Exploiting Wireless Broadcast Advantage for Energy Efficient Packet Overhearing in WiFi

Bing Feng, Chi Zhang, *Member, IEEE,* Haichuan Ding, and Yuguang Fang, *Fellow, IEEE*

***Abstract*—The energy efficiency of WiFi interfaces has become a significant concern for battery-powered mobile stations. In WiFi networks, a station's backoff procedure is interrupted by other stations' channel activities to receive and decode useless packets that are not intended for it, which consumes a non-negligible amount of its energy. To reduce energy consumption induced by such packet overhearing, we propose a novel transmission scheme, DSS (Data Symbol Silence), which inserts silent data symbols into the front part of a data packet at the physical layer to encode information. By exploiting the wireless broadcast advantage, the embedded information is received by neighboring stations. Data symbol level energy detection is used to extract the embedded information without receiving and decoding the whole data packet. Thus, unintended receivers can quickly stop useless packet receiving and processing. The SNR gap in current WiFi networks is utilized to ensure the inserted silent data symbols do not affect the correct decoding of the original data packet. We analyze the minimum SNR required for effective DSS and implement DSS to validate its feasibility. Compared with the 802.11 standards, our studies show that DSS significantly improves the energy efficiency of WiFi interfaces.**

***Index Terms*—WiFi, energy efficiency, packet overhearing, SNR gap, channel coding.**

I. Introduction

WIFI has become a major approach to accessing the Internet for mobile stations, such as laptops, smartphones, and tablets. However, WiFi network interface cards (NICs) are power hungry while mobile stations are mostly battery-powered. Previous works show that WiFi NIC imposes a significant energy drainage on battery-powered mobile stations [2] [3]. Thus, enhancing the energy efficiency of WiFi NIC is essential for extending mobile station's battery life.

The power saving mode (PSM) is included in the IEEE 802.11 standards to reduce energy consumption of WiFi NICs. When a station has no data packets to transmit, PSM allows its WiFi NIC to enter sleep state and periodically wake up to retrieve downlink data packets. Thus, PSM reduces a station's energy waste in waiting for unpredictable downlink data packets. Various scheduling algorithms are proposed in variants of PSM to determine when WiFi NIC needs to wake up [4]–[6]. However, PSM and its variants are only suitable for the scenario where a station has no data packets to transmit. When a station needs to transmit data, its WiFi NIC has to stay in active state to contend for the channel. Therefore, how to reduce WiFi NIC's energy consumption during channel contention periods is still a challenging task.

Packet overhearing accounts for a significant fraction of the total energy consumption during channel contention [7] [8]. In the IEEE 802.11 standards, the medium access control (MAC) scheme is distributed coordination function (DCF) where stations contend for the channel using carrier sense multiple access with collision avoidance (CSMA/CA). Specifically, a station randomly selects a backoff counter to conduct backoff procedure before transmitting. The backoff counter decrement process may be interrupted by other stations' packet transmissions. In this case, the station freezes its backoff counter and stays in the active state because it itself may be the intended receiver. As wireless channel has inherent broadcast nature, the station can overhear other stations' packet transmissions even if it is not the intended receiver. In the current WiFi standards, a station needs to receive and decode the whole MAC frame[1] before getting the destination address of the MAC frame because there is no individual checksum field for the MAC header and the common cyclic redundancy check (CRC) locates at the end of a MAC frame [8]. Thus, the station will waste energy to decode received packets at the physical layer even if it is not the intended receiver. Since the backoff freezing probability increases with the number of stations in WiFi networks, the unnecessary energy waste induced by packet overhearing becomes more significant.

To address the packet overhearing problem and avoid unnecessary energy consumption, an intuitive solution is to let unintended receivers stop receiving and processing a useless packet, as early as possible. Moreover, if unintended receivers have the knowledge of the packet transmission duration, they can switch to a low power state and return to active state before the packet transmission ends. Therefore, we need effective schemes to let the unintended receivers learn about the extra information, destination address and packet transmission duration as early as possible. Previous works [8] [9] add new control fields in the MAC frame header so that the unintended receivers only decode the MAC frame header to get the extra information, which changes the existing MAC frame structure. Legacy 802.11 stations cannot understand the modified MAC frames. With this method, unintended receivers may not be able to correctly decode the MAC frame header because they

B. Feng and C. Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, P. R. China (e-mail: fengice@mail.ustc.edu.cn, chizhang@ustc.edu.cn).

H. Ding and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida 32611, USA (e-mail: dhcbit@gmail.com, fang@ece.ufl.edu).

A preliminary version of this paper appeared in IEEE ICC'18 [1].

[1] Throughout this paper, we use "MAC frame" and "data packet" interchangeably.

may have worse channel conditions than the intended receiver while the data rate of the MAC frame is selected according to the intended receiver's channel condition. Adopting dedicated control packets [10] or adding a new physical preamble in front of the MAC frame [2] [9] can ensure robust transmission of the extra information to unintended receivers, but these methods are at the cost of consuming extra channel resources.

The above observations motivate us to propose a novel communication scheme so that the aforementioned extra information can be conveyed concurrently with the unicast data packet transmission without consuming extra channel resources. This communication scheme is referred to as DSS (Data Symbol Silence). Specifically, DSS intentionally erases some data symbols by loading zero transmission power in the front part of a data packet at the physical layer. The data symbols erased by DSS are referred to as silent data symbols. Unlike the traditional schemes, DSS embeds the extra information into the intervals between adjacent silent data symbols. Notice that DSS does not modify the existing data packet structure defined in the IEEE 802.11 standards. Legacy 802.11 stations can communicate with stations employing DSS but ignore the embedded information.

With DSS, stations only need to receive the front part of data transmission at the physical layer to extract the embedded information by data symbol level energy detection without decoding the whole data packet. Thus, unintended receivers can immediately abort useless packet receiving and processing and switch to a low-power state to reduce unnecessary energy waste. Different from the traditional broadcast scheme that transmits dedicated broadcast packet at the lowest data rate, the broadcast transmission achieved by DSS is overlaid on top of the unicast data packet transmission, but the interpretation of the embedded information in DSS is as robust as using dedicated broadcast packet because physical layer energy detection requires lower signal-to-noise ratio (SNR) than packet decoding. Wireless broadcast advantage is the feature that the signal transmitted by a station is received by all stations due to the broadcast nature of wireless channel [11], which enables neighboring stations to receive the embedded information.

In DSS, the silent data symbols are data symbol errors. To ensure the correct decoding of the original data packet, we exploit the SNR gap in WiFi transmissions to design DSS. In WiFi networks, the most suitable data rate of a data packet is selected to fight against transmission error according to the intended receiver's channel SNR. However, the number of available data rates is limited in a practical wireless communication system. As a result, the practical data rate adjustment is discrete while the channel SNR is continuous. Each data rate has a corresponding minimum SNR required for selecting the data rate. Thus, there exists an SNR gap between the actual channel SNR and the minimum required SNR of the selected data rate, which results in the number of bit errors induced by wireless transmission is less than what the correcting capability of the adopted channel coding can correct. Thus, the redundancy in the channel coding is under-utilized, which can be exploited to correct the silent data symbols that are intentionally introduced by DSS.

The free communications provided by DSS is opportunistic because the value of the SNR gap varies with various channel conditions and depends on the instantaneous channel condition. Based on the knowledge of the intended receiver's channel SNR, the station deployed with DSS decides whether to adopt DSS before transmitting. DSS is adopted as long as the current SNR gap ensures DSS does not harm the correct decoding of the original data packet. If the SNR gap is too small to adopt DSS, the station deployed with DSS works as a standard WiFi station without using DSS. Notice that the WiFi stations deployed with DSS can coexist with the existing WiFi stations that treat embedded information as noise and work as usual. Thus, the communication scheme provided by DSS is transparent to the existing WiFi stations, and DSS can be directly integrated into the existing WiFi networks, which maintains backward compatibility.

In this paper, the minimum SNR required for applying DSS is theoretically analyzed. We evaluate the energy-saving performance of DSS by simulations, and our results show that compared with the IEEE 802.11a standard, DSS significantly reduces energy consumption in packet overhearing.

The contributions of this paper are summarized as follows:

- We present an energy model for the IEEE 802.11 DCF to show packet overhearing is a major source of the total energy consumption during channel contention.
- We design DSS targeting at creating lightweight broadcast communications on the unicast data packet transmission to reduce energy consumption induced by packet overhearing without consuming extra channel resources.
- We theoretically analyze the minimum SNRs required for applying DSS in various channel conditions and implement DSS. We also conduct simulations to show that DSS significantly reduces energy waste compared with the IEEE 802.11 standards.

The rest of this paper is organized as follows. Section II presents our proposed energy model for the IEEE 802.11 DCF. Section III presents an overview of DSS. Section IV describes the detailed design of DSS. Section V theoretically analyzes the minimum SNR required for designing DSS. Section VI evaluates DSS. Section VII reviews the related work. Finally, Section VIII concludes this paper.

II. Motivations

In this section, we motivate DSS by showing the impact of packet overhearing on a station's energy spent on channel contention. An accurate energy model is proposed to decompose the energy consumption into different components.

A. An Accurate Energy Model

Existing energy models [12] [13] for the IEEE 802.11 DCF ignore the fact that there may be multiple channel activities within a backoff freezing process, which fails to accurately model the backoff freezing process where packet overhearing occurs. Following the existing Markov models [14] [15], an accurate energy model is proposed to investigate the energy consumption caused by packet overhearing.

We consider a WiFi network with N saturated stations including one tagged station and $N-1$ non-tagged stations.

