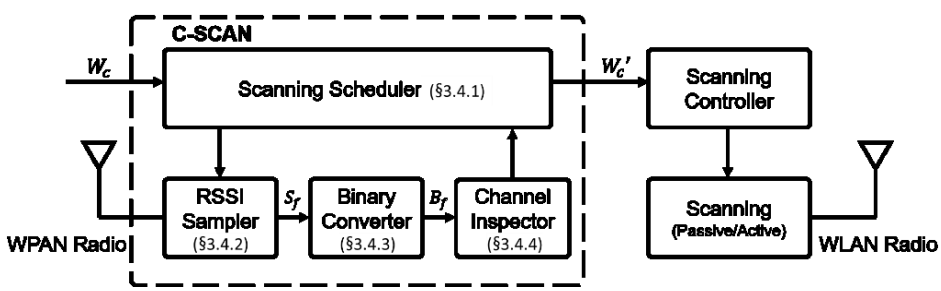


Figure 3.5: C-SCAN's high-level architecture.

for ZigBee [87] coexisting in a device. We consider the problem of Wi-Fi channel scanning in modern mobile devices.

Basically, the target Wi-Fi channels are provided by the original Wi-Fi management module of the system, e.g., WiFiManager in the case of Android smartphones [88]. Let W_C denote a set of target channels. For a given W_C, CSCAN inspects the presence of Wi-Fi signals based on the RSSI values obtained from the Bluetooth radio over the frequency range of target Wi-Fi channels. It first selects a Wi-Fi channel $w_i \in W_C$ and then tests the presence of an AP on w_i. When the operation of C-SCAN is complete, the scanning results, i.e., a set of discovered channels, denoted by $W_{available}$, are delivered to the Wi-Fi module. Then, the Wi-Fi scanning manager excludes the channels determined to be empty from WC and constructs a set of new target channels, $W_C' \leftarrow W_C \cap W_{available}$. Thus, it can perform scanning only on available Wi-Fi channels.

Wi-Fi and Bluetooth are available on most modern mobile devices as their connectivity are becoming more common. Even entry-level phones have Bluetooth connectivity. Hence, we envision that the proposed method can be applied to various types of Wi-Fi enabled systems, including smartphones, wearables, and IoT devices.

3.3.2 Background and Notations

The 2.4 GHz ISM band is by far the most crowded unlicensed ISM band, occupied by various wireless technologies, such as IEEE 802.11b/g/n Wi-Fi, ZigBee, and Bluetooth [89]. Fig. 3.6 depicts the channel map allocated for Wi-Fi and Bluetooth in the 2.4 GHz ISM band. The IEEE 802.11b/g/n defines 13 channels within this band, numbered 1–13. Each possesses a bandwidth of 20/22 MHz and they are spaced 5 MHz apart. The Bluetooth protocol divides the band into 79 channels (each

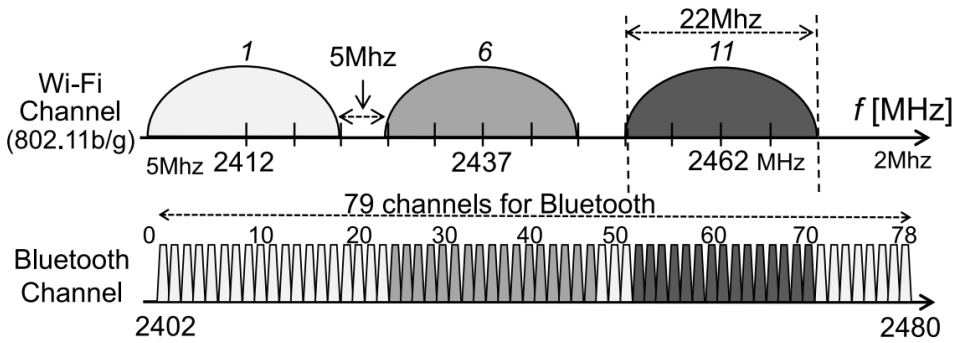


Figure 3.6: Wi-Fi and Bluetooth frequency channels in the 2.4 GHz ISM band

1 MHz wide), operating from 2.402 to 2.4835 GHz.

In the rest of this work, we use the notation w_i to represent a Wi-Fi channel i, and c_k to denote a Bluetooth channel k. For a Wi-Fi channel w and Bluetooth channel c, let $f_c(w)$ and $f_c(c)$ denote their center frequencies, respectively. Let ω denote a set of entire Wi-Fi channels, i.e., $\omega = \{w_1, w_2, ..., w_{13}\}$ and C represent a set of Bluetooth channels $\mathsf{C} = \{c_0, c_1, ..., c_{78}\}$. To describe the relation between Wi-Fi and Bluetooth channels, we define a *binary relation* $R_{overlap}$ on a Cartesian product $\omega \times \mathsf{C}$ as $R_{overlap} = \{(w, c) |$ Wi-Fi channel $w \in \omega$ is overlapping with Bluetooth channel $c \in \mathsf{C}\}$.

Definition 1 (Overlapping Relation): Given a Wi-Fi channel w, we define the set of all overlapping Bluetooth channels as $O_c(w) = \{c | (w, c) \in R_{overlap}\}$, e.g., $O_c(w_1) = \{c_0, c_1, ..., c_{19}\}$, as shown in Fig. 3.6.

Definition 2 (Channel Relation): Given a Wi-Fi channel $w_i \in \omega$, we also define a mapping function $g_c(w_i)$ that converts w_i to the Bluetooth channel c_k whose center frequency $f_c(c_k)$ is identical to $f_c(w_i)$. We can easily derive $k = 5i + 5$ as shown in Fig. 3.6. For example, $g_c(w_6) = c_{35}$.

3.3.3 Channel Identification Using Bluetooth Radio

The effectiveness of our approach hinges on accurate detection of Wi-Fi signals and their frequency bands, i.e., Wi-Fi channels, via a Bluetooth radio interface.

1) Pilot Experiment: To study the feasibility, we conducted a pilot experiment that uses Ubertooth, an open source Bluetooth sniffer [53], to measure the RSSI values of wideband Wi-Fi signals over narrowband Bluetooth channels. For the measurement, we deployed one AP and one Wi-Fi client on channel 6, w_6, in a large underground parking lot where no radio interference signal in the 2.4 GHz band was present. We set the AP as a sender and generated UDP traffic using iPerf3 with a constant packet generation rate of 5 Mbps. For the frequency range of Wi-Fi channel w_6, i.e., $O_{map}(w_6)$, the RSSI values are measured by the Bluetooth-compliant CC2400 wireless transceiver in the Ubertooth when the Wi-Fi signal is present. Hence, we designed the experiment to collect RSSI values on the Bluetooth channels $[c_{20}, c_{50}]$ (as depicted in Fig. 3.6) at four different distances (i.e., 1, 5, 15, and 35 m) from the AP. For data processing, we used a carrier sense (CS) threshold of -80 dBm to filter out low RSSI values.

2) Results and Observations: Fig. 3.7 plots the average RSSI for Wi-Fi frames sensed by a Bluetooth interface and a Wi-Fi interface. Fig. 3.8 plots the distributions of the RSSI values collected from the channels $\forall_c \in [c_{20}, c_{50}]$. From Fig. 3.7 and 3.8, we make the following observations.

O1: Although a high degree of variation is observed in the RSSI values, it is clear that Bluetooth radios can sense Wi-Fi signals and their strength in the range of overlapping frequencies, i.e., $O_c(w_6)$, in spite of different characteristics, such as transmission range and sensitivity. Further, as expected, we observe a wide "flat" section in the measured RSSI values, which correctly characterizes the coherence

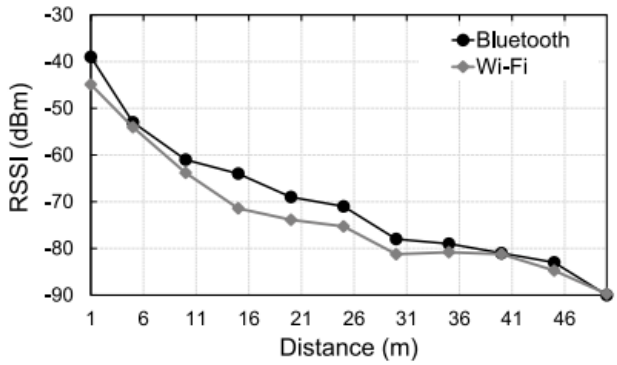


Figure 3.7: Average RSSI for Wi-Fi frames sent from an AP at different distances, each sensed by a Bluetooth interface and a Wi-Fi interface.

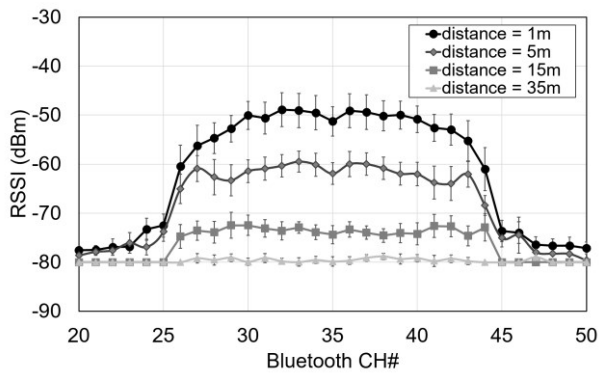


Figure 3.8: Measurement results of RSSI values of Wi-Fi signals on channel 6 using an open source Bluetooth sniffer at different distances.

bandwidth[3] of Wi-Fi signals with the center frequency of channel w_6

O2: The observed coherence bandwidth of Wi-Fi signals is larger than 10 MHz but smaller than 20 MHz. Therefore, the RSSI values measured on the three adjacent channels $w_{i-1(=5)}$, $w_{i(=6)}$, and $w_{i+1(=7)}$ spaced 5 MHz apart, have approximately

[3] Coherence bandwidth is the range of frequencies where the channel is flat [90].

identical distributions regardless of the distances. Meanwhile, although RSSI values greater than the CS threshold are observed on the channels 10 MHz apart from $f_c(w_6)$, their RSSI distributions have smaller values.

The above-mentioned observations inspire us to design an algorithm to pinpoint the operating channel number of Wi-Fi signals via a Bluetooth radio. Consider a Wi-Fi system (e.g., AP) operating on channel w. Clearly, in the range of Bluetooth channels $O_c(w)$, a Bluetooth radio can sense its transmissions (e.g., data and beacon frames) and construct time-series of RSSI samples. Then, the RSSI sequences measured on two channels $c_1, c_2 \in O_c(w)$ $(c_1 \neq c_2)$ will be strongly correlated (here, the degree of correlation between two time series will vary depending on the target channels). Therefore, by inspecting the RSSI sequences obtained on $O_c(w)$ with a proper algorithm, we can identify the presence of any Wi-Fi system(s) on channel w. Based on this motivation, we design our solution and explain the details in what follows.

3.4 C-SCAN Design

In this section, we explain the details of C-SCAN based on its system architecture shown in Fig. 3.5.

3.4.1 Scanning Scheduler

Scanning scheduler is the main component of C-SCAN that manages flow control between components. Given a set of candidate Wi-Fi channels $W_C \subset \omega$ received from the original scanning service [4] (e.g., WifiManager), *scanning scheduler*

[4] Due to the space limit, we omit the details on the scan procedure in the IEEE 802.11 standard. The interested reader can refer to [11] for further details.

identifies the set of available channels $W_{available} \subset W_C$. To this end, it sets the sequence of target frequencies and executes *RSSI sampler* accordingly.

In C-SCAN's baseline algorithm, all channels in W_C are scanned one by one to determine the presence of available APs. For example, for $W_C = \{w_4, w_6, w_1\}$ the scan is performed on the WPAN frequency band corresponding to w_4, and the process proceeds to channel w_6 and w_1 in order. Therefore, the algorithm is executed $|W_C|$ times.

When the scanning and inspection over target channels are complete, *scanning scheduler* generates a set of available channels $W_{available}$, a subset of W_C. The result is passed to the Wi-Fi scanning manager, as illustrated in Fig. 3.5. Hence, the manager performs scanning *selectively* with the set of new candidate channels $W_C' \leftarrow W_C \cap W_{available}$.

3.4.2 RSSI Sampler

RSSI sampler is invoked by *scanning scheduler* with a target Wi-Fi channel w, and it performs RSSI measurements on n scanning points $(f_1, f_2, ..., f_n) \in O_c(w)$ via the collocated Bluetooth radio.

Fig. 3.9 and 3.10 show the RSSI sampling mechanism adopted by CSCAN. *RSSI sampler* measures RSSI values on a channel $f \in (f_1, f_2, ..., f_n)$ every τ and changes the channel from one to the next one in a circular fashion over the n scanning points. The parameter τ is the sampling period or slot time, and is set to 160 us in our implementation. Let $f(t)$ denote the sampling channel of *RSSI sampler* at time slot t, which is given by

$$f(t) = f_{(t \text{ modulo } n)+1},$$

where $t = 0, 1, 2,...$ and $f(t) \in (f_1, f_2, ..., f_n)$. A n-tuple of RSSI samples,

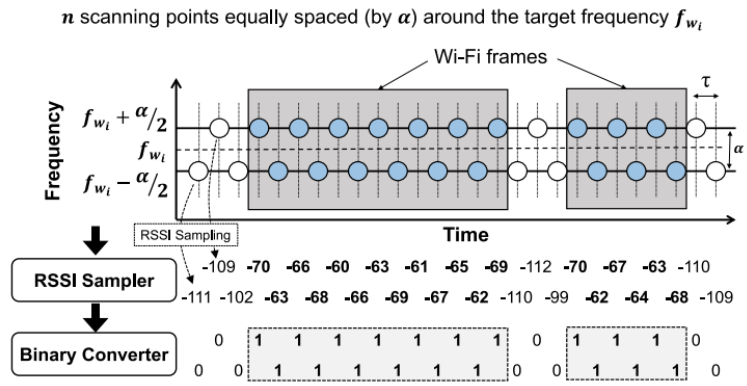


Figure 3.9: Basic operations of C-SCAN algorithm with two scanning points.

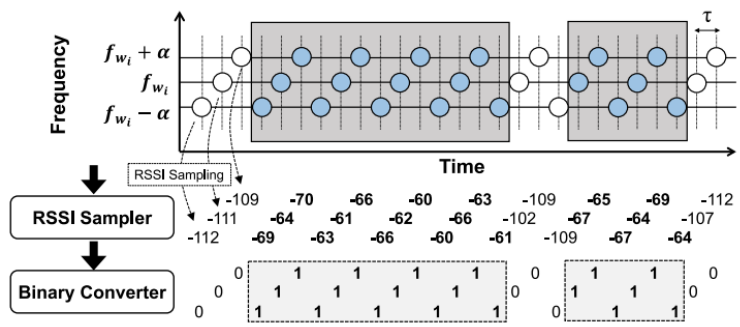


Figure 3.10: Basic operations of C-SCAN algorithm with three scanning points.

denoted by (s_{f_1}, s_{f_2}, ..., s_{f_n}), is generated and each element is updated every slot time τ. Here, s_{f_i} indicates an RSSI value measured on f_i. Then, the tuple is constantly fed to C-SCAN's *channel inspector* for further examination. The parameter n should be at least two, whereas the minimum distance between each scanning point should be wider than 2 MHz to ensure Wi-Fi signal detection. Hence, unless the channel is overloaded, signals from narrowband protocols such as ZigBee or Bluetooth cannot affect more than two scanning points simultaneously. We design the *RSSI sampler* so that its scanning points are distributed around the center

frequency of the target Wi-Fi channel with a constant interval, α, as depicted in Fig. 3.9 and 3.10.

In the baseline algorithm, the number of scanning points is set to the minimum value 2 for simplicity and the distance between them ($|f_2 - f_1|$) is set to 10 MHz. To be specific, when a target Wi-Fi channel i with center frequency f_{w_i} is passed down, a tuple of scanning points (f_1, f_2) is mapped onto ($f_{w_i} - 5$, $f_{w_i} + 5$). Since the Wi-Fi channels are spaced 5 MHz apart from adjacent channels, the tuple can be represented as ($f_{w_{i-1}}$, $f_{w_{i+1}}$). The reason is to consider the effective bandwidth of the Wi-Fi's signal and make every scanning points independent of neighboring Wi-Fi channels' signals.

3.4.3 Binary Converter

Binary converter takes series of raw RSSI sample sequences and converts each value into a binary digit, 0 or 1, forming a binary string of length n: (b_{f_1}, b_{f_2}, …, b_{f_n}) where b_{f_i} denotes a binary bit that represents the state of frequency f_i. If an RSSI value collected from a scanning point shows no traces of signals, the corresponding bit is set to 0 to indicate that the frequency is idle. Otherwise, if the sample exhibits any signal presence, the bit is set to 1 to indicate the busy state of the channel.

Although there are several ways to decide whether a frequency is idle or busy, the baseline algorithm chooses the most intuitive criteria: the CS threshold (Th_{cs}) of −80 dBm. Specifically, a bit is set to 0 if the RSSI value is less than the CS threshold and 1 otherwise. The converted binary strings are passed to *channel inspector*

Binary strings of length n have 2^n different combinations, but only small

portions are meaningful. A converted binary string may indicate a sign of Wi-Fi signal when it contains consecutive 1's. Since signals from a Wi-Fi channel range over multiple adjacent scanning points, strings with a single 1 or no 1's, and strings with no consecutive 1's are safely ignored during further AP detection process.

3.4.4 Channel Inspector

Channel inspector receives a series of length n binary strings to make the final decision on presence or absence of Wi-Fi APs in a channel. As mentioned, a binary string may indicate a sign of Wi-Fi signal only when it contains consecutive 1's. The baseline algorithm assumes that there is a Wi-Fi signal when both scanning points are considered busy (i.e., a binary string of "11"). We then can infer the length of a signal by looking at the number of successive signs. For an example, if a binary string 11 is observed three times in a row, the length of signal should be around $3 \cdot \tau$. The *channel inspector* keeps track of the length of valid signals and counts occurrences of signals based on length, during T.

Short signals are likely to appear as noise in dynamic environments and, conversely, signals that last relatively longer imply that they are reliable signs of Wi-Fi APs. Therefore, we score the likelihood of AP presence in a channel by reflecting the weight in accordance with the length of the detected signals. The equation that scores the likelihood of AP presence on channel w_i is as follows:

$$\text{Score}_{w_i} = \sum_{l=1}^{L} l \cdot N_{w_i}^{l}$$

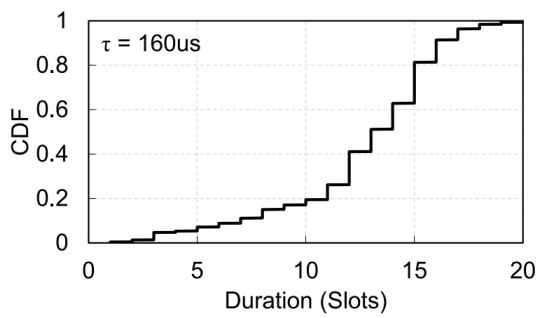
where l denotes the length of a signal observed during T, and $N_{w_i}^{l}$ is the number of times w_i's signal with length l has been detected during the scanning period T. The larger the l is, the more the count contributes to the final score. We obtain the

threshold(θ) of the score to judge the AP presence from heuristic data we obtained from beacon length experiments, as shown in Fig. 3.11.

Therefore, false negative (FN) rates are kept under 5%. In the experiments, a number of beacons were collected from various places like subway stations, cafes, and offices to reflect real-world environments. The beacon length distribution is represented in form of a CDF. Since the majority of beacons prolonged on air fairly long, we safely assumed that the signals of length equal to or less than 4 slots are noises with high probability. A beacon is transmitted (T/beacon interval) times during the scan and hence, θ is set to 4·(T/beacon interval). If the target Wi-Fi channel scores higher than θ, it is considered to contain at least one AP. *Channel inspector* passes the result to *scanning scheduler* so that the target channel is included in the available channel set ($W_{available}$) sorted by its score.

3.4.5 Experiments

We first implement the baseline algorithm on the application layer to validate its effectiveness for a massive RSSI dataset we collected from a set of controlled environments. To closely observe the behavior of the algorithm, we run C-SCAN's *RSSI sampler* using an Ubertooth transceiver in an anechoic chamber where wireless



Slots	CDF
1	0.004199
2	0.013296
3	0.046886
4	0.053184
. . .	
17	0.963611
18	0.983905
19	0.993002
20	1

Figure 3.11: CDF of beacon lengths at various locations (e.g., subway, cafe, etc.)

signals from outside are completely blocked. To cover different channel densities, traffic densities, and network topologies, we vary number of APs and mobile clients, traffic loads, and their placements in terms of distance. In a single-AP scenario, one AP is used to test the baseline algorithm varying few external environmental factors. The AP is connected with a laptop by wire to generate 0 or 5 Mbps UDP traffic. Additionally, it is paired with a mobile client and their distance from the Ubertooth transceiver is varied {0, 5, and 15 m}. During the whole experiment, two scanning points of the *RSSI sampler* are fixed to detect the target Wi-Fi channel 6 while the AP is located on channel 5–7 in each single-AP scenario, respectively. In multiple-AP scenarios, two APs are located on channels {4, 8}, {5, 7}, {6, 7}, and {6, 8}, and three APs are deployed on channels {5, 6, 7}, again varying the UDP loads and distances. For each scenario, we run the algorithm over the dataset 100 times and measure the average score and detection ratio.

