EasyPass: Combating IoT Delay with Multiple Access Wireless Side Channels

Haoyang Lu
University of Pittsburgh
haoyanglu@pitt.edu

Ruirong Chen
University of Pittsburgh
ruc28@pitt.edu

Wei Gao
University of Pittsburgh
weigao@pitt.edu

ABSTRACT

Many IoT applications have stringent requirements on wireless transmission delay, but have to compete for channel access with other wireless traffic. Traditional techniques enable multiple access to wireless channels, but yield severe delay when the channel is congested. In this paper, we present *EasyPass*, a wireless PHY technique that allows multiple IoT devices to simultaneously transmit data over a congested wireless link without being delayed. The key idea of EasyPass is to exploit the excessive SNR margin in a wireless channel as a dedicated side channel for IoT traffic, and allow multiple access to the side channel by separating signals from different transmitters on the air. We implemented EasyPass on software-defined radio platforms. Experiment results demonstrate that EasyPass reduces the data transmission delay in congested IoT networks by 90%, but provides a throughput up to 2.5 Mbps over a narrowband 20MHz wireless link that can be accessed by more than 100 IoT devices.

CCS CONCEPTS

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

KEYWORDS

Internet of Things, Delay, Channel Contention, Wireless Side Channel, Multiple Access, SNR Margin

ACM Reference Format:
Haoyang Lu, Ruirong Chen, and Wei Gao. 2019. EasyPass: Combating IoT Delay with Multiple Access Wireless Side Channels. In *The 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '19), December 9–12, 2019, Orlando, FL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3359989.3365421

1 INTRODUCTION

The Internet of Things (IoT) consists of heterogeneous sensing devices that interact with and collect data from the physical world. Delay of transmitting sensory data in IoT, in many cases, should be minimum for prompt system response. For example, real-time data transfer is required in smart factories for precise manufacturing control [58]. Timely feedback from distributed sensors is also the key enabler for smart home and smart city applications [5, 28, 34].

Unfortunately, IoT traffic can be easily delayed in practice, when competing with other data traffic for wireless channel access. Current wireless MAC protocols based on CSMA [7, 19, 29, 37, 39] are usually incapable of minimizing such delay due to their transmission backoff. Cross-technology communication improves the MAC efficiency with explicit device coordination [13, 30, 35], but produces significant coordination overhead in a large IoT network and results in extra delay. Exploiting new spectrum such as the mm-wave band [53, 54, 65], on the other hand, has constraints on signal propagation and may not fit to many IoT contexts.

Instead, a better alternative is to avoid delay at the PHY layer by transmitting delay-sensitive IoT traffic through a dedicated *wireless side channel*, which operates over the spectrum being occupied but under-utilized by an existing wireless channel[1] due to its in-band *SNR margin*: the main channel's SNR is usually higher than the SNR required to support the link data rate being used, due to discrete link rates and conservative rate adaptation. Building such a side channel is feasible due to the unique feature of IoT, in which data traffic from each device is highly bursty but usually low in volume. For example, sending a remote control message only costs a few bytes [33], and most embedded sensors (e.g., light, temperature and motion sensors) create kilobytes of data traffic every second [55, 66]. This feature allows transmitting IoT data with a moderate amount of extra throughput being exploited from the main channel's SNR margin. These transmissions never wait for main channel's idleness, and hence result in minimum delay over congested wireless links.

The major challenge of using a wireless side channel in IoT, however, is to support multiple access to the side channel from many IoT devices. Most of existing research builds the side channel over WiFi [15, 38] and ZigBee [63] by alternating the main channel characteristics [10, 21, 25, 27, 41, 56], but limits the side channel to be operated by the same device that operates the main channel. The RF signal of the side channel, when being transmitted from multiple devices, will be appended to the main channel signal as *extra interference* and difficult to be correctly demodulated. The dynamic conditions of main channel may further result in intermittent availability or fluctuating amount of SNR margin, making it hard to allocate the SNR margin among multiple IoT devices.

In this paper, we present *EasyPass*, a new wireless PHY technique that realizes multiple access wireless side channels. EasyPass allows precise side channel demodulation via controlled asynchrony between transmissions of the side channel and the main channel. A side channel transmitter holds its transmission of each frame until the end of a main channel frame's preamble, which hence is not mixed with any side channel signal. This preamble, then, is used by the side channel to estimate the main channel's impulse response

[1]The existing wireless channel, whose spectrum is utilized to build the side channel, is referred as the "main channel" in the rest of this paper.

CoNEXT '19, December 9–12, 2019, Orlando, FL, USA

ACM ISBN 978-1-4503-6998-5/19/12…$15.00
https://doi.org/10.1145/3359989.3365421

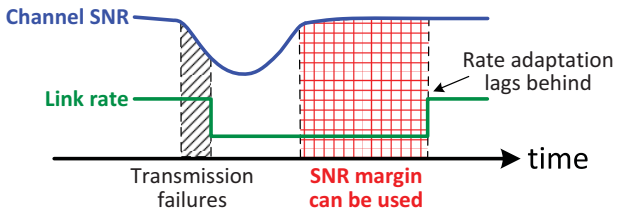


Figure 1: Illustration of usable SNR margin

and remove the main channel signal from the mixed signal being received. Based on such asynchronous transmissions, EasyPass achieves multiple accessibility in the side channel by exploiting the main channel's OFDM-based wireless PHY[2]: it applies extra interference from each side channel transmitter to a different set of main channel's OFDM subcarriers.

To minimize the impact of side channel operations to the main channel, the amount of extra interference from side channel transmitters must be within the available SNR margin. The side channel signal, hence, has to be transmitted with much weaker power, which lowers its effective SNR[3] and results in demodulation errors. To address this challenge, EasyPass adaptively controls the number of OFDM subcarriers that the side channel operates, so that stronger power can be applied to each subcarrier for correct demodulation. In practice, EasyPass provides a side channel throughput up to 2.5 Mbps over a 20MHz main channel.

To our best knowledge, EasyPass is the first that allows multiple IoT devices to simultaneously transmit over a congested wireless link without being delayed or interrupted. Its major contributions are as follows:

- EasyPass is highly *adaptive*. Side channel's operations always adapt to the available SNR margin and transmission patterns in the main channel, ensuring reliable data transmission even with strong noise or channel distortions. EasyPass well adapts to various IoT dynamics in channel conditions, device population and traffic demands, to flexibly allocate the available SNR margin among multiple IoT devices.
- EasyPass is *lightweight*. A side channel in EasyPass uses the same channel detection, modulation and coding techniques as those in commodity wireless devices. It does not require any explicit coordination with the main channel or decode any main channel's transmitted data. No explicit coordination is required among side channel users, either. Hence, EasyPass incurs negligible computation and communication overhead.
- We implemented EasyPass over both USRP and WARP devices that operate on a 2.4GHz band. Our evaluation shows that EasyPass can reduce the transmission delay of >100 IoT devices by more than 90% over a highly congested 20MHz link, and also suppresses the delay jitter with heterogeneous IoT traffic patterns. Further, we also applied EasyPass to practical smart home IoT scenarios, and demonstrates that EasyPass can significantly improve their performance.

[2]OFDM is the PHY modulation method in mainstream IoT wireless techniques such as WiFi and narrowband IoT (NB-IoT) [45], and is the foundation of next-generation IoT wireless access such as 802.11ax [4] and 5G [57].
[3]From the side channel's perspective, the main channel signal is also considered as noise.

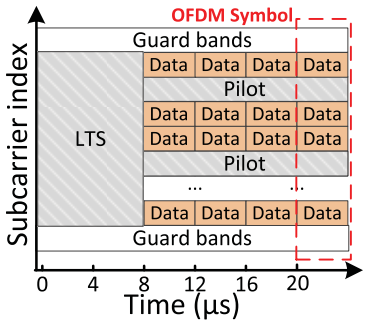


Figure 2: OFDM PHY

2 MOTIVATION & BACKGROUND

In this section, we motivate the design of EasyPass by showing the availability of SNR margin in commodity wireless networks, and explain how today's wireless PHY works to help understand the design of EasyPass.

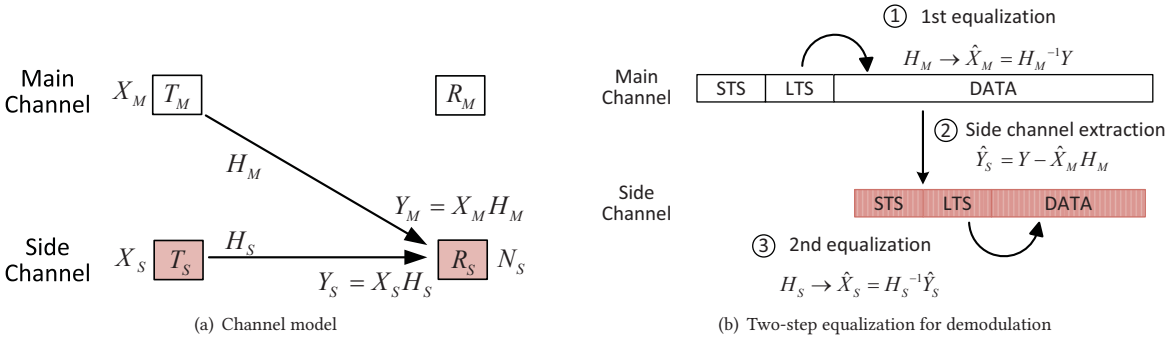
2.1 SNR Margin

We define the SNR margin as the difference between the instantaneous channel SNR and the minimum channel SNR required to support the current link data rate. Such SNR margin widely exists in current wireless networks, because of their discrete choices of data rates. For example, Table 1 lists the minimum SNR required by the IEEE 802.11 standard with a 1% packet error rate [15, 44]. For example, if the channel SNR is 14 dB, the channel can only operate at 24 Mbps, resulting in a 2 dB SNR margin.