All stations only have uplink transmissions without receiving packets from the access point (AP). According to DCF employing truncated binary exponential backoff, in the ith backoff stage, the contention window size W_i is defined as

$$W_i = \begin{cases} 2^i W_0, & i = 0, ..., m-1, \\ 2^m W_0, & i = m, ..., L, \end{cases} \tag{1}$$

where W_0 is the initial contention window size and L is the maximum retry threshold before dropping a packet.

As in [14] [15], the probability τ that a station transmits is

$$\begin{aligned} \tau =& \frac{1-p^L}{\left[\sum_{i=0}^{L}\left(1+\frac{1}{1-p}\sum_{k=1}^{W_i-1}\frac{W_i-k}{W_i}\right)p^i\right](1-p)} \\ =& \frac{1-p^L}{\left[\sum_{i=0}^{L}\left(1+\frac{W_i-1}{2(1-p)}\right)p^i\right](1-p)}, \end{aligned} \tag{2}$$

where p is the collision probability. The p also represents the backoff freezing probability that the tagged station that is being in backoff procedure freezes its backoff counter due to other stations' channel activities. Given τ, we have [14]

$$p = 1-(1-\tau)^{N-1}. \tag{3}$$

A function is established by equations (2) and (3), and we can numerically solve τ and p by searching for the fixed points of this function. Denote $1-(1-\tau)^{N-1}$ in (3) as $g_p(\tau)$. The convergence of the fixed point iteration is proved in [16] [17] by showing that $g_p'(\tau)$ is bounded, where $g_p'(\tau)$ is the derivative of $g_p(\tau)$ with respect to p.

Because of the $N-1$ non-tagged stations' channel activities, the tagged station's backoff counter decrement process is interrupted (i.e., backoff freezing) by a successful transmission or a collision. We define p_s and p_c as the probability that the backoff freezing is induced by a successful transmission and a collision, respectively. Knowing $p = p_s + p_c$, we have $p_s = \binom{N-1}{1}\tau(1-\tau)^{N-2}$ and $p_c = p - p_s$. Within a backoff freezing process where the tagged station freezes its backoff counter, the channel activities (successful transmissions and collisions) of non-tagged stations contribute to the tagged station's energy consumption due to packet overhearing. The energy waste can be significantly large if interruptions to the tagged station's backoff procedure occur frequently.

The backoff freezing process caused by a successful transmission means that exactly one non-tagged station is involved in the successful transmission. After the non-tagged station successfully transmits, it will randomly select a backoff counter from $[0, W_0-1]$. In this case, it may have another successful transmission if its newly backoff counter is zero, which is with the probability $p_{ss} = \frac{1}{W_0}$. Thus, the average number of successive successful transmissions during such a backoff freezing process is $\frac{1}{1-p_{ss}}$. Let E_{os} denote the energy consumed by the tagged station to overhear a successful transmission not involving itself. The total energy that the tagged station consumes during such a backoff freezing process is

$$A_s = \frac{1}{1-p_{ss}}E_{os}. \tag{4}$$

On the other hand, the backoff freezing process caused by a collision means that at least two non-tagged stations are involved in the collision. After the collision ends, there may be other channel activities during such a backoff freezing process, which depends on the newly selected backoff counters of the colliding stations. If only one of the colliding stations selects zero as its new backoff counter, there is a successful transmission after the collision, and the corresponding probability is denoted by p_{cs}. Given k colliding stations, the probability that a colliding station selects zero as its new backoff counter is $\delta = \sum_{j=0}^{L}\frac{1}{W_j}\eta(j)$ where $\eta(j) = p^j(1-p)/(1-p^{L+1})$ is the probability that a packet transmits successfully after the jth retry. Obviously, $p_{cs}(k) = \binom{k}{1}\delta\,(1-\delta)^{k-1}$. Then, we get $p_{cs} = \sum_{k=2}^{N-1}P(k)p_{cs}(k)$, where $P(k) = \binom{N-1}{k}\tau^k(1-\tau)^{N-1-k}$ is the probability that k out of $N-1$ non-tagged stations are involved in the collision. If all colliding stations do not select zero as their new backoff counters, the channel becomes idle (no transmissions) after the collision, and the corresponding probability is $p_{ci} = \sum_{k=2}^{N-1}P(k)p_{ci}(k)$ where $p_{ci}(k) = (1-\delta)^k$. If two or more out of the colliding stations select the same backoff counter, there is another collision after the collision, and the corresponding probability is $p_{cc} = \sum_{k=2}^{N-1}P(k)(1-p_{cs}(k)-p_{ci}(k))$.

Within the backoff freezing process caused by a collision, the average number of successive collisions and successful transmissions are $\frac{1}{1-p_{cc}}$ and $\frac{p_{cs}}{1-p_{cc}}\frac{1}{1-p_{ss}}$, respectively. Let E_{oc} denote the energy consumed by the tagged station to overhear a collision not involving itself. The total energy that the tagged station consumes during such a backoff freezing process is

$$A_c = \frac{1}{1-p_{cc}}E_{oc} + \frac{p_{cs}}{1-p_{cc}}\frac{1}{1-p_{ss}}E_{os}. \tag{5}$$

Let $U(j) = \sum_{i=0}^{j}\frac{W_i-1}{2}$ be the average number of idle backoff slots that the tagged station encounters before its packet is transmitted successfully at the jth retry. Based on (4) and (5), we derive the average energy spent on channel contention as

$$\begin{aligned} E =& \left(\frac{1}{1-p_{cc}}E_{oc} + \frac{p_{cs}}{1-p_{cc}}\frac{1}{1-p_{ss}}E_{os}\right)p_c\sum_{j=0}^{L}\eta(j)U(j) \\ &+ \frac{1}{1-p_{ss}}E_{os}p_s\sum_{j=0}^{L}\eta(j)U(j) \\ &+ E_i\sum_{j=0}^{L}\eta(j)U(j) \\ &+ E_{tc}\sum_{j=0}^{L}j\eta(j) + E_{ts}, \end{aligned} \tag{6}$$

where E_i, E_{tc}, and E_{ts} are the energy that the tagged station spends on an idle backoff slot, a collision, and a successful transmission involving the tagged station, respectively, and $\sum_{j=0}^{L}j\eta(j)$ is the average number of collisions before a packet of the tagged station is transmitted successfully.

In (6), $E_i\sum_{j=0}^{L}\eta(j)U(j)$, $E_{tc}\sum_{j=0}^{L}j\eta(j)$, and E_{ts} represent energy consumptions of the channel activities involving the tagged station. The $N_i = \sum_{j=0}^{L}\eta(j)U(j)$ is the number of idle backoff slots that the tagged station encounters

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2897399, IEEE Transactions on Vehicular Technology

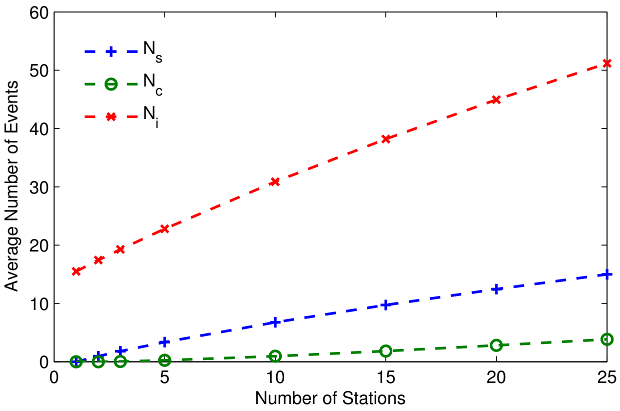


Fig. 1. The numbers of various interruptions within a tagged station's backoff procedure.

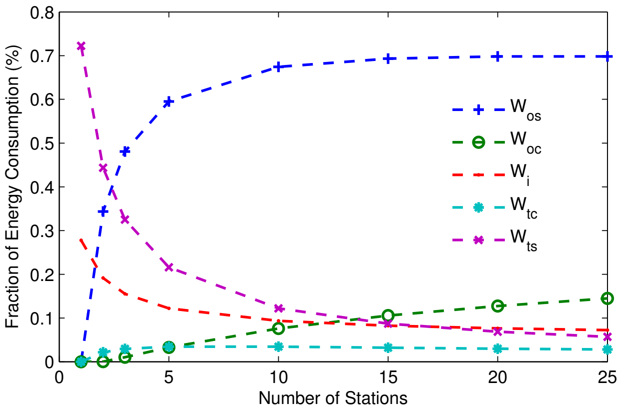


Fig. 2. The fractions of various energy consumption components within the backoff procedure used by the tagged station to successfully transmit a packet.

before its packet is transmitted successfully. The numbers of interruptions caused by collisions and successful transmissions not involving the tagged station are $N_c = p_c \sum_{j=0}^{L} \eta(j)U(j)$ and $N_s = p_s \sum_{j=0}^{L} \eta(j)U(j)$, respectively. Because of packet overhearing, the tagged station's extra energy consumption $A_c p_c \sum_{j=0}^{L} \eta(j)U(j) + A_s p_s \sum_{j=0}^{L} \eta(j)U(j)$ is wasted to receive and process useless packets not addressed to it, which is caused by the $N-1$ non-tagged stations' channel activities and can be decomposed into two parts. The one caused by overhearing successful transmissions is

$$B_{os} = \left(\frac{p_{cs}}{1-p_{cc}} \frac{1}{1-p_{ss}} p_c + \frac{1}{1-p_{ss}} p_s \right) E_{os} \sum_{j=0}^{L} \eta(j)U(j), \tag{7}$$

and the other one caused by overhearing collisions is

$$B_{oc} = \frac{1}{1-p_{cc}} E_{oc} p_c \sum_{j=0}^{L} \eta(j)U(j). \tag{8}$$

B. Analyzing Energy Waste in Packet Overhearing

In this subsection, we exploit the proposed energy model to analyze energy consumption components based on the power consumption parameters of a typical Atheros WiFi NIC [8]. Our results show that the most energy-consuming event is overhearing other stations' successful transmissions.