Our experimental results (Fig. 3.12 and 3.13) show that the baseline CSCAN can reliably identify the presence of an AP on a target channel and reject channels that involve no Wi-Fi activities under both light and heavy traffics. In a multiple-AP scenario, the baseline C-SCAN can always detect the AP on the target channel, as depicted in Fig. 3.13. The false positive detection rate (FPR) is kept relatively low, but it grows rapidly under environments where APs are closely located and traffics are heavily loaded.

3.4.6 Discussion

Although the experimental results seem promising in controlled environments, there are still challenges in exploiting the baseline algorithm in real-world environments.

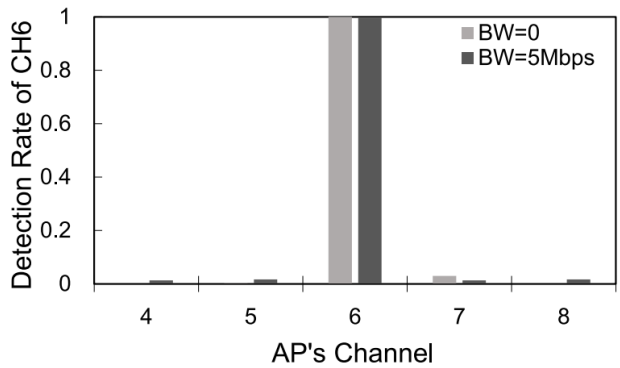


Figure 3.12: Detection rate for channel 6 using the baseline algorithm in controlled environments with a single AP located nearby.

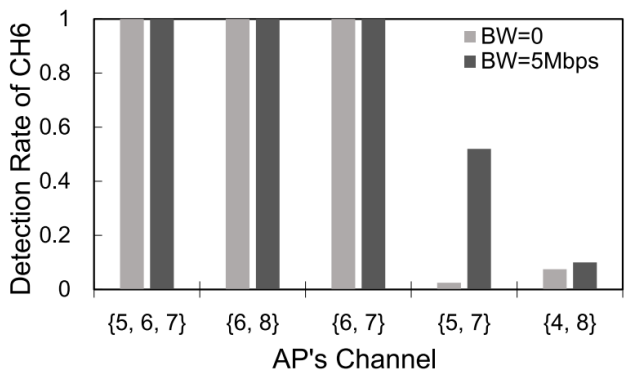


Figure 3.13: Detection rate for channel 6 using the baseline algorithm in controlled environments with multiple APs located nearby.

1) O(n) Time Complexity: For a given target channel set W_C, the latency of the baseline C-SCAN algorithm, which works in the middle of existing Wi-Fi scanning procedure and stays on each channel $w \in W_C$ for T to check any AP presence, is not ideal for a scanning service that requires a certain degree of agility, because the scanning delay increases in proportion to the number of target channels $|W_C|$. If it incurs too much time overhead, incorporating C-SCAN has no merits.

2) Impact of Neighboring Interference: Signals from neighboring channels often affect scanning points located outside their effective bandwidth. Due to imperfections of hardware and different techniques of manufacturers, the Wi-Fi's signal tends to overflow out of its 20 MHz bandwidth, especially at close distances. If APs on adjacent channels release a signal that confuses both scanning points of the baseline algorithm, it will ring a false positive (FP) alarm in a situation in which no AP exists on the target channel. Even in the ideal situation where neighboring APs do not affect both scanning points, problems can still occur. In a lightweight traffic scenario, signals from different APs are unlikely to collide, potentially aided by CSMA/CA. However, the effect of CSMA/CA is limited (e.g., due to hidden terminal problem) and the problem gets worse in a heavy traffic scenario, inducing collisions between signals. When signals from neighboring APs located on the channel at both ends (i.e., w_{i-1} and w_{i+1}) of the target channel (w_i) coincide, both scanning points are affected and generate an FP detection of channel w_i.

3.5 Enhanced C-SCAN

In this section, we propose an enhanced version of the C-SCAN algorithm that performs low-latency operations and achieves excellent levels of stability in heavy-traffic scenarios with multiple APs.

3.5.1 Minimization of Scanning Points

In the baseline algorithm of C-SCAN, we use the number of scanning points $n = 2$ for RSSI sampling on each Wi-Fi channel, which is the minimum number required to verify the presence of an AP on a *single* Wi-Fi channel. However, as described

above, this approach may suffer from the increased scanning delay when the number of target channels is large.

Our key observation to minimize the scanning latency is that, with a certain degree of scanning points, e.g., $n = 3$, we can inspect more than one channel at a time. For example, consider a situation where C-SCAN scans and inspects a Wi-Fi channel w_i with three scanning points ($f_{w_{i-1}}$, f_{w_i}, $f_{w_{i+1}}$) for a given target channel w_i as shown in Fig. 3.14. *RSSI sampler* will generate a sequence of RSSI samples, ($s_{f_{w_{i-1}}}$, $s_{f_{w_i}}$, $s_{f_{w_{i+1}}}$). In an ideal case, a sequence of consecutive binary stings of "110," "111," and "011" imply the presence of APs on channels w_{i-1}, w_i, and w_{i+1}, respectively. This implies that we can monitor three consecutive Wi-Fi channels, i.e., w_{i-1}, w_i, and w_{i+1}, with a single execution of C-SCAN.

It is possible to monitor more channels at once by increasing the number of scanning points. However, it is not cost-free and there is a tradeoff between monitoring granularity and monitoring scope. Since we are trying to collect samples over multiple frequencies using a single radio, if the number of frequencies to be hopped increases, the sampling period regarding each scanning point also increases naturally. This in turn reduces the monitoring granularity. Specifically, given a sampling period of a WPAN radio, the slot time τ, it takes $n \cdot \tau$ to collect two

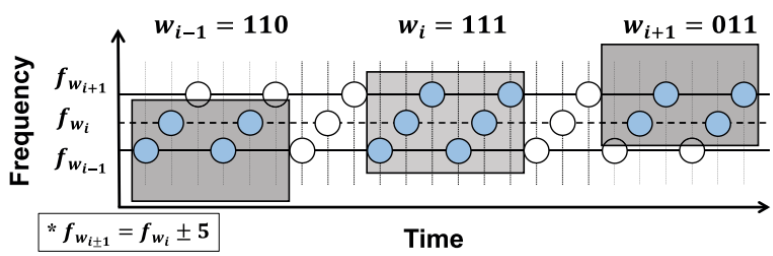


Figure 3.14: Multiple channel scan with binary string classification in the enhanced C-SCAN algorithm.

consecutive RSSI samples from a single target frequency. If signals of lengths shorter than $n \cdot \tau$ move in the air, they are unlikely to be detected. Thus, adopting large n may cause too many signals uncatchable, raising FN rates.

Based on this reason, we choose $n = 3$ for the enhanced algorithm since it holds a good balance between monitoring granularity and monitoring scope. To inspect all candidate Wi-Fi channels $\forall w \in W_C$, *scanning scheduler* bundles as many groups of three consecutive Wi-Fi channels as possible, and schedules C-SCAN operations at the center frequencies of the center channels belonging to three Wi-Fi channels. For each iteration, *RSSI sampler* maps three scanning points (f_1, f_2, f_3) onto ($f_{w_{i-1}}$, f_{w_i}, $f_{w_{i+1}}$) for given target channel w_i, and generates a sequence of RSSI samples ($s_{f_{w_{i-1}}}$, $s_{f_{w_i}}$, $s_{f_{w_{i+1}}}$) by alternating between three sampling points in the same circular manner as depicted in Fig. 3.14. Note that the scanning points are now spaced 5 MHz apart from their neighbors (10 MHz space was used in the baseline algorithm).

3.5.2 Sample Normalization and Similarity Analysis

One challenging issue to realize this approach is that APs on neighbor channels can cause FP alarms. For example, although Wi-Fi signals are on channel w_{i-1} (e.g., the first gray box in Fig. 3.14), all RSSI values measured on three sampling points ($f_{w_{i-1}}$, f_{w_i}, $f_{w_{i+1}}$) could be above a predefined threshold, e.g., -80 dBm due to the time-varying nature of wireless media or co-channel interference. Then, its binary string will be represented as 111 instead of 110, which is falsely recognized as w_i.

To address this issue, we present a min–max-based sample normalization and similarity analysis method. As observed from our pilot experiments in Section 3.3.3,

the coherence bandwidth of Wi-Fi signals is larger than 10 MHz but smaller than 20 MHz. To exploit this, we analyze the similarity between RSSI values in a sample ($s_{f_{w_{i-1}}}$, $s_{f_{w_i}}$, $s_{f_{w_{i+1}}}$) to generate a binary string from the measured RSSI values, instead of simply comparing their values to a predetermined threshold. Prior to similarity check, we normalize them in order to alleviate the effect of RSSI variation caused by differing distances to the APs. Among the various normalization techniques in data mining [91], *min–max*-based normalization better conforms with the requirements of our implementation for the following reasons.

1) *Min–max* algorithm, also known as *feature scaling*, is suitable for standardizing features of data whose range varies widely. It correctly identifies the frequency-domain similarity between n scanning points, where overlapping wideband Wi-Fi channels with different distance provoke different ranges of RSSI data.

2) It provides computational simplicity since the sample size is always fixed at n. Normalizing RSSI values in each sample is done extremely fast and efficiently, which is a crucial property in detecting Wi-Fi signals in real-time.

3) Noisy RSSI signals are effectively ignored and their influence is minimized to ensure stability, since normalization and similarity comparison only focus on good signals, i.e., signals with RSSI values above the CS threshold (-80 dBm).

The threshold value was carefully chosen from our pilot experiment Section 3.3.3. In evaluation part, we verify that the threshold of -80 dBm can effectively find nearly all available APs in vicinity. Few APs, which C-SCAN could not detect, were neither found by original Wi-Fi scan nor usable due to bad link quality. The min–max-based normalization can be written as

$$s' = (s - Min)/(Max - Min)$$

where s is a raw RSSI value in a sample, which is greater than -80 dBm, and s' is

normalized value of range [0, 1]. Max and Min are determined as follows:

$$Min = \begin{cases} \min\left(S_{f_{w_{i-1}}} \cup S_{f_{w_i}} \cup S_{f_{w_{i+1}}}\right), & if\ \min\left(S_{f_{w_{i-1}}} \cup S_{f_{w_i}} \cup S_{f_{w_{i+1}}}\right) > -80 \\ -80, & otherwise \end{cases},$$

$$Max = \max(S_{f_{w_{i-1}}} \cup S_{f_{w_i}} \cup S_{f_{w_{i+1}}}).$$

where $S_{f_{w_i}}$ denotes the set of RSSI samples collected from f_{w_i} during the scanning period T. *Binary converter* confirms with a similarity check if active signals are actually from Wi-Fi channels or if they are simply composite noises. Similarity between two RSSI samples is simply computed by the following equation:

$$\mathrm{Sim}(s_i, s_j) = 1 - |s_i' - s_j'|.$$

Signals with mask 110, 111, and 011 imply the presence of APs on channels w_{i-1}, w_i, and w_{i+1}, respectively, for a given target channel w_i. When all RSSI values are above -80 dBm, the similarity check is done for all three possible pairs: if $\mathrm{Sim}\left(s_{f_{w_{i-1}}}, s_{f_{w_i}}\right)$, $\mathrm{Sim}\left(s_{f_{w_i}}, s_{f_{w_{i+1}}}\right)$, and $\mathrm{Sim}\left(s_{f_{w_{i-1}}}, s_{f_{w_{i+1}}}\right) \geq \delta$, with δ as a predefined threshold, it is confirmed to be a binary string of 111. If the first or last two RSSI values are above -80 dBm, the similarity check $\mathrm{Sim}\left(s_{f_{w_{i-1}}}, s_{f_{w_i}}\right) \geq \delta$ is done for 110 or $\mathrm{Sim}\left(s_{f_{w_i}}, s_{f_{w_{i+1}}}\right) \geq \delta$ is done for 011. This step alleviates FP alarms caused by neighboring channel dependency. Among the available signal masks 110, 111, and 011 in the *binary converter*, only the ones confirmed to be similar are counted for scoring in the next step.

To search for the optimal value of the threshold δ, we measured the detection performance of enhanced C-SCAN in true positive rate (TPR) and FP rate (FPR), as shown in Fig. 3.15, 3.16, and 3.17. Under controlled environments, where an AP was located at the target channel, a channel away from the target, and two channels away from the target, the similarity δ of value 0.6 showed the highest TPR as well as the

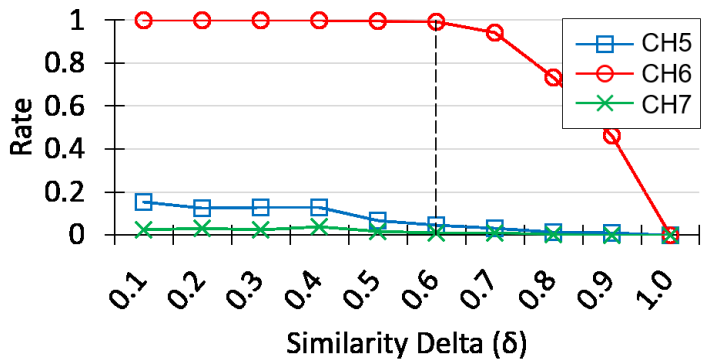


Figure 3.15: TPR of CH6 and FPR of CH5 and CH7 when AP is on channel 6.

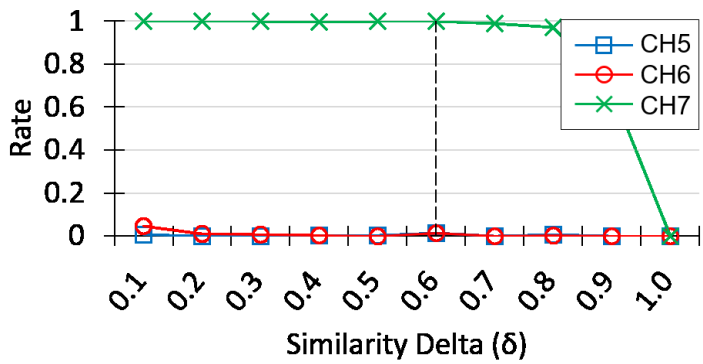


Figure 3.16: TPR of CH7 and FPR of CH5 and CH6 when AP is on channel 7.

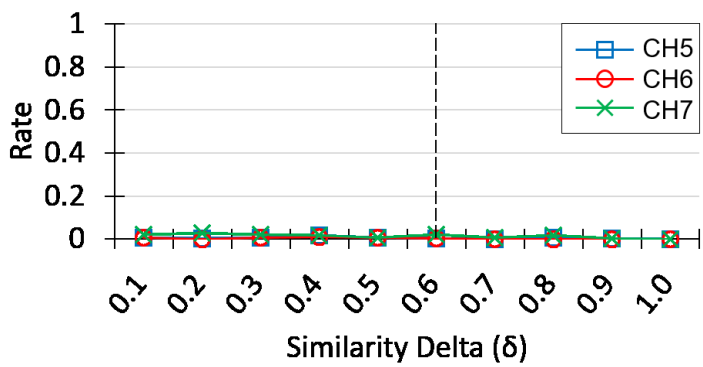


Figure 3.17: FPR of CH5, CH6, and CH7 when AP is on channel 8.

lowest FPR. Hence, in subsequent performance evaluations of enhanced C-SCAN, the threshold value δ is set to 0.6.

3.5.3 Results

Under the same controlled environments as in the baseline C-SCAN algorithm, including both single- and multiple-AP scenarios with varying bandwidth and distance, we again run *RSSI sampler* using an Ubertooth transceiver on the target Wi-Fi channel 6. Then, enhanced C-SCAN is shown that the neighboring channels, 5 and 7, can also be monitored. In Fig. 3.18 and 3.19, the results show that the enhanced C-SCAN can cover three channels at a time, reliably identifying the presence of an AP on a target channel and rejecting channels that involve no Wi-Fi activities, under both light and heavy traffic. The enhanced C-SCAN achieves high detection performance even in the multiple-AP scenario. The FPR is kept low in most cases, but it increases lightly under the environments in which APs are closely located and heavy traffic appears. Note that unlike the real-world environment where APs are randomly distributed, the distances between APs and Ubertooth transceiver are kept approximately constant in multiple-AP controlled environments, which makes it difficult to perform accurate similarity analysis. In Section 3.7, we evaluate the performance of enhanced C-SCAN in real-world environments and show that FPR is effectively mitigated by a similarity analysis.

3.6 Implementation of C-SCAN in Android

We implemented a prototype of C-SCAN by using Ubertooth One, an open-source 2.4 GHz wireless development board, in Android devices. We developed all

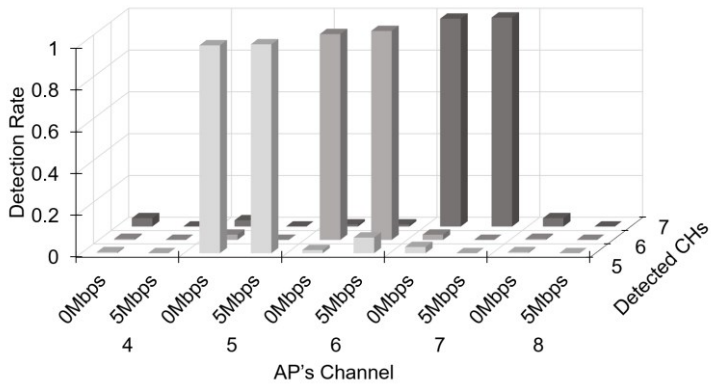


Figure 3.18: Detection rate of channels 5–7 using the enhanced algorithm in controlled environments with a single-AP located nearby.

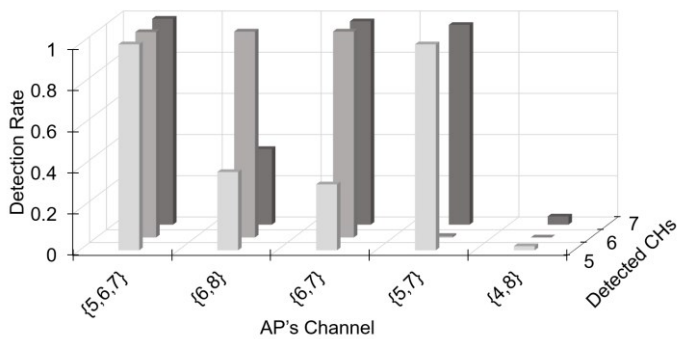


Figure 3.19: Detection rate of channels 5–7 using the enhanced algorithm in controlled environments with multiple APs located nearby.

components described in Sections 3.4 and 3.5 as Android application and/or Ubertooth firmware. For the target mobile device, we used an Android tablet (Google Nexus 7 - 2nd version) equipped with a Qualcomm Atheros WCN3660 Wi-Fi chipset. Then, Ubertooth was attached to the tablet via a Micro USB Gender, as shown in Fig. 3.20.

Fig. 3.21 depicts the implementation architecture of C-SCAN on Android platform. Two core components of it, *scanning scheduler* and *channel inspector*, are

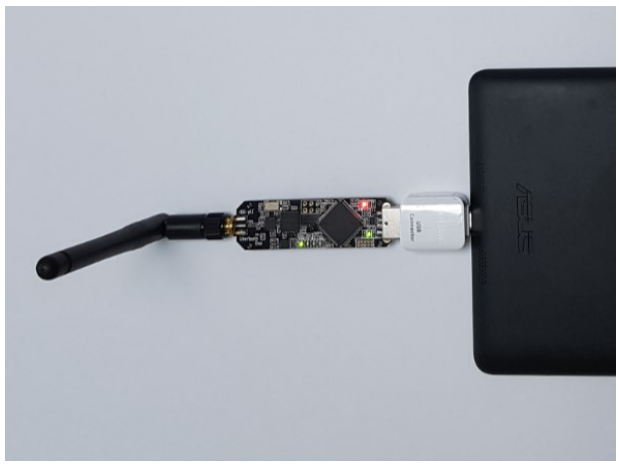


Figure 3.20: Prototype of C-SCAN that consists of a tablet and Ubertooth connected through a micro OTG gender.

