

Out-of-Band Sensing with ZigBee for Dynamic Channel Assignment in On-the-Move Hotspots

Jaehyuk Choi^{†*} and Kang G. Shin^{*}

[†]Dept. of Software Design & Management, Kyungwon University, Seongnam, Korea 461-701

Email: jchoi@kyungwon.ac.kr

^{*}Real-Time Computing Laboratory, Dept. of EECS, The University of Michigan, Ann Arbor, MI 48109-2121

Email: kgshin@eecs.umich.edu

Abstract—Mobile WiFi hotspots have become increasingly popular as a new innovative wireless Internet access technology. Although it has received little attention, adaptive, dynamic (re)assignment of channels in a mobile hotspot router—typically, a smartphone or a laptop or a tablet equipped with heterogeneous network interfaces like 3G/4G, WiFi, Bluetooth or ZigBee—is key to mobile hotspot performance. To fill this important gap, we present a novel scheme, called *Ex2R*, that finds the best WiFi service-channel by harnessing secondary low-power ZigBee radios accompanied in mobile hotspot systems, enabling intelligent and seamless dynamic channel reconfiguration. *Ex2R* exploits the RSSI values and clear channel assessment (CCA) outputs from the ZigBee radio sampling interface to measure idle-time fractions on candidate WiFi channels, and determines the channel that can provide the highest hotspot capacity. *Ex2R* thus provides a mobile hotspot router accurate information necessary for the selection of the best channel. We have implemented and evaluated *Ex2R*, demonstrating its effectiveness; *Ex2R* accurately ranks WiFi channels for a mobile hotspot router to dynamically select and switch to the best channel available, thus improving hotspot performance significantly.

I. INTRODUCTION

Recently, mobile WiFi hotspots are becoming popular as a new innovative wireless Internet access technology. A *mobile WiFi hotspot* offers on-the-move Internet access to WiFi end-devices over IEEE 802.11-based wireless local area networks (WLANs) using a *mobile Access Point (AP)*—typically, a tethering-capable smartphone or a dedicated hardware known as *mobile router* [28], equipped with heterogeneous network interfaces like 3G/4G, WiFi, Bluetooth or ZigBee—connected to an Internet service provider via a broadband data network. The chief advantage of a mobile hotspot is that it provides ubiquitous and secure, yet high-speed (up to 6 Mbps download and 1 Mbps upload speed,¹ even on the move [26]) Internet connectivity anywhere and anytime.

Unlike traditional stationary WiFi systems in which APs are immobile and experience infrequent topology changes, a mobile WiFi hotspot can be turned on at any location and at any time, thanks to its portability. The network condition for hotspot service on a given channel exhibits significant temporal and spatial variations, depending on the deployment status of nearby WiFi systems [4], [17]. Therefore, the best-conditioned channel of a mobile hotspot also varies with time

and location. One can thus improve hotspot performance and capacity significantly by dynamic and adaptive assignment of channels to a mobile router based on the current level of interference and contention on each channel.

Despite its potential for high performance and capacity, however, dynamic channel (re) assignment in mobile hotspot environments has received little attention. Most of current commercial mobile routers [28] are not designed for dynamic and adaptive channel reconfiguration. They usually initialize operating channels based on manual input, often yielding unsatisfactory performance, including as relatively high transmission errors and low network capacity. Although considerable research efforts have been made on intelligent channel assignment [4], [17], the state-of-the-art solutions are not applicable to mobile routers as they focus only on *static* environments. Moreover, most of them are based on a single interface and use an active (i.e., in use) WiFi interface for probing channels. However, probing channels takes time and prevents hotspot clients from communication with the router, thus incurring a significant delay to the clients. Such a delay is usually unacceptable to delay-sensitive applications like VoIP.

In this paper, we tackle the problem of dynamic channel selection for mobile hotspot systems. Specifically, we present a novel scheme, called *Exploiting Secondary Radio (Ex2R)*, that provides channel information to a mobile hotspot router for dynamic (re)assignment of the best channel to its WiFi interface, thereby improving hotspot performance. *Ex2R* is motivated by the rapid spread of tethering-capable mobile devices (e.g., smartphones, laptops, and tablets) equipped with multiple heterogeneous network interfaces, including a low-power wireless personal area network (WPAN) interface like ZigBee or Bluetooth, and a WiFi interface [23]. *Ex2R* harnesses the fact that WiFi signals in the 2.4 GHz ISM band can be sensed via a low-power WPAN radio (i.e., IEEE 802.15.4 ZigBee), and thus, the state of a target WiFi channel can be monitored via ZigBee channels that overlap with the target WiFi channel. In particular, *Ex2R* senses the idle time between energy bursts by using a built-in RSSI sampling mechanism of ZigBee radios, and estimates the available capacity of a mobile hotspot on each candidate WiFi channel. As a result, *Ex2R* provides a mobile hotspot router accurate information necessary for its selection of the best channel, thereby improving the mobile hotspot performance.

¹The 4G wireless technology, such as LTE (Long Term Evolution) and WiMAX, will provide 10 times faster Internet speed than 3G [21].

We have prototyped Ex2R on a Linux platform composed of a CC2420-based MicaZ and an 802.11 network card, demonstrating its efficacy and efficiency. Our experimental results show that Ex2R accurately ranks candidate WiFi channels for dynamic channel switching to improve hotspot performance. Although we utilize the RSSI sampling interface of a ZigBee radio for the implementation of Ex2R in the rest of this paper, Ex2R is general enough to be implemented on other platforms that offer the RSS sampling interface for the 2.4GHz ISM spectrum band, such as Bluetooth radios.

This paper makes the following three main contributions. First, we present a simple and novel dynamic channel reconfiguration framework, Ex2R, that provides accurate channel information to mobile hotspot systems for effective selection of the best channel. Second, we analytically and experimentally show that the idle-time fraction on each channel accurately reflects the available capacity of a mobile hotspot. In addition, we present an integrated technique for computing the idle-time fraction on a WiFi channel based on the idle-time fractions for the four ZigBee channels that overlap with the WiFi channel. To achieve these, we overcome the limitation of ZigBee radio’s coarse-grained sensing capability, and develop an accurate channel measurement scheme, called *Idle-Sampler*. Finally, we have implemented a prototype of Ex2R and conducted an extensive experimental evaluation, demonstrating Ex2R’s ranking of WiFi candidate channels with high accuracy.

The rest of the paper is organized as follows. Section II summarizes the work related to our approach. Section III presents an overview of Ex2R architecture and discusses the channel-selection strategy of Ex2R. Section IV elaborates on the design of Ex2R. Section V presents a detailed evaluation of Ex2R. Finally, Section VI discusses future extensions to Ex2R and concludes the paper.

II. RELATED WORK

Related work falls under two categories: (i) techniques for improving WiFi performance by intelligent channel allocation, and (ii) techniques harnessing heterogeneous radios for optimizing application performance and minimizing device power consumption.

There have been considerable research efforts on intelligent channel selection in 802.11 WiFi networks [4], [9], [17], [19]. Many commercial AP vendors implement a simple distributed scheme, commonly called the *least congested channel search* (LCCS) [9]. With LCCS, an AP searches for the least-loaded channel, i.e., the channel with the least amount of traffic. The authors of [17] presented a dynamic channel hopping protocol that distributively assigns the channel of an AP in order to resolve the unfairness problem in uncoordinated wireless environments. It provides good fairness and can also take advantage of partially-overlapped channels. A recent study in [19] considered the traffic pattern for the selection of channels, showing that traffic-aware channel assignment can significantly improve the quality of channel assignment. These approaches have potential for achieving high throughput of a WiFi hotspot, but most of them focused on *stationary*

environments, such as campus and enterprise networks, and they are network-driven solutions. In addition, most of them are single interface-based approaches [4], [9], [17], [19] that use an active WiFi interface to probe channels for channel reassignment. However, mobile routers have to suspend their service during their probing, which may incur significant delays to hotspot clients. We address this issue by out-of-band sensing with coexisting WPAN interfaces.