Data rate (Mbps)	Min. SNR in IEEE std. (dB)	Avg. SNR margin in our experiment (dB)
6 (BPSK 1/2)	3.5	4.3
9 (BPSK 3/4)	4.5	4.9
12 (QPSK 1/2)	5.5	5.7
18 (QPSK 3/4)	9.5	5.2
24 (16QAM 1/2)	12.0	6.4
36 (16QAM 3/4)	15.5	6.8
48 (64QAM 2/3)	20.0	7.1
54 (64QAM 3/4)	21.0	7.5

Table 1: WiFi data rates and SNR margins

In practice, the SNR margin is much higher due to conservative decisions of wireless rate adaptation, which always lag behind the actual fluctuation of channel conditions. For example, most rate adaptation algorithms decide the link rate based on past statistics of packet delivery [8]. The link rate will only increase after a significant number of consecutively successful frame transmissions, but will immediately drop with any two consecutive transmission failures. To investigate such SNR margin, we conducted an experiment by streaming different contents of HBO Go video [3] from a Netgear N300 router to three MacBook Air laptops, and then monitor the reception of >10k WiFi frames in various practical channel conditions. The laptops are evenly placed at different locations 10 meters away from the router, with a mutual distance of 5 meters. Each laptop has line-of-sight connection to the router without blockage. Results in Table 1 show that the average SNR margin is at least 4



(a) Channel model (b) Two-step equalization for demodulation

Figure 3: Asynchronous data transmissions in EasyPass

dB, and would increase to >7 dB with higher link rates. We also observed that the SNR margin is higher than 5 dB for >80% of WiFi frames, when the link rate varies between 6 Mbps and 54 Mbps.

As shown in Figure 1, such SNR margin could be opportunistically used to transmit extra data when the channel condition improves but rate adaptation lags behind for a few frames [2, 32, 60], especially if the main channel's link rate is restricted due to previous transmission failures. The key to avoid impacting the main channel includes 1) precisely controlling the amount of SNR margin being used, and 2) fast utilization of the intermittently available SNR margin. We will present technical details on these two aspects in Section 4 and Section 5, respectively. In Section 7.2, we will also experimentally investigate how EasyPass can timely adapt to the fluctuation of the available SNR margin over time, in various practical wireless network scenarios.

2.2 Wireless PHY Primer

OFDM. EasyPass builds on the OFDM-based PHY in the main channel. As shown in Figure 2, OFDM divides the spectrum into evenly spaced subbands, called subcarriers, and data is correspondingly split into parallel streams. Since the frequency bands of OFDM subcarriers are orthogonal, side channel users in EasyPass have the minimum impact on each other when operating over different sets of subcarriers.

Channel equalization. The channel equalizer is critical for a wireless receiver to recover the received signal from multipath fading. For each received data frame, it estimates the channel impulse response by measuring the distortion of certain reference signal in the frame, and then compensates such distortion by applying the inverse of this channel response. For example, WiFi uses Long Training Sequence (LTS) in the frame preamble as the reference signal, which consists of two OFDM symbols and can be used for channel estimation with various estimators. The commonly used least-square estimator estimates the channel response as $H = X^{-1}Y$, where X and Y are the LTS signal sent and received, respectively. EasyPass utilizes this reference signal to estimate the main channel response, based on which the side channel signal is extracted.

3 EASYPASS DESIGN

The design of EasyPass builds on asymmetry of common IoT configurations, which consist of many weak sensing devices as transmitters and few strong data processing units as receivers. For example, a surveillance camera system has many cameras that send their data back to a single base station. Similarly, data obtained by sensors in a smart factory will be processed at the factory's central computer. EasyPass offloads the side channel operations to the receiver as much as possible, hence minimizing the computation and communication overhead at resource-constrained transmitters.

For a mixed channel transmission model in Figure 3(a), a side channel Tx transmits data frames in the same structure as the main channel Tx uses. Its only side channel operation is to overhear the main channel and hold its transmission of each frame until the end of LTS preamble in a main channel's data frame. Then, as shown in Figure 3(b), the side channel Rx estimates the main channel's impulse response in each subcarrier before the side channel frame starts, and uses this knowledge to remove the main channel signal from the received signal on the air. In this way, even if the main channel is occupied, the transmission latency in the side channel is limited to the duration of one data frame.

As described in Section 2.2, the LTS preamble has its unique bit structure and can be identified by passively overhearing the channel. Asynchronous data transmissions in EasyPass, hence, does not require any active time synchronization or coordination between the side channel Tx and the main channel Tx. Since the side channel Tx does not decode the main channel's transmitted data, it produces similar amounts of computation and communication overhead with that of commodity WiFi. The computation overhead at the side channel Rx, on the other hand, depends on the number of OFDM subcarriers used by the side channel.

3.1 Two-step Equalization

Such asynchronous transmission is achieved by two steps of channel equalization at the side channel Rx. For each subcarrier that the side channel operates, as shown in Figure 3(a), the mixed signal Y received by the side channel Rx is

$$Y = Y_M + Y_S + N_S = X_M H_M + X_S H_S + N_S, \quad (1)$$

where H_M and H_S indicate channel responses and N_S indicates channel noise. Since the side channel Tx only starts transmitting its frame after the LTS of the main channel frame ends, the side channel Rx estimates the main channel response H_M in advance, and further use H_M to demodulate the main channel signal as $\hat{X_M} = H_M^{-1} Y$ by considering Y_S as background noise after the side channel frame starts. We subtract $\hat{X_M}(H_M)$ from Y, and detect the

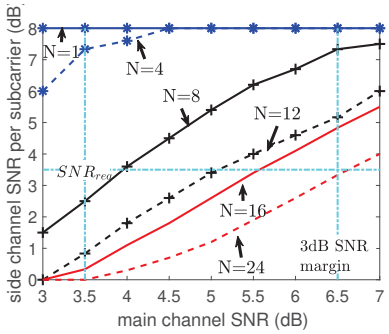
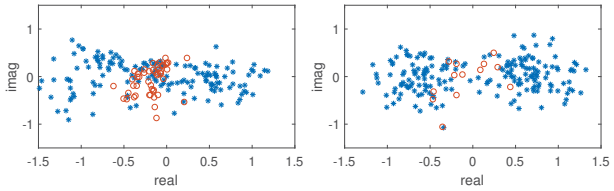


Figure 4: Multiple accessibility of side channel



(a) Without pilot tones (b) With pilot tones

Figure 5: Side channel signals after the 2nd equalization. ('○' indicates demodulation error)

preamble of the side channel frame from the residual signal $\hat{Y}_S$ to estimate H_S. Then, the 2nd channel equalization over the residual signal is able to demodulate the side channel data as $\hat{X}_S = H_S^{-1}\hat{Y}_S$.

When the main channel frame ends, the side channel Tx will keep transmitting if the spectrum becomes idle. Otherwise, it will stop transmitting if the spectrum is occupied by other main channel signaling (e.g., beacon or ACK frames), and wait until the next main channel's data frame starts.

3.2 Multiple Accessibility of Side Channel

In EasyPass, side channel transmitters access different subsets of the main channel's OFDM subcarriers. Multiple accessibility to the side channel, hence, depends on the number of subcarriers that can be used by the side channel without impacting the main channel.

To experimentally investigate such multiple accessibility, we operate the main channel and the side channel between two pairs of USRPs, which are placed 3 meters away and both transmit at 6 Mbps. The side channel is only applied to N subcarriers with 3 dB lower power. Figure 4 plots the correlation between the main channel SNR and side channel SNR with different values of N, and demonstrates that N relates to the main channel's SNR margin. As the main channel SNR improves, stronger power can be used in the side channel as extra interference, which can be distributed to more subcarriers. For example, when the main channel SNR improves from 3.5 dB (the minimum SNR_{req} required for 6 Mbps) to 6.5 dB and results in a 3 dB SNR margin, it allows N to increase from 4 to 16[4]. When the main channel's link quality degrades, the side channel can also ensure correct data transmission by reducing the value of N and increasing the per-subcarrier transmit power. Details on balancing between these two aspects are described in Section 4.

[4]The value of N must ensure that the side channel SNR also exceeds 3.5 dB.

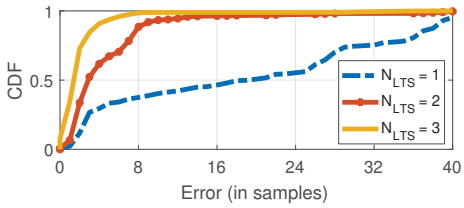


Figure 6: Error of symbol alignment

3.3 Minimizing Symbol Errors

To investigate the accuracy of side channel demodulation, we did an experiment by operating the main channel at 6 Mbps and using 3 subcarriers for the side channel. Figure 5(a) plots the side channel signals extracted by two-step equalization, and demonstrates non-negligible demodulation errors. The major reason for these errors is that the channel response could vary over time in practice, especially in fast-fading channels. The estimated channel response during two-step equalization, hence, could be obsolete and inaccurate when demodulating some symbols in a frame. Commodity wireless solves this problem by continuously tracking the channel through coherent signal detection in specialized pilot subcarriers, which are however, infeasible for a side channel user that only operates few subcarriers.