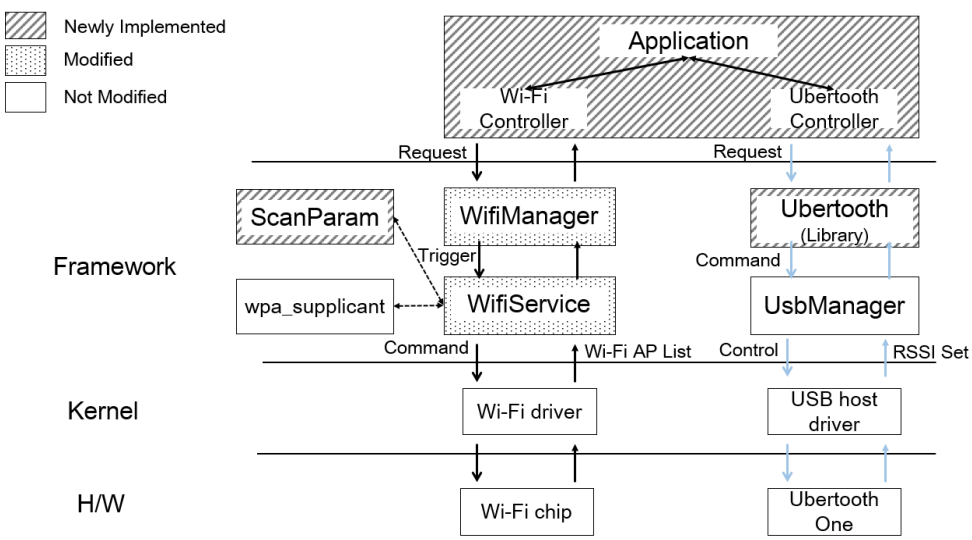


Figure 3.21: Implementation architecture of C-SCAN on Android.

implemented in Java and run in a standalone Android application. By implementing an additional integrated controller on the application layer, we enable *scanning scheduler* to interact with Ubertooth and the Wi-Fi chipset through the USB interface and Android's functional API, respectively. *scanning scheduler* obtains a list of target scanning channels from the system configuration file or user specification. It invokes *RSSI sampler* in the firmware of Ubertooth via the USB interface with a set of target channels and the sampling period as parameters.

In addition, we modified Android's scanning module at the framework layer to implement *selective scanning*. By default, Android is designed to scan the full set of Wi-Fi channels, and the related scan parameters defined in wpa_supplicant, such as interval and timeout values, are fixed and not configurable at run-time. We added new features to the framework layer to scan only the specified set of Wi-Fi channels using user-configurable scan parameters, instead of the entire channels. *RSSI sampler* is implemented in the firmware of Ubertooth. We extend the original firmware to accomplish circular channel hopping and RSSI sampling. In particular, the sampler reads an RSSI value from the CC2400 transceiver every $\tau = 160$ us and hops to the next channel. The sampler stores the RSSI samples into a buffer and transfers them to *channel inspector* via the USB interface.

3.7 Performance Evaluation

In this section, we present the performance evaluation results of two C-SCAN algorithms: 1) *baseline* and 2) *enhanced*. We compare the performance of C-SCAN with two Wi-Fi standard scanning methods: 1) active and 2) passive. *Legacy* denotes the original Wi-Fi scan method of an Android device. We first describe the experimental setups and parameters, and then discuss the results in terms of various

performance metrics: detection accuracy, detection latency, energy consumption, and throughput.

3.7.1 Experimental Setups and Parameters

In order to demonstrate the feasibility of our solution, we conduct experiments in real-world environments with various channels conditions and network topologies. We carried out the experiments in lots of places, including subway platforms, cafes, and office buildings in Seoul, South Korea, where the channel distributions of deployed APs are very diverse. Depending on the degree of channel occupancy, the experiment results are classified into three states: 1) *sparse*; 2) *moderate*; and 3) *dense*. *Sparse* represents environments where less than 30% of all Wi-Fi channels are occupied by nearby APs. The percentage of occupied channels in *dense* environments are 50% or higher, and *moderate* environments have a percentage between 30% and 50%. We configure system parameters of C-SCAN with reference to the Android devices' default settings. The details of parameters for Ubertooth and Wi-Fi are given in Table 3.1.

3.7.2 Detection Accuracy

First, we evaluate the detection accuracy of the two algorithms (baseline and enhanced). We conduct experiments in multiple places and repeat them 100 times at each place. For the evaluation, we use two standard accuracy metrics: 1) FP and 2) FN. We define an FP as the result when C-SCAN detects an active channel when there is no AP on the target channels. An FN is defined as the result when C-SCAN incorrectly indicates that a channel is inactive (i.e., no APs detected) but at least one AP exists on the channel.

Table 3.1: Parameter settings for C-SCAN.

Parameter	***Values***
Carrier Sensing (CS) Threshold (Th_{cs})	-80 dBm
Sampling Timer (T)	102.4 ms
Maximum Round (α)	3
Similarity Delta (δ)	0.6
Theta (θ)	10
Scan Interval (I)	10 s
MinChTime ($MinChTime$)	10 ms
MaxChTime ($MaxChTime$)	102.4 ms

Fig. 3.22 and 3.23 illustrate the detection accuracy of the baseline and enhanced algorithms based on FP and FN rates for different channel conditions depending on the occupancy of nearby Wi-Fi APs: sparse, moderate, and dense. The result shows that both algorithms achieve very low FN rates (approximately zero) in sparse and moderate environments. Further, the results also show that the overall error rate of the baseline algorithm, as expected, is greater than that of the enhanced algorithm because of its simple *binary converter* which causes FPs on adjacent channels. The FP rate of the enhanced algorithm is significantly decreased compared with that of the baseline algorithm: 16% (sparse), 30% (moderate), and 15% (dense) on average.

In dense environments, however, both fail to maintain accuracy. This is mainly due to the occurrence of FNs on some active channels. On these channels, there were very few frames whose signal strengths are stronger than the CS threshold.

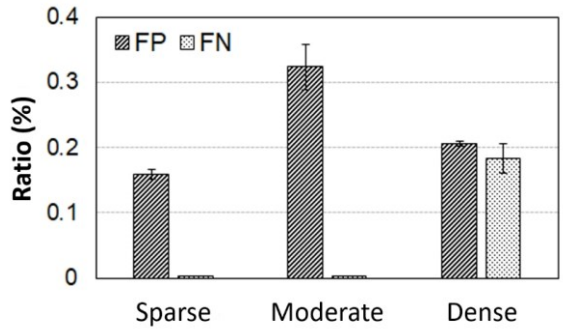


Figure 3.22: Detection accuracy of C-SCAN baseline algorithm.

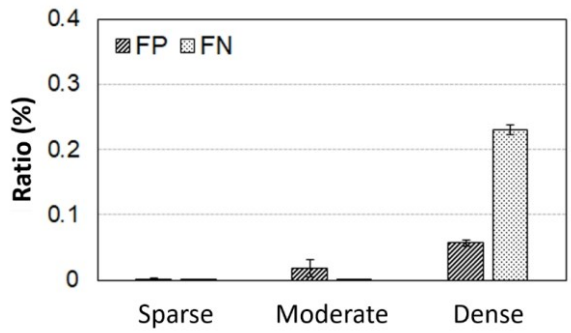


Figure 3.23: Detection accuracy of C-SCAN enhanced algorithm.

Fig. 3.24 shows the cause of poor performance in dense environments. We gather AP scan results using the Wi-Fi interface at the same places and compare the number of detected Wi-Fi APs on each channel: 1) Wi-Fi interface; 2) baseline algorithm; and 3) enhanced algorithm via the Bluetooth-compliant transceiver. During experiments, the number of identified APs varies according to the received signal strength of frames from APs on each channel. In this environment, six channels are occupied by nearby APs: 1, 3, 4, 7, 9, and 11. In particular, on channel 4, the average number of APs being identified via the Wi-Fi interface is only 5.25. Compared with the scan results of the Wi-Fi interface, the average FN rates of baseline and enhanced algorithms decrease from 18.3% to 10.4% and from 23% to 15.3%, respectively.

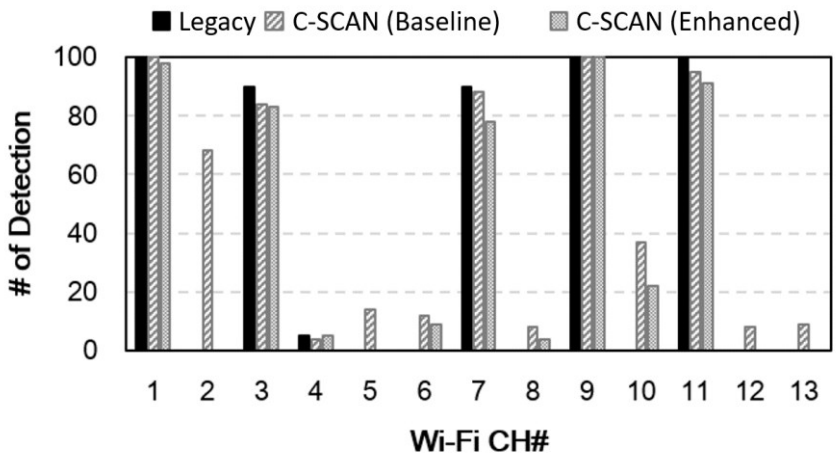


Figure 3.24: Comparison of AP detection counts between Wi-Fi interface and C-SCAN (baseline and enhanced) in dense environment.

3.7.3 Detection Latency

Detection latency is one of the most important performance metrics in Wi-Fi scanning. We measure the total time taken to obtain a scanning result of C-SCAN, i.e., the Wi-Fi channel information of vicinity. In particular, latency is defined as the time difference between when C-SCAN starts channel scanning for all Wi-Fi channels and the end. Here, C-SCAN operates the channel discovery until the sampling timer expires and gives several tries to identify the absence of AP if no Wi-Fi signal is detected on the channel. We compare the difference between the latency of baseline and enhanced algorithm.

We first compare the delay performance of C-SCAN with legacy. For active Wi-Fi scanning (Fig. 3.25), the scan times of both C-SCAN algorithms are shorter than that of legacy because the first do not send unnecessary probe requests on channels where no AP exists. We observe that the latency of the enhanced algorithm is shortest due

to low FPs that occur on adjacent channels. Similarly, when the Wi-Fi interface does AP scanning passively (Fig. 3.26), the latency of C-SCAN is significantly reduced in all experimental cases. Note that the performance gain on latency is noticeable in passive scanning, which needs a much longer waiting time than active scanning to listen to periodic beacon frames.

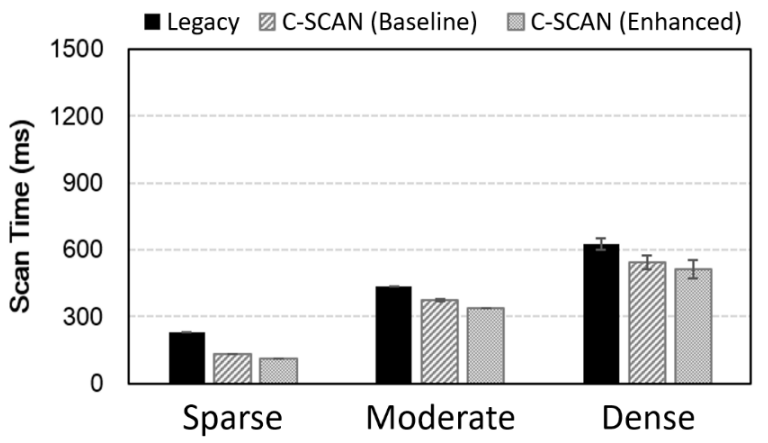


Figure 3.25: Detection latency of active Wi-Fi scanning.

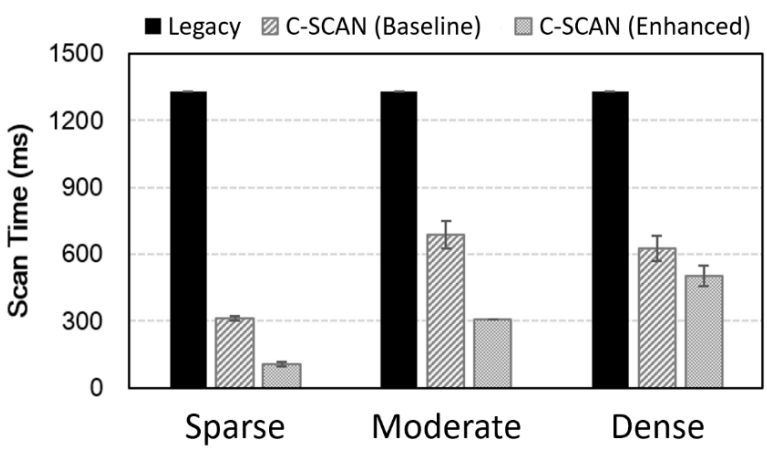


Figure 3.26: Detection latency of passive Wi-Fi scanning.

We investigate how the enhanced algorithm improves performance in terms of latency compared to the baseline algorithm. Fig. 3.27 depicts the cumulative distribution of the detection latency difference of the baseline and enhanced algorithm for different levels of channel occupancy. The latency gap implies the reduced detection time by adopting the enhanced algorithm instead of the baseline algorithm. The negative gap suggests that the detection time of the enhanced algorithm is faster than that of baseline algorithm. Our results indicate that the Wi-Fi discovery of the enhanced algorithm finishes more quickly than that of baseline algorithm in all environments under various channel conditions. This is because the enhanced algorithm operates RSSI sampling on multiple WPAN channels and determines the presence of Wi-Fi signal for multiple Wi-Fi channels. Specially, in sparse environment, the enhanced algorithm is 2.3 times faster than the baseline algorithm (baseline: 3.53 s and enhanced: 1.54 s).

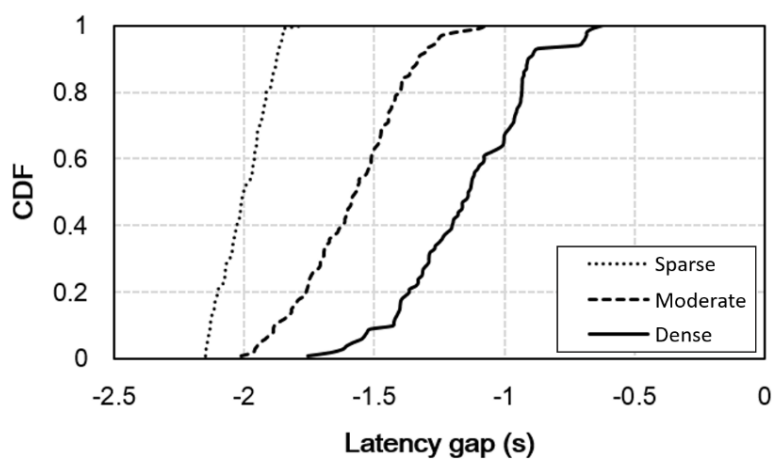


Figure 3.27: Comparison of detection latency at different levels of channel occupancy in real environments.

3.7.4 Energy Consumption

To evaluate the energy efficiency of C-SCAN compared with legacy, we conduct experiments in multiple places, including sparse, moderate, and dense environments. We use a Monsoon power monitor and its PowerTool software [92] to measure the energy consumption during AP scanning of the Wi-Fi interface. It records the energy consumption every 200 ms. Fig. 3.28 shows the setup of our power measurement experiments. The battery of our test device is non-removable and is integrated with a battery control chipset. To supply power with the power monitor, it needs a physical modification (detaching a battery control chipset from battery). After modification, the test device is connected to the power monitor through the chipset.

In the experiments, we restrict our device with the following settings to maintain reasonable experimental results. We set the backlight level of the phone screen to its

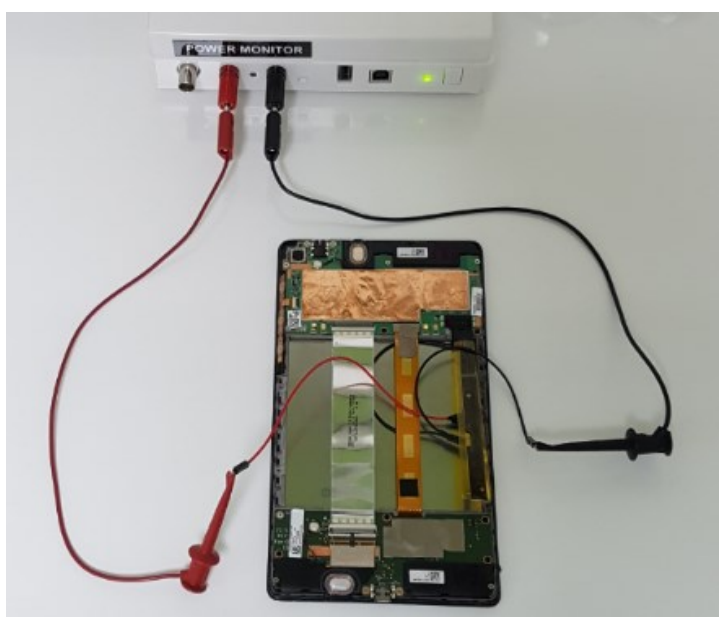


Figure 3.28: Setup of energy measurements using the Nexus 7 device.

lowest possible value and close all unnecessary running apps and services. In addition, we also disable all radio and other interfaces (e.g., NFC, GPS, and Sync) except for Wi-Fi.

We first measure the base energy consumption for 300 s, when the device contains its newly implemented application with the aforementioned restrictions while the Wi-Fi interface is turned off. The average and standard deviation of power consumption are 8155.57 and 0.21 uAh, respectively. We conduct the measurement for each scan method (C-SCAN and legacy) in various settings for equal duration. The average energy consumption of each scan method (active Wi-Fi scanning) can be approximated by subtracting the average base consumption from the energy consumption.

Fig. 3.29 shows the average energy consumption for the Wi-Fi interface. We focus on the active Wi-Fi scanning. Legacy consumes the highest energy in all settings since it scans full channel sets regardless of the channel conditions. In comparison, the C-SCAN uses lower energy by dispensing with unnecessary scanning of AP-free Wi-Fi channels. Notably, the sparse environment presents a 26% (baseline) and 40%

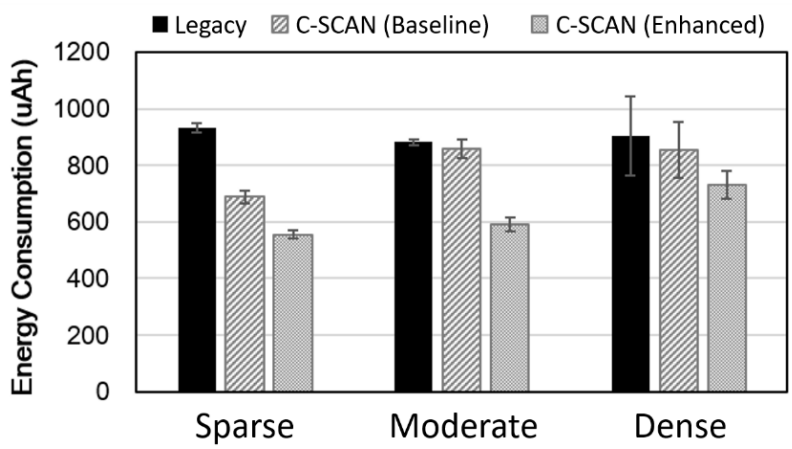


Figure 3.29: Energy consumption.

(enhanced) decrease in energy consumption for legacy despite of the FP cases. We observe that the average consumption of the enhanced algorithm is further reduced than that of baseline due to low FPs for the mentioned settings.

3.7.5 Throughput

To identify any throughput improvement of C-SCAN in comparison with legacy, we conduct experiments using iPerf3, configured for TCP, in a downlink scenario. We use a TP-Link AP, which supports 802.11 b/g/n in 2.4 GHz, as a WLAN network. The testing device is the only node associated with the AP and its distance from the AP is roughly 1 m. A Samsung laptop, which generates downlink traffic for TCP, is directly connected to the AP through a Gigabit Ethernet. Since the throughput can be affected by signal interference, we design a controlled environment where no signal interference exists (i.e., anechoic chamber). Each experiment is repeated 100 times with a measurement duration of 100 s.

Fig. 3.30 presents the throughput measurements of C-SCAN and legacy for different scan intervals. The latter is gradually increased from 1 to 10 s. As shown in the figure, both C-SCAN algorithms maintain high throughput even with aggressive scan interval. This indicates that C-SCAN has a minimal impact on the throughput since it prevents unnecessary probing. Notably, in the shortest scan interval, the performance improvement is the largest among the various scan intervals: 19.3 Mbps for legacy, 21.9 Mbps for baseline, and 22.4 Mbps for the enhanced algorithm on average. However, the improvement decreases as the scan interval increases. Even though the improvement for longer intervals is not higher than that for the shortest interval, C-SCAN still achieves a performance gain in the longer scan intervals. When the scan interval is set to 10 s, the throughput improvements for legacy are ~2.9%

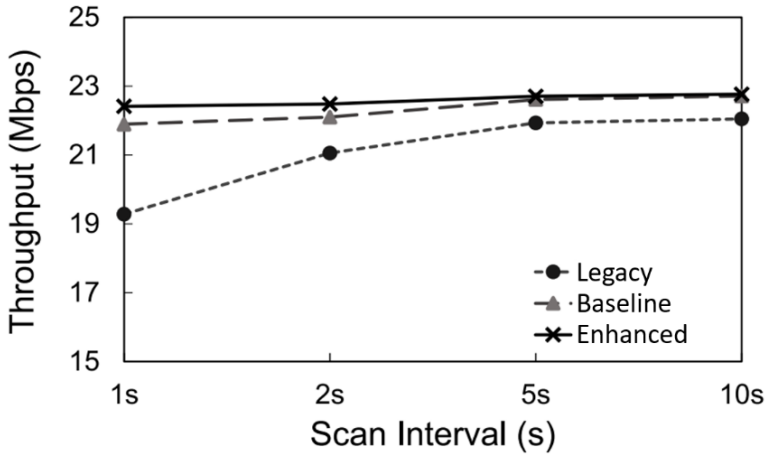


Figure 3.30: Throughput comparison: C-SCAN versus legacy.