Recently, several approaches utilizing secondary coexisting radios, such as Bluetooth and ZigBee, have been proposed to optimize application performance and minimize device power consumption [3], [5]–[7], [20], [22]. The idea of using a secondary low-power radio for saving device power consumption was first proposed in [7]. Two recent research efforts [5], [22] exploit interface heterogeneity to reduce energy cost by reducing the number of unnecessary energy-consuming scans of WiFi interfaces. In [5], Blue-Fi uses Bluetooth contact-patterns as context identifiers for predicting WiFi availability. Footprint in [22] leverages cellular information, such as the overheard cellular tower IDs and signal strength, to guide the WiFi scan.

Two recent proposals introduced in [6], [23] are most relevant to Ex2R as they harness WiFi and ZigBee cooperation. In [23], the authors proposed a system, called ZiFi, that utilizes a low-power ZigBee radio to discover available WiFi Internet connectivity. ZiFi senses WiFi signals by using a ZigBee’s RSSI interface and identifies unique signatures of periodic WiFi beacons. In [6], the authors proposed Essense that uses the ZigBee’s energy sensing feature to enable communication between ZigBee and WiFi devices which have fundamentally different physical layers.

Although these techniques can be used to enhance our solution, none of them directly addresses the design of dynamic channel reconfiguration that harnesses a secondary low-power radio in a mobile hotspot system. To the best of our knowledge, Ex2R is the first to consider dynamic channel reconfiguration in a mobile hotspot system by exploiting a low power radio.

III. PROBLEM CONTEXT

In this section, we first overview the problem. We then discuss the channel-selection strategy of Ex2R.

A. Overview

We consider the channel selection problem in a mobile hotspot system. In this paper, we will use the term “out-of-band sensing” to mean measuring the state of the candidate WiFi channels that are not being used at a hotspot. Ex2R uses a ZigBee interface—i.e., an 802.15.4 ZigBee-compliant radio, like a CC2420 RF transceiver [29]—to perform out-of-band sensing and provides the information on the best channel to the hotspot system for dynamic channel (re)assignment. Ex2R can be implemented in various types of mobile hotspot system, such as tethering-capable platforms (e.g., smartphones, tablets, or laptops) that are equipped with multiple types of network interface—e.g., 802.11-based WiFi, 3G/4G mobile broadband,

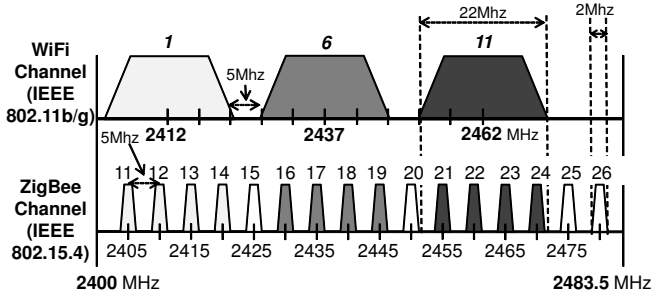


Fig. 1. IEEE 802.11b/g and IEEE 802.15.4 frequency channels in the 2.4 GHz ISM band. Each 802.11 channel (22 MHz wide) is composed of four 802.15.4 channels (each 2 MHz wide) overlapping with the 802.11 channel.

and low-power ZigBee interfaces—and dedicated broadband mobile routers (e.g., Mi-Fi [28] [30]) that can connect to an external ZigBee interface. A mote-based board can be designed as a dongle to be connected to off-the-shelf WiFi routers, e.g., AirLink AR660W3G [24].

For a link i of a target WiFi hotspot, the optimal channel-selection problem can be formulated as:

$$c_i^* = \arg \max_{c \in \mathcal{C}} S_i(c), \quad (1)$$

where \mathcal{C} is the set of channels for the WiFi hotspot, e.g., $\mathcal{C} = \{1, 2, \dots, 11\}$ for 11 channels available in 802.11b/g WLANs, and $S_i(c)$ is the available throughput capacity of link i on channel c . In other words, Ex2R selects the channel $c \in \mathcal{C}$ that maximizes the expected achievable capacity, $S_i(c)$.

Spectrum Sharing in 2.4-GHz ISM Band: Fig. 1 depicts the frequency allocations of IEEE 802.11b/g and IEEE 802.15.4 in the 2.4 GHz ISM band [1], [2]. The 802.11b/g WLAN standard defines 11 channels within this band, numbered 1 to 11, each with bandwidth of 22 MHz and channel separation of 5 MHz. The 802.15.4 standard operates on a total of 16 channels in the band, numbered 11 to 26, each 2 MHz wide with channel separation of 5 MHz. Each WiFi (802.11) channel overlaps with 4 consecutive ZigBee (802.15.4) channels. The 802.15.4 ZigBee radio offers a built-in mechanism for sampling the RSSI and reporting the clear-channel assessment (CCA) output that indicates if the RSSI is above a programmable (i.e., carrier sense) threshold. Therefore, the WiFi signals in the 2.4 GHz band can be sensed via ZigBee, and Ex2R exploits this capability to sense and assess the state of a WiFi channel via one or more overlapping ZigBee channels [23].

Notations: We use the notation c to represent a WiFi channel, and u will denote a ZigBee channel. Let $\mathcal{Z}_{map}(c)$ denote the set of ZigBee channels overlapping with WiFi channel c , e.g., $\mathcal{Z}_{map}(1) = \{11, 12, 13, 14\}$, and $\mathcal{Z}_{map}(6) = \{16, 17, 18, 19\}$ as shown in Fig. 1.

B. Channel-selection using Coexisting ZigBee Radio

The effectiveness of a channel assignment hinges heavily on accurate assessment of the achievable capacity of a given 802.11 wireless link on each candidate channel. Therefore, we present a methodology that can estimate the available

throughput capacity of the link between a mobile router (AP) and its client nodes.

1) *Available 802.11 Link Capacity:* In an 802.11 WLAN, the capacity of a link is determined primarily by two factors: (a) PHY-layer *link*² *quality* and (b) MAC-layer *neighboring contention* [18]. The physical link quality determines the bit error rate for a given PHY-layer modulation scheme (i.e., transmission rate). The neighboring contention or interference is caused by traffic from multiple surrounding nodes within the sensing range. Our goal here is to estimate the maximum available throughput capacity of a target link by capturing the PHY- and MAC-layer behavior of 802.11 WLANs.

Let's consider a target link i on channel c . From the perspective of the sending node of link i , the link's activity can be characterized with the following three different channel states [8]: (a) *self channel* occupied by the node's own transmission, (b) *busy channel* due to the activity of other neighboring nodes, and (c) *idle channel* when the channel is not used by any node. Let $X_i(c)$, $Y_i(c)$, and $Z_i(c)$ denote, respectively, the fraction of time link i stays in these three states on channel c . These variables are determined by the PHY- and MAC-layer behavior of nearby 802.11 links, including link i and neighboring links, on channel c . Thus, the packet-error transmission probability of link i depends on the state of channel c . Let $p_i(c)$ denote the conditional transmission error probability³ (due to faulty link condition or collision) of a packet transmitted by link i 's sending node on channel c . Then, we have the following proposition.