In Fig. 1, we plot N_i, N_s, and N_c under various numbers of stations. More stations in a WiFi network means more successful transmissions not involving the tagged station, so there exist more interruptions within a backoff procedure of the tagged station. We can clearly observe that N_s increases with the number of stations. In a WiFi network with 15 stations, the backoff procedure of the tagged station is interrupted 9.7 times on average due to non-tagged stations' successful transmissions before it successfully transmits a packet.

Let W_{os} and W_{oc} denote the ratios of the energy consumption caused by overhearing successful transmissions and collisions not involved with the tagged station to the total energy consumption, respectively. Let W_i, W_{tc}, and W_{ts} denote the ratios of the energy consumption caused by idle backoff slots, successful transmissions and collisions involved with the tagged station to the total energy consumption, respectively. Fig. 2 plots the ratios of various energy consumption components to the total energy consumption under various numbers of stations. Fig. 2 confirms that the energy spent on channel contention is dominated by W_{os}. For example, when the number of stations is 15, each station spends 69.31% of the total energy consumption on overhearing other stations' successful transmissions while the fraction of energy spent on overhearing collisions is the second source of energy consumption and accounts for less than 10.56% of the total energy consumption. Since it is difficult to extract information from a colliding packet, DSS focuses on reducing energy consumption caused by overhearing successful transmissions.

III. DSS Design

A. Overview of DSS

In the current WiFi standards, the physical layer technique adopts orthogonal frequency-division multiplexing (OFDM). The design of DSS is based on OFDM that divides the wireless channel into orthogonal subcarriers. For example, according to the IEEE 802.11a standard, a wireless channel includes 64 orthogonal subcarriers, of which 48 subcarriers are data subcarriers to carry data symbols in parallel. As shown in Fig. **??**, the 48 data subcarriers are numbered logically from 1 to 48. With OFDM, wireless channel resources are divided into time-frequency resource blocks and the smallest resource unit is a data symbol generated by a modulation scheme such as BPSK, QPSK, or 16QAM. The duration of a data symbol, i.e., a time slot, is 4 μs in IEEE 802.11a. In Fig. **??**, a regular data symbol is a complex value in the constellation diagram of a modulation scheme.

To embed extra information into a data packet, DSS intentionally erases a controlled number of data symbols on 48 data subcarriers. A data symbol (regular data symbol) erased by loading zero transmit power is called a silent data symbol and can be located with data symbol level energy detection. In Fig. 3, a data symbol is denoted by $S_{i,j}$ where i is the time slot index and j is the subcarrier index. $S_{1,1}$ is a silent data symbol indicating the start of embedded information. The number of regular data symbols between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2897399, IEEE Transactions on Vehicular Technology

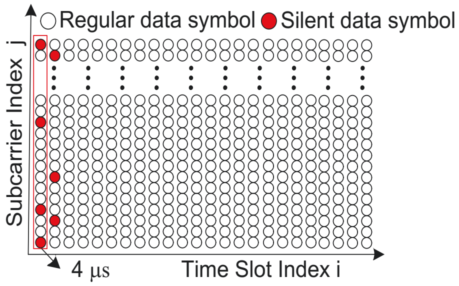


Fig. 3. The illustration of DSS based on the OFDM-based IEEE 802.11a. In DSS, the intervals between silent data symbols are used to encode information. DSS only needs two time slots (8 μs) to embed information.

two adjacent silent data symbols is defined as the interval length that is used to encode information. We embed 3 bits into an interval, so the maximum interval length is 7. As shown in Fig. 3, the interval length between silent data symbols $S_{1,4}$ and $S_{1,12}$ is 7, which means that 3 bit "111" is embedded into the corresponding interval. To convey the 12 bit "001011111101" message, we divide it into four groups where $\{$"001" $\rightarrow$ 1, "011" $\rightarrow$ 3, "111" $\rightarrow$ 7, "101" $\rightarrow$ 5$\}$. Through detecting silent data symbols, receivers interpret the embedded information at the physical layer without decoding data packets. Notice that DSS does not utilize the physical preamble to embed information because the physical preamble is important for OFDM frequency synchronization. In the practical design, DSS can start from the physical layer header of data packets to embed information.

With DSS, the transmitter of a data packet embeds two extra information, destination address and duration of the data packet transmission, into the front part of the data packet. To shorten the length of the destination address information, we adopt the 15 bit association identifier (AID) instead of the original 48 bit MAC address to logically identify a station in the WiFi network. In the scenario of multiple WiFi networks, stations may overhear useless data packets from neighboring WiFi networks. The last 4 bits of the AID is used to identify different WiFi networks. When the association request of a station to the access point (AP) is granted, the station is assigned a unique AID. The AP maintains the association list to ensure each associated station has a distinct AID value and updates it whenever a station joins or leaves the WiFi network. We use 15 bits to specify the duration of data packet transmission in μs. If the WiFi networks support the Transmission Opportunity (TXOP) mechanism where a station can transmit multiple data packets till the end of TXOP, we can use the 15 bits to specify the TXOP duration instead of the duration of a single data packet transmission. In addition, an individual 8 bit CRC is added to check the correctness of the embedded information.

The total message embedded by DSS includes 38 bits. Because we use 3 bits to encode an interval, the total message with 38 bits can always be encoded using 13 intervals, which requires inserting 14 silent data symbols into the front part of the data packet. Since the number of available data symbols within a time slot is 48 while the maximum interval length is 7, at most two time slots (i.e., 8 μs in IEEE 802.11a) are needed to achieve 13 intervals. In other words, receivers only need to receive the front part of the data packet (two time slots) to extract embedded information at the physical layer. Unintended receivers can quickly abort useless packet receiving and processing. Based on the knowledge of the data packet transmission duration, unintended receivers can decide whether to switch to a sleep state. If the data packet transmission duration is large (more than a threshold), unintended receivers can switch to sleep state, and they will switch back to an active state before the data packet transmission ends. Otherwise, they stay in the idle state. With DSS, the unintended receivers' energy consumption caused by packet overhearing can be reduced.

TABLE I
DATA RATES DEFINED IN IEEE 802.11A.

Min Required SNR	Data Rate	Modulation	Coding Rate
3.5 dB	6 Mbps	BPSK	1/2
4.5 dB	9 Mbps	BPSK	3/4
5 dB	12 Mbps	QPSK	1/2
9.5 dB	18 Mbps	QPSK	3/4
12 dB	24 Mbps	16QAM	1/2
17.5 dB	36 Mbps	16QAM	3/4
21 dB	48 Mbps	64QAM	2/3
22 dB	54 Mbps	64QAM	3/4

B. Channel Coding Redundancy

In DSS, the inserted silent data symbols are data symbols erased deliberately by loading zero transmission power. The data symbol erasing process is equivalent to intentionally adding a controlled number of bit errors into the original data packet at the transmitter side. The design of DSS should ensure the original data packet can be decoded correctly at the intended receiver side while not affecting the achievable data rate. We will show how to exploit the wasted channel coding redundancy existing in practical WiFi systems to recover the data symbols damaged by the inserted silent data symbols.

In wireless communications, data rate is determined by adopted modulation and channel coding schemes. A lower data rate means the adoption of a lower order modulation scheme and a channel coding with lower coding rate. Based on the SNR at the intended receiver, rate adaption scheme selects the maximum data rate while ensuring reliable transmissions. Due to hardware constraints, the number of data rates is limited in a practical communication system, whereas SNR is continuous. As a result, we cannot make a perfect one-to-one matching from SNRs to data rates, which results in a stair-case rate adjustment. A data rate can be selected when the SNR is not lower than its minimum required SNR. Thus, there exists an SNR gap between the actual channel SNR and the minimum required SNR of the selected data rate.

The IEEE 802.11a standard specifies eight available data rates and corresponding minimum required SNRs, which is shown in Table I. If the SNR at the intended receiver is 14 dB, falling between 12 dB and 17.5 dB, the transmitter selects the data rate of 24 Mbps according to Table I. Since the minimum required SNR for selecting 24 Mbps is 12 dB in Table I, the SNR gap is 2 dB. The 24 Mbps data rate

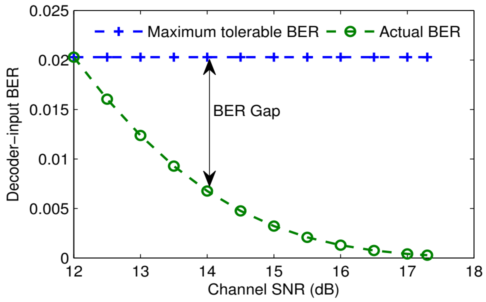


Fig. 4. Given data rate of 24 Mbps corresponding to the SNR range from 12 dB to 17.5 dB, the BER gap between the maximum tolerable BER and the actual BER.

is a result of 16QAM modulation and $1/2$ channel coding rate. During 16QAM demodulation stage, lower SNR results in higher bit error rate (BER). Given the selected data rate of 24 Mbps, the maximum tolerable BER that the channel coding can handle is obtained at 12 dB corresponding to the minimum required SNR. To better understand the relationship between the SNR gap and BER, Fig. 4 plots the decoder-input BERs in the SNR range from 12 dB to 17.5 dB. For the channel SNR of 14 dB, we can clearly observe the BER gap between the maximum tolerable BER and the BER at 14 dB due to the SNR gap, which means the actual number of bit errors induced by wireless transmission is less than what the channel coding can correct. In other words, the channel coding redundancy is under-utilized and the decoder can tolerate more bit errors. Therefore, DSS can exploit the SNR gap that results in under-utilized correction capability of the existing channel coding to correct the inserted silent data symbols.