Instead, EasyPass uses block-type pilots [49, 51], which reserve one OFDM symbol every K symbols to deliver the pilot tones in the time domain. H_S, in this case, will be updated as the distortion of pilot tones that continuously tracks the fading channel. In EasyPass, we choose $K = 10$ and Figure 5(b) demonstrates that the majority of demodulation error can be successfully eliminated.

Another source of demodulation error comes from the possible mis-detection of symbol boundary. Current techniques detect such boundary via frequency-domain correlation over LTS, but fail when applied to the side channel accessed by multiple users: LTSs from different users will collide if each of them spans all OFDM subcarriers, but provide insufficient frequency diversity of precise detection if being transmitted over only few subcarriers. Our experiment results in Figure 6 with 2.5 dB main channel SNR show that a LTS should be transmitted over at least $N_{LTS} = 3$ subcarriers to restrain the error of symbol boundary under 8 samples, which is half of the cyclic prefix that resists against symbol misalignment. Hence, EasyPass allocates a minimum of 3 subcarriers to each side channel user. Details of such distributed allocation is described in Section 5.

4 TRANSMIT POWER CONTROL

EasyPass controls the side channel's transmit power by adjusting the number of OFDM subcarriers used by the side channel. As shown in Figure 7, its maximum transmit power ($P^S_{\max}$) should be within the main channel's SNR margin ($\overline{SNR}^M$), to avoid affecting the main channel. Its minimum transmit power *per subcarrier* ($P^S_{\min}$) should be sufficient for correct data reception. The number of subcarriers that the side channel can operate, hence, is decided as $\lfloor P^S_{\max}/P^S_{\min} \rfloor$.

4.1 Determine $P^S_{\max}$

As shown in Figure 8, for each main channel Rx, the side channel's received power ($P^S_{\max} - PL_{S\to M}$) should equal to the minimum

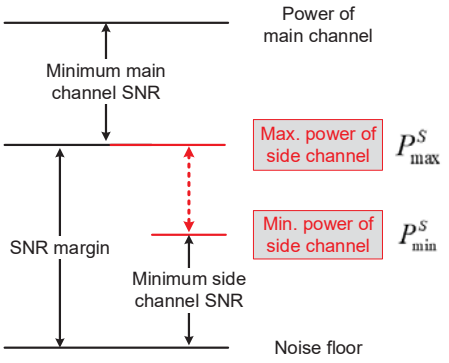


Figure 7: Power bounds of side channel

SNR ($\overline{SNR}^M + N_M$) required by the main channel Rx (R_M), where $PL_{S\to M}$ indicates the path loss from T_S to R_M, and N_M indicates the noise floor at R_M. Hence, $P^S_{\max}$ could be calculated as

$$P^S_{\max} = PL_{S\to M} + \overline{SNR}^M + N_M. \quad (2)$$

When there are multiple main channel receivers, the side channel Tx evaluates its $P^S_{\max}$ with respect to each main channel Rx and pick up the minimum one.

One solution to obtain $P^S_{\max}$ is that R_M reports $\overline{SNR}^M$ and N_M to the side channel, but requires hardware modification on main channel devices. Instead, we indirectly estimate the quantities in Eq. (2) at side channel users, which actively probe the main channel using commodity protocols.

Estimating $PL_{S\to M}$: EasyPass leverages the standard power control mechanisms being widely used in existing wireless networks (e.g., Transmit Power Control in 802.11 [42, 43] and Loop Power Control in LTE [22, 52]), which allow a sender to decide its transmit power based on the link path loss reported from the receiver. The side channel Tx sends a power control (PC) request frame to the main channel Rx, which responds with a PC response frame containing its transmit power for sending this frame. $PL_{S\to M}$ is then calculated as the difference between the transmit and receive power of PC response frame.

In practice, such PC request frames only need to be sent when the channel SNR has noticeably changed. The extra communication overhead caused by sending PC request frames is hence negligible due to the small size of these frames, which is similar to the size of the widely used RTS/CTS frames in 802.11.

Estimating N_M: Estimating the noise floor (N_M) at the main channel Rx from the side channel is much harder, because the side channel has limited knowledge about the main channel's propagation. Such estimation, however, is necessary for EasyPass to efficiently operate with the minimum bit error rates (BER) in dynamic wireless network conditions. In EasyPass, the side channel Tx probes the main channel Rx by sending multiple PC requests with increasing levels of transmit power P^S, and calculate the frame delivery ratio (FDR) of the corresponding PC responses to estimate the receiving SNR at the main channel Rx as $f(FDR)$. N_M can then be decided as

$$N_M = P^S - PL_{S\to M} - f(FDR) \quad (3)$$

In theory, FDR always increases when channel SNR improves. To precisely correlate between FDR and the corresponding channel

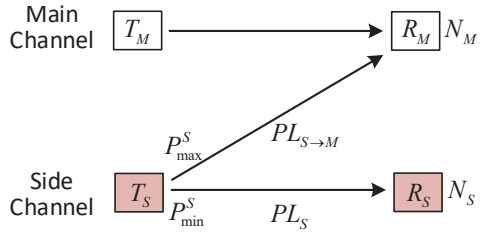


Figure 8: Adaptive transmit power control

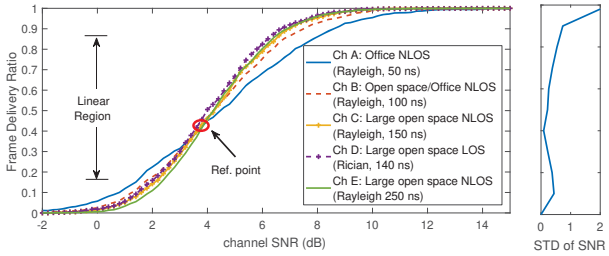


Figure 9: FDR vs. SNR timing indicates the r.m.s delay spread of the tapped delay.

SNR, we tested the FDR under different channel SNR with HIPERLAN/2 fading models [16, 17], which emulate the physical-layer characteristics of 802.11 networks in various practical link scenarios (e.g., with a communication range up to 50 meters and mobility up to 2m/s). The multipath fading is modeled with tapped delay line models that emulate multiple echoes from the transmitted signal. As shown in Figure 9(a), each channel model requires a different SNR to reach a certain FDR, with a variance up to 2 dB when FDR>85%. The standard deviation of such variance is shown in Figure 9(b), and reaches the minimum (0.09 dB) when FDR is 0.4. In other words, given P^S that yields an FDR of 0.4, the channel SNR (3.8 dB) is nearly constant for all channel models. Hence, we use $(SNR_{ref}, FDR_{ref}) = (3.8, 0.4)$ as the reference point: P^S is adjusted until FDR reaches FDR_{ref}. N_M can then be calculated as

$$N_M = P^S_{FDR_{ref}} - PL_{S\to M} - SNR_{ref}, \quad (4)$$

where $P^S_{FDR_{ref}}$ is the side channel transmit power to reach FDR_{ref} at the main channel Rx. Note that N_M is usually constant and only needs to be estimated once. In practice, when the link fading characteristics vary, similar methods can be applied to the corresponding channel fading model to estimate the main channel's noise floor.

4.2 Determine $P^S_{\min}$

$P^S_{\min}$ can be determined in a similar way as described in Section 4.1, but requires less operations by piggybacking the required information about link characteristics in the side channel Rx's ACK frames back to the side channel Tx. Since $P^S_{\min}$ only depends on the characteristics of side channel itself, it can be calculated as

$$P^S_{\min} = PL_S + SNR^S_{\min} + N_S, \quad (5)$$

where $SNR^S_{\min}$ indicates the minimum side channel SNR required that can be decided by the side channel Rx according to the error rate of data demodulation.

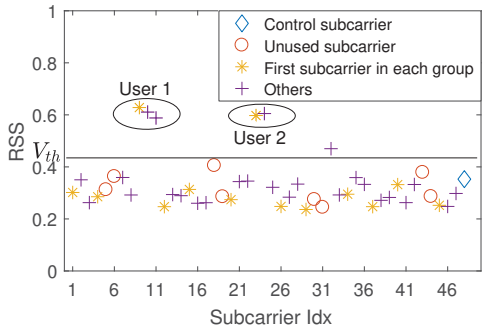


Figure 10: Subcarrier occupancy sensing. $V_{th} = 0.45$ when $\overline{V} = 0.32$ and $\alpha = 1.4$.

5 DISTRIBUTED SUBCARRIER ALLOCATION

EasyPass allocates the available SNR margin to side channel users in a fully distributed manner, and each user independently decides the subcarriers to use based on its local channel sensing, so as to avoid any explicit coordination with other users. Since the main channel's SNR margin is intermittently available, EasyPass also minimizes the latency incurred by such subcarrier allocation, to ensure that any available SNR margin can be instantaneously utilized.