Proposition 1 (Achievable Throughput Capacity) We obtain the throughput capacity $S_i(c)$ of link i on channel c as

$$S_i(c) = Z_i(c) \times G(p_i(c)) \times (1 - p_i(c)) \times R_i(c) \times T_{payload}, \quad (2)$$

where $R_i(c)$ is the data transmission rate, $T_{payload}$ is the average transmission time of the data payload, and let $G(p_i(c))$ model the attempt rate per idle slot for a given transmission-error probability $p_i(c)$ [14].

Proof: See Appendix A. ■

Note that the data rate $R_i(c)$ is determined by the physical link condition of link i .

We also have the following proposition regarding the relation between the link capacity and the idle-time fraction $Z_i(c)$.

Proposition 2 (Property of Link Capacity) The achievable throughput capacity $S_i(c)$ of link i on channel c is proportional to $Z_i(c)$ on a given channel c .

Proof: See Appendix B. ■

²In general, a *channel* refers either to a physical transmission medium, or to a specific radio frequency. In this paper, a *channel* refers to a specific radio frequency or band of frequencies (e.g., 11 channels in the 2.4 GHz). We use the term of *physical link* to refer to a physical wireless medium that connects a source to its sinks for clarity.

³The probability of an error seen by a packet being transmitted on the channel [14].

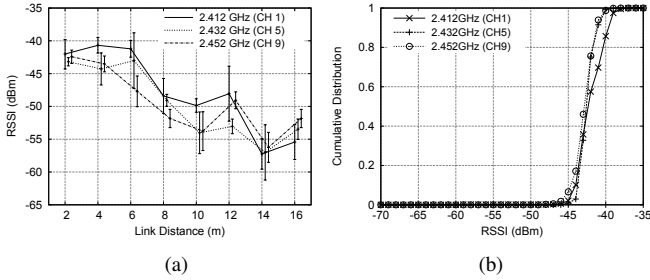


Fig. 2. Measurement results of RSSI at different link distances on different channels.

2) *Idle Period-based Channel Selection*: One way to find the best channel is to directly estimate $S_i(c)$ for $\forall c \in \mathcal{C}$. For this, we need a measure for the idle-time fraction (i.e., $Z_i(c)$) as well as link-quality information (i.e., $p_i(c)$) to quantify the link capacity $S_i(c)$ as shown in Eq. (2). However, it is challenging to accurately assess the 802.11 link quality via out-of-band sensing with a heterogeneous secondary radio interface (e.g., ZigBee and Bluetooth), since it is infeasible to probe the state of a WiFi channel using such a heterogeneous radio technology.

The key observation to address this issue is that the distance between communicating devices in mobile hotspot systems is usually very short (less than 10 m/30 ft [26], [28]), and thus, their physical link conditions are relatively good and are not severely affected by the channel, i.e., they have the nearly same average link condition for all the frequency channels. To evaluate the effect of channels, we measure the RSSI of a WiFi link’s signal using an off-the-self 4G mobile router, KWD-B2600 [25], at different link distances on three different channels.⁴ Figs. 2(a) and 2(b) plot the ranges of measured RSSI at different distances and the CDF of measured link’s signal (RSSI) at the distance of 3m, for different operating channels, respectively. The RSSI at a receiver represents its link quality since the received signal strength is the dominant factor in determining the Signal-To-Noise Ratio (SNR) [16]. We can observe that the link quality is fairly constant across channels and their average channel conditions are nearly identical regardless of the channel used for communication.

On the other hand, the MAC-layer contention due to the neighbor nodes’ transmission activity varies with the channel in use, since the underlying traffic loads across channels are independent and usually unbalanced [19]. For example, Fig. 3 shows the number of foreign (background) traffic—i.e., packets that are sent by other nodes—over time measured by a WiFi sniffer for three different channels. The result shows that the distribution of traffic loads across different channels is not uniform, implying that the MAC-layer neighboring contention on each channel is different.

As a result, as long as the quality of a wireless link is not significantly affected by the channel, the MAC-layer neighboring contention is the dominant factor that determines the achievable link capacity $S_i(c)$.

⁴The router and the client node are placed in each other’s line of sight.

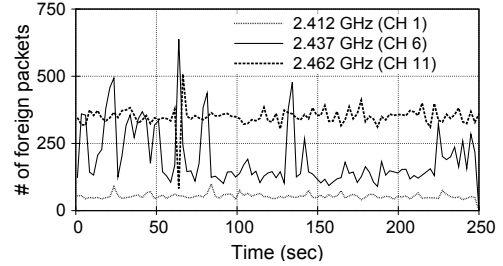


Fig. 3. Measurement results of a foreign packet (that is not sent by itself) on different channels.

Based on these observations and Proposition 2, therefore, the channel-selection problem can be restated as:

$$\mathbf{c}_i^* = \arg \max_{c \in \mathcal{C}} Z_i(c). \quad (3)$$

One can, therefore, obtain sufficient information necessary for comparing the achievable link capacity of a channel, based only on the measured idle-time period $Z_i(c)$, without the extensive measurement of capacity $S_i(c)$ for the comparison in Eq. (1).

IV. Ex2R DESIGN

Ex2R is designed on the basis of the channel-selection strategy in Section III-B. We first give an overview of the three main components of Ex2R. Then, we explain the details of its three main components: Idle-Sampler, Ex2R Core, and Channel-Transition Trigger.

A. Ex2R Overview

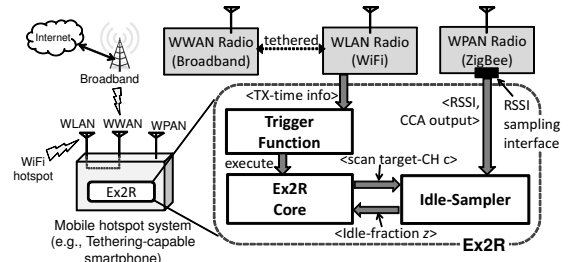


Fig. 4. Architecture of Ex2R

Fig. 4 illustrates the interactions among the components.

- *Idle-Sampler*: measures the fraction of a target channel’s idle period, by exploiting the RSSI values and CCA outputs obtained from the RSSI sampling interface of a ZigBee (Section IV-B).
- *Ex2R Core*: determines the best WiFi channel and initiates a switch to that channel (Section IV-C), using the Idle-Sampler’s idle-time fraction measurements on all candidate channels.
- *Trigger Function*: monitors the current WiFi channel condition and detects network congestion, if any. It executes Ex2R Core when the network congestion exceeds a certain level (Section IV-D).

Given below are the details of these components.

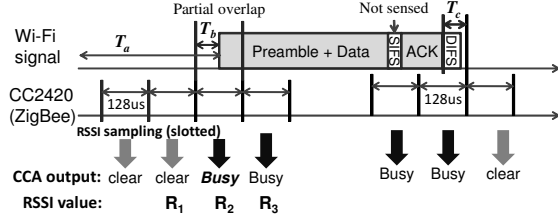


Fig. 5. Partially-overlapped sensing: a WiFi signal may be sensed partially in a ZigBee’s sensing period.

B. Idle-Sampler: Estimation of Idle Period

In Ex2R, a Zigbee interface is used to sense the WiFi channel state and measure its idle period. The 802.15.4 ZigBee radio offers a built-in mechanism for sampling the RSSI and reporting the clear-channel assessment (CCA) output. Idle-Sampler exploits the RSSI value and the CCA output to infer the 802.11 channel state and measure idle-time fraction $z(u)$.⁵

A practical challenge in the design of Ex2R is that the ZigBee radio’s sensing accuracy is limited by its coarse-grained sensing capability. The RSSI value (in dBm) reported by the ZigBee radio is typically the average received radio energy over a long sensing period, 8 symbols ($=128\mu s$ on CC2420). Therefore, any approach based on the RSSI value may be inaccurate in estimating channel occupancy, especially under-estimating the idle periods. For instance, Fig. 5 illustrates the process of sampling the RSSI values for a WiFi signal. The WiFi signal may partially overlap with the ZigBee’s long sensing period, e.g., the 3rd sensing slot in Fig. 5. Each of its measured RSSI values is the average received radio energy over an idle period, T_b , and the rest of a sensing period (i.e., $128\mu s - T_b$) for the overlapping WiFi signal. Nevertheless, the CCA output for this partially-overlapped sensing period indicates that the channel becomes busy, since the RSSI value is larger than the carrier sense threshold. This may lead to under-estimation of the idle period as $T_a - T_b$ instead of T_a , where T_a denotes the actual duration of the 802.11 channel being idle and T_b the overlapping duration in Fig. 5.