In DSS, we intentionally corrupt 14 data symbols within a data packet. To correctly decode the original data packet, the SNR gap should ensure the correction capability of the channel coding can handle the total bit errors including the errors induced by our inserted silent data symbols and the errors naturally incurred by wireless transmissions. Therefore, given a data rate, there exists a threshold to opportunistically take advantage of the SNR gap to convey free messages without consuming extra channel resources. If the SNR at the intended receiver satisfies the threshold, the transmitter deployed with DSS embeds extra information to support the function of energy efficient packet overhearing. Otherwise, it just works as the standard WiFi station without supporting DSS. In Section V, we will present the thresholds of SNRs, i.e., the minimum SNRs required for applying DSS, for various data rates based on our theoretical analysis.

IV. Detailed Design

In this section, we present the overall system architecture of DSS and the detailed implementations.

A. Overall System Architecture

The overall system architecture of DSS is illustrated in Fig. 5. DSS does not change the architecture of the 802.11 standards above the physical layer but adds new components to embed and extract extra information at the physical layer.

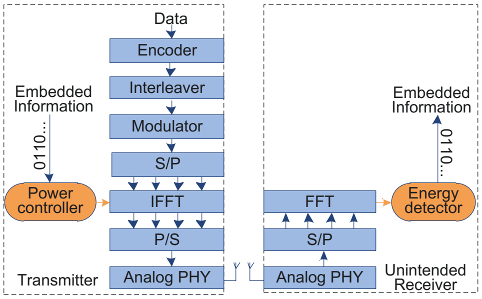


Fig. 5. Architecture of DSS based on the IEEE 802.11a.

In Fig. 5, the power controller module at the transmitter side embeds extra information by controlling the interval lengths between the inserted silent data symbols that are constructed based on the Inverse Fast Fourier Transform (IFFT). Because of packet overhearing, all neighboring stations including the intended receiver in the WiFi network can receive the unicast data packet. Initially, receivers conduct information extraction and packet reception in parallel. Once receivers extract and interpret the embedded information at the physical layer and know the destination address, the unintended receivers immediately abort the remaining packet receiving and processing while the intended receivers work as standard WiFi stations to receive and process the whole data packet. Extracting the embedded information is achieved with the data symbol level energy detection that is realized by the Fast Fourier Transform (FFT). With DSS, unintended receivers reduce useless energy consumption in the RF-front end and baseband processing, such as demodulation, de-interleaving, and decoding.

B. Construction and Detection of Silent Data Symbols

In an OFDM system, the IFFT operation transforms N data symbols in the frequency domain into time domain OFDM signals. Let $X[n]$ be the data symbol carried on n-th data subcarrier, for $n = 0, 1, ..., N-1$ where N is the number of data subcarriers ($N = 48$ in the IEEE 802.11a standard). The vector $\mathbf{X} = [X[0], X[1], ..., X[N-1]]$ is the IFFT input. Thus, an OFDM symbol $x[k]$ at sample time k is obtained by

$$x[k] = \frac{1}{N}\sum_{n=0}^{N-1} X[n]e^{j2\pi\frac{kn}{N}}, \quad k = 0, 1, ..., N-1 \tag{9}$$

In the regular IFFT operation, the data symbol $X[n]$ carried on n-th data subcarrier is a complex value. For example, $X[n]$ is $1+0i$, $0+i$, $-1+0i$ or $0-i$ in the constellation diagram of the QPSK modulation. If a silent data symbol is inserted into the n-th data subcarrier, $X[n]$ is set to 0 rather than the complex value of the data symbol, which deactivates the corresponding data subcarrier and results in zero transmit power on the n-th data subcarrier. Thus, all inactive data subcarriers do not carry actual signal and remain silent.

At the receiver, the OFDM modulation is inverted by the FFT operation that transforms the received time domain OFDM signal into N data symbols in the frequency domain. The FFT result reveals energy distribution of the received signal in the frequency domain, i.e., among different data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2897399, IEEE Transactions on Vehicular Technology

subcarriers. We can use a fairly simple method to determine the locations of silent data symbols by analyzing the energy distribution on different data subcarriers of the received OFDM signal. On a data subcarrier carrying a regular data symbol, i.e., an active data subcarrier, high energy will be detected. Since an inactive data subcarrier only contains noise without an actual signal, the detected energy will not be zero, but it is very small compared with an active data subcarrier. In a constellation diagram, the received data symbols are silent data symbols if they fall around the constellation position of $0 + 0i$ and are regular data symbols if they fall around their corresponding constellation positions. Thus, locating silent data symbols can be realized with data symbol level energy detection.

The design of DSS does not affect the frequency synchronization of the OFDM system, so the data subcarriers are still orthogonal. In DSS, the power of a data subcarrier carrying regular data symbols does not leak energy into adjacent data subcarriers carrying silent data symbols, which ensures accurate detection of silent data symbols.

C. Energy Detection Threshold

The detailed data symbol level energy detection in the frequency domain is implemented by using threshold test to identify the presence of silent data symbols. The energy detection threshold is set to the noise floor. Since the noise floor varies in various wireless scenarios, this threshold should be adaptively chosen by sampling the noise floor. Therefore, we propose an estimation scheme to estimate the noise floor.

In the IEEE 802.11a standard, there are 4 pilot subcarriers that are distributed evenly across all data subcarriers. The pilot subcarriers carry known data symbols. We extract the noise floor by the pilot subcarriers to set the detection threshold. In practical situations, the threshold value is slightly larger than the estimated value. The energy of a regular data symbol is much larger than the noise floor. If the energy of a data subcarrier is significantly larger than the threshold, we declare it as an active data subcarrier carrying a regular data symbol. Otherwise, it carries a silent data symbol.

D. Erasure Viterbi Decoding

DSS is an opportunistic communication scheme. If the SNR gap is too small to ensure correct packet decoding at the intended receiver, we do not use DSS. To create more opportunities to use our proposed communication scheme, we can apply a more powerful decoding scheme to enhance decoding performance so that we can still insert silent data symbols even if the SNR gap is small.

In the IEEE 802.11 standards, the channel coding at the physical layer is convolutional code and the decoding adopts Viterbi algorithm. Since the existing Viterbi algorithm is an error-only decoding scheme, it simply treats silent data symbols as erroneous symbols induced by wireless fading even though they result from erasing the original data symbols. As it is well-known, more erasures than errors can be corrected by the forward error correction code. Previous works [18] [19] have shown that with the side information of the reliability of data symbols, erasing unreliable data symbols prior to the decoding process improves decoding performance. In DSS, the inserted silent data symbols are low-reliability data symbols. Thanks to data symbol level energy detection that provides idle erasure information, we can design error-and-erasure decoding without extra overhead.

The erasure Viterbi decoding (EVD) scheme is proposed to incorporate erasure decoding into the conventional Viterbi decoding algorithm. Let x_i be the i-th transmitted data symbol. For the data symbol x_i modulated by M-ary constellation, it contains $m = \log_2 M$ bits. Let $d_i^j = b (b \in \{0, 1\})$ denote the j-th bit contained in x_i, for $j = 1, 2, ...m$. At the receiver, the i-th received data symbol is y_i after the FFT operation. Based on the detection of silent data symbols, we use erasure indicator e_i for i-th received data symbol to indicate whether it is a silent data symbol. The $e_i = 0$ means y_i is marked as a silent data symbol while $e_i = 1$ means y_i is a regular data symbol. With the pair (y_i, e_i) indicating y_i with e_i, the demodulator calculates bit metric for every bit $d_i^j = b$ by

$$\lambda(d_i^j = b) = \begin{cases} \log P(y_i | d_i^j = b), & \text{if } e_i = 1 \\ 0, & \text{if } e_i = 0 \end{cases} \tag{10}$$

where the bit metrics of the bits contained in a silent data symbol ($e_i = 0$) are zero (i.e., $\lambda(d_i^j = b) = 0$) while the bit metrics of the bits contained in a regular data symbol ($e_i = 1$) are calculated by a log likelihood function. As in [20] [21], the log likelihood function is

$$\begin{aligned} \log P(y_i | d_i^j = b) &= \log \sum_{x_i \in \chi_b^i} P(y_i | x_i) \\ &\approx \max_{x_i \in \chi_b^j} \log P(y_i | x_i), \end{aligned} \tag{11}$$

Let Φ denote a signal set including all constellation points in the M-ary constellation. The size of Φ is $M = 2^m$. χ_b^j is the subset of Φ whose j-th bit d_i^j is $b \in \{0, 1\}$.

Notice that the de-interleaver at the receiver breaks correlation among data bits contained in the same data symbol [20]. Thus, the de-interleaving makes data bits contained in a silent data symbol spread across various positions in a codeword. In our erasure Viterbi decoding, the key operation is to mark silent data symbols as erasures and set the bit metrics of the bits contained in silent data symbols to zero. Because of the detection of silent data symbols, the proposed erasure decoding avoids the complexity of deciding which data symbols are to be erased. In addition, we only modify the bit metric calculation in the existing Viterbi decoding procedure. Therefore, the proposed error-and-erasure decoding scheme can be directly built upon the existing Viterbi decoder.