In practice, EasyPass adapts such allocation to how the main channel utilizes the spectrum. For example, a 20MHz 802.11g channel uses 48 subcarriers for data transmission, 4 subcarriers as pilots, and the remaining as guard bands. Among the 48 data subcarriers, EasyPass reserves the 8 subcarriers next to pilots to ensure correct main channel equalization, and use 1 subcarrier as the control plane. Thus, 39 subcarriers can be used by EasyPass to support multiple access to the side channel. As discussed in Section 3.3, to guarantee the side channel performance, each side channel user should be assigned with at least one group of 3 consecutive subcarriers. Thus, a maximum of 13 concurrent IoT transmitters can be support by EasyPass. In this way, EasyPass allows the main channel to efficiently eliminate concentrated bit errors from these interfered subcarriers via interleaver shuffling across the frequency band [1].

Similarly, EasyPass could provide extra multiple accessibility to the side channel, if the main channel utilizes the spectrum in a more fine-grained manner. For example, 802.11n/ac uses 52 data subcarriers in a 20MHz main channel and supports 14 concurrent side channel users. Such multiple accessibility further increases to 72 users when a wideband 80MHz main channel with 234 data subcarriers is used.

5.1 Subcarrier Occupancy Sensing

Before transmitting each frame, a side channel user senses all the subcarriers in the main channel and decide whether they are being used by another side channel user, by measuring its level of energy after two-step equalization. Figure 10 plots the energy level of 48 data subcarriers in a 802.11g channel, and demonstrates that the subcarriers being used by side channel users have significantly higher levels of energy. In practice, we use a RSS threshold $V_{th} = \alpha\overline{V}$ to decide whether a subcarrier is occupied, where $\overline{V}$ denotes the average power of the 8 unused data subcarriers and α is the scaling factor that varies with the channel conditions. Figure 10 shows that $\alpha = 1.4$ ensures $> 95\%$ detection ratio of subcarrier occupancy when the side channel SNR is 3 dB.

To minimize the sensing latency in practice, EasyPass monitors the overall energy level in the entire frequency band of the main channel, and then divides the outcome into individual OFDM subcarriers after FFT. In this way, such sensing latency can be effectively controlled within a few microseconds (see Section 7.5).

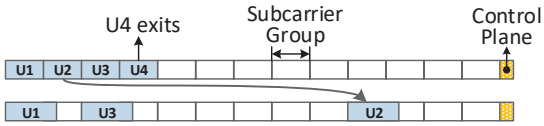


Figure 11: Subcarrier reallocation when User 4 exits. User 2 is reallocated since both of its neighboring groups are occupied.

5.2 Subcarrier Allocation

The current number of existing side channel users in the network, indicated by m, is critical to appropriate subcarrier allocation, and each side chanenl transmitter autonomously maintains its own record about m. More specifically, when a new device accesses the side channel, it announces itself by transmitting in the control subcarrier, so that all users are aware about the update of m and autonomously reduce their occupancy of subcarrier groups by $\lfloor 39/m \rfloor - \lfloor 39/(m+1) \rfloor$.

In addition, side channel users keep monitoring the occupancy of each subcarrier group, in order to detect completions of data transmission and consequent availability of new subcarriers. In these cases, the value of m will be reduced accordingly, and the spare subcarrier groups will be reallocated to other uses for maximum utilization of the SNR margin. For example in Figure 11, when U4 exits, both U1 and U2 cannot expand due to the subcarrier collision. To address this problem, each user calculates the number of its neighboring groups that are occupied, and adjusts their utilization to the vacant subcarriers.

To evaluate the effectiveness of such allocation scheme, we conducted experiments using 3 side channel users with dynamic traffic patterns, and then collect the real-time subcarrier occupancy from the side channel Rx. Results in Figure 12 show that users initiate the link by accessing the control plane, such as User 3 does at frame 8. When User 1 remains inactive for a certain period (5 seconds), User 2 and 3 would enlarge their subcarrier occupancy (Frame 11) to fully utilize the SNR margin. These results demonstrate that EasyPass can promptly adapt to the dynamic traffic patterns in IoT with high utilization to the intermittently available SNR margin: the overall throughput of the side channel could reach 2.25 Mbps (18 subcarriers) at Frame 12.

5.3 Addressing Conflicts

When multiple users access the side channel simultaneously, it is likely that a subcarrier is accessed by more than one user due to incorrect subcarrier occupancy sensing or the hidden terminal problem, resulting in conflict and transmission error. Such conflict has a higher chance to occur within a large group of users or fast changing channel conditions. To address such conflict, the

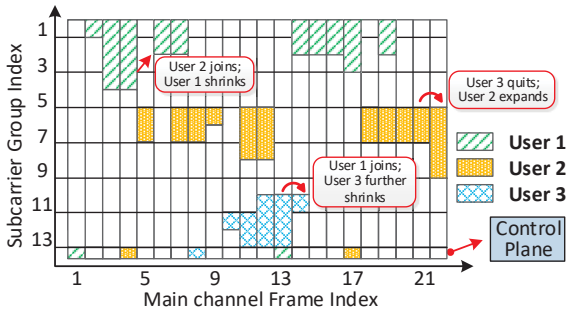


Figure 12: Subcarrier allocation

side channel users keep track of the occupation of all subcarrier groups, and maintain a list of consecutive vacant groups. A weight $w_n = 2^n$ is then assigned to each of the n vacant groups. Upon detecting conflicts from successive data transmission failures, the user randomly reselects its occupancy from the list according to their weights, hence preventing conflicting users from jumping to the same group again.

6 IMPLEMENTATION

We implemented EasyPass over USRP to realize a 802.11g transceiver over the 2.4GHz band [9]. EasyPass only modifies the side channel transceivers but retains the main channel transceivers intact.

6.1 PHY Implementation

The PHY implementation of EasyPass is shown in Figure 13, with newly added or modified modules highlighted in blue. At the transmitter, we modify the subcarrier allocator module of WiFi, which assembles OFDM symbols by filling data into subcarriers. When being implemented on FPGA-based RF radio platforms such as WARP, these extra modules correspond to less than 1% of the logical interconnects being involved, while keeping the majority of the 802.11 PHY structure intact. Therefore, the hardware modification in order to implement EasyPass is minimum.

Since EasyPass only transmits at a subset of data subcarriers and does not transmit pilots, the modified subcarrier allocator module allocates subcarriers among the selected ones that are decided upon Clear Channel Assessment (CCA). This implementation could also be easily adopted to other wireless PHY with narrower channel bandwidth, such as ZigBee or Bluetooth, by aligning the side channel transmitter's spectrum to the main channel's subcarriers being allocated for side channel operations.

At the receiver, the key challenge is to estimate and remove the the main channel's signal from the mixed signal being received, which are decided by 1) the estimation of main channel Tx signals, 2) the channel distortion, and 3) Carrier Frequency Offset (CFO). In order to obtain these, we pass the received signal through the regular WiFi decoding procedure, and store the estimated channel response and CFO for later use. Afterwards, we recover the main channel Tx signals by passing the decoded bits into regular WiFi encoding procedure. Meanwhile, the channel distortion and CFO are re-applied through De-equalizer and CFO re-applier modules, respectively. The residual signals of the side channel can then be subtracted from mixed signal being received. Finally, we use side channel detector to detect the arrival of the side channel signal for decoding.

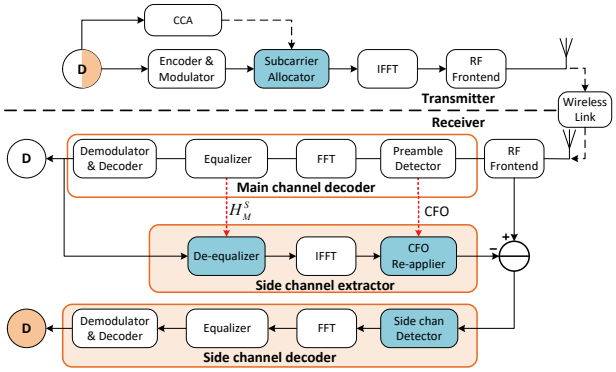


Figure 13: PHY implementation of EasyPass

6.2 MAC Implementation

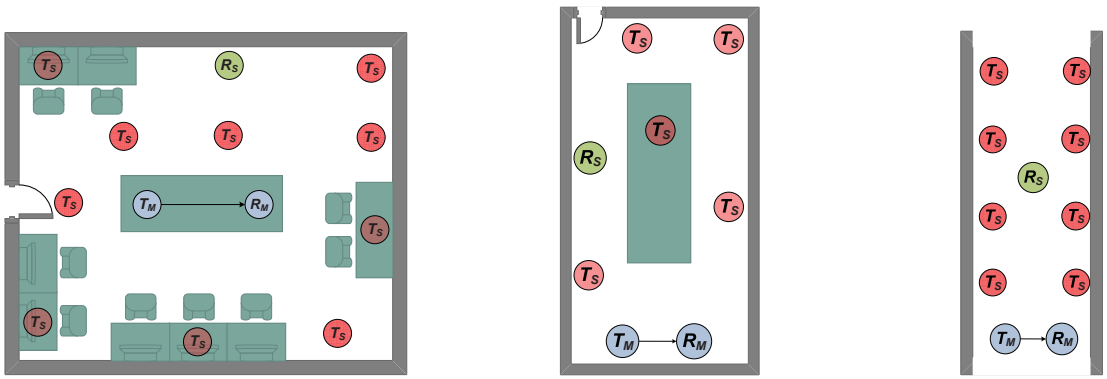
We implemented a complete WiFi MAC layer with functionalities such as carrier sensing and retransmission [24]. The IEEE 802.11 standards do not specify the rate adaptation algorithms being used. Without loss of generality, we integrated the Minstrel algorithm [2], which is one of the most popular rate adaption methods and has been widely used in popular Linux-based WiFi drivers such as MadWiFi, Ath5k and Ath9k [64], into our implementation. In this way, we support the realistic MAC behaviors of commodity wireless devices, and also emulate the main channel's available SNR margin in practical wireless scenarios.