To address the partially-overlapped sensing problem for accurate measurement of an actual idle period, we study how the overlapping duration T_b relates to the RSSI value. Since the RSSI value is averaged over a sensing period of $128\mu s$, it is straightforward to show that the partially-overlapping duration T_b can be calculated by

$$T_b = 128\mu s \times (R_3 - R_2)/(R_3 - R_1), \quad (4)$$

where R_1 , R_2 , and R_3 are the RSSI values measured over three consecutive slots for a partially overlapped signal in Fig. 5; R_1 is the RSSI value for the first idle slot (i.e., noise level), R_2 is measured during a partially-overlapped sensing period, and R_3 is for the third slot that is fully occupied by a WiFi signal.

We conducted a experiment measuring the RSSI values of a partially-overlapped sensing period. We monitored WiFi

⁵We use the notation $Z(c)$ ($z(u)$) to represent the idle-time fraction of WiFi channel c (ZigBee channel u).

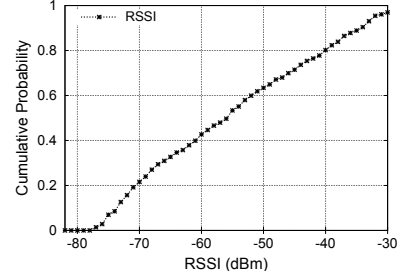


Fig. 6. The CDF of the RSSI value R_2 measured in a partially-overlapped sensing period.

packets from a single source node and recorded the time and the RSSI value of each packet by using a MicaZ mote. Fig. 6 shows the cumulative distribution of the RSSI values measured in a partially-overlapped sensing period (i.e., R_2) in Fig. 5. We can observe that R_2 is uniformly distributed within the range of $[R_1, R_3]$, meaning that the duration T_b of partially-overlapped sensing is also uniformly distributed from Eq. (4). Thus, we can obtain

$$E[T_b] = \frac{\text{sensing period of ZigBee}(=128\mu s)}{2}. \quad (5)$$

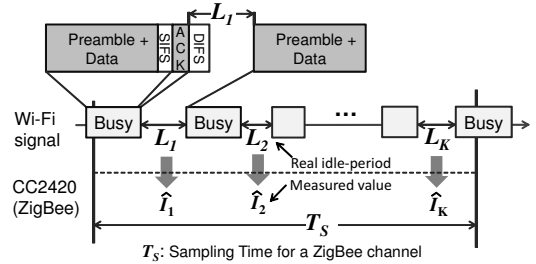


Fig. 7. Idle period sampling.

To measure the idle-time fraction $z(u)$ for ZigBee channel u , Idle-Sampler samples RSSIs and CCA outputs from a ZigBee radio every $128\mu s$ for the sampling interval T_s , as shown in Fig. 7. Let K be the total number of consecutive idle periods during the total sampling time interval. L_j is the idle-time duration between two consecutive WiFi signals, and \hat{I}_j is the measurement of L_j by the ZigBee Radio, where a measurement may involve two partially-overlapping sensing periods (e.g., T_b and T_c in Fig. 5), i.e., $L_j - 2 \times 128\mu s < \hat{I}_j \leq L_j$. Since the duration of DIFS is included in L_j , the total idle period T_z during T_s can be calculated as

$$T_z = \sum_{j=1}^K (L_j - DIFS) \approx \sum_{j=1}^K (\hat{I}_j + 2E[T_b] - DIFS). \quad (6)$$

Finally, the idle-time fraction $z(u)$ is calculated as $z(u) = T_z/T_s$.

C. Ex2R Core

Ex2R Core computes the idle-time fraction $Z(c)$ for a candidate WiFi channel $c \in C_{candidate}$ by combining idle-time fractions for four overlapping ZigBee channels measured by Idle-Sampler, where the set $C_{candidate}$ represents all target

candidate channels. Then, Ex2R Core determines the best WiFi channel and initiates a switch to that channel. First, we elaborate on how to combine the measured idle-time fractions on ZigBee channels for computing the idle-time fraction on a WiFi channel. Then, we describe how to determine the new best channel.

To compute the idle-time fraction $Z(c)$ on WiFi channel c in Eq. (3), Ex2R Core combines the idle-time fractions measured on four ZigBee channels $\mathcal{Z}_{map}(c)$. Clearly, the idle-time fractions $z(u)$ for $\forall u \in \mathcal{Z}_{map}(c)$, reflect the state of WiFi channel c . However, they may have very different values, because each $z(u)$ is affected by one or more (up to 4) different WiFi channels. For example, among the ZigBee channels overlapping with WiFi channel 2, i.e., $\mathcal{Z}_{map}(2)$, $z(15)$ is likely to be larger than $z(12)$, $z(13)$, and $z(14)$ when WiFi channel 1 is more congested than channel 2 [13]. One observation to address this overlapping problem is that during the time a WiFi node senses channel c to be idle, all the overlapping ZigBee channels $\forall u \in \mathcal{Z}_{map}(c)$ also will be sensed to be idle, but not vice versa. If there exists any ZigBee channel $\exists u_0 \in \mathcal{Z}_{map}(c)$ whose underlying power level is above the WiFi's CCA threshold (i.e., strong enough to trigger the CCA check on 802.11 nodes [15]), the WiFi node senses channel c to be busy, while other overlapping ZigBee channels $u (\neq u_0) \in \mathcal{Z}_{map}(c)$ can remain idle. For example, let us consider WiFi channel 4 that overlaps with ZigBee channels 14, 15, 16, and 17, i.e., $\mathcal{Z}_{map}(4) = \{14, 15, 16, 17\}$, in Fig. 1. When there are WiFi signals on the overlapping WiFi channel 6, only ZigBee channels 16 and 17 become busy while the other channels 14 and 15 can remain idle. In this case, the WiFi node operating on channel 4 will likely declare the channel to be busy.⁶ This implies that the idle period of a WiFi channel c is bounded by the minimum of $z(u)$ for $u \in \mathcal{Z}_{map}(c)$.

Based on this observation, we employ a simple conservative combination policy to compute $Z(c)$:

$$Z(c) = \min_{u \in \mathcal{Z}_{map}(c)} \rho(u, c) \cdot z(u), \quad (7)$$

where $\rho(u, c)$ is a weight factor that considers the impact of overlapping frequencies between ZigBee channel u and WiFi channel c . Our experimental result shows that the four idle-time fractions for a WiFi channel are almost identical. Therefore, we define:

$$\begin{cases} \rho(u, c) = 1, & \text{if } u \in \mathcal{Z}_{map}(c), \\ \rho(u, c) = 0, & \text{otherwise.} \end{cases} \quad (8)$$

We select the best WiFi channel through three steps. First, Ex2R measures the idle-time fraction $z(u)$ on channel $\forall u \in \mathcal{Z}_{map}(c)$ for $\forall c \in C_{candidate}$, with Idle-Sampler. Here, $C_{candidate}$ represents the set of all candidate channels excluding the mobile router's operating WiFi channel and its partially-overlapping neighbor WiFi channels. Next, the idle-time fraction, $Z(c)$, on WiFi channel c is computed for

⁶A recent study [15] has experimentally shown that 802.11 nodes can sense signals of 802.15.4 transmissions on an overlapping ZigBee channel, and thus, their CCA declares the channel to be busy.