E. Selection of Low-Power State

Within the packet transmission duration, the unintended receivers can switch to the idle or sleep state. The energy consumption of the sleep state is significantly lower than that of the idle state. However, switching between the sleep state and the active state (transmitting, receiving or idle) consumes extra time and energy. With the knowledge of packet transmission duration, the unintended receivers can obtain the remaining packet transmission duration and decide whether to switch to the sleep state. Notice that various WiFi NICs provided by various vendors have different hardware parameters. The time for switching from receiving state to sleep state may range from several microseconds to a few hundreds of microseconds. Therefore, stations should individually make their own decisions to select a low-power state (idle or sleep).

Let $T_{s,a}$ and $T_{a,s}$ denote the time for switching from sleep state to active state and from active state to sleep state, respectively. Given the condition that the remaining packet transmission duration T_t satisfies $T_t > T_{a,s} + T_{s,a}$, the unintended receivers calculate which state (idle or sleep) consumes less energy. Let $P_{s,a}$ and $P_{a,s}$ denote the power consumption spent on switching from sleep state to active state and that spent on switching from active state to sleep state, respectively. Then, if an unintended receiver switch to sleep state, the corresponding energy consumption is estimated by

$$E_{sleep} = (T_t - T_{a,s} - T_{s,a}) * P_s + T_{a,s} * P_{a,s} + T_{s,a} * P_{s,a}, \quad (12)$$

where P_s is the power consumption of sleep state. Since the idle state is one of the active states, the time and energy consumption for switching between an idle state and an active state are negligible. Thus, the energy consumption for staying in the idle state within the remaining packet transmission duration is estimated by

$$E_{idle} = T_t * P_i, \quad (13)$$

where P_i is the idle state's power consumption. The following conditions should be satisfied to switch to sleep state,

$$\begin{cases} T_t > T_{a,s} + T_{s,a} \\ E_{sleep} < E_{idle} \end{cases} \quad (14)$$

V. Theoretical Analysis

In this section, we theoretically analyze the decoder-output BER of our proposed communication scheme, DSS, that inserts 14 silent data symbols into a data packet. Then, we present the minimum SNRs required for designing DSS. Our conference paper [1] does not present theoretical analysis to show the feasibility of DSS. As shown in Table I, we consider all data rates specified in the IEEE 802.11a standard.

A. Bit Error Rate

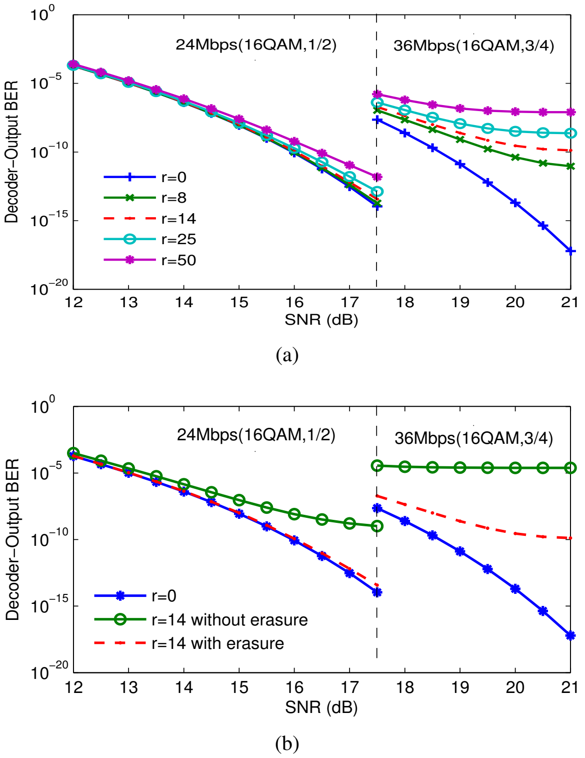
Without erasure decoding, the union bound on the BER of a $R = k/n$ convolutional code with Viterbi decoding can be obtained in [22]. Let $P(d)$ be the pairwise error probability that the Viterbi decoding selects an incorrect path having Hamming weight d. As in [23], $P(d)$ is given by

$$P(d) = \begin{cases} \sum_{i=(d+1)/2}^{d} \binom{d}{i} p^i q^{d-i}, & d \text{ odd} \\ \frac{1}{2}\binom{d}{d/2} p^{d/2} q^{d/2} + \sum_{i=d/2+1}^{d} \binom{d}{i} p^i q^{d-i}, & d \text{ even} \end{cases} \quad (15)$$

where p is the decoder-input BER and $q = 1 - p$. Without inserting any silent data symbols, p only depends on the demodulation BER $\tilde{p}(s)$ that is a function of data symbol SNR denoted by s. Given r silent data symbols are inserted into a data packet of L Bytes, the ratio of the number of inserted silent data symbols to the number of total data symbols is $\theta = \frac{rN_{BPSC}}{8L}$, where N_{BPSC} is the number of bits per data symbol. Thus, p also depends on θ in DSS. For a specific modulation scheme, p is a function of θ and s, denoted as $p(\theta, s)$. We can calculate $p(\theta, s)$ by

$$p(\theta, s) = \theta\tilde{p}(0) + (1-\theta)\tilde{p}(s), \quad (16)$$

where $\tilde{p}(0)$ means the SNR of a silent data symbol is zero because it only contains noise, and Table II presents the values of $\tilde{p}(s)$ for the four modulation schemes specified in the IEEE 802.11a standard. Notice that $p(0, s)$ means DSS does not insert silent data symbols.



(a)

(b)

Fig. 6. Decoder-output BER $\overline{P}_b$ of DSS for two data rates. (a) Effect of the number of inserted silent data symbols r on $\overline{P}_b$. (b) Effect of erasure decoding on $\overline{P}_b$.

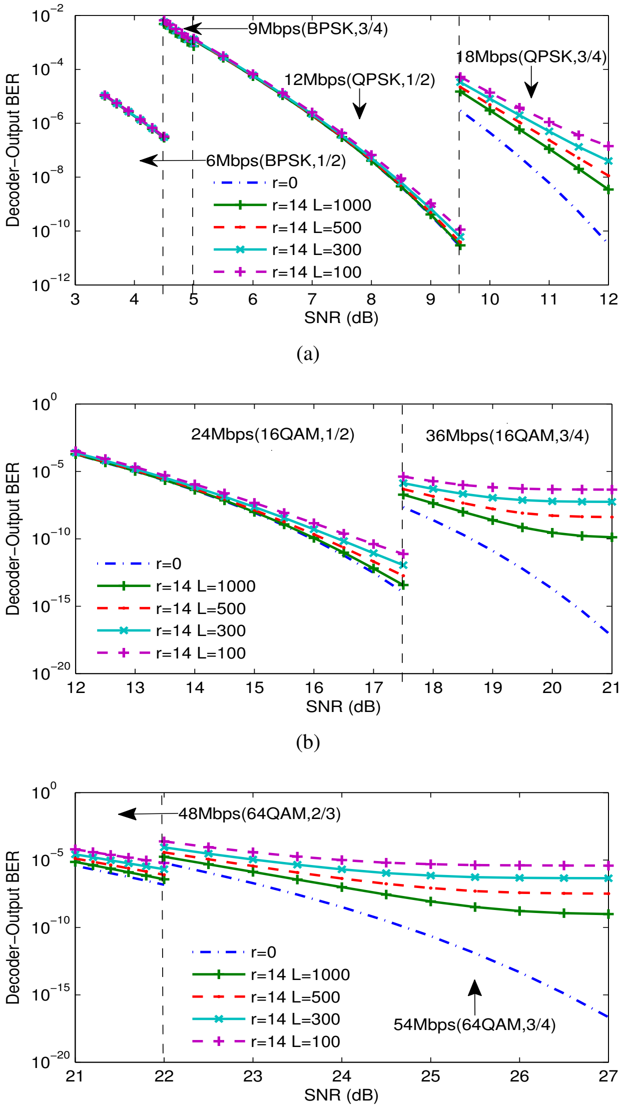
TABLE II
DEMODULATION BER $\tilde{p}(s)$ FOR VARIOUS MODULATION SCHEMES. Q IS THE Q FUNCTION.

Modulation	BPSK	QPSK	16QAM	64QAM
$\tilde{p}(s)$	$Q(\sqrt{2s})$	$Q(\sqrt{s})$	$\frac{3}{4}Q(\sqrt{s/5})$	$\frac{7}{12}Q(\sqrt{s/21})$

With erasure decoding, the bit metrics of the data bits contained in silent data symbols are set to zero, so the codeword having Hamming weight d can reduce its original Hamming weight to $d - e$ where e is the number of marked bits having bit metric value of zero. Given θ, the probability of reducing Hamming weight to $d - e$ is $\binom{d}{e}\theta^e(1-\theta)^{d-e}$. Accordingly, the $P(d)$ in (15) becomes [19] [20]

$$\overline{P}(d) = \sum_{e=0}^{d} \binom{d}{e} \theta^e (1-\theta)^{d-e} P(d-e). \quad (17)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2897399, IEEE Transactions on Vehicular Technology



(a)

(b)

(c)

Fig. 7. Decoder-output BER $\overline{P}_b$ as a function of SNR and L. (a) In data rates of 6Mbps, 9Mbps, 12Mbps, and 18Mbps. (b) In data rates of 24Mbps and 36Mbps. (c) In data rates of 48Mbps and 54Mbps.