(0.6 Mbps) and ~3.2% (0.7 Mbps) for the baseline and enhanced algorithm, respectively.

3.8 Summary

In this work, we presented C-SCAN, a novel and lightweight scheme for mobile devices, that utilizes coexisting low-power WPAN radios to identify available Wi-Fi channels. We implemented a prototype of C-SCAN using a Bluetooth-compliant wireless transceiver and demonstrated that C-SCAN achieves a high detection accuracy with only a short delay.

Chapter 4

D-SCAN: Toward Collaborative Multi-Radio Coexistence in Mobile Devices via Deep Learning

4.1 Introduction

Cross technology interference (CTI) in the unlicensed ISM bands has become a major issue as several different technologies, such as Wi-Fi, Bluetooth, and ZigBee, share the same frequency band. Many mechanisms have been proposed over the past several years to resolve the network-level external CTI issue. These include the detection of heterogeneous interferers and prevention of their collisions [19-23]. However, as the number of IoT devices exponentially increases and some IoT applications even rely on multiple heterogeneous radios in a single device, the internal CTI issue has also been aggravated.

To alleviate the problem of Wi-Fi and Bluetooth interfering with each other internally, TDM (Time-Division Multiplexing) based medium sharing approaches have been proposed [24-26]. That is, Wi-Fi and Bluetooth take turns to access the

2.4 GHz ISM band. The most representative case would be *combo-module* solutions which integrate Wi-Fi and Bluetooth interfaces into a single SoC (System on Chip) circuit. Most smartphones and many IoT devices (e.g., Samsung Galaxy, Apple iPhone, and Raspberry Pi) adopt these combo-modules due to their cost-effectiveness in terms of form factor. However, the inherent nature of antenna sharing causes severe performance degradation, which is reported repeatedly [93, 94] in common application scenarios where the two wireless interfaces are simultaneously used. The reason, which will further be elaborated, is mainly that the external and internal CTI mitigation schemes may inadvertently impede each other and lead to the failure of both.

In this work, we exploit the pros and cons of different radios in a collaborative manner if both Wi-Fi and Bluetooth are available. We leverage the fact that a Bluetooth radio in a combo-module can sense Wi-Fi signals as Bluetooth channels overlap with Wi-Fi ones in the 2.4 GHz unlicensed band. We infer several useful Wi-Fi information, which includes the identification of which Wi-Fi channels are used or not as well as the utilization of each Wi-Fi channel, efficiently with the low-power Bluetooth radio. Then, we facilitate the reliable coexistence of Wi-Fi and Bluetooth within the same device by coordinating the heterogeneous radios to effectively share the same antenna and spectrum.

There have been many attempts to utilize a low-power secondary radio to acquire Wi-Fi information. Wake-on-WLAN [17], S-WOW [27], and Esense [18] allowed cross-technology communications between ZigBee and Wi-Fi using special codes to exchange their channel information. Others have suggested Wi-Fi and Bluetooth combo APs embed the Wi-Fi channel information into Bluetooth broadcast packets [28, 29]. Zi-Fi [13], BlueScan [14], and C-SCAN [95] have suggested hands-off listening techniques to detect Wi-Fi APs. For example, Zi-Fi searches for beacon

frames by analyzing the periodicity of RSSI (Received Signal Strength Indicator) peaks. Choi *et al.* [52] and Jung *et al.* [95] have proposed lightweight algorithms to detect Wi-Fi signals via the correlation analysis between RSSI samples. These prior studies made contributions to collecting the detailed information about Wi-Fi signals; however, a few drawbacks still remain. The need for broad applicability demands a mechanism that (i) minimizes software or hardware modification, (ii) operates efficiently in terms of energy consumption, and (iii) adapts to dynamically changing environments.

To this end, we introduce a new data-driven approach, called D-SCAN, that models the unique temporal and spectral features of Wi-Fi signals from RSSIs measured by a Bluetooth radio via deep learning techniques. D-SCAN is widely applicable as it does not require hardware modifications, and is energy-efficient as the low-power Bluetooth radio offsets the overhead in Wi-Fi operations. Unlike prior heuristic approaches, it uses DNNs (Deep Neural Networks) to reliably infer Wi-Fi information even under network dynamics. However, it is not trivial to realize D-SCAN due to the following challenges. First, measuring the RSSIs of Wi-Fi signals with a Bluetooth radio involves a wide search space. In addition, it requires a vast number of training data that covers complex signal and interference patterns from the real world. Finally, D-SCAN should capture Wi-Fi signal patterns within fluctuating RSSI samples. Note that the problem becomes difficult especially as the density of APs increases.

The key ideas of D-SCAN that enable the accurate analysis across 13 Wi-Fi channels[5] are as follows. Normally, a large number of possible outcomes hinder the accurate classification by DNNs due to the high complexity. We overcome this

[5] Under the regulation by individual countries, the United States uses 11 Wi-Fi channels defined in the 2.4 GHz band, while most other countries use 13 Wi-Fi channels.

challenge using a divide-and-conquer approach. D-SCAN divides the 2.4 GHz spectrum into tractable narrow frequency bands to render training and decision-making manageable. Then, we extend two deep learning models to solve the scanning problem. We first apply the LSTM (Long Short-Term Memory) model to take sequential RSSI measurements of 20 MHz bands and to learn temporal features of Wi-Fi signals. However, LSTM exhibits limited performance for short and sparse Wi-Fi signals due to the small correlation between consecutive RSSI samples. We address this problem with a simple data representation method, termed *edge projection*, that accumulates the RSSI data for a target period of time. The edge projection preserves temporal and spectral correlations of Wi-Fi signals in 2-D matrices, and thus can leverage the CNN (Convolutional Neural Network) model effectively.

The main contributions are summarized as follows:

- We propose D-SCAN mechanism, which exploits a Bluetooth radio to efficiently obtain Wi-Fi channel information including Wi-Fi channel usage, signal strength, and channel utilization.
- We introduce a new 2-D RSSI data representation and effectively extend deep learning models to learn temporal and spectral features of Wi-Fi signals, achieving robust and adaptive estimation performance.
- We implement a prototype of D-SCAN for three use cases, where the obtained Wi-Fi information is used to improve the efficiency of both Wi-Fi scanning and Wi-Fi handover and to help Bluetooth coexist with Wi-Fi by promptly adapting to Wi-Fi interference.
- We generalize D-SCAN, which uses a low-power radio to learn the

distinguished features of wideband and narrowband signals, and demonstrates its utility in IoT devices over different frequency bands.

The rest of this chapter is organized as follows. Section 4.2 provides preliminaries on our research. In Section 4.3, we explain the overview and design of D-SCAN. In Section 4.4, we describe use cases of D-SCAN and their implementations to improve Wi-Fi and Bluetooth performance. Section 4.5 presents the evaluation results which emphasize the applicability of D-SCAN in real-world scenarios. In Section 4.7, we give a brief summary of this chapter.

4.2 Preliminaries

4.2.1 Wi-Fi Characteristics

To validate D-SCAN, we collected data about real Wi-Fi usage on various public locations such as parks, cafes, restaurants, and subway stations in Seoul, South Korea. Fig. 4.1 shows the ratio of the number of places where a given number of APs are observed to the total 168 visited places. Fig. 4.2 shows the ratio of the number of places where a given number of Wi-Fi channels are actively used to the number of total visited places. Here, we count the numbers of all the detected APs, fair APs with RSSIs above -80 dBm, and fine APs with RSSIs above -60 dBm, respectively, to differentiate the quality of Wi-Fi signals from the APs.

Many Wi-Fi APs, more than 35 in about 25% of the visited places, are deployed over multiple channels and form complex interference patterns in the 2.4 GHz band. However, when we only consider fine APs, which are useful in actual communications, the number of detected APs is significantly low. In terms of the number of used channels, four or fewer active channels are most commonly found.

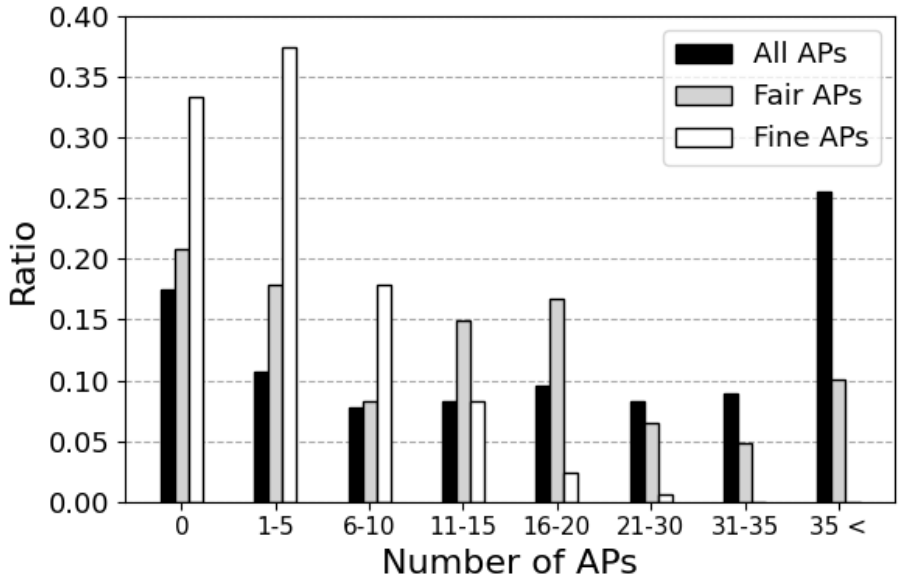


Figure 4.1: Detection ratio of adjacent APs.

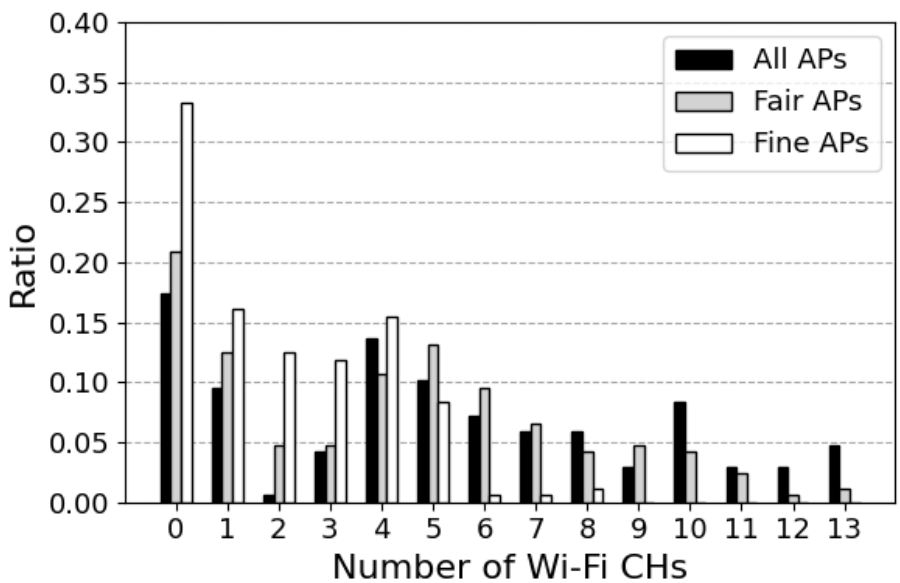


Figure 4.2: Detection ratio of active Wi-Fi channels.

If the Wi-Fi information, such as channel distribution of Wi-Fi APs, their signal strengths, and channel utilization, are obtained in advance by D-SCAN, Wi-Fi devices can reduce unnecessary scanning overhead by performing selective scans on channels that are confirmed to have active Wi-Fi APs. In addition, heterogeneous IoT devices or protocols that share the same frequency band can mitigate Wi-Fi interference by observing the Wi-Fi usage patterns.

4.2.2 Wi-Fi and Bluetooth Coexistence in a Combo-Module

Wi-Fi and Bluetooth in a combo-module share a single antenna as well as the ISM band in a TDM manner, generating new interference patterns that result in performance degradation. More specifically, both protocols fail to deal with nearby interferers and their collision rates mutually increase, when Wi-Fi and Bluetooth are used simultaneously.

Bluetooth uses an adaptive frequency hopping (AFH) strategy to avoid interference. Each pair of Bluetooth devices holds an AFH channel map that simply tells which of the channels are good to use or not based on their local channel assessments. However, the AFH, which requires rather complex channel assessments and master-slave handshaking, cannot appropriately update the AFH map while the radio is used by Wi-Fi. Then, Bluetooth fails to adapt to wireless communication dynamics and wastes a significant portion of radio occupancy upon collisions. Furthermore, impairing Wi-Fi frames by Bluetooth signals trigger “Avalanche effect” [93], where Wi-Fi transmission rates are lowered by the rate adaptation mechanism.

For example, even a user's watching video may require both Wi-Fi and Bluetooth; streaming down through Wi-Fi, and transmitting the audio to a headset via Bluetooth. Significant deterioration of video and audio quality in such use cases has been

reported [94] with no clear solution other than turning off Bluetooth. We believe that D-SCAN can mitigate this coexistence issue.

4.3 D-SCAN

4.3.1 Overview

D-SCAN aims to facilitate the collaborative coexistence of Wi-Fi and Bluetooth on a single device by harnessing the fact that a Bluetooth radio can measure the RSSI values of Wi-Fi signals over its Bluetooth channels, although it cannot decode the Wi-Fi signals. D-SCAN employs a data-driven approach that models the unique temporal and spectral features of Wi-Fi signals via deep learning techniques to extract useful Wi-Fi channel information such as Wi-Fi signal presence, signal strengths, and channel utilization.

With D-SCAN, we can improve the efficiency of Wi-Fi operations, such as Wi-Fi scanning and Wi-Fi handover, and also enable the synergy of multi-radio coexistence. For instance, the channel information obtained by D-SCAN is used by the Wi-Fi radio to perform actual Wi-Fi scanning only on the channels where the presence of APs is identified. It also helps the Wi-Fi radio (of the device) to associate with a better AP, by considering both Wi-Fi signal strengths and channel utilization data. Finally, D-SCAN, by informing Bluetooth of the used Wi-Fi channels and their utilization, helps to reduce collisions between two technologies. The use cases and their implementations are detailed in Section 4.4.

The overall flow of D-SCAN is illustrated in Fig. 4.3. It consists of four core components: Scheduler, Sampler, Data Formatter, and Channel Inspector. The Scheduler is invoked by the system or applications to reserve and initiate D-SCAN

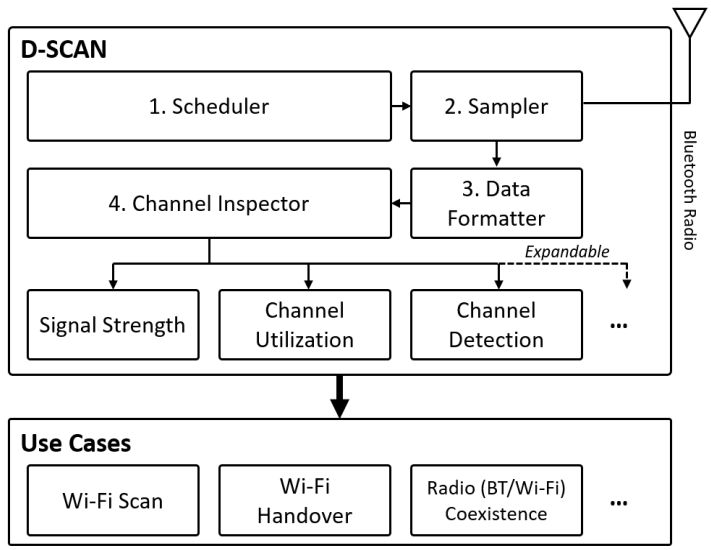


Figure 4.3: D-SCAN architecture.

operations. The Sampler measures RSSIs over the 2.4 GHz band using a Bluetooth radio. These raw RSSIs are continuously fed to the Data Formatter to pre-process the input to the neural networks. The Channel Inspector, which includes pre-trained LSTM or CNN models, analyzes the input (by running its neural networks) and infers Wi-Fi information.

4.3.2 Divide-and-Conquer Approach

To efficiently train a DNN for D-SCAN that encompasses various Wi-Fi environments, a large amount of training data and time are required. It involves numerous RSSI samples measured with Bluetooth radios to obtain comprehensive data across the whole Wi-Fi channels, i.e., 13 channels in our settings. To address this challenge, we characterize the waveforms of Wi-Fi signals on the 2.4 GHz band by collecting the RSSIs on the Bluetooth channels overlapping with the Wi-Fi ones. We can observe the recurring pattern of Wi-Fi signals on RSSI samples measured

from the 20 MHz range that corresponds to a Wi-Fi channel. For example, when we capture RSSIs of Wi-Fi signals when there are Wi-Fi APs on channels 1, 6, and 11, the signal characteristics in 2402–2422 MHz are similar to those in 2427–2447 MHz and 2452–2472 MHz (Fig. 4.4). This implies that the same method can be applied to infer the different Wi-Fi channels across the ISM frequency band.

This observation motivates us to employ a divide-and-conquer approach that simplifies the problem of training a DNN model for the whole Wi-Fi frequency band into smaller sub-problems of modeling a single Wi-Fi channel. In other words, we focus on the development of a deep learning model for a 20 MHz rather than the entire 84 MHz (2.4－2.4835 GHz) in the ISM band, and then apply the model multiple times to infer information on all the 13 Wi-Fi channels.

This divide-and-conquer approach has the following benefits: It lowers the complexity of the deep learning problem and exponentially reduces the amount of training data required. We can collect thirteen 20 MHz-wide training instances, which are partially overlapping but unique, from the 2.4 GHz band of each measurement site. We will show that the divide-and-conquer approach enables the trained neural network to achieve high accuracy under various scenarios including multiple AP signals.

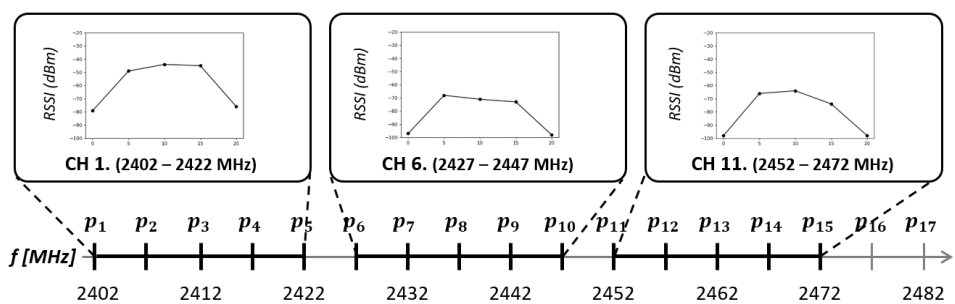


Figure 4.4: The recurring pattern of Wi-Fi signals on different channels.

4.3.3 Temporal and Spectral Features of Wi-Fi Signals

To effectively design deep learning models for obtaining Wi-Fi information, we have to understand the unique features of Wi-Fi signals distinguished from those of other protocols or noise. This brings us to focus on the temporal and spectral correlations between measured RSSIs during the sampling procedure (Fig. 4.5).

From a receiver's point of view, a valid Wi-Fi signal occurs at an arbitrary time and lasts for a certain duration (i.e., airtime, T). Then, the RSSIs measured consecutively for a period of time, T, at a frequency inside the 20 MHz range of the Wi-Fi signal have similar values and exhibit temporal correlation. The spectral feature, however, across the bandwidth of a Wi-Fi signal should also be employed to filter out the signals of other narrowband protocols; the channel widths of 1-2 MHz for Bluetooth and 2 MHz for ZigBee are relatively small compared to 20 MHz, that of Wi-Fi. The RSSIs of Wi-Fi signals, measured over the 20 MHz channel bandwidth for the period of time T, exhibit spectral correlation, showing a bell-shaped arrangement around the center frequency.