Algorithm 1 Selecting the best channel c^*

procedure *findChannel()*

```

1: //  $c_0$  : WiFi channel in use
2:  $\mathcal{Z}_{scan} \leftarrow \emptyset$  //  $\mathcal{Z}$ : set of ZigBee channels to scan
3:  $C_{candidate} \leftarrow \mathcal{C} - \{c_0, c_0 \pm 1, c_0 \pm 2, c_0 \pm 3, c_0 \pm 4\}$ 
4: // Obtain a set of ZigBee channels for Idle-Sampling
5: for all  $c \in C_{candidate}$  do
6:    $\mathcal{Z}_{map}(c) \leftarrow MapWiFi\text{-}to\text{-}ZigBee(c)$ 
7:    $\mathcal{Z}_{scan} \leftarrow \mathcal{Z}_{scan} \cup \mathcal{Z}_{map}(c)$ 
8: end for
9: // Sampling : Idle scan for  $\forall u \in \mathcal{Z}_{scan}$ 
10: for all  $u \in \mathcal{Z}_{scan}$  do
11:    $z(u) \leftarrow IdleSampler(u)$ 
12: end for
13: // First step:
14: // find a ZigBee channel  $\tilde{u}(c)$  symbolizing each  $c \in C_{candidate}$ 
15: for all  $c \in C_{candidate}$  do
16:   // channel with the minimum  $z$  among 4 overlapping channels
17:    $\tilde{u}(c) \leftarrow \arg \min_{u \in \mathcal{Z}_{map}(c)} z(u)$  // Eq. (7)
18:    $\tilde{c}(\tilde{u}(c)) \leftarrow \tilde{c}(\tilde{u}(c)) \cup c$ 
19: end for
20: // Second step: find the best ZigBee channel
21:  $\tilde{u}^* = \arg \max_{\tilde{u}(c), \forall c \in C_{candidate}} z(\tilde{u}(c))$ .
22: // Final step: find the best WiFi channel
23: if  $|\tilde{c}(\tilde{u}^*)| > 1$  then
24:    $c^* \leftarrow \arg \max_{c \in \tilde{c}(\tilde{u}^*)} \sum_{k \in \mathcal{Z}_{map}(c)} z(k)$ 
25: else
26:    $c^* \leftarrow$  the single element of  $\tilde{c}(\tilde{u}^*)$ 
27: end if
28: return  $c^*$ 

```

procedure *MapWiFi-to-ZigBee(c)*

```

1:  $\mathcal{Z}_{map} \leftarrow \{c + 10, c + 11, c + 12, c + 13\}$  // from Fig. 1.
2: return  $\mathcal{Z}_{map}$ 

```

$\forall c \in C_{candidate}$ based on Eq. (7). Given the obtained $Z(c)$ for $c \in C_{candidate}$, we can then find the best channel c_i^* according to Eq. (3), i.e., the channel with the maximum of $Z(c)$ is selected. If two or more channels have the same idle-time fraction $Z(c)$, Ex2R Core compares the sum of all four ZigBee idle-time fractions $z(u)$ of these WiFi channels, and chooses the channel with the maximum sum as the best channel. For example, assume that WiFi channels 10 and 11 (i.e., $Z(10)$ and $Z(11)$) have the same idle-time fraction ($=z(22)$) according to Eq. (7). In this case, Ex2R Core compares their sums of all idle-time fractions, i.e., $\sum_{u \in \mathcal{Z}_{map}(10)} z(u)$ and $\sum_{u \in \mathcal{Z}_{map}(11)} z(u)$, and selects the channel with a larger sum as the best channel c_i^* . Finally, Ex2R Core initiates a switch to the selected channel. Algorithm 1 details this selection process.

D. Channel-Transition Trigger

The objective of the trigger function is to diagnose the network condition, or detect network congestion, if any, and notify the diagnosis result to the mobile router, so as to initiate Ex2R Core for a channel switch. Ex2R trigger function runs on the WiFi interface of a mobile router and monitors the state of outgoing transmissions. It keeps track of the time taken to successfully transmit an 802.11 packet and uses the time as

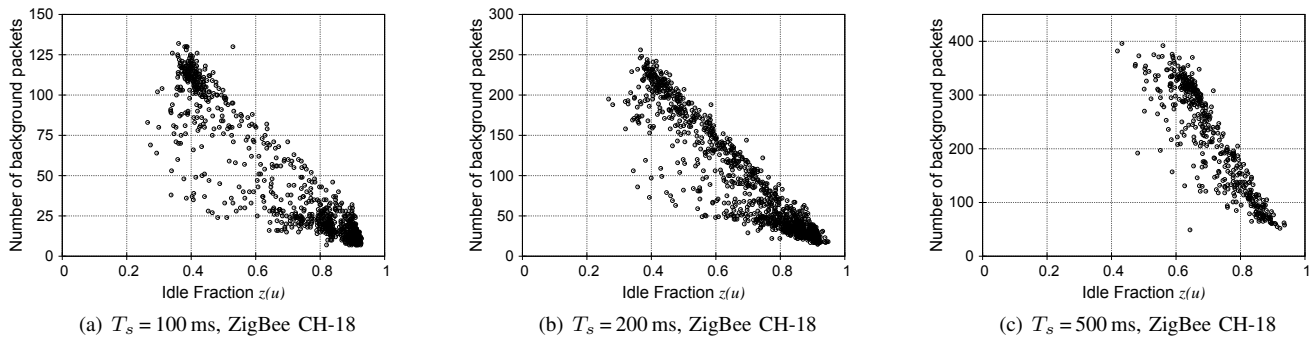


Fig. 8. Correlation between idle-time fraction and MAC interference, measured in WiFi channel 6 and ZigBee channel 18: (a) $T_s = 100$ ms, (b) $T_s = 200$ ms, and (c) $T_s = 500$ ms.

a sample sequence for congestion diagnosis.⁷ Let s_j denote the time taken for successfully transmitting the j -th packet. Under the normal condition (i.e., uncongested condition), s_j is stationary and has a certain upper bound \mathcal{U} which is a design parameter. When network congestion occurs, s_j will suddenly have a large value, and will likely exceed the bound \mathcal{U} . To quantify how long the congestion lasts, we adopt the CUSUM (CUmulative SUMmary) algorithm [10], which has been widely adopted for the detection of state changes.

We define the CUSUM change detection filter g_j as

$$g_j = \max(0, g_{j-1} + \hat{s}_j), \quad (9)$$

$$g_0 = 0,$$

where $\hat{s}_j = s_j - \mathcal{U}$ such that it has a negative mean during normal operation. When the network becomes congested, \hat{s}_j will suddenly become a large positive number due to the increased value of s_j . g_j in Eq. (9) is interpreted as the maximum cumulative increment until sequence j . If a change in the network state occurs, the magnitude of g_j tends to increase continuously. In other words, a large g_j is a strong indication of network congestion. Thus, an additional CUSUM test parameter, θ_H , called *alarm threshold*, is defined, i.e., the change detection filter sends an alarm when $g_j > \theta_H$. In other words, Trigger Function executes Ex2R Core when $g_j > \theta_H$.

V. EVALUATION

In this section, we implement Ex2R's dynamic channel reconfiguration, and then evaluate its efficiency on the prototype implementation in a WiFi hotspot testbed.