7 PERFORMANCE EVALUATION

As shown in Figure 14, we evaluate the performance of EasyPass in three indoor scenarios, whose parameters are listed in Table 2. In all these scenarios, multiple pairs of main channel transmitters and receivers are being deployed to generate strong wireless interference to the side channel transmitters. Such interference, however, differs with the specific indoor layouts and the communication distance between the transmitters and receivers. In all experiments, we use USRP N210 with UBX-40 RF daughterboards for both main channel and side channel transceivers, each of which operates on a 20 MHz channel. Every three USRPs are connected to a Dell desktop PC for packet processing and overhead measurements.

We evaluate the performance of EasyPass in a highly challenging wireless network scenario, where the available wireless spectrum is fully occupied by the main channel traffic. More specifically, the main channel Tx continuously transmits frames with payloads of 1500 bytes to the main channel Rx, and will generate a new wireless frame right after the previous frame transmission finishes. We modify the MAC behavior of main channel Tx, such that it bypasses CSMA and transmits frames at all times even if the main channel is occupied.

In this way, we emulate a 100% occupied main channel. The delay-sensitive traffic being sent to the side channel Rx, if being sent using commodity WiFi protocols, will be fully interfered and seriously delayed by the concurrent main channel traffic on the air. Such over-the-air interference, on the other hand, could not be addressed by existing traffic prioritization schemes such as QoS



(a) Dense deployment (b) Sparse deployment (c) Hallway

Figure 14: Experiment scenarios. T_M and R_M indicate the main channel Tx and Rx. T_S and R_S indicate the side channel Tx and Rx.

	Room size	No. side channel Tx	Avg. side channel comm. distance	Avg. distance to main channel Rx	Main channel placement	Indoor furniture
Dense deployment	9m × 9m	10	4m	2.5m	Room center	Many
Sparse deployment	18m × 10m	5	7.5m	10.7m	Side of room	Few
Hallway	25m × 4m	8	6m	14.5m	End of hallway	None

Table 2: Parameters of experiment scenarios

control, which is limited to address MAC-layer contention among different traffic flows at a local device.

We fix the distance between the main channel Tx and Rx to be 3.5m, and place multiple side channel receivers at the same location in some experiments. We tune the main channel SNR by adjusting the USRP Tx Gain that ranges between 0 dB and 31.5 dB at the step of 0.5 dB. The maximum Tx power is equivalent to 13 dBm. Only one side channel Rx is used, but it handles transmissions from different side channel Tx separately with dedicated computing resources.

7.1 Data Transmission Delay

In this experiment, each side channel Tx transmits 700 frames. The side channel is disabled for the first 350 frames, which are transmitted by commodity WiFi with CSMA and received by the side channel Rx through the main channel. Afterwards, each side channel Tx will enable EasyPass by bypassing CSMA and transmitting frames regardless channel occupancy. These frames are mixed with the main channel traffic on the air, and decoded by the side channel Rx with two-step equalization described in Section 3. In this way, the performance of EasyPass is compared with that of commodity WiFi under the same channel condition.

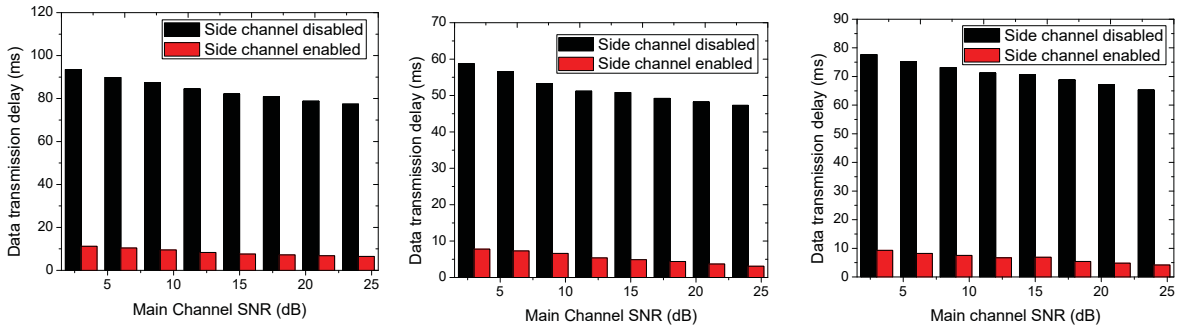
Figure 15 shows the side channel's data transmission delay. Since the link propagation delay in our experiments is around 2ms, Figure 15 shows that without EasyPass, congestion in the main channel increases the side channel's transmission delay by up to 50 times. In contrast, EasyPass controls the transmission delay within 10ms and achieves >90% delay reduction over this congested wireless link in all experiment scenarios, because transmissions in the side channel will never be delayed by traffic in the main channel. When the main channel's SNR degrades, extra channel noise may distort the main channel signal and hence results in more side channel demodulation errors. These errors lead to frame retransmissions in the side channel and increase the delay. However, Figure 15 shows that such delay increase in EasyPass is less than 20% even if the main channel SNR degrades to <5 dB, and hence demonstrates reliable data reception in the side channel.

Furthermore, such reliable data reception also minimizes the delay jitter, which is also important in many IoT applications. As shown in Figure 16, EasyPass almost avoids any overlong transmission delay (>20ms) due to retransmissions, no matter how such delay in commodity WiFi will be impacted by link congestion.

7.2 Side Channel Throughput

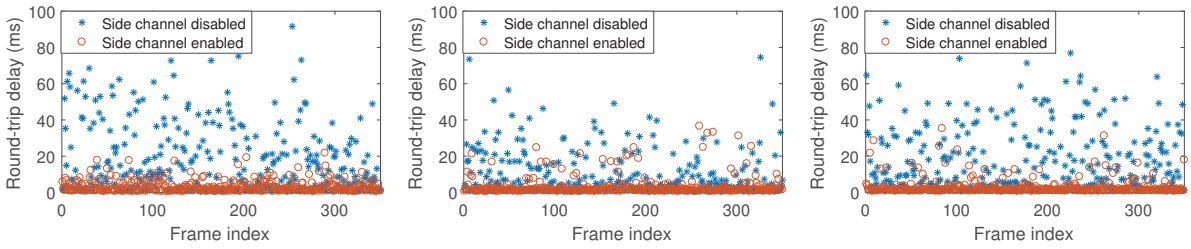
The side channel throughput is mainly affected by the main channel's condition, which influences the side channel's transmit power and determines the number of subcarriers being used by the side channel as described in Section 4. As shown in Figure 17(a), EasyPass achieves a maximum throughput of 2.5 Mbps in the Hallway scenario. In this scenario, the high path loss from the side channel Tx to the main channel Rx results in higher $P^S_{\max}$ according to Section 4.1, and the long communication distance in the side channel results in lower $P^S_{\min}$ according to Section 4.2. EasyPass, hence, can use the maximum number of subcarriers for side channel transmissions. On the other hand, the Dense deployment scenario results in the lowest side channel throughput because of short communication distance and low path loss.

In practice when the main channel condition is highly dynamic, EasyPass can also exploit a smaller amount of SNR margin for better transmission reliability and less impact to the main channel. As shown in Table 3, when the amount of SNR margin being used varies



(a) Dense deployment (b) Sparse deployment (c) Hallway

Figure 15: Data transmission delay



(a) Dense deployment (b) Sparse deployment (c) Hallway

Figure 16: Delay jitter with main channel SNR=12dB

from 1 dB to 3 dB, our proposed power control schemes in Section 4 precisely adapt the side channel's transmit power, resulting in a different number of OFDM subcarriers (N) being used.

	$\overline{SNR}^M$ (dB)	$P^S_{\max}$ (dBm)	$P^S_{\min}$ (dBm)	N	Throughput (Mbps)
Dense	1	-47	-44	2	0.25
	2	-43		5	0.625
	3	-41		9	1.125
Sparse	1	-42	-40	3	0.375
	2	-36		7	0.875
	3	-34		12	1.5
Hallway	1	-47	-54	5	0.625
	2	-43		11	1.375
	3	-41		19	2.375

Table 3: Side channel throughput with different amounts of SNR margin being used

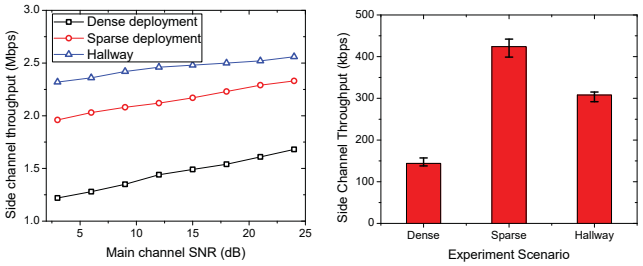
Meanwhile, we also evaluated the side channel throughput provided to each user. Results in Figure 17(b) with a main channel SNR of 12 dB show that the side channel throughputs at different users match our expectation and has low variation. Hence, EasyPass achieves maximum fairness when allocating the available side channel throughput to multiple IoT devices, without requiring any centralized or distributed coordination as described in Section 5.