In D-SCAN, we reduce the search space from 84 to 17 frequencies, which we call *sampling points* (p_1, p_2, …, p_{17}), by spacing out the sampling frequencies apart by

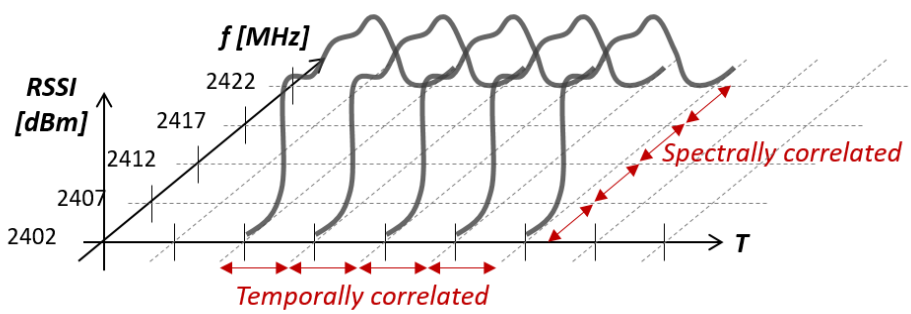


Figure 4.5: Temporal and spectral correlations in Wi-Fi signals on CH1.

5 MHz so they are aligned with center frequencies of Wi-Fi channels as in Fig. 4.4. This coarse-grained sampling strategy with 5 MHz space significantly reduces the sampling cost and the RSSI measurement delay while preserving both the temporal and spectral features of Wi-Fi signals. It can also filter out the narrowband signals as they cannot affect two or more sampling points at the same time. Note that using a fine-grained sampling, e.g., spaced 1 MHz apart instead of 5 MHz, in a limited amount of time may lead to partial data redundancy in the spectrum domain and/or incomplete spectrum characterization in the time domain. If the temporal and spectral features are well preserved in sampled data regardless of fluctuations in RSSI values, DNNs of D-SCAN can effectively capture specific patterns and adapt to network dynamics.

Based on the insights above, we design an RSSI sampling strategy for D-SCAN as follows. D-SCAN collects RSSI values by hopping five sampling points in a cyclical manner (say, from p_3 to p_7 and starting over from p_3 again as in Fig. 4.6) every τ. Here, τ is the sampling period, or slot time, which is ~160 us in the implementation using the CC2400 chipset. Once sampling begins, it lasts for a beacon interval, 102.4 ms, to ensure the beacon detection [96]. Wi-Fi APs can be sensed even without data exchange between APs and clients, as their beacon frames are periodically broadcasted. The upshot of this sampling domain and principle forms a solid foundation for DNN training and decision-making.

4.3.4 Sweeping the Entire 2.4 GHz Band

Legacy Wi-Fi scanning algorithms embed inefficiency by nature due to lack of prior knowledge on the APs in the vicinity, and thus perform a full-channel search on 13 channels. As a result, legacy Wi-Fi scanning takes about 1,300 ms for passive

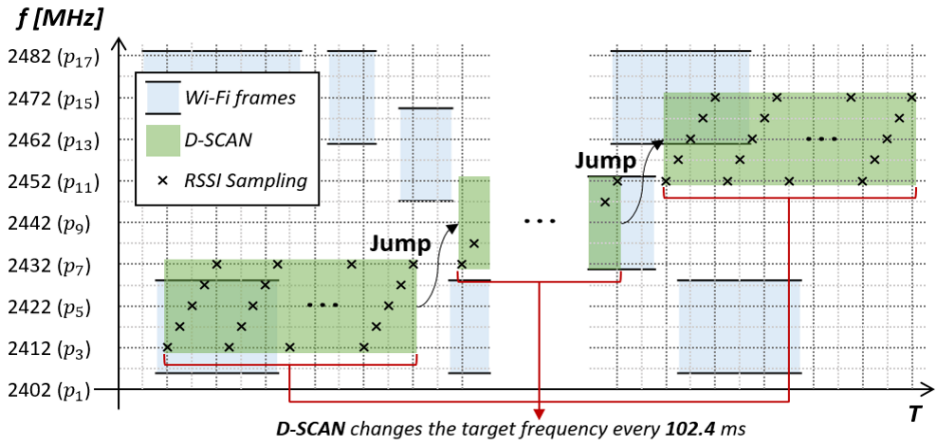


Figure 4.6: Three sampling ranges of D-SCAN to cover the 2.4 GHz band.

scanning and about 520 ms for active scanning in the 2.4 GHz band [97].

As explained earlier, D-SCAN employs a divide-and-conquer approach and uses the pre-trained deep learning models to infer the entire Wi-Fi channel information. The models are designed to take input from five sampling points to infer the Wi-Fi channel information over a 20 MHz range.

Here, it is worth noting that, in order to obtain information on all 13 Wi-Fi channels, D-SCAN only needs to be executed three times, not thirteen times (Fig. 4.6), unlike the legacy Wi-Fi scanning algorithms. This is because D-SCAN targeted on a Wi-Fi channel can infer Wi-Fi information of five Wi-Fi channels at once; one for the target channel and four for the adjacent overlapping channels. For example, by observing the five sampling points (p_3, p_4, p_5, p_6, p_7) targeted on Wi-Fi channel 3, we can also infer Wi-Fi information of channels 1, 2, 4, and 5. Out of those five sampling points, Wi-Fi signals on channel 1 affect the RSSI values of three sampling points, (p_3, p_4, p_5), Wi-Fi signals on channel 2 affect the RSSI values of four sampling points, (p_3, p_4, p_5, p_6), and so on. Thus, D-SCAN can analyze these neighboring Wi-Fi channels, albeit with fewer hints in the input data.

In turn, we set the three sampling ranges, 2412–2432, 2432–2452, and 2452–2472 MHz and obtain information of Wi-Fi channels {1, 2, 3, 4, 5}, {5, 6, 7, 8, 9}, and {9, 10, 11, 12, 13}, respectively. Channels 5 and 9 benefit from additional verification through redundant analysis. To sum up, D-SCAN is highly efficient in terms of both latency and energy, as the scan over the entire 2.4 GHz spectrum only lasts ~300 ms using a low-power Bluetooth radio. Still, note that training dataset consists of RSSI measurements from all 13 Wi-Fi channels.

4.3.5 Deep Learning Models

We introduce extended designs of two renowned deep learning models, LSTM and CNN, for D-SCAN. The inputs are from five sampling points of 20 MHz range but the format varies, as each model is designed for its own specialized purpose. Instead, we set the output layer size of both neural networks to 5 to infer five-channel information by taking the input from 20 MHz range. Here, the simultaneous analysis of multiple Wi-Fi channels barely affects the prediction accuracy in our experiments.

1) Time Series RSSI Measurements for LSTM: LSTM networks are designed to find patterns in long time-series data [98-100]. Therefore, the ability to memorize characteristic Wi-Fi fragments in an RSSI sequence is important in the learning process. As an input to the LSTM model, we collect RSSI data for a beacon interval, 102.4 ms, sequentially from five sampling points using a sliding window of window size 5, as in Fig. 4.7. The resulting matrix of 5x640 is used for the input. The RSSIs are normalized with minimum and maximum boundaries of -100 dBm and -20 dBm before we feed the data to LSTM. Generally, APs of RSSIs less than -90 dBm are considered unusable, and greater than -30 dBm are considered prodigious.

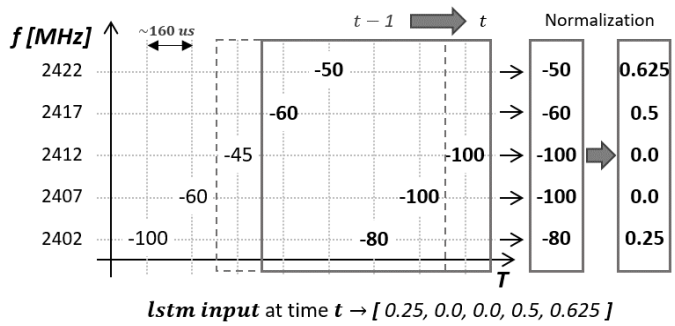


Figure 4.7: Sequential RSSI measurements for LSTM.

2) LSTM Architecture: The LSTM model of D-SCAN is designed to have 5 stages. Table 4.1 lists the parameters of our model. It has two LSTM layers with 256 hidden units, followed by two fully connected (FC) layers of size 32 and 5, respectively. We apply batch normalization (BN) and Rectified Linear Unit (ReLU) activation between the fully connected layers for standardized non-linear transformation on each element. In the last layer, a sigmoid activation function is applied to infer the Wi-Fi signal strength or channel utilization with a value between 0 and 1.

The LSTM model receives the RSSI sequence from five sampling points as input. The first two LSTM layers find characteristic Wi-Fi fragments and learn long-term relationships between time series data. Two fully connected layers learn the nonlinear complex relationship between the hidden unit and the final output. We expected the LSTM model to infer the Wi-Fi signal strength by recognizing the shape of the encoded Wi-Fi signal, or to infer the channel utilization by tracing the history of signal discovery.

In our experiments, the LSTM-based D-SCAN provides decent estimations on Wi-Fi channel information overall, but its performance is surpassed by that of CNN. The reason is that the short and sporadic Wi-Fi signals leave only a few temporal patterns in RSSI samples for LSTM to capture.

Table 4.1: RNN-LSTM architecture.

Stage	***Layer Description***	***Shape***
	Sequence input w/ 5 dim	5
1	LSTM with 256 hidden units	256
2	LSTM with 256 hidden units	256
3	FC 32	32
	BN + ReLU	32
4	FC 5	5
5	Sigmoid	5
	Prediction output	5

3) Edges and Projections for CNN: The original purpose of CNN is to analyze multi-dimensional images with spatial features [101-103]. We hypothesize that encoding temporal and spectral correlations between highly fluctuating (and, in some cases, seemingly random) RSSI samples as an image would form a solid pattern and allow us to discern meaningful Wi-Fi signals from noise. However, converting sequential 1-D RSSI data into discernable images poses a challenge.

We address this problem by developing a novel data representation, termed *edge projection*, where the focus is on the edges between two adjacent sampling points rather than the points themselves. There are 16 different edges (e_1, e_2, …, e_{16}) between 17 sampling points. An *edge* sample consists of two RSSI values measured successively from two adjacent sampling points. Accumulating all edge samples obtained during the sampling period into 2-D planes results in 16 edge projections (ep_1, ep_2, …, ep_{16}). More specifically, one can plot a point (corresponding to an edge sample) on a plane by using the RSSI value ($rssi_{lf}$) from a relatively low-

frequency sampling point as the x coordinate, and the RSSI value ($rssi_{hf}$) from a relatively high-frequency sampling point as the y coordinate. Here, we bin RSSI values between -100 and -20 dBm. Therefore, an edge projection is represented by an 80-by-80 matrix, whose elements are the count of edge occurrences.

Fig. 4.8 depicts an example of edge projections when two APs reside in different channels, 6 and 7. An edge, e_6, is constructed with two RSSI values from p_6 and p_7, which are measured consecutively and projected onto ep_6. The $rssi_{p_6}$ from the relatively low-frequency, p_6, and $rssi_{p_7}$ from the relatively high-frequency, p_7, are respectively used as the x and y coordinates of e_6 in ep_6. In this way, the aforementioned temporal and spectral correlations are preserved; two consecutively measured RSSI samples are integrated into a single point and their relative and absolute magnitudes are encoded as a position on a 2-D plane. The edge projection embodies the likelihood and consistency of a Wi-Fi signal by recording the number of signal occurrences onto a matrix corresponding to the edge samples' coordinates. In Fig. 4.8, the edges being accumulated are expressed as the colored points which become darker as the points are repeatedly plotted at the same or adjacent locations.

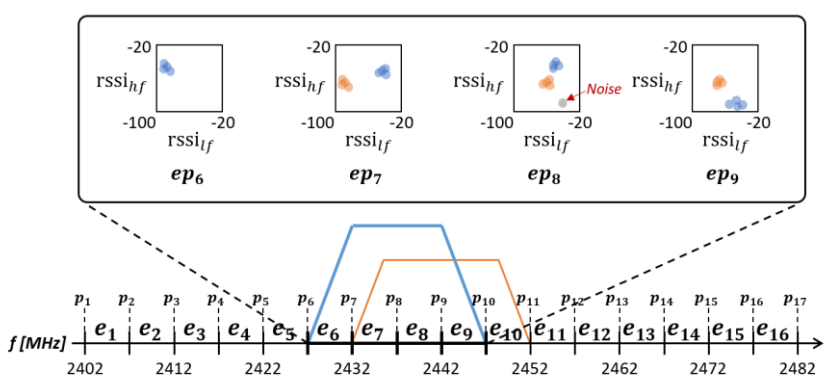


Figure 4.8: Unique cluster pattern projected by Wi-Fi signals on corresponding four edge projections.

Wi-Fi signals leave distinctive patterns in the edge projections. As shown in Fig. 4.8, when the signals from an AP in channel 6 are projected as edges, the weights are accumulated on ep_6, ep_7, ep_8, and ep_9. In an edge projection, edges from the same AP are plotted close to each other to form a unique cluster pattern, and the combination of these clusters across multiple edge projections characterizes that AP. Here, the distance of the clusters to the origin (-100, -100) represents the signal strength: the farther the distance, the stronger the Wi-Fi signals. Furthermore, the weights accumulated by edges imply the degree of medium utilization in the frequency range of that edge projection. Signals from protocols other than Wi-Fi would only affect a single edge projection, and noise would not form a cluster.

4) CNN Architecture: Depending on the device, Wi-Fi signals show a variety of similar but different bell-shaped spectral patterns and CNN is specialized to distinguish them effectively. Our CNN consists of 7 stages; four convolutional layers, two fully connected layers, and the output layer (Table 4.2). We apply BN and ReLU activation between the convolutional and fully connected layers. The 2x2 average pooling filters that follow the convolutional layers are applied with a stride of 2 to downsample the feature maps.

The four edge projections of shape 4x80x80 are provided as input to the CNN, with the second and fourth edge projections transposed to facilitate cluster combination search. The first two convolutional layers are designed to spot clusters in individual edge projections. The convolutional layers on stage 3 are designed to detect cluster combinations across edge projections by applying vertically and horizontally long filters. The last convolutional layer finds meaningful combinations among them. The following fully connected layers learn relevance with the output layer to approximate the Wi-Fi channel information. We expected CNN to infer (i) the strongest Wi-Fi RSSI by recognizing the outer-most cluster combinations of a

Table 4.2: CNN architecture.

Stage	***Layer Description***	***Shape***
	Four-edge projection	4x80x80
1	Conv 3x3	16x80x80
	BN + ReLU + AvgPool(2,2)	16x40x40
2	Conv 3x3	16x40x40
	BN + ReLU + AvgPool(2,2)	32x20x20
3-1	Conv 3x11	128x20x20
3-2	Conv 11x3	128x20x20
	BN + ReLU + AvgPool(2,2)	256x10x10
4	Conv 3x3	256x10x10
	BN + ReLU + AvgPool(2,2)	256x5x5
5	FC 128	128
	BN + ReLU	128
6	FC 5	5
7	Sigmoid	5
	Prediction output	5

valid Wi-Fi signal and (ii) the channel utilization by observing the accumulated edge occurrences.

5) Hyperparameter Tuning: In applying neural networks, there are no strict rules for selecting network parameters. Therefore, we experimentally searched for the parameters that optimize the performance and remove unnecessary computations by adjusting network widths and depths [107, 108]. In fully connected layers, we set the dropout rate to 0.5 to prevent overfitting and generalize the networks. Finally, the

weight of networks was trained using the Adam optimizer with a learning rate of 2e-4. The prediction error through backpropagation was minimized by using SSE (Sum of Squared Error) to satisfy the following objective function;

$$\min_{\theta} Loss_{SSE}(\hat{y}, y)$$

$$Loss_{SSE}(\hat{y}, y) = \sum_{i} (\hat{y}[i] - y[i])^2$$

$\hat{y}$ is the set of predicted output on i-th channel while y denotes the set of i-th channel's actual value. We implemented our deep learning architectures using Pytorch on a system with four NVIDIA CUDA-enabled Titan GPUs.

D-SCAN is lightweight when it comes to the cost of running the DNNs. The CNN-based design has 1.7M network parameters and requires 0.18B FLOPs (Floating-Point Operations) for each runtime. This is a small number compared to the renowned image analysis CNNs. The exact time and energy required for D-SCAN depend on hardware specifications such as memory and processor. For each prediction on Wi-Fi information, our GPU system spends only 1.2 ms. D-SCAN is also applicable to low-spec devices, such as Raspberry Pi 3, and it takes 16 ms. The LSTM-based design has fewer parameters than CNN. However, the CNN-based D-SCAN is faster by its design, since data in CNN can be processed in parallel, while data in LSTM has to be processed sequentially.

4.4 D-SCAN Use Cases

In this section, we present three use cases where D-SCAN overcomes the limitations caused by Wi-Fi and Bluetooth coexistence on a single device. Specifically, we demonstrate the utility of D-SCAN in providing high performance in Wi-Fi operations and Bluetooth transmissions.

4.4.1 Wi-Fi Scanning

Passive Scan: With passive Wi-Fi scanning, a device listens for beacon frames periodically broadcasted by APs in the vicinity on all channels. This method consumes a lot of time and energy, as the device has to stay on each channel at least for a beacon interval to ensure that all APs are detected. Many Wi-Fi channels, however, are often vacant as we observed in Fig. 4.1 and 4.2, and it is wasteful to operate an expensive Wi-Fi radio on channels where no AP exists.

D-SCAN, to suppress unnecessary Wi-Fi scanning on AP-free channels, triggers *selective* passive scan. The information regarding AP presence on each Wi-Fi channel is acquired by D-SCAN in a fast and efficient manner thanks to the low-cost Bluetooth spectrum scanning.

Active Scan: With active Wi-Fi scanning, a device actively broadcasts probe requests on all channels to collect probe responses from nearby APs. This method shows much shorter latency compared to passive scanning as the device does not need to wait for beacon frames. Active scans, however, may substantially deteriorate the QoS (Quality of Service) of Wi-Fi networks by making channels overloaded with probe messages [60, 61, 106, 107]. Moreover, active scans, which are periodically invoked from client devices to ensure reliable wireless connectivity, increase Wi-Fi latency.

D-SCAN, to reduce spectrum wastage and to provide shorter active scanning delay, limits the number of probe messages. D-SCAN utilizes the Wi-Fi signal strength and channel utilization to rank Wi-Fi channels and selectively scans on a few of them, e.g., top-3 channels, which are considered to have the most suitable APs to associate with. The assessment of Wi-Fi channels is based on the achievable throughput described in Section 4.4.2.

D-SCAN is invoked right before the scheduled Wi-Fi scans, both passive and active, to obtain the Wi-Fi information in advance. Then, the reduced use of Wi-Fi radio thanks to the selective scan greatly reduces scanning latency and energy consumption.

4.4.2 Wi-Fi Handover

Wi-Fi handover is the process of a client device disconnecting from the currently associated AP and re-associating with another AP. In Wi-Fi, the handover mechanism primarily depends on signal strengths. However, an AP with a higher RSSI value does not always provide higher bandwidth when its channel is saturated with traffics from connected devices. Therefore, channel utilization must be considered together. We compute the achievable throughput (AT) as in BLEND [28]. Although BLEND required Wi-Fi/Bluetooth combo AP to broadcast channel utilization in the Bluetooth advertising packets, D-SCAN can acquire this information by simple and short sweeps over the spectrum.

$$AT = PR \cdot (1 - CU)$$

PR is the achievable PHY rate and is approximated via measured RSSI. Table 4.3 shows the PHY rates of 802.11n Wi-Fi and their Rx sensitivities, which are the minimum RSSIs that satisfy 90% packet delivery ratio (PDR). We consider the PR to be the maximum PHY rate that satisfies 90% of PDR given estimated RSSI. Then,

Table 4.3: PHY rate and Rx sensitivity (802.11n).

PHY rate (Mbps)	6.5	13	19.5	26	39	52	58.5	65
Rx sensitivity (dBm)	-94	-91.7	-89.2	-86.1	-82.5	-77.9	-76.3	-74.7

we take into account the idle channel ratio, which is the opposite concept of channel utilization (CU), denoted as $(1 - CU)$. The achievable throughput is computed by multiplying the PR by the idle channel ratio. When a device is in the disconnected state, D-SCAN connects it to the AP with the largest AT. During Wi-Fi handover where a device is already connected to an AP, D-SCAN re-associates it to the AP with AT greater than AT of the current AP by Δ.

$$AT_{current} + \Delta < AT_{target}$$

$$\Delta = (1 - CU_{current}) \cdot \delta$$

The threshold, Δ, is to prevent the ping-pong effect, where handover repeatedly occurs between two APs with similar ATs. The value of δ is set to be 6.5 (Mbps) in 802.11n scenarios, as it is the minimum throughput difference between the PHY rates. It is also possible to employ different PHY rate values depending on the embedded Wi-Fi chipset.

4.4.3 Synergy for Wi-Fi and Bluetooth Coexistence

Most modern devices equipped with combo-modules experience huge performance degradation in a congested environment where Wi-Fi and Bluetooth are used together as mentioned in Section 4.2.2. The main reason is that the increased number of retransmission attempts of Bluetooth radio due to CTI can reduce the time that Wi-Fi radio uses the shared 2.4 GHz antennas.