A. Implementation of Dynamic Channel Reconfiguration

We have implemented a prototype of Ex2R using MicaZ/TinyOS 2.1 [11] and open-source WiFi driver [27] on a Linux platform. The MicaZ mote is equipped with an 802.15.4 standard-compliant TI CC2420 radio [29] operating in the 2.4 GHz ISM band. The prototype consists of three main Ex2R components: *Idle-Sampler*, *Ex2R Core*, and *Trigger Function*. *Idle-Sampler* is implemented in TinyOS 2.1 [11] and runs on MicaZ motes. The other two components run on the Linux

platform. Ex2R Core is implemented in Java. Trigger Function is implemented in an open source driver, i.e., MadWiFi driver [27]. Ex2R Core communicates with Idle-Sampler via serial port communication, and with Trigger Function via a shared file in Linux.

We also implemented Ex2R's dynamic channel reconfiguration. Once Trigger Function on the MadWiFi driver diagnosed the current WiFi hotspot channel to be congested, it periodically reads a shared configuration file to obtain the information on the best candidate channel updated by Ex2R Core. If the thus-obtained channel is different from the one currently in use, Ex2R initiates a channel switch.

B. Measurement Accuracy

1) *Evaluation of Idle Sampler*: Idle-Sampler is a critical component based upon which Ex2R determines the best WiFi channel. So, we first evaluate the efficiency of Idle-Sampler, and show that the idle-time fraction accurately estimates the degree of MAC interference. We also study the impact of sampling interval T_s for measuring the idle-time fraction $z(u)$ on ZigBee channel u .

We conduct measurements and explore the relation between the idle-time fraction and the amount of foreign (background) traffic. We monitored the WiFi packets transmitted by WiFi APs and nodes in the vicinity by using both WiFi packet sniffer and Idle-Sampler. The WiFi packet sniffer recorded the arrival time of 802.11 packets, and Idle-Sampler measured the idle-time fraction and reported it to Ex2R Core every T_s seconds. Ex2R Core recorded the time of each report. We merged the two separate records. We sensed WiFi channel 6 and two overlapping ZigBee channels 18 and 16.

Fig. 8 plots the measurement results for ZigBee channel 18 with three different sampling intervals, $T_s = 100$ ms, 200 ms, and 500 ms. One can observe a linear inverse proportional relationship between the idle-time fraction and the degree of MAC-layer interference, indicating that a large idle-time fraction leads to higher achievable capacity of channel c , and vice versa. The results demonstrate the accuracy of the idle-fraction measurement as well as the efficiency of Ex2R, i.e., our simple method can provide high accuracy in predicting achievable hotspot capacity.

⁷Note that the time spent for a successful transmission reflects both current wireless channel and background traffic conditions: the less time it takes, the better channel condition, and vice versa.

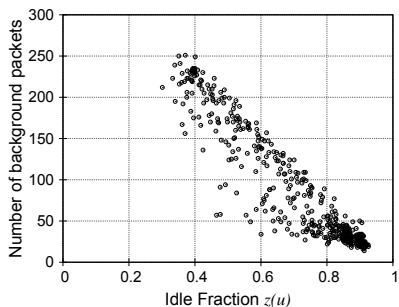


Fig. 9. Correlation between idle-time fraction and MAC interference, measured in WiFi channel 6 and ZigBee channel 16: $T_s = 200$ ms.

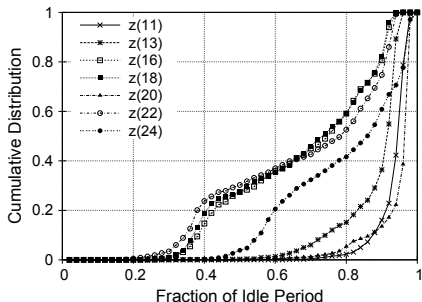


Fig. 10. CDF of idle-time fractions on ZigBee channels measured in our testbed.

We also observe the impact of sampling duration T_s of Idle-Sampler on prediction accuracy. When the sampling duration is small, i.e., $T_s = 100$ ms in Fig. 8(a), the dispersion of samples is relatively high. When we adopt longer sampling durations $T_s = 200$ ms and 500 ms, the dispersion of samples becomes narrower, as shown in Figs. 8(b) and 8(c), where a narrower dispersion implies higher prediction accuracy. However, there is a tradeoff between sampling interval (T_s) and prediction accuracy. A long sampling period increases the total time required to determine the best channel.

Fig. 9 plots the measurement results for ZigBee channel 16 with the sampling duration of $T_s = 200$. Note that ZigBee channel 16 resides at the edge of WiFi channel 6 and ZigBee channel 18 in Fig. 8 is at the middle. The results also show a close correlation between the idle-time fraction and the degree of MAC-layer interference as shown in Fig. 8. This means that the idle-time fractions measured on all overlapped ZigBee channels are equally useful in inferring the capacity of a WiFi channel, regardless of where a ZigBee channel overlaps with the WiFi channel. This observation justifies our equal-gain combination policy in Eq. (7); we use the same weight factor, $\rho(u, c)$ in Eq. (8) for all the overlapping WiFi channels $u \in \mathcal{Z}_{map}(c)$.

2) *Impact of Partial Channel Overlap*: Next, we study the impact of overlapping WiFi channels on the measurement of idle-time fraction. Fig. 10 presents the cumulative distribution of idle-time fraction $z(u)$ on ZigBee channels $u = \{11, 13, 16, 18, 20, 22, 24\}$ measured in our testbed (at a fixed location) for 10 seconds. The WiFi channel environment nearby our testbed is shown in Table I, where the condition of nearby WiFi networks directly affects the measurement results. The

large numbers of APs on channels 1, 6, and 11 in Table I mostly belong to a university WiFi network, where our CSE department offices are covered by the University WiFi network which is active on 802.11 channels 1, 6, and 11. To study the impact of a partial channel overlap of WiFi channels, we operated APs on channels 2 and 10.

TABLE I
THE NUMBER OF APs IN THE VICINITY OF OUR TESTBED.

channel	1	2	3	4	5	6	7	8	9	10	11
number	11	1	0	0	0	12	0	0	0	1	9

From Fig. 10, we observe three clusters; (i) lower region: ZigBee channels 20, 11, and 13, (ii) intermediate region: channel 24, (iii) upper region: channels 16, 18, and 22. We make the following observations based from this figure: those ZigBee channels overlapping with the same WiFi channel can have different distributions of idle-time fractions when the WiFi channel is partially overlapping with other WiFi channels, but they have the same distribution when the WiFi channel does not overlap with other WiFi channels. In our experiment, ZigBee channels 16 and 18 overlap only with WiFi channel 6. So, they have an identical distribution. On the other hands, ZigBee channels 11 and 13 (22 and 24) overlap with different WiFi channels 1 and 2 (10 and 11), even though they overlapped with WiFi channel 1 (11) together, respectively. As a result, ZigBee channels 11 and 13 (22 and 24) have different idle-time fraction distributions. Especially, the ZigBee channels overlapping with more WiFi channels will likely have lower idle-time fractions. For example, ZigBee channel 13 (22)—which overlaps with two WiFi channels 1 and 2 (10 and 11)—has lower distribution than that of ZigBee channel 11 (24)—which overlaps with only one channel 1 (2). This observation provides a reason for our conservative combination policy for computing idle-time fraction $Z(c)$ of a WiFi channel shown in Eq. (7).

C. Effectiveness of Dynamic Channel Reconfiguration

Next, we evaluate the performance of Ex2R's dynamic channel reconfiguration.

First, we show via real experiments the usability of Trigger Function's network diagnosis introduced in Section IV-D. Note that timely and accurate diagnosis of network congestion on the current hotspot channel is essential to intelligent dynamic reconfiguration.