Since the silent data symbols are erased during erasure decoding process, p is $p(0, s)$ for the calculation of $\overline{P}(d)$. The BER of a $R = k/n$ convolutional code with our proposed erasure decoding is bounded by

$$\overline{P}_b \leq \frac{1}{k} \sum_{d=d_f}^{\infty} B_d \overline{P}(d), \tag{18}$$

where d_f is the minimum free distance of the convolutional code and B_d is the total number of error bits of all incorrect paths having Hamming weight d. For the three coding rates specified in IEEE 802.11a, the values of d_f and B_d can be obtained from [24].

B. BER Analysis

Given a data rate, the decoder-output BER of our proposed DSS $\overline{P}_b$ depends on the number of inserted silent data symbols r and packet length L. Given $L = 1000$, Fig. 6(a) plots decoder-output BER $\overline{P}_b$ as a function of SNR for various values of r. For the data rate of 24Mbps corresponding to a combination (16QAM, 1/2), given an SNR, $\overline{P}_b$ slightly increases with r because decoder-input BER in (16QAM, 1/2) is dominated by demodulation BER. The increase of r has a slight effect of decoder-input BER. With the increase of SNR, demodulation BER reduces sharply. In contrast, when the data rate is 36Mbps corresponding to a combination (16QAM, 3/4), decoder-input BER is dominated by inserted silent data symbols and $\overline{P}_b$ significantly increases with r for a given SNR. Given a r value, the decreases of $\overline{P}_b$ in both (16QAM, 1/2) and (16QAM, 3/4) are due to the increase of SNR. The decrease of demodulation BER implies that more correcting capability of the channel coding is under-utilized, Thus, more silent data symbols can be corrected. However, we can observe that in (16QAM, 3/4), given r, $\overline{P}_b$ initially decreases with the increase of SNR. After SNR reaching a threshold, $\overline{P}_b$ decreases slightly or stops decreasing. The reason is that when demodulation BER decreases to zero due to large SNR and the correcting capability of the channel code has been totally used for the decoder-input BER induced by inserted silent data symbols, the increase of SNR does not further decrease $\overline{P}_b$.

Next, we compare our proposed erasure decoding with conventional Viterbi decoding that simply treats silent data symbols as erroneous symbols induced by wireless fading. Given $r = 14$ and $L = 1000$, Fig. 6(b) plots decoder-output BER $\overline{P}_b$ as a function of SNR for two decoding schemes. We can clearly observe that with the perfect erasure of silent data symbols, the proposed erasure decoding significantly outperforms the conventional Viterbi decoding and the performance gain of erasure decoding over conventional Viterbi decoding becomes larger at high SNR. Within a large SNR range, the decoder-output BER of the proposed erasure decoding is slightly degraded comparing to the case when $r = 0$.

Given r, $\overline{P}_b$ also depends on L that affects θ, the fraction of inserted silent data symbols within a data packet. Given $r = 14$, Fig. 7 plots decoder-output BER $\overline{P}_b$ as a function of SNR for various L. Smaller L means larger θ, so higher $\overline{P}_b$.

C. Minimum Required SNR

In DSS, we need to insert $r = 14$ silent data symbols to embed extra information while ensuring correct decoding of the original data packet. Given $r = 14$, we investigate the minimum SNR required for ensuring correct decoding of the original data packet when applying DSS. Fig. 7 plots decoder-output BER $\overline{P}_b$ as a function of SNR and L for eight data rates specified in IEEE 802.11a. For each data rate, the decoder-output BER of the original data packet that does not have silent data symbols (i.e., $r = 0$) at the minimum required SNR of the data rate is used as the performance metric to determine the minimum required SNR of applying DSS. To ensure that the correct decoding of the original data packet is not affected by the inserted silent data symbols, we should guarantee that the performance metric can be achieved at the intended receiver despite the insertion of silent data symbols. For example, in Fig. 7(a), the minimum required SNR of 18Mbps is 9.5dB, so

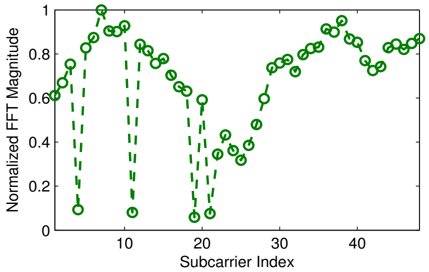


Fig. 8. Normalized FFT result of 48 data subcarriers with detected silent data symbols on data subcarriers [4,11,19,21].

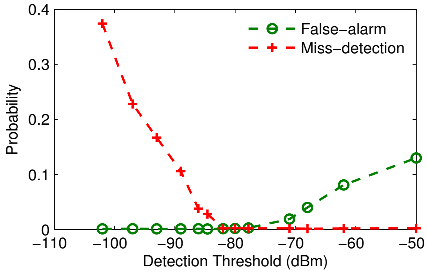


Fig. 9. The impact of detection threshold on P_f and P_m.

the value of $\overline{P}_b$ with $r = 0$ at 9.5dB, 2.98×10^{-6}, is selected as the performance metric for the SNR range corresponding to 18Mbps. Then, for $r = 14$ and $L = 1000$, when SNR is not lower than 10dB, the corresponding $\overline{P}_b$ is not higher than the performance metric of 2.98×10^{-6}. Similarly, given the data rate of 36Mbps of which the minimum required SNR is 17.5dB, for $r = 14$ and $L = 500$, the corresponding $\overline{P}_b$ should be not higher than the performance metric of 2.283×10^{-8}, which requires SNR to be not lower than 18.8dB.

TABLE III
MINIMUM REQUIRED SNRS FOR DESIGNING DSS WITH VARIOUS PACKET LENGTHS.

Data Rate	Minimum required SNR			
	L=100	L=300	L=500	L=1000
6 Mbps	3.51 dB	3.50 dB	3.50 dB	3.50 dB
9 Mbps	4.60 dB	4.57 dB	4.54 dB	4.52 dB
12 Mbps	5.00 dB	5.00 dB	5.00 dB	5.00 dB
18 Mbps	10.59 dB	10.37 dB	10.16 dB	10.00 dB
24 Mbps	12.18 dB	12.06 dB	12.01 dB	12.00 dB
36 Mbps	-	-	18.40 dB	18.22 dB
48 Mbps	-	21.77 dB	21.45 dB	21.23 dB
54 Mbps	24.59 dB	23.34 dB	22.78 dB	22.42 dB

Table III presents the minimum required SNRs in various data rate scenarios with various packet lengths. We can clearly observe that given a data rate, the minimum required SNR decreases with the packet length L. In the SNR range corresponding to 36 Mbps, we cannot find an SNR to satisfy the corresponding performance metric when L is 100 or 300. Since DSS only requires $r = 14$ that is much small compared with the total number of data symbols within a data packet, we can exploit the existing SNR gap presented in Section III to design DSS in most channel conditions.

D. Discussions

The implementation of the proposed communication scheme DSS in this work is based on IEEE 802.11a, but DSS can be implemented in other IEEE 802.11 standards (e.g., 802.11g/n/ac) because the physical layer in most IEEE 802.11 standards adopts OFDM technique. Applying DSS to different IEEE 802.11 standards does not affect the construction and detection of silent data symbols. However, the minimum SNRs required for applying DSS in different IEEE 802.11 standards are different because different modulation schemes and coding rates specified in different IEEE 802.11 standards lead to different SNR gaps. In addition, in the detailed design of DSS, we should consider the features specified in different IEEE 802.11 standards such as channel bandwidth, data frame structure, and frame aggregation. For example, the bandwidth of a channel in 802.11a/g is fixed to 20 MHz while that in 802.11n and 802.11ac can be up to 40 MHz and 80 MHz, respectively. In a channel with wider bandwidth, we can exploit more data subcarriers to design DSS.

In the WiFi network including heterogeneous stations, stations employing different 802.11 standards cannot understand each other's MAC frame headers because different 802.11 standards specify different data packet structures. In our future work, we will investigate the feasibility of communications among stations employing different 802.11 standards. Since DSS extracts the embedded information at the physical layer without packet decoding, stations employing different 802.11 standards may communicate with each other as long as they are capable of OFDM operation.

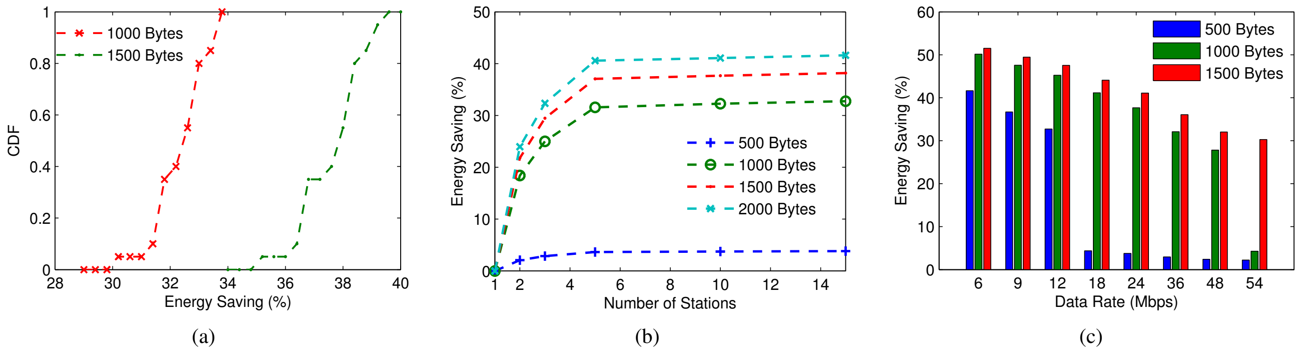
VI. Evaluation

In this section, we validate the feasibility of our DSS. Our network-level simulations are based on OPNET 14.5 to evaluate the energy saving of DSS.