Through spectrum analysis, D-SCAN acquires Wi-Fi channel information, a major factor impeding Bluetooth communication. So we use D-SCAN to promptly update the AFH map and help Bluetooth avoid collisions with the Wi-Fi signals. As shown in Fig. 4.9, the collision probabilities of each Bluetooth channel can be estimated by the utilization information of corresponding Wi-Fi channels. Then, D-SCAN

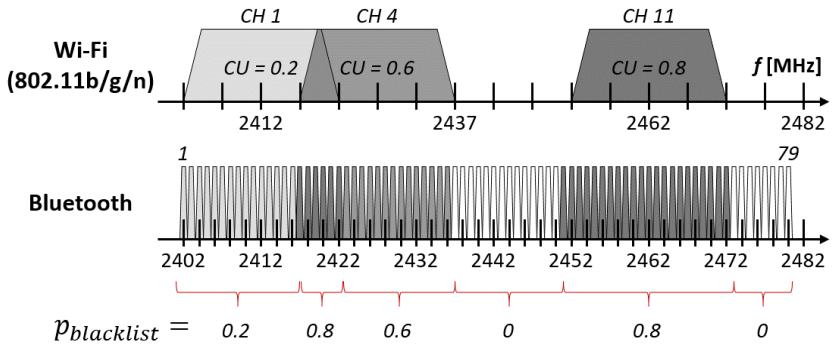


Figure 4.9: Probabilities of Bluetooth channels to be blacklisted ($p_{blacklist}$) by D-SCAN based on Wi-Fi channel usage.

blacklists each Bluetooth channel with the collision probability, dubbed $p_{blacklist}$. This strategy can be easily applied by using an existing command implemented in Bluetooth's host controller interface (HCI) [24]. We invoke D-SCAN to update the AFH map every 5 seconds in scenarios where Wi-Fi and Bluetooth are used simultaneously.

4.5 Performance Evaluations

We demonstrate the accuracy of D-SCAN's prediction of signal strength and channel utilization. We then establish the performance gain of D-SCAN on three practical use cases; Wi-Fi scan, handover, and Bluetooth coexistence. Unless otherwise stated, the results are of CNN-based design, as it performed better than LSTM in many experimental scenarios. D-SCAN is implemented on an Ubuntu 18.04 laptop equipped with Qualcomm Atheros AR5B22 chipset. We modified the ath9k device driver of backports 5.6.8-1 [108] to operate D-SCAN. Ubertooth [53], an open-source Bluetooth platform equipped with a CC2400 transceiver, is attached to the laptop through a USB port for RSSI sampling via Bluetooth radio.

4.5.1 Training and Testing Data Collection

To demonstrate the feasibility of D-SCAN, we collected data and conducted experiments from real-world environments. We visited various places throughout Seoul to collect diverse data. To be specific, the data is from 168 different spots of over 50 measurement sites, including cafes, restaurants, malls, parks, university buildings, subway stations, apartments, etc. To enrich the training data for each measurement site, we collected time-series RSSI from 5 sampling points for 100 ms on 13 different target frequency ranges that cover the whole 2.4 GHz spectrum. We labeled the samples with ground-truth Wi-Fi scan results, which include the list of APs residing in each channel and their RSSIs. In addition, we appended the channel utilization measured by multiple synchronized Ubertooth devices. The RSSI and channel utilization labels were normalized to values between 0 and 1. The size of the dataset reached about 100,000, and we performed 5-fold cross-validation for parameter tuning and performance evaluations. To further refine D-SCAN as a realistic solution, *federated learning* techniques, where a central server and edge devices collaborate to effectively form a rich dataset and train models, can be applied in the future.

4.5.2 Signal Strength and Channel Utilization

We evaluate the accuracy of D-SCAN in estimating strongest signal strength (*SS*) and channel utilization (*CU*) in Fig. 4.10 and 4.11. We use 20-by-20 normalized confusion matrices to visually represent how close the predicted values are to the real ones. D-SCAN accurately predicts the absence of a Wi-Fi signal. Weak signals with RSSI less than 0.25 (-80 dBm) are considered absent. The SS prediction works

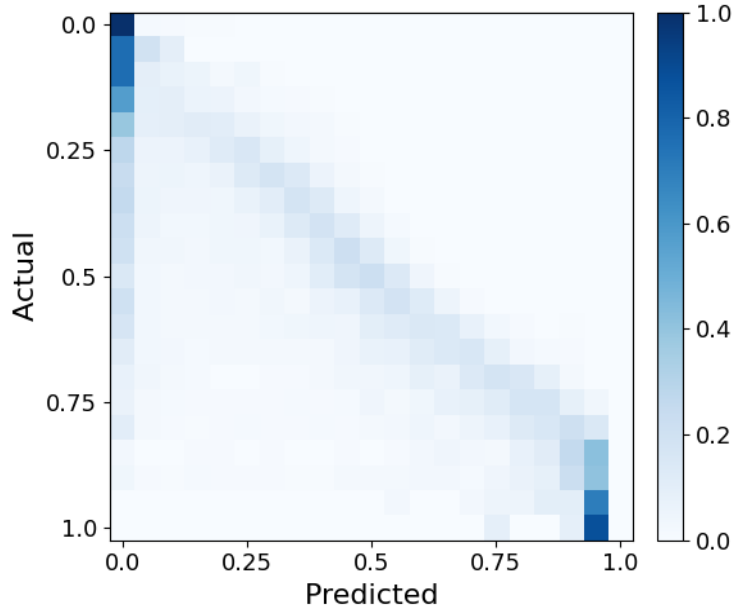


Figure 4.10: Normalized confusion matrices on signal strength estimation.

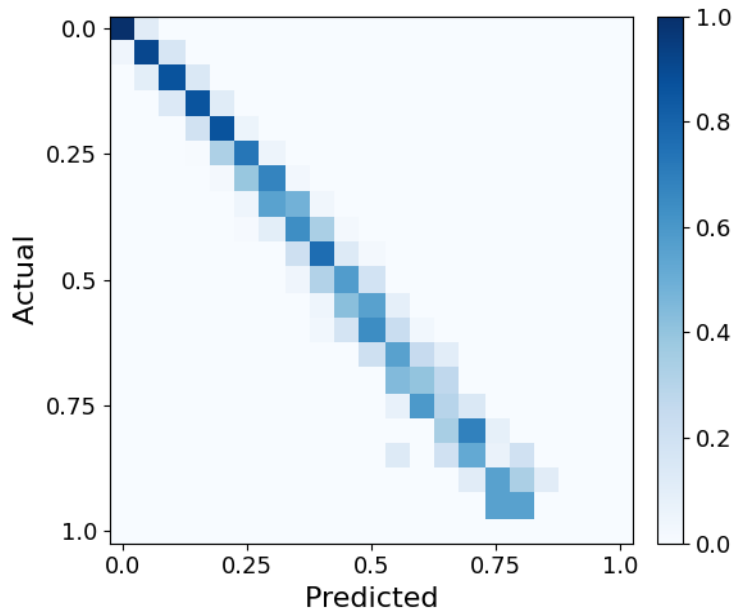


Figure 4.11: Normalized confusion matrices on channel utilization estimation.

well when the Wi-Fi signal is very strong around 1 (-20 dBm). Predictions of moderate signal strength are slightly deviated, but remained valid with the average loss of ~0.07 (5.6 dBm). CU predictions clearly reveal diagonal on the confusion matrix, implying a high degree of accuracy. It shows loss values lower than 0.02 in most cases. Inferring the CU of a Wi-Fi channel is considered a relatively easy problem, as it is analogous to finding the proportion of RSSI samples affected by Wi-Fi signals out of the total samples. The overall performance of D-SCAN based on CNN and LSTM are recorded on Table 4.4.

4.5.3 Wi-Fi Scanning

We evaluate the performance of D-SCAN in determining the AP presence of each channel. If it accurately identifies the active Wi-Fi channels, the performance of Wi-Fi scanning in terms of latency, energy, and throughput can be improved. We use the full scan result of legacy Wi-Fi as a ground-truth evaluation criterion. We compare the performance with that of the latest C-SCAN method [95], which also detects Wi-Fi APs using a Bluetooth radio. C-SCAN determines the AP presence based on a heuristically designed likelihood score; it records the length and number of occurrence of wideband signals to determine whether they are Wi-Fi signals.

Table 4.4: Evaluation of signal strength and channel utilization predictions by CNN- and LSTM-based D-SCAN.

	CNN-SS	***LSTM-SS***	***CNN-CU***	***LSTM-CU***
RMSE	0.152	0.163	0.019	0.025
MAE	0.069	0.081	0.011	0.015

D-SCAN determines that an AP exists in the corresponding channel when the estimated Wi-Fi signal strength exceeds a certain threshold. We illustrate a ROC (Receiver Operating Characteristic) curve in Fig. 4.12 to find the threshold and to compare the diagnostic ability of each method. FPR (False Positive Rate) and TPR (True Positive Rate) are plotted as the discrimination thresholds are varied. A good classifier yields points closer to the upper left corner. D-SCAN (CNN) is the finest, followed by D-SCAN (LSTM) and C-SCAN, with area under the curves (AUC) of 0.9, 0.86, and 0.73. We then compare the accuracy after having each method adopt the threshold that exhibited the highest accuracy (Fig. 4.13); each model uses 0.025, 0.1, and 26. The overall accuracy of each model is 0.9, 0.84, and 0.77. We classify the real-world test data into *sparse*, *moderate*, and *dense* scenarios according to the number of active channels. The sparse, moderate, and dense scenarios contain APs in 0-4 channels, 5-9 channels, and 10-13 channels, respectively. All methods have difficulties in determining the AP presence as the density increases. Among them, D-SCAN (CNN) shows the highest accuracy regardless of channel density and it is about

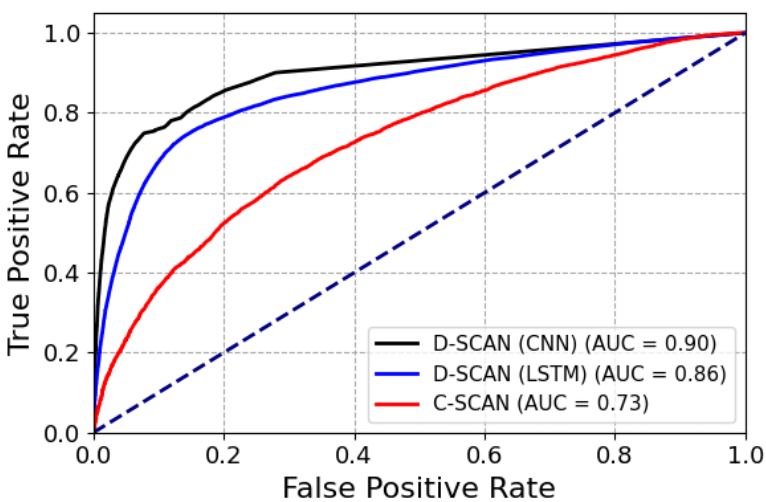


Figure 4.12: ROC curves on channel detection.

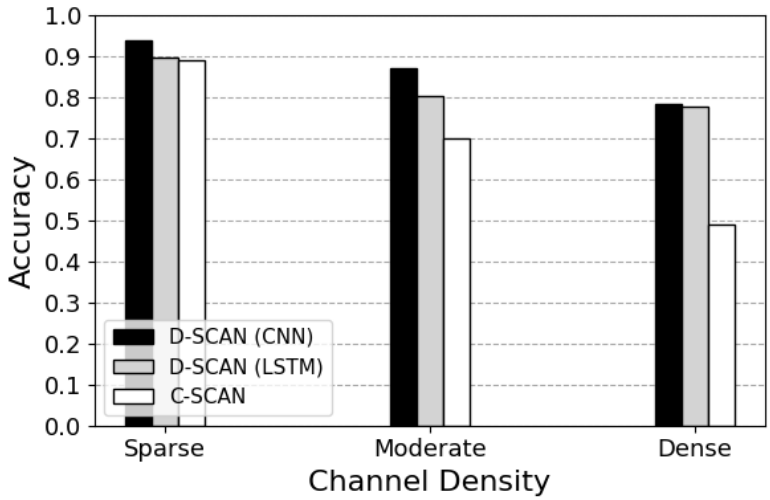


Figure 4.13: Channel detection accuracies.

40% higher than that of C-SCAN in the worst-case scenario. D-SCAN (LSTM) exhibits good resistance to changing densities as higher channel density provides more temporal patterns. C-SCAN, a fixed heuristic protocol, does not seem to adapt well to diverse signal patterns. D-SCAN (CNN) performs well, as it focuses more on spectral characteristics; the cluster shape in spatial coordinates facilitates the detection of Wi-Fi signals over noises.

We next observe scan latency and energy consumption in both passive and active cases (Fig. 4.14). In the passive scan experiment, D-SCAN takes ~300 ms to perform a full scan and triggers Wi-Fi to selectively scan on active Wi-Fi channels. D-SCAN detects an average of 7.2 active channels, resulting in total scan latency of about 1,000 ms, a 23% time-saving. In particular, the performance gain of D-SCAN in terms of energy is even more pronounced thanks to the employment of inexpensive Bluetooth instead of Wi-Fi. Bluetooth acquires channel information with small energy of about 20 mJ and reduces the energy consumption by about 45% compared to the legacy. In active scan, legacy Wi-Fi shows short scan latency with active probe

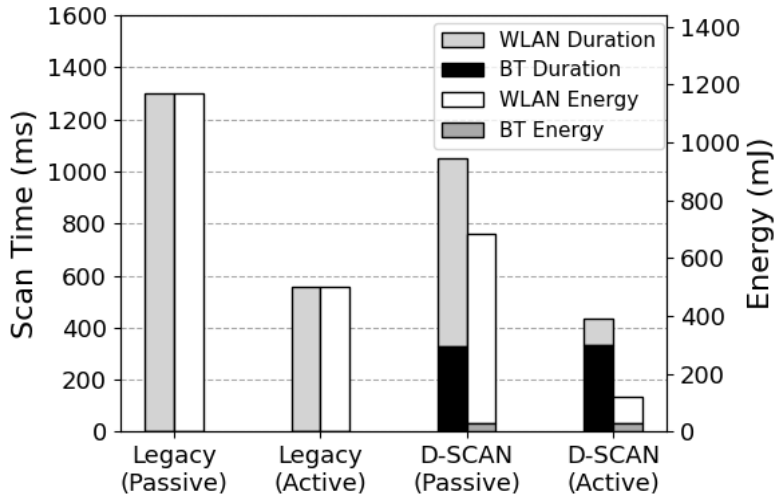


Figure 4.14: Wi-Fi scan latency and energy consumption.

exchanges. D-SCAN achieves lower scan latency by exchanging probe messages only on up to 3 channels. By further restricting the use of the Wi-Fi radio, it consumes 86% less energy.

Repetitive use of D-SCAN obtains fresh Wi-Fi channel information and contributes to sustained improvement of Wi-Fi connection. Since D-SCAN is lightweight in terms of latency and energy, these repeats of D-SCAN after the initial connection complements the periodic active scan. Fig. 4.15 presents the throughput measurements of D-SCAN and legacy using different intervals for active scans. As D-SCAN eliminates unnecessary overheads in scan operations, an adverse impact of periodic scan on Wi-Fi communication (i.e., throughput) is reduced. The shorter the scan interval, the greater the throughput gain.

Additional Experiments: We further investigated several prior selective Wi-Fi scanning methods, BLESS [29], BLEND [28], BlueScan [14], and C-SCAN [95], to evaluate the performance of D-SCAN on acquiring Wi-Fi information. These

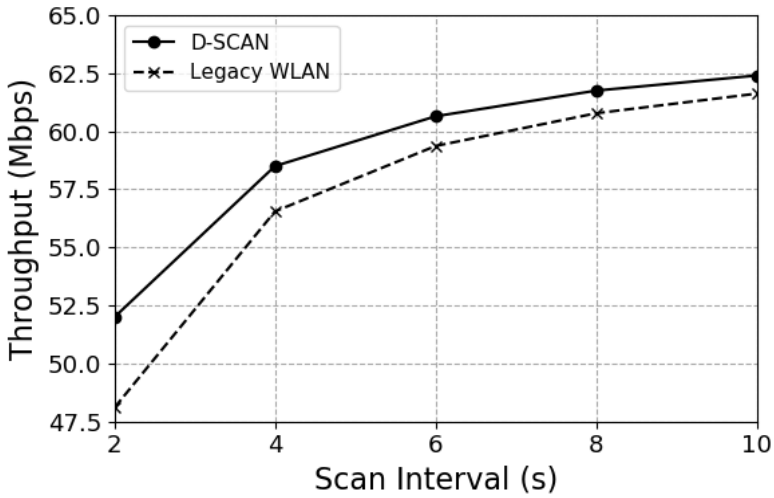


Figure 4.15: Throughput comparison.

methods commonly exploit Bluetooth radio to obtain Wi-Fi information and suppress unnecessary scanning attempts, offsetting the overhead in legacy Wi-Fi scanning.

BLESS and BLEND require Wi-Fi and Bluetooth combo functionality on both AP side and client side. The combo APs embed Wi-Fi beacon information into BLE (Bluetooth Low Energy) advertising packets, which are periodically broadcasted to nearby stations. A station directly acquires nearby Wi-Fi AP information by collecting BLE advertising packets for ~200 ms. On the other hand, BlueScan, C-SCAN, and D-SCAN infer Wi-Fi information from the RSSI traces which are sequentially measured with Bluetooth radio. For BlueScan, which attempts to spot beacon frames by observing periodicity in RSSI peaks, it takes more than 2.5 s to fully scan 13 Wi-Fi channels in the 2.4 GHz band. For C-SCAN and D-SCAN, as they analyze 3 and 5 Wi-Fi channels at once within ~100 ms, it takes ~500 and ~300 ms to scan 13 Wi-Fi channels, respectively.

Upon acquisition of Wi-Fi information, BLESS computes the optimal scanning sequence for beacon receptions and performs passive scans on the right channel at the right moment, otherwise, it stays in the idle mode. BLEND estimates the achievable throughput of each AP with signal strength and channel utilization, and performs an active scan on a single channel with the best AP. BlueScan, C-SCAN, and D-SCAN simply perform either passive or active scans on channels, which are identified to have active Wi-Fi APs. As an exception in active scan mode, D-SCAN only scans top-3 Wi-Fi channels judged to have good link quality, as it can also rank Wi-Fi channels by inferred signal strength and channel utilization (like BLEND).

Fig 4.16 and 4.17 depict the simulation results on scan latency and energy consumption, respectively, as the number of APs varies from 6 to 12. Here, the average number of occupied Wi-Fi channels is 4.96, 6.16, 7.16, and 8.03. In the experiment, as it takes a significant amount of time for BlueScan to obtain Wi-Fi information (more than x2.5 than legacy), we assume that BlueScan simply represents the selective passive scanning approaches and, like BLESS and BLEND, exploits the BLE advertising packets. Here, note that the scan time of BLESS depends on the number of APs while the scan time of BlueScan, C-SCAN, and passive D-SCAN depends on the number of occupied Wi-Fi channels. The scan time of BLEND and active D-SCAN, on the other hand, is constant as they scan on fixed number of channels.

Both passive (P) and active (A) scanning methods shorten the duration of Wi-Fi scanning, which in turn saves a significant amount of energy. The scan latency of BLESS, which is a passive scanning method, is very short just like other active (A) scanning methods, BLEND, C-SCAN (A), D-SCAN (A), and legacy (A), as it optimizes the scanning sequence from the beacon timing information. However, the

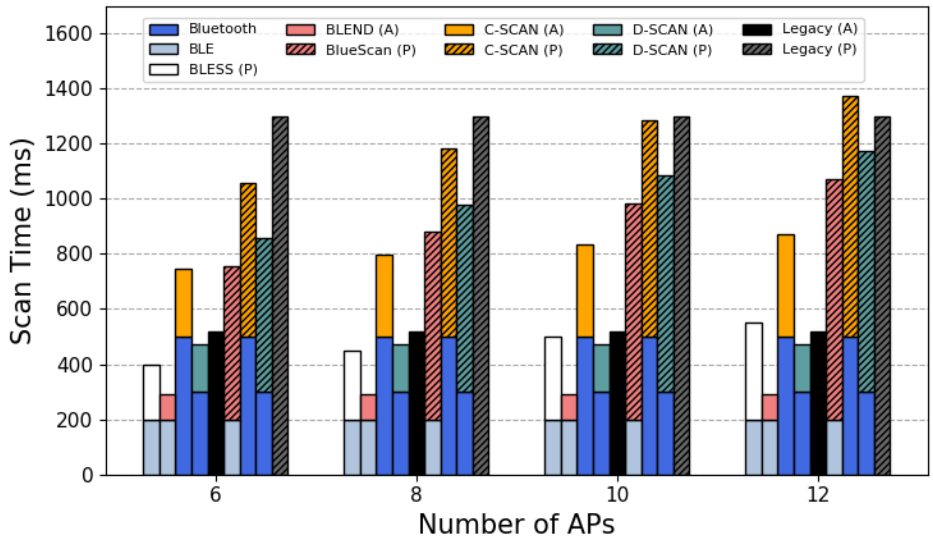


Figure 4.16: Experimental results of different Wi-Fi scan methods on scan latency.