Throughput with dynamic channel condition. We further investigated the variation of the side channel throughput when the wireless channel condition fluctuates in the Hallway scenario. More specifically, we configure the Tx gain of the main channel for sending the N-th data frame as $12(1 + 6sin(2\pi\omega N))$, where $\omega = 0.01$ (i.e. the period of the Tx gain change is 100 frames) and 12 is the average level of the Tx gain being configured between 6dB and 18dB. As shown in Figure 18(a), when the Tx gain fluctuates, the main channel SNR and SNR margin varies accordingly and leads to different side channel throughput over time with small delay. However, even when the main channel SNR drops to nearly 0 dB, EasyPass can still retain a minimum side channel throughput of 0.5 Mbps by using the minimal ($\leq$3) OFDM subcarriers, and hence exhibits great adaptability and applicability to severe environmental contexts in IoT applications.

No. main receivers	1	2	3	4	5
Dense	1.44	1.37	1.28	1.22	1.14
Sparse	2.12	2.03	1.94	1.88	1.81
Hallway	2.46	2.31	2.19	2.10	1.98

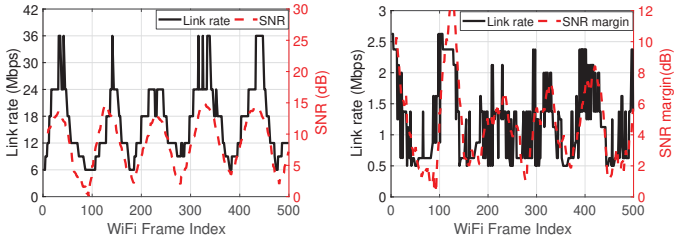
Table 4: Side channel throughput with multiple main channel receivers and SNR=12 dB

Throughput with multiple main channel receivers. As discussed in Section 4, when there are multiple receivers in the main channel, the side channel Tx uses the minimum value of $P^S_{\max}$ among them. The number of subcarriers used by the side channel hence reduces, resulting in lower throughput. As shown in Table 4, such throughput reduction could be 20% in Dense deployment and Hallway scenarios where the distance among devices is smaller. EasyPass provides at least 1 Mbps throughput even in a densely placed main channel network with 5 receivers, and can ensure 2 Mbps throughput in Hallway scenario with similar settings.



(a) Cumulative throughput (b) Per-user throughput

Figure 17: Side channel throughput



(a) dynamic channel conditions (b) Side channel throughput

Figure 18: Side channel throughput with dynamic channel conditions in Hallway scenario

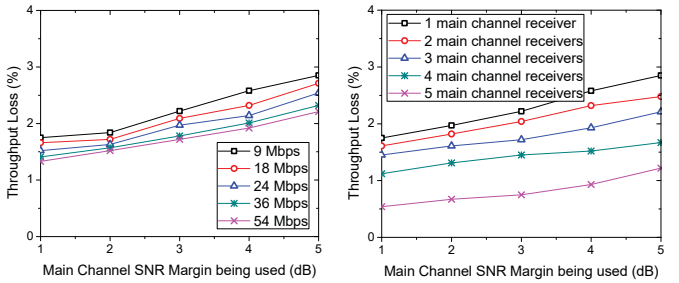
7.3 Impact on the Main Channel

The side channel operations of EasyPass should have the minimum impact on the main channel. To evaluate the main channel's throughput loss due to side channel operations, we vary the main channel's condition in Dense deployment scenario with the shortest distance to the main channel Rx, and aggressively utilize all the available SNR margin in the side channel. Experiment results in Figure 19(a) show that side channel operations cause at most 3% throughput loss in the main channel, when the main channel transmits at 9 Mbps but the side channel fully uses the 5 dB SNR margin. As the side channel uses less SNR margin, such impact on the main channel further decreases and will be less than 2%. The main channel's throughout loss also decreases, when the main channel adopts a higher data rate and hence provides more SNR margin for the side channel.

The existence of multiple main channel receivers, as described in Section 7.2, reduces the amount of subcarriers that can be used by the side channel and hence results in less impact on the main channel. As shown in Figure 19(b), when 3 main channel receivers are being placed, the average impact on the main channel can be reduced to 2%, and such impact can be reduced to <1% if there are 5 main channel receivers, even if 5 dB SNR margin is exploited.

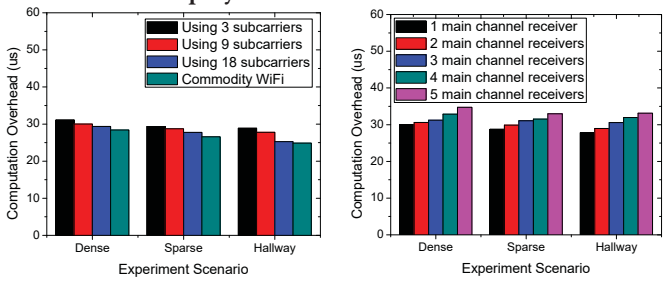
7.4 Computation Overhead

Computation overhead of side channel Tx. We evaluate the computation overhead of EasyPass by measuring the runtime of signal processing blocks over the PC. Figure 20(a) shows the Tx computational time for transmitting one frame. It demonstrates that with different numbers of subcarriers used by the side channel, the computation overhead at a side channel Tx is always at the same level with that of commodity WiFi, and the extra computation



(a) Dynamic main channel conditions (b) Multiple main channel receivers operating at 9 Mbps

Figure 19: Main channel's throughput loss due to side channel in Dense deployment scenario



(a) Using different subcarriers (b) with multiple main channel receivers

Figure 20: Computation overhead of side channel Tx

overhead due to side channel operations is less than 5%. In particular, the Tx processing time slightly increases when fewer subcarriers are being used. This is because that the computation overhead at Tx is mainly decided by the size of data payload. Since a side channel Tx only uses a portion of subcarriers and hence requires more OFDM symbols to carry the same amount of data, it requires more IFFT operations to convert each symbol into the time domain and hence incurs more overhead.

The side channel Tx's computation overhead with multiple main channel receivers is shown in Figure 20(b), with 9 subcarriers used by the side channel. As discussed in Section 4, the side channel Tx separately computes $P^S_{\max}$ for each main channel Rx, and incurs more computations. However, EasyPass restrains such overhead increase within 15%, even when there are 5 main channel receivers being placed.

Main channel data rate	No. of subcarriers			Main channel overhead
	3	6	9	
6 Mbps	653	852	1100	464
12 Mbps	839	1010	1259	632

Table 5: Rx computation overhead (μs)

Computation overhead of side channel Rx. Table 5 lists the side channel Rx's computation overhead, assuming that the main channel utilizes the same number of OFDM symbols as the side channel. Overhead of decoding and removing the main channel signal at the side channel Rx is only decided by the main channel's frame length and data rate, since the traditional equalization and decoding schemes are applied to the mixed signal. After removing the main channel signal, the residual signal will be demodulated

and decoded using the built-in blocks shown in Figure 13. Such processing time is proportional to the size of side channel payload that is decided by the number of subcarriers it exploits.

7.5 Scalability of EasyPass

As described in Section 5, EasyPass enables up to 13 IoT devices to concurrently transmit data without being delayed. In practice, data traffic in IoTs is mostly event-driven rather than continuous, and EasyPass can hence combat delays in a much larger IoT network, where wireless devices do not always transmit data at the same time. To investigate such scalability, we deploy more side channel transmitters but set them to transmit with different volumes of traffic: a volume v indicates that a side channel Tx will delay its next frame transmission for $(1-v)T$ every time after it finishes transmitting a frame that lasts for T, and the size of payloads is randomly varying between 500 to 1500 bytes to mimic traffic asynchrony in practice. To ensure proper placement of these transmitters, we place multiple transmitters at the same location, with one next to each other. In these experiments, the main channel's SNR is set as 20 dB.

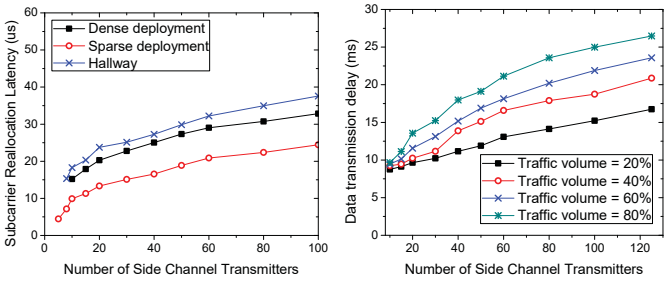
We first examine the timeliness of EasyPass on occupying the available subcarriers for side channel access, which is crucial to utilize the intermittently available SNR margin. As shown in Figure 21(a), whenever a side channel Tx finishes its transmission, EasyPass could promptly sense the new availability of subcarriers and reallocate them to another side channel Tx, and such reallocation latency can be controlled within 35us even in a IoT network with 100 devices. According to Figure 20, such latency is at the same level with the Tx's computation overhead for transmission, and hence ensures prompt adaptation to the SNR margin fluctuation.