A. Detection Accuracy of Silent Data Symbols

The Sora platform [25] provides a software WiFi driver, SoftWiFi, which is used to validate the feasibility of constructing and detecting of silent data symbols. We adopt the default values of IEEE 802.11a to set system parameters. The false-alarm probability (P_f) and miss-detection probability (P_m) are used to quantify the detection accuracy of silent data symbol. P_f is the probability that DSS treats a regular data symbol as a silent data symbol while P_m is the probability that DSS fails to detect a silent data symbol.

Magnitudes of data subcarriers: The inserted silent data symbols are detected by measuring the energy on each data subcarrier, which does not require demodulating data symbols. Based on the FFT result presenting the magnitudes of all data subcarriers, we can identify data subcarriers with low energy. Fig. 8 presents the normalized FFT result of 48 data subcarriers, revealing that 4 data subcarriers remain idle, while other data subcarriers have actually transmitted data symbols. Because of frequency selective fading, data subcarriers have different magnitudes. In Fig. 8, we can clearly observe the magnitude of a data subcarrier carrying a silent data symbol is much smaller than that of a data subcarrier carrying a regular



(a)

(b)

(c)

Fig. 10. (a) Cumulative distribution of energy saving of DSS in a WiFi network with 20 stations. (b) The impact of packet length and number of stations on energy saving of DSS. (c) The impact of data rate on energy saving of DSS.

data symbol. With simple threshold check on data subcarrier energy, we can clearly identify which data subcarriers carry silent data symbols. In Fig. 8, silent data symbols are detected on data subcarriers [4,11,19,21].

Impact of detection threshold: A suitable threshold is important to accurately conduct energy detection. The practical threshold for energy detection is larger than the estimated noise floor. Without interference, Fig. **??** presents the effect of detection threshold on P_f and P_m. Within the detection threshold range [-84.2dBm -77.8dBm], both P_f and P_m approach zero. In Fig. 9, we can observe that the decrease of detection threshold results in a significant increase of P_m while the increase of detection threshold results in a slight increase of P_f. Since the magnitude difference between a regular data symbol and a silent data symbol is large, P_m is more sensitive to the detection threshold than P_f. However, the detection threshold cannot be too high because regular data symbols with deep fading may be falsely regarded as silent data symbols.

B. Energy Saving of DSS

In this subsection, we present simulation results to show the energy saved by our DSS scheme. The AP has no packets to transmit while all stations have saturated traffic and employ IEEE 802.11a to contend for the channel. According to the energy profile of a typical Atheros WiFi NIC [8], the power consumption parameters of transmitting, idle, receiving and sleep in our simulations are set to 1.35 W, 0.89 W, 1.02 W and 0.16W respectively. We define the energy saving as the ratio of the energy saved by DSS to the total energy consumed by the IEEE 802.11a for channel access.

Given 20 stations in a WiFi network, Fig. 10(a) plots the energy saving of DSS. For packet length of 1000 Bytes, the energy saving ranges between 30.35% and 33.93%. For 90% of the stations, DSS saves 31.4% or more energy compared with the IEEE 802.11a, and the energy saving is around 32.6%. With longer packet length of 1500 Bytes, the energy saving falls between 35.28% and 39.50%, and DSS saves above 36.4% of energy for more than 90% of stations.

Given data rate of 24 Mbps, Fig. 10(b) shows the energy saving of DSS as a function of packet length and the number of stations. Since longer packet length allows unintended receivers to spend much more time in the sleep state, the energy saving increases with packet length. The packet length may be very long in the frame aggregation scheme that is widely used in current WiFi standards, so energy waste induced by packet overhearing becomes more serious. For packet length of 500 Bytes, unintended receivers stay in idle state rather than sleep state due to short packet transmission duration. The power consumption of idle state is comparable with that of receiving state, so we can observe a significant decrease in energy saving when packet length decreases from 1000 Bytes to 500 Bytes. For example, given 5 stations, the energy saving is 31.56% for 1000 Bytes while the energy saving is 3.64% for 500 Bytes. Because the number of successful transmissions in a WiFi network initially increases with the number of stations, unintended receivers can overhear more successful transmissions. We observe that the energy saving initially increases with the number of stations in Fig. 10(b). However, the increase of the number of stations has little effect on the energy saving after the network reaches saturated state.

Given 10 stations, Fig. 10(c) shows the impact of data rates on energy saving. With the increase of data rate, the energy saving of various packet lengths decreases. For packet length of 500 Bytes, we can observe that the energy saving drops sharply at the data rate of 18 Mbps that forces unintended receivers to stay in idle state. For packet length of 1500 Bytes, the energy saving ranges from 51.5% to 30.3% when data rate increases from 6 Mbps to 54 Mbps. Since unintended receivers switch to sleep state at all data rates, the energy saving decreases slightly.

Coexistence with the existing WiFi stations: The WiFi stations supporting DSS can coexist with the existing WiFi stations. The more WiFi stations supporting DSS in the network, the more energy can be saved. A WiFi station supporting DSS can fully utilize the DSS benefits. Since a legacy WiFi station does not take advantage of DSS to embed information, unintended receivers cannot benefit from the information exchange to reduce energy wasted by packet overhearing. We define η as the ratio of the number of WiFi stations supporting DSS to the total number of stations in the network. In the simulation, the total number of stations is always 20.

Fig. 11 plots the energy saving of DSS as a function of

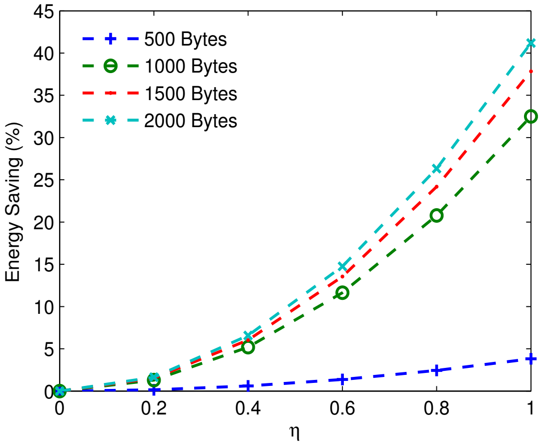


Fig. 11. The energy saving of DSS as a function of η.

η. As η increases from 0 to 1, the energy saving increases significantly because more stations can take advantage of DSS to reduce energy wasted by packet overhearing. When $\eta = 0$, DSS becomes useless and the overall network degrades to the standard WiFi network, so the corresponding energy saving is zero. When $\eta = 1$, the proposed communication scheme is fully utilized.

VII. Related Work

There is a large body of work attempting to reduce energy consumption in WiFi networks.

PSM sleep scheduling: With PSM, a station stays in the sleep state for most of the time. Efficient scheduling scheme is important to determine when it needs to wake up. Since competing background traffic affects the power performance of a station in PSM, many schemes have been proposed to isolate PSM traffic from competing background traffic. Rozner et al. [5] proposed NAPman to achieve energy-aware fair scheduling at the AP. Based on the mechanism of time division, He et al. [4] proposed scheduled PSM to divide a beacon period into multiple time slices, so a station in PSM only needs to wake up at the appointed time slices. Masri et al. [26] proposed a novel QoE-aware PSM that exploits the silence periods of a VoIP session to increase the sleeping time. Since PSM results in many multicast packet losses, Lee et al. [27] developed a practical AP-side solution to prevent stations from operating in PSM.

Creating sleep opportunities: Many works aim at creating more sleep opportunities to save energy. Liu et al. [28] presented experiments to show the existence of short idle intervals between MAC frames due to low rate applications, and they put WiFi interface into sleep state based on the prediction of the next frame arrival time. Since an VoIP call includes silence periods, Pyles et al. [29] predicted the future silence periods and switched WiFi interfaces into sleep state.

The above works do not save energy during contention periods and our work is orthogonal to these works.

PHY-assisted energy saving: If the WiFi interface works at a lower clock rate, significant energy can be saved. Zhang et al. [2] proposed E-MiLi that enables the receiver to work with a lower sampling clock rate during idle listening while ensuring accurate packet detection. Lu et al. [30] proposed SloMo that allowed transceivers to work at a lower clock rate during reception and transmission while ensuring accurate communications. Different from [2] and [30], Wang et al. [31] exploited rateless codes to allow the receiver to scale down its sampling rate without modifying legacy packets or requiring PHY redundancy. Chung et al. [32] reduced energy consumption induced by WiFi scanning using collocated Bluetooth radios. To solve packet overhearing problem, the idea of exploiting physical preamble to convey the intended receiver's address information has been used in many works [2] [9] [33]. This method adds a special preamble in front of the original packet and uses signature self-correlation detection to extract information. However, the new preamble consumes extra channel resources and the correlation process incurs extra delay of packet reception. The power consumption of this method is theoretically analyzed in [34].

SNR gap: The existence of SNR gap has been analyzed in [35] that proposed a rate compatible modulation to achieve seamless rate adaptation. The SNR gap means the receiver can tolerate additional interference. Therefore, it is used to enhance spatial reuse in ad hoc networks [36] and to let the secondary user get more opportunities to access the licensed spectrum in cognitive radio networks [37]. The idea of inserting silent data symbols to encode information is also proposed in [38]. However, [38] focuses on access coordination while DSS is proposed for energy saving. To achieve physical layer user authentication, Borle et al. [39] intentionally introduced a constant-length offset of the constellation points to embed extra information. Wang et al. [40] exploited inference signal to encode channel usage information for combating hidden and exposed station problems. These two works also exploit SNR gap to recover transmission errors. However, DSS adopts a different communication scheme and aims for a different application.