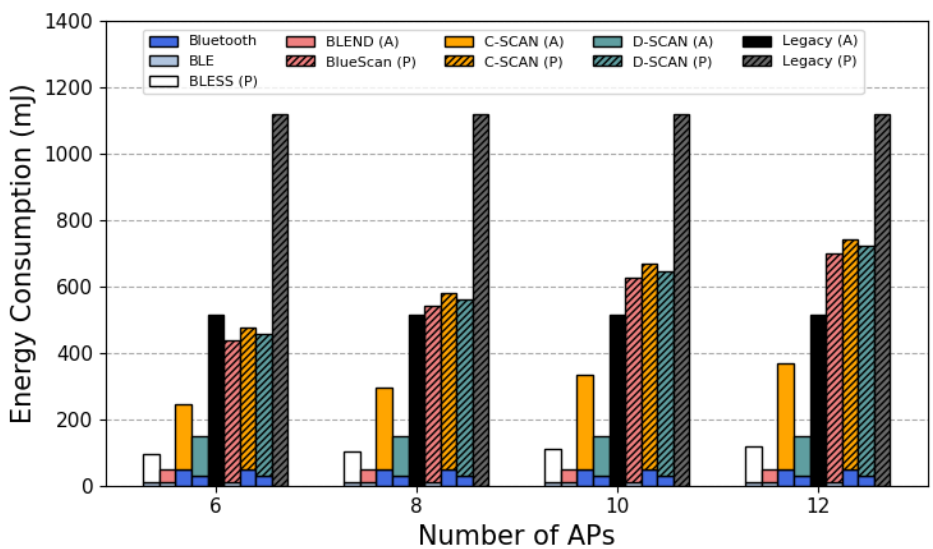


Figure 4.17: Experimental results of different Wi-Fi scan methods on energy.

scan latency of BLESS increases proportionally to the number of APs to discover all the APs. The scan latency of selective passive scanning methods, BlueScan (P), C-SCAN (P), and D-SCAN (P), depends on the number of occupied Wi-Fi channels, and it is shown that unnecessary scan operations on the empty Wi-Fi channels are effectively suppressed. In terms of energy, BLESS (P), BLEND (A), and BlueScan (P) take advantage of using an efficient BLE chipset. However, since C-SCAN and D-SCAN can also employ BLE radio, the performance gap can be overcome.

In short, each scanning method has pros and cons that should be further investigated in future works. The selective scanning methods that employ combo APs, such as BLESS and BLEND, are fast and energy-efficient. However, they are not practical as it is hard to widely deploy APs with combo functionality. C-SCAN and BlueScan require no additional infrastructure, but either the scanning takes too much time or the scan results are not accurate enough. In the meanwhile, D-SCAN stands as a practical and reliable solution, by taking the balance between the deployment cost and the performance.

4.5.4 Wi-Fi Handover

We examine the performance improvement by D-SCAN during Wi-Fi handover in a controlled moderate environment, where three APs are configured to use non-overlapping channels (Fig. 4.18). A target Wi-Fi device is moved from P1 to P3, and we observe the throughput changes as it initially connects to AP1 and re-associates to either AP2 or AP3 at P2. We adjust the transmit power and locations of AP2 and AP3 so that respective RSSIs of -50 dBm and -80 dBm are observed during the handover. In addition, we deploy an interferer that occupies the medium of AP2, which has a relatively better signal quality; interferer CUs of 0.1, 0.2, …, and, 0.9

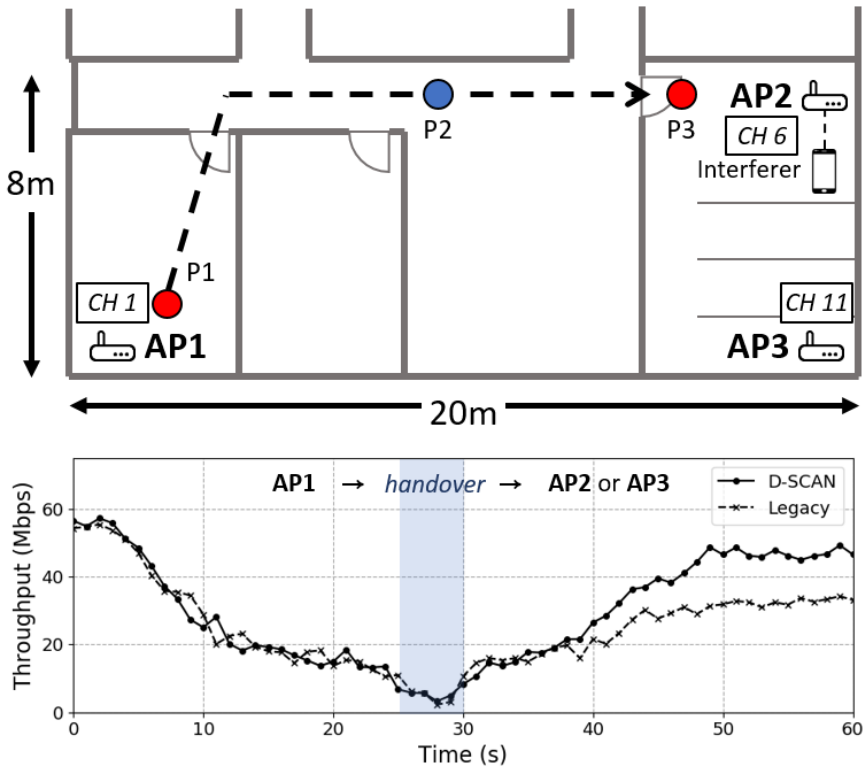


Figure 4.18: Wi-Fi handover by D-SCAN, taking both SS and CU into account.

are used for each scenario. Since the legacy Wi-Fi handover method selects an AP based on RSSI regardless of channel conditions, the presence of the interferer does not affect the selection. As a result, it always chooses AP2 and the throughput is limited to 32.7 Mbps on average. D-SCAN selects AP2 when the CU of interferer on AP2 is less than 0.3, and chooses AP3 afterward, achieving an average throughput of 46.8 Mbps. D-SCAN shows 43% higher throughput compared to the legacy as it uses Wi-Fi channel information obtained to estimate the achievable throughput and selects a better AP.

4.5.5 Synergy for Wi-Fi and Bluetooth Coexistence

A unique advantage of D-SCAN is its ability to synergistically improve Bluetooth (BT) performance in combo devices. We first compare how standalone BT, combo-module BT, and combo-module BT aided by D-SCAN respond to interference. For the experiment, we generate a jamming signal over a 20 MHz-wide Wi-Fi channel to impede BT and measure the latency from the beginning of the jamming until the AFH map successfully excludes the Wi-Fi channel from BT's hopping sequence.

From the result in Fig. 4.19, we observe that combo BT experiences more than 6 times longer AFH map update latency, while the combo BT with D-SCAN only incurs a small additional delay of about 3 seconds. Unlike standalone BT, where BT is always active, combo BT operates with Wi-Fi in a time-division manner, which makes it difficult to achieve precise channel quality assessment. However, with the aid of D-SCAN, the AFH map of combo BT can also be rapidly updated to mitigate CTI using detailed information about Wi-Fi channels. The reasons D-SCAN combo BT has slightly higher latency than standalone BT are from (i) the overheads of D-SCAN being triggered periodically and (ii) the delay between issuing the HCI command and completing the master-slave handshaking.

If the AFH map is not updated properly in the presence of interference, BT faces collisions and makes retransmission attempts. This increases the portion of medium taken by BT and eventually deteriorates Wi-Fi performance as well, shortening the medium occupation by Wi-Fi. We conduct an experiment to evaluate the impact of BT/Wi-Fi coexistence (Fig. 4.20). We measure the saturated Wi-Fi throughput under three different scenarios: (i) only Wi-Fi in a static environment, (ii) Wi-Fi and BT in a static environment, and (iii) Wi-Fi and BT in a dynamic environment. When BT is off in a static environment, combo Wi-Fi with and without D-SCAN can achieve its

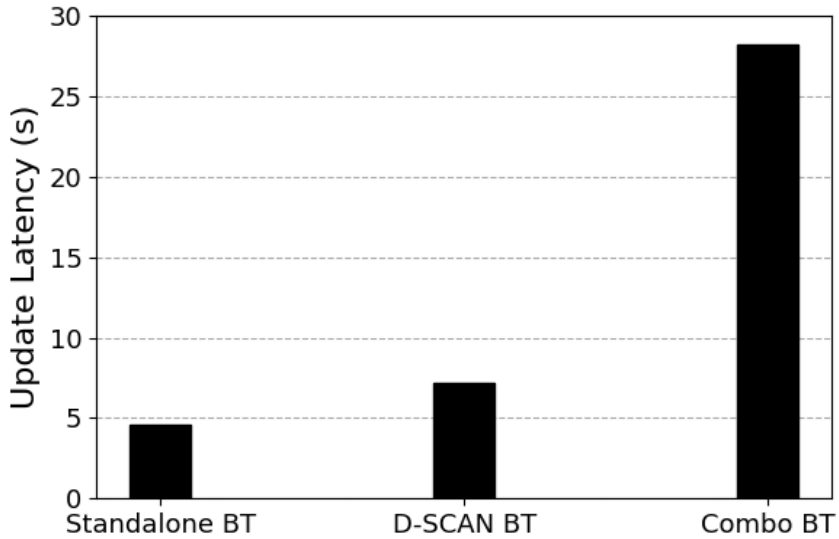


Figure 4.19: AFH map update latency.

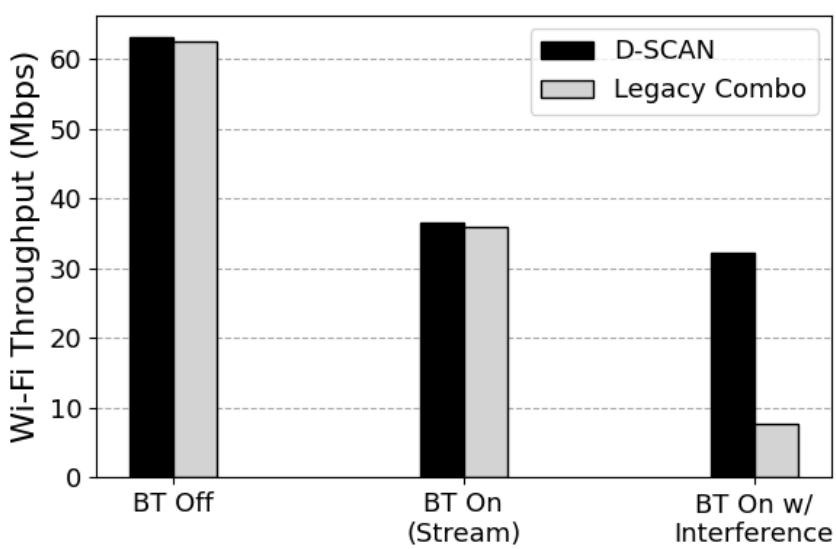


Figure 4.20: Wi-Fi throughput on different combo-module scenarios.

maximum data rate. However, when we turn on BT to stream music on a headset, the throughput becomes almost halved. Finally, when we alternately generate jamming signals on neighboring Wi-Fi channels every 30 seconds, the performance gap between combo Wi-Fi with and without D-SCAN becomes prominent. The music streamed on a headset with the legacy combo BT experiences high levels of noise and delay throughout the experiment, and Wi-Fi throughput is significantly dropped to 8 Mbps, 26.7% compared to the interference-free scenario. The combo BT with D-SCAN also experiences noise and delay early in the interference but soon stabilizes as the updated AFH map promptly responds to the CTI. Thus, we confirm that D-SCAN improves the immunity of the combo-module to interference and synergistically advances Bluetooth and Wi-Fi performance. Other IoT protocols that require cost-effective spectrum analysis can also employ the variants of D-SCAN.

4.6 Discussion

4.6.1 Sub-1 GHz and 5 GHz Wi-Fi

Wi-Fi and IoT protocols coexist in various frequency bands depending on the generation (version) and usage. The applicability of D-SCAN extends beyond the 2.4 GHz band.

In other bandwidths, Wi-Fi signal characteristics are also distinct. 802.11ah channel widths (1, 2, 4, 8, and 16 MHz) are wider compared to those of coexisting low-power wide-range protocols in Sub-1 GHz unlicensed band, such as LoRa (<500 kHz) and Sigfox (100 Hz). Wi-Fi and other heterogeneous protocols such as LTE, radar, and DSRC (dedicated short-range communication) in 5 GHz have different channel plans and are possibly distinguished. Similarly, 802.11ax, the new

generation of Wi-Fi standards, inherits these channel characteristics. We will be able to generalize D-SCAN based on the Wi-Fi characteristics in the future by arranging the sampling points with an appropriate width for Wi-Fi existing in other bands and collecting RSSIs with a low-power radio [109].

4.6.2 Bonded Wi-Fi Channels

Since 802.11n, the concept of channel bonding where two adjacent channels in 2.4 and 5 GHz spectrums are combined to increase the data rate is implemented. In the 2.4 GHz band, the channel width can be up to 40 MHz, and in the 5 GHz band, it can be up to 160 MHz.

Technically, 40 MHz signals can be detected by D-SCAN, since they also leave their traces on successive four edge projections. The strength of deep learning methods is that they learn features on their own given data with appropriate labels. In the worst case, bonded channels may bring about false-positive detections, and the inspection range of D-SCAN should be extended to 40 MHz to prevent this. However, since the presence of Bluetooth, Zigbee, and Wi-Fi on other overlapping channels often limits the availability of channel bonding, the impact on D-SCAN's detection accuracy is negligible.

4.6.3 Differentiation of APs and Clients

D-SCAN detects Wi-Fi signals well, but cannot tell if they are from APs or clients. This is a kind of exposed node problem and can cause D-SCAN devices to make false-positive decisions on AP presence, but fortunately, the rate has been kept low in our experiment. Actually, detection of close-enough Wi-Fi clients suggests that the AP it

communicates with can also empirically provide satisfactory service to the D-SCAN device. In particular, the estimated channel utilization is valid regardless of the source of the signal, providing a good inference on the achievable throughput.

4.7 Summary

In this work, we presented D-SCAN, which efficiently obtains comprehensive Wi-Fi channel information using a Bluetooth radio and synergistically improves the performance of both Wi-Fi and Bluetooth. The well-defined problem of D-SCAN and novel RSSI representation, called *edge projection*, allowed CNN to deliver robust predictions on Wi-Fi channel information even under network dynamics.

Chapter 5

Conclusion

5.1 Research Contributions

In this dissertation, we address resource management issues in three IoT protocols: LoRa/LoRaWAN, Wi-Fi, and Bluetooth. We take environment-aware approaches to diagnose radio properties and extract useful features that improve the efficiency in energy, computation, and radio resources.

In Chapter 2, we present EARN, an enhanced ADR mechanism that predicts the link performance and assigns the best parameter set in terms of energy and delivery ratio. EARN verifies its effectiveness in small-scale real-world experiments, and outperforms the conventional method in large-scale simulations.

In Chapter 3, we propose C-SCAN, which offloads the Wi-Fi scan procedure to a low-cost Bluetooth radio, and enable low-delay and energy-efficient Wi-Fi scanning. The experiments demonstrate that C-SCAN pinpoints available Wi-Fi channels with high accuracy in real-world environments.

In Chapter 4, we propose D-SCAN, which exploits a Bluetooth radio to efficiently

obtain detailed Wi-Fi channel information. A prototype of D-SCAN exploits the obtained Wi-Fi information to improve the efficiency of Wi-Fi scanning and handover, and to help Bluetooth promptly adapt to Wi-Fi interference.

5.2 Future Research Directions

Based on the findings of this dissertation, there are several future research directions to be further investigated as follows.

First, regarding the LoRa/LoRaWAN parameter allocation, we would like to further sophisticate the link performance models to reflect the processing power of the gateways, the impact of downlink messages, and other unknown variables. In this regard, we plan on applying deep learning techniques to design an ADR that learns the better use of transmission parameters from rich empirical and theoretical LoRaWAN data. We also envision ED-side ADR improvements to address the case where an ED should autonomously adjust its parameters as the downlink control messages from the server are lost.

Regarding the efficient acquirement of Wi-Fi information and coexistence of Wi-Fi and Bluetooth, we anticipate that the positive impact of C-SCAN and D-SCAN to extend beyond these two wireless technologies and beyond the 2.4 GHz band. The general strategy of using low-power wireless radios to accurately collect broad information about more energy-intensive wireless technology can be applied to other ISM bands including the widely used Sub-1 GHz, 2.4 GHz, and 5 GHz. In particular, for D-SCAN to be more effective in various environments, *federated learning* techniques, where a central server and edge devices collaborate for data collection and model training, can be applied in the future.

Bibliography

[1] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani. (2017, October). EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations. In 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 1-8). IEEE.

[2] J. T. Lim, and Y. Han. (2018). Spreading factor allocation for massive connectivity in LoRa systems. IEEE Communications Letters, 22(4), 800-803.

[3] M. Bor, and U. Roedig. (2017, June). LoRa transmission parameter selection. In 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 27-34). IEEE.

[4] B. Reynders, W. Meert, and S. Pollin. (2017, May). Power and spreading factor control in low power wide area networks. In 2017 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[5] K. Q. Abdelfadeel, V. Cionca, and D. Pesch. (2018, June). Fair adaptive data rate allocation and power control in LoRaWAN. In 2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM) (pp. 14-15). IEEE.

[6] M. F. Tuysuz, and H.A. Mantar. (2013, September). Smart channel scanning with minimized communication interruptions over IEEE 802.11 WLANs. In 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) (pp. 2218-2222). IEEE.

[7] G. Castignani, A. Arcia, and N. Montavont. (2011). A study of the discovery process in 802.11 networks. ACM SIGMOBILE Mobile Computing and Communications Review, 15(1), 25-36.

[8] A. J. Nicholson, and B. D. Noble. (2008, September). Breadcrumbs: forecasting mobile connectivity. In Proceedings of the 14th ACM international conference on Mobile computing and networking (pp. 46-57).

[9] K. H. Kim, A. W. Min, D. Gupta, P. Mohapatra, and J. P. Singh. (2011, April). Improving energy efficiency of Wi-Fi sensing on smartphones. In 2011 Proceedings IEEE INFOCOM (pp. 2930-2938). IEEE.

[10] K. Doppler, C. B. Ribeiro, and J. Kneckt. (2011, March). On efficient discovery of next generation local area networks. In 2011 IEEE Wireless Communications and Networking Conference (pp. 269-274). IEEE.

[11] N. Montavont, A. Arcia-Moret, and G. Castignani. (2013, September). On the selection of scanning parameters in IEEE 802.11 networks. In 2013 IEEE 24th annual international symposium on personal, indoor, and mobile radio communications (PIMRC) (pp. 2137-2141). IEEE.

[12] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne. (2004, October). Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs. In Proceedings of the second international workshop on Mobility management

& wireless access protocols (pp. 19-26).

[13] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. (2010, September). ZiFi: Wireless LAN discovery via ZigBee interference signatures. In Proceedings of the sixteenth annual international conference on Mobile computing and networking (pp. 49-60).

[14] J. Yi, W. Sun, J. Koo, S. Byeon, J. Choi, and S. Choi. (2018, June). BlueScan: Boosting Wi-Fi scanning efficiency using bluetooth radio. In 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) (pp. 1-9). IEEE.

[15] C. F. Chiasserini, and R. R. Rao. (2001). Improving battery performance by using traffic shaping techniques. IEEE Journal on Selected Areas in Communications, 19(7), 1385-1394.

[16] G. Ananthanarayanan, and I. Stoica. (2009, June). Blue-Fi: enhancing Wi-Fi performance using bluetooth signals. In Proceedings of the 7th international conference on Mobile systems, applications, and services (pp. 249-262).

[17] N. Mishra, K. Chebrolu, B. Raman, and A. Pathak. (2006, May). Wake-on-WLAN. In Proceedings of the 15th international conference on World Wide Web (pp. 761-769).

[18] K. Chebrolu, and A. Dhekne, A. (2009, September). Esense: Communication through energy sensing. In Proceedings of the 15th annual international conference on Mobile computing and networking (pp. 85-96).

[19] M. Hou, F. Ren, C. Lin, and M. Miao. (2014, June). HEIR: Heterogeneous interference recognition for wireless sensor networks. In Proceeding of IEEE

International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014 (pp. 1-9). IEEE.

[20] C. F. Chiasserini, and R. R. Rao. (2003). Coexistence mechanisms for interference mitigation in the 2.4-GHz ISM band. IEEE Transactions on Wireless Communications, 2(5), 964-975.