Trigger Function has two design parameters: \mathcal{U} , the upper bound in case of normal operation, and θ_H , the decision threshold. The choice of these parameters can affect the detection of network congestion. We use different diagnosis thresholds of Ex2R's Trigger Function in order to study the detection sensitivity. We tested three upper bound parameters \mathcal{U} for $\mathcal{U} = 3, 5, \text{ and } 10$ ms.

Figure 11 plots the variation of transmission time during the measurement period, and the detection results. We measured the time taken for a successfully transmission by an AP in our testbed. We added a simple measurement routine in the MadWifi driver. The time taken for a successfully transmission

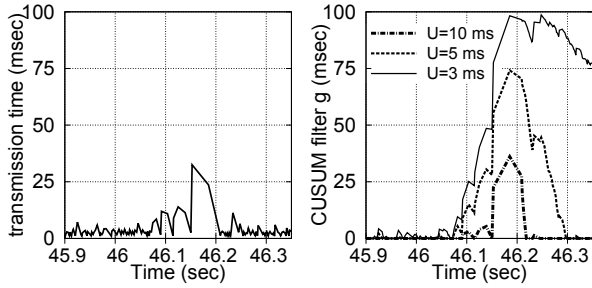


Fig. 11. Measurements of transmission time (left), i.e., the time taken for a successfully transmission, and the CUSUM test statistics in Ex2R Trigger (right). $(\mathcal{U}, \theta_H) = (5 \text{ ms}, 50 \text{ ms})$ is used in the evaluation.

is usually found to be less than 2 ms under normal condition. But, when the network becomes congested, the transmission time begins to increase beyond 5 ms (as shown at around 46.1 seconds in Figure 11).

The accumulative effect of different parameter values \mathcal{U} is also clearly shown in the figure. In the case of small \mathcal{U} , i.e., $\mathcal{U} = 3 \text{ ms}$, the CUSUM value q (sequential cumulative sum) increases readily. In the case of $\mathcal{U} = 10 \text{ ms}$, the CUSUM filter reacts only to high transmission delays, and increases conservatively. Trigger Function initiates Ex2R Core when the CUSUM value q becomes larger than θ_H . Therefore, the greater the value θ_H (and smaller the value \mathcal{U}), the less frequent execution of Ex2R, and vice versa. We used $(\mathcal{U}, \theta_H) = (5 \text{ ms}, 50 \text{ ms})$ in our experiments, but the selection of these design parameters can be adapted to the application QoS requirements or the link capacity of 3G/4G broadband connection of a mobile router.

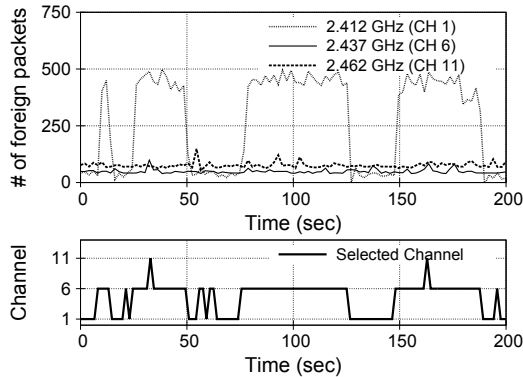


Fig. 12. Accuracy of Ex2R's channel selection.

Next, we evaluate the accuracy of the proposed channel selection strategy of Ex2R. We monitored WiFi channels 1, 6, and 11 by using WiFi sniffers and measured the amount of foreign (background) traffic. We execute the prototype Ex2R every 2 seconds and pick the best channel. For the experiment, we generated on/off background traffic on channel 1 by using a laptop. Fig. 12 shows the measured amount of foreign traffic and the best channel selected by Ex2R as a function of time. Ex2R is shown to accurately select the least congested channel during the measurement period.

We also conducted experiments to capture the benefits

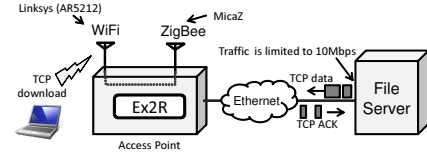


Fig. 13. Experiment topology.

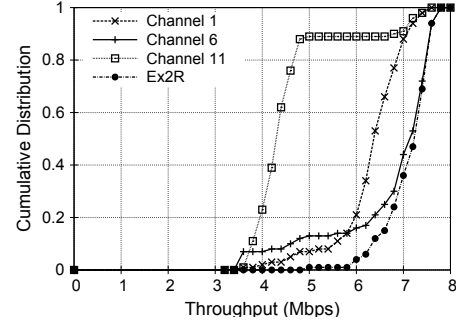


Fig. 14. Performance comparison: CDF of download throughput of static channel assignment and Ex2R's dynamic channel reconfiguration.

of dynamic reconfiguration using the Ex2R prototype. We consider a mobile hotspot with the Ex2R AP and one laptop node, as shown in Fig. 13. The WiFi channel condition in our testbed is shown in Table I. The broadband WAN connection of a mobile hotspot router is emulated with the traffic control module at a file server. The download traffic is limited to 10 Mbps.

We compare the performance of static channel assignment with that of Ex2R's dynamic reconfiguration. The laptop client downloads a file using TCP from the file server. For static channel assignments, the channel of the testbed hotspot is fixed for three different channels 1, 6, and 11. We repeated experiments 100 times and measured the download throughput for four cases: Ex2R's dynamic reconfiguration, and three static channels 1, 6, and 11.

Fig. 14 plots the CDF of download throughput of static channel assignment and Ex2R's dynamic channel assignment. The result shows that Ex2R's dynamic reconfiguration outperforms the static channel assignment, demonstrating the efficiency of Ex2R.

VI. CONCLUSION

In this paper, we proposed a new, practical channel assignment scheme, *Ex2R*, that harnesses secondary low-power ZigBee radios accompanied in mobile hotspot systems. We have shown analytically and experimentally that a channel's idle-time fraction accurately reflects available hotspot capacity. We have also addressed the problem of ZigBee radio's limited sensing capability, and developed an accurate channel measurement technique. Our evaluation results have shown that the proposed Ex2R accurately and dynamically selects the best channel, thus improving the hotspot performance significantly. We plan to apply Ex2R for a wide range of scenarios, including moving nodes, rate adaptation, and AP selection when 3G/4G smartphones are tethered as mobile hotspot routers.

ACKNOWLEDGEMENTS

We thank our shepherd Mooi Choo Chuah and the anonymous reviewers for their insightful comments and discussion.

APPENDIX A

PROOF OF PROPOSITION 1

Proof: The available throughput capacity $S_i(c)$ of link i on channel c is given by

$$S_i(c) = X_i(c) \times (1 - p_i(c)) \times R_i(c) \times \frac{T_{payload}}{T_{tx}}, \quad (10)$$

where $T_{tx} = T_{Header} + T_{payload} + DIFS + SIFS + ACK$ is the average overall transmission time (in number of time slots) for a packet, including PHY and MAC headers, data payload and ACK, as well as DCF Inter-frame Space (DIFS) and Shortest Inter-frame Space (SIFS). Since we are only interested in the *maximum* available capacity of link i , we assume that the node always has backlogged packets to transmit.