VIII. Conclusion

In WiFi networks, packet overhearing exists in the backoff freezing process of a station due to broadcast nature of the wireless channel. The station spends much energy in receiving and decoding packets that are not destined to it, which may significantly reduce battery lifetime. Our energy model for IEEE 802.11 DCF demonstrates that packet overhearing is a major source of energy consumption of WiFi interfaces. To address this issue, we design DSS that exploits the SNR gap existed in WiFi communications to embed lightweight information into the front part of a data packet at the physical layer. With DSS, unintended receivers quickly extract information at the physical layer without receiving and decoding the whole data packet, so they can stop useless packet receptions to save energy. Our simulation results show that DSS provides significant energy saving over the IEEE 802.11 systems.

Acknowledgment

This work was supported by the Natural Science Foundation of China (NSFC) under Grants 61871362 and 61201240. The work of H. Ding and Y. Fang was partially supported by National Science Foundation under CNS-1717736 and CNS-1409797.

References

[1] B. Feng, C. Zhang, H. Ding, and Y. Fang, "Phycast: Towards energy efficient packet overhearing in wifi networks," in *Proc. IEEE ICC*, 2018.

[2] X. Zhang and K. G. Shin, "E-mili: energy-minimizing idle listening in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1441–1454, 2012.

[3] L. Sun, R. K. Sheshadri, W. Zheng, and D. Koutsonikolas, "Modeling wifi active power/energy consumption in smartphones," in *Proc. IEEE ICDCS*, 2014, pp. 41–51.

[4] Y. He and R. Yuan, "A novel scheduled power saving mechanism for 802.11 wireless lans," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1368–1383, 2009.

[5] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "Napman: network-assisted power management for wifi devices," in *Proc. ACM MobiSys*, 2010, pp. 91–106.

[6] J. Wang, Y. Fang, and D. Wu, "A power-saving multi-radio multi-channel mac protocol for wireless local area networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[7] Z. Zeng, Y. Gao, and P. Kumar, "Sofa: A sleep-optimal fair-attention scheduler for the power-saving mode of wlans," in *Proc. IEEE ICDCS*, 2011, pp. 87–98.

[8] B. Balaji, B. R. Tamma, and B. Manoj, "A novel power saving strategy for greening ieee 802.11 based wireless networks," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.

[9] S. Sen, R. R. Choudhury, and B. Radunovic, "Phy-assisted energy management for mobile devices," *ACM MobiSys Poster Session*, 2010.

[10] S. Biswas and S. Datta, "Reducing overhearing energy in 802.11 networks by low-power interface idling," in *Proc. IEEE PCC*, 2004, pp. 695–700.

[11] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, 2000, pp. 585–594.

[12] M. Ergen and P. Varaiya, "Decomposition of energy consumption in ieee 802.11," in *Proc. IEEE ICC*, 2007, pp. 403–408.

[13] W.-K. Kuo, "Energy efficiency modeling for ieee 802.11 dcf system without retry limits," *Computer communications,Elsevier*, vol. 30, no. 4, pp. 856–862, 2007.

[14] H. Zhai, Y. Kwon, and Y. Fang, "Performance analysis of ieee 802.11 mac protocols in wireless lans," *Wireless communications and mobile computing, Wiley Online Library*, vol. 4, no. 8, pp. 917–931, 2004.

[15] 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 Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3256–3266, 2011.

[16] 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.*, no. 3, pp. 588–601, 2007.

[17] Q. Zhao, D. H. Tsang, and T. Sakurai, "A simple and approximate model for nonsaturated ieee 802.11 dcf," *IEEE Trans. Mobile Comput.*, vol. 8, no. 11, pp. 1539–1553, 2009.

[18] L.-L. Yang and L. Hanzo, "Performance analysis of coded m-ary orthogonal signaling using errors-and-erasures decoding over frequency-selective fading channels," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 2, pp. 211–221, 2001.

[19] J. Geist and J. Cain, "Viterbi decoder performance in gaussian noise and periodic erasure bursts," *IEEE Trans. Commun.*, vol. 28, no. 8, pp. 1417–1422, 1980.

[20] T. Li, W. H. Mow, and M. Siu, "Joint erasure marking and viterbi decoding algorithm for unknown impulsive noise channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, 2008.

[21] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, 1997.

[22] A. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 751–772, 1971.

[23] J. C. Moreira and P. G. Farrell, *Essentials of error-control coding*. John Wiley & Sons, 2006.

[24] D. Haccoun and G. Bégin, "High-rate punctured convolutional codes for viterbi and sequential decoding," *IEEE Trans. Commun.*, vol. 37, no. 11, pp. 1113–1125, 1989.

[25] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, "Sora: high performance software radio using general purpose multi-core processors," in *Proc. USENIX NSDI*, 2009, pp. 75–90.

[26] M. A. Al Masri, A. B. Sesay, and F. M. Ghannouchi, "Energy/quality-of-experience tradeoff of power saving modes for voice over ip services," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4009–4022, 2017.

[27] G. Lee, Y. Shin, J. Koo, J. Choi, and S. Choi, "Act-ap: Activator access point for multicast over wlan," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

[28] J. Liu and L. Zhong, "Micro power management of active 802.11 interfaces," in *Proc. ACM MobiSys*, 2008, pp. 146–159.

[29] A. J. Pyles, Z. Ren, G. Zhou, and X. Liu, "Sifi: exploiting voip silence for wifi energy savings insmart phones," in *Proc. ACM UbiComp*, 2011, pp. 325–334.

[30] F. Lu, G. M. Voelker, and A. C. Snoeren, "Slomo: Downclocking wifi communication." in *Proc. NSDI*, 2013, pp. 255–268.

[31] W. Wang, Y. Chen, L. Wang, and Q. Zhang, "Sampleless wi-fi: Bringing low power to wi-fi communications," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1663–1672, 2017.

[32] J. Chung, J. Park, C.-k. Kim, and J. Choi, "C-scan: Wi-fi scan offloading via collocated low-power radios," *IEEE Internet of Things Journal*, 2018.

[33] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proc. ACM SenSys*, 2006, pp. 307–320.

[34] L. Feng and J. Yang, "A novel analysis of delay and power consumption for polling with phy-assisted power management," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3610–3620, 2018.

[35] H. Cui, C. Luo, J. Wu, C. W. Chen, and F. Wu, "Compressive coded modulation for seamless rate adaptation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4892–4904, 2013.

[36] T.-S. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *Proc. ACM MobiCom*, 2006, pp. 366–377.

[37] Y. Yang and S. Aissa, "Achievable data rate in spectrum-sharing channels with variable-rate variable-power primary users," *IEEE Wireless Commun. Lett*, vol. 1, no. 4, pp. 312–315, 2012.

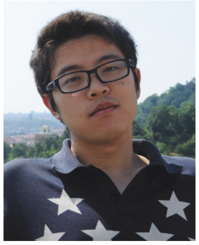
[38] B. Feng, C. Zhang, J. Liu, and Y. Fang, "Turning waste into wealth: Free control message transmissions in indoor wifi networks," *IEEE Trans. Mobile Comput.*, submitted for publication.

[39] K. M. Borle, B. Chen, and W. Du, "A physical layer authentication scheme for countering primary user emulation attack," in *Proc. IEEE ICASSP*, 2013, pp. 2935–2939.

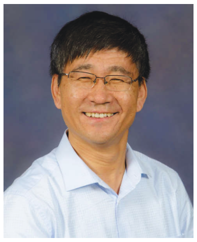
[40] L. Wang, K. Wu, and M. Hamdi, "Combating hidden and exposed terminal problems in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 4204–4213, 2012.

Bing Feng is currently a Ph.D. student at the University of Science and Technology of China. He received the B.E. degrees from the Anhui University, Hefei, China, in 2013. His research interests include wireless communication and wireless local area networks.

Chi Zhang received the B.E. and M.E. degrees in electrical and information engineering from the Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida in 2011. He joined the School of Information Science and Technology, University of Science and Technology of China, as an Associate Professor in 2011. His research interests include the areas of network protocol design and performance analysis and network security particularly for wireless networks and social networks.



Haichuan Ding is currently a Ph.D. student at the University of Florida. He received the B.Eng. and M.S. degrees in electrical engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2011 and 2014, respectively. From 2012 to 2014, he was with the Department of Electrical and Computer Engineering, the University of Macau, as a visiting student. His current research is focused on cognitive radio networks, vehicular networks, and security and privacy in distributed systems.



Yuguang ”Michael” Fang (F’08) received an MS degree from Qufu Normal University, Shandong, China in 1987, a PhD degree from Case Western Reserve University in 1994 and a PhD degree from Boston University in 1997. He joined Department of Electrical and Computer Engineering at University of Florida in 2000 and has been a full professor since 2005. He held a University of Florida Research Foundation (UFRF) Professorship (2006-2009), a Changjiang Scholar Chair Professorship (Xidian University, Xi’an, China, 2008-2011; Dalian Maritime University, Dalian, China, 2015-present), and a Guest Chair Professorship with Tsinghua University, China (2009-2012). Dr. Fang received the US National Science Foundation Career Award in 2001, the Office of Naval Research Young Investigator Award in 2002, the 2015 IEEE Communications Society CISTC Technical Recognition Award, the 2014 IEEE Communications Society WTC Recognition Award, the Best Paper Award from IEEE ICNP (2006), and the 2010-2011 UF Doctoral Dissertation Advisor/Mentoring Award. He is the Editor-in-Chief of IEEE Transactions on Vehicular Technology, was the Editor-in-Chief of IEEE Wireless Communications (2009-2012), serves/served on several editorial boards of technical journals. He is a fellow of the IEEE and the AAAS.