[21] C. D. M. Cordeiro, S. Abhyankar, R. Toshiwal, and D. P. Agrawal. (2004). BlueStar: enabling efficient integration between Bluetooth WPANs and IEEE 802.11 WLANs. Mobile Networks and Applications, 9(4), 409-422.

[22] X. Zhang, and K. G. Shin. (2013, April). Gap sense: Lightweight coordination of heterogeneous wireless devices. In 2013 Proceedings IEEE INFOCOM (pp. 3094-3101). IEEE.

[23] Y. Wang, Q. Wang, G. Zheng, Z. Zeng, R. Zheng, and Q. Zhang. (2013). WiCop: Engineering WiFi temporal white-spaces for safe operations of wireless personal area networks in medical applications. IEEE transactions on mobile computing, 13(5), 1145-1158.

[24] Specification of the Bluetooth System, Covered Core Package Version: 4.2, The Bluetooth Special Interest Group (SIG), 2013.

[25] L. Ophir, Y. Bitran, and I. Sherman. (2004, September). Wi-Fi (IEEE 802.11) and Bluetooth coexistence: Issues and solutions. In 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754) (Vol. 2, pp. 847-852). IEEE.

[26] A. E. Xhafa, and Y. Sun. (2013, January). Mechanisms for coexistence of collocated WLAN and bluetooth in the same device. In 2013 International

Conference on Computing, Networking and Communications (ICNC) (pp. 905-910). IEEE.

[27] N. Mishra, D. Golcha, A. Bhadauria, B. Raman, and K. Chebrolu. (2007, January). S-WOW: Signature based wake-on-WLAN. In 2007 2nd International Conference on Communication Systems Software and Middleware (pp. 1-8). IEEE.

[28] J. Choi, G. Lee, Y. Shin, J. Koo, M. Jang, and S. Choi. (2018, June). Blend: BLE beacon-aided fast wifi handoff for smartphones. In 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) (pp. 1-9). IEEE.

[29] W. Park, D. Ryoo, C. Joo, and S. Bahk. (2021, May). BLESS: BLE-aided Swift Wi-Fi Scanning in Multi-protocol IoT Networks. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications (pp. 1-10). IEEE.

[30] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso. (2016, November). Do LoRa low-power wide-area networks scale?. In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 59-67).

[31] O. Georgiou, and U. Raza. (2017). Low power wide area network analysis: Can LoRa scale?. IEEE Wireless Communications Letters, 6(2), 162-165.

[32] B. Ghena, J. Adkins, L. Shangguan, K. Jamieson, P. Levis, and P. Dutta. (2019, October). Challenge: Unlicensed lpwans are not yet the path to ubiquitous connectivity. In The 25th Annual International Conference on Mobile Computing and Networking (pp. 1-12).

[33] LoRa Alliance, “A technical overview of lora and lorawan,” White paper, 2015.

[34] Semtech, “SX1272/73 - 860 MHz to 1020 MHz low power long range transceiver,” https://www.semtech.com/uploads/documents/SX1272 DS V4.pdf, 2015.

[35] LoRa Alliance, “LoRaWANTM 1.1 Regional Parameters,” In: LoRa Alliance, 2017.

[36] The Things Network, “The Thing Network Wiki: Adaptive Data Rate,” https://www.thethingsnetwork.org/wiki/LoRaWAN/ADR, 2017.

[37] LoRa Alliance, “LoRaWANTM 1.1 Specification,” In: LoRa Alliance, 2017.

[38] A. Hoeller, R. D. Souza, O. L. A. López, H. Alves, M. de Noronha Neto, and G. Brante. (2018, August). Exploiting time diversity of LoRa networks through optimum message replication. In 2018 15th International Symposium on Wireless Communication Systems (ISWCS) (pp. 1-5). IEEE.

[39] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo. (2015, December). On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. In 2015 14th international conference on its telecommunications (itst) (pp. 55-59). IEEE.

[40] M. Cattani, C. A. Boano, and K. Römer. (2017). An experimental evaluation of the reliability of lora long-range low-power wireless communication. Journal of Sensor and Actuator Networks, 6(2), 7.

[41] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan. (2017, August). Empowering low-power wide area networks in urban settings. In Proceedings of the

Conference of the ACM Special Interest Group on Data Communication (pp. 309-321).

[42] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, ... and A. Rowe. (2018, April). Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks. In 2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) (pp. 60-71). IEEE.

[43] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota. (2017). Lora backscatter: Enabling the vision of ubiquitous connectivity. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(3), 1-24.

[44] A. Varshney, O. Harms, C. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt. (2017, November). Lorea: A backscatter architecture that achieves a long communication range. In Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (pp. 1-14).

[45] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, ... and K. Jamieson. (2018, August). PLoRa: A passive long-range data network from ambient LoRa transmissions. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (pp. 147-160).

[46] T. Polonelli, D. Brunelli, and L. Benini. (2018, October). Slotted aloha overlay on lorawan-a distributed synchronization approach. In 2018 IEEE 16th international conference on embedded and ubiquitous computing (EUC) (pp. 129-132). IEEE.

[47] T. H. To, and A. Duda. (2018, May). Simulation of lora in ns-3: Improving lora performance with csma. In 2018 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.

[48] B. Reynders, W. Meert, and S. Pollin. (2016, May). Range and coexistence analysis of long range unlicensed communication. In 2016 23rd International Conference on Telecommunications (ICT) (pp. 1-6). IEEE.

[49] Simpy, "Event discrete simulation for Python," https://simpy.readthedocs.io., 2016.

[50] "Cisco visual networking index: Forecast and methodology, 2016–2021," San Jose, CA, USA, Cisco, White Paper, Jun. 2017.

[51] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni. (2012). CSI-based indoor localization. IEEE Transactions on Parallel and Distributed Systems, 24(7), 1300-1309.

[52] J. Choi. (2016). WidthSense: Wi-Fi discovery via distance-based correlation analysis. IEEE Communications Letters, 21(2), 422-425.

[53] Ubertooth-One. Accessed: Jan. 10, 2017. [Online]. Available: http://ubertooth.sourceforge.net/

[54] How Wi-Fi Drains Your Cell Phone. Accessed: Jun. 24, 2010. [Online]. Available: http://www.technologyreview.com/

[55] A. Gupta, and P. Mohapatra. (2007, June). Energy consumption and conservation in wifi based phones: A measurement-based study. In 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (pp. 122-131). IEEE.

[56] R. Raghavendra, E. M. Belding, K. Papagiannaki, and K. C. Almeroth. (2010). Unwanted link layer traffic in large IEEE 802.11 wireless networks. IEEE Transactions on Mobile Computing, 9(9), 1212-1225.

[57] A. Gupta, J. Min, and I. Rhee. (2012, December). WiFox: Scaling WiFi performance for large audience environments. In Proceedings of the 8th international conference on Emerging networking experiments and technologies (pp. 217-228).

[58] J. Yeo, M. Youssef, and A. Agrawala. (2004, October). A framework for wireless LAN monitoring and its applications. In Proceedings of the 3rd ACM workshop on Wireless security (pp. 70-79).

[59] S. Rayanchu, A. Patro, and S. Banerjee. (2012). Catching whales and minnows using wifinet: Deconstructing non-wifi interference using wifi hardware. In 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12) (pp. 57-70).

[60] X. Hu, L. Song, D. Van Bruggen, and A. Striegel. (2015, October). Is there WiFi yet? How aggressive probe requests deteriorate energy and throughput. In Proceedings of the 2015 Internet Measurement Conference (pp. 317-323).

[61] W. Wang, M. Motani, and V. Srinivasan. (2009). Opportunistic energy-efficient contact probing in delay-tolerant applications. IEEE/ACM Transactions on Networking, 17(5), 1592-1605.

[62] H. Velayos, and G. Karlsson. (2004, June). Techniques to reduce the IEEE 802.11 b handoff time. In 2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577) (Vol. 7, pp. 3844-3848). IEEE.

[63] H. Wu, K. Tan, Y. Zhang, and Q. Zhang. (2007, May). Proactive scan: Fast handoff with smart triggers for 802.11 wireless LAN. In IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications (pp. 749-757). IEEE.

[64] J. Teng, C. Xu, W. Jia, and D. Xuan. (2009, April). D-scan: Enabling fast and smooth handoffs in ap-dense 802.11 wireless networks. In IEEE INFOCOM 2009 (pp. 2616-2620). IEEE.

[65] J. Eriksson, H. Balakrishnan, and S. Madden. (2008, September). Cabernet: Vehicular content delivery using WiFi. In Proceedings of the 14th ACM international conference on Mobile computing and networking (pp. 199-210).

[66] N. Chakraborty, B. N. P. Sinha, S. H. Nizamie, V. K. Sinha, S. Akhtar, J. Beck, and B. Binha. (2006). Effectiveness of continuing nursing education program in child psychiatry. Journal of Child and Adolescent Psychiatric Nursing, 19(1), 21-28.

[67] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das. (2009, June). Predictive methods for improved vehicular WiFi access. In Proceedings of the 7th international conference on Mobile systems, applications, and services (pp. 263-276).

[68] N. Poosamani, and I. Rhee. (2015, August). Wi-fi hotspot auto-discovery: A practical & energy-aware system for smart objects using cellular signals. In proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (pp. 90-99).

[69] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, V. Bahl, and R. Gupta. (2009). Somniloquy: augmenting network interfaces to reduce pc energy usage.

[70] A. Currid. (2004). TCP Offload to the Rescue: Getting a toehold on TCP offload engines—and why we need them. Queue, 2(3), 58-65.

[71] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. (2012, June). FM-based indoor localization. In Proceedings of the 10th international conference on Mobile systems, applications, and services (pp. 169-182).

[72] T. Li, C. An, R. Chandra, A. T. Campbell, and X. Zhou. (2015, September). Low-power pervasive wi-fi connectivity using WiScan. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 409-420).

[73] S. Cui, A. J. Goldsmith, and A. Bahai. (2004). Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks. IEEE Journal on selected areas in communications, 22(6), 1089-1098.

[74] P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar. (2006). Cooperative wireless communications: A cross-layer approach. IEEE Wireless communications, 13(4), 84-92.

[75] J. Park, E. Song, and W. Sung. (2009). Capacity analysis for distributed antenna systems using cooperative transmission schemes in fading channels. IEEE transactions on wireless communications, 8(2), 586-592.

[76] H. Wang, X. G. Xia, and Q. Yin. (2009). Distributed space-frequency codes for cooperative communication systems with multiple carrier frequency offsets. IEEE Transactions on Wireless Communications, 8(2), 1045-1055.

[77] B. Sirkeci-Mergen, and A. Scaglione. (2007). Randomized space-time coding for distributed cooperative communication. IEEE Transactions on Signal Processing, 55(10), 5003-5017.

[78] X. Li, T. Jiang, S. Cui, J. An, and Q. Zhang. (2010). Cooperative communications based on rateless network coding in distributed MIMO systems [coordinated and distributed MIMO]. IEEE wireless communications, 17(3), 60-67.

[79] Q. Zhang, J. Jia, and J. Zhang. (2009). Cooperative relay to improve diversity in cognitive radio networks. IEEE Communications Magazine, 47(2), 111-117.

[80] J. Jia, J. Zhang, and Q. Zhang. (2009, April). Cooperative relay for cognitive radio networks. In IEEE INFOCOM 2009 (pp. 2304-2312). IEEE.

[81] L. Li, X. Zhou, H. Xu, G. Y. Li, D. Wang, and A. Soong. (2010). Simplified relay selection and power allocation in cooperative cognitive radio systems. IEEE Transactions on Wireless Communications, 10(1), 33-36.

[82] Y. Zou, Y. D. Yao, and B. Zheng. (2012). Cooperative relay techniques for cognitive radio systems: Spectrum sensing and secondary user transmissions. IEEE Communications Magazine, 50(4), 98-103.

[83] E. Shih, P. Bahl, and M. J. Sinclair. (2002, September). Wake on wireless: An event driven energy saving strategy for battery operated devices. In Proceedings of the 8th annual international conference on Mobile computing and networking (pp. 160-171).

[84] J. Sorber, N. Banerjee, M. D. Corner, and S. Rollins. (2005, June). Turducken: Hierarchical power management for mobile devices. In Proceedings of the 3rd

international conference on Mobile systems, applications, and services (pp. 261-274).

[85] X. Zhang, and K. G. Shin. (2012). E-MiLi: Energy-minimizing idle listening in wireless networks. IEEE Transactions on Mobile Computing, 11(9), 1441-1454.

[86] W. Wang, Y. Chen, L. Wang, and Q. Zhang. (2017). Sampleless wi-fi: Bringing low power to wi-fi communications. IEEE/ACM Transactions on Networking, 25(3), 1663-1672.

[87] Texas Instruments. CC2420 IEEE 802.15.4 Compliant and ZigBee Ready RF Transceiver. Accessed: May 1, 2016. [Online]. Available: http://www.ti.com/product/CC2420.

[88] Android WiFiManager. Accessed: May 1, 2016. [Online]. Available: https://developer.android.com/reference/android/net/wifi/WifiManager.html.

[89] List of 2.4 GHz Radio Use. Accessed: May 1, 2016. [Online]. Available: https://en.wikipedia.org/wiki/List_of_2.4_GHz_radio_use.

[90] T. S. Rappaport. (1996). Wireless communications: principles and practice (Vol. 2). New Jersey: prentice hall PTR.

[91] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh. (2006). Data mining: A preprocessing engine. Journal of Computer Science, 2(9), 735-739.

[92] Monsoon Power Monitor. Accessed: Mar. 1, 2017. [Online]. Available: https://www.msoon.com/LabEquipment/PowerMonitor/.

[93] A. E. Xhafa, X. Lu, and D. P. Shaver. (2008, October). Coexistence of

collocated ieee 802.11 and bluetooth technologies in 2.4 ghz ism band. In International Conference on Access Networks (pp. 138-145). Springer, Berlin, Heidelberg.

[94] T. Kessler. “turn off Bluetooth to fix Airplay mirroring bug in OS X”, http://www.cnet.com/how-to/, 2014.

[95] J. Chung, J. Park, C. K. Kim, and J. Choi. (2018). C-SCAN: Wi-Fi scan offloading via collocated low-power radios. IEEE Internet of Things Journal, 5(2), 1142-1155.

[96] IEEE 802.11, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Std., Mar. 2012.

[97] X. Chen, and D. Qiao. (2010, March). HaND: Fast handoff with null dwell time for IEEE 802.11 networks. In 2010 Proceedings IEEE INFOCOM (pp. 1-9). IEEE.

[98] A. Graves, N. Jaitly, and A. R. Mohamed. (2013, December). Hybrid speech recognition with deep bidirectional LSTM. In 2013 IEEE workshop on automatic speech recognition and understanding (pp. 273-278). IEEE.

[99] R. Johnson, and T. Zhang. (2016, June). Supervised and semi-supervised text categorization using LSTM for region embeddings. In International Conference on Machine Learning (pp. 526-534). PMLR.

[100] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang. (2017, May). Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In IEEE INFOCOM 2017-IEEE Conference on Computer Communications (pp. 1-9). IEEE.

[101] A. Krizhevsky, I. Sutskever, and G. E. Hinton. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[102] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, ... and A. Rabinovich. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[103] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury. (2019, April). Oracle: Optimized radio classification through convolutional neural networks. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications (pp. 370-378). IEEE.

[104] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, ... and H. Adam. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[105] M. Tan, and Q. Le. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114). PMLR.

[106] D. Jaisinghani, V. Naik, S. K. Kaul, and S. Roy. (2015, May). Realtime detection of degradation in WiFi network's goodput due to probe traffic. In 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt) (pp. 42-47). IEEE.

[107] H. D. Balbi, D. Passos, R.C. Carrano, L. C. Magalhães, and C. V. Albuquerque. (2020). Association stability and handoff latency tradeoff in dense IEEE 802.11 networks: A case study. Computer Communications, 159, 175-185.

[108] Linux backports. https://backports.wiki.kernel.org/.

[109] M. Hessar, A. Najafi, V. Iyer, and S. Gollakota. (2020). TinySDR: Low-Power SDR Platform for Over-the-Air Programmable IoT Testbeds. In 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20) (pp. 1031-1046).

초 록

수많은 기기가 서로 연결되는 초연결 사회를 현실화하기 위해 다양한 사물인터넷(IoT) 프로토콜이 등장했다. 특히 LoRa/LoRaWAN, Wi-Fi, 그리고 Bluetooth와 같은 비면허 대역 프로토콜은 저렴한 비용으로 공공 및 사설 네트워크의 구축을 용이하게 하여, 도시 규모에서 근거리에 이르기까지 광범위한 서비스에 널리 적용되었다. 그러나 이기종 IoT 프로토콜에서는 통신 기기들의 안정적인 연결을 보장하는 동시에, 에너지, 연산 및 무선 자원을 효율적으로 관리토록 하는 연구 과제를 제시한다. 해당 과제를 수행함에 있어, 구성 가능한 다양한 무선 매개변수 조합, 그로 인한 링크 성능상의 복잡한 절충 관계, 그리고 끊임없이 변화하는 무선 환경이 주요 장애물로 작용한다. 이러한 특성 때문에 일반적으로 효과적인 자원 관리 전략을 고안하기 위해 자칫 불규칙적으로 보이는 무선 속성을 정확하게 진단할 수 있는 네트워크 전문 지식이 필요로 된다.

이 논문에서는 1) LoRa/LoRaWAN의 전송 매개변수 제어, 2) 저전력 라디오를 통한 Wi-Fi 스캔 오프로딩, 3) Wi-Fi 및 Bluetooth의 협동적 공존 세 가지 연구 주제를 다루며, 이 과정에서 우리는 무선 신호에서 추출한 유용한 상황 정보를 활용하여 기기들의 자원을 효율적으로 관리한다.

첫째, 비면허 대역에서 작동하는 가장 유망한 LPWA 프로토콜 중 하나인 LoRa는 무선 자원 및 링크 성능을 관리하기 위해 적절하게 제어해야 하는 전송 매개변수 집합을 제시한다. 이 연구에서 우리는 프레임 전달율과 에너지 소비 간의 균형을 최적화하기 위해 CR (Coding Rate)을 적응적으로 활용하는 향상된 ADR (Adaptive Data Rate) 메커니즘인 EARN을 제안한다. EARN은 먼저 이론적으로 LoRaWAN의 링크 성능을 모델링하여 최상의 매개변수 집합을 찾고, 캡처 효과를 이용하여 충돌하는 신호의 생존율을 높인다. 우리는 실제 실험에서 적응적 CR 설정의 타당성을 검증하고, 대규모 시뮬레이션을 통해 EARN이 기존 방식을 능가하는 것을 보인다.

둘째, 스마트 기기에서 고품질의 무선 연결을 위한 기본 기능인 Wi-Fi 스캐닝은 액세스 포인트(AP)가 존재하지 않는 채널에 대해 불필요한 검색을 수행하여 자원 및 성능상의 낭비를 초래한다. 우리는 함께

배치된 저전력 무선 인터페이스를 활용하여 이러한 Wi-Fi 스캐닝 오버헤드를 상쇄하는 C-SCAN 방법을 제안한다. C-SCAN은 Bluetooth 라디오로 2.4 GHz 공유 스펙트럼을 검사하고 실제 Wi-Fi 스캔 전에 사용 중인 Wi-Fi 채널을 식별한다. 비어 있는 것으로 판단된 채널을 검색 대상에서 제외함으로써 Wi-Fi 스캐닝의 대기 시간과 에너지를 절약할 수 있다. 실험 결과는 우리가 구현한 C-SCAN의 프로토타입이 밀집된 Wi-Fi 환경에서도 사용 중인 Wi-Fi 채널을 정확히 탐지하는 것을 보인다.

마지막으로, 최신 스마트 기기에 함께 배치된 Wi-Fi와 Bluetooth는 단일 안테나와 스펙트럼을 공유하기 때문에 내외부적으로 기술 간 간섭(CTI)을 겪는다. 우리는 이 연구에서 새로운 협동적 공존 메커니즘인 D-SCAN을 제안한다. D-SCAN은 Bluetooth 라디오로 주변 Wi-Fi에 대한 포괄적인 정보를 효율적으로 추론하여 주요 Wi-Fi 기능의 오버헤드를 상쇄하고 나아가 Wi-Fi와 Bluetooth 간의 충돌을 방지한다. 이를 위해 D-SCAN은 Bluetooth의 스펙트럼 측정 결과에 심층 신경망을 활용하여 Wi-Fi 신호의 고유한 시간 및 스펙트럼 특성을 캡처하는 데이터 기반 접근 방식을 채택한다. 실제 실험에서 D-SCAN 프로토타입은 기존 Wi-Fi 스캐닝의 대기 시간과 에너지 소비를 줄이고 Bluetooth의 간섭 회피를 촉진한다.

주요어: 비면허대역, 로라, 와이파이, 블루투스, 자원 관리, 공존.
학 번: 2013-23115