Based on such capability of fast subcarrier sensing and reallocation, Figure 21(b) further shows that EasyPass can provide an average transmission delay of <30 ms in the side channel, even when more than 100 wireless transmitters are being placed in the Dense deployment scenario and their traffic volume is 80%. In particular, the delay increase is only noticeable when there are more than 50 devices in the network or their traffic volume is higher than 50%. Based on these results, we are confident that EasyPass can concurrently minimize the transmission delay of >100 devices, over a single 20MHz wireless link. EasyPass can also be extended to support more IoT devices based on fine-grained spectrum multiplexing in the main channel, such as 802.11n/ac or OFDMA [50, 59].

8 REAL-WORLD EXPERIMENTATION

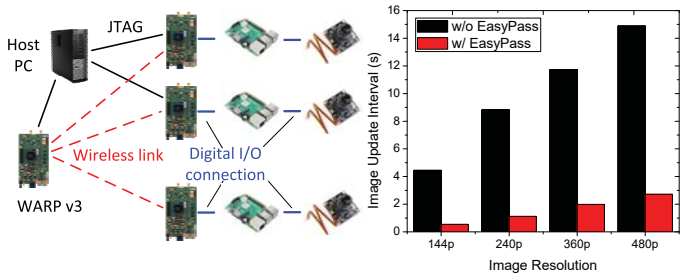
In this section, we examine the applicability of EasyPass in real-world IoT scenarios, by emulating a home surveillance camera system. By evaluating its performance with specific metrics, we demonstrate that EasyPass can greatly improve the performance of delay-sensitive IoT systems.

To transmit the camera's data at real-time, we first migrated the implementation of EasyPass described in Section 6 to the WARP v3 boards [23] by customizing its WiFi reference design. Then, as shown in Figure 22(a), we deploy three low-power CMOS cameras and connect each low-power CMOS camera module to a Raspberry Pi 3 board. We then use WARPs as EasyPass transceivers to make wireless connection between the Raspberry Pi 3 and the host PC. To ensure correct evaluation, the WARP transmitters and receivers are placed 5 meters away from each other with line of sight connection.



(a) Subcarrier allcation latency (b) Data transmission delay

Figure 21: Scalability of EasyPass



(a) System setup (b) Image update interval

Figure 22: Experiments with surveillance cameras

Based on this implementation, we use a Dell Optiplex 5810 workstation with modified WiFi firmware to keep streaming data to a Netgear Nighthawk AC1900 wireless router, so as to create fully congested main channels. Then, we use the above system to transmit the captured images with different resolutions from the Raspberry Pi 3 to the host PC via the wireless channel build by EasyPass from WARP boards, and measure the average interval for such images to be fully received at the PC[5]. Results in Figure 22(b) show that EasyPass can fundamentally prevent such delay-sensitive data transmissions from being delayed by wireless link congestion, and hence controls the update interval of images within 2 seconds at all resolutions. When 480p resolution is being used, such update interval is reduced by up to 85%. In practice, reducing the interval of updating images with 480p resolution to 2 seconds would enable various camera surveillance applications, such as home security and city monitoring [11].

9 RELATED WORK

Side Channel Design. Prior designs of the in-band side channel are mainly applied for delivering coordination and control information, so as to improve the network performance without introducing extra costs compared to the out-of-band approaches. Most schemes hide information into the side channel for private communication, by alternating the physical characteristics of the main channel such as signal duration [10], the position of interfered chips [62], timing of periodic beacon frames [30], intentionally corrupted checksums [40] and the position of the blank subcarriers [38]. These side channel designs can only provide limited throughputs and do

[5]Most commercial systems also adopt this method to update images at a fixed interval (e.g., a few seconds), to save bandwidth and ensure reliability.

not support multi-access operations due to their dependency with the main channel. hJam [61] exploits the preamble redundancy in OFDM, and detects side channel signals using those over blank subcarriers in frame preamble as the reference. The main channel data is recovered through interference cancellation, which however, requires firmware modifications of main channel receivers.

Interference Cancellation. Existing research on interference cancellation techniques gives insights to our work but is different from EasyPass. Full duplex radio [6, 14, 18, 26, 46] allows simultaneous transmit and receive by estimating the phase and amplitude of its transmitted signals. The full-duplex design, however, is not suitable for our proposed work for two reasons. First, the side channel Rx aims at extracting the useful information from the main channel signals, rather than the locally transmitted signals. Second, implementation of full-duplex usually requires modifications on the RF front end, and hence doesn't meet our objective. Zigzag decoding [20] utilizes interference cancellation technique to recover information from collided frames. It exploits the asynchrony of successive collisions caused by the hidden terminal. These successive transmissions of both channels, though, is unavailable in our work. EmBee [12] removes interference across different wireless technologies by reserving wireless spectrum for low-power wireless devices. Such explicit spectrum reservation, however, cannot be adopted to IoT devices that operate the same wireless technology.

Subcarrier Allocation. Subcarrier allocation has been widely studied to optimize system metrics, such as power efficiency [59] and channel throughput [50], among which almost all require centralized control. [31] suggests using average SNR for each user to decide the total number of subcarriers to be assigned. SNOW [47, 48] allows single-hop communication between the base station and sensors in sensor networks, by utilizing the untapped TV whitespace spectrum. Sensors are assigned non-overlapping narrowband orthogonal subcarriers to avoid collisions, which is centrally controlled by BS. BS also controls the timing of data transmission in a TDMA fashion. Different from our work, each subcarrier of the TV whitespace spectrum is occupied only by SNOW nodes. Interference control is hence not required. Recent technical advances such as OFDMA [36] [31] and 802.11ax [4], on the other hand, achieve fine-grained subcarrier allocation through centralized coordination. This coordination, however, may not be feasible in large-scale IoT scenarios with large population of autonomous devices.

10 DISCUSSIONS

In this section, we discuss how EasyPass could be further expanded to operate over a larger collection of wireless networks, as well as its resistance against other sources of wireless interference and possible hidden terminals.

10.1 Expandability and Compatibility

As discussed in Section 5, EasyPass well adapts to different spectrum multiplexing (e.g., 802.11n/ac or OFDMA) in the main channel, by expanding its subcarrier allocation to a wider collection of OFDM subcarriers. Similarly, EasyPass is expandable to wireless MIMO, by individually sensing and utilizing the SNR margin in each data stream. In particular, the SNR margins in different data streams may differ due to their heterogeneous multipath conditions, and an IoT transmitter could adaptively choose the ones with the maximum SNR margin to operate the side channel.

EasyPass can also be expanded to new generations of WiFi standards with wider channel bandwidth (e.g., 802.11n/ac that allows a single channel's bandwidth to reach 160MHz via channel bonding), because the same OFDM technique is being adopted by these new standards in modulation. Furthermore, exploiting the extra OFDM subcarriers provided in these newer WiFi standards allows EasyPass to achieve higher thoughput in the side channel or support more side channel transmitters at the same time.

EasyPass builds on the main channel's OFDM PHY, and is capable of providing side channel access to IoT transmitters using ZigBee or Bluetooth, even though their PHY layers do not use OFDM. The key reason of such compatibility is the narrowband channels of ZigBee (2MHz) and Bluetooth (1MHz), which allow such a transmitter to apply its signal to a selected set of main channel's subcarriers. Such side channel data could be decoded from Zigbee or Bluetooth based on the various signal patterns received by Zigbee or Bluetooth. Data decoding technique in Section 3, then, can be similarly applied to the side channel receiver via PHY customization.

10.2 Other Interfering Sources

In practice, IoT may be diverse with various interfering sources, such as cordless phones or baby monitors that share the unlicensed ISM bands with unknown interference patterns. Our key insight of addressing these interfering sources, however, is that they will also affect the main channel at the same time. Hence, EasyPass can identify these interfering sources by monitoring the main channel's behavior in the similar way as in Section 3, and only transmits when the channel is clear for the main channel Tx to transmit.

10.3 Tackling Hidden Terminal Problem

The performance of EasyPass may be impaired by the hidden terminal problem in the main channel. As discussed in Section 3, its correct data modulation relies on overhearing the frame preambles transmitted from the main channel Tx, which however, may be a hidden terminal for the side channel Tx. EasyPass can leverage the existing mechanisms in commodity wireless networks to combat such problem. For example, the main channel's frame preambles can instead be overheard by the side channel Rx, which coordinates data transmissions with the side channel Tx using RTS/CTS mechanism in WiFi. More advanced techniques such as ZigZag decoding [20] can also be used on this problem.

11 CONCLUSION

In this paper, we present EasyPass that minimizes wireless latency from multiple IoT devices over a congested link. These users concurrently access a wireless side channel without impacting the main channel. We implemented and evaluated our design over practical SDR platforms, and demonstrate more than 90% delay reduction over multiple IoT devices.

ACKNOWLEDGMENTS

We thank our shepherd Lakshminarayanan Subramanian and anonymous reviewers for their comments and feedback. This work was supported in part by the National Science Foundation (NSF) under grant number CNS-1617198, CNS-1812399 and CNS-1812407.

REFERENCES

[1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Standard 802.11*, 2012.

[2] Minstrel rate control algorithm for 802.11 networks, 2017. https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel.

[3] Hbo go, 2018. https://play.hbogo.com/.