In 802.11 DCF, nodes attempt to transmit only during an idle slot. In other words, the sender node counts down its back-off timer to transmit a packet only during idle periods, and it defers the count-down process whenever the channel is sensed busy. Thus, if we let $G(p)$ model the attempt rate per idle slot for a given transmission-error probability p [14], then the normalized self airtime $X_i(c)$ is proportional to the normalized idle-time $Z_i(c)$ and can be expressed as:

$$X_i(c) = Z_i(c) \times G(p_i(c)) \times T_{tx}. \quad (11)$$

Therefore, we obtain the throughput capacity $S_i(c)$ of link i on channel c as

$$S_i(c) = Z_i(c) \times G(p_i(c)) \times (1 - p_i(c)) \times R_i(c) \times T_{payload}. \quad (12)$$

APPENDIX B

PROOF OF PROPOSITION 2

Proof: Let us define $e_i(c)$, the frame error rate (FER) due to PHY-layer channel error (i.e., channel quality), and $b_i(c)$, the probability that the channel becomes busy due to the activity of other nodes on channel c . Let $\tau_i(c)$ denote $G(p_i(c))$, the attempt rate per idle slot for a given transmission-error probability p_i , where $G(p_i(c))$ is a function of p_i given in [14]. The probability $b_i(c)$ can be expressed as $b_i(c) = 1 - \prod_{k=1}^{N_c} (1 - \tau_k(c))$, where N_c denotes the number of neighbor nodes in the carrier sensing range of the target node. Then, the conditional packet transmission error probability $p_i(c)$ is given by $p_i(c) = 1 - (1 - e_i(c))(1 - b_i(c))$, where $0 \leq p_i(c) \leq 1$. Let $P_X(c)$, $P_Y(c)$, and $P_Z(c)$ denote the probabilities of being in state *self channel*, *busy channel*, and *idle channel*, respectively, when the link is on channel c . Then, we can obtain these probabilities as $P_X(c) = \tau_i(c)$, $P_Y(c) = (1 - \tau_i(c))b_i(c)$, and $P_Z(c) = (1 - \tau_i(c))(1 - b_i(c))$. Then, $X_i(c)$, $Y_i(c)$, and $Z_i(c)$ can be expressed as

$$\begin{aligned} X_i(c) &= P_X T_X / (P_X(c)T_X + P_Y(c)T_Y + P_Z(c)T_Z), \\ Y_i(c) &= P_Y T_Y / (P_X(c)T_X + P_Y(c)T_Y + P_Z(c)T_Z), \\ Z_i(c) &= 1 - X_i(c) - Y_i(c), \end{aligned} \quad (13)$$

where T_X , T_Y , and T_Z denote the time periods which the link remains in each of the three states, respectively.

Based on Eq. (13), one can easily show that the available throughput capacity $S_i(c)$ in Eq. (12) can be expressed as a function $f(b_i(c), e_i(c))$ of $b_i(c)$ and $e_i(c)$. This is because $\tau_i(c)$ is a function of $p_i(c)$, that is, a function of $b_i(c)$ and $e_i(c)$, and $Z_i(c)$ can be expressed as $(1 - \tau_i(c))(1 - b_i(c))$. For a given frame error rate e_i , $S_i(c)$ is inversely proportional to the probability $b_i(c)$. Then, one can easily see that $Z_i(c)$ is inversely proportional to $b_i(c)$ based on Eq. (13). Since $Z_i(c)$ is a dependent variable of $b_i(c)$, the achievable throughput capacity $S_i(c)$ of link i on channel c is proportional to $Z_i(c)$. ■

REFERENCES

- [1] IEEE 802.11 WG. IEEE Std 802.11-2007 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-1999, 2007.
- [2] IEEE Computer Society. Local and metropolitan area networks - Specific requirements Part 15.1: Wireless medium access control and physical layer specifications for wireless personal area networks (WPANs), 2003.
- [3] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, Wireless wakeups revisited: Energy management for voip over wi-fi smartphones. In *Proc. ACM/USENIX Mobisys*, June 2007.
- [4] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, Self-Management in Chaotic Wireless Deployments. In *Proc. ACM MOBICOM*, Sep. 2005.
- [5] G. Ananthanarayana, and I. Stoica, Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals. In *Proc. ACM/USENIX Mobisys*, 2009.
- [6] K. Chebroli, and A. Dhekne, Esense: Communication through Energy Sensing. In *Proc. ACM MOBICOM*, Sep. 2009.
- [7] C. F. Chiasserini and R. Rao, Combining Paging with Dynamic Power Management. In *Proc. IEEE INFOCOM'01*, pages 12.19, April 2001.
- [8] Y. Gao, J. C. S. Lui, and D. Chiu, Determining the End-to-end Throughput Capacity in Multi-hop Networks: Methodology and Applications. In *Proc. SIGMETRICS/Performance 2006*, June 2006.
- [9] J. Geier, Assigning 802.11b Access Point Channels. In *WiFi planet*, Feb. 2002.
- [10] F. Gustafsson, Adaptive filtering and change detection. *John Wiley & Sons, Ltd*, 2000.
- [11] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, System architecture directions for network sensors. In *Proc. ASPLOS-IX*, Nov. 2000.
- [12] M. Kershaw, GNU WiFi Sniffer Kismet. <http://www.kismetwireless.net>.
- [13] J. Ko, T. Gao, A. Terzis, Empirical Study of a Medical Sensor Application in an Urban Emergency Department. In *Proc. BodyNets'09*, 2009.
- [14] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, New Insights from A Fixed-point Analysis of Single Cell IEEE 802.11 WLANs. *IEEE/ACM Trans. Netw.*, 15(3):588–601, June 2007.
- [15] C.-M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, Surviving Wi-Fi Interference in Low Power ZigBee Networks. In *Proc. IEEE SenSys'10*, Nov. 2010.
- [16] M.-H. Lu, P. Steenkiste, and T. Chen, Design, Implementation and Evaluation of an Efficient Opportunistic Retransmission Protocol. In *Proc. ACM MOBICOM*, Sep. 2009.
- [17] A. Mishra, V. Shrivastava, D. Agarwal, and S. Banerjee, Distributed Channel Management in Uncoordinated Wireless Environments. In *Proc. ACM MOBICOM*, Sep. 2006.
- [18] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, A General Model of Wireless Interference. In *Proc. ACM MobiCom*, Sep 2007.
- [19] E. Rozner, Y. Mehta, A. Akella, and L. Qiu, Traffic-Aware Channel Assignment in Enterprise Wireless LANs. In *Proc. IEEE ICNP*, Oct 2007.
- [20] E. Shih, P. Bahl, and M. J. Sinclair, Wake on wireless: An event driven energy saving strategy for battery operated devices. In *Proc. ACM MOBICOM*, Sep. 2002.
- [21] G. Strachan, Is 4G Better than 3G?, *Discovery News*, <http://news.discovery.com/tech/is-4g-better-than-3g.html>.
- [22] H. Wu, K. Tan, J. Liu, and Y. Zhang, Footprint: Cellular Assisted Wi-Fi AP Discovery on Mobile Phones for Energy Saving. In *Proc. ACM WinTech (Mobicom workshop)*, Sep. 2009.
- [23] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, Zifi: Wireless LAN Discovery via ZigBee Interference Signatures. In *Proc. ACM MOBICOM*, Sep. 2010.
- [24] AirLink, AR660W3G Wireless N 300 3G/3.5G Mobile Router. <http://http://airlink101.com/products/ar660w3g.php>.
- [25] KT WiBro 4G egg, BWD-2600 Wibro 4G mobile router. <http://wibro.olleh.com>.
- [26] Mobile Internet Comparison. <http://www.mobile-broadband-reviews.com/broadband-card-comparison.html>.
- [27] Multiband Atheros Driver For WiFi, <http://madwifi-project.org/>.
- [28] Novatel MiFi. <http://www.novatelwireless.com/>.
- [29] Texas Instruments Inc., CC2420: Single-Chip 2.4 GHz IEEE 802.15.4 Compliant RF Transceiver.
- [30] Verizon Wireless Mobile Broadband. <http://www.verizonwireless.com/b2c/mobilebroadband/>.