[4] B. Bellalta. IEEE 802.11ax: High-efficiency WLANs. *IEEE Wireless Communications*, 23(1):38–46, 2016.

[5] M. Benjamin. *Drone warfare: Killing by remote control*. Verso Books, 2013.

[6] D. Bharadia, E. McMilin, and S. Katti. Full duplex radios. *ACM SIGCOMM Computer Communication Review*, 43(4):375–386, 2013.

[7] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans. In *Proceedings of the 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications(PIMRC)*, pages 392–396. IEEE, 1996.

[8] S. Biaz and S. Wu. Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison. In *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, pages 130–136, 2008.

[9] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. An IEEE 802.11a/g/p OFDM Receiver for GNU Radio. In *ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF)*, pages 9–16, 2013.

[10] K. Chebrolu and A. Dhekne. ESense: communication through energy sensing. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 85–96. ACM, 2009.

[11] N. Chen, Y. Chen, X. Ye, H. Ling, S. Song, and C.-T. Huang. Smart city surveillance in fog computing. In *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, pages 203–226. Springer, 2017.

[12] R. Chen and W. Gao. Enabling cross-technology coexistence for extremely weak wireless devices. In *IEEE INFOCOM*, pages 253–261, 2019.

[13] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu. B2w2: N-way concurrent communication for iot devices. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, pages 245–258, 2016.

[14] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *ACM MobiCom*, 2010.

[15] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath. Flashback: Decoupled lightweight wireless control. *ACM SIGCOMM Computer Communication Review*, 42(4):223–234, 2012.

[16] E. N. Committee et al. Channel models for hiper-lan/2 in different indoor scenarios. *ETSI EP BRAN 3ERI085B*, 1998.

[17] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson. A comparison of the hiperlan/2 and ieee 802.11 a wireless lan standards. *IEEE Communications magazine*, 40(5):172–180, 2002.

[18] M. Duarte and A. Sabharwal. Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results. In *Proceedings of the 44th Asilomar Conference on Signals, Systems and Computers*, pages 1558–1562. IEEE, 2010.

[19] E. Felemban and E. Ekici. Single hop ieee 802.11 dcf analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases. *IEEE Transactions on Wireless Communications*, 10(10):3256–3266, 2011.

[20] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. 38(4), 2008.

[21] S. Gollakota and D. Katabi. Physical layer wireless security made fast and channel independent. In *Proceedings of IEEE INFOCOM*, pages 1125–1133. IEEE, 2011.

[22] J. Gora, K. I. Pedersen, A. Szufarska, and S. Strzyz. Cell-specific uplink power control for heterogeneous networks in lte. In *IEEE 72nd Vehicular Technology Conference Fall*, pages 1–5. IEEE, 2010.

[23] S. Gupta, C. Hunter, P. Murphy, and A. Sabharwal. WARPnet: Clean slate research on deployed wireless networks. In *Proceedings of the 10th ACM international symposium on Mobile ad hoc networking and computing*, pages 331–332. ACM, 2009.

[24] J. R. Gutierrez-Agullo, B. Coll-Perales, and J. Gozalvez. An IEEE 802.11 MAC software defined radio implementation for experimental wireless communications and networking research. In *Wireless Days (WD), 2010 IFIP*, 2010.

[25] N. Homma, S. Nagashima, Y. Imai, T. Aoki, and A. Satoh. High-resolution side-channel attack using phase-based waveform matching. In *Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems*, pages 187–200. 2006.

[26] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. Practical, real-time, full duplex wireless. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 301–312. ACM, 2011.

[27] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 321–332. ACM, 2009.

[28] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami. An information framework for creating a smart city through internet of things. *IEEE Internet of Things journal*, 1(2):112–121, 2014.

[29] C. Joo, X. Lin, J. Ryu, and N. B. Shroff. Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels. In *Proceedings of ACM MobiHoc*, pages 89–98. ACM, 2013.

[30] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 317–330. ACM, 2015.

[31] D. Kivanc, G. Li, and H. Liu. Computationally efficient bandwidth allocation and power control for ofdma. *IEEE transactions on wireless communications*, 2(6):1150–1158, 2003.

[32] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 rate adaptation: A practical approach. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2004.

[33] I. Lee and K. Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015.

[34] Y. Li and W. Gao. Interconnecting heterogeneous devices in the personal mobile cloud. In *IEEE INFOCOM*, pages 1–9. IEEE, 2017.

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

[36] D. López-Pérez, A. Valcarce, G. De La Roche, and J. Zhang. OFDMA femtocells: A roadmap on interference avoidance. *IEEE Communications Magazine*, 47(9), 2009.

[37] H. Lu and W. Gao. Scheduling dynamic wireless networks with limited operations. In *Proceedings of IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, 2016.

[38] H. Lu and W. Gao. Supporting real-time wireless traffic through a High-Throughput side channel. In *In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2016.

[39] H. Lu and W. Gao. Continuous wireless link rates for internet of things. In *Procedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 48–59. IEEE, 2018.

[40] A. Najafizadeh, R. Liscano, M. V. Martin, P. Mason, and M. Salmanian. Challenges in the implementation and simulation for wireless side-channel based on intentionally corrupted fcs. *Procedia Computer Science*, 5:165–172, 2011.

[41] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. Secret key extraction from wireless signal strength in real environments. *IEEE Transactions on Mobile Computing*, 12(5):917–930, 2013.

[42] D. Qiao, S. Choi, A. Jain, and K. G. Shin. Miser: an optimal low-energy transmission strategy for ieee 802.11 a/h. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 161–175. ACM, 2003.

[43] D. Qiao, S. Choi, and K. G. Shin. Interference analysis and transmit power control in ieee 802.11 a/h wireless lans. *IEEE/ACM Transactions on Networking*, 15(5):1007–1020, 2007.

[44] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini. Frequency-aware rate adaptation and mac protocols. In *ACM MobiCom*, 2009.

[45] R. Ratasuk, N. Mangalvedhe, Y. Zhang, M. Robert, and J.-P. Koskinen. Overview of narrowband iot in lte rel-13. In *IEEE conference on standards for communications and networking (CSCN)*. IEEE, 2016.

[46] A. Sahai, G. Patel, and A. Sabharwal. Pushing the limits of full-duplex: Design and real-time implementation. *arXiv preprint arXiv:1107.0607*, 2011.

[47] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu. Snow: Sensor network over white spaces. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, pages 272–285. ACM, 2016.

[48] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra. Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces. SenSys, 2017.

[49] Y. Shen and E. Martinez. Channel estimation in ofdm systems. *Freescale Semiconductor*, 2(6), 2006.

[50] Z. Shen, J. G. Andrews, and B. L. Evans. Adaptive resource allocation in multiuser ofdm systems with proportional rate constraints. *IEEE transactions on wireless communications*, 4(6):2726–2737, 2005.

[51] L. Shi, B. Guo, and L. Zhao. Block-type pilot channel estimation for OFDM systems under frequency selective fading channels. In *Proceedings of the International Communication Conference on Wireless Mobile and Computing*, pages 21–24, 2009.

[52] A. Simonsson and A. Furuskar. Uplink power control in lte-overview and performance, subtitle: principles and benefits of utilizing rather than compensating for sinr variations. In *IEEE 68th Vehicular Technology Conference*. IEEE, 2008.

[53] P. F. Smulders. Statistical characterization of 60-ghz indoor radio channels. *IEEE Transactions on Antennas and Propagation*, 57(10):2820–2829, 2009.

[54] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan. 60 ghz indoor networking through flexible beams: A link-level profiling. In *ACM SIGMETRICS Performance Evaluation Review*, volume 43, pages 71–84. ACM, 2015.

[55] F. Viani, F. Robol, A. Polo, P. Rocca, G. Oliveri, and A. Massa. Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation. *Proceedings of the IEEE*, 101(11):2381–2396, 2013.

[56] Q. Wang, H. Su, K. Ren, and K. Kim. Fast and scalable secret key generation exploiting channel phase randomness in wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1422–1430. IEEE, 2011.

[57] T. Wild, F. Schaich, and Y. Chen. 5G air interface design based on universal filtered (UF-)OFDM. In *Proceedings of the International Conference on Digital Signal Processing (DSP)*, pages 699–704, 2014.

[58] M. Wollschlaeger, T. Sauter, and J. Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, 2017.

[59] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch. Multiuser ofdm with adaptive subcarrier, bit, and power allocation. *IEEE Journal on selected areas in communications*, 17(10):1747–1758, 1999.

[60] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 146–157, 2006.

[61] K. Wu, H. Li, L. Wang, Y. Yi, Y. Liu, Q. Zhang, and L. Ni. HJam: Attachment transmission in WLANs. In *Proceedings of IEEE INFOCOM*, pages 1449–1457, 2012.

[62] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni. Side channel: Bits over interference. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, page 13, 2010.

[63] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. M. Ni. Side channel: bits over interference. *IEEE Transactions on Mobile Computing*, 11(8):1317–1330, 2012.

[64] D. Xia, J. Hart, and Q. Fu. On the performance of rate control algorithm minstrel. In *Proceedings of the 23rd Int'l Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*, 2012.

[65] H. Xu, V. Kukshya, and T. S. Rappaport. Spatial and temporal characteristics of 60-ghz indoor channels. *IEEE Journal on selected areas in communications*, 20(3):620–630, 2002.

[66